MOBILE ROBOT NAVIGATION BASED ON CIRCLE RECOGNITION

Peter Pásztó — Peter Hubinský [*]

This paper presents a navigation method for a mobile robot using a visual system. Circular marks with specific colors are used for marking the significant points of the mobile robot's trajectory that it needs to pass. The colors of the used marks are signalizing the way of their bypassing with the mobile robot (from the left or right side). The mobile robot uses only one camera for the marks recognition task and it is able to determine its own relative position from the detected marks. The image processing and the mobile robot's trajectory planning algorithm working in real-time are described in this paper.

K e y w o r d s: circular mark detection, mobile robot trajectory planning, image processing, Hough transform

1 INTRODUCTION

The aim of this paper is to propose a navigation method for a mobile robot using a visual system that consists of only one ordinary camera (web camera). The algorithm should be able to determine the relative position between the mobile robot and the detected marks and to plan the mobile robot's trajectory to bypass the marks from a desired side. Most of visual system applications are using stereo vision techniques for object's position determination. In this paper the position of the mobile robot relative to the detected mark is computed using only one camera with unknown optical parameters.

The navigation consists of two main parts: the image processing part and the trajectory planning part. Image processing part is a combination of more image processing algorithms (steps) and its goal is to detect all the circular marks in an unknown environment while the changing of light conditions is possible. This part also determines the colors of the detected marks. Changing of the light conditions is an important disturbance bringing noise into the image processing. Therefore the image processing application needs to be designed the way to be able to troubleshoot this disturbance.

Trajectory planning part is computing the positions of the detected marks relative to the position of the mobile robot using the data obtained with image processing part. After it the mobile robot's trajectory is planned using Bézier curves [1].

The navigation process is real-time and the image processing uses the free OpenCV library [2].

2 CONTROL SCHEME

The navigated mobile robot is iRobot Create [3] platform. The control scheme is designed to avoid unnecessary wires. An ASUS Eee PC 901 is embedded to the iRobot Create's body. This notebook has low weight and small size therefore it is easily fixable to the mobile robot, but it has not enough computation power to handle the image processing algorithms. Therefore an external computer with higher computation power is needed to control the mobile robot.

The control scheme is shown on Fig. 1. The ASUS Eee PC 901 embedded to the iRobot Create is used as a server and its task is to capture the input image from the connected web camera and send it to the controlling computer. These two computers are connected over wireless network [13]. The controlling computer ASUS K73SV controls the mobile robot over Bluetooth.

3 IMAGE PROCESSING

This part consists of several steps and the output should be the data about all the marks in the captured image. First of all the marks are needed to be recognized in variable light conditions (because the variability is caused even only by robot movement). During the robot movement the angle between it and the mark is also changing. This takes effect on the projection of the mark into the image of the camera. The projected mark has a shape of an ellipse. The image processing algorithm needs to use image processing techniques those can adapt to the light conditions variability and can detect circular or elliptical objects in the image. Some data about the marks (center position, radius, color, *etc*) should be also known because these data will be used for mark's relative position calculation.

The image processing algorithm scheme is shown on Fig. 2. It uses the HSV color space, Hough transform, edge and contour detection image processing techniques. Important techniques will be described in the following

---

* Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology STU, Ilkovičova 3, 812 19 Bratislava, Slovak Republic, peter.paszto@stuba.sk, peter.hubinsky@stuba.sk
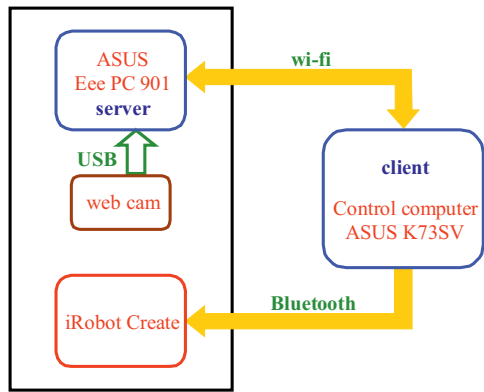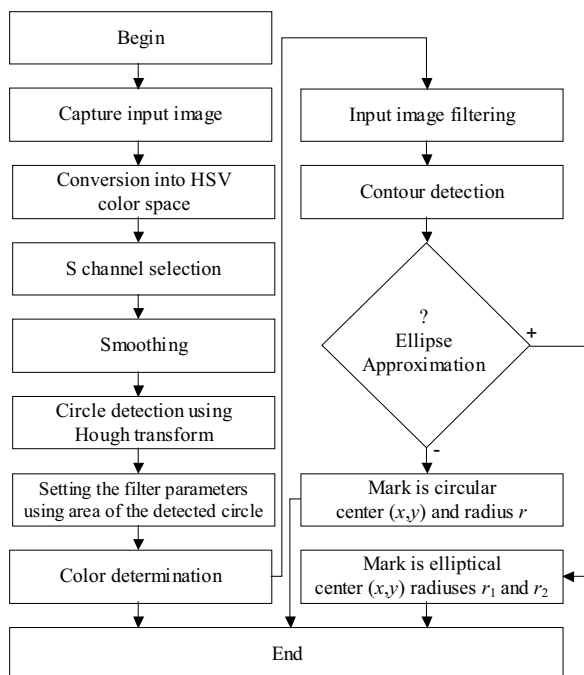
Fig. 1. Control scheme

Fig. 2. Image processing algorithm scheme

subsections. The output of the image processing part is the information about the shape of the recognized mark (circle or ellipse), the position of the center of the mark in the image and its radius (in case of circular mark) or radiuses (in case of elliptical mark).

3.1 Projection of the mark

Assume that all the marks are at the same height as the camera embedded to the mobile robot.

The projection of the mark into the image of the camera (Fig. 3.) can be described by

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1}$$

where (X, Y, Z) are the coordinates of a point in the environment projected into the image of the camera with

coordinates (x, y), (o_x, o_y) are the coordinates of the principal point of the image of the camera (most of the current imaging systems define the origin of the pixel coordinate system at the top-left pixel of the image). The parameter $\lambda = Z$ is the homogeneous scaling factor and f is the focal length of the camera [4].

The mark is projected to the image of the camera as an ellipse if the image plane and the mark plane are not parallel. Otherwise the projection of the mark to the camera image is circular.

If the real radius of the mark in the environment is r_p, it's distance from the camera image plane is d_{yp}, then the radius of the mark in the image if the mark plane and the camera image plan are parallel is

$$r = f \frac{r_p}{d_{yp}} \, . \tag{2}$$

Rotating the mark plane by an angle α around the z axis of the global coordinate system, the radius of the mark in the image of the camera is

$$r = f \frac{r_p \cos \alpha}{d_{yp} + r_p \sin \alpha} \, . \tag{3}$$

Note that in this case the unit of the radius of the mark in the image of the camera r is the same as the unit of the real radius of the mark in the environment r_p.

The radius of the mark in the camera image is a function of the angle α between the camera image plane and the mark plane

$$r = f(\alpha) = f \frac{\cos \alpha}{(d_{yp}/r_p) + \sin \alpha} \, . \tag{4}$$

3.2 HSV color space

It is necessary to transform the RGB color space used with the camera into HSV color space; because the RGB color space has a large variability of the colors of the marks in changing light conditions. The color changes of the area of the mark are mathematically nondescript. In HSV color model H is Hue, S is Saturation and V is value. Only the V channel is rapidly changing with changing light conditions. A minimal change in pixels intensity values in the S channel of HSV image while switching on and off the light source in the environment is shown on Fig. 4.

The RGB color space transformation into HSV color space is [5]

$$a = \frac{(R - G) + (R - B)}{\sqrt{(R - B)^2 + (R - B)(G - B)}} \, , \tag{5}$$

$$H = \begin{cases} \cos^{-1}(a) \, , & B \leq G \, , \\ 2\pi - \cos^{-1}(a) \, , & B > G \, , \end{cases} \tag{6}$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R + B + G)} \, , \tag{7}$$

$$V = \frac{\max(R, G, B)}{255} \, . \tag{8}$$

Fig. 3. Perspective projection

Fig. 4. The minimal change in pixels intensity values in the S channel of HSV image in different light conditions

3.3 Hough transform

The principle of the Hough transform can be explained on line detection [6–8, 11]. Consider a line passing through the points $A = (x_1, x_2)$ and $B = (x_1', x_2')$. All lines passing through the point A meet the equation of the line

$$x_2 = kx_1 + q. \tag{9}$$

Because x_1, x_2 are constant in the point A, the equation can be understood as an equation in a parameter space k, q. In the parameter space all lines passing through the point A in the image space are described by

$$q = x_2 - kx_1. \tag{10}$$

All the lines passing through the point B can be described with equation in the parameter space

$$q = x_2' - kx_1'. \tag{11}$$

The only common point of both lines in the parameter space is the point (k, q) that is responding to the line passing through A and B points in the image space.

Fig. 5. Image with circles

If Hough transform is used for circle detection, the equation of a circle

$$(x - a)^2 + (y - b)^2 = r^2 \tag{12}$$

instead of the line equation has to be used.

In case of circle detection there are three parameters (a, b, r) and the Hough transform accumulator has three dimensions. Consider an input image of size 160×120 pixels. There are three circles in the input image (Fig. 5.), one with radius $r = 40$ px and two with radiuses $r = 20$ px.

Figures 6 and 7 are showing the cuttings through the three dimensional accumulator at corresponding radius values ($r = 40px$ and $r = 20px$). A color map for these figures is shown on Fig. 8. The local maxima in every cutting through the accumulator can be seen on figures.
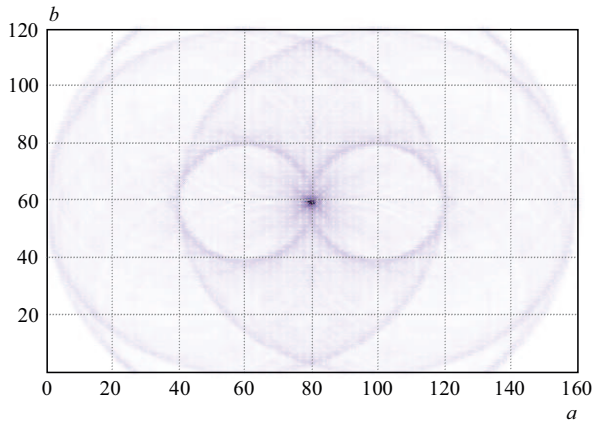
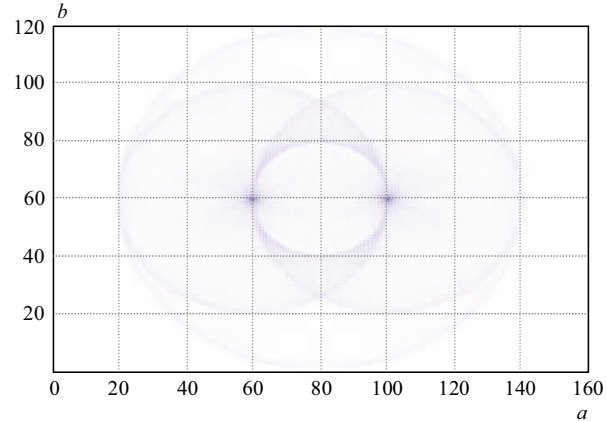**Fig. 6.** A cut of the accumulator at $r = 40$ px

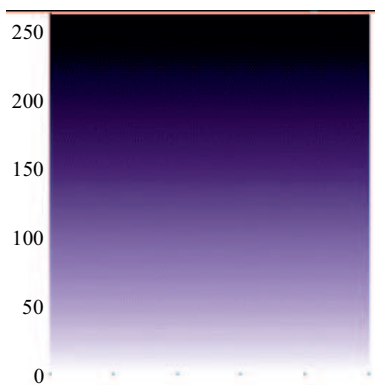**Fig. 7.** A cut of the accumulator at $r = 20$ px
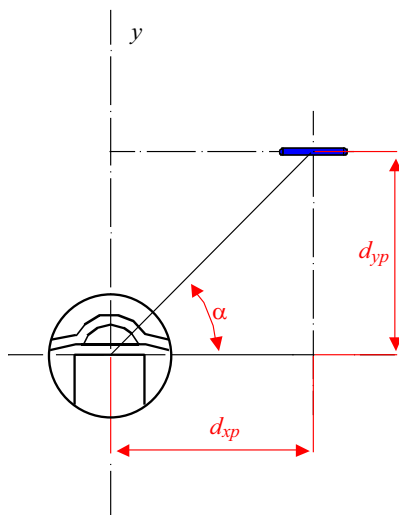

Fig. 8. Color map for Figs.6 and 7

Fig. 9. Calculation of the relative position between the mark and the mobile robot

These local maxima (a, b) with the corresponding r belong to the circles those can be found in the camera image.

The greater is the radius the greater is the local maximum. Its value should be approximately equal with the circumference of the circle (depends on sampling period and the resolution of the image).

Hough transform can be used for detection of objects those can be expressed with a mathematical equation. Its advantages for the purposes of this paper are that after object detection, the parameters of the object are known (center position and radius) and it can detect an incomplete object in the image (for example a standing out or a covered mark).

The real input image to the Hough transform is a single channel image (after edge detection). The OpenCV's Hough transform function is using implicitly a Canny edge detector.

OpenCV library does not implement a Hough transform for ellipse detection therefore the image processing algorithm (Fig. 2.) is also using a Hough transform for circle detection for this purpose. An ellipse can be interpreted as an incomplete circle; there is an assumption that the Hough transform for circle detection can also detect the ellipses. After circular object detection a contour detection is done for all the detected objects. If a contour approximated with the nearest ellipse is successful, the detected circular object is considered to be an ellipse.

3.4 Edge detection

Edge detection algorithms are based on evaluation of the differences in brightness values of neighboring pixels. Image is a two dimensional function and its gradient [9] is computed

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} i_x + \frac{\partial f(x,y)}{\partial y} i_y \qquad (13)$$

where x and y are the coordinates of the point in which the gradient is computed, i_x and i_y are unit vectors at the directions of x and y axes. Since an image is a digital signal $f(n_1, n_2)$, the gradient should be computed using differences instead of derivatives. An example of replacing derivative $\frac{\partial f(x,y)}{\partial x}$ with G_{n1} is showing the equation

$$G_{n1} = \frac{f(n_1, n_2) - f(n_1 - 1, n_2)}{T} . \qquad (14)$$

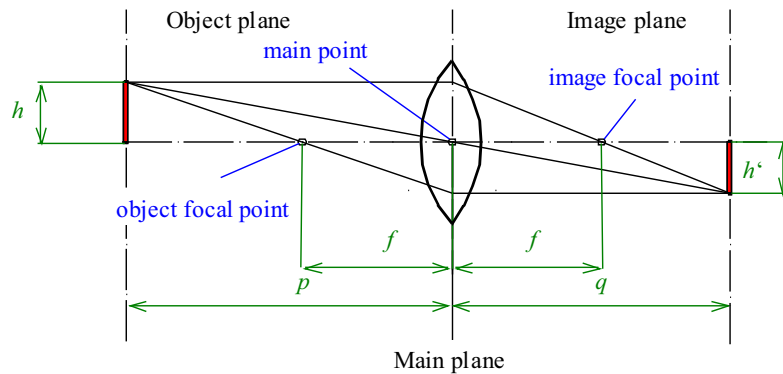OpenCV's Hough transform function uses a Canny edge detector. This detector uses a convolution of an

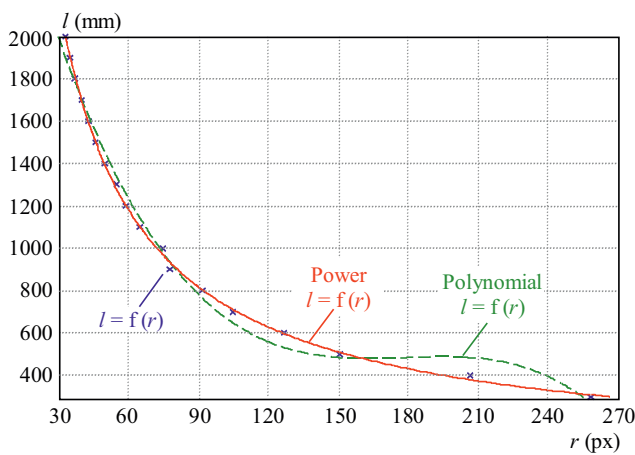**Fig. 10.** Projection of an object with lens

Fig. 11. Polynomial and power approximation of the measured data

Table 1. Dependence of the distance of the mark from the mobile robot (Dist) on its radius (Rad) in the image of the camera

Rad (pixels)	36	38	40	43	46	49	53	58	62
Dist (mm)	2000	1900	1800	1700	1600	1500	1400	1300	1200
Rad (pixels)	68	78	81	95	108	130	154	210	262
Dist (mm)	1100	1000	900	800	700	600	500	400	300

image with an operator G_n which is the first derivative of a two dimensional Gaussian G in some direction n

$$G_n = \frac{\partial G}{\partial n} = n \cdot \nabla G \,. \tag{15}$$

4 MOBILE ROBOT NAVIGATION

After the image processing part the parameters of all detected marks are known (shape — circular or elliptical, radius or radiuses in the image of the camera, position of

the center in the image of the camera). These data are used for the relative position computation between the mobile robot and the marks. After position estimation the trajectory planning part is applied.

4.5 Mobile robot and mark relative position determination

The task is to compute the position (d_{xp}, d_{yp}) between the centre of the mobile robot and the mark (Fig. 9) from the known data obtained from the image processing part.

For d_{yp} computation the laws of optics could be used. The radius of the circle in the image of the camera depends on the distance d_{yp} of the mark from the center of the mobile robot.

A projection of an object to the image of the camera is shown on Fig. 10., where f is the focal length, p is the distance between the object plane and the main plane, q is the distance between the image plane and the main plane, h is the real height of the object and h' is the height of the object projected to the image of the camera.

Using the laws of optics, h' can be calculated [8]

$$\frac{1}{q} + \frac{1}{p} = \frac{1}{f} \,, \tag{16}$$

$$\frac{h'}{h} = \frac{q}{p} = \frac{q - f}{f} \,, \tag{17}$$

$$h' = \frac{h}{(p/f) - 1} \,. \tag{18}$$

Equation (18) is describing the dependence of the radius of the mark in the image (h') on the distance of the mark from the camera (or from the mobile robot). This equation is not linear. The needed parameters of the camera (such as focal length) are not always known therefore the dependence described with the equation (18) should be measured for the used camera.

The measurement was done by placing the mark in front of the camera (the way that the z axis of the camera was passing through the center of the mark) and changing
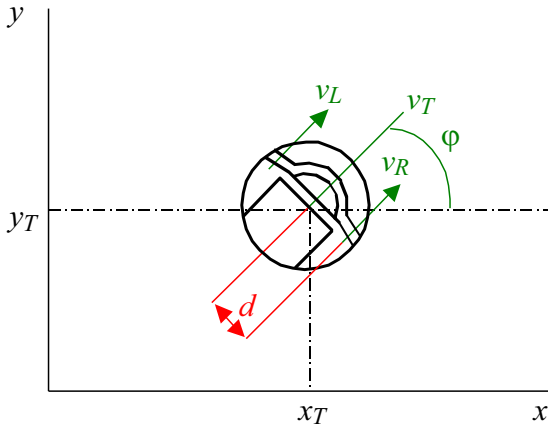
Fig. 12. Two wheeled mobile robot kinematics

the distance between the mark and the camera from 2 m down to 300 mm with step 100 mm (Table 1).

The measurement from Table 1. has to be approximated with a function for the ability of distance computation also between the measured points in this table. Different types of approximations were done, but the power approximation fits the best (Fig. 11.).

According to Fig. 11. the dependence of the distance of the mark from the mobile robot on its radius in the image (for a concrete camera) is

$$y = 58958x^{-0.9434}. \qquad (19)$$

Using (19) and substituting the radius of the recognized mark (radius of the detected circle with Hough transform) in *pixels* instead of x, the distance between the mark and the mobile robot d_{yp} (on Fig. 9.) in mm can be computed.

For computation of d_{xp} (Fig. 9.) similar triangles can be used. Let (x_0, y_0) be the coordinates of the center of the detected mark (circle) in the image, r_o the radius of the mark in the image and x_{co} the center of the image in the x axis. The ratio between the distance of the mark from the y axis in the environment and its real radius is the same as the ratio between the distances on the x axis of the center of the mark in the image from the center of the image on the x axis

$$\frac{d_{xp}}{r_p} = \frac{x_o - x_{co}}{r_o}, \qquad (20)$$

$$d_{xp} = \frac{x_o - x_{co}}{r_o} r_p. \qquad (21)$$

4.6 Mobile robot trajectory planning

If the distance between the mobile robot and the nearest mark is greater than a desired threshold then the mobile robot is moving straight to the mark with an effort to hold the center of the mark (in the image) in the center of the image.

If the distance reaches the threshold, the mobile robot starts a movement over a trajectory described with a Bézier curve. This trajectory is computed the way that

the end point of the curve is situated on the needed side of the mark at the same distance as is the distance of the mark from the mobile robot d_{yp}.

Quadratic Bézier curves are used for path planning. A quadratic Bzier curve is a path traced by he function $C(t)$, given points P_0, P_1 and P_2 where P_0 and P_2 are the start and the end points and P_1 is the control point

$$C(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2, \quad t \in \langle 0, 1 \rangle \quad (22)$$

Substituting the coordinates of the points, this equation can be rewritten

$$\begin{aligned} C_x(t) &= (1-t)^2 x_0 + 2(1-t)t x_1 + t^2 x_2, \\ C_y(t) &= (1-t)^2 y_0 + 2(1-t)t y_1 + t^2 y_2. \end{aligned} \qquad (23)$$

To make the mobile robot follow a desired trajectory, the trajectory should be divided into time intervals and the velocities of the wheels of the mobile robot should be calculated in each of these intervals. In other words in the first step the robot should move from a start position $[C_x(t), C_y(t)]$; $t = 0$ into a new position $[C_x(t + \Delta t), C_y(t+\Delta t)]$; $t = 0$, where Δt is a time (sampling) interval which divides the interval $t \in \langle 0, 1 \rangle$. The next step of the mobile robot is similar to the first one, but starts with $t = t + \Delta t$. To make a move from the start position into a new position, the needed velocities of the wheels of the mobile robot should be computed. The length of any Bézier curve is defined with its start, control and the end point, but $t \in \langle 0, 1 \rangle$ for any length of the planned curve. It could be a problem for a mobile robot with a maximal speed $v_{\max}$ to move from a start position to the end position in a time defined by Δt, if the distance is greater than the mobile robot can travel with its maximal speed during this time period. Therefore a transformation of a step on the computed Bézier curve (with time interval Δt) into a real step in Cartesian coordinates (with time interval Δt_r) was done

$$\Delta t_r = \frac{\sqrt{(x_T - x_0)^2 + (y_T - y_0)^2}}{v_{\max}}. \qquad (24)$$

This equation computes the time needed for a mobile robot with a known maximal speed $v_{\max}$ to drive the distance between two points on the desired Bézier curve, where $[x_0, y_0] = [C_x(t), C_y(t)]$; $t = 0$ is the start position and $[x_T, y_T] = [C_x(t + \Delta t), C_y(t + \Delta t)]$; $t = 0$ is the position where the mobile robot has to drive into.

The general equations of the two wheeled mobile robot kinematics [12] can be used for the calculation of the needed velocities of the wheels of the mobile robot in each step
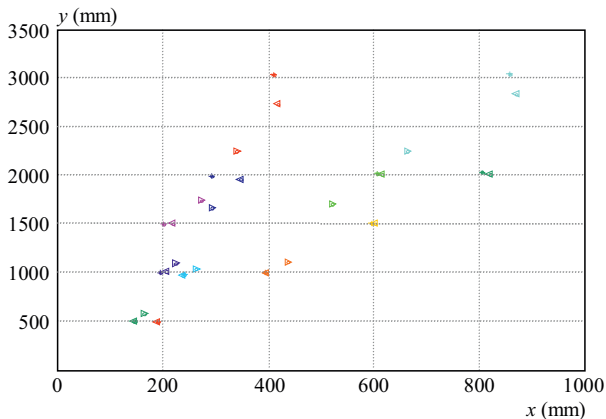
$$v_x = \frac{v_L + v_R}{2} \cos \varphi, \qquad (25)$$

$$v_y = \frac{v_L + v_R}{2} \sin \varphi, \qquad (26)$$

$$v_\varphi = \frac{v_R - v_L}{2d}, \qquad (27)$$

$$\Delta x_T = v_x \cdot \Delta t, \qquad (28)$$

$$\Delta y_T = v_y \cdot \Delta t, \qquad (29)$$

$$\Delta \varphi = v_\varphi \cdot \Delta t. \qquad (30)$$

Fig. 13. The graph of the accuracy of mark's position estimation with one camera

Table 2. List of selected parameters of Hokuyo URG-04LX laser scanner

Parameter	Description
Measuring range	60 to 4095 mm, 240°
Accuracy	60 to 1000 mm: ±10 mm 1000 to 4095 mm: 1 % of measurement
Angular resolution	Step angle 0.36° (360/1024 steps)

Here Δx_T and Δy_T are the differences between the robot's start and end position on x and y axes if the robot moves with the velocities v_L and v_R of its wheels for a time Δt. These differences should be the same as the differences on the computed Bézier curve with step Δt. During the movement of the mobile robot, its angle φ is also changing. The starting orientation and position of the mobile robot on a trajectory generated with a Bézier curve is always assumed to be $[x_0, y_0] = [0, 0]$ and $\varphi_0 = \frac{\pi}{2}$. The robot has to move to position $[x_T, y_T]$ and has to change its starting angle to the new angle φ_T. This angle is computed using a C++ function atan2 which returns the principal value of the arc tangent of $\frac{y}{x}$ expressed in radians (this function computes the value depending on the quadrant)

$$\varphi_T = a \tan 2\big((y_T - y_0), (x_T - x_0)\big). \qquad (31)$$

Substitute the robot's start and end positions and angles into the system of kinematics equations

$$(x_T - x_0) = \frac{v_L + v_R}{2} \cos \varphi_0 \Delta t, \qquad (32)$$

$$(y_T - y_0) = \frac{v_L + v_R}{2} \sin \varphi_0 \Delta t, \qquad (33)$$

$$(\varphi_T - \varphi_0) = \frac{v_R - v_L}{2d} \Delta t. \qquad (34)$$

The needed velocities of the left and the right wheel of the mobile robot are computed as follows (remember to use Δt_r instead of Δt because the two wheeled mobile robot kinematics equations were written in general form)

$$v_L = \frac{1}{\Delta t_r}\left[\frac{(x_T - x_0)}{\cos \varphi_0} - d(\varphi_T - \varphi_0)\right], \qquad (35)$$

$$v_R = \frac{2d(\varphi_T - \varphi_0)}{\Delta t_r} + v_L. \qquad (36)$$

The new position and the angle of the mobile robot are

$$x_T = x_0 + v_x \Delta t_r, \qquad (37)$$

$$y_T = y_0 + v_y \Delta t_r, \qquad (38)$$

$$\varphi_T = \varphi_0 + v_\varphi \Delta t_r. \qquad (39)$$

The next step starts with setting x_0 and y_0 to reached x_T and y_T. Then the computation of new desired x_T and y_T from the Bézier curve equation with step Δt follows.

5 EXPERIMENTS

Experiments were done to verify the accuracy of the position estimation between the detected mark and the mobile robot with visual system. The designed method can determine the position of the mark in front of the mobile robot on x and y axes of the mobile robot's coordinate system using only one camera. If the accuracy of the position estimation would be low, the mobile robot could collide with the mark.

During the experiment the mark was placed in front of the mobile robot into different positions on the x and y axes. The accurate position of the mark was measured with Hokuyo URG-04LX laser scanner [10]. The list of its selected parameters is shown in Table 2.

The accurate position of the mark was compared with the position determined by the visual system. The result is shown in Fig. 13. Every measured position of the mark is shown in this figure with different color and the accurate position of the mark determined with laser scanner is marked with a star. For every position of the mark a series of 100 measurements with visual system was done. The best and the worst position estimation with visual system for every position of the mark were noted in the graph. The worst position estimation is marked with right-triangle and the best position estimation is marked with left-triangle.

In the measurements the error of position determination with visual system towards laser scanner is calculated as

$$e = \sqrt{(x_{\text{visual}} - x_{\text{laser}})^2 + (y_{\text{visual}} - y_{\text{laser}})^2} \qquad (40)$$

where $[x_{\text{visual}}, y_{\text{visual}}]$ is the position of the mark measured with visual system and $[x_{\text{laser}}, y_{\text{laser}}]$ is the position of the mark measured with laser scanner.

The measurement is showing that the accuracy of position determination of the mark with visual system consisting of only one camera is more accurate at lower distances of the mark from the mobile robot. But greater variances of the accuracy of position estimation with visual system start to occur at the distance 3000 mm of the mark from the mobile robot on the y axis. The accuracy of position determination to distances lower than 1500 mm is accurate enough to rely on for the mobile robot to bypass the mark with a planned trajectory with Bézier curve. The safety bypassing distance of the end point of the Bézier curve does not need to be uselessly high.

6 CONCLUSION

In this paper a wireless real time mobile robot navigation application based on circular marks recognition with visual system consisting of only one camera was created. The experiments show that instead of using stereo vision techniques a one camera visual system is accurate enough for the marks position determination relatively to the position of the mobile robot in the environment up to the distances of 1500 mm or 2000 mm between the mark and the mobile robot.

The navigation application is able to recognize circular and elliptical marks in variable light conditions in different types of environments.

References

[1] ZHOU, F.—SONG, B.—TIAN, G.: Bézier Curve Based Smooth Path Planning for Mobile Robot, Journal of Information & Computational Science **8** No. 12 (2011), 2441–24450.

[2] BRADSKI, G.—KAEHLER, A.: Learning OpenCV, Sebastopol: O'Reilly Media, Inc., 2008.

[3] http://www.irobot.com/filelibrary/pdfs/hrd/create/ Create%20Manual_Final.pdf, [online] – cited 29.06.2012.

[4] MORVAN, Y.: Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video. June 9, 2009//, http: www.epixea.com/research/multi-view-coding-thesisse8.html. [online] – cited 12.03.2013.

[5] SHIH-YAO, JUANG—JIH-GAU—JUANG: Real-Time Indoor Surveillance Based on Smartphone and Mobile Robot, 10th IEEE International Conference on Industrial Informatics (INDIN), 25-27 July 2012, 2012, pp. 475–480.

[6] ŠONKA, M.—HLAVÁČ, V.: Computer Vision (Počítačové vidění), Grada, Praha, 1992. (in Czech)

[7] HLAVÁČ, V.—SEDLÁČEK, M. Signal and Image Processing (Zpracování signálu a obrazu): Vydavatelství ČVUT, Praha.

[8] ŠONKA, M.—HLAVÁČ, V.—BOYLE, R. Image Processing, Analysis and Machine Vision: Brooks/Cole Publishing Company.

[9] PAVLOVIČOVÁ, J.—PARTYK, M.—POLEC, J.: Numerical Image Processing (Číslicové spracovanie obrazu), Vydavateľstvo STU v Bratislave.

[10] DUCHOŇ, F.—DEKAN, M.—JURIŠICA, L.—VITKO, A.: Some Applications of Laser Rangefinder in Mobile Robotics, Journal of Control Engineering and Applied Informatics vol 14 No. 2 (2012), 50–57.

[11] HARGAŠ, L.—HRIANKA, M.—KONIAR, D.: Image Processing and Analysis: a Practical Approach, text book, Žilina: University of Žilina, 2008 (CD-ROM).

[12] MIKOVÁ,.—GMITERKO, A.—KELEMEN, M.: Kinematic Structures of one- two- and three-wheels mobile robots (Kinematické štruktúry jedno-, dvoj- a trojkolesových mobilných robotov), ATP Journal No. 2 (2012), 46–48lang Slovak.

[13] ŽIDEK, K.—ŠEMINSKÝ, J.—DOVICA, M.—HOŠOVSKÝ, A.: Wireless Mobile Equipment with Integrated Function of Image Processing (Bezdrôtové mobilné zariadenie s integrovanou funkciou spracovania obrazu), In: Automatizace, regulace a procesy: Sborınk přednášek z 8. technické konference ARaP, Praha, 20. – 21.11.2012., DIMART, 2012, pp. 37–42. (Slovak)

Peter Pásztó born in 1984, received the Engineer degree in cybernetics from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava in 2009. In 2012 he received the PhD degree from the same University and he is working at this time as an assistant. His research and activities are in the areas of visual systems, robotics and automation.

Peter Hubinský born in 1962, received the Engineer degree in technical cybernetics from the Faculty of Electrical Engineering of the Slovak University of Technology, Bratislava in 1985. In 1992 he received the PhD degree from the same University. He specialized further in system theory and drive technology and control of robotic systems and worked as an assistant at the Faculty of Electrical Engineering in this time. Since 1999 he has been Assoc. Professor at the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava. In 2011 he became a a full Professor at the same Faculty. His research and lecture activities are in the areas of servo systems, theory of dynamical systems, motion control systems, robotics and automation.