ACTA IMEKO
ISSN: 2221-870X
March 2018, Volume 7, Number 1, 57-62

A Novel Design of an Augmented Reality Based Navigation System & its Industrial Applications

Timotei István Erdei, Zsolt Molnár, Nwachukwu C. Obinna, Géza Husi

University of Debrecen, Electrical Engineering and Mechatronics Department H-4028 Debrecen, Ótemető street 2-4

ABSTRACT
This paper proposes a design of an augmented reality based navigation system, as well as investigate its potential areas of application within the industry. With the ascent of Industry 4.0 (IoT), systems such as Augmented Reality and Virtual Reality benefit from the availability of bulk sensory data.

Section: RESEARCH PAPER

Keywords: augmented reality; virtual reality; automated guided vehicle; KUKA robotics

Citation: Timotei István Erdei, Zsolt Molnár, Nwachukwu C. Obinna, Géza Husi, A Novel Design of an Augmented Reality Based Navigation System & its Industrial Applications, Acta IMEKO, vol. 7, no. 1, article 10, March 2018, identifier: IMEKO-ACTA-07 (2018)-01-10

Section Editor: Lorenzo Ciani, University of Florence, Italy

Received December 12, 2017; **In final form** January 16, 2018; **Published** March 2018

Funding: The project was supported by University of Debrecen, Electrical Engineering and Mechatronics Department

Corresponding author: Timotei István Erdei, e-mail: timoteierdei@gmail.com

1. INTRODUCTION

Maintenance activities are integral within an industrial setting. The efficiency of these activities is associated with the total productivity of an industrial process/machine. A highly efficient maintenance policy/strategy usually results in relatively high levels of plant productivity. Augmented Reality AR and Virtual Reality VR may be incorporated into a maintenance strategy. AR and VR technology would enhance maintenance activities and facilitate somewhat complex tasks.

A maintenance strategy includes a range of activities and may be categorized into administrative, technical and management processes. AR technology contains digital data, as well as other technical details, and is able to provide information about industrial machinery-equipment, without the need for equipment disassembling.

In this regard, we employed AR technology in developing a unique navigation system to replace/reduce the installation costs of traditional Automatic Guided Vehicles (AGV) navigation systems. The proposed AR system consists of a camera, which observes the QR-Code/Markers, and a processing unit.

Augmented Reality enables visualisation of any data and information, as well as control of a running process. It means it is possible to read various data within any equipment, during its operation and in real-time. This facilitates analysis of "black box" systems.

2. AR IMAGE PROCESSING - OPENCV

Our main goal was developing a Cyber-Physical System (CPS) for an industrial robot laboratory and find ways of implementing Internet of Things (IoT).

The department robotic laboratory includes a KUKA KR5 medium payload industrial welding robot, a KUKA KR3, FANUC spider and a Sony Scara SRX-611 [1].

The data and parameters have been encoded into QR-Codes and other unique markers. In the first test we use these codes as position markers and an instruction set to control a prototype AGV robot. The AGV uses an IP camera as vision sensors.

The Building Mechatronic Research Centre features several functional IP cameras, which were interfaced to the CPS system.

A computer serves as the CPS system processing unit. The computer has an Intel Core i7 CPU, 16 GB DDR3 RAM, 2 TB HDD & 2 × NVIDIA GeForce GTX 650 Ti. On the other hand we chose OpenCV, which is a vision library software with numerous algorithms, for image processing [1].

In order to implement the AR algorithm, a special plugin/library known as ARma was required.

AR gives an informative view of the world, and the main parts include a camera, a QR-Code [2] with hidden information and an image processing software/hardware to decode a QR-Code and detect the position of the code in real-time (Figure 1).

The proposed AR system is required to track the motion of a point in a predetermined 3D coordinate system. Motion tracking involves the measurement of an object velocity and orientation (Figure 2).

This object tracking method is a rigorous one, because first you need to design the 3D model in a 3D CAD software, and after it you need to decrease the polygon number of the 3D model. Although this has to be done with circumspection. In our case we used two Nvidia GeForce GTX 650 Ti graphics cards. The main parameters for these graphics cards are the maximum polygon number and maximum image frame drawings per second. Regarding the functioning of the graphics card, it is worth to recall that computer-generated images are made up of triangles. For example, a square will be made up by two quadrant triangles. In case of complex geometrical forms, like a sphere or curved surfaces, these triangles are so small and dense, that in appearance they do not bother the user, in other words, they cannot be seen. The bigger the number of the triangles, the smaller they are, and the better the quality of our 3D model: this is a requirement to be met in order to outdraw competitors.

The goal is to have 60 picture refreshments per second, this is a value that cannot be noticed by the human eye, and it gives constant movement in case of higher speeds as well.

In the case of the graphics card, the performance can also be characterized in triangle/seconds. This means the triangle (polygon) number of a given picture is reduced by 1, the polygon drawings will be reduced by 60 each second. In case of bigger numbers, this implies a significant lessening of demand for the calculations.

As we can see in Figure 3, in the beginning we tried running the system with 50000 polygons, which showed a nice graphic, but because of the dynamic properties of the model, this graphic was useless, since the refreshing of the image dropped to 25 pictures/second. With the polygon number being dropped drastically to 1000 polygons, the model became graphically useless. On the other hand, by turning off the limitations of the graphics card (V-sync) the numerical accomplishment surpassed the value of 2300 picture refreshments.

By running more tests, the optimal polygon number came to be 25000. With this number, the 3D model is suitable and a stable 60 picture refreshments per second can also be attained and it permits other objects to be loaded beside the robot arm.

High polygon number values are computationally expensive. We designed all 3D models in 3D CAD software, and after that, we decreased the numbers of the polygons to make our program run more efficiently [3].

The next step of our research was designing the AGV prototype. The AR AGV navigation system would include trajectory planning and tracking, as well as collision avoidance. The most important characteristic of the robot is that there are no sensors on the robot itself, but it uses only the IP cameras of the robotics laboratory for trajectory tracking.

First, we designed the AGV concept in a 3D CAD modelling software and made a part list with the necessary equipment [4]. This 3D model is shown by Figure 4.

The vehicle guide part is composed of an Arduino Nano panel. This makes up the required PWM signal for the servo motors and communicates with the ESP Wi-Fi module through communication lines.

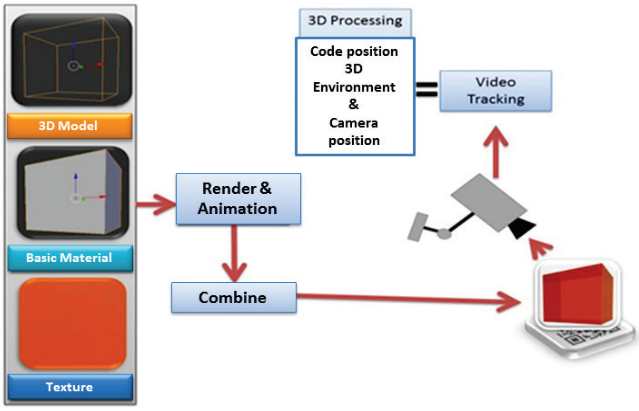

Figure 1. ARma and OpenCV – working.

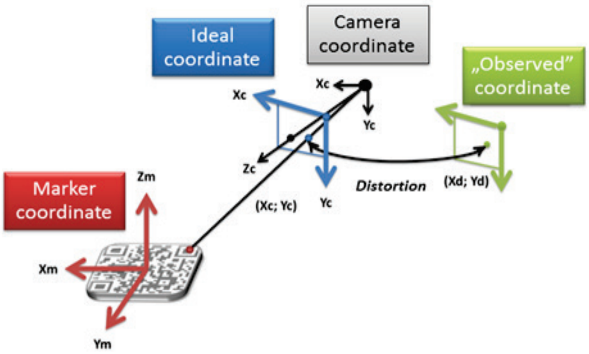

Figure 2. Camera view & position detection of object.

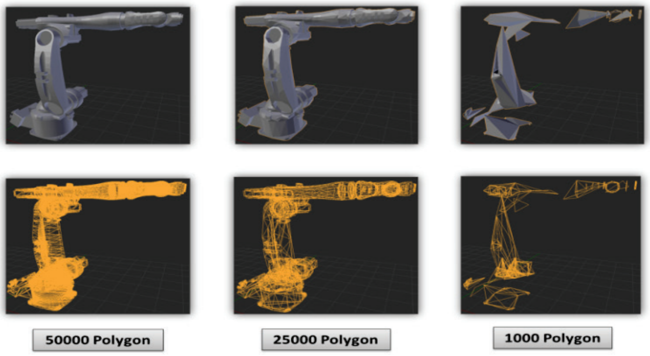

Figure 3. Polygon reduction.

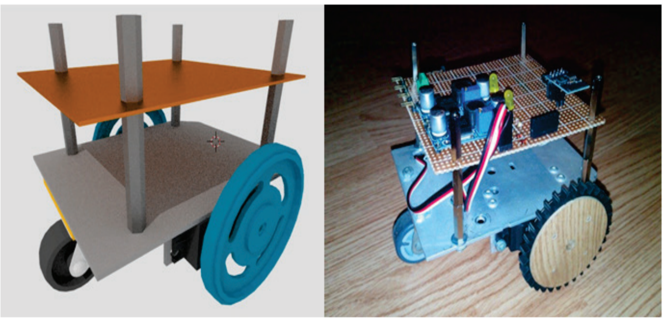

Figure 4. AGV prototype robot; left panel: 3D model; right panel: picture.

In addition, two continuous rotating servo motors make the shell of the vehicle move.

The communications are ensured by the ESP Wi-Fi module which, after the power-supply voltage is supplied, automatically connects to the Wi-Fi router and transmits the received commands to the Arduino Nano Panel.

Besides this, on the AVG a Li-Ion accumulator with two cells was placed.

Two DC-DC step-down converters provide voltage stabilization.

The step-down converters have high power conversion rate, even 90% efficiency is reachable, which is why they were chosen.

The block diagram of the AGV robot is shown on Figure 5.

The core of the AGV prototype robot is an Arduino Nano, two LM2596 DC step down converter and a Wi-Fi-Serial module.

The weight of the robot with batteries is 483 grams. Two Li-Ion batteries were used for power supply, each providing 3.6 V voltage and 3600 mAh capacity.

Two rotating servomotors [5] were used for drive. They can deliver 0.27 Nm torque at 6 V power. The maximum current consumption is of 200 mA each.

The high battery capacity and the low power consumption of the AGV prototype robot make possible a long battery life. During the tests, depending on the operating speed and load, the continuous operation was 8 to 10 hours.

3. ROBOT LABORATORY AS AR ISOLATED ENVIRONMENT

The Robot Laboratory of the Building Mechatronic Research Center serves as an AR isolated environment, and it is a part of the building surveillance and security system network. Furthermore, we can use the installed IP cameras as vision sensor using a desktop pc which can store the recordings and which is powerful enough to make CUDA based image processing/decoding [6].

Thus, it is possible to use QR-Code/Markers placed in the field of vision of the cameras and thus making possible to identify their positions. Once the identification is done, we can add other commands to the QR-Code/Markers to perform tasks.

The diagram in Figure 6 shows the structure of the Robot Laboratory in a control task. The system consists of IP cameras and their QR-Code and the managed device. Images of IP cameras are processed by the central high performance computer.

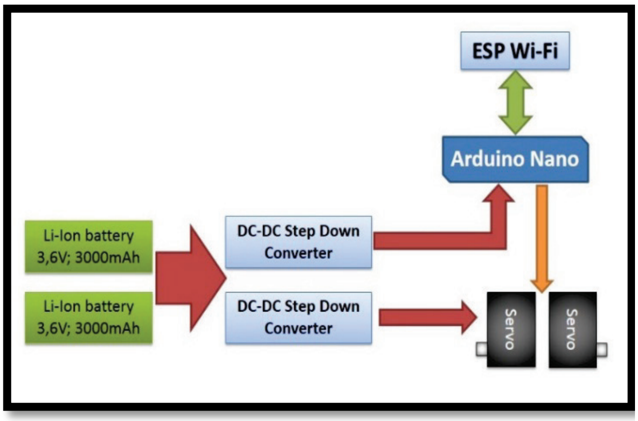
ESP Wi-Fi
Arduino Nano
Li-Ion battery 3,6V; 3000mAh
Li-Ion battery 3,6V; 3000mAh
DC-DC Step Down Converter
DC-DC Step Down Converter
Servo
Servo

Figure 5. AGV robot block diagram.

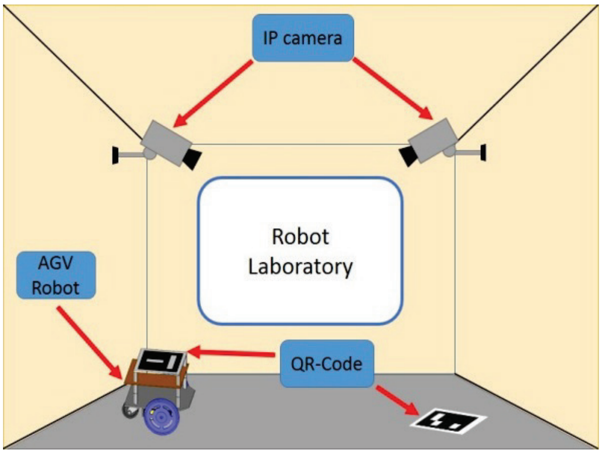

Figure 6. AR isolated environment.

On the controlled device, in our case the AGV robot, no detector is found: its position is determined by the cameras in the Robot Laboratory. IP cameras are available on both floors of the entire Building Research Centre, whose images are available from the central computer and processed. This allows the AGV robot to operate in any part of the building, if necessary [7].

The IP cameras used in the Lab are type TCIP-LPro213WDRMDN and have the following parameters:

- Sony 1/3, 1.3 Mpx. Progressive Exmor CMOS sensor
- 1280x1024 (SXGA); H.264, MPEG4, M-JPEG
- Lux (Automatic Infrared LED Lighting below 2 Lux)
- 2D noise reduction and sensitivity enhancement
- Motion Detection and Alarm
- Two-way audio and multicast
- Built-in Aspherical Day / Night Lens (f3.8 9.5 mm)
- 12 V supply voltage [8].

The cameras characteristics made them suitable for the controlling of the AGV robot. Even though, this also comes with certain setbacks.

One of these is connected with the automatic light adjustment. In the case of the cameras used it works like this: it uses all of the light power of the acquired image to compute a value, which is then taken as reference for further adjustments.

The other set-back comes from the used lens. They give a wider point of view, but in the same time distort the edges of the image.

After the designing and building of the AGV prototype robot, we needed to test the system detection stability.

In that aspect we analysed the detection ability of all installed IP cameras in the Robot Laboratory. In this context we made measurements of the illumination which is the most important thing in detection.

The graph below illustrates the measurable illumination in the centre of the Lab depending on the number of lights on.

The measurements have shown (Figure 7) that there must be at least 200 lux above the exposure of the area, so that the TCIP-LPro213WDRMDN camera can detect the codes. The camera can see lower brightness but the image is not clear enough to recognize the codes. In addition, in the absence of illumination, the camera will automatically turn on the infrared LED. The infrared light diffracts from different surfaces differently from to the visible light, giving simply a black and

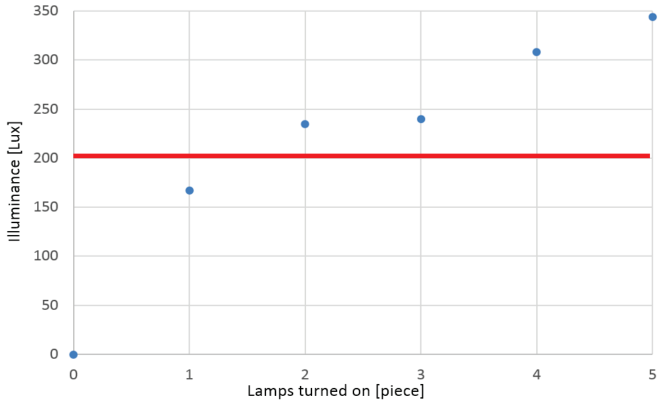

Figure 7. Illumination of the centre of the Robot Laboratory.

white image, making it unsuitable for code recognition.

4. AR BASED NAVIGATION SYSTEM

In the first part of our project we found a suitable place to test the system which was the robotics laboratory in the Building Mechatronic Research Centre. Its dimensions are: width: 4.3 m; length: 6.5 m; height: 2.8 m.

We designed the QR-Codes/Markers and printed them out in different sizes and placed them in the right order to make the path for the robot.

The task of the AGV prototype robot is to leave the “Start” position, go to the 1st position and wait until the KUKA KR5 robotic arm goes to the position of the AGV and puts on it the aluminium cube (Figure 8). After it reach the 2nd, 3rd 4rd, 5rd position and the Home position with the cargo. We printed a code for identifying the AGV’s orientation.

The distance between the positions, the time required to make the distance and the average speed are calculated in Table 1.

Distances have been determined by the software during route planning. The recording of time was determined by the time elapsed between touching the two targets.

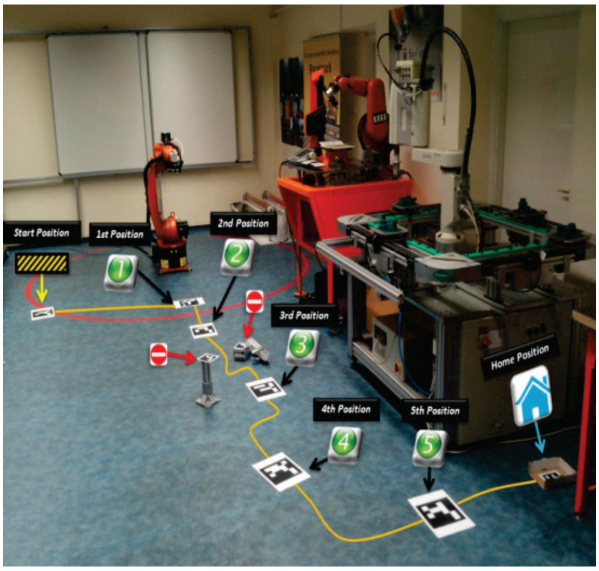

Figure 8. Robot Laboratory as AR environment.

Table 1. Distance, time, speed values.

Path	Distance [cm]	Time [s]	Speed [cm/s]
Start – 1	94	41.5	2.27
1 – 2	46	28.6	1.6
2 – 3	87	67	1.3
3 – 4	75	30.2	2.48
4 – 5	98	40.8	2.4
5 – Home	53	24.7	2.15

There are differences between average speed values. The reason is that for the straight paths, the applied algorithm simply determines the path and the robot moves without rotations. For paths with obstructions such as the 2-3 route, the algorithm bypasses the obstacles, determining where the robot needs to rotate, which takes time and decreases the average speed.

The IP cameras of Robot Laboratory see the codes and stream data to the notebook via Wi-Fi. On the desktop computer, OpenCV and ARma recognize the position of the code and decode it, send the information to the AGV prototype robot which plans and tracks the trajectory.

The algorithm used to design the route is the Rapidly-Exploring Random Tree (RRT), originally written for humanoid robots to design motion. The RRT algorithm has advantage in many cases, providing a solution for 2D and 3D route planning.

The operation of RRT can easily be described. The algorithm grows from the initial configuration to "root" as a tree in the search space to detect the target configuration. If it cannot find a viable route, it will continue to search in the open space, while taking into account the criteria (barriers, limits, etc.). The size of the "roots" can be determined by the growth factor. The RRT algorithm will continue to search until it finds the optimal path between the start and destination configurations.

In the first test we launched/programmed the KUKA KR5 industrial robot manipulator with our electronic PCB, the TiMo board, to control the robot gripper. Furthermore, we established a new Wi-Fi network with a TP-Link router to make the Wi-Fi signal stronger in the Robot Laboratory. This was very important because we sent all signals trough Wi-Fi and we wanted to avoid the lag of the stream of the mobile phone.

The structure of the communication network used for testing is shown on Figure 9.

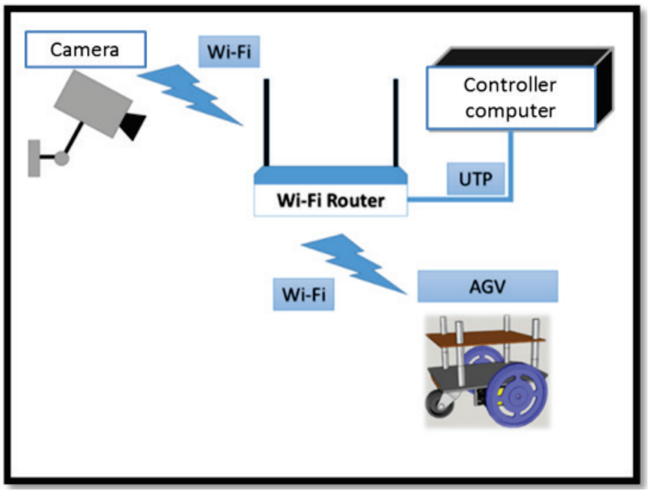

Figure 9. Network communication system.

For the first time we noticed a lag in the Desktop computer. Fortunately, we could set the dedicated Nvidia graphic card to calculate instead of the core i7 CPU which was slower in this instance.

When we began the first test, the AGV prototype robot was in the "Start" position, then it started to go to the 1st position and waited there until the KUKA KR5 moved to the AGV position and opened the gripper to put the aluminium cube on the AGV. The next step was to reach the 3rd position marker and avoid the collisions with the other objects. To realize this we made codes with other 3D models and we set the avoidance distance in the ARma software. Knowing this data, the AGV could avoid them during the motion (Figure 10). When the AGV reached the HOME position it stopped.

The project task required the re-creation of the KUKA KR5 gripping system to be suitable for material handling. To the end a Japanese, Humphrey H040-4E2 type pre-driven electro-pneumatic bistable valve has been selected, which runs from 2 to 7 bar and 24 volts [9]. The control valve was installed into a gripper GRIP GmbH. To control the gripper, we used our own self-developed TiMO Board controller system, which is suitable for controlling industrial pneumatic/hydraulic valves and has already been proven to control smaller production line cells [10].

TiMo Board was designed for Industry 4.0 / IoT tasks. The other part of the TiMo board is a Raspberry Pi B+ minimized desktop computer [11] and the Timotei-Robotics Linux distribution Os which was installed on it. The structure of the programmable control system is shown on Figure 11.

When programming the control panel (TiMo Board), the program was written on the Timotei-Robotics Os, in standard PLC ladder diagram. The TiMo board control panel has been installed on the pneumatic gripping electric circuit of the KUKA KR5 industrial robot.

5. AR APPLICATION AREAS & DEVELOPMENT

In the first part of our project we tested the AR based navigation system successfully, but there are many other ways to use QR-Codes and AR technology in development or in education.

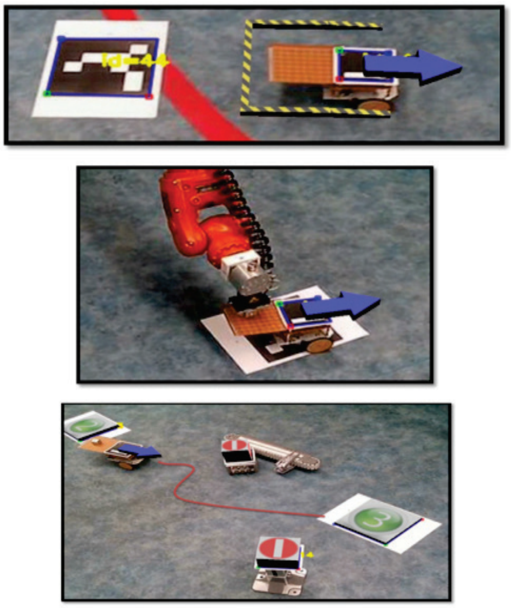

Figure 10. AGV Prototype first test.

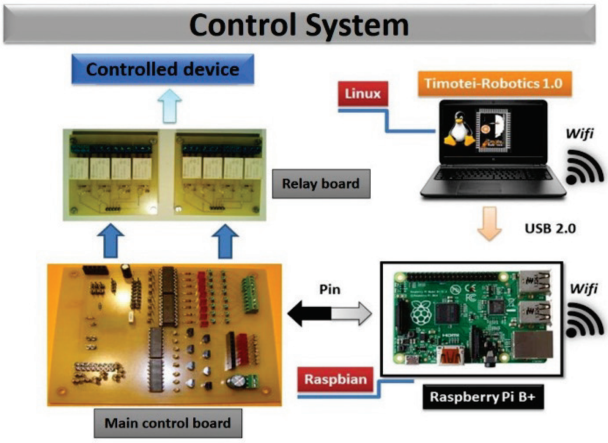

Figure 11. Programmable Control System – TiMo Board.

In the industrial robotics laboratory we installed robots of various types. Maintenance of all these robot is relatively difficult, because we need to register all of them. We collected all the datasheets, uploaded to our private network system and embedded them into QR-Codes. The core of the private network system is a Raspberry Pi [12] mini desktop computer, which has a Raspbian Linux distribution system and a 64GB SDHC card. When someone needs information about a robot in the robot laboratory he has to decode the QR-Code and in that moment we receive a notification that someone decoded one of our dynamic QR-Code, so that we can see on a diagram which code have been decoded (Figure 12).

On the other hand, we can use the QR-Codes as AR markers in order to visualize the axes positions of KUKA KR5 or set dynamically its position in the workspace (Figure 13).

The AR technology is also able to test standalone systems and items. We developed a CAD gripper for the KUKA KR5, we reduced the polygon number of it and mounted "virtually"

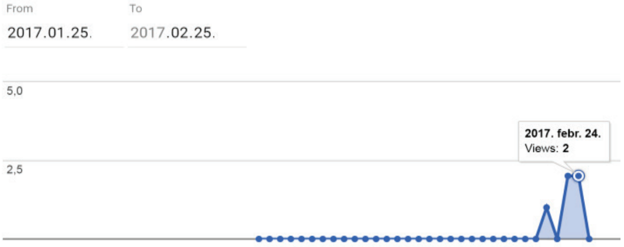

Figure 12. QR-Code decoding - monitoring.

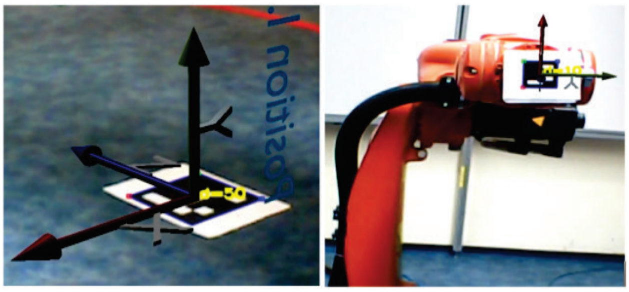

Figure 13. AR code to set position & axis visualization.

on to the robot arm.

Furthermore, the codes can be used to establish interactive connections between each other.

In one of our tests we used another marker which consists of a column 3D model (Figure 14).

When we moved the real KUKA KR3 with the virtual mounted gripper**,** it could push away the virtual column (Figure 15).

6. CONCLUSIONS

The Augmented Reality technique is potentially effective because we can set up a firewall system protecting information embedded in the codes. Furthermore**,** we can encrypt commands for robots as we did to command a prototype AGV. Moreover, we can upgrade the robots virtually and perform tests for collision avoidance, gripper replacement testing, etc.

During the tests we used GPU image processing because the integrated VGA was not fast enough for calculations.

Unfortunately, we can clearly say for these tests we had to use a relatively fast processing computer because it is computationally expensive.

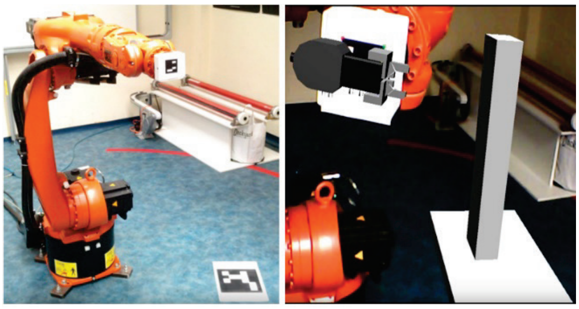

Figure 14. Virtual AR items.

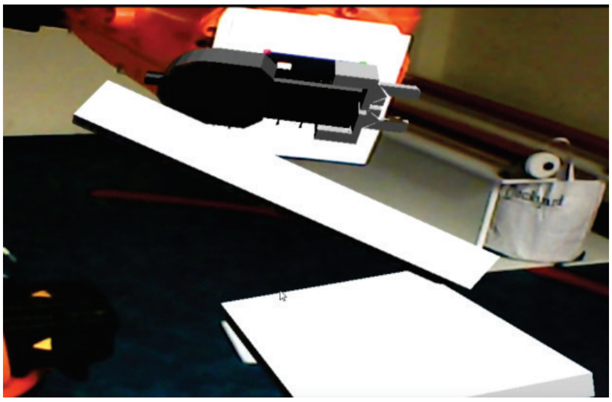

Figure 15. Virtual collision.

The system that we created does not only recognize QR Code in read-only information, but can also recognize these codes as so-called Markers that can represent the status/position of an AGV robot or the axes of robot arms. This has enabled the control of AGV robot and the attachment of virtual AR devices to physical machines.

The AR based navigation system of the AGV is fully autonomous so that no human intervention is required.

The AR technology requires huge knowledge in 3D modelling, polygon reduction and image processing.

Further improvements are underway to enhance these areas of application.

ACKNOWLEDGEMENT

The work is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund.

REFERENCES

[1] N.C. Obinna, T.I Erdei, Zs. Molnár, G. Husi, LabVIEW Motion Planning and Tracking of an Industrial Robotic Manipulator (KUKA KR5 arc): Design, Modelling, and Simulating the Robot's Controller Unit - Electrical Engineering and Mechatronics Conference, 2017.03.12.

[2] Mauricio Marengoni, High Level Computer Vision Using OpenCV, Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI.

[3] J. Rubinstein, Signal Delay in RC Tree Networks, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1983.

[4] T.I. Erdei, Zs. Molnár, N.C. Obinna, T. Gönczi, Industrial KUKA Robot Manipulator in Simulation Environment and Position Read-back, 1st International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS 2016).

[5] Zs. Molnár, G. Husi, Prototípus alkatrészek elkészítésére kifejlesztett CNC távvezérlése és távfelügyelete, 10.17667/riim.2017.si/2.

[6] Parallax servo motor parameters official manual, https://www.parallax.com/product/900-00008 2017.04.14. – 15:35.

[7] N. Zlatanov, CUDA and GPU Acceleration of Image Processing, IEEE Computer Society, 2016.03.

[8] TCIP-LPro213WD-Offical Manual {offline} 2017.02.24., 18:42).

[9] N.C. Obinna, T.I Erdei, Zs. Molnár, G. Husi, Surveillance and Security System in the Building Mechatronics Research Center - Conference: 1st International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS 2016).

[10] Humphrey H040-4E2 – https://www.asseta.com/parts/humphrey-h040-4e2 2017.03.30. – 16:34.

[11] T.I. Erdei, Zs. Molnár, G. Husi, Saját fejlesztésű vezérlőrendszer Open-Source alapokon - IJEMS.2016.2.13.

[12] E. Upton, Raspberry Pi - IEEE Computer Society 31.10.2013, 10.1109/MC.2013.349.