ISSN 1392 – 1215 ELECTRONICS AND ELECTRICAL ENGINEERING 2010. No. 4(100)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 125 AUTOMATION, ROBOTICS
AUTOMATIZAVIMAS, ROBOTECHNIKA

"iHouse" for Advanced Environment Control

A. Kairys
Department of Control Technology, Kaunas University of Technology, Lithuania
Studentų str. 48, Kaunas, phone: +37060882642 e-mail: arturas.kairys@ktu.lt

V. Raudonis, R. Simutis
Department of Process Control, Kaunas University of Technology, Lithuania
Studentų str. 50, Kaunas, phone: 300296, e-mail: rimvydas.simutis@ktu.lt

Introduction

On the last decade one of the most popular developments in the control systems area was implementation of smart environment and, as a part of it, smart home paradigm. Such system improves living quality mostly not for common consumer, but for older peoples and the ones with impairments. Today there are lots of such applications released, they are used not only to improve living quality, but as well to improve healthcare. Many research groups are still focused on the development of novel products for smart home applications.

Though, there is a big supply of commercial products with very good technical characteristics, popularity of such system is limited by its' price. Price is determined by expensive hardware, such as high quality video camera. So main problem supplying consumers with such systems is price; such products are hardly affordable for consumer itself and authorities are unable to provide it for everyone in need. Therefore, need of creating cheaper and not less effective systems is extensive.

Simple method and technical implementation allows proposal of inexpensive assistive domotics- "iHouse", system. Costs are reduced by using of the shelf solutions, e.g. ordinary web-cams, customary garage gate controls set. A low profile technical solution enables proposed system to be used with common personal computer that has USB connection.

Intuitiveness is an advantage that helps to accurately select device, that user wants to control. System, that randomly selects controlled devices, is not as effective as it could be, when intuitiveness is applied. Even control tasks can be faster accomplished, as not all devices in viewed area have to be checked for necessity of control. This feature gives an advantage against similar systems, where random device detection is used, or selectivity is achieved by using complicated methods.

"iHouse" system is proposed in this work, it recognizes 6 different markers, whom any device can be assigned. Markers are color coded with main colors (Red, Green and Blue) in RGB color system. Marker detection algorithm finds certain amount of spots to be a marker-candidate. If mouse cursor is in the same quarter like the candidate's center, spot is proceeded to further check. When marker is detected, pop-up window appears, using the eye tracking systems consumer can set a certain output that can be used to control external device. No calibration is required as all the necessary data is set on algorithm.

Related works

Interest within smart environment as well as artificial intelligence has been growing for more than decade. Lots of smart environment realizations are based on image processing. There are various ways to extract useful data from image by processing it but only some of them will be introduced.

Data processing techniques can be divided into two bulk groups: simple ones and the complex ones. For simple recognition no transformations or modern methods are necessary. The very simplest way of detecting lighter or darker spots in gray scale image is logical indexing; binary image is extracted during logical pixel value comparison with certain threshold value. In paper [1] logical indexing is used to extract road markers in order to autonomously control vehicle. Additional image pre-processing is necessary in order to accomplish more difficult tasks. Inverse perspective mapping [2] is commonly used for image pixel position recalculation in order to get top-view picture. Objects in such image can be recognized more precisely. Chain-code algorithms [3] are used for logical object shape description, when the shape is more complex than basic geometric figures, for example arrow on the road. Logical indexing can be used to accomplish even difficult tasks though additional methods are necessary for more complex object detection and recognition. Drawback of this method is that pixel quantity is the main factor that determines speed of image processing. Often this method is used for noise elimination or as a method to extract main data [4, 5, 6].

Straight lines can be detected from edge-extracted image using linear Hough transform (HT) [7]; this kind of image transform is useful when object is constructed from straight lines. Generalized Hough transform (GHT) [8] is more suitable to recognize object containing various curves that's base is ellipse or other basic geometrical shapes. Article [9] presents handwritten Chinese character recognition algorithm, though there are more complex problems to be solved prior to recognition. One of the main problems is slope of text, estimation and correction of this issue is presented in paper [10], and skew correction is proposed in article [11]. Even though HT or GHT can be used to solve such complex tasks like text character recognition, more problems emerge as image data has to be pre-processed and objects that are extracted has to be handled. Mentioned transform is useful when object has certain shape and is in contrast with background. Method has limited usability in real time image processing, because transformation cause time delays.

Most common method used for skin detection and face recognition is cluster analysis [12]. Cluster analysis-method that is based on statistical computations, all data is divided into clusters or segments and only the ones with a certain features are significant. Often cluster analysis is used as a part of more complex image processing algorithm. Even though this method is useful when it is difficult to extract data, it has a drawback as it is necessary to have some prior knowledge about object that has to be detected.

Fuzzy logic is a membership function based method that enables to use human-like perception in image processing [13]. Usage of fuzzy logic in some way is similar to logical indexing, difference is between data sets, logical indexing uses grayscale image and fuzzy uses different set of pixel data. Precisely the data set composition is based on human contemplation, in this way more informative data is obtained. Though fuzzy logic can be used to solve tasks like image filtering [14], it requires some prior knowledge that is used to create membership functions and according to it fuzzy data set.

Fuzzy set and functions can be compiled automatically according to training set. To accomplish this task artificial neural network (ANN) [15] is used. ANN is based on processes that occur in live organisms brains, in this way it is learning based method. Operation of ANN is based on rules that are assigned to network during training phase. This popular method is widely used for object detection in images, still it requires training phase, which is the main drawback in real time applications.

All above mentioned image processing methods are widely used for smart environment related applications. Mentioned drawbacks help picking the most suitable technique not only for marker detection. It helps wisely design a set of markers that would be easily recognized.

"iHouse" algorithm

Easily achieved round object detection in binary image leads to selection of round markers (*a)shape in* Fig. 2). Marker includes three areas: central circle, color segments and outside area. Points from all these areas are used in recognition process. Central circle and outside area are used for premature recognition; by default those two areas are white. Information about controlled device can be coded using color segments' and different color combinations. Combinations are generated using either two or three main RGB colors. Thick black centers' border enhances candidate detection.

Marker-candidate detection overall is round object detection. As mentioned before logical indexing is suitable for such task. Before candidates can be detected logical image has to be extracted, algorithm (Algorithm 1.) explains how it is accomplished.

Algorithm 1. Logical image extraction

step 1. RGB to grayscale conversion (image height-h, width-w):

$$RGB = \begin{bmatrix} i_{1,1} & \cdots & i_{1,w} \\ \vdots & \ddots & \vdots \\ i_{h,1} & \cdots & i_{h,w} \end{bmatrix} \Rightarrow gray = \begin{bmatrix} m_{1,1} & \cdots & m_{1,w} \\ \vdots & \ddots & \vdots \\ m_{h,1} & \cdots & m_{h,w} \end{bmatrix},$$

$$i_{1,1} = \begin{bmatrix} r_{1,1} \\ g_{1,1} \\ b_{1,1} \end{bmatrix}, \quad m_{1,1} = 0{,}2989 \cdot r_{1,1} + 0{,}587 \cdot g_{1,1} + 0{,}114 \cdot b_{1,1};$$

step 2. Binarization:

$$gray = \begin{bmatrix} m_{1,1} & \cdots & m_{1,w} \\ \vdots & \ddots & \vdots \\ m_{h,1} & \cdots & m_{h,w} \end{bmatrix} \Rightarrow bw = \begin{bmatrix} b_{1,1} & \cdots & b_{1,w} \\ \vdots & \ddots & \vdots \\ b_{h,1} & \cdots & b_{h,w} \end{bmatrix},$$

$$if \quad m_{1,1} > thresh \quad then \quad b_{1,1} = 1; \quad else \quad b_{1,1} = 0; \quad end,$$

$$thresh = 0,1,\ldots,255;$$

In the first algorithm RGB is first converted to grayscale image, such conversion is proceeded using weighted sum of R, G and B color components. Grayscale image contains pixels that are defined using lightness numeric value (0÷255). According to this value binary image can be created, each pixels' lightness is compared to pre-defined threshold value. Each pixel gets new value, according to the comparison result it can be 0 or 1.

Algorithm 2. Candidate detection

step 1. Binary image labeling:

$$regions = (a_1, \ldots, a_i) \Leftarrow bw, \quad i \in Z;$$

step 2. Labeled white regions' property extraction (perimeter(ρ), diameter(Δ), center coordinates(ς)):

$$\left. \begin{matrix} \varsigma_i \\ \Delta_i \\ \rho_i \end{matrix} \right\} \Leftarrow a_1, \quad \varsigma_i = (xc_i, yc_i);$$

step 3. Diameter size of each region is checked:

$$if \quad \alpha < \Delta_i < \beta \quad then \quad k_n = i; \quad end,$$
$$ind1 = (k_1, k_2, \ldots, k_n), \quad \alpha, \beta, n \in Z;$$

step 4. White region roundness evaluation:

$$ind1(n) = k_n, \quad \xi_n = \frac{\Delta(ind1(n)) \cdot \pi}{\rho(ind1(n))},$$
$$if \quad \gamma < \xi_n < \delta \quad then \quad l_m = ind1(n); \quad end,$$
$$m \le n, \quad ind2 = (l_1, l_2, \ldots, l_m), \quad m \in Z, \quad \gamma, \delta \in (0, \ldots 1];$$

step 5. Adaptive threshold formation and evaluation:

$$if \quad m > \varepsilon \quad then \quad thresh = thresh - st; \quad end,$$
$$if \quad m < \theta \quad then \quad thresh = thresh + st; \quad end, \quad \varepsilon, \theta \in Z;$$

Second algorithm reveals steps of candidate detection. Before processing binary image, it has to be labeled; it means all white areas get a numeric label that helps to calculate their numeric characteristics (center coordinates, equivalent diameter and perimeter) one by one. Center coordinates, equivalent diameter and perimeter are calculated so that all regions can be tested simultaneously. Diameter is checked in means to reject noise and small areas as well as those sectors that are too big. All areas that passed diameter evaluation is then verified to be round objects, simple proportion in *step 3* helps to determine which objects are round ones. Afterwards the number of round objects is used to determine whether used threshold was of the right value.

Marker-candidates are checked using nine pixels (*a)shape in* Fig. 2.) color components' numeric values. Marker recognition process is not only used for marker recognition, need of recognition is checked as well. The need assessment makes whole system intuitive as user can indicate if recognition has to be proceeded. Need assessment, pixel position determination and candidate recognition process is explained in next algorithm (Algorithm 3.).

Algorithm 3. Marker recognition

step 1. Figure windows' (see Fig. 1) position on screen estimation:

$$\begin{bmatrix} x_f \\ y_f \\ w_f \\ h_f \end{bmatrix} \Leftarrow figure\_window;$$

step 2. Computer mouse position on screen estimation using Java methods:

$$mouse = MouseInfoGetPointerInfo(),$$
$$coordinate = GetLocation(mouse),$$
$$x_m = GetX(coordinate), \quad y_m = GetY(coordinate);$$

step 3. Mouse cursor over figure window verification:

$$if \quad x_f < x_m \,\&\&\, x_m < (x_f + w_f) \quad then \quad if \quad y_f < (w_s - y_m)$$
$$\&\&(w_s - y_m) < (y_f - h_f) \quad then \quad *proceed\_further*;$$
$$else \quad *terminate*; \quad end; \quad end, \quad w_s (see \quad Figure \quad 1);$$

step 4. Nine pixel position determination:

$$A = C \Leftarrow (xc, yc), \quad (B, C, D, E) \Leftarrow \Gamma(\varsigma, \Delta, \varphi),$$
$$(F, G, H, I) \Leftarrow \Gamma(\varsigma, \Delta, \phi), \quad (B, C, D, E) - corol \quad segments',$$
$$(F, G, H, I) - outside \quad areas' \quad pixels, \quad \varphi, \phi \in R;$$

step 5. Pixel positions are examined not to be out of bounds:

$$0 \le x \le w_f, \quad 0 \le y \le h_f, \quad h_f (see \quad Figure \quad 1);$$

step 6. Mouse cursor over figure window position comparison with candidates' center position in figure window (example of first quarter (Fig. 2)):

$$x_f \le x_m \le (x_f + 0.5 \cdot w_f) \,\&\&\, y_f \le (h_s - y_m) \le (y_f + h_f \cdot 0.5$$
$$\&\&0 \le xc \le (0.5 * w) \,\&\&\, 0 \le yc \le 0.5 * h;$$

step 7. Pixel color value comparison for white areas determination in RGB image:

$$\chi < r \,\&\&\, \chi < g \,\&\&\, \chi < b, \quad \chi \in (0, 1, \ldots, 255);$$

step 8. Color segment color determination (red color):

$$g < r \,\&\&\, b < r \,\&\&\, \chi < r;$$

Third algorithm reveals concept of intuitiveness that is assured using tracking of objects and their position comparing. Mouse cursor is used to determine if user wants to use system and which device he wants to control (in case there are more than one device visible). Intuitiveness is achieved by tracking figure windows' (Fig. 2) position, getting mouse cursors' coordinates and checking if mouse is over that window.

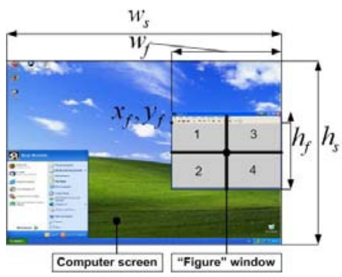

Fig. 1. Figure window on computer screen

If mouse cursor is not over figure window no marker recognition is proceeded, on the opposite occurrence, algorithm checks are there any candidates on the same quarter in which mouse cursor is situated. On candidates existance, depending on their label, specific pixels are examined not to be out of bounds. Supposing candidate is not out of bounds, candidate evaluation is proceeded. Firstly *A*, *F*, *G*, *H*, *I* (*A*- marker center, *F*, *G*, *H*, *I*-pixels from outside are) pixels are tested to be relative white. Color segment evaluation is performed in a bit different way, color component are examined to be less than the rest two and to have bigger numeric value than pre-defined value (χ-threshold value). When all three checks are positive the color segment has a color relative to color segment, that was examined.

Using all three algorithms marker detection-recognition system is created, it is used for image processing in order to detect and recognize markers.

Complicated parts of this algorithm (Fig. 2.) were fully explained on above sections so only narrow explanation of it would be produced.

Presented algorithm (Fig. 2.) only previews how a single frame is processed, there are no elements that preview algorithms' continuous operation. At the moment of execution, computer screen resolution is determined in order to create "Figure" window in proper position. User-defined default values are collected. Before turning camera on, the capturing parameters are set. Captured frame is processed by algorithm of binary image extraction and candidate detection. Note that those two algorithms are as well used for threshold adaptation, which is performed when "while" loop is used. Cycle is terminated when threshold goes out of bounds (new frame then is processed as not relevant amount of candidates was found, default threshold value is used for new frame processing). Termination of cycle also occurs when the amount of candidates is in certain bounds (current threshold value is stored in memory for next frames' processing), which means that candidate detection was successful. When positive recognition occurs, popup window (*b)shape in* Fig. 2.) appears. After all candidates have been checked new frame is captured.

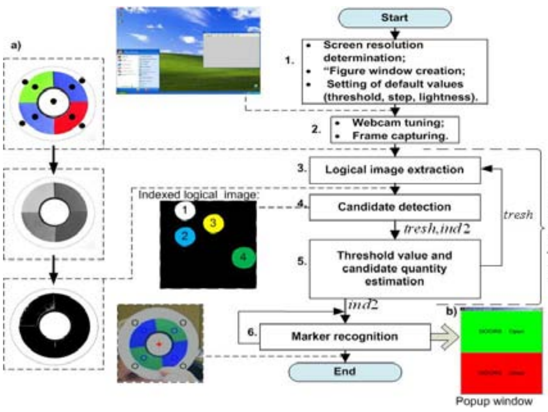

Fig. 2. Image processing algorithm

Implementation of intuitive environment

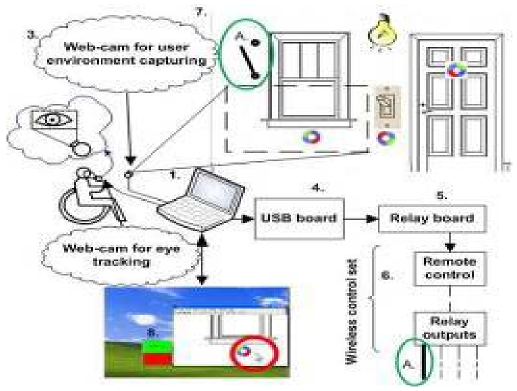

Fig. 3. System configuration

The functional scheme of intuitive environment is shown in Fig. 3. System consists of:

1. Computer. All the 2D data computations for marker detection and recognition as well as eye tracking systems' calculations are proceeded in computer.
2. Near IR web-cam. Video capturing device, that works in near infrared light, is used in eye tracker.
3. Simple web-cam. Captures 2D data for marker detection and recognition.
4. USB board. Data acquisition board is used to write output signals through USB interface.
5. Relay board. Relay are used to automatically control (commutate) remote control unit from wireless communication set.
6. Wireless communication set. It is used to wirelessly control devices as receiver of this set has relay outputs.
7. Executive facility. It's either button for TV, door or window control board, light switch and etc.
8. Popup windows. Enables user to make decision by pressing one of two buttons.

Opportunity to control all devices wirelessly makes this system more flexible. Untraditional decisions improve efficiency and makes design simpler. Although low price and ease of design is obtained, system won't work properly if detection and recognition doesn't succeed.

Testing of algorithm

Experiments took place in order to investigate potentiality of detection/recognition algorithm. Speed and accuracy tests were performed. Results enclose some characteristics of proposed image processing algorithm.

Detection rate depends on pixel quantity in image, therefore video used in experiments has dimensions of 320X240 pixels. Video containing 300 frames, was created by using 30 fps rate for capturing, marker was fully visible in all frames, distance between marker and camera was alternating. Webcams' digital zoom, was used to improve results. Though this feature reduces frame quality it is useful because marker becomes easier to detect, as it appears bigger in image.

Variables, that were exploited in order to examine operation, were step (*st*) and default threshold (*thresh*) - the primary threshold value, and lightness value (χ). Frame processing and marker recognition depends on above mentioned variables. Record was proceeded through algorithm using more than ten different threshold, step and lightness values. A bulk amount of data was accumulated, though some of it had to be rejected because of scattering. Three data sets were separated: processing speed, accuracy dependence on threshold and step value, and accuracy dependence on lightness value, used for marker recognition.

Processing rate is the main characteristic for all real time systems. Linear characteristics (Fig. 4.) present experimental data. Data shows that relative concurrences, in terms of slower frame processing, occurs using default threshold equal to 150 of grayness (Gy). It can be easily explained with the fact that positive marker recognition calls additional actions, e.g. drawing visual elements on image, which slows down whole processing. Differences between characteristics "1", "5" and "9" with value 150 Gy determine that frame processing speed has obvious relation with step value. The bigger step value is, the faster threshold is changed and, as well, frame processed. The rest four characteristics, that has no concurrency with first three, only shows that positive recognition can be achieved with thresholds from a certain range of values which can't be obtained using default threshold (150 Gy) and step values bigger than 9. Other spikes just prove that values from that range can be reached faster by using other thresholds and bigger step values.

Precision is characteristic that determines effectiveness of recognition. Dependence between recognition precision, threshold value and step is represented in Fig. 5. Precision in this experiment was represented like percentage positive recognition. Results claim that high recognition can be achieved. Biggest concurrency is obtained when default threshold is 150 Gy. Such occurrence can be explained with possibility, that value of 150 Gy is appropriate for positive recognition and no threshold adaptation is necessary. Low recognition level and scattered data, when default threshold was bigger than 150 Gy, can be explained using principles of operation in candidate detection algorithm during the adaptation of threshold. When adaptation stops, because proper number of candidates is achieved and none of them is recognized as marker, marker in that frame is missed.

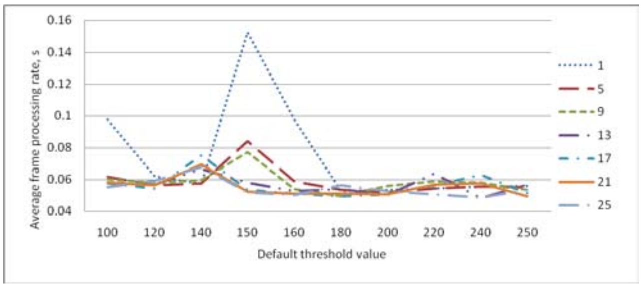

Fig. 4. Dependence between processing rate and default threshold over various step values

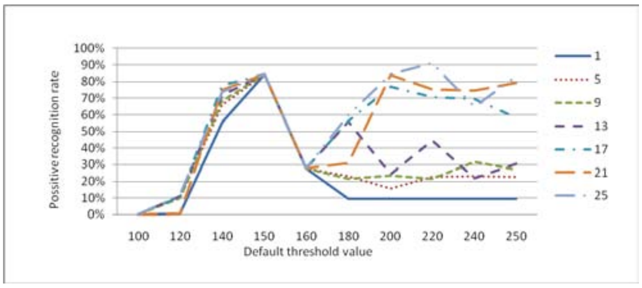

Fig. 5. Dependence between recognition and default threshold over various step values

Recognition is affected by magnitude of lightness value. Experiment was performed by changing lightness value and processing whole record each time using default threshold equal to 150 Gy and 10 as a step value. Test results in Fig. 6. can be used to determine optimal lightness value.

It is obvious that reference of lightness value influences recognition. It makes a limitation for area to be recognized as a relatively white and becomes a limit to the pixels' color component value during color segments decoding. Results claim that the best recognition percentage can be reached using lightness values as big as 100 Gy. For bigger values, percentage decrease. Such reliance is explained by noting, that lightness value is a bottom limitation to pixel color components. The lover is this value, the bigger chance that misrecognition will occur, therefore this value should not be smaller than 80. Certainly due to lightning conditions, limitation can be changed.

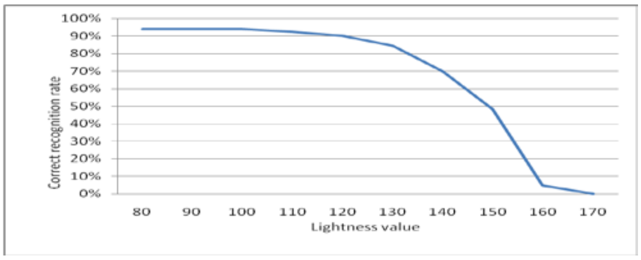

Fig. 6. Dependence between recognition and lightness value

Conclusion and future work

Though experiments were performed using record in which size of marker in frames was alternating, results were appropriate, recognition was fast and accurate. Presented algorithm can recognize 6 markers, thats' design is simple and more markers in the same way can be made. Experiments proved that performance of algorithm can be improved by changing three main numeric values: step, default threshold and lightness numeric representation.

Tests show that more than 10 frames can be fully processed (marker can be recognized and additional action performed) over second, when default threshold value is 150 Gy and step value is nine or less. Processing rate decreases, as more threshold adaptation operations are performed, therefore step value mostly influences speed, though it is not directly visible in experimental results. Higher than 80 % recognition rate was achieved when step value was 10, default threshold- 150 Gy and lightness-90 Gy. Mentioned values are not constant as whole image processing is tightly dependant on marker illumination.

Even though experiments claim "iHouse" system suitability for real time applications, more tests should be taken. Presented system is based on image processing so the main aim is to supply algorithm with high quality frames. Lightness is the main problem on most of this kind application and future work should be related to testing the utility of using additional light. Future work would be, as well, related with creating methods for device identification, when no marker is necessary.

Acknowledgment

The authors appreciate the financial support of the Lithuanian state science and studies foundation.

References

1. **Jing P., Chaojian S.** Recognizing road environment for vision system on autonomous vehicles// Proceedings ITST '07. – 7th International Conference on ITS 6–8 June, 2007. – P. 1–6.

2. **Muad A. M., Hussain A., Samad S. A., Mustaffa M. M., Majlis B. Y.** Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system // Proceedings TENCON’2004. – P. 207–210.
3. **Lele Z., Zahir S.** A new efficient context– based relative–directional chain coding // Signal processing and information technology, 2006. IEEE International Symposium, 2006. – P. 787–790.
4. **Fuksis R., Greitans M., Nikisins O., Pudzs M.** Infrared Imaging System for Analysis of Blood Vessel Structure // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 1(97). – P. 45–49.
5. **Mednieks I., A. Skageris A.** Real Time Image Processing for Object Detection // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 4(92). – P. 33–37.
6. **Laptik R., Navakauskas D.,** Application of Ant Colony Optimization for Image Segmentation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 8(80). – P. 13–19.
7. **Vuillemin J. E.** Fast Linear Hough transform // Proceedings of Application Specific Array Processors, 1994. – P. 1–9.
8. **Pina-Fu Fung, Wing-Sze Lee, King I.** Randomized Generalized Hough Transform for 2–D grayscale object detection // Proceedings of Pattern Recognition. – The 13th International Conference, 1996. – P. 511 – 515.
9. **Ming-Jing Li; Ru-Wei Dai,** A Personal Handwritten Chinese Character Recognition Algorithm Based on the Generalized Hough Transform // Proceedings of the Third International Conference Document Analysis and recognition. – P. 828–831.
10. **Farooq F.,** Govindaraju V., Perrone, M. Pre–processing methods for handwritten Arabic documents // Proceedings of Document Analysis and Recognition, 2005. – P. 267–271.
11. **Xiaoyan Zhu, Xiaoxin Yin.** A new textual/non–textual classifier for document skew correction // Proceedings of Pattern Recognition, 2002. – P. 480 – 482.
12. **Paukštaitis V., Dosinas A.** Multilayer Transformation of Different Resolution for Colour Image Analysis and Segmentation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 8(72). – P. 49–55.
13. **Mario I., Chacon M., Alma D., Corral S.** Image complexity measure: a human criterion free approach // Fuzzy Information Processing Society, 2005. NAFIPS 2005. – P. 241–246.
14. **Chang–Shing Lee, Shu–Mei Guo, Chin–Yuan Hsu.** Genetic–Based Fuzzy image filter and its application to image processing // IEEE Transactions, 2005. – Vol. 35, no. 4. – P. 694–711.
15. **Paukštaitis V., Dosinas A.** Pulsed Neural Networks for Image Processing // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 7(95). – P. 15–21.

Received 2010 03 10

A. Kairys, V. Raudonis, R. Simutis. “iHouse” for Advanced Environment Control //Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 4(100). – P. 37–42.

In this paper intuitive environment is proposed, image processing software was connected with eye tracking system in order to wirelessly control electric devices. The system use web-cam mounted to wheelchair. Accordingly a set of special markers was designed, each marker means a different device such as doors, window, light and etc. Markers are recognized from video frames that are taken from surroundings. Marker position in a frame is compared with mouse cursor position for intuitive control. Accuracy, speed and reliability experiments took place. System is able to process video frame in less than 0.1 s, 84% accuracy is achieved. Experiments claim that intuitive system for smart home is relevant to be used for real time applications. Ill. 6, bibl. 15 (in English; abstracts in English, Russian and Lithuanian).

А. Кайрис, В. Раудонис, Р. Симутис. "iHouse" для расширенного контроля окружающей среды // Электроника и электротехника. - Каунас: Технология, 2010. – № 4(100). – С. 37–42.

Предлагается программная система „iHouse“ для обработки изображений полученных глазовой системой отслеживания. Созданное передвижное устройство позволяет управлять двери, окна, свет и т.д. Видеокадры окружающей среды обеспечивает точность 84%, когда скорость обработки программной информации составляет менее 0,1с. Приведены обширные экспериментальные результаты так называемой „интуитивной среды“. Ил. 6, библ. 15 (на английском языке; рефераты на английском, русском и литовском яз.)

A. Kairys, V. Raudonis, R. Simutis. “iHouse” – pažangus gyvenamosios aplinkos valdymas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 4(100). – P. 37–42.

Siūloma „iHouse“ sistema, kurioje duomenų apdorojimo programinė įranga ir akies padėties sekimo sistema panaudojama bevieliam elektros įrenginių valdymui. Sistemoje naudojama prie invalido vežimėlio pritvirtinta internetinė kamera. Specialiai sistemai buvo sukurtas žymeklių rinkinys. Kiekvienas žymeklis naudojamas skirtingiems įrenginiams: elektrinėms durims, langams, šviesai, šildytuvui ir kt., identifikuoti. Žymekliai aplinkoje atpažįstami apdorojant kadrus, kurie užfiksuojami minėtaja kamera. Atpažinto žymeklio centro koordinatės palyginamos su pelės žymeklio koordinatėmis, kurios nustatomos akies sekimo sistema, ir taip intuityviai pasirenkami ir valdomi įrenginiai. Buvo atlikti tikslumo, greitaveikos ir patikimumo tyrimai. Aplinkos vaizdo kadras apdorojamas greičiau nei per 0,1 s, o žymekliai atpažįstami 84 % tikslumu. Eksperimentiniai duomenys rodo, kad intuityvi sistema yra tinkama naudoti realiu laiku. Il. 6, bibl. 15 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).