Optical and Acoustical Methods in Science and Technology 2013

Pattern Recognition Algorithm for Eye Tracker Sensor Video Data Analysis

K. Murawski[a,*] and K. Różanowski[b]

[a]Military University of Technology, Inst. of Teleinformatics and Automatics, S. Kaliskiego 2, 00-908 Warsaw, Poland

[b]Military Institute of Aviation Medicine, Aviation Bioengineering Dept., Z. Krasińskiego 54, 01-755 Warsaw, Poland

The aim of the paper is to discuss research conducted at the Military University of Technology as well as the Military Institute of Aviation Medicine. The essence of the research is to develop a pattern recognition algorithm for the analysis of data received from an eye tracker sensor. Data analysis can be done using computer vision algorithms. In the article the state of the art pattern recognition algorithms were presented. Particular attention was paid to the possibilities and limitations of their use. The main part of the paper is the presentation of the author's pattern recognition algorithm. In the paper we have also shown the results obtained by other techniques of image processing, which are frequently used to determine the position of the pupil of the eye on the image.

DOI: 10.12693/APhysPolA.124.509

PACS: 42.30.Sy, 42.30.Va, 07.05.Pj

1. Introduction

Sensory systems are an interesting combination of hardware and software. Measuring the value of the measured magnitude in these systems is conducted by hardware systems. For this purpose detectors using physical compounds and phenomena are used. Due to properties sensor solutions performing non-invasive and non-contact measurement are of particular interest. These include sensors recording light parameter changes resulting from the impact of measured external factors, recording the intensity of reflected light from the surface of the tested object and others. Representatives of the first group of sensors are: fiber-optic sensors for measuring the current [1], fiber-optic sensors monitoring the security level of the object [2], optical sensors for monitoring the operation of artificial chambers of the heart [3, 4]. The resulting operation of this type of sensor is the measurement result expressed as a number or as a binary signal. The second group consists of imaging sensors operating in the visual band, near and far infrared [5–6], using laser techniques [7] and terahertz technology [8]. These are the main types of imaging sensors.

The detectors used convert the recorded energy in the form of photons into electrical signals. These signals, after processing into digital form, depending on the organization of the detectors, are presented in the form of matrixes or a vector of numbers. A representative of these types of elements is the charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS) image sensors used in cameras, Fig. 1. Due to the speed during continuous operation (e.g. 340 readings per second from 2,228,224 detectors — Basler acA2000-340 km NIR) in the vast majority of cameras CMOS sensors are used. These image sensors also offer access to detectors (pixels) in any order. This allows users to define a specific region of interest (ROI), whose scan rate can be even higher. This property is used in the production of industrial cameras and sensors for motion-capture applications. This property is also used in the intelligent vision sensors such as the iVu TG sensor developed by the Banner Engineering company.

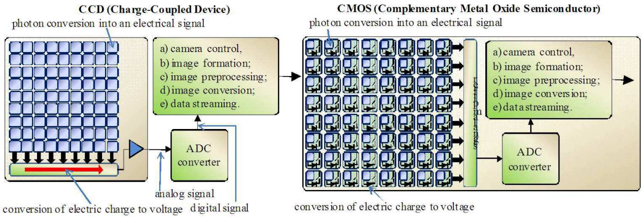

Fig. 1. Organization of CCD and CMOS sensors.

The development of smart sensors caused that part of the function is performed by the software [9, 10]. This trend is particularly noticeable in the area of sensing using optical sensors. One of the possible area of use of such sensors is eye-tracking [11] and laser photography [12]. In this case, the hardware of the sensor provides a picture frame, as well as frequently performing functions of image pre-processing. The appropriate processing is usually performed by specialized software.

The aim of the paper is to present the pattern recognition algorithm developed by the authors. In the paper we have also shown the results obtained by other techniques of image processing, which can be used to determine the position of the pupil of the eye on the image.

The developed algorithm was tested at a laboratory testing station, in a truck simulator as well as in real conditions. Images recorded in that visual band were obtained from a Logitech 9000Pro camera. In the infrared band images were provided by the sensor presented in [5].

[*]corresponding author; e-mail: k.murawski@ita.wat.edu.pl

2. State of the art video-based search algorithms of the head and eyes

There are many universal and specialized pattern recognition techniques that are presented in the literature. Some of them directly work on numerical data [13–15]. Other calculations use the interpretation of the data in the form of an image [16–20]. A small group of algorithms [16] is implemented in devices such as cameras.

Among the algorithms that can be used in the construction of a system for testing a driver's oculomotor activity (PBAO), prospective was: the support vector machine (SVM) [21], eigenvalue and eigenface [20], Viola Jones technique [16] and active contour [18].

As stated in [21], SVM has good recognition performance, which is supported by theoretical proof. The operation of SVM techniques depend explicitly on the number and image quality of searched for patterns. For the proper operation of SVM a process of learning based on images of the recognized object (patterns) and alternative images different from the pattern is required. In the case of PBAO software a pattern could be an image of the pupil. As a result, the algorithm should define its position on the image and determine the required state of the eye. Unfortunately, other items of the literature, e.g., [22], indicate that the learning algorithm and the determination of the position of the pattern on the image during normal operation requires a considerable amount of time. Therefore, SVM applications use patterns with a larger area of initially defining the area of interest. The area is then analyzed again by the SVM or processed by other techniques. Initial determination of the position of the eyes can also be implemented using the technique for determining the eigenvalue or eigenface. The authors implemented an algorithm for determining the minimum eigenvalue value. Processed color image was acquired from the Logitech 9000 Pro camera.

The result of processing is presented in Fig. 2. Image points determined in the calculations are represented by yellow dots.

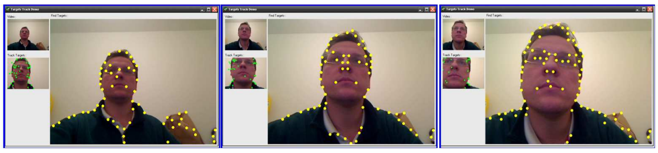

Fig. 2. Graphical representation of minimal eigenvalue.

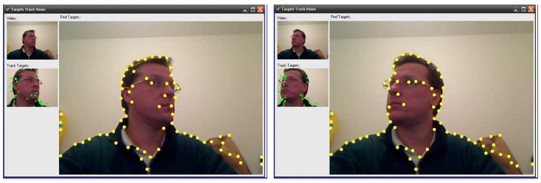

Fig. 3. Graphical representation of minimal eigenvalue during rotation of the head.

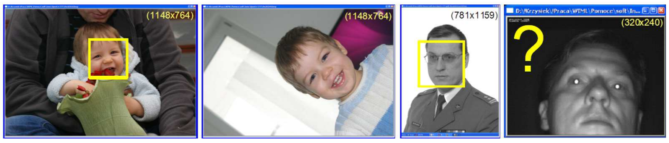

Fig. 4. Result of research of Viola Jones technique.

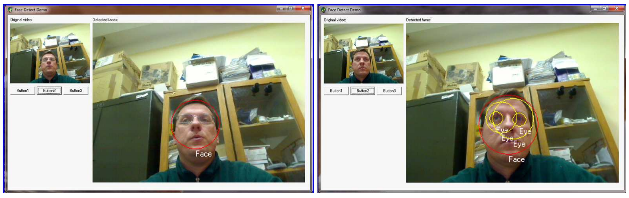

Fig. 5. Determining the position of the face and eyes in the case of motion.

Processing speed was achieved at the rate of 25 fps, which was a result of the digital camera used. It was noted that some of the characteristic elements of the face such as the corners of the mouth or nose base are almost always properly determined. These points were also visible during considerable limited visibility when turning the head, Fig. 3.

The results obtained during the day in visible light were very promising. Unfortunately, they could not be repeated at night. The built-in camera filter effectively cut out the infrared band, and its removal caused significant problems with the auto focus.

Equally high hopes were associated with the Viola Jones technique. This algorithm is often used to search for faces in the raster image. The study of its properties is relatively easy. It is implemented in the OpenCv library, which makes it available to a wide range of users. The testing of the algorithm was started by checking its performance on static images of different resolutions, Fig. 4.

As shown in Fig. 5 the algorithm correctly assigned the position of a face, when it was completely visible, and the size was contained in a 24 × 24 pixels square. In other cases, regardless of the image resolution, no identification was obtained or there were false readings. The obtained results are completely justified in the operation of the algorithm [16]. No identification was due to the type of classifiers implemented in the OpenCv library, which does not take into account picture rotation. The result of the algorithm operation was also affected by the learning process, and in fact the database of patterns on which teaching was carried out. Ultimately, the algorithm test was repeated. The experiment focused on the examination of the impact of scale as well as the motion of the searched object against the search result.

In the first case (scale effect) the study was carried out on an image, which was scaled to two different resolutions. In the experiment, all the faces on the images were properly determined. After scaling the image almost 50% of the people had, so-called, double faces. The second experiment ended in failure, in which the algorithm was tested during movement of the analyzed object. The facial area was almost always correctly indicated in the study, but in the case of the eyes their presence was not ascertained or an excess number of areas were indicated, Fig. 5.

The last checked was the Active Contour algorithm. The Active Contour searches for characteristic elements of an image using pattern recognition techniques, such as the face. It was considered to use Seeingmachines and Luxland commercial libraries in the project. In these libraries, all characteristic points were defined allowing determining not only the state of the eyes, but also facial expressions, Fig. 6a and b. Significantly better results were obtained using Luxland SDK in the study, Fig. 6c.

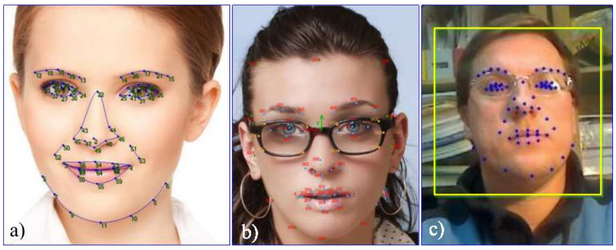

Fig. 6. Searching for characteristic points: Luxland library (a), Seeingmachines (b), the result of searching for facial elements using Luxland SDK (c).

This result would be entirely satisfactory, if not the requirements that must be met for the image processing to take place in real time. The manufacturer recommends the use of the Intel® Core$^{\text{TM}}$ i7 with a minimum of 2 GB of RAM. In the discussed studies these requirements have been fully met. An Intel® Core$^{\text{TM}}$ i7 920 computer was used clocked at 2.66 GHz for testing. During determining the characteristic facial points, Fig. 6c, eight processor cores were operating. Its usage was on an average of 51%, and memory usage was at 1.89 GB.

The requirements set by Luxland SDK and the results obtained from measurements caused that other solutions started to be looked for. As a result, an algorithm was developed that searches for the pupil regardless of the object rotating, the scale, and at a wide range of changes in illumination. The algorithm was tested on a computer with an Intel® Core$^{\text{TM}}$ i5 – 2500 (4 cores) clocked at a frequency of 3.30 GHz. Power usage of the CPU during normal operation (25 fps) averaged 13%.

3. Pattern recognition algorithm for eye tracker sensor

In the proposed algorithm, Fig. 7, the activity of the eyes is determined based on processing a single image frame. It is taken from the infrared camera with a minimum frequency equal to 25 Hz. The frame is then automatically normalized and analyzed to determine the value of the control signal for setting the camera exposure time and to determine the state, as well as location of the pupil in the image. The image is received in the RGB standard.

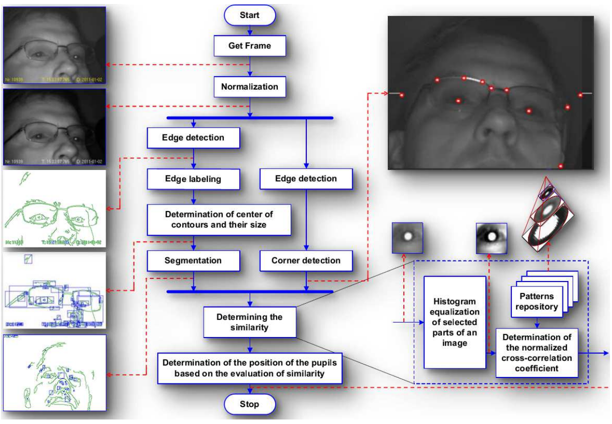
Start
Get Frame
Normalization
Edge detection
Edge labeling
Edge detection
Determination of center of contours and their size
Segmentation
Corner detection
Determining the similarity
Determination of the position of the pupils based on the evaluation of similarity
Stop
Histogram equalization of selected parts of an image
Patterns repository
Determination of the normalized cross-correlation coefficient

Fig. 7. Pattern recognition algorithm for the eye tracker sensor video data analysis.

Further processing of the image is performed only on the red image channel. During normalization the image is standardised to get a minimum (0) and maximum (255) constant value of brightness for the pixels. The image frame is then processed in parallel. At the same time, contours (Canny edge detector) and corners (Harris corner detector) are detected. The contours detected by the Canny edge detector are marked by the fast labeling algorithm. In the next step of the proposed algorithm the areas occupied by the contours are calculated. The image segmentation is performed taking into account the designated features and anthropometric properties of the eyes. During segmentation, contours with dimensions greater than 25 × 25 pixels are eliminated. Each contour whose width and height of the area differs more than 2 pixels is also removed from the list of potential positions of the pupil. The center of the area is calculated for the rest. The received points combined with the results of the Harris corner detector operator determine the probable location of the pupil, Fig. 8a. The final position of the pupil, Fig. 8b, is determined by calculating a measure of similarity between the patterns and analyzed image slices.

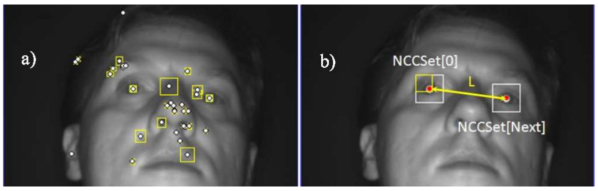

Fig. 8. The projected location of the pupil (a), result of calculation (b).

A few patterns were used in the algorithm. The patterns repository contains three images. Each image has the same width and height, but each represents different sizes of the pupil of the eye, Fig. 7. Pupil size represented by the patterns were established on the basis of the results given in [24], as well as during experiments conducted in simulated environment and real traffic. The similarity of the patterns to the image slices is specified by the values of normalized cross-corelation (NCC). The obtained values are sorted in descending order. In the last step of the algorithm there shall be designated areas: the first — with the highest value of cross-correlation coefficient and the second — with the next value of NCC and distant L from the first one by at least 30 pixels, Fig. 8b.

4. Result of experiments

The presented algorithm was tested together with the sensor for eye tracking in a car with a harsh environment. The study was conducted in real and laboratory conditions. The experiment is described in [5]. The research included 20 randomly selected individuals of different age (from 18 to 25 years of age) and gender. The results showed that in such conditions the developed algorithm worked properly. During the experiments in the laboratory the location and the status of the eyes were correctly specified. There were a few cases when only the position of one eye was determined or the eye was not detected. This was usually when the drivers did not look at the front windshield of the vehicle. In other situations, positions of the pupils and the state of the eyes were identified correctly. Sample results produced by the software were included in [5]. In a separate experiment (10 persons, 5 women and 5 men aged 26 years to 45 years) the impact of the color of the eye, age, contact lenses and glasses on the result of the determination of the eye position were tested. The obtained results are very promising. There was practically no significant effect of these factors on the result of the algorithm. Detailed results are shown in [25]. It means that the algorithm is able to work in real environment.

5. Conclusions

The paper presents a pattern recognition algorithm for the eye tracker sensor video data analysis. The task of the developed algorithm and sensor presented in [5] is to track the activity of the eyes and determine the psycho-physical condition characteristics of the driver. The algorithm has been implemented in the following languages: Object Pascal (Delphi) as well as .NET (Visual Studio 2010). Afterwards, the algorithm was tested along with the sensor in the laboratory environment and in road conditions during the day and night. The obtained results, including the photo, are presented in [5]. During the study we have also specified the impact of eye color, age, contact lenses and glasses on the result of algorithm's operation. Performed experiments have shown that the algorithm works as required. During the day, the algorithm was able to process up to 40 fps and save the result of work, including video data, on a SSD drive. At night, this value increased to 60 fps.

Acknowledgments

The project was co-financed by the European Union through the European Regional Development Fund under the Innovative Economy Program. Development Project contract no. POIG.01.03.01-10-085/09.

References

[1] K. Barczak, T. Pustelny, D. Dorosz, J. Dorosz, *Europ. Phys. J., Spec Top.* **154**, 11 (2008).

[2] M. Życzkowski, *Acta Phys. Pol. A* **122**, 933 (2012).

[3] G. Konieczny, Z. Opilski, T. Pustelny, E. Maciak, *Acta Phys. Pol. A* **116**, 344 (2009).

[4] G. Konieczny, T. Pustelny, *Acta Phys. Pol. A* **122**, 962 (2012).

[5] K. Różanowski, K. Murawski, *Acta Phys. Pol. A* **122**, 874 (2012).

[6] K. Murawski, R. Różycki, P. Murawski, A. Matyja, M. Rękas, *Acta Phys. Pol. A* **124**, 517 (2013).

[7] M. Piszczek, M. Kowalski, M. Szustakowski, K. Rutyna, K. Ludwikowski, *Acta Phys. Pol. A* **122**, 862 (2012).

[8] M. Piszczek, M. Kowalski, M. Szustakowski, *Acta Phys. Pol. A* **120**, 720 (2011).

[9] K. Murawski, *Przegląd Elektrotechniczny* **9**, 184 (2010) (in Polish).

[10] K. Murawski, in: *Proc. XV Int. Conf. on Methods and Models in Automation and Robotics (MMAR), Międzyzdroje (Poland)*, 2010, p. 356.

[11] M. Böhme, A. Meyer, T. Martinetz, E. Barth, in: *Proc. II Conf. on Communication by Gaze Interaction — COGAIN, Turin (Italy)*, 2006, p. 1.

[12] M. Piszczek, M. Kowalski, M. Karol, K. Rutyna, M. Zarzycki, M. Szustakowski, K. Ludwikowski, *Acta Phys. Pol. A* **124**, 550 (2013).

[13] C. Tyszkiewicz, A. Szpakowski, T. Pustelny, *Acta Phys. Pol. A* **116**, 419 (2009).

[14] P. Marczyński, A. Szpakowski, C. Tyszkiewicz, T. Pustelny, *Acta Phys. Pol. A* **122**, 847 (2012).

[15] C. Burgers, *Data Min. Knowl. Discov.* **2**, 121 (1998).

[16] P. Viola, M. Jones, in: *Proc. IEEE Workshop on Statistical and Computation Theories of Vision*, IEEE, Vancouver 2001.

[17] K. Mikołajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, *Int. J. Comput. Vision* **65**, 43 (2005).

[18] `http://research.microsoft.com/en-us/um/people/ablake/contours/`, 2011.

[19] M. Kass, A. Witkin, D. Terzopoulos, *Int. J. Comput. Vision* **1**, 321 (1988).

[20] M.A. Turk, A.P. Pentland, in: *Proc. Computer Vision and Pattern Recognition*, p. 586 (1991).

[21] E. Osuna, R. Freund, F. Girosi, in: *Proc. IEEE NNSP'97, Amelia Island*, 1997, p. 24.

[22] `http://faculty.ucmerced.edu/mhyang/papers/icip03_tutorial`, 2013.

[23] P. Campadelli, R. Lanzarotti, G. Lipori, *Biometr. Issue NATO Sci. Series* **18**, 234 (2007).

[24] K. Różanowski, K. Murawski *Acta Phys. Pol. A* **124**, 558 (2013).

[25] K. Murawski, K. Różanowski, M. Krej, *Acta Phys. Pol. A* **124**, 513 (2013).