

The Development and Research of the Indoor Navigation System for a Mobile Robot with the Possibility of Obstacle Detection

Alexander Tyukin, Ilya Lebedev, Andew Priorov,
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
tyukin.alex@gmail.com, ilyaleb@gmail.com, andcat@yandex.ru

Abstract—This work describes a new approach to the mobile robot orientation in space and its obstacle determination. The method uses color beacons placed on the environmental objects for position determination. For obstacle detection the computer vision algorithm is used. All the system functions using one single camera. Both algorithms developed are investigated under different conditions.

I. INTRODUCTION

Nowadays automatic systems are becoming more and more sought-after because of their ability to fulfill routine tasks without user's interference. Some tasks require highly skilled specialists to be drawn into work. For production cost reduction a "trained" robot can be used. Autonomous systems can also be of help when it's necessary to execute huge amount of work on a real-time basis (to discover suspect items in the field) [1]. In many cases computer vision systems already complete tasks of man accompanying and substitution in the business areas related to visual information collection and analysis [2, 3].

Today one of the most popular systems is the system of robot beacons navigation. The global navigation tasks are successfully solved but some problems may arise with mobile robot indoor orientation [4]. It occurs due to the fact that working indoors is characterized by multiple barriers - from illumination unevenness to problems of radio signals reflection. In this case the environment is considered ill-conditioned, with unreliable communication line, fundamental inaccuracy and uncertainty. Thus, the necessity to research orientation methods able to function in a highly noised environment is evident [5].

The modern scientific and technical literature analysis shows that in many cases the most reliable communication channel is the optical one [6]. In such navigation systems beacons with color code are used. They represent a passive device with three fields of different color. Unlike ultrasonic, infrared and laser ones, beacons with color code

are simple to be made and don't require any power-supply sources what lets them stay operable during indefinite time. A usual color digital video camera can serve as a beacon optical signal receiver [7].

The color code beacon navigation system can be used for such tasks as mobile autonomous platforms and robots navigation; operation process automation; quality control; distance, angle and speed measurement; tracing; virtual reality creation; actors' movements and gestures digitization.

Obstacle detection is an important task for modern mobile robots. Nowadays there are a lot of robots relying on transducers and sensors to measure distances to obstacles [8]. However none of these sensors is ideal. Ultrasonic sensors are cheap but have small angular resolution and there's a problem with mirror reflection. Laser distance gauges and radars provide better resolution but are more expensive and complicated. In addition to their individual disadvantages all these methods are not good in detecting small or flat objects lying on the surface. In spite of small objects and different types of bottoming surfaces being difficult to detect by transducers measuring distances, in most cases they can easily be detected using color information [9, 10]. Stereoscopic data processing requires huge compute capacities [11]. For this reason there was developed and researched the computer vision algorithm that distinguishes the surface color attributes using one single camera.

II. NAVIGATION SYSTEM

A. Requirements for the navigation algorithm

The main tasks are to discern the call-off quantity of three-color beacons in the video stream (the colors are set separately); to locate the beacons in space relative to the video camera matrix (in the relational map) and to find the autonomous platform coordinates in the absolute map (Fig. 1).

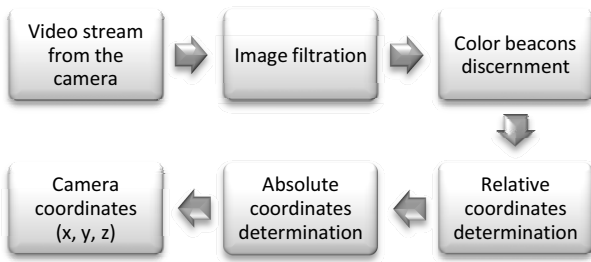


Fig.1. The scheme of the navigation algorithm work

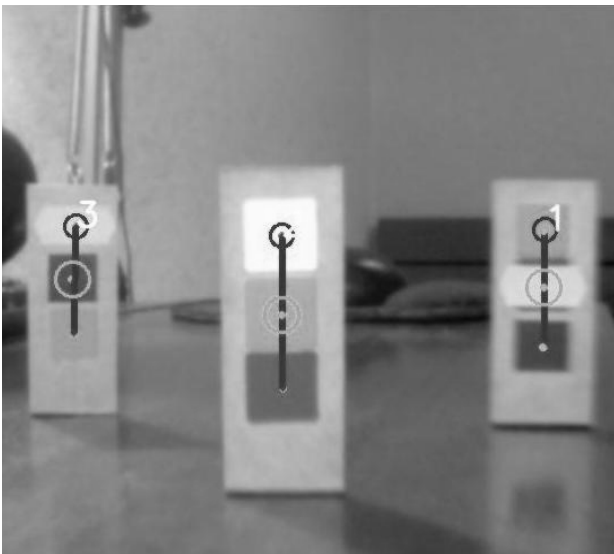


Fig. 2. The example of beacons

The color beacons represent a combination of three color areas allocated in-line in close proximity to each other (Fig. 2). The chosen quantity of color areas is ideal in the context of the algorithm fast response time and detection high reliability. With more of them the response time will be lost, with less of them the detection reliability will.

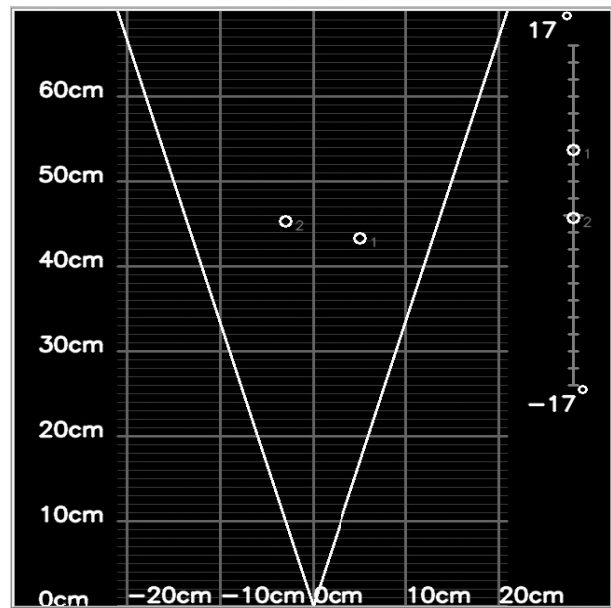
The beacons have following restrictions that define the algorithm working efficiency:

- all three beacon colors should be visually identifiable;
- the color areas centers shall be allocated in one line and equally-spaced from each other;
- the beacons surface should be matt.

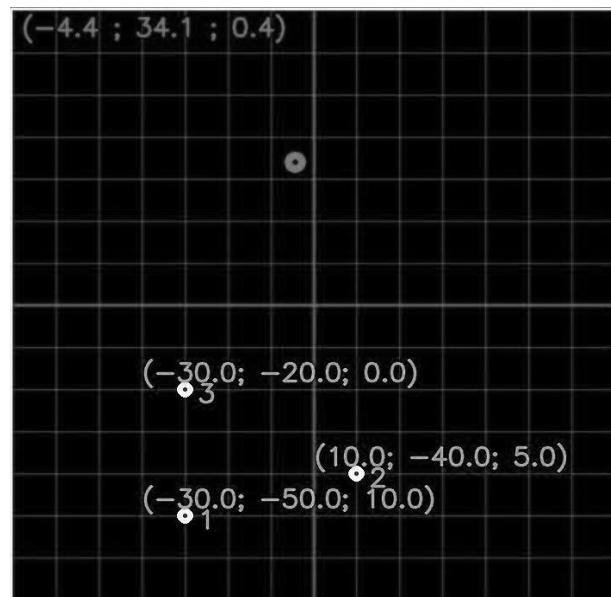
The vertical beacon position is also preferable as the distance between color areas will not change while horizontal movement of the autonomous platform.

B. The navigation algorithm structure

The relational map (Fig. 3a) is an image with color beacons locations indicated by dots. Also the index number and coordinates of each are indicated.



(a)



(b)

Fig. 3 Beacon location (a – relational map, b – absolute map)

The beacon coordinates are calculated by the algorithm in the Cartesian reference system. With reference to them the beacon location in relation to the camera (coordinates x, y) and the vertical level at which the beacon can be seen (coordinate z) are drawn.

For the calculation of the beacon three-dimensional coordinates several values should be set a priori – the beacon height which equals to the distance between the centers of the two outermost color areas and the video camera lens aperture. The height equality of all beacons

that are simultaneously present in the frame is an essential condition.

Another prior parameter is the lens aperture equal to the tangent of the lens aperture half angle empirically measured by the received image horizontal. The measurements should be taken specifically by the image received from the video camera. It helps to receive measurements of a higher accuracy [12].

The absolute map (Fig. 3b) is an image of a room or space plan where the autonomous platform moves. The absolute map is tightly bent to the Cartesian reference system.

Due to three coordinates set a priori for each beacon that characterize its location in space unambiguously, the video camera (mobile platform) coordinates are defined.

To solve this task it's necessary to set the following values:

- 1) beacon quantity;
- 2) three coordinates for each beacon;
- 3) color value for each area of the three-color beacon.

C. The work stages of the navigation algorithm

The following steps can describe the work of the color discernment algorithm:

1) At first each stream video frame is filtered to smooth image defects and eliminate distortions. For this task Gaussian filter with 5x5 mask is used. This filtration method is quite fast and effective to fight distortions in the image.

2) The image is converted from the RGB color model (red, green, blue) into the HSV model (Hue, Saturation, Brightness) to facilitate further image processing as HSV is easier to work with colors.

3) For each HSV channel smooth analog function is used: for saturation and brightness – logistic sigmoid function with scalable parameters of curvature k and bias Δ [13]:

$$f(x) = \frac{255}{1 + \exp(-k(x - m + \Delta))} \quad (1)$$

and for hue – Gaussian curve with scalable dispersion D :

$$f(x) = 255 \exp\left(-\frac{(x - m)^2}{2D}\right) \quad (2)$$

where m is a value of hue, saturation or brightness accordingly, received during the algorithm habituation stage. At the function output we receive three images in grey scale with pixel value from 0 to 255.

The usage of smoothly varying functions instead of threshold functions is explained by stability augmentation and algorithm operation constancy [14].

To boost response time a very important improvement has been made. Instead of calculating the values of logistic sigmoid functions and Gaussian curve for every pixel of the image, the values of these functions are calculated only once and are written into one-dimension array with the length of 256. The number of the array element corresponds to the pixel intensity and the element value – to the threshold function value.

Thus, during one algorithm launch only 3N of arrays are defined, where N is a number of the arrays.

4) Then these three images are multiplied pixel by pixel with the cube root extraction. This action produces a “color mask” for one particular color (Fig. 4a).

5) In the “color mask” the pixel with maximum intensity value is detected and the area around is “filled” by pixels the intensity value of which meets the following condition:

$$1 \leq x \leq x_{\max} + 200 \quad (3)$$

where x is the intensity of this pixel, x_{\max} is the intensity of the pixel with which the “filling” started (pixel intensity is normalized to 255). The usage of this condition and the 8-connected area while “filling” lets us define the areas of one color from another well.

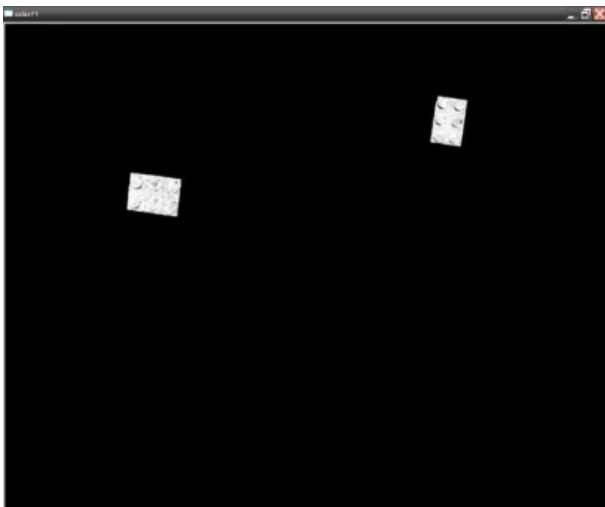
This procedure is repeated several times without taking previously “filled” pixels into account while “filling” new areas. Iterations continue until the total filled area exceeds 90% of the image or until the quantity of the areas of less than 100 pixels is less than five. The manual restriction of the filled areas number for response time boosting is also possible.

6) The coordinates of centers of each filled area are calculated and written into the array.

7) Steps 3) – 6) are repeated for every chosen color (fig. 4b)

8) The next step is to calculate the lengths and angles of the tilts of the vectors connecting the filled areas centers. The total vector quantity $(n_2 \cdot (n_1 + n_3))$ where n with the k -index is a number of k -colored areas found. The total of $(n_1 \cdot n_2 \cdot n_3)$ combinations of two vectors can give us a “skeleton” of the sought three-color beacon (as the beacon consists of three colors, there are two vectors that connect them in sequence).

9) The matching combination of these two can be found by the difference in lengths and angles of two vectors connecting three color areas of the beacon. If these differences are less than the preset threshold then the color areas that correspond to these vectors are marked as the



(a)



(b)



(c)

Fig.4 Color masks (a – of one of the colors, b – of the image, c – with the vectors applied)

sought beacon indicating the beacon orientation (turn) and index number (Fig. 4b).

10) Knowing the video camera lens aperture and the beacons dimensions the beacons location in space is determined (top view and vertical level in relation to the lens visual axis) (Fig. 3a).

11) The autonomous platform coordinates are calculated and its location is indicated in the absolute map. It's achieved due to the three-dimensional affine transformations usage (parallel transfer, turn, scaling) that permits to fulfill transit from the camera coordinates system to the world coordinates. At this stage the algorithm turns to the absolute beacon coordinates set before the algorithm launch.

D. The analysis of the illumination type influence on the navigation algorithm work

The color discernment procedure forms the basis of the algorithm work. The color of the area the video camera lens points to, doesn't depend on the reflecting surface physical aspects only, but also on the impinging light spectral structure. That's why at first the algorithm work quality with reference to the external illumination type was researched.

The dispersion of the beacon geometric center coordinates is measured in (pixel²). The beacon was placed still in front of the camera for the time of the experiment. In the ideal case the dispersion should be equal zero but due to the distortions of the image beacon coordinates are of random nature.

The working scene was sequentially lighted by five different light sources of correspondingly different spectral structure of the light emission.

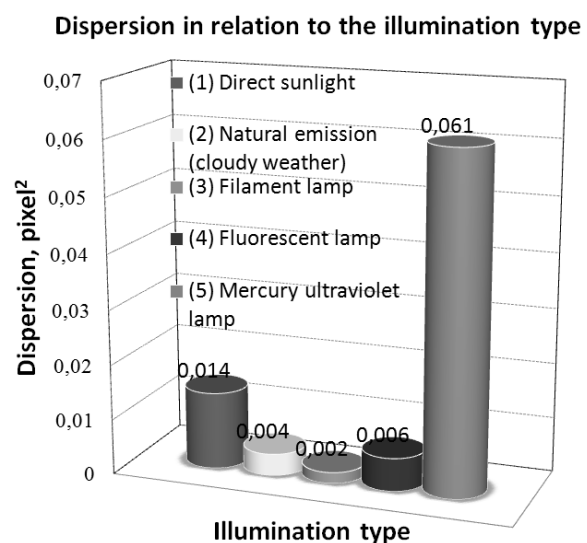


Fig.5 Dependence of beacon dispersion coordinates on the illumination type

The experiment used direct sunlight, natural emission (cloudy weather), filament lamp, fluorescent lamp, mercury ultraviolet lamp. Fig.5 shows the dependence of beacon dispersion coordinates on the illumination type.

The histogram shows that the worst results were received while using mercury ultraviolet lamp, fluorescent lamp and direct sunlight. It can be explained by the limitations of the lamp light spectral structure and glare of intensive sunlight. The best results were received while using filament lamps and natural emission. These light sources give steady luminance and wide range of the radiated light that guarantees high values of color purity. But for all light sources dispersion values are not that high and the algorithm preserves its working capacity.

E. The analysis of the luminance influence on the navigation algorithm work

Another external parameter that characterizes the scene under research is luminance measured in lux (lx). At low values of luminance the video camera matrix isn't able to recognize colors and it switches to the grayscale mode. In this mode only the information about pixel intensity is transferred, but not about color. Vice versa, at very high values of luminance the image can get light exposed and the color information will be lost.

The dependence of beacon coordinates dispersion on the luminance of the scene under research is shown in fig. 6.

The diagram shows that while the luminance is decreasing the coordinates dispersion is rising, and at the luminance of about 22 lx the algorithm comes out of action.

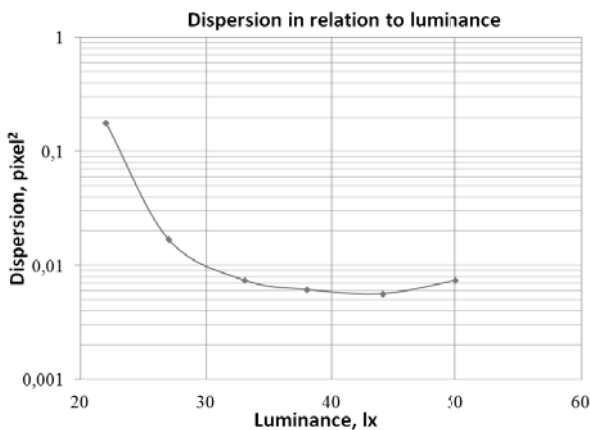


Fig.6 The dependence of beacon coordinates dispersion on the luminance

F. The analysis of the beacon – camera distance influence on the navigation algorithm work

Due to the mobility of the platform on which the given algorithm is planned to be used, the distance between camera and beacons can strongly vary. This factor is very important because if the distance increases the beacon relative area in the image will decrease. If the space

occupied by some color area will get less than 100 pixels it will be classified as “rubbish” and won't participate in discernment.

Thus, the maximum distance till the beacon as a parameter of the algorithm working capacity is relative. It depends on the beacon size (its height). The experiment used beacons of 32 mm high.

To define the absolute value we need to divide the minimal area of the beacon which still will be discerned, by the whole image area. The minimal relative area of the beacon is 300 pixels (the beacon consists of three color areas). The whole image area is $640 \cdot 480 = 307200$ pixels. Thus, if the beacon occupies the image part of $300 : 307200 = 1 : 1024$ it will be discerned.

The dependence of the beacon coordinates dispersion on the video camera – beacon distance is shown in the fig.7.

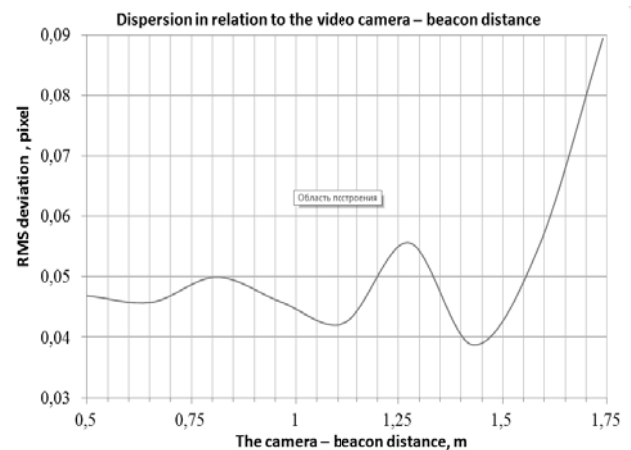


Fig.7 The dependence of the beacon coordinates dispersion on the video camera

III. THE OBSTACLE DETECTION ALGORITHM

The navigation algorithm described above lets the robot orientate in space but to make it move more confidently the obstacle detection ability is needed. For this there was developed a new system that discerns surface color attributes using one single camera (Fig. 8).

The core of the obstacle detection algorithm is discerning pixels that differ by color from the bottoming surface and classifying them as obstacles. The algorithm works on a real-time basis in different conditions providing images of high definition at the output. The system is also easily taught.

A. The conceptual description of the obstacle detection algorithm

The developed obstacle detection scheme is entirely based on the freestanding pixel outside appearance. Any pixel that's different from the ground by its outside appearance is classified as an obstacle.

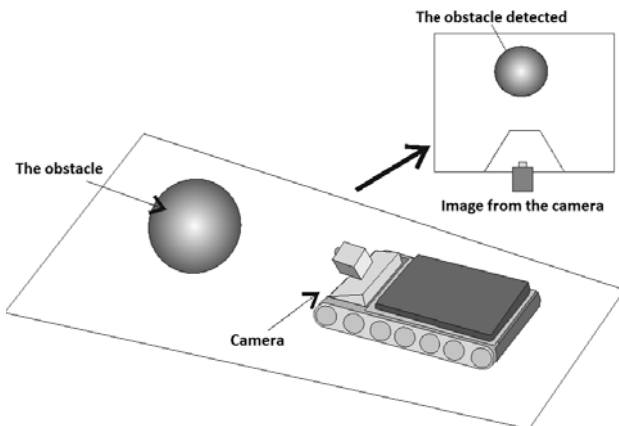


Fig. 8 The offered scheme of the algorithm work

The method is based on three suppositions that are reasonable for different internal and external conditions:

- 1) Obstacles differ from the ground by their outside appearance.
- 2) The ground is relatively flat.
- 3) There are no overhanging obstacles.

The first supposition lets us distinguish obstacles from the ground and the second and the third ones let us estimate the distance between the detected obstacles and camera.

For many applications it's important to estimate the distance from the camera to the pixel that represents an obstacle. With monocular vision the general approach to the distance estimation consists of the supposition of the ground being relatively flat and no overhanging obstacles present. If these two suppositions are right the distance is a monotonically increasing function of the pixel height in the image.

B. The technical implementation of the obstacle detection algorithm

To start work we need to receive an incoming image: it can be video sequence or just a static image.

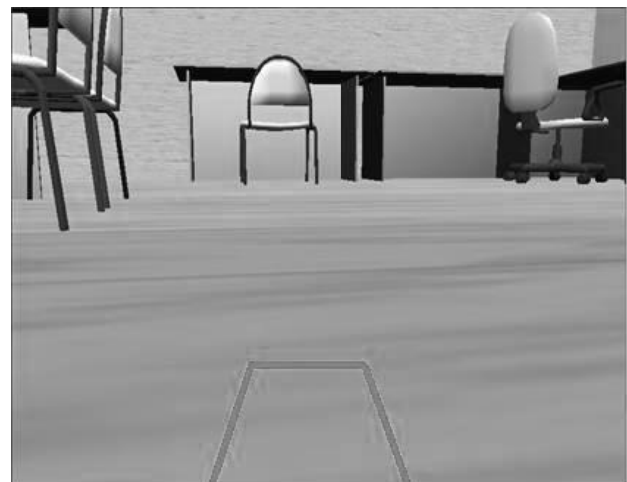
The next step is image filtration. Filter is a usual scheme of decimation and interpolation (lower pass filter implementation, odd information removal, signal enhancing by blank readings and interpolation using lower pass filter). The usage of this filter is conditioned by the fact that after its implementation the areas similar in color will be supremely homogeneous and it is essential for the clustering tasks. [15]

The filtered image can be transferred to the HSV color system. The way this transition influences the algorithm work will be shown later.

For the analysis of the bottoming surface with no obstacles in the image the trapezoid is drawn. This procedure is shown in the Fig. 9.

The trapezoid area is divided into 3 clusters using the K-means algorithm. On the basis of each cluster we create a taught model (on its basis the system will be taught for the following environment exploration). This model will include:

- a number of trapezoid pixels that got into the given cluster;
- the percentage of the pixels of the cluster in relation to the total number of pixels;
- cluster covariance matrix;
- the average value of the pixels of the cluster by corresponding color components.



(a)



(b)

Fig.9 Trapezoid drawing

For all image pixels we calculate the distance of the mahalanobis d to the closest researched model (researched model is a color model of the bottoming surface that has been accumulated during several frames). For each cluster of the trapezoid we calculate the average distance of the mahalanobis d' for all dots belonging to the cluster. Then the pixels not forming obstacles, are marked. If the

following condition is fulfilled the pixel is considered bottoming surface, otherwise – an obstacle:

$$d - d' < \tau \tag{4}$$

Then the researched models are renewed using training models.

C. The analysis of two obstacle detection methods

The survey concerned two modifications of the algorithm of the obstacle detection at the bottoming surface: in RBG and HSV color systems. The dispersion of the number of the dots discovered as bottoming surface was used as the work criteria. The experimental algorithm research has shown the less this value is the more reliable obstacle detection will be.

The algorithms were analysed by the following parameters:

- 1) Dependence on the Mahalanobis distance.
- 2) Dependence on the input image contrast ratio.
- 3) Dependence on the input image distortions.

The dependence of the results on the Mahalanobis distance is represented in the Fig. 10. As it's shown the function has absolute minimum and local minimums. Subsequently, to define the Mahalanobis distance that gives the best discernment automatically it's necessary to plot the whole chart and it takes a lot of time (to define the dispersion not less than 15 frames are needed that equals to 1 sec at the layout speed FPS = 15, so plotting a chart will take 10 sec.). It should be noted that the algorithm of image processing in RGB is of a less dispersion at the minimum point.

We have evaluated the distortion influence on the detection correctness. The results dispersion dependence on the peak signal to noise ratio is shown in the table 1. The experiment data show that at moderate distortion the algorithms work at the quite same level. But at the higher distortion we see a big difference in the dispersions HSV and RGB (the RGB dispersion is a sequence higher).

The results dispersion dependence on the image contrast ratio is represented in the table 2. The contrast ratio has been calculated using the formula:

$$c = \frac{2\sigma}{Y_{\max}}, \tag{5}$$

where

$$\sigma^2 = \frac{1}{N} \sum_{p=1}^N \left(Y_p - Y \right)^2 \tag{6}$$

The chart shows that the dispersion minimum falls to the contrast ratio 60. It means that it's possible to change the contrast ratio of the input image deliberately to achieve the

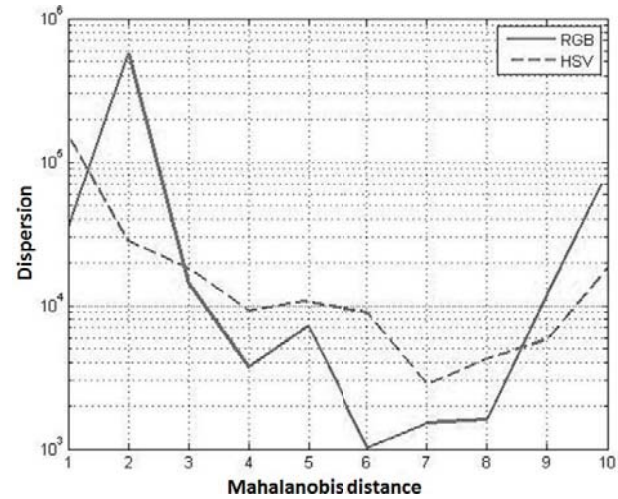


Fig. 10 The results dispersion dependence on the Mahalanobis distance

TABLE I. THE RESULTS DISPERSION DEPENDENCE ON THE PEAK SIGNAL TO NOISE RATIO

N ₀	PSNR (dB)	D (RGB)	D (HSV)
1	∞	1,5 * 10 ³	2,6 * 10 ³
2	47	5,6 * 10 ³	7,3 * 10 ³
3	38	2,6 * 10 ⁴	1,8 * 10 ⁴
4	30	2,7 * 10 ⁵	1,2 * 10 ⁵
5	25	2,4 * 10 ⁶	1,5 * 10 ⁵
6	20	6,2 * 10 ⁶	2,1 * 10 ⁵
7	14	3,4 * 10 ⁷	1,5 * 10 ⁶

TABLE II. THE RESULTS DISPERSION DEPENDENCE ON THE INPUT IMAGE CONTRAST RATIO

N ₀	C	RGB (*10 ³)	HSV (*10 ³)
1	5	3,4	6,3
2	10	3,3	5,8
3	20	3,2	5,5
4	40	2,7	4,5
5	60	2,3	3,9
6	80	3,9	4,7
7	90	4,1	7,6

dispersion minimum. The difference between the dispersions of RGB and HSV differ nominally but the HSV dispersion indeed is higher.

The noise influence on the detection correctness has been evaluated. The experiment proved that at moderate distortion the algorithms work at the quite same level. But at the PSNR < 30dB there's a huge difference between HSV and RGB dispersions (the RGB dispersion is a sequence higher).

IV. CONCLUSION

In the present work the algorithm of the autonomous mobile platform navigation has been developed. Beacons with color coding were chosen as "benchmarks" for the navigation system. The computer vision system discerns the beacons mentioned above basing only on the color component of the incoming video sequence.

The system work analysis has shown that the computer vision algorithm is sensitive to the luminance and outside lightning conditions. It was determined that the computer vision algorithm is capable to discern the beacons correctly only at the luminance higher than 22 lx. The preferable light sources are the following:

- filament lamp;
- daylight, without direct sunlight.

While studying the camera – beacon distance influence it was found out that the algorithm can detect the beacon only at the distance not bigger than 1,7 m. but the experiment was carried out using the beacons 32 mm high and the distance mentioned above is only a relative value. To enlarge the system operation range it's necessary to enlarge the beacon size. Subsequently the beacon height should be chosen with reference to the assigned task scales.

While developing and studying the obstacle detection algorithm the following steps were taken:

- 1) Implementation of the algorithms of obstacle detection and avoidance by mobile platform on the basis of bottoming surface discernment using RGB and HSV images.
- 2) Creation of the specific virtual environment for the computer vision algorithm analysis without hardware tools.

- 3) Analysis of the outcoming data of the implemented computer vision algorithms, comparative analysis of the implemented algorithms.

During the obstacle detection algorithm analysis it was shown that there exists such a Mahalanobis distance at which the system works optimally. The research of the noise influence on the algorithm has shown that at PSNR > 38 dB the algorithm using the HSV color scheme is preferable and at PSNR < 38 dB it's preferable to use RGB.

REFERENCES

- [1] O.P. Kooznetsov, "Artificial intellect and cognitive sciences", *Informational measuring and managing systems*, 2013, № 5, p. 16-24 (In Russian).
- [2] L. Shapiro and J. Stockman. M., *Computer vision*. 2006.
- [3] A. Gonzalez and R. Woods *Digital image processing*. 2005.
- [4] A.V. Ivanov "Navigation systems of mobile surface facilities. Algorithms of information processing in the angle channel", *Radiotekhnika*, 2013, №4 (In Russian).
- [5] A.A. Selyaev and B.A. Alpatov, "Algorithm of the object location estimation in two-dimensional image", *Izvestiya vuzov. Priborostroenie*, 1988, №5, p. 3-5. (In Russian).
- [6] B. A. Alpatov, A.A. Selyaev and A.I. Stepashkin "Digital image processing in the task of moving object tracking", *Izvestiya vuzov. Priborostroenie*, 1985, №2, p.39-43. (In Russian).
- [7] A.I. Baibakov, V.M. Garbooz, V.I. Lobeiko, F.P. Stepanov and N.N. Tarasenko "Methods of calculating parameters of air objects relative position", *Radiotekhnika*, 2007, №6 (In Russian)
- [8] S. V. Krooglikov "Scientific method device of justification of promising directions of automated managing systems development", *Informational measuring and managing systems*, 2013, № 12, p. 44-51 (In Russian).
- [9] N.J. Nilsson *Shakey the Robot*. TechnicalNote: 1984.
- [10] H.R. Everett *Sensors for Mobile Robots: Theory and Applications*. Massachusetts. 1995.
- [11] A. A. Novokreschenov and V.P. Khranilov. "Hardware-software means of improvement of calculative resources usage efficiency in fully exhaustive tasks of automated elements placing", *Informational measuring and managing systems*, 2013, № 7, p. 77-80 (In Russian).
- [12] B.A. Alpatov, "Evaluation of moving object parameters in the sequence of changing two-dimensional images", *Avtometriya*, 1991, №3, p. 21-24. (In Russian).
- [13] O. Faugeras, L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin and Zoghلامي. *3-D reconstruction of urban scenes from image sequences*. *Comput. Vision and Image Understanding*. 1998. p. 292-309.
- [14] P.A. Bakoot, G.S. Kolmogorov and I.E. Vornovitskiy "Image segmentation: Thresholding methods", *Zarubezhnaya radioelektronika*, 1987, №10, p. 6-24. (In Russian).
- [15] A.L. Priorov, V.V. Khryashev, M.N. Golubev. "Removal of impulsive noise with impulses random values from the images", *Radiotekhnika*, 2010, №5 (In Russian).