Image Thresholding and Contour Detection with Dynamic Background Selection for Inspection Tasks in Machine Vision

K. Židek, A. Hošovský

***Abstract* -** The paper deals with a new method of thresholding especially for machine vision systems. This new method is based on creating dynamic background for machine vision tasks. The main advantage is resistivity to: non-homogenous environment light, surface texture failure with same color as background and shadows which can disrupt the recognized edge and shape contour. We can reach precise contour of recognized objects by this technics. The solution is primarily aimed for recognition tasks in automatized lines with conveyor belts.

***Keywords* -** machine vision, thresholding, contour detection.

I. Introduction

THIS document introduces new approach to image thresholding in machine vision systems for usability in automatized production lines. This new method is based on creating dynamic background by bottom lighting passing through transparent belt. The method is primarily intended for flat parts with small and middle height. This approach can be used for precise shape inspection in engineering industry for example to inspect sheet metal blanks (cutting, low profile extrusion), fasteners, covers …etc. The next usability is expected in food industry for example to check borders quality of flat pastry products. The first chapter of presented paper deals with methodology of thresholding technics and methods which are currently used in industrial machine vision systems. Next chapter introduces main principle of a new approach of thresholding based on dynamic background with detailed description of the used algorithm. The next section describes hardware and software solution of the especially designed for dynamic background system based on embedded systems. The last chapter shows some experiments with suggested method for precise contour detection with ideal reference part and part with some surface errors (defects). Resume contains simplified algorithm for detection object contour only and comparison of introduced thresholding method with other standard methods (OTSU thresholding and adaptive thresholding methods …etc.).

K. Židek, A. Hošovský is with the Technical University of Kosice, Faculty of Manufacturing technologies with a seat in Presov, Department of Mathematics, Informatics and Cybernetics, Bayerova 1, 08001, Prešov, Slovakia, phone: +421 55 602 6422; e-mail: kamil.zidek@tuke.sk, alexander.hosovsky@tuke.sk. The research work is supported by the Project of the Structural Funds of the EU, Operational Program Research and Development. Research and development of the intelligent non-conventional actuators based on artificial muscles, ITMS code: 26220220103.

II. Thresholding, contour detection and problems

A. Basic thresholding methods

Thresholding is the basic segmentation method. Basic thresholding can be divided to three methods: binary, truncate and threshold to zero. These methods can be used with inverting function. The advanced technics of thresholding are band and multispectral thresholding [1]. Thresholding usually uses grayscale image. In special cases if we cannot extract the data from grayscale image, the multispectral threshold is used. Multispectral image is created from color image with three separate channels (R-Red, G-Green, B-Blue). The next development in another thresholding methods are described [2], [3].

B. Dynamic thresholding: OTSU, Adaptive thresholding

An OTSU method is based on histogram processing to find minimum between foreground and background [4]. The main condition is bimodal histogram, otherwise algorithm fails. If we cannot ensure homogenous lighting we can use adaptive thresholding methods [5]. Adaptive thresholding creates separate threshold value for every segment in the image. The main disadvantage of this method is very noisy output which is not so simple to filter with standard methods [6]. Thresholding must be optimal, because it is a basic step in recognition process and generates the first input for all other postprocessing filters.

C. Contour detection

Contour detection is very strongly tied with optimal thresholding. Contour detection extracts quantity and shape of objects detected in image. The next task of contour detection is description of object hierarchy inside the object (holes, defects, cuts …etc.) [7], [8].

The next development in thresholding methods continues with the combination of standard methods with advanced technics for example: wavelet transformation, pyramid processing [9], [10], [11], [12] or methods based on artificial intelligence, for example: fuzzy logic, neural networks and genetic algorithms [13], [14], [15], [16].

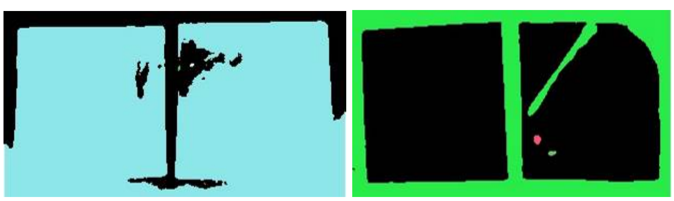
D. Typical problem in thresholding and contour detection

For optimal thresholding in industrial machine vision we must choose ideal background color which is on opposite side scale as recognized object color. The histogram must be bimodal for optimal thresholding. This approach will work only for one type of parts texture in conveyor belt. If we must recognize more parts with different surface color on opposite side of scale, this approach fails. Another example is bad selected background for recognition process. The Fig. 1a shows wrong chosen background. The next problems of standard thresholding are parts with more surface colors (defects, assemblies …etc.) Fig. 1b.

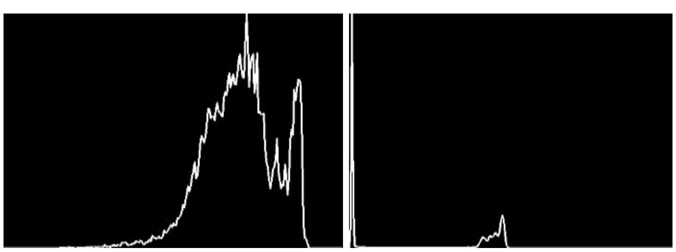
a) b)

Image grab with two opposite color background

OTSU Thresholding



Contour detection



Unusable grayscale histograms (are not bimodal)

Fig. 1 Example of image recognition errors: white/black background, OTSU thresholding, contour detection and histograms.

The higher objects create shadows around edges by side surfaces on lighter backgrounds. Fig. 2 shows problems with shadows generated around object edges. If basic thresholding is not optimal and creates inaccurate edges other filters cannot remove these errors, which pass to next image processing (canny edge filter, contour detection).

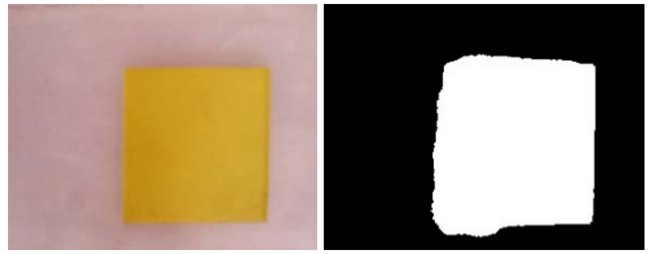
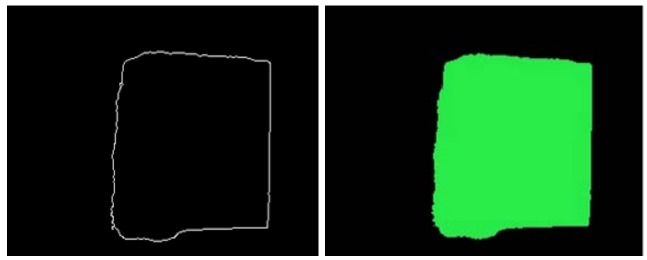


Image grab OTSU thresholding



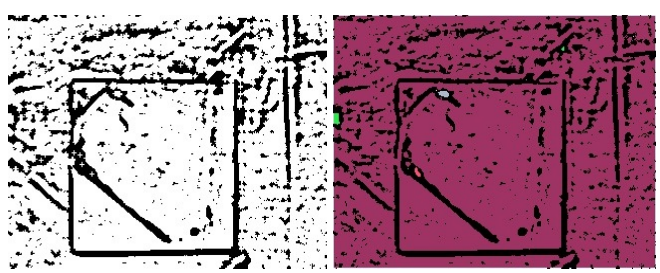
Edge detection Contour detection

Fig. 2 Image of recognizing errors with shadows.

If we cannot ensure homogenous upper lighting to camera view area, possible solution is usability of adaptive thresholding algorithm. We are tested two basic adaptive filter thresholding:

- Adaptive thresholding combined with mean filter
- Adaptive thresholding with gauss filter

The edges detected by first adaptive filter were without shadow distortion, but both filters create significant noise. The created noise was irregular which is hard to remove by smooth or other advanced filters for example Wiener or Kuwahara …etc Filters. Machine vision algorithm for next inspection of error detection considers the generated noise as surface defects. The next problems were discontinuous edges. The contour detection algorithm requires only closed shapes or surfaces for correct function. The first experiment with adaptive thresholding with mean filter and contour detection is shown on the Fig 3.



a) b)

Fig. 3 Adaptive thresholding with mean filter.

The experiment with adaptive thresholding with gaus filter and contour detection is shown on the Fig 4.

Gauss filter significantly removes created noise but in other side degrades edges continuous for contour detection. The result of this experiments is that adaptive filtering cannot be used for machine vision task where is surface errors detection as next inspection task.

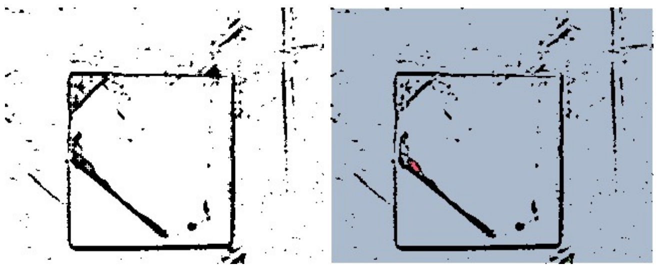


Fig. 4 Adaptive thresholding with gauss filter.

III. CURRENT METHODOLOGY OF THRESHOLDING IN MACHINE VISION

Optimal thresholding is the most important step in image processing especially in machine vision tasks.

Current status in machine vision design process is: Application Engineer usually selects manually the color of the background in vision module according surface color of recognition part.

The main problem in standard thresholding in machine vision is, if:

- the color of the object changed slightly shade during production,
- environment light is changeable,
- environment light is not homogenous,
- sometime it is necessary recognize more parts with different surface colors.

Our solution: create dynamic background in machine vision task by low light RGB system.

The problems with thresholding are solved using the proposed algorithm with dynamic background color according color of parts surface. The detailed description of algorithm is specified in next chapter.

IV. PRINCIPLE OF SOLUTION

The main principle of solution is dynamic background color selection. Optimal distance D_{max} is maximum distance from object color to color of RGB cube corner. Other two distances d are alternative solutions which can replace optimal distance in special cases. Other color systems for example HSL and HSV is not suitable, because doesn't contains black color which is one of the most used color for background in machine vision systems.

The Optimal distance is calculated from Euclidian distance for every corner in RGB cube color space. RGB color spaces have eight basic colors in corners. The highest value of the distance represents optimal background. Example of optimal background color selection for yellow color surface is shown in Fig. 5.

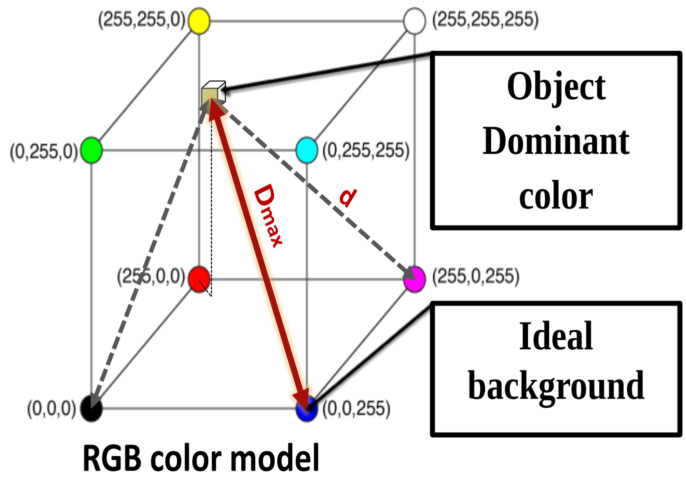


Fig. 5 Background color selection scheme.

- Example color of background
Dmax – ideal background color
d – optional alternative background

Background color selection (cube corners):

- Basic color Red, Green, Blue,
- Color White (RGB), Black by raster deflector,
- Combination of basic color Yellow, cyan, magenta.

The visual example of distance calculation color distance d (D_{max}) in Cartesian coordinate system by Euclidian distance is shown on the Fig. 6.

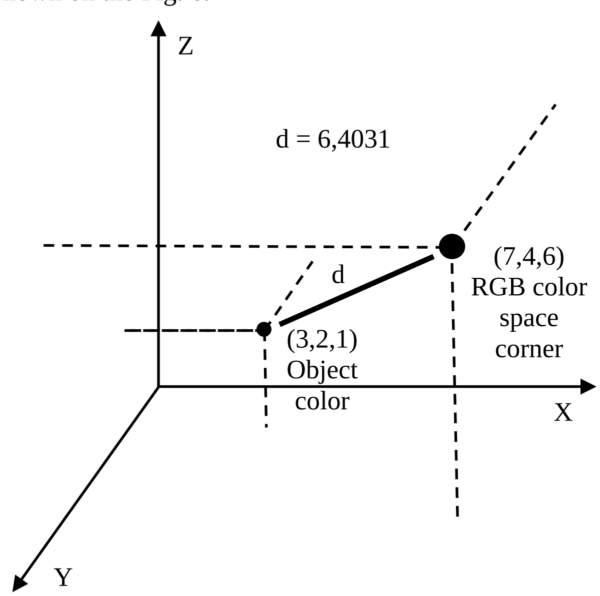


Fig. 6 Cartesian coordinate system for optimal distance calculation.

Maximum distance from dominant color to background color in RGB color space is calculated in two steps:

a) euclidian distance must by counted 6 times for every cube corner color,
b) background color is selected from highest d value.

Calculating optimal background color is based on Euclidian distance equation (1).

$$d = \sqrt{\underset{B}{(x_2 - x_1)^2} + \underset{R}{(y_2 - y_1)^2} + \underset{G}{(z_2 - z_1)^2}} \qquad (1)$$

We can save energy consumed by RGB bottom and top lighting during image processing and delay required by FPS requirement by pulse light.

Time delay diagram for RGB dynamic background lighting and image thresholding is shown on the Fig. 7.

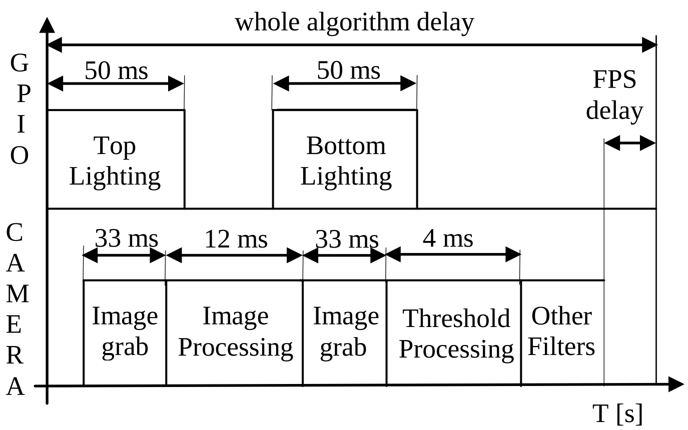


Fig. 7 Time delay diagram of thresholding with dynamic RGB background.

Principle of vision recognition process consists from five basic steps:

- detection of object surface color by identification of dominant color [17],
- computation of background color from RGB Color space,
- OTSU thresholding for dynamic threshold selection,
- contour creating with generating of contour shape mask,
- merging images, combine texture image to contour mask.

Flowchart diagram for image thresholding with relevant delays is shown on the Fig. 8.

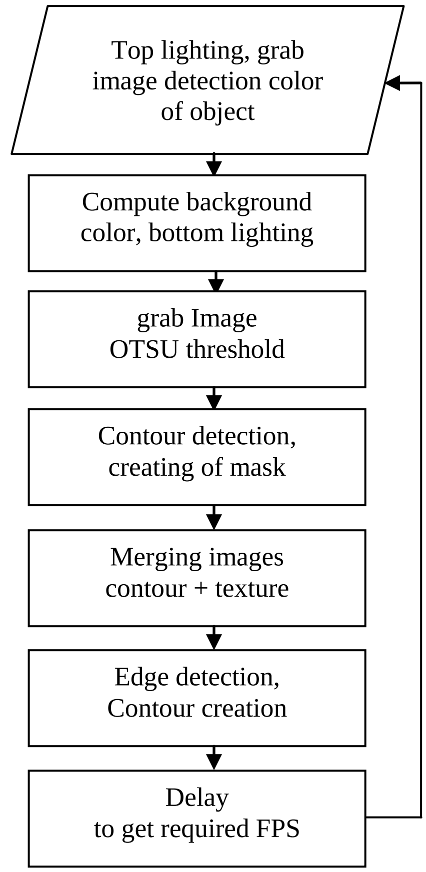
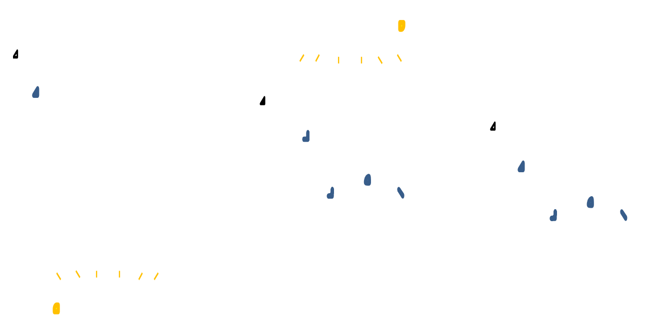


Fig. 8 Flowchart diagram of dynamic thresholding and contour detection.

Basic principle of thresholding, contour detection and extraction is shown in Fig. 9a, top surface texture is extracted in Fig. 9b, and merging of images is shown on the Fig. 9c.



a) Acquiring contour b) Capturing of texture c) Merging

Fig. 9 Main principle of dynamic threshold method.

Our algorithm assume dynamic light environment in these conditions it is possible to use manual thresholding. We conducted successful experiment with OTSU algorithm.

Thresholding in ideal bimodal histogram is reliable with OTSU Threshold algorithm, which consists from:

a) finding the local extremes,
b) Select threshold value in between acquired local extremes.

Example of ideal histogram with threshold selected by OTSU algorithm is shown on Fig. 10.

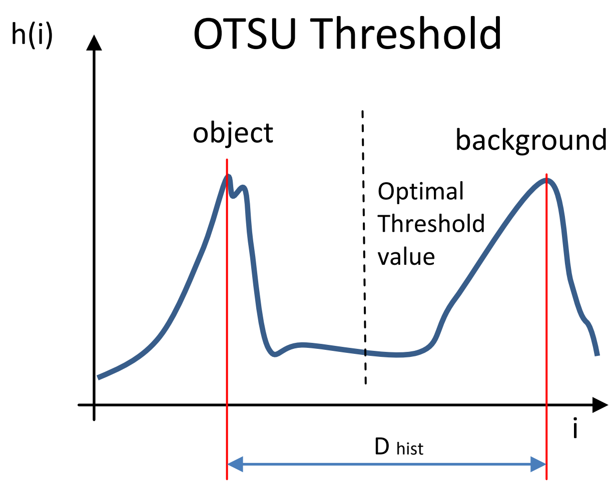


Fig. 10 OTSU thresholding principle.

V. Hardware

Introduced vision system is not fixed to any computational device or hardware platform. The developed software can be used in Wireless Access Point with Linux (MIPS), embedded board (ARM) or any Personal Computer x86(x64) architecture based.

Expected combination of devices with camera interfaces is shown on the Fig. 11.

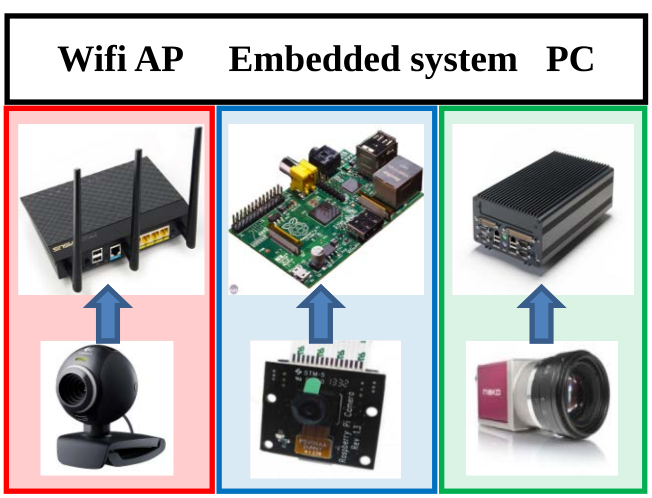


Fig. 11 Hardware of vision system with conveyor belt.

We can use any camera with basic communication interfaces (CMOS or CCD):

- USB 2.0, (USB 3.0) CAMERA
- CSI CAMERA
- FIREWARE CAMERA

Each camera creates different sensor and optics distortion which can be reduced by software calibration which is described in next software section.

Hardware used for bottom lighting thresholding and contour recognition was based on embedded platform based on ARM architecture. The algorithm was ported after testing on x86 (x64) architecture to Raspberry PI embedded board with two connected cameras (USB 2.0 web and CSI video camera).

The system consists from two lighting systems: top white/RGB/InfraRed/Ultraviolet lighting and bottom RGB lighting.

Bottom lighting creates color background thru carrier Plexiglas part with chessboard deflector and transparent belt. We can generate background dynamically by combination of basic colors with PWM signal. The intensity of the lighting can be regulated too. The complete vision system for image processing based on Raspberry PI platform is shown on the Fig. 12.

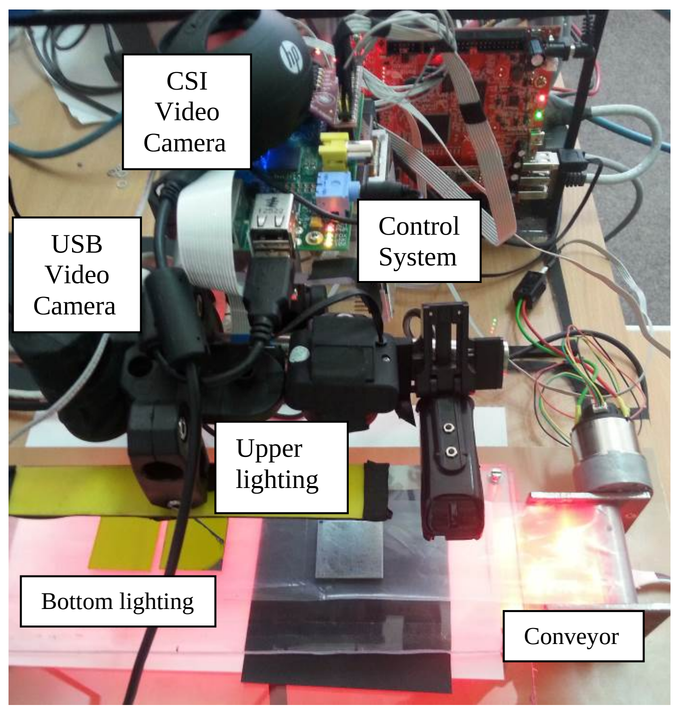


Fig. 12 Hardware of vision system with conveyor belt.

The Conveyor is created from standard DC motor with gearbox and incremental encoder to regulate velocity of transparent belt with recognized parts. Belt velocity is controlled by GPIO from Raspberry PI by external library Wiring PI thought soft PWM function. Camera can be set in axis X, Y, Z and three skew angles α, β, γ (6 DOF).

Introduced solution for vision system consists from:

1. Conveyer with transparent belt
2. Holder of deflector
3. Raster deflector sheet (Fig. 15)
4. Low light RGB system
5. Actuator with gearbox
6. Camera system

Isometric view to conveyor with RGB dynamic background lighting system is shown on the Fig. 13.

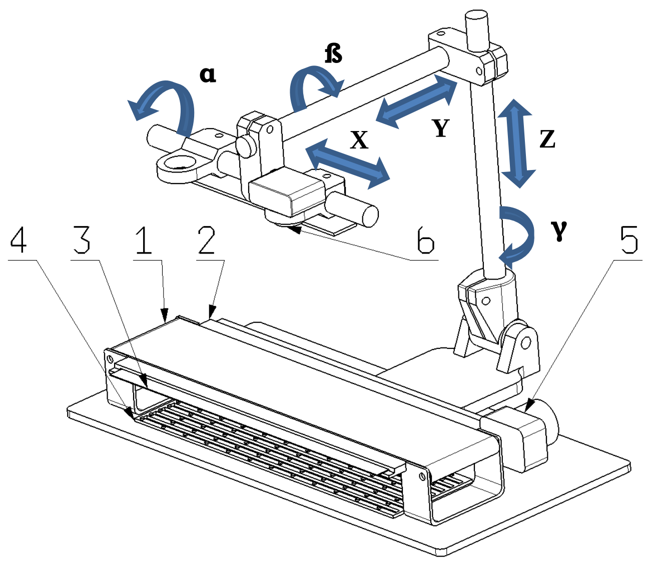


Fig. 13 Hardware of vision system with conveyor belt.

The section view of modified conveyor with bottom light system for better illustration is shown on the Fig. 14.

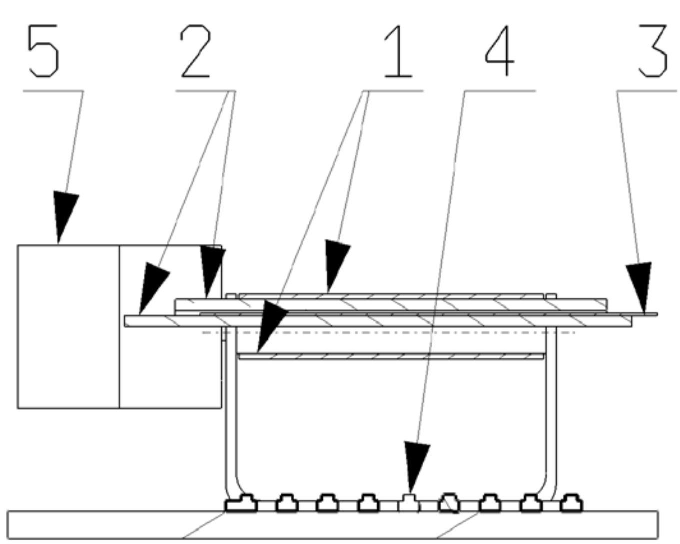


Fig. 14 Section of conveyor belt with low backlight system.

Raster deflector used for acquiring black (dark gray color near black) background color (Fig. 15a) and result background with only top white lighting is shown on the Fig. 15b.

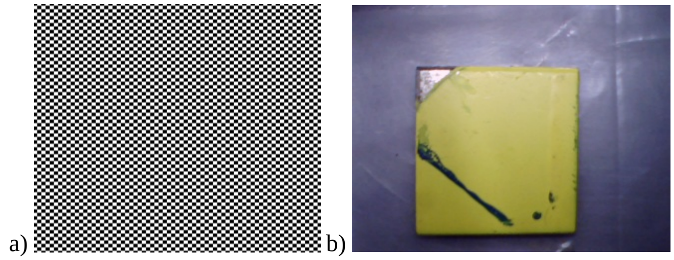


Fig. 15 Raster map for generating dark gray color.

The experiment with green color background lighting thru chessboard deflector and black background (no top lighting only environmental light) is shown on the Fig. 16.

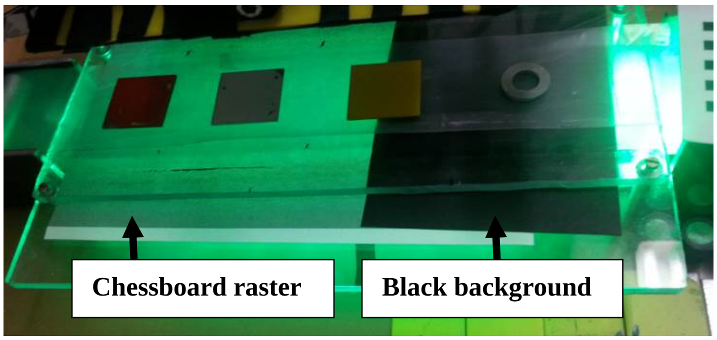


Fig. 16 RGB background light system set to green color.

Detailed scheme of hardware system based on raspberry PI embedded board is shown on the Fig. 17.

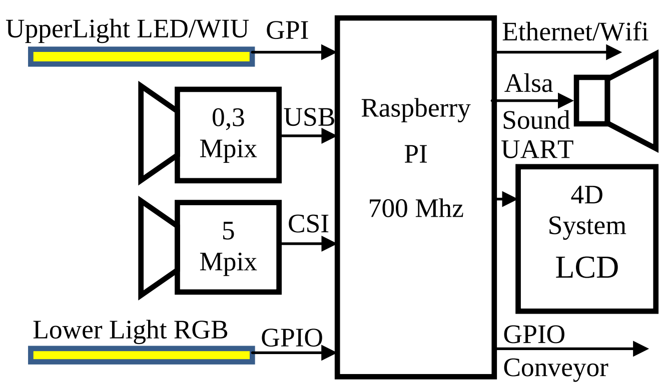


Fig. 17 Vision system based on Raspberry PI.

The presented visual system has some extended function for example: LCD for easy setup recognition process, Ethernet or Wi-Fi connection possibilities, sound alarm system and dynamic SoC overclocking (700-1000 Mhz).

SOFTWARE

Image processing is based on open source library OpenCV [18]. Main recognition loop is written in C++ language. User interface is based on HTML/JavaScript (jQuery) language with support of PHP scripting language to access low level routines in C++. Graphical user interface runs thru internal web server Apache, which is accessible by Ethernet (Wi-Fi) and TCP/IP protocol.

Framework of vision software solution with GUI is shown on the Fig. 18.

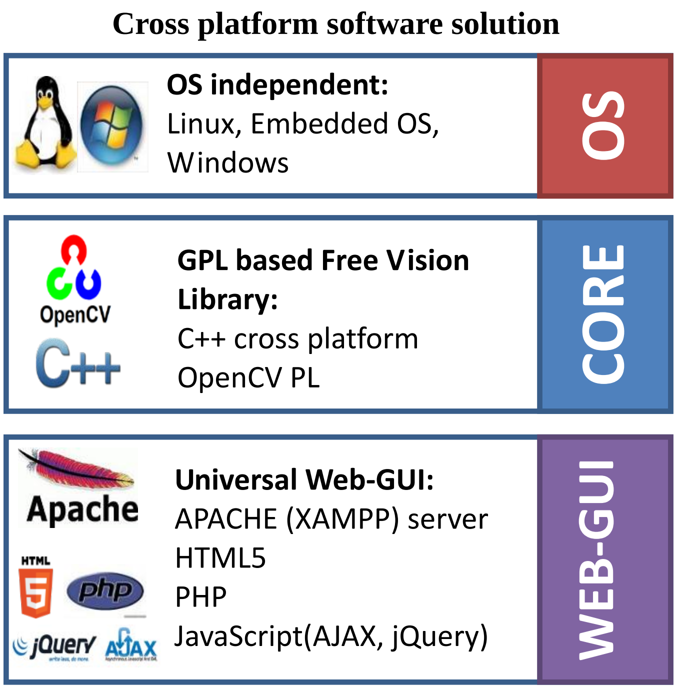


Fig. 18 Software framework of vision system.

The whole software solution was designed universally and can be executed on any OS distribution based Linux kernel or Windows. The Linux software solution was tested on Debian, Openwrt (embedded OS for Linux routers), Fedora and Raspberrian distribution. Thus the software can be easily ported to any Linux platforms.

Currently the vision software system was ported to these devices:

- Access points MIPS based: TP-Link WR703N and Asus WL520gU,
- Embedded ARM boards: Olinuxino IMX233, Olinuxio A13 and Raspberry PI,
- x86 (x64) PC: HP Server Proliant (Intel CPU), Heavy Horse Server (AMD CPU) and virtual CPU by KVM with x86 architecture.

User interface for manual selecting background RGB light color is shown on the Fig. 19a. You can select these illuminations Red, Green, Blue and its combination, separate white infrared and ultraviolet top light. Because the system is not fixed to any camera hardware we must calibrate camera to remove distortion by chessboard pattern. Calibration interface is shown on the Fig. 19b.

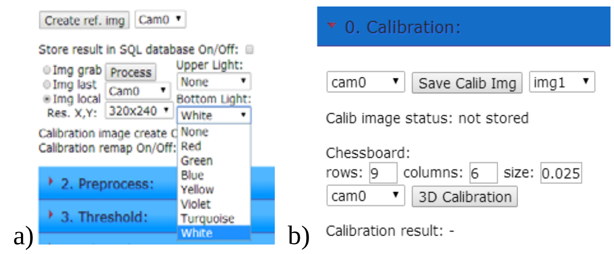


Fig. 19 Some examples of designed GUI (HTML5/jquery)

The monitoring web interface for all connected vision systems is shown on the Fig. 20. The devices architectures are grouped to three color group: PC (green), Routers (red), Embedded boards (blue).

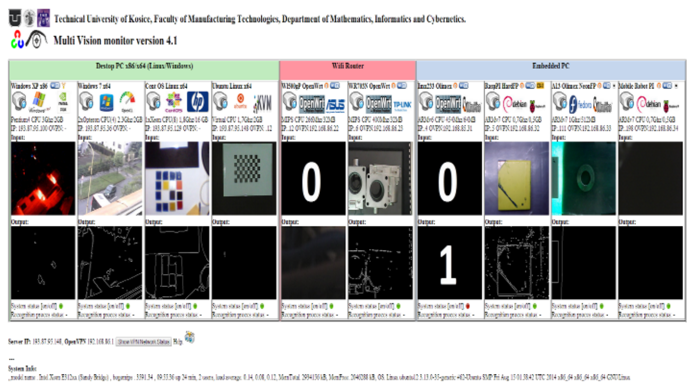


Fig. 20 Monitoring software for all vision systems on the network.

The web user interface for image processing integrated in Raspberry PI embedded board is shown on the Fig. 21.

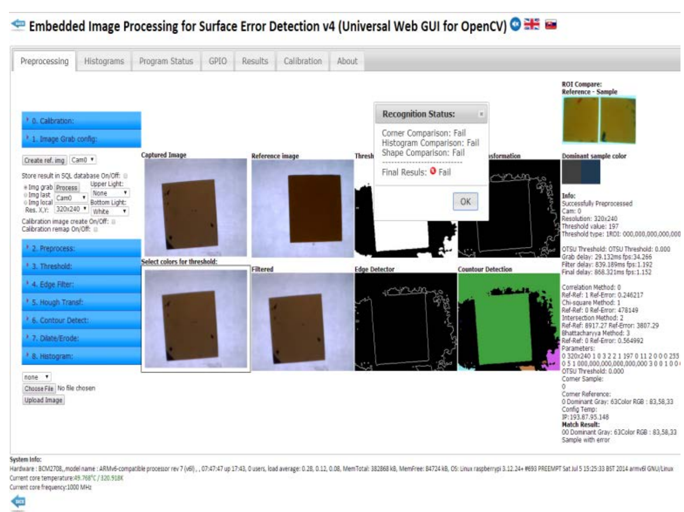


Fig. 21 Image processing GUI (Raspberry PI).

Because free IPv4 address are currently limited and public IP are frequently under flood attack, our system separates vision systems to virtual private network based on OPENVPN. For external access to any vision system in OPENVPN network we need only one public address for monitoring server.

Design of multi machine vision system network is shown on the Fig. 22.

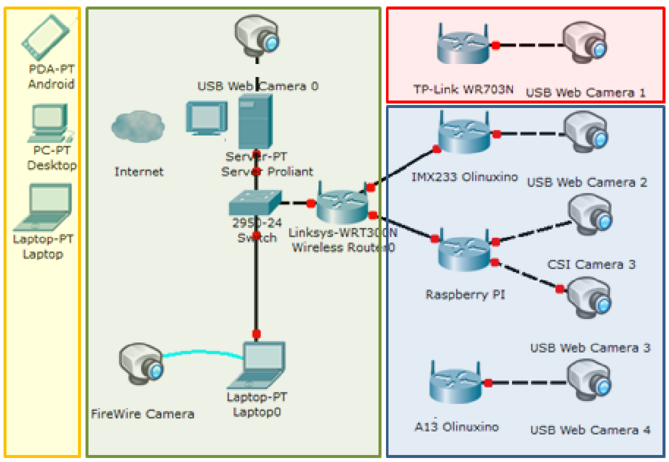


Fig. 22 Design of multi machine vision network by Packet Tracer

We can check active machine vision system on OPENVPN network.

Example of connected client monitor output is shown on the Fig. 23.

Server IP: 193.87.95.148, **OpenVPN** 192.168.86.1 Show VPN Network Status Help:

OpenVPN Info:
Updated,Fri Jul 11 11:02:04 2014

Common Name	Real Address	Bytes Received	Bytes Sent	Connected Since
client32	193.87.95.141:63843	44847707	33149102	Thu Jul 10 16:04:40 2014
client2	193.87.95.141:61447	2906576	1605925	Fri Jul 11 06:40:42 2014
client33	193.87.95.141:59771	647357	578074	Thu Jul 10 14:07:22 2014
client23	193.87.95.141:59745	1492172	765763	Thu Jul 10 14:07:11 2014
client34	193.87.95.141:59765	8874787	8591327	Thu Jul 10 14:07:17 2014

ROUTING TABLE

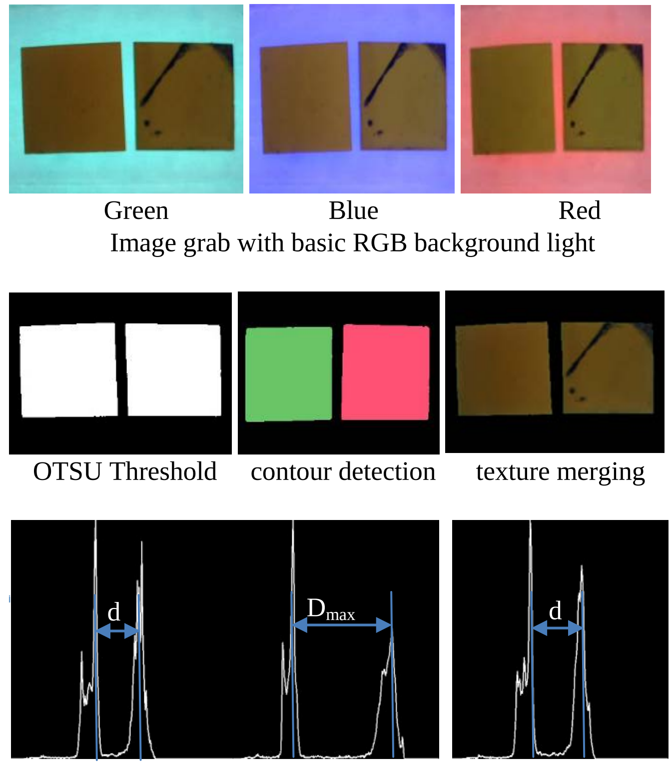
Virtual Address	Common Name	Real Address	Last Ref
192.168.86.32	client32	193.87.95.141:63843	Fri Jul 11 11:02:03 2014
192.168.86.2	client2	193.87.95.141:61447	Fri Jul 11 11:02:03 2014
192.168.86.33	client33	193.87.95.141:59771	Fri Jul 11 11:01:34 2014
192.168.86.34	client34	193.87.95.141:59765	Fri Jul 11 11:01:34 2014
192.168.86.23	client23	193.87.95.141:59745	Fri Jul 11 11:01:34 2014

Fig. 23 Web GUI for monitoring active devices and its bandwidth

VI. BOTTOM LIGHTING EXPERIMENT

The experiments were conducted with yellow color object and three basic color for background generated primarily by RGB Led.

The results of experiments are shown on the Fig. 24. The first line of the figure shows two parts (reference part and part with some surface defects) with three different background colors: green, blue, red. Second line represents results of OTSU thresholding, contour detection and texture and contour merging. The third line of the figure shows grayscale histograms for all background examples.

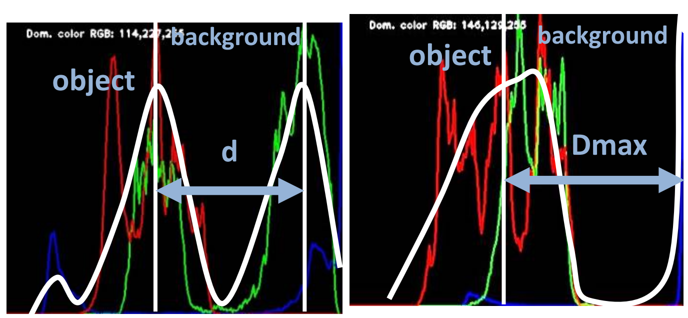


Grayscale histograms

Fig. 24 Bottom lighting experiments for precise contour detection

Grayscale histograms have set multiplicity of shades to 8bit resolution (256 bins). The reliability of algorithm can be checked in grayscale or in color histogram too, because the distance D_{max} is visible as the space between two peaks which represents object color and color of the background.

The maximum distance D_{max} in grayscale histogram was acquired by blue color background in grayscale and in 3 channel RGB histogram. The color histograms for green and blue color are shown on the Fig. 25. The peaks in color histogram are created by averaging its values.



Green background Blue background

Fig. 25 Selection of background color in 3-channel histogram

The background with red color is unusable because the detected object color contains significant part of red channel.

The researches deals with next inspection surface task about texture recognition were described in these articles [19], [20], [21], [22].

USB camera reaches frame rate about 30 fps. Algorithm was tested with resolutions 320x240 and 640x480 pixels. Higher resolutions in these experiments don't provide better result in precision of contour. A whole algorithm delay in tested hardware (Raspberry PI) was about 97 ms with resolution 320x240. Embedded platform Raspberry PI can process about 10 frames per second.

VII. SIMPLIFIED THRESHOLD ALGORITHM

This chapter describes experiment with white color background lighting and simplified algorithm only for edge detection.

In case you do not need texture on recognized object it is possible to significantly simplify whole algorithm. Top lighting will be not used, we do not need acquire dominant color of the part and we can use white color for background lighting. The algorithm delay is reduced about two times and precision is same as with full algorithm. This simplification can reduce delay of original algorithm to about 39 ms with resolution 320x240. This algorithm for precise edge detection will works with any color object, because if we don't use top lighting the front object color is changed to black.

Evolution diagram of simplified algorithm for dynamic background is shown on the Fig. 26

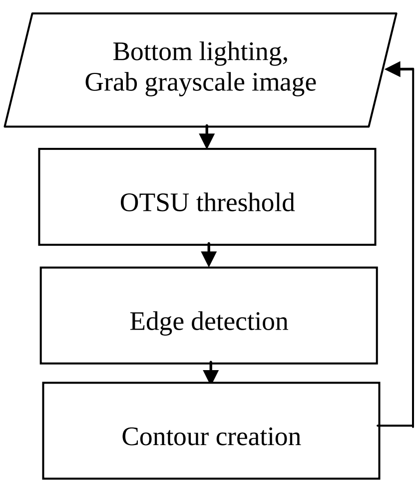


Fig. 26 Flowchart diagram for simplified algorithm.

We can reduce color after image grab in image matrix to grayscale for next image processing. Our tested sample has some surface defects for example wrong color or peeling paint. These defects create with standard thresholding methods false edges and inaccurate detected edges.

The experiments with simplified algorithm are shown on the Fig. 27.

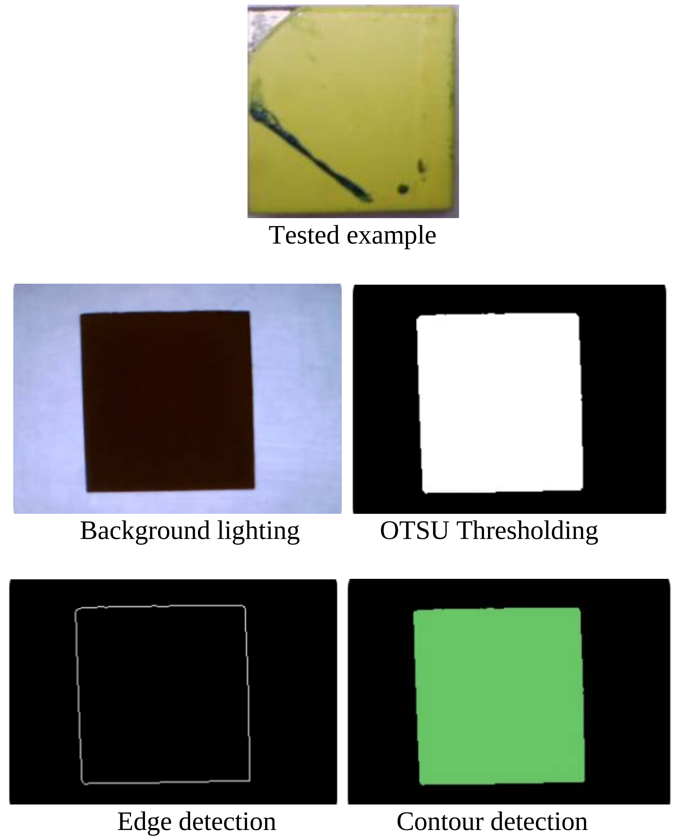


Fig. 27 Experiment with simplified algorithm.

The result of recognition process is always ideal bimodal histogram because the color of object surface will be very near black color (although not in isolation ambient lighting).

The resulting grayscale histogram is shown on the Fig. 28.

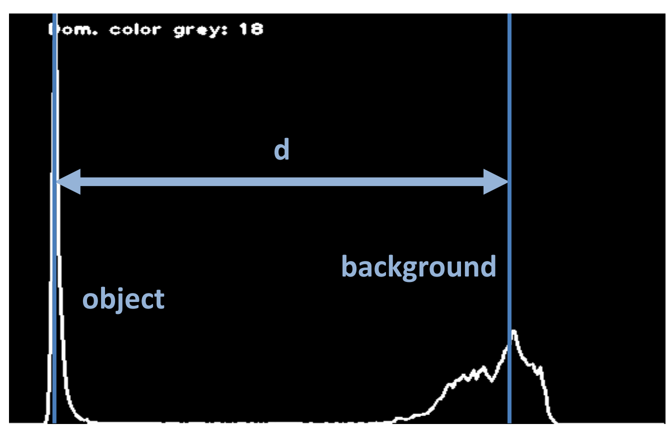


Fig. 28 Bimodal grayscale histogram for simplified algorithm.

VIII. COMPARISON WITH OTHER THRESHOLDING METHODS

Comparison of basic thresholding methods with introduced bottom lighting algorithm is shown on the Tab. 1.

Tab.1. Comparison of thresholding methods

Thresholding method	Grab + Delay (320x240)	Ambient light	Noise
Dynamic background thresholding	33 + 49 ms	independent	Low
Standard OTSU thresholding	33 + 4,3 ms	Depend on homogeneity of light	Low
Adaptive thresholding	23 ms (Mean) 21 (Gauss)	independent	High

Introduced algorithm is resistant to ambient light and has high reliability in precise edge detection. The main disadvantage is computational difficulty and two level image grab.

IX. Conclusion

The introduced dynamic thresholding algorithm based on bottom lighting is primarily designed for flat parts with smaller and middle height. The main disadvantage is usability only in industrial area for automatized lines where are inspected parts transported in conveyor belt. The algorithm delay is about two times slower than standard methods but provides high reliability and precise edges recognition. We can improve the delay of the whole algorithm porting solution to another embedded SoC platform for example Odroid U3 with quad core CPU. The next experiments of precise contour detection with this methods will be conducted with shiny material (metal parts without matte surface), because standard thresholding and contour detection usually fails in this case.

The introduced vision system can be used in other areas of image processing, for example for object detection in unknown environments [23], [24].

Acknowledgment

The research project is supported by the Project of the Structural Funds of the EU, Operational Program Research and Development, Measure 2.2 Transfer of knowledge and technology from research and development into practice: Title of the project: Research and development of the intelligent non-conventional actuators based on artificial muscles ITMS code: 26220220103.

Európska únia

Operačný program VÝSKUM a VÝVOJ

We are support research activities in Slovakia / Project is cofounded from sources of ES.

References

[1] Pal, N.R., Pal, S.K., "A review on image segmentation techniques", in *Pattern Recognition*, volume 26, issue 9, 1993, pp. 1277-1294.

[2] A. M. Atto, D. Pastor, G. Mercier, "Wavelet shrinkage: unification of basic thresholding functions and thresholds", in *Signal, Image and Video Processing*, volume 5, issue 1, March 2011, pp. 11-28.

[3] A. Singh, "Digital change detection techniques using remotely-sensed data", in *International Journal of Remote Sensing*, volume 10, issue 6, 1989, pp. 989-1003.

[4] L. Gopinath, M. Anitha, N. Nithya, N.R. Raajan, "Comparitive analysis of OTSU's method and active contour model", in *International Journal of Applied Engineering Research*, volume 9, issue 2, 2014, pp. 277-282.

[5] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 1999, pp. 246-252.

[6] D. Gaceb, F. Lebourgeois, J. Duong, "Adaptative smart-binarization method: For images of business documents", in *Proceedings of the International Conference on Document Analysis and Recognition*, ICDAR, 2013, article number 6628596, pp. 118-122.

[7] H. Jiang, L. Zeng, B. Bi, "A comprehensive method of contour extraction for industrial computed tomography images", in Optics and Lasers in Engineering, Volume 51, Issue 3, March 2013, pp. 286-293.

[8] R. Medina-Carnicer, , R. Muñoz-Salinas, A. Carmona-Poyato, F.J. Madrid-Cuevas, "A novel histogram transformation to improve the performance of thresholding methods in edge detection", in *Pattern Recognition Letters*, Volume 32, Issue 5, 2011, pp. 676-693.

[9] J. Piteľ, T. Saloky, "Intelligent robots and computer vision", in *International Carpathian Control Conference ICCC'2004 – Volume II.*, Zakopane, May 25-28, 2004. Krakow: AGH - UST, 2004. pp. 243-246.

[10] K.-Z. Tang, B.-X Liu, H.-Y Xu, X. Xiao, Z.-Y. Li, "A minimum cross entropy threshold selection method based on genetic algorithm", in *Kongzhi yu Juece/Control and Decision*, volume 28, issue 12, 2013, pp. 1805-1810.

[11] Y. Yu, J. Cao, Y. Tang, J. Gan, "Fast wavelet thresholding algorithms for face image inpainting", in *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, article number 6682266, pp. 1445-1449.

[12] Andrei Tigora, "A Watershed-Based Thresholding Approach for Image Binarization", NAUN, International Journal of Circuits, Systems and Signal Processing, Volume 8, 2014, pp. 246-252.

[13] M. Tuba, I. Brajevic, "Modified Seeker Optimization Algorithm for Image Segmentation by Multilevel Thresholding", NAUN, International Journal of Mathematical Models and Methods in Applied Sciences, Volume 7, 2013, pp. 370-378.

[14] S. Nashat, A. Abdullah, M. Z. Abdullah, "Machine vision for crack inspection of biscuits featuring pyramid detection scheme", in *Journal of Food Engineering*, volume 120, Issue 1, 2014, pp. 233-247.

[15] "Fuzzy entropy-based MR brain image segmentation using modified particle swarm optimization" in *International Journal of Imaging Systems and Technology*, volume 23, issue 4, 2013, pp. 281-288.

[16] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", in *Computer Vision, Graphics & Image Processing*, volume 29, issue 3, 1985, pp. 273-285.

[17] Szabolcs Sergyan, "Parameter Optimization of Dominant Color Histogram Descriptor in Content-Based Image Retrieval Systems", NAUN, International Journal of Circuits, Systems and Signal Processing, Volume 8, 2014, pp. 286-291.

[18] Open Source Computer Vision (OpenCv) Library, [Online]. Available: http://opencv.willowgarage.com/wiki.

[19] T. Leung, J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons", in *International Journal of Computer Vision 43*, 2001, Kluwer Academic Publishers. Manufactured in The Netherlands, pp. 29–44.

[20] Q. Zaidi, "Visual inferences of material changes: color as clue and distraction", in *Wiley Interdiscip Rev Cogn Sci*, 2011, pp. 686–700.

[21] K. Židek, E. Rigasová, "Diagnostics of Products by Vision System, Applied Mechanics and Materials", in *Trans Tech Publications*, Switzerland, Vol. 308, 2013, pp. 33-38.

[22] Lead-based paint: Testing methods. Missoula, MT, United States Department of Agriculture Forest Service, Technology & Development Program, 1996 (http://www.fs.fed.us/eng/pubs/htmlpubs/htm96712353/, accessed 20 December 2010).

[23] F. Duchoň, M. Kľúčik, L. Jurišica, „Reactive Navigation of Mobile Robot with Visual System" in *Acta Mechanica Slovaca*, 13, č. 2-A, 2009, pp. 47-52.

[24] K. Židek, J. Šeminský, M. Dovica, A. Hošovský, "Wireless mobile device with integrated image processing", in *Automation, control and processes: Proceedings of the 8th technical conference ARaP*, 2012, Prague: Dimart, pp. 37-42.