An Efficiency Analysis of Augmented Reality Marker Recognition Algorithm

Dovilė Kurpytė (*Doctoral Student, Vilnius Gediminas Technical University*),
Dalius Navakauskas (*Professor, Vilnius Gediminas Technical University*)

Abstract **– The article reports on the investigation of augmented reality system which is designed for identification and augmentation of 100 different square markers. Marker recognition efficiency was investigated by rotating markers along *x* and *y* axis directions in range from −90° to 90°. Virtual simulations of four environments were developed: a) an intense source of light, b) an intense source of light falling from the left side, c) the non-intensive light source falling from the left side, d) equally falling shadows. The graphics were created using the OpenGL graphics computer hardware interface; image processing was programmed in C++ language using OpenCV, while augmented reality was developed in Java programming language using NyARToolKit. The obtained results demonstrate that augmented reality marker recognition algorithm is accurate and reliable in the case of changing lighting conditions and rotational angles – only 4 % markers were unidentified. Assessment of marker recognition efficiency let to propose marker classification strategy in order to use it for grouping various markers into distinct markers' groups possessing similar recognition properties.**

Keywords **– Computers and information processing; Computational efficiency; Augmented reality; Image recognition; Open source software.**

I. Introduction

Augmented reality (AR) is one of main future technological tools. It is more and more successfully applied to many areas of life [1], [2], for example: learning [3], playing games, in a military, architecture, science, medicine [4], and in mobile technology [5] – [7]. Augmented reality can inject a joyful element/detail into everyday product. It adds a sense of excitement to the process which appeals to a greater number of customers, and this can be the aim of marketing [8].

AR system without accurate image registration will not be accepted in many applications [9]. So one of the most important problems in AR to be perfected is real and augmented view matching, i.e., linking of different scene coordinates and their object coordinates with general coordinates [10]. Moreover, in order to reach the most effective AR results it is also important to assess the camera and used locations of objects. The features for the analysis usually are taken from view textures or collected data from sensors, such as: GPS [11], compass, gyroscope, magnetometer and linear accelerometers [12]. Non-marker AR systems are being oriented by objects, planes, their movements in the images although these kinds of AR systems are less accurate [1].

One of the most frequently used approach for image analysis in obtaining the necessary data are markers, providing benchmarks and distinguishing internal marker elements according to which the objects can be identified and focused on. To add more, markers are important AR tool because of their high degree of accuracy to capture [1], [4], [13]. The system should also be able to calculate the position of the camera using the detected marker. Four known points are sufficient to calculate the pose of a camera uniquely [14] and the simplest shape to acquire them as a square [1]. Also they allow pinpointing the coordinates for the environment and localizing the objects to be augmented [15].

It is of primary importance to define the conditions under which the AR system will operate correctly and to analyze the marker recognition algorithm effectiveness condition either [1], [13]. Several factors are available which can be analyzed while correctly identifying markers, i.e. the distance to the marker [13], [16], rotational angles and lighting conditions [2].

Those are the reasons why the markers rotational angles, different conditions for lighting with even/uneven light and shadows are observed and investigated in the following material. The recognition of 100 different square markers rotated by angle from −90° to 90° along the two *x* and *y* axes is performed. Under these conditions efficiency of marker recognition algorithm is assessed. The efficiency of marker recognition index is proposed and basing on it the markers are divided into three classes. The recommendations for markers internal elements textures are presented.

Throughout the research an open source library for different functions is being used. Open source software is free and built by passion communities of developers, and that makes innovations appear faster than closed source software. Another great reason to choose open source libraries is flexibility, because anyone can insert new source and adjust the algorithm to adapt the necessary operating conditions. And finally an open source model can result in powerful, secure, useful, industrial-strength software. For example, open source library NyARToolKit [17] is constantly improved and widely applied therefore it was used here, too.

II. Marker Recognition Algorithm Investigation Technique

In order to describe investigation technique of AR marker recognition algorithm primary we need to give comments on employed markers and their identity, then the outline selected marker recognition algorithm, afterwards present created virtual environments and finally in details present created AR system implementation.

A. Markers and their Identity

Different types of markers can be used for reality augmentation, where according to [1] and [13] the most common markers are: bars (e.g., QR, *DataMatrix*, *PDF-417*), circles (e.g., *CircleInner*, *CircleSplit*, *CircleOuter*, *SpotCode*), squares (ID markers, binary), and images (e.g., *StudierStube*). In [1] it is shown that square marker with a thick border is the most accurate for marker-based systems. This is because of:

- marker form – can be easily found even using different methods or algorithms;
- color contrast – even with different cameras color differences are easily noticed;
- border type – is more robust in most of the situations which can be applied.

Square marker can be created with different number of internal elements (4 × 4, 5 × 5 and up to 177 × 177) which directly influences the processing time. That is why in current research we used 4 × 4 internal elements markers taken from the University of Utah developed databases [18]. A composition frame of used markers is shown in Fig. 1.

All the markers have the same thickness border. Each employed marker has the ID assigned according to the Fig. 2 (where the first ID digit is the row number, the second – the column number), resulting in ID from 00 to 99.

All the markers are resistant to turns along z axis, i.e., turning the marker onwards (upwards) will not influence marker recognition accuracy [2].

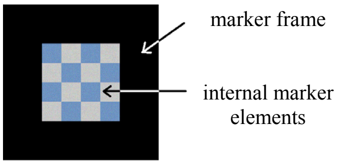


Fig. 1. Marker frame with outlined internal elements inside it.

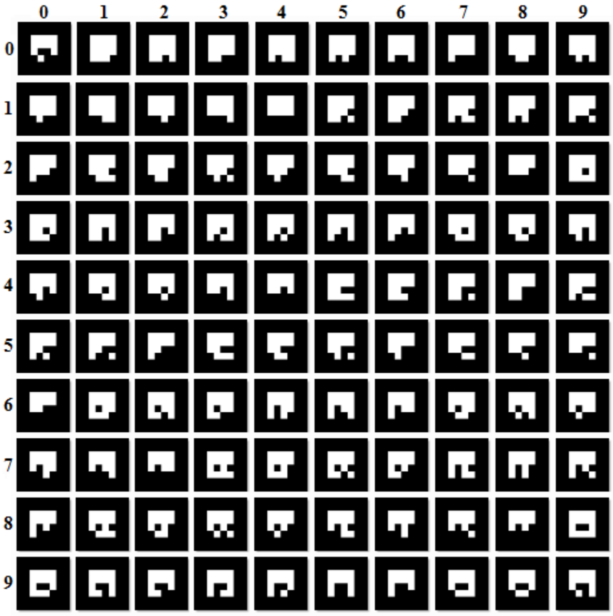


Fig. 2. Investigated markers and their corresponding ID = (row nr.; column nr.).

B. Marker Recognition Algorithm

The marker recognition algorithm is based on NyARToolKit library which roots from ARToolKit library that was adopted for Java programming language, as well as for other programming languages like C # and operating systems (e. g. Android) that support virtual machines. The main advantage of the use of NyARToolKit library is in the fact that here user viewpoint tracking problem is solved. Functions from the library allow tracking the view angle without losing the virtual object. This ensures a view of clear augmented representation.

Original marker recognition algorithm was proposed in [16]. Let us briefly comment its main processing stages. The algorithm starts with the input of image which afterwards using a lighting threshold is binarized and labeled. After that, when component list is received, starts contour detection. Based on contour detection results, sub-pixel corner detection is performed and found patterns are normalized. Based on camera intrinsic parameters and found components list marker regions list is built. Then a template matching is performed. Marker templates are taken from in advance prepared individual .patt files [19]. Marker size and lines parameters list taken from sub-pixel corner detection is processed to compute homography. After applying camera transformation markers transformation is received and used for final recognition step. In the case of successful recognition the recognized marker's ID number is output, otherwise recognition failure is indicated.

C. Virtual Environment Setup and Implementation

Virtual scenes prepared for experimental investigation are shown in Fig. 3. The following objects (with their possible coordinate systems) can be seen: the scene, the marker with ID 00, a camera and light sources. In order to simplify image processing, camera coordinate system was reduced (fixed).

During the experimentation four virtual scenes different in illumination were considered:

1. Light – with natural ambient light (Fig. 3, a) [2];
2. Twilight – with uniformly distributed shadow and blurred ambient lighting (Fig. 3, a).
3. Shadow 1 – with a bright light source from the left side (Fig. 3, b);
4. Shadow 2 – with the blurred light source from the left side (Fig. 3, b);

Processing in virtual environment is realized as follows. Every 5–25 ms marker (starting from marker with ID 00) is rotated by 1° along x axis from −90° to 90° in the first virtual environment scene Shadow 1, after that the rotation is performed along y axis. Next marker is selected and processed with a similar sequence of actions. After all 100 markers processing the recognition results are recorded in a matrix of size 181 × 200 elements (each element holds recognized marker index or failure sign). The chosen directed light source is from the left side of camera. Permanent place is created to determine the changes in characteristics of lighting intensity while the degree is changed. Afterwards virtual environment scene is changed and all the processing is repeated.

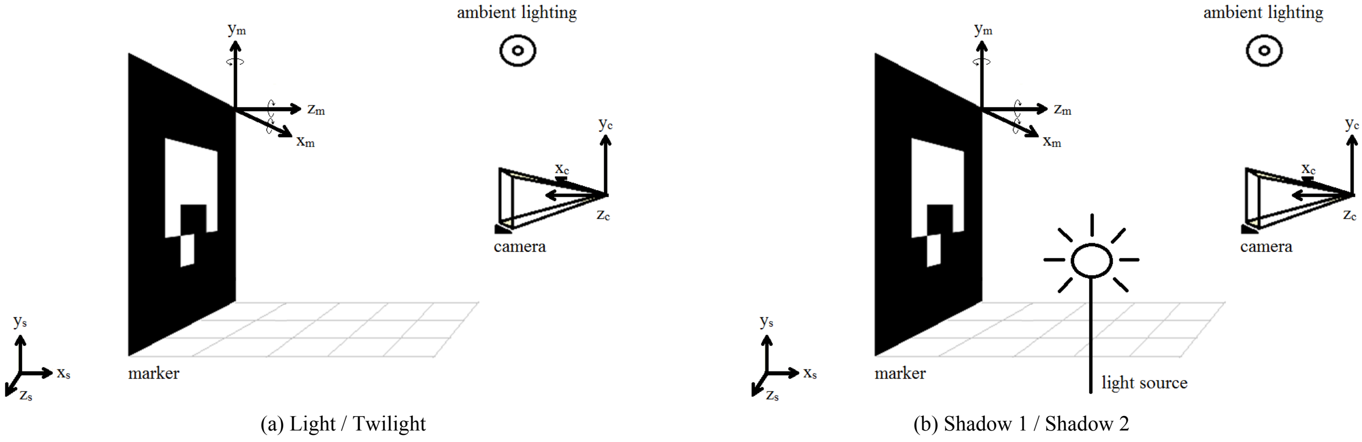


Fig. 3. Scenes of investigation in virtual environment.

D. Augmented Reality System Implementation

Augmented reality system can be created using computer and various gadgets with camera. Most popular gadgets for AR systems are: head-mounted, eyeglasses, contact lenses, virtual retinal display, handheld. In reality however, preparation for the test is difficult, because it has to have guaranteed coordinate system, completely controlled lighting conditions, maintain rotational angles along both axes and accuracy (every one degree), and all used tools errors shall not exceed the permissible. That is why, e.g., in [20] only few rotation angles (30°, 45°, 60°, 75°, 90°) were observed. All of the above mentioned problems are minimized or diminished if AR system testing is performed in a virtual environment.

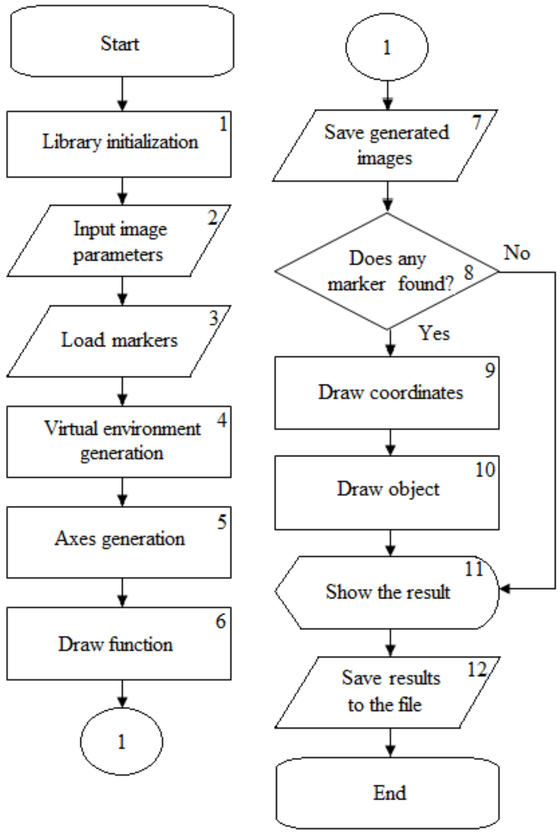


Fig. 4. Flowchart of augmented reality system implementation.

For investigation purposes created augmented reality system involves the following functional parts:

1. Image loading.
2. Image preprocessing.
3. Image processing with features extraction.
4. Augmented video data loading.
5. AR system video streaming.

Each part of the system according to its own operating specifics may be modified or revoked, depending on the desired results.

One of the ways to develop this kind of augmented reality system is shown in flowchart at Fig. 4. At Steps 1 and 2 libraries are initialized and image parameters are inputted. At Step 3 100 markers (shown in Fig. 2) are loaded. At Steps 4–7 virtual environment is generated, markers are rotated and processed (details were already discussed in Section II.B).

At Step 8 marker recognition is preformed (see Section II.B). At Steps 9–11 the coordinates and the object (in this case simple colorful cube with marker ID on top) are drawn, by showing if marker is detected and if detection is correct. At the final Step 12 all the data about markers recognition is written to a file.

III. The Experimental Investigation Results

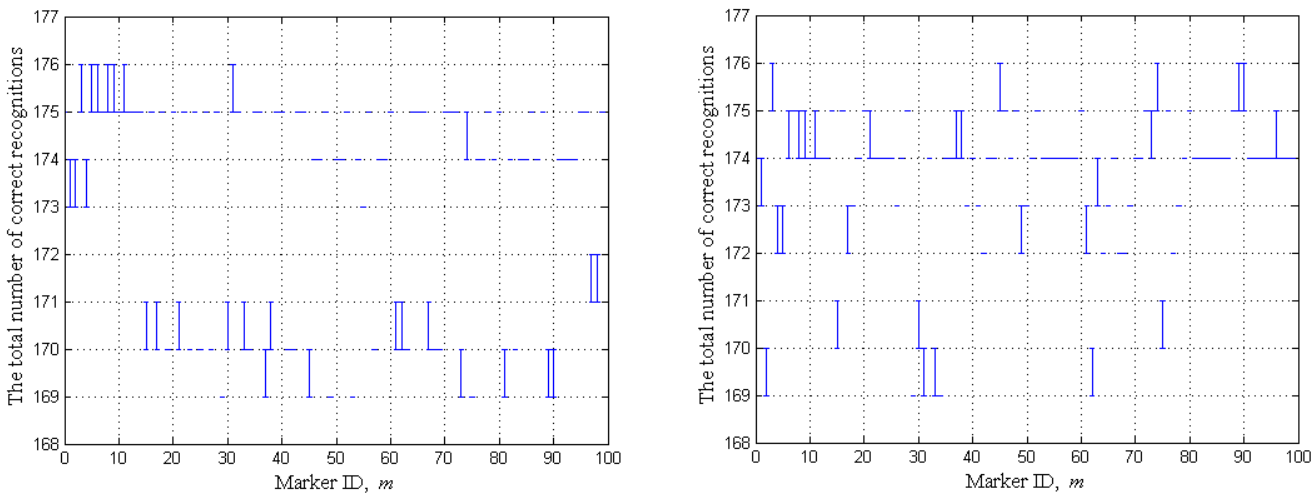
Presentation of experimental investigation results we start with the analysis of individual marker recognition results, then introduce to assessment of marker recognition efficiency, propose markers classification according to recognition efficiency index strategy and finally present markers classification results.

A. Individual Marker Recognition Results

Experiments were carried out in four different virtual environment scenes. A total number of created marker images were 144,800, calculated by:

$$N_{\mathrm{Total}} = N_{\mathrm{M}} \cdot N_{\mathrm{R}} \cdot N_{\mathrm{A}} \cdot N_{\mathrm{E}}, \qquad (1)$$

here: N_{M} – the number of used markers $N_{\mathrm{M}} = 100$; N_{R} – the number of rotations, $N_{\mathrm{R}} = 90 \cdot 2 + 1 = 181$; N_{A} – the number of axis, $N_{\mathrm{A}} = 2$; N_{E} – the number of environment scenes, $N_{\mathrm{E}} = 4$.



(a) Rotating along *x* axis

(b) Rotating along *y* axis

Fig. 5. Limits of the total number of correct recognitions of individual markers evaluated under marker rotation in four virtual environment scenes.

TABLE I

The Total Number of Correctly Recognized Markers in Four Virtual Environment Scenes While Rotating Markers Along *x* and *y* Axis

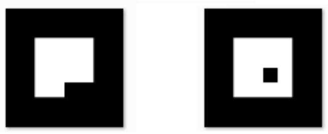
Values / Environment	The total number of recognized markers under rotation					
	Both axis		x axis		y axis	
	min	max	min	max	min	max
Light	169	176	169	**176**	169	176
Twilight	169	**175**	169	175	169	**175**
Shadow 1	169	176	169	175	169	176
Shadow 2	169	176	169	175	169	176

Fig. 5 presents the results on the total number of correct recognitions of individual markers evaluated under rotation (part a – along *x* axis, part b – along *y* axis) in four virtual environment scenes.

Table 1 summarizes the same investigation results presenting the total number of correctly recognized markers in four scenes while rotating markers along *x* and *y* axis.

The influence of the environment on the algorithm precision is small, as differences of the total number of correct recognitions of individual markers varies only up to one unit. Most of the markers (72 % in *x* axis and 73 % in *y* axis rotation direction) are resistant to the changes in all four environments (shown as dots in Fig. 5).

Markers were correctly identified by turning them in the range from −87° to 87°, an average of unrecognized rotational angles is just 7° per marker out of 181°. It could be noted that in all four considered virtual environment scenes, the same markers (see Fig. 6) were found to be the best (ID 03) and the worst (ID 29).



(a) ID 03 – best (b) ID 29 – worst

Fig. 6. The best (a) and the worst (b) recognized markers.

B. Assessment of Marker Recognition Efficiency

Minimal angle along which rotated in all directions (along *x* and *y* axis) in a specific virtual environment scene *e* marker *m* is still recognized should be expressed by:

$$\delta_m^{\min}\left(e\right)=\min\left(\underset{|\alpha|,|\beta|}{\operatorname{argmin}}\left(r_m\left(\alpha,\beta,e\right)=1\right)\right), \qquad (2)$$

here: α and β are angles of rotations along *x* and *y* axis correspondingly; *e* – virtual environment scene index; r_m – true markers *m* recognition result (1 – recognized; 0 – unrecognized).

Maximal angle along which rotated in at least one direction (along *x* and *y* axis) in a specific virtual environment scene *e* marker *m* is still recognized:

$$\delta_m^{\max}\left(e\right)=\max\left(\underset{|\alpha|,|\beta|}{\operatorname{argmax}}\left(r_m\left(\alpha,\beta,e\right)=1\right)\right). \qquad (3)$$

Keep noted that there is no guarantee that specific marker rotated by angle from $\delta_m^{\max}(e)$ till $\delta_{m.}^{\min}(e)$ will be correctly recognized, i.e., angles $[+\delta_m^{\min}(e),-\delta_m^{\min}(e)]$ could be thought as most conservative (guaranteed in all considered directions), while angles $[+\delta_m^{\max}(e),-\delta_m^{\max}(e)]$ – most optimistic (results will be acceptable at least in one rotation direction).

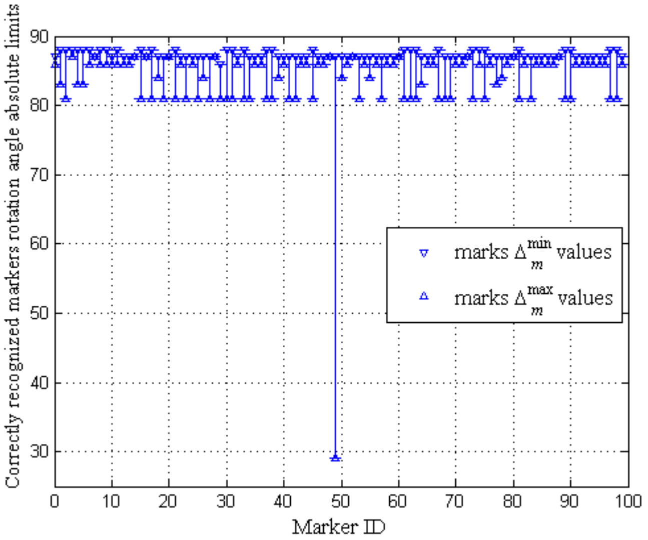


Fig. 7. Limiting angles along which rotated in all considered directions and virtual environment scenes markers are still recognized.

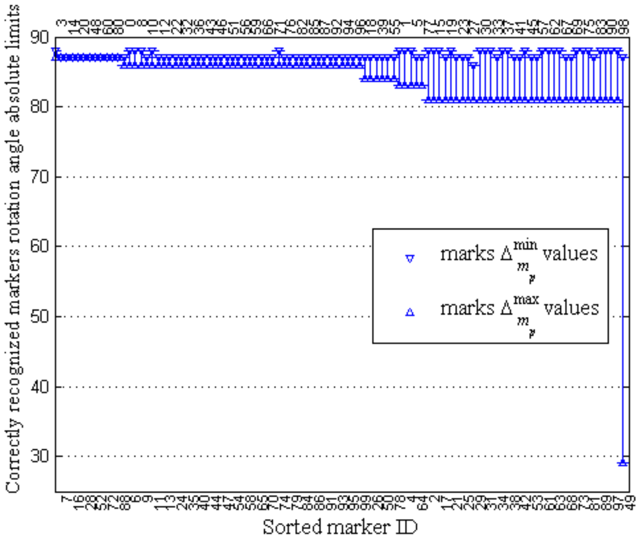


Fig. 8. Marker recognition results sorted using single criteria (4a). Note that marker IDs are shown pair wise at the bottom and the top of axis.

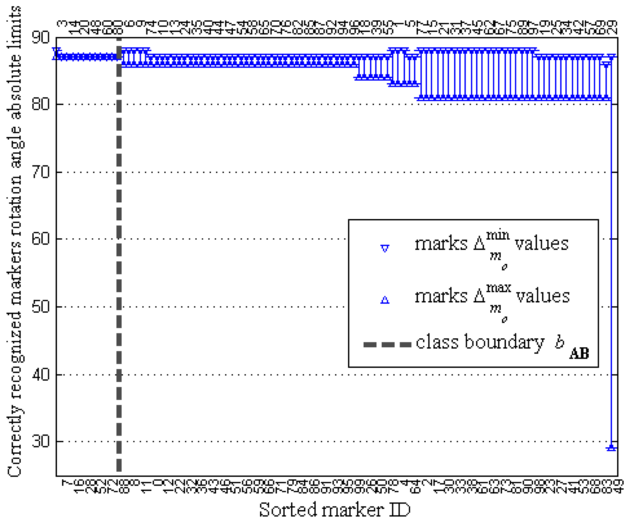


Fig. 9. Marker recognition results sorted using two-fold criteria: (4a) and (5) . Note: marker IDs are shown pair wise at the bottom and the top of axis; boundary between **A** and **B** marker classes is indicated by dash line.

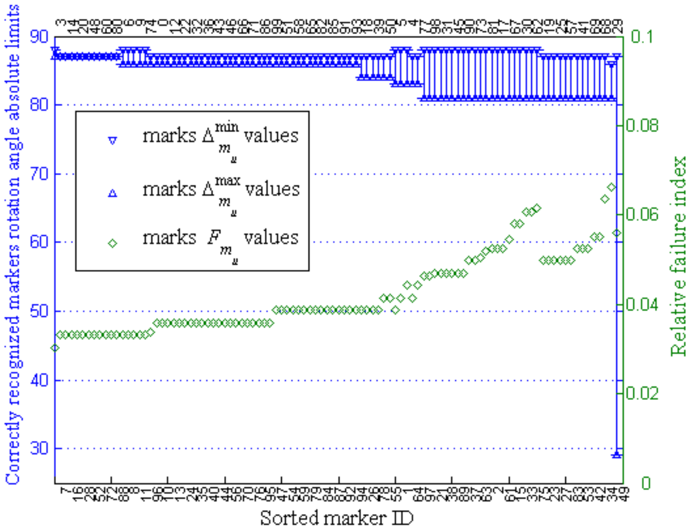


Fig. 10. Marker recognition results sorted using three-fold criteria: (4a), (5) , and (8). Note: marker IDs are shown pair wise at the bottom and the top of axis; relative failure index F_m values are indicated at the right side of axis.

In order to take into account variability of recognition results in all considered virtual environments let us introduce two more derivative parameters:

$$\Delta_m^{\min} = \min_e \delta_m^{\min}(e), \tag{4a}$$

$$\Delta_m^{\max} = \max_e \delta_m^{\max}(e). \tag{4b}$$

Here $\Delta_m^{\min}$ is minimal and $\Delta_m^{\max}$ is a maximal angle along which rotated in all considered directions and virtual environment scenes marker m is still recognized. Results computed according to (4) are presented graphically in Fig. 7.

Then angular discrepancy (sort of risk) associated with particular marker to be unrecognized could be expressed as:

$$\Delta_m = \Delta_m^{\max} - \Delta_m^{\min} - 1. \tag{5}$$

Based on $\Delta_m^{\min}$ and Δ_m values individual marker's recognition efficiency can be assessed.

C. Markers Classification Strategy

Marker recognition results according to $\Delta_m^{\min}$ can be simply sorted as follows:

$$m_p = \underset{m}{\operatorname{argsort}}\left(\Delta_m^{\min} \to 0\right), \ \forall\, \Delta_{m_p}^{\min} \ge \Delta_{m_{p+1}}^{\min},\ p \in [0, 99]. \tag{6}$$

However such approach will yield ambiguous results (see Fig. 8, where Δ_{mp} chaotic changes are noticeable).

More consistent results could be achieved by two-fold sorting, namely:

$$m_o = \underset{m}{\operatorname{argsort}}\left(\Delta_m^{\min} \to 0\Big|_{\Delta_{m_o} \ge \Delta_{m_{o+1}}}\right),$$
$$\forall\, \Delta_{m_o}^{\min} \ge \Delta_{m_{o+1}}^{\min},\ o \in [0, 99]. \tag{7}$$

Marker sorting results according to (7) are illustrated in Fig. 9.

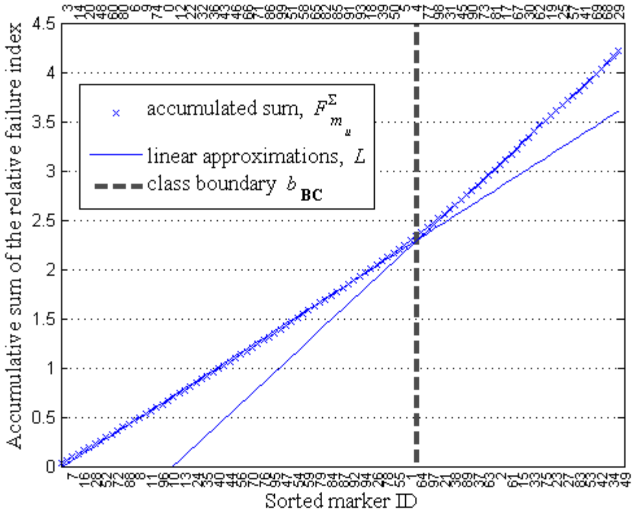


Fig. 11. Accumulative sum of the relative failure index dependency on marker recognition results sorted using three-fold criteria. Note: marker IDs are shown pair wise at the bottom and the top of axis; the best LMS approximation by two linear functions is indicated by solid lines; boundary between **B** and **C** marker classes is indicated by dash line.

The angular discrepancy associated with particular marker presented in Fig. 9 could be accorded with one more parameter – relative failure index – the total number of angular positions in which investigated recognition algorithm fails to give correct recognition results:

$$F_m = \frac{1}{4}\sum_{e=1}^{4} f_m(e), \tag{8a}$$

$$f_m(e) = \frac{1}{362}\sum_{\alpha=-90}^{90}\sum_{\beta=-90}^{90}\left(1 - r_m(\alpha,\beta,e)\right). \tag{8b}$$

Taking the third worth for sorting factor into account sorting expression (7) can be augmented by:

$$m_u = \underset{m}{\operatorname{argsort}}\left(\Delta^{\min}_{m} \to 0\Big|_{\Delta_{m_u} \ge \Delta_{m_{u+1}} \cap\, F_{m_u} \le F_{m_{u+1}}}\right),$$
$$\forall\, \Delta^{\min}_{m_u} \ge \Delta^{\min}_{m_{u+1}},\, u \in [0, 99]. \tag{9}$$

The relative failure index aligned with previous data is presented in Fig. 10.

Accumulative sum of the relative failure index

$$F^{\Sigma}_{m_u} = \sum_{i=0}^{u} F_{m_i}, \quad u \in [0, 99], \tag{10}$$

is not linear (see Fig. 11) thus could serve as indicator for markers division into classes.

D. Marker Classification Results

According to the proposed strategy for markers classification, all considered marker sets can be divided into three marker classes: **A**, **B** and **C**, rated by marker recognition efficiency in decreasing order, correspondingly. **A** marker class is the most reliable, while **B** and **C** marker classes possess some associated risks.

Boundary between **A** and **B** marker classes is found based on most conservative criteria – $\Delta^{\min}_{m_o}$, i.e.:

$$b_{\mathbf{AB}} = \underset{m_o}{\operatorname{argmax}}\, \Delta^{\min}_{m_o}. \tag{11}$$

According to the considered marker set $b_{\mathbf{AB}} = 11$ (see Fig. 9 where found boundary is indicated by dash line).

Boundary between **B** and **C** marker classes is found by the following optimization procedure:

$$b_{\mathbf{BC}} = \underset{u_{\mathbf{BC}}}{\arg}\left(\min_{\substack{a_1, b_1\\ u \in [0,\, u_{\mathbf{BC}}]}}\left[\frac{1}{u_{\mathbf{BC}}+1}\left(F^{\Sigma}_{m_u} - L_1(a_1, b_1; m_u)\right)^2\right] + \right.$$
$$\left. + \min_{\substack{a_2, b_2\\ u \in [u_{\mathbf{BC}},\, 99]}}\left[\frac{1}{(99-u_{\mathbf{BC}})}\left(F^{\Sigma}_{m_u} - L_2(a_2, b_2; m_u)\right)^2\right]\right) =$$
$$= \underset{u_{\mathbf{BC}}}{\arg}\left(\min_{\substack{a_1, b_1\\ u \in [0,\, u_{\mathbf{BC}}]}}\left[\frac{1}{u_{\mathbf{BC}}+1}\left(F^{\Sigma}_{m_u} - a_1 m_u + b_1\right)^2\right] + \right.$$
$$\left. + \min_{\substack{a_2, b_2\\ u \in [u_{\mathbf{BC}},\, 99]}}\left[\frac{1}{(99-u_{\mathbf{BC}})}\left(F^{\Sigma}_{m_u} - a_2 m_u + b_2\right)^2\right]\right), \tag{12}$$

where L_1 and L_2 are two linear functions with corresponding parameters a_n and b_n, that fit by not overlapping pieces

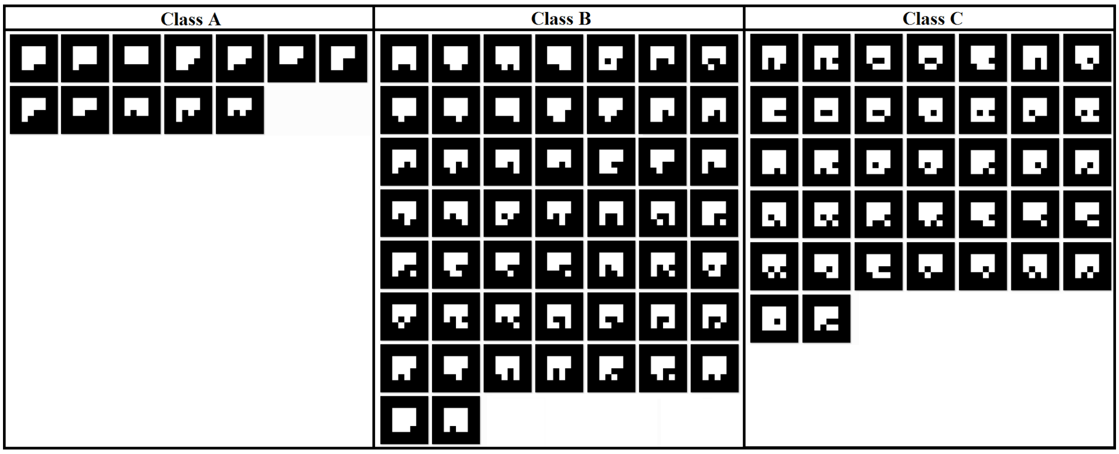


Fig 12. Markers classified into three classes according to the recognition efficiency: **A** class – most reliable; **B** class – reliable; **C** class – most risky.

in the least mean square sense to the accumulative sum of the relative failure index $F^{\Sigma}_{m_u}$; $u_{\mathbf{BC}}$ is a point where linear functions intersect and guarantee optimum.

Performed by (11) optimization yielded the following results: $a_1 = 0.0367$; $b_1 = -0.0137$; $a_2 = 0.0518$; $b_2 = -0.09353$; $u_{\mathbf{BC}} = 62$. In Fig. 11 graphical interpretation of the final result is shown, where lines are presented by solid line, boundary between marker classes – by thick dashed line.

Markers classification results are summarized in Fig. 12, where all markers belonging to a particular class are pictured.

Marker recognition average in class **A** is 96.7 %, in class **B** – 96.3 %, and in class **C** – 94.8 %. Class **A** marker's internal elements have common points with the frame and have existing horizontal/vertical central line symmetry or equivalent asymmetry. Class **C** markers have "flying" internal marker elements, none or fewer connections with marker's frame.

IV. Conclusions

1. Specialized augmented reality system exploiting virtual environment scene capabilities for individual square markers' properties investigation was created and researched. The system is suitable for other kinds of markers investigation.

2. Experimental investigation of 100 square markers recognition efficiency under changes in lighting and rotation angles was carried out. Individual markers' recognition efficiency was assessed.

3. Novel strategy for markers classification was proposed and investigated marker set was classified according it into three classes.

4. While developing a resistant to rotational angles and different lighting conditions marker it is recommended to keep the vertical and horizontal middle aspect symmetry and for internal marker elements to have common points with marker border.

References

[1] S. Siltanen, Theory and applications of marker-based augmented reality, 2012.

[2] D. Kurpytė, "Papildytos realybės algoritmo efektyvumo tyrimas," Mokslas – Lietuvos ateitis, Vol. 5, Issue 2, pp. 79–83, 2012.

[3] T. Yamabe and T. Nakajima, "Playful training with augmented reality games: case studies towards reality-oriented system design," Multimedia Tools and Applications, Vol. 62, Issue 1, pp. 259–286, 2013.

[4] M. Mahvash and L. Besharati Tabrizi, "A novel augmented reality system of image projection for image-guided neurosurgery," Acta Neurochir, Vol. 155, Issue 5, pp. 943–947, 2013.

[5] S. Martin, G. Diaz, E. Sancristobal, R. Gil, M. Castro and J. Peire, "New technology trends in education: Seven years of forecasts and convergence," Computers & Education, Vol. 57, Issue 3, pp. 1893–1906, 2011.

[6] R. Azuma, M. Billinghurst and G. Klinker, "Special Section on Mobile Augmented Reality," Computers & Graphics, Vol. 35, pp. VII–VIII, 2011.

[7] H.-Y. Chang, H.-K. Wu and Y.-S. Hsu, "Integrating a mobile augmented reality activity to contextualize student learning of a socioscientific issue," British Journal of Educational Technology, Vol. 44, Issue 3, pp. 95–99, 2013.

[8] M. Bulearca and D. Tamarjan, "Augmented Reality: A Sustainable Marketing Tool?" Global Business and Management Research: An International Journal (GBMR), Vol. 2, Issue 2, pp. 237–252, 2010.

[9] R. Azuma, "A survey of Augmented reality," In Presence: Teleoperators and Virtual Environments, Vol. 6, Issue 4, pp. 355–385, 1997.

[10] R. Azuma, Y. Balliot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, "Recent Advances in Augmented Reality," Computer Graphics and Applications, IEEE, Vol. 21, Issue 6, pp. 34–47, 2001.

[11] R. Behringer, Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors, Virtual Reality, IEEE, pp. 244–251, March 13–17, 1999, Houston, Texas, USA.

[12] T. Langlotz, C. Degendorfer, A. Mulloni, G. Schall, G. Reitmayr and D. Schmalstieg, "Robust Detection and Tracking of Annotations for Outdoor Augmented Reality Browsing," Computers & Graphics, Vol. 35, Issue 4, pp. 831–840, 2011.

[13] A. C. Rice, R. K. Harle and A. R. Beresford, "Analyzing fundamental properties of marker-based vision system designs," Pervasive and Mobile Computing, Vol. 2, Issue 4, pp. 453–471, 2006.

[14] R. ~I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[15] J. Yeol Lee, D. Woo Seo and G. Won Rhee, "Tangible Authoring of 3D Virtual Scenes in Dynamic Augmented Reality Environment," Computers in Industry, Vol. 62, Issue 1, pp. 107–119, 2011.

[16] H. Kato and M. Billinghurst, Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, 2nd IEEE and ACM International Workshop on Augmented Reality, pp. 85–94, October 20–21, 1999. San Francisco, California, USA.

[17] NyARToolKit, "プロジェクト," 2012. [Online]. Available: http://nyatla.jp/nyartoolkit/wp/?page_id=55. [Accessed: Oct. 22, 2013].

[18] University of Utah, "Augmented reality – ARToolKit Patternmaker," July 2002. [Online]. Available: http://www.cs.utah.edu/gdc/projects/augmentedreality/ [Accessed: Oct. 22, 2013].

[19] M. Tahir, A. M. Arshad, A. Hafeez, M. Yazdanie and S. Ziauddin, "Interactive Slide Navigation: An Approach for Manipulating Slides with Augmented Reality Markers," Smart Computing Review, Vol. 2, Issue 2, pp. 138–148, 2012.

[20] X. Zhang, S. Fronz and N. Navab, Visual Marker Detection and Decoding in AR Systems: A Comparative Study, International Symposium on Mixed and Augmented Reality, pp. 97–106, 2002.

Dovilė Kurpytė received her BSc degree in Electronics engineering and MSc with honour degree in Informatics engineering from Vilnius Gediminas Technical University, Lithuania, in 2009 and 2013, respectively. Also received BSc degree in Management and Business Administration from Vilnius University, Lithuania, in 2013.

She currently is a doctoral student and works as a laboratory assistant at Vilnius Gediminas Technical University, Lithuania. Her research interests include image processing, bioengineering and 3D graphics.

She is a member of the IEEE, IEEE Women in Engineering (WIE) and IEEE Engineering in Medicine and Biology (EMBS) Societies.

Address: Vilnius Gediminas Technical University, Faculty of Electronics, LT-03227, Naugarduko str. 41–413, Vilnius, Lithuania.

Email: dovile.kurpyte@vgtu.lt

Dalius Navakauskas received honour diploma of Radio-Electronics Engineer in 1992, MSc in Electronics degree in 1994, doctor of Electrical and Electronic Engineering degree in 1999, passed habilitation procedure in Informatics Engineering in 2005, received professor title in 2008 all at Vilnius Gediminas Technical University.

He is a professor in the Department of Electronic Systems, Vilnius Gediminas Technical University, Lithuania. Main research interests include computational intelligence, signal and image processing, bioinformatics.

D. Navakauskas is a senior member of the IEEE, active member of IEEE Computational Intelligence and Signal Processing Societies, currently serves as IEEE Lithuania Section Chair.

Address: Vilnius Gediminas Technical University, Faculty of Electronics, LT-03227, Naugarduko str. 41–422, Vilnius, Lithuania.

Email: dalius.navakauskas@vgtu.lt