Ján VACHÁLEK[*]**, Marian GÉCI**[**]**, Oliver ROVNÝ**[***]**, Tomáš VOLENSKÝ**[****]

LOCALIZATION OF OBJECTS USING THE MS WINDOWS KINECT 3D OPTICAL DEVICE WITH UTILIZATION OF THE DEPTH IMAGE TECHNOLOGY

LOKALIZÁCIA OBJEKTOV ZA POMOCI 3D OPTICKÉHO ZARIADENIA MS WINDOWS KINECT S VYUŽITÍM HĹBKOVÝCH MÁP OBRAZU

Abstract

The paper deals with the problem of object recognition for the needs of mobile robotic systems (MRS). The emphasis was placed on the segmentation of an in-depth image and noise filtration. MS Kinect was used to evaluate the potential of object location taking advantage of the in-depth image. This tool, being an affordable alternative to expensive devices based on 3D laser scanning, was deployed in series of experiments focused on object location in its field of vision. In our case, balls with fixed diameter were used as objects for 3D location.

Abstrakt

Príspevok sa zaoberá problémom rozoznávania objektov pre potreby mobilnej robotiky a bol vypracovaný s dôrazom na hĺbkový obraz pri segmentácii a filtrácii rušivých informácií. Pre posúdenie potenciálu hľadania objektu na hĺbkovom obrázku bol použitý MS Kinect, nakoľko je to lacná alternatíva k drahým laserovým 3D snímačom. Za pomoci neho boli vykonane experimenty orientované na vyhľadávanie objektov v jeho zornom poli. V konkrétnom prípade riešeného v tomto príspevku sa jedná o loptičky s pevne stanoveným priemerom.

Keywords

Localization, 3D optical device, Kinect, HSV color model, Depth maps

---

[*] doc. Ing., PhD., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail jan.vachalek@stuba.sk

[**] Ing., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail marián.geci@stuba.sk

[***] Ing., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail oliver.rovny@stuba.sk

[****] Ing., PhD., Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, Námestie Slobody 1, Bratislava, tel. (+421) 2 5249 7193, e-mail tomas.volensky@stuba.sk

1   INTRODUCTION

In past, both visual inspection of objects and control of robotic systems in environment with changing conditions had to rely on human vision and abilities of operator to decide and control. Deployment of human senses in industrial processes is often expensive and leads to varying quality of products. Moreover, teach-in process of operators is time-consuming and thus inefficient mainly in the environments where the tasks are often alternated. Another problem is that human capabilities to decide and control are relatively slow. All these facts led to development of a new field called computer vision, which is dedicated to problem of real time information acquisition from an image. Significant progress in this field allowed an implementation of new technologies such as measurement, inspection and process control aided by imaging devices and another electronics. These new systems allow real time processing of huge amount of data. Hand in hand with these technologies comes a possibility to provide mobile robotic systems with higher level of autonomy in interaction with their surroundings as well as there is a need to provide students of robotics an opportunity to learn these new systems and to gain valuable practical experience with devices which meet the current level of industrial control systems.

The article is dedicated to evaluation of the potential of object location based on the in-depth image processing. Currently, the in-depth images are mostly used for distance measurement whereas coloured images are used for 3D object location. However, the in-depth image processing seems to be good alternative to methods of object location involving coloured images. MS Kinect was used for image acquisition, being an affordable alternative to expensive industrial 3D laser scanners.

2   DESCRIPTION OF THE MS WINDOWS KINECT 3D OPTICAL DEVICE

Microsoft Kinect is probably the most affordable optical device, often used for robotic applications. Its advantage in comparison with professional counterparts lies in relative simplicity and an obvious economic advantage. As this system is primarily meant for the videogame industry, several modifications were necessary for its utilization on the MRS [1], including its integration to different software environments, especially Matlab.

The main components of this device are a proprietary single use DSP chip, a CCD camera and a depth-sensing camera. The depth-sensing camera is created by an infrared camera and an infrared emitter module. The Kinect device uses the USB 2.0 interface to communicate with the computer. The first optical camera allows a user set resolution of either 640 x 480 pixels or 1280 x 960 pixels. In addition, it allows the user to switch between different colourspaces. The depth measurement implemented by the second camera is possible in two working regimes: the first is used by the X-box gaming console, having an approximate range of 0.8 to 4 m. The second mode of operation is the so-called "near" regime, having a range from 0.4 m up to 3 m. This is available only on a special modified version of the Kinect, named Windows Kinect (used for the MS Windows operating system), which is also considered in this work. The resulting distance information is the perpendicular position of the measured object from the face of the Kinect sensor. The output data are matrices measuring 640 x 480 for both cameras, which are then fed and evaluated inside the Matlab environment (Fig. 1).
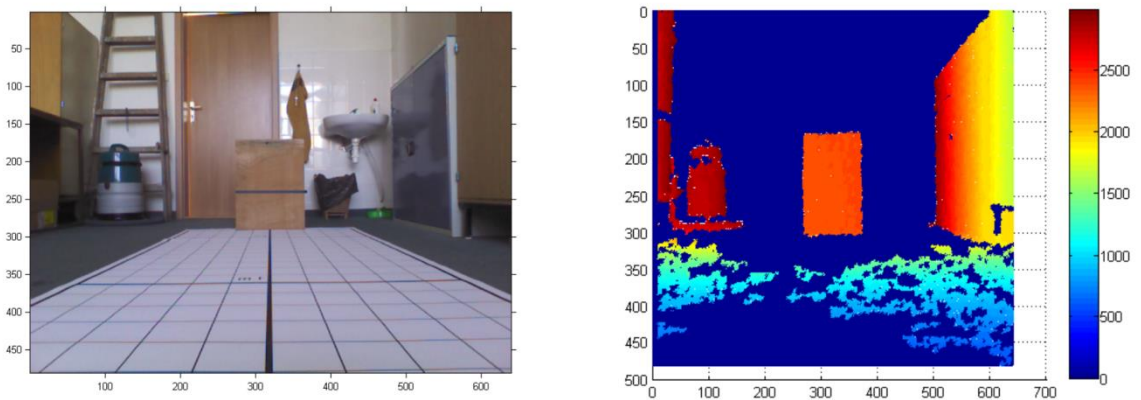
Fig. 1 Output from Kinect, overlaid with a pseudo color representation of the measured depth field in Matlab

3 LOCALIZATION OF OBJECTS

For the exact evaluation of coordinates, the internal parameters of the CCD camera must be calibrated [2,3]. Fortunately, a ready-made tool called the Camera Calibrator Toolbox is available in the Matlab environment. This toolbox was designed to compute focal distances and it requires a complex calibration procedure. The approach in the heart of the procedure is the use of a specific calibration screen with a certain shape and dimensions. This is black and white screen, which must be sensed from different pre-determined angles. The number of images taken with the camera equal to 36. One field of the calibration screen has the dimensions of 30 mm by 30 mm. After using the resulting matrices in the mentioned toolbox, the focal distances f along with the center of the frame c will be computed. Using this procedure, we have calibrated the camera according the data shown in Tab. 1.

Tab. 1 Optical parameters of the first camera

Focal points	
f_y	517,3
f_z	517,4
Center	
c_y	318,5
c_z	239,4

The next step is to calibrate the position of the Kinect device in relation to the global coordinate system, which was placed into its hypothetic center. For this, a program was created to transform different coordinate systems inside Matlab. Amongst others, this program automatically visualizes a crosshair into the footage, with its center given at c_y and c_z. This crosshair is necessary for the correct configuration of the coordinate system, since it helps to align the field of view on the

calibration screens (Fig. 2). This procedure ensures that the Kinect system is turned in the correct angle in relation to the global coordinate system. The object localized in this work is a small yellow sponge ball with a radius of $r_l = 34mm$.
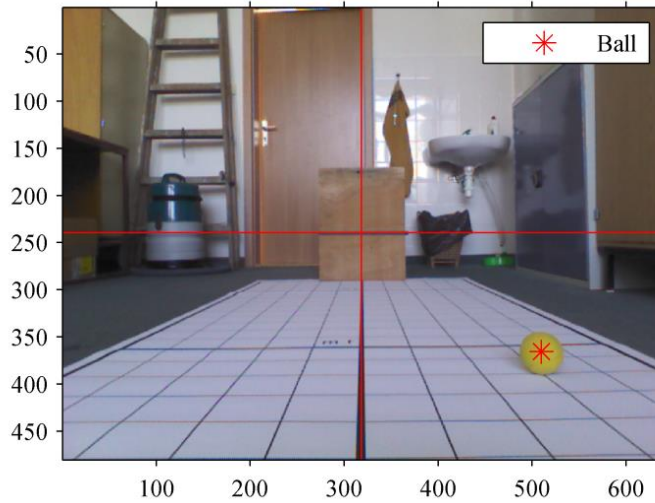

Fig. 2 Localization of the ball using the RGB colorspace

Unlike the common approaches deploying colourful images in process of 3D object location, our solution places the emphasis on the in-depth image segmentation. The main advantage of such approach is that the influence of lighting conditions is strongly suppressed. Thus, image segmentation and recognition of particular objects in the image becomes the crucial part of the whole process of object location. The most significant disadvantage of this method compared to conventional methods using colourful (not depth) images is that the depth resolution is constrained and thus, objects smaller than certain threshold could not be successfully recognized. Moreover, the depth resolution decreases with growing distance from the in-depth camera. However, in our case the field-of-vision of the Kinect head is constrained to distance of 2.2 m (on flat surface) by its position and orientation on the mobile robot. As the diameter of sponge ball is fixed to be 68 mm while the depth resolution of Kinect in distance of 2 m is 14 mm, the ball should be successfully recognized using this method. The advantages and disadvantages of both methods are summarized in the table Tab. 2.

Tab. 2 Comparison of coloured image and depth image method of object location

| Coloured image	Depth image
Advantages	Advantages
Distance of recognition	Depth is known directly
Number of libraries available	Simple method of edge identification
Sharper edges of the object	Independancy on lighting conditions
Disadvantages	Disadvantages
Difficult calculation of distances	Distance of recognition
Necessity of unification	Blur edges and white areas
Dependancy on the lighting conditions	
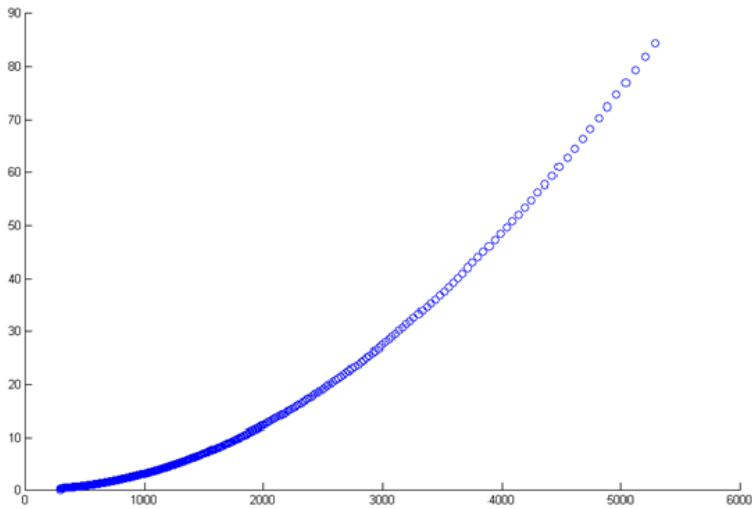
Fig. 3 Depth resolution of the Y axis of the camera depending on the distance from the camera (X axis) in millimetres

The Kinect device is equipped with both, emitter of infra-red beams and infra-red camera by means of which the distance measurement is accomplished. Numerous infra-red beams are projected to the space in pseudo-random pattern and simultaneously, the reflections are captured by infra-red camera. The pseudo-random patterns of beam projection are known to Kinect and can be reproduced. Thanks to industrial calibration of the Kinect sensor, the calculation of distance for each pixel of camera is based on comparison of reflections produced by different patterns projected. This calculation is performed by so called Header DSP chip, being a special chip intended to fast signal processing. The principle of depth measurement is as follows: when a spotlight is pointed slightly from right side to some plain perpendicular to camera axis and the plane moves towards the camera, the light spot moves slightly to right. When the spotlight is pointed from left side, spot moves to left. The pattern consisting of thousands of light spots is projected to the scene by IR emitter and the reflections are captured by IR camera simultaneously. The capture of new reflections is compared to the pattern which has been previously saved in memory. The method of structured light beams which was described is the manner how the in-depth image is generated in each cycle of this continuous repeating process. The information about distance from an object to plane of camera is stored by each pixel of resulting depth image.

Depth image is available in following resolutions: 80x60, 320x240 and 640x480 (default). The in-depth resolution is 16 bit. Depth sensor can work in normal or close mode. The distance range in normal mode (default) is 0.8 m up to 8 m, whereas in the close mode the range is 0.4 m up to 8 m with lower resolution. In optimal conditions, Kinect is able to achieve accuracy in millimetres. Pixels with unknown distance store value "0". The data gained using Kinect in Matlab were in compliance with information claimed in datasheets, however it was necessary to ensure optimal conditions i.e. the surfaces perceived by the sensor should not be shiny and the IR beams should not impact the surface at adverse angle. The normal mode of Kinect sensor is sufficient for our needs. Depth resolution varies from 1.5 mm depending on the distance (Fig. 3).
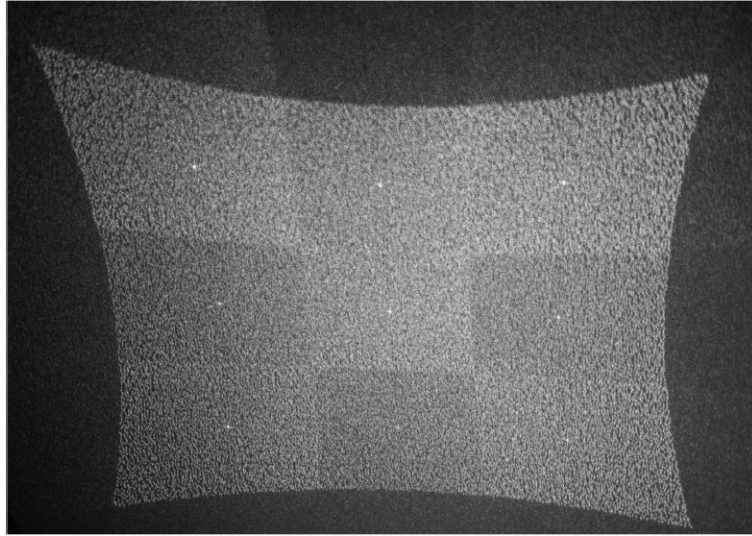
Fig. 4 Infra-red pattern captured by an external camera

4   OUR SOLUTION OF THE TASK

The problem is stated as recognizing of objects with certain shape and colour. The goal is to evaluate the coordinates of the object in the local coordinate system of Kinect and to transform them to global coordinate system of the MRS [1]. The emphasis was placed on the in-depth image segmentation. In our study, a 68 mm plastic foam ball was employed as the object to be identified and located. The algorithm of identification of the object in the depth image can be split to following steps [4]:

1.   Finding interesting pixels.
2.   Path segments growing and checking the geometry of segments.
3.   Colour checking and coordinates transformation to the global coordinate system.

The following sections are devoted to the particular algorithm of ball identification in the depth image and coordinates calculation. The solution is demonstrated in the figures Fig. 5 and Fig. 6, which were acquired by MS Kinect.
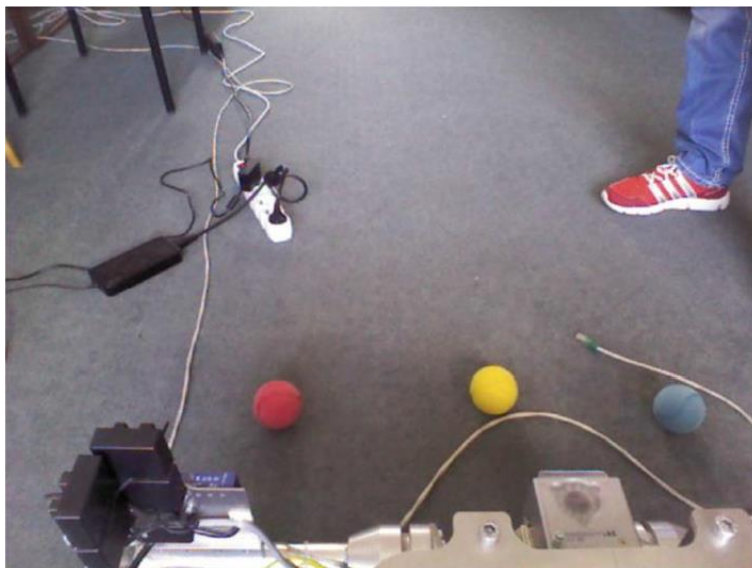
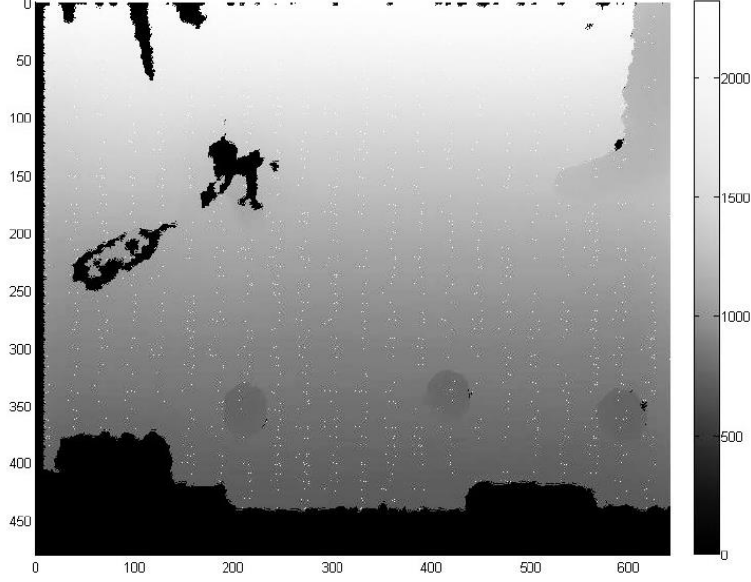Fig. 5 Coloured image of the cut out captured

Fig. 6 Depth image of the cut out corresponding to that shown in the previous figure

5   SEARCHING FOR OBJECTS OF INTEREST IN FORM OF PIXELS

Primary attention is payed to search for pixels which might belong to the object (ball) [2, 7]. The further growth of the segments is than initiated from this clue pixels. Path-following is the simplest as well as the most reliable manner of segments growth. Moreover, the segments gained by this method are convenient for further processing. It is necessary to state proper criteria for identification of clue pixels. The difference between distances of both, the ball outline and the background seems to be a proper criteria for identification of clue pixels. Particular columns of pixels are processed comparing each pixel with previous one. The difference between pixels is compared to interval of interest:

$$l_{min} < pix_i - pix_{i+1} < l_{max} \tag{1}$$

The range of distance differences estimated between ball outline and background is defined by l_{min} and l_{max} values. The simple scheme how the estimated difference in distances is calculated is displayed in the figure Fig. 7. Following equations are used to calculate the distance l:

$$\cos \beta = \frac{H}{L} \tag{2}$$

$$l = \frac{h}{\cos \beta} \tag{3}$$

The distance L in equation 2 is substituted by the greatest and the smallest feasible value and thus the range of estimated values of l is predefined. Calculated for L = 800 mm, corresponding l_{min} = 75.3 mm and for L = 2300mm, l_{max} = 216.6 mm.
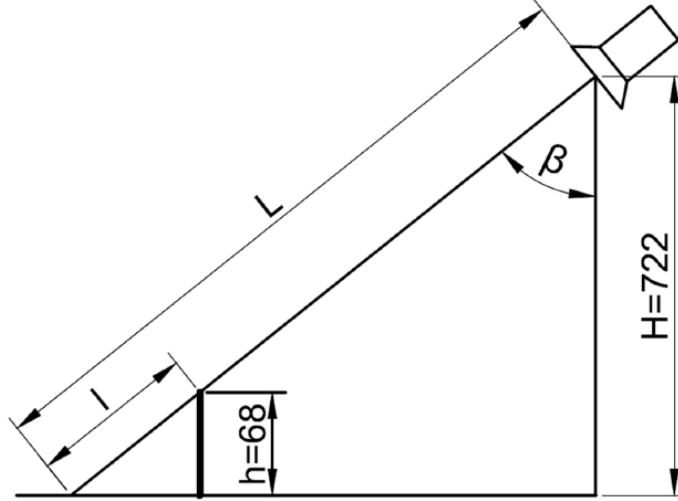

Fig. 7 Basic geometry for calculation of distance l depending on L

H – vertical position of the Kinect device,

L – distance between the device and background above the object,

l – distance between the device and the outline of the object,

h ≈ D – height of the object (equals to ball diameter in our particular case),

β – angle between optical axis and vertical

It is necessary to determine optimal sampling in the horizontal direction to shorten the time needed for code execution. Determination of the optimal sampling was based on the number of pixels within the ball diameter. The number of pixels within ball diameter was identified for the greatest distance L which can be achieved in our setup since the number of pixels is the least in that case. The search process is carried out with respect to the upper contour of the ball, hence at least 1 pixel has to be visible between the ball outline and the upper border of the image. The calculation is based on the geometry shown in the figure Fig. 8 and is as follows:

Width of the area captured in the image in the distance of L_{max}:

$$y_{mm} = 2 * L_{max} * \tan(\alpha/2) \tag{4}$$

Number of pixels corresponding to 1 mm in the distance of L_{max}:

$$c = \frac{y_{mm}}{640} \tag{5}$$

Width of the ball in pixels:

$$D_{ypx} = \frac{D_{mm}}{c} \tag{6}$$

70

The maximal feasible distance L_{max} is 2.3 m. Corresponding width of the ball in pixels is $D_{ypx} = 17.42$ px. In the next step, 10 pixels are substracted from the upper border of the image and the distance L is read. The calculation is repeated for the new value of L and thus the new value of $Dyp = 18.09$ px is gained. Hence, the value of horizontal sampling of 18 pixels is considered to be optimal: no more accurate calculation is necessary regards to the camera inaccuracy and resolution. The function described is called "*zaujimave_pix*" and it is our own function. The coordinates of the pixels found by this function are saved into the structure called "*segmenty*". These pixels, being the first pixels of the outlines of objects captured, are the pixels which initialize the growth of segments.
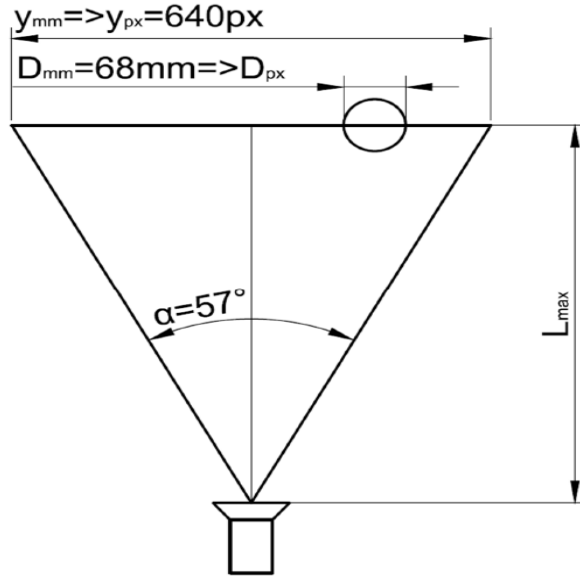

Fig. 8 Recalculation of the object width from milimetres to pixels

6   THE GROWTH OF SEGMENTS ALONG THE OBJECT OUTLINE

The growth of the particular segment is ensured by another function developed within our project. The function is meant to add another relevant pixels to the pixel germs. Thus the outline contour of the particular object is sequentially created. As described in section 5, the identification of the pixels belonging to the outline contour is based on the difference between the values of neighbouring pixels. All 8 neighbouring pixels are examined using this method. The growth of segment is stopped at the moment when either the condition (1) is met by no another neighbouring pixel or the segment is too long to be considered belonging to the outline of the ball. An instance of affiliation of pixels in the typical part of contour is displayed in the table Tab. 3. The particular pixels are represented by the cells where [5]:

- White cells (pixels) belong to the background
- Green cells are those by which segment is formed
- Cells with numbers displayed in bold are considered to belong to the ball
- Grey cell marks the initial pixel

Tab. 3 Identification of the cells (pixels) belonging to the outline contour of the object

1838	1848	1848	1848	1838	1838	1838
1838	1838	1838	1838	1848	1848	1848
1828	1828	1828	1838	1838	1848	**1638**
1809	1809	1828	1828	**1653**	**1653**	**1653**
1800	1800	1809	**1669**	**1669**	**1661**	**1661**
1800	1800	**1661**	**1669**	**1669**	**1661**	**1661**
1800	**1669**	**1661**	**1661**	**1669**	**1661**	**1661**
1669	**1669**	**1669**	**1661**	**1661**	**1661**	**1661**
1669	**1669**	**1661**	**1661**	**1661**	**1653**	**1653**

The segments associated to the particular in-depth image are displayed as yellow contours in the figure Fig. 9.
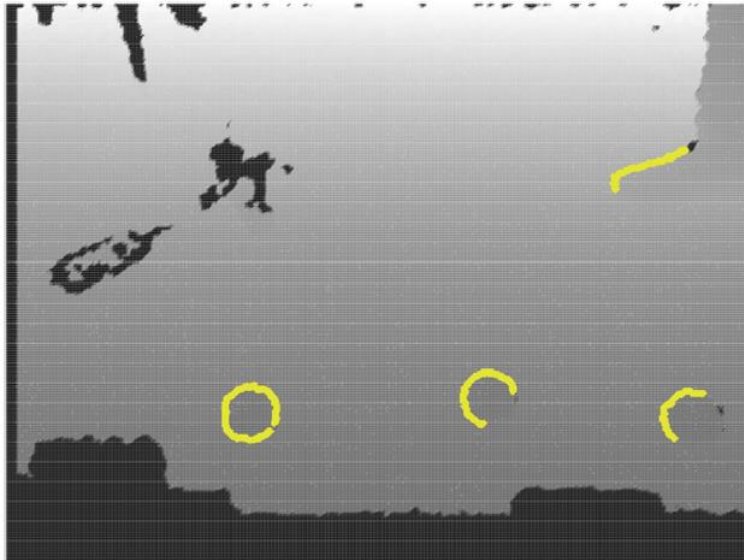

Fig. 9 The segments associated to the in-depth image captured by the Kinect device

7   GEOMETRY CHECK OF THE SEGMENTS CREATED

As there is a necessity to get rid of segments not belonging to the ball, the appropriate filtration of segments has to be undertaken. This is the role of the function called "*circlefit*" which is used to determine whether the segment curve can be approximated by a circle of diameter which is close to ball diameter or not. But just before deploying this function, it is reasonable to get rid of

segments which are either too long or too short to be considered as outline curves of balls. In the next step the remaining segments are approximated by circles. After that the depth of the centre pixel of each circle is determined. The circle approximation is done using the least squares method. The parameters centre, radius and depth are than saved to the "*segments*" structure. The resulting circles after approximation are shown in the figure Fig. 10.
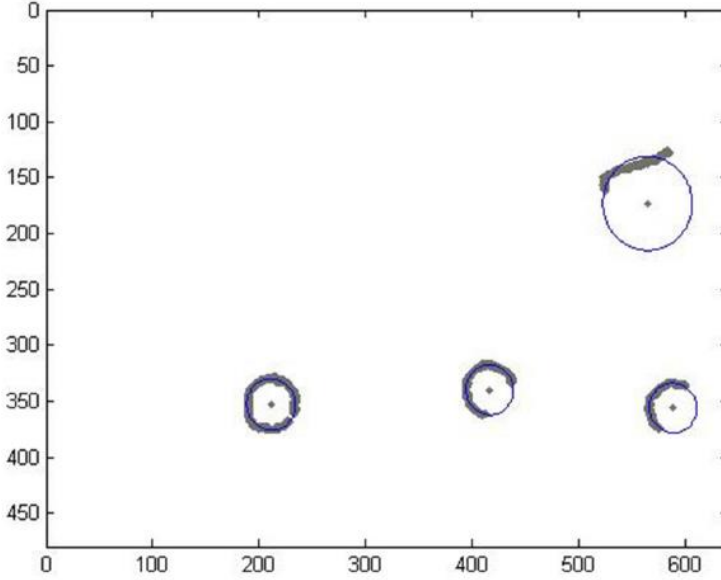

Fig. 10 Circle approximation via segments

The final step is to calculate the error of approximation and to remove the segments which plausibly do not belong to the ball and are therefore rejected. The average of summed radius deviations is used to calculate the error of approximation.

$$E = \frac{\sum_{i=1}^{n}(|R-R_i|)}{n} \tag{7}$$

R – radius of the circle

R_i – distance of the i-th pixel from the circle centre,

n – number of pixels within segment

It was discovered experimentally that the error of approximation does not exceed the value of 1.45 if the segment belongs to the ball. Therefore, segments characterized by greater value of this error are rejected. Further filtration of the segments gathered is based on the radius check. According to equations (4), (5) and (6), minimum and maximum radii are 9 px and 26 px. For instance, the third segment from the table Tab. 3 is rejected after the geometry check since the error of approximation is too big as well as the radius itself.

Tab. 4 Structure "segments" with all parameters assigned

Fields	rows	cols	Tx	Ty	R	value	error
1	1x371 double	1x371 double	211	353	22,8418	842	0,990733
2	1x264 double	1x264 double	416	340	22,28403	859	0,793337
3	1x219 double	1x219 double	564	173	41,89111	1319	3,630786
4	1x187 double	1x187 double	588	356	22,2038	834	0,94205

The meaning of parameters in the table Tab. 3 is as follows:

Fields – numbers assigned to segments

rows, cols – coordinates of the pixels in the depth image

Tx, Ty – coordinates of the circle centre

R – circle radius

Value – the distance of the point with coordinates [Tx,Ty]

error – error of approximation

8 COLOUR CHECK OF THE SEGMENTS

The last condition used to filter the selected objects is the colour match. To verify this condition it is necessary to check the colour of the pixels in the coloured image which correspond to the position of interesting objects identified in the depth image [4]. Hence, transformation of the coordinates of each circle centre from the depth image to the coloured image is necessary. The transformation is accomplished using the data in the table Tab. 5.

Tab. 5 Parameters of the coloured camera [8]

	Horizontal	Vertical
Focal lengths	$f_y = 517{,}25$	$f_z = 517{,}42$
View angles	63.5°	49.8°

The images from the depth camera are captured from different viewpoint than those of the coloured camera. Moreover, the view angles of the cameras are not the same. From these two reasons there are two kinds of mismatch between the images captured by these cameras:

- Image shift
- Image size mismatch

The mismatch from image shift is characterised by simple constants, whereas the calculation of correction for different view angles is described by the figure Fig. 12:
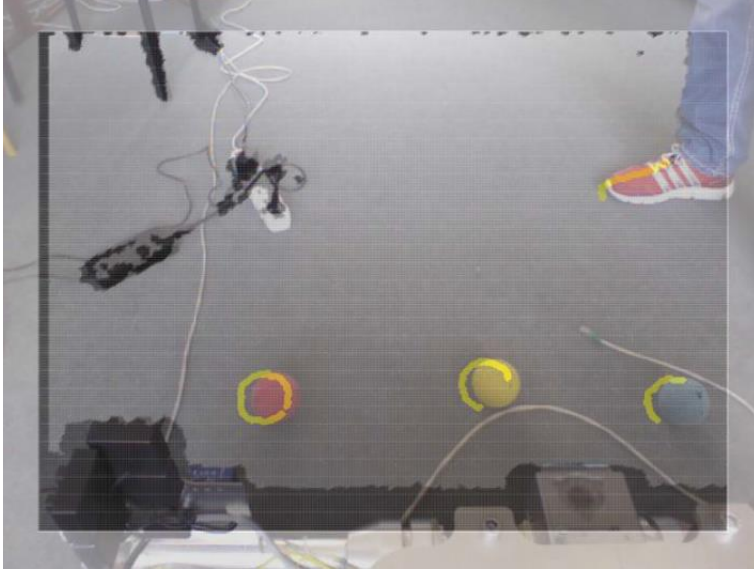
Fig. 11 An example of overlapping images demonstrating the misalignment in concentric display caused by different viewpoints of the two cameras: the size ratio of the images correspond to the ratio of view angles.

$$b_{rgb} = \frac{\tan(\alpha_{rgb}*0,5)}{\tan(\alpha_{depth}*0,5)} * b_{depth} \tag{8}$$

Resulting transform equations of *[Tx,Ty]* coordinates to *[Txrgb,Tyrgb]*:

$$Tx_{rgb} = 320 + \frac{\tan(\alpha_{rgb}*0,5)}{\tan(\alpha_{depth}*0,5)} * (Tx - 320) + 4 \tag{9}$$

$$Ty_{rgb} = 240 + \frac{\tan(\alpha_{rgb}*0,5)}{\tan(\alpha_{depth}*0,5)} * (Ty - 240) - 6 \tag{10}$$

After recalculation of the coordinates, colour at the particular point is checked. The colour check is carried out using HSV colour model. Hence, the colour acquired from the RGB camera has to be transformed to the HSV model analogy. It is necessary to determine the hue (H component) of the ball and its tolerance since this component of the HSV model is only weakly affected by lighting conditions. A special program has been developed which allows to read the image, to transform it to several colour models and to find the points with defined properties.
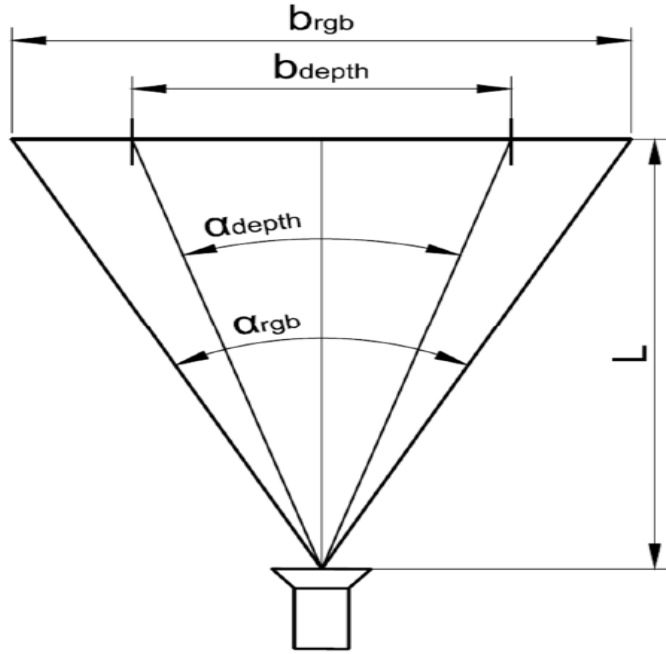
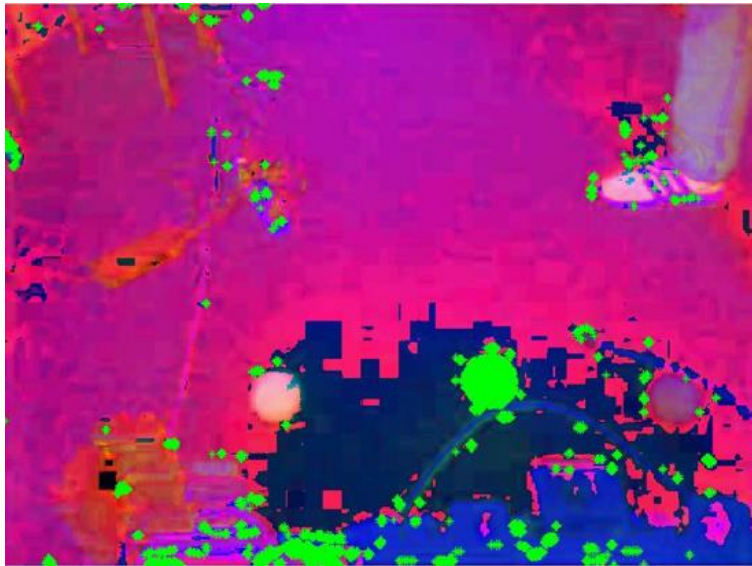Fig. 12 Correction of the view angles

Fig. 13 HSV image with determined points

Figure 13 shows an example of the picture inserted in the program in HSV display. As the values of saturation (S) and brightness (V) are strongly affected by lighting conditions, the tolerances are set wide to neglect their influence on the result. The hue component is decisive in the phase of colour check. The identified point is marked with a green spot. It turned out that when H is set to 0.12 with tolerance of 0.02, every point of the ball is marked. The results of the same experiment with another image led to values H = 0.16 with tolerance 0.02. A series of experiments revealed that the optimal values are H = 0.14 with tolerance of 0.04. After execution of the function for colour check which was developed within our study, there should remain just one last segment in the structure "*segments*", corresponding to the ball [2].

9 TRANSFORMATION OF THE COORDINATES TO THE GLOBAL COORDINATE SYSTEM

Transformation of the coordinates of the ball in the local coordinate system of the Kinect device to the global coordinate system is the last step of the complex location procedure. The global coordinate system is fixed to the base platform of the MRS and the parameters of the relative position of the device mounted on the MRS are well known. The Denavit-Hartenberg method has been implemented to perform the transformation. The Denavit-Hartenberg parameters have been determined and thus the transformation matrix was set (11) [6]:

$$
\begin{bmatrix}
\cos\alpha & 0 & \sin\alpha & x_p + x_{kw}\cos\alpha + z_{kw}\sin\alpha \\
0 & 1 & 0 & y_p + y_{kw} \\
-\sin\alpha & 0 & \cos\alpha & z_p + z_{kw}\cos\alpha + x\sin\alpha \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{11}
$$

8 CONCLUSION

The aim of our study was to evaluate the potential of object location based on the in-depth image processing. Two methods of evaluation of our solution were used. The first method was performed using a board with chess-pattern and it was deployed to test the accuracy in x direction. The vertical coordinate was gradually changed from 240 px to 420 px. The average deviation was 1.38 mm. However, this value is a guide only since there is an error due to the resolution of camera and this error cannot be neglected. The second evaluation was carried out as a comparison to the solution which has already been created in past. Very similar values in X and Y axes were obtained. On the other hand, Z values were different. This was caused by the fact that the calculation was carried out using the value α which was calculated using the values from Kinect, whereas in past the angle α was directly measured. Finally, the difference between the values gained by method involving the depth image and the coloured method were in the position of the centres of balls. However, the difference is only slight. More significant differences were observed in the case when the lighting conditions were changed. Based on the results of the series of experiments performed, we can claim that thanks to the approach deploying depth image in the object location process, it is possible to increase the tolerance of the colour during search. Thus the negative influence of changing lighting conditions during object location with aid of computer vision is suppressed.

REFERENCES

[1] VACHÁLEK, J., TÓTH, F., KRASŇANSKÝ, P., ČAPUCHA, Ľ.: *Design and construction of a robotic vehicle with omni-directional mecanum wheels*. In Sborník vědeckých prací Vysoké školy báňské - Technické univerzity Ostrava. Roč. 60, č. 1 (2014), s. 97-103. ISSN 1210-0471.

[2] VACHÁLEK, J., ČAPUCHA, Ľ., KRASŇANSKÝ, P., TÓTH, F.: *Collision-free manipulation of a robotic arm using the MS Windows Kinect 3D optical system.* In Process control 2015 : 20th International Conference on Process Control. Štrbské Pleso, Slovak Republic. June 9-12, 2015. [s.l.] : IEEE, 2015, S. 96-106. ISBN 978-1-4673-6627-4.

[3] FOSSATI, A., GALL, J., GRABNER, H., REN, X., KONOLIGE, K.: *Consumer Depth Cameras for Computer Vision*, Advances in Computer Vision and Pattern Recognition. Springer, London, 2013.

[4] ŠONKA, M., HLAVÁČ, V., BOYLE, R.: *Image Proccesing, Analysis and Machine Vision*, International Student Edition. 3nd. United States of America 2008.

[5] JOHNSY, A.: *IMAGE PROCESSING. Find Area, Perimeter,Centroid, Equivdiameter, Roundness and Bounding Box without Using MATLAB Function 'regionprops'*. [cit. 3. Február 2015] Dostupné na internete: http://angeljohnsy.blogspot.com/2012/05/find-areaperimeter-centroid.

[6] VACHÁLEK, J., TAKÁCS, G.: *Robotika*. 1. vyd. Bratislava : Nakladateľstvo STU, 2014. 166 s., 96 obr., 2 tab. ISBN 978-80-227-4163-7.

[7] VACHÁLEK, J., KRASŇANSKÝ, P., VAJSÁBEL, M.: *Využitie 3D optického kamerového systému MS Windows Kinect pre potreby manipulácie robotického ramena mobilného robotického systému.* In *Metrológia a skúšobníctvo*. Roč. 19, č. 2 (2014), s. 7-13. ISSN 1335-2768.