Application of distance measuring with Matlab/Simulink

Mircea Coman[1], Sergiu-Dan Stan[1], Milos Manic[2], Radu Balan[1]
[1]Dept. of Mechatronics, Technical University of Cluj-Napoca, Cluj-Napoca, Romania
[2] University of Idaho, Department of Computer Science
mtr_lestat@yahoo.com, sergiustan@.ieee.org, misko@ieee.org, radubalan@yahoo.com

***Abstract*- The paper presents the implementation in the Matlab/Simulink environment of an application for measuring distances using a video camera. Some of the advantages of using image processing as a method of measurement and of the Matlab for designing de application. The principles that where use to obtain de distance calculated where presented. Also the steps of the application implementation in Simulink where described.**

***Keywords*: distance measurements, Matlab, image processing**

I. INTRODUCTION

THE use of optical sensors in controlled systems gives a high level of flexibility. With the aid of optical sensor static or in movement objects can be observed. From the images obtained by a video camera the geometry and speed of objects analyzed can be obtained. In the case of using this type of sensors, in environmental analysis process, the operations can be classified in two categories of lower and higher level. Lower level operations involve operations that are designed to capture and represent of images, to correct the image brightness, noise removal, edge detection.

High level operations are using more than knowledge of the objects studied or of the environment. As a result of high-level operations implementation the system obtains information about the geometry of observed objects, their position, and their speed or can be identified according to a previously created database.

The implementation of image processing systems can be achieved within a numerous areas without varying much the type of equipment used, in some cases, changing only the processing algorithm. By using video cameras, objects can be studied without influencing their structure or behaviour, a crucial property for the measurement systems.

The same algorithm, as in the case of systems equipped with normal cameras, can be used if the information entered into the system is not part of visible spectrum, such an example is the use of camera that processes the thermal radiation.

II. MATLAB, IMAGE ACQUISITION TOOLBOX

Many systems that implement image processing algorithms use programs focused solely on that direction or high level programming environments to meet the large volume of information that is provided by video cameras. This paper presents the implementation in Matlab development environment.

This approach allows, because the program capabilities, communication with other systems to control them or for adding additional information to the computational algorithm.

Matlab environment offers several options for implementing image processing algorithms. These algorithms can be develop using the Simulink environment in which to design the application a block diagram interface is used.

Matlab also offers for the developers the option of application implementation using the M code programming language, or alternatively, a combination of the two methods can be used. Additional to those capabilities that make it advisable to operate with a variety of signals, the program has designated in Matlab Simulink two toolboxes dedicated for operation with signals containing images or video data.

To obtain one video signal in Simulink the “From the video device” block will be used. With this block the user will get a video signal in real time, from cameras connected to the computer. The advantage of this approach is that it gives you immediate access to information that is collected by video cameras connected to the computer. In this block the developer can choose which of the video cameras connected to the computer will be use to obtain images. It is possible to simultaneously capture and process signals from multiple video cameras, this is essential when estimating the distance in the case of using stereo vision system or even for more than two cameras system. In this block it is possible to set the output signal type, and also to choose the output signal resolution (with the upper limit depending on video camera characteristics).

Also it can choose between the RGB or Y, Cb, Cr formats, according to the preferences or needs of application developers.

Block sample time will determine the frequency with which the block can to generate video signal. If we are to isolate a certain portion of the images taken, that region can be selected from the block only by setting regions limits and, as a result, a resulting matrix will be transmitted to the output port thus reducing the amount of data to be processed and the number of blocks in the model. Another feature that can be set in this block is the precision with which light intensity will be represented.

You can also choose whether the three output signals, when using the standard RGB, will use three output ports from the block or a single port which will provide these data in a single signal that will contain all three signals multiplexed.

Also, for verification is provided the opportunity to view the images taken by the video camera, without needing to be run the application.

In figure 1 is presented the graphical interface of the block mentioned above.

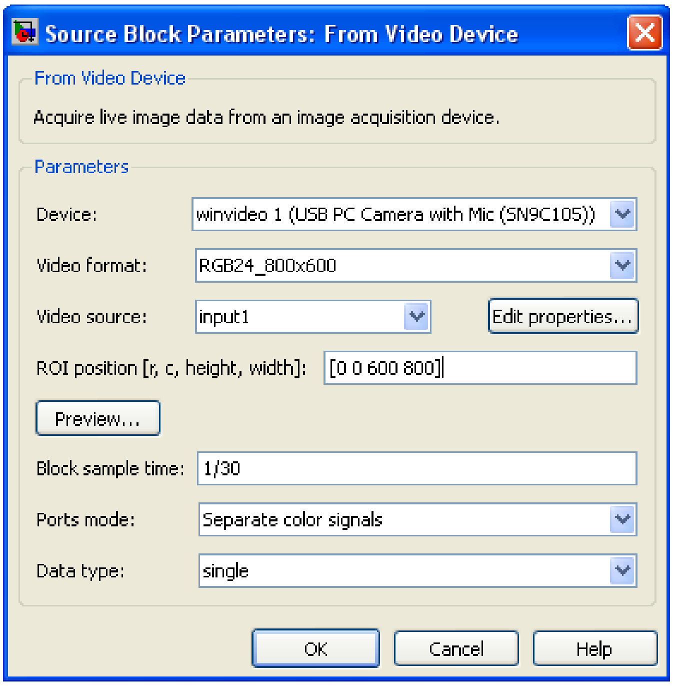

Figure. 1. “From the video device”

III. IMPLEMENTATION OF A DISTANCE MEASUREMENT APPLICATION

In the application presented in this paper, the model identifies and measures a surface to determine the distance between the object and the video camera.

The surface has a different colour to the background for easily separation of the object studied.

This choice was made to reduce errors that occur during measurement.

To reduce the volume of information processed, the signal obtained from the video camera is converted in black and white signal that contains only data about the pixel brightness, regardless of their colour.

To adjust the light intensity of the whole image a gain block was used for amplification / reduction in brightness. This value can be controlled interactively with a slider during the runtime of the application.

The detection of surface to be measured will be done by the next block.

This block takes as input a signal containing information on the image and a limit that will be used to determine the threshold that help to delimit the black area measured.

The following figure (Fig. 2) shows the subsystem that performs the separation of the surface measured from the rest of the image.

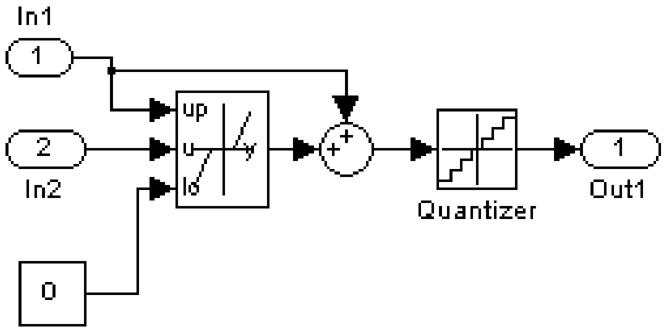

Figure 2. Separation of black regions subsystem

The signal thus obtained is sent to two viewing blocks, one that will show only an image of the surface resulting in the identification pursued, and the second block will show the original image taken by the camera but having the identified area highlighted by green.

The second view, which has the highlighted area, is particularly useful during the calibration application because the overall information on the data provided by the camera.

The two views are presented in the following figures.

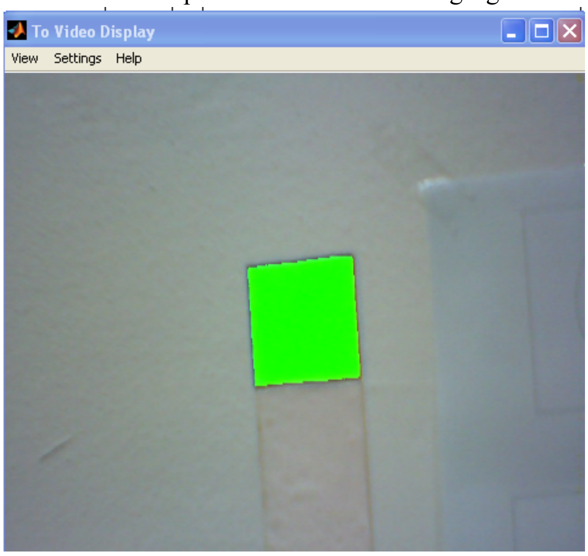

Figure 3. Image whit the resulted surfaces highlighted

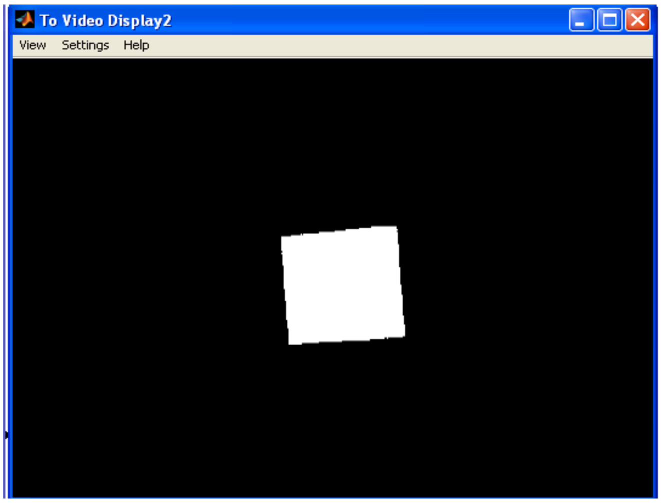

Figure 4. Image resulted after the separation

Following the identification of the area that is used for measuring the distance and converting the signal into a binary image, a matrix will be obtained that has the pixels representing the surface with a value of 1 and the rest of the image will be filled with zeros. As a result, for determining the surface area is sufficient to make the sum of all elements of the matrix. To stabilize the result seven consecutive measurements values are used. These values will me mediated thus resulting the average value of the area measured. Through this mediation the fluctuations in value arising from the noise that affects the measurement are greatly reduced. The following figure presents the structure subsystem to calculate this average value.

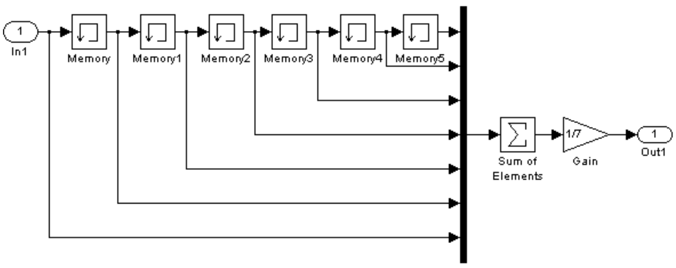

Figure 5. Averaging values subsystem

Because the measured surface is a square to the side length is sufficient the extraction of square root. The calibration measurement is done for determining the focal length of the video device. The focal distance is then use in a subsystem that calculates the distance from the surface to the camera. To calculate the focal length it is necessary to make two measurements for the distance that will cause a doubling of the image scale on the camera sensor. These distances will be used in the following algorithm for calculating the focal length:

M – *scaling factor*
f – *focal distance*
S_{11},S_{12} – *distances from the object to the camera*
S_2 – *distance from the lens to the image*

$$\frac{1}{S_1}+\frac{1}{S_2}=\frac{1}{f} \qquad (1)$$

$$M=-\frac{S_2}{S_1}=\frac{f}{f-S_1} \qquad (2)$$

$$(2)\Rightarrow\begin{cases} M_1=\dfrac{f}{f-S_{11}} \\ M_2=\dfrac{f}{f-S_{12}} \end{cases} \qquad (3)$$

$$M_1=2M_2 \qquad (4)$$

$$(3),(4)\Rightarrow\frac{f}{f-S_{11}}=2\frac{f}{f-S_{12}} \qquad (5)$$

$$f=2S_{11}-S_{12} \qquad (6)$$

With the focal length calculated the next step is to determine de magnification scale of the application. This is done by a simple division between the theoretical and the measured one. This ratio is also introduced in the distance measurement subsystem. The formula for obtaining the distance to the object in view is presented below:

$$S_1=f\left(1-\frac{1}{M}\right) \qquad (7)$$

The resulting subsystem that calculates the distance based on the calibration made and the area of the square surface is presented in figure 5.

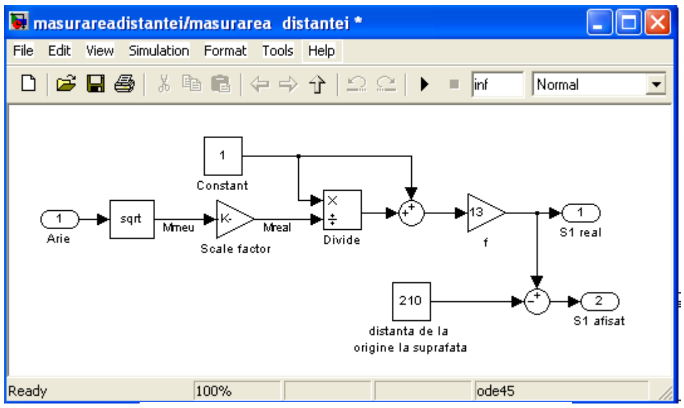

Figure 6. Distance measuring subsystem

The entire model that includes all components plus a histogram to illustrate the distribution of gray levels in the image is presented in the following figure:

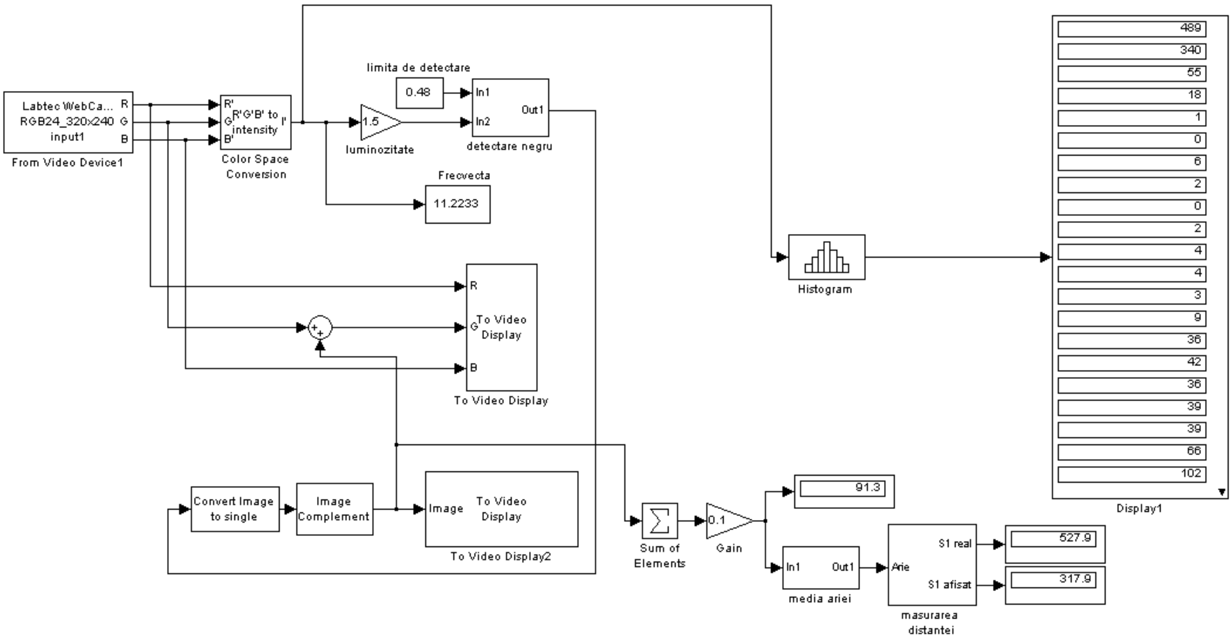

Figure. 7. Simulink model

V. SERIAL COMUNICATION

To verify the positioning of the video camera the serial robot Mitsubishi RV-2AJ was used. In the end effector of the robot arm it was placed the camera used to obtain the images. As image-taking equipment a webcam connected to a computer was used. This design was chosen because it allows a high precision positioning of the camera, accuracy determined by the precision of the robotic arm.

Figure 8. serial robot with mounted camera

Another advantage of this configuration is that the robot great workspace is transformed into a high flexibility of positioning for the camera, making it possible to test the functionality of the algorithm for distances measuring over a wide range of values. It is also possible to use this system for measuring the cameras orientation, but these measurements are not treated in this paper. Inserting the robot arm into the camera positioning measurement system significantly facilitate the desired positioning within a small interval of time. The distance measurement between the camera and the object is done by determining the position of robotic arm end effector and then compensating the distance between the object and reference system of the robot.

To receive the end effector position in the computer it was made an application, which used the serial interface to communicate between the robot controller and Matlab. Communication is carried on COM1 serial port using RS232 communication protocol. To order the robot, in the controller was placed a program that positioned his end effector in predefined positions, and then forward its position on the serial interface communication. Program that performed these operations is presented below:

```
10 OPEN "COM1:" AS #1
20 MOV P1
30 print #1,p_curr
40 DLY 5
50 MOV P2
60 print #1,p_curr
...
200 MOV P7
210 print #1,p_curr
220 DLY 5
230 MOV P8
310 print #1,p_curr
320 DLY 5
```

The program will run in the loop to verify the accuracy of measurements for multiples instances.

For communication with the robotic arm controller was developed an application that interprets the data sent to it, transforming them from ASCII characters into value of the position. To do this conversion, the application examines each character sequentially and then, depending on the value and the transmission data format, transforms the string of characters received into position and orientation of the end effector. The subsystem that carried out this conversion is shown in the following figure.

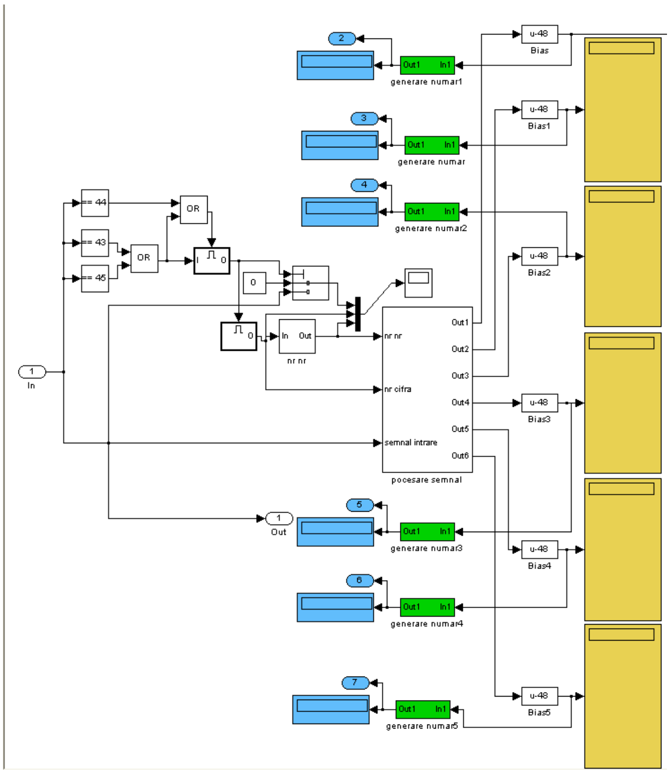

Figure 9. Subsystem that converts the ASCII signals received to position and orientation

This subsystem is designed to perform with the following type of format: *+1,(+297.64,+105.24,+286.62,-85.74,+176.44 , +0.00)(6,0).*

The subsystem will include a serial interface configuration block and a block that will enter into the model data received on the COM1port. The signal thus obtained will be compared with the calculated distance from the camera. This application examine only the Ox axis movement, movement on the other axes are ignored.

VI. OTHER APPLICATIONS

Beside this application of distance measurement and verification of the results we have been designed some other applications in Simulink.

One of them involves detecting objects having a certain colour. The colour to be detected is chosen in an interactive way using three sliders. The three sliders define the brightness for each of the following colours: red, green, and blue. Thus by changing these brightnesses it is possible to choose any hue depending on user options. With the colour produced, a comparison is made between it and the difference in light intensity for each colour (red, green, blue), and if the amount of colour difference is less than the predefined value than it is considered that the pixel has the colour desired.

Errors of tone, representing the intensity differences are taken in absolute terms to avoid their mutual cancellation and then summarized and compared with a threshold. As a result of this comparison with the threshold we will get a binary image that will illustrate portions of the image that have the chosen colour. It was also implement a window that displays the colour chosen for more clarity during the choice. The following figure illustrated the application at runtime; the chosen colour is red.

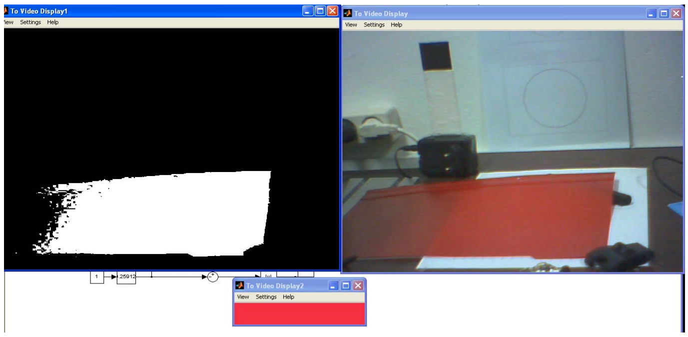
Figura 10. Color detection application

The Simulink model, together with the sliders that are used for colour selection are presented in the following figure:

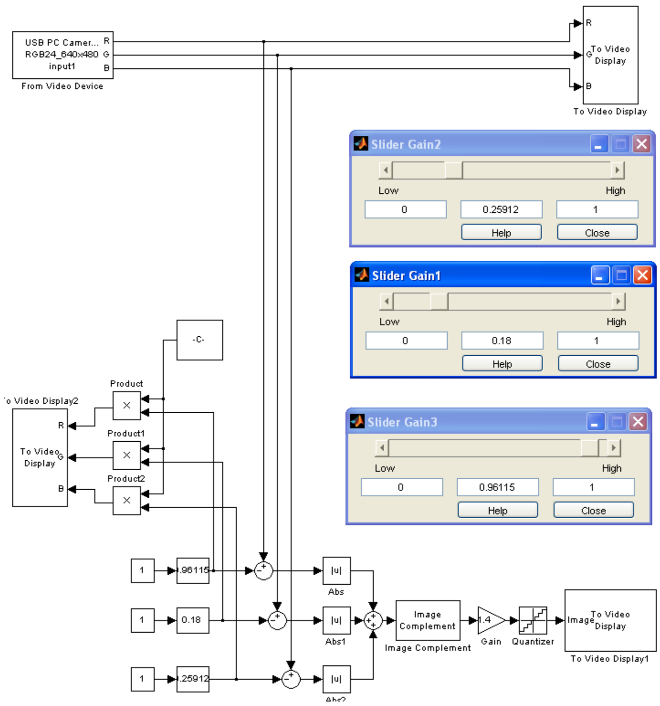

Figure 11. Simulink model for colour detection application

There have also been implemented in Simulink several algorithms for edge detection in order to analyze their effectiveness. As in the application for distance measurement we used a window to reveal the original image with the results highlighted. The following figure illustrates this application and the results obtained by using the next algorithms: Sobel, Prewitt, Roberts and Canny.

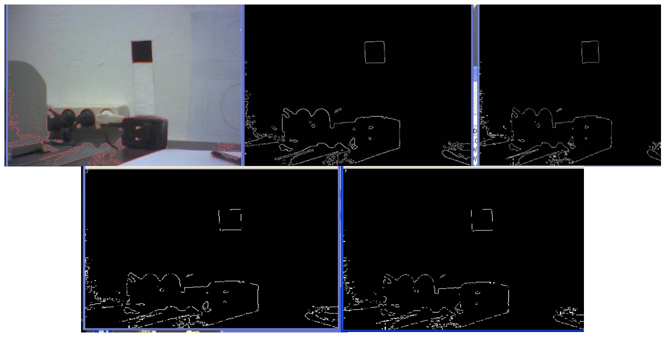

Figure 12. Result for edge detector using different algorithms

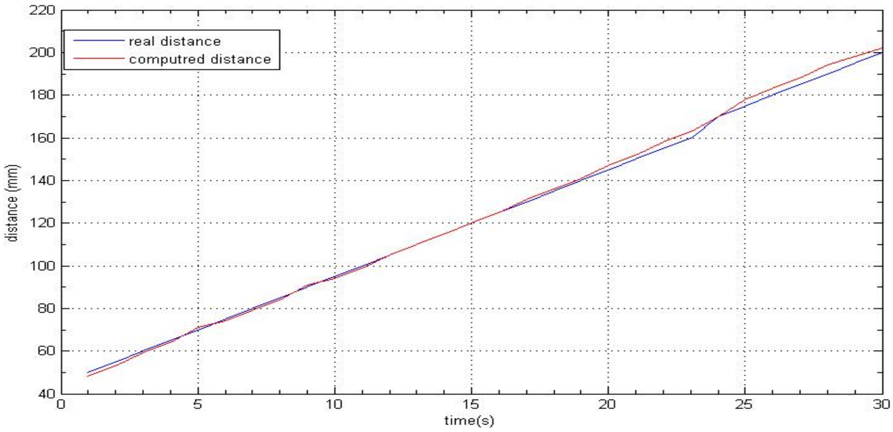

Figure 13. Measured and real distance between the object and the video camera

IV RESULTS

The graphic in figure 13 illustrates the estimation of the distance and the actual position of the video camera. We hope to improve these results by a using a camera whit a higher resolution and by a better removal of noise in the obtained image. Also a camera whit a higher frame rate would improve the application dynamic performance.

V CONCLUSIONS

This application can be the starting point for developing a more complex set of image processing applications. Its practical applicability is clearly visible and do not require major changes if the object used for measuring change. If he remains one square a recalibration is enough for the system to operate at the same parameters. If another type of surface is chosen, it is essential to make the correlations between its surface area and a line contained in it (for example, the diameter, if the surface is a circle $D = \sqrt{A/4\pi}$).

ACKNOWLEDGMENT

Financial support for this work was supported in part by the CNMP under Grant no. 72-197/1.10.2008 PARTENERIATE, title of the project 'Complex mechatronics systems for medical applications'.

VI REFERENCES

[1] M. Coman, R. Bălan Controlul robotului serial RV-2AJ, CNET&TE, Cluj – Napoca, 2009 ISSN 2066-446X

[2] Arthur Schuster, An Introduction to the Theory of Optics, London: Edward Arnold, 1904

[3] Greivenkamp, John E. (2004). Field Guide to Geometrical Optics. SPIE Field Guides vol. FG01. SPIE. pp. 19–20. ISBN 0-8194-5294-7

[4] Ariel Lipson, Stephen G. Lipson, Henry Lipson, Optical Physics 4th Edition, "Cambridge University Press", ISBN-13: 9780521493451

[5] Ray Sydney Applied photographic optics: lens and optical systems for photography, film , video digital imaging. 2002 ISBN 0240515404

[6] M. Born and E.Wolf (1999). Principles of Optics (7th ed.). Pergamon Press.

[7] E. Hecht (2001). Optics (4th ed.). Pearson Education. ISBN 0805385665.

[8] Fowles, Grant R. (1989). Introduction to Modern Optics. Dover Publications. ISBN 0-486-65957-7

[9] S. G. Lipson (1995). Optical Physics (3rd ed.). Cambridge University Press. ISBN 0521436311

[10] John E. Greivenkamp (2004). Field Guide to Geometrical Optics. SPIE Press. p. 6–9. ISBN 9780819452948.

[11] Leslie D. Stroebel View Camera Technique. Focal Press. (1999). ISBN 9780240803456.

[12] Deepak Malani, REC Calicut ,Anant Malewar, Introduction to Video analysis using MATLAB

[13] Matworks, Image Processing Toolbox™ 6 User's Guide.

[14] Stan, S., Workspace optimization of a two degree of freedom mini parallel robot, IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics – AQTR 2006 (THETA 15), Cluj-Napoca, Romania, pp. 278-283, 2006.

[15] Stan, S.-D, Manic, M., Mătieş, M., Bălan, R., "Evolutionary Approach to Optimal Design of 3 DOF Translation Exoskeleton and Medical Parallel Robots", HSI 2008, IEEE Conference on Human System Interaction, Krakow, Poland, May 25-27, 2008.