

Optical Driving for a Computer System with Augmented Reality Features

Tomasz Pałys
Military University of
Technology
Kaliskiego Str. 2,
01-489 Warsaw, Poland,
Email:
tomasz.palys@wat.edu.pl

Krzysztof Murawski
Military University of
Technology
Kaliskiego Str. 2,
01-489 Warsaw, Poland,
IEEE Member # 92707852
Email:
krzysztof.murawski@wat.edu.pl

Artur Arciuch
Military University of
Technology
Kaliskiego Str. 2,
01-489 Warsaw, Poland,
Email:
artur.arciuch@wat.edu.pl

Andrzej Walczak
Military University of
Technology
Kaliskiego Str. 2,
01-489 Warsaw, Poland,
Email:
andrzej.walczak@wat.edu.pl

Abstract— This article proposes a laser beam encoding method that is used to control an augmented reality system. Experiments were performed using a red laser emitting a wavelength of $\lambda = 650$ nm and a power of $P = 3$ mW. The purpose of the study was to investigate the methods of modulation and demodulation of the encoded laser signal, and to examine the influence of parameters such as laser pulse duration, camera image resolution, the number of recorded frames per second on the demodulation result of the optical signal.

The results show that the proposed coding method provides the transmission of the necessary information in a single laser beam (no less than 36 codes with a decoding efficiency of 99.9%). The developed coding method enables, based on the sequence analysis of video images, the influence on the course of the simulation performed in augmented reality, including distinguishing players and actions taken by them. This is an important advancement in relation to interaction systems used to influence augmented reality.

I. INTRODUCTION

The impact on augmented reality (AR) can occur while using different types of devices [1 – 3]. It consists in producing intentional and previously planned possible behavior of the system, which for the “player” will create the impression of interaction [4]. The coding and decoding system [5] plays a key role in such communication. Its most important parameters include the code capacity and the time of encoding and decoding the transmitted information. In the case under consideration, the data to be encoded is obtained from the user. They arise as a result of interacting with him/her through patterns or objects presented in the image that force the user to behave in a certain way. This behavior is recognized by it and analyzes the AR system. For this purpose, markers [6, 7], 9 DOF sensors [8] or camera systems including 3D cameras [9, 10] are used. In the augmented reality system that is based on tags, the player is “stuck” with markers and then observed by an optical system consisting of multiple cameras. A similar effect is obtained when a player uses 9 DOF sensors to visualize its movement, Fig. 1. The position and activity of the player is then determined on the basis of data received from

accelerometers, gyroscopes and magnetometers [11]. In the solution shown in Fig. 1, 14 sensors were used to determine the position and movement of the player. In other designs, motion is determined using methods that determine local depth maps [12]. On their basis information undergoes synthesization and selected behaviors of the player are identified [13].

The commercial AR system used in the study has been enhanced with a proprietary control system, the main components of which are the video camera (receiver), the signal decoder (microprocessor device) and the manipulator (mock-up weapon) along with the encoder (encoder). The transmitter uses amplitude modulation of the laser beam. The laser beam is reflected from the screen and recorded by the video camera. The attractiveness of the suggested approach is the ability to influence AR even when the distance between the screen and the player reaches 600 m.

The developed coding method eliminates problems identified as essential in [14]. The coding presented in paper [14] was based on the data transmission protocol used in the RS-232 standard.

II. HARDWARE DESCRIPTION

The laser beam modulation is implemented by the encoder, Fig. 2. The main component of the encoder is the AT89s8253 microcontroller clocked at $f = 22.1184$ MHz. The encoder performs the signal coding algorithm discussed in point IV. The result of the algorithm operation, depicted in the form of a sequence of logic signals (binary values), is

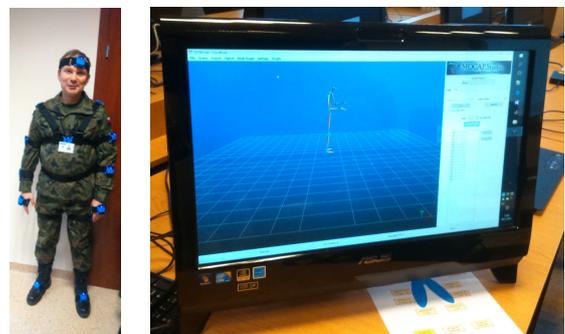


Fig. 1 View of player equipped with a 9 DOF sensors system

This work was not supported by any organization

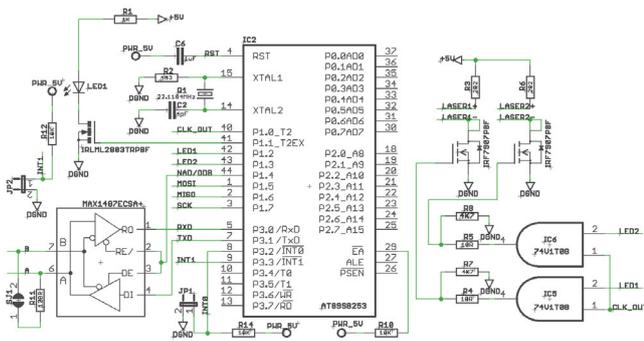


Fig. 2 Diagram of the laser beam modulation system (main elements)

taken out in series to the microcontroller terminals (indicated during configuration) – P1.2 or P1.3. These terminals, in the case of a "1" logic input, add power to the laser diodes power supply (not shown in Fig. 2). The developed version of the system and software makes it possible to simultaneously control two laser beams. In addition, they can be keyed by a binary signal or a rectangular wave generated by the microcontroller on the P1.0 terminal. This property was used in work [15 - 18] to control the brightness of the near-infrared illuminator. Activation of P1.2 or P1.3 terminals occurs after identifying the falling edge of the signal at the P3.2 terminal of the microcontroller (JP1 connector). In the presented layout, the button attached to the JP1 connector was taken out for the user. Similarly, in order to enable manual and remote configuration of the system, a button attached to the JP2 connector was taken out as well as communication connectors operating in the RS - 485 standard. Giving a high signal on line P0.4 (as a result of JP2 connectors short-circuiting) puts the system into the configuration state. The button attached to the JP1 connector is then used to select the player identifier (*Player ID*) in the computer system. The selection is signaled to the user by a LED attached to the P1.1 terminal of the microcontroller and stored in a non-volatile memory (*EEPROM*). These actions can also be performed from the parent program when an encoder device is attached to a PC computer using a communications connector.

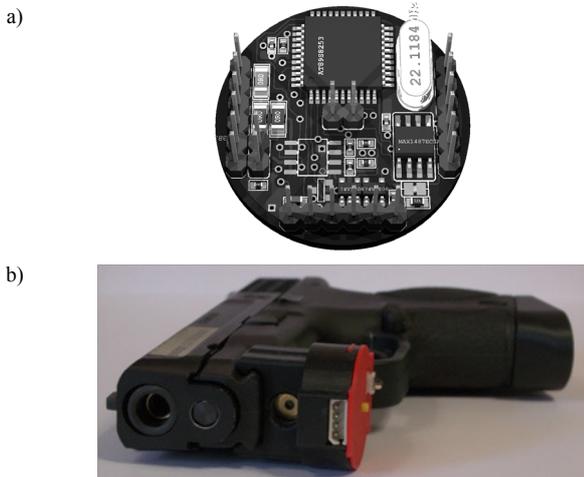


Fig. 3 Laser signal encoding module (a), view of installed device (b)

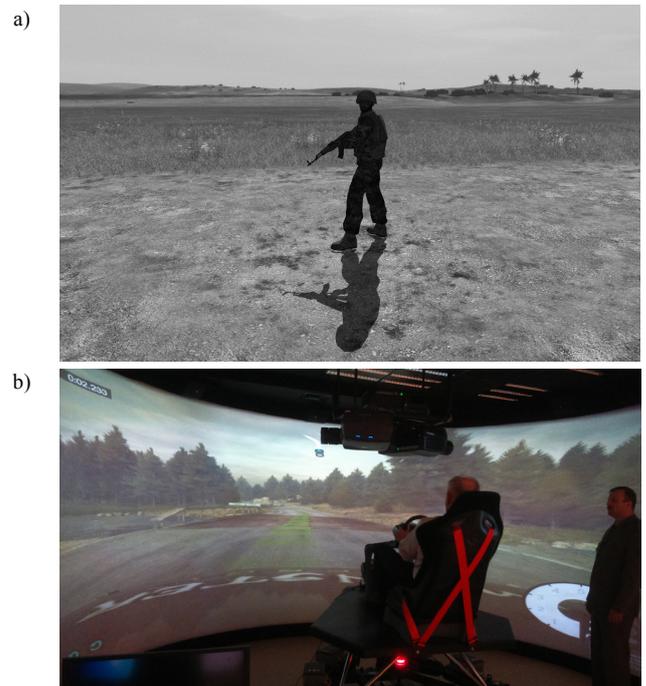


Fig. 4 Augmented reality environment: frame example (a), visualization system along with the screen in the shape of a half cylinder (b)

The view of the laser coding system is shown in Fig. 3a, and its location on a model of a gun is shown in Fig. 3b.

III. MEASUREMENT SYSTEM CONFIGURATION

The study of laser beam coding and decoding techniques was performed under controlled laboratory conditions. The experiment was conducted using the Manta G - 201 Allied Vision camera equipped with a focal length lens of $F = 8$ mm and a bandpass filter for which λ was equal 650 nm and the window width $\Delta\lambda$ was ± 10 nm, and the OptiTrack 120 Slim camera with a lens with a focal length of $F = 8$ mm. The laser beam produced by a laser module with a power of $P \approx 3$ mW emitting a wavelength of $\lambda = 650$ nm was subjected to modulation. The laboratory stand was created using commercial Virtual Battlespace 3 system (VBS3), which has been supplemented by encoding device developed by the authors, Fig. 3, as well as laser beam decoding device. VBS3 software was used to generate scenarios in which the AR system affected the user. An example scene is shown in Fig. 4a. The generated image was transferred to the 7th Sense Delta 2208 multimedia server. The server split the input image between the three projectors that displayed it at 1920 px x 1200 px. ProjectionDesign F35 AS3D WUXGA projectors were used in the study as well as a half cylindrical screen, Fig. 4b, measuring 12 m x 3 m. The user was about 3 m from the surface of the screen. During the test, the player was equipped with a Smith & Wesson Springfield weapon model, supplemented with the coding system shown in Fig. 3a. The task of the player was to shoot in strictly defined situations, enforced by the AR system. These shots were observed by the camera and then decoded in real time.

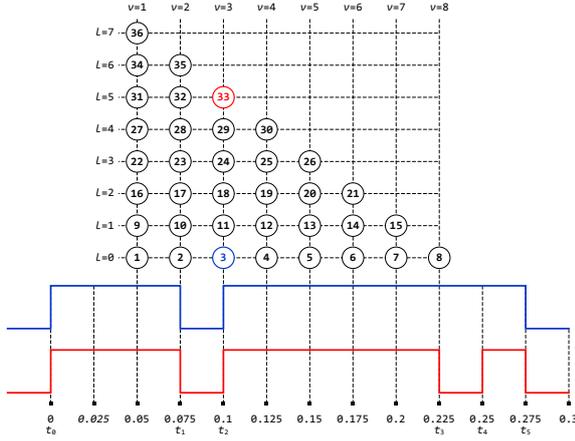
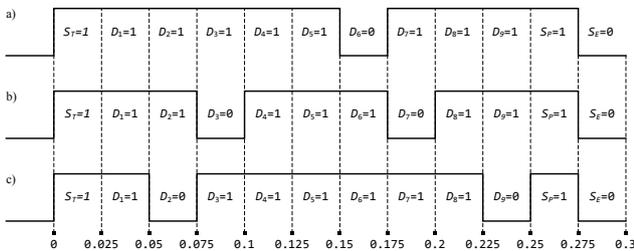


Fig. 5 Diagram of encoding symbols and code frame for symbol 33

IV. METHOD OF LASER BEAM CODING

During developing the laser signal coding method, $N=36$ encoded symbols were assumed. The default state of the laser module is “off”, which is manifested by the lack of a laser spot on the screen, and thus on the image. At the same time, it was assumed that displaying a single bit frame of code T_W , frame format $[D_T, D_1, \dots, D_{L+1}, D_P, D_E]$, where D_T – start bit, D_P – stop bit, D_E – frame end bit, D_k – code bit for $1 \leq k \leq L+1$. Parameter L is dependent on N and is equal to $L=\text{round}(\text{sqrt}(2N))$. In the layout of Fig. 2, the next frame bits of code is formed by terminal P1.2, which is responsible for turning on and off the laser module. The start bit and stop bit always have a value of one. The D_E bit always assumes zero. This corresponds to the transition from the state of emission of subsequent frame bits to the deactivation of the laser module. The impact of the T_W bit emission time on the effectiveness of decoding transmitted symbols has been verified in the studies of the coding system. The dependency of the required time of emitting the T_W bit as a function of the frequency of images obtained from the camera was also determined. The principle of encoding a frame is presented in Fig. 5. Thirty six symbols, $N=36$, were arranged on eight levels $L=8$. At the level of index l equaling zero, eight symbols were placed. Each higher level contains one symbol less. At the last level $l=L-1$ only one symbol is placed. For each symbol, for each level, a sequence number v is assigned. The coded symbol can then be written in the form

Fig. 6 Coding example: a) 6 symbols ($v=6; l=0$); b) 24 symbols ($v=3, l=3$); c) 35 symbols ($v=2; l=6$)

of $S = v + 0.5l(2L + 1 - l)$, where $0 \leq l < L$. The pulse width depends on parameters v and l , and the pause time between pulses does not change and equals T_W . An example of symbol encoding from level $l=5$ (first from the bottom) and a symbol from level $l=0$ (second from the bottom) is shown in Fig. 5. In this example, time T_W was equal to 0.025 s.

The encoding of parameter v is done by assigning a value of one in the code frame for bits D_1 to D_{v-1} and the value of zero for D_v . If the symbol is at level $l=0$ then the bits from D_{v+1} to D_{L+1} are assigned a value of one. Otherwise, bits from D_{v+1} to D_{v+l} are set to one and the D_{v+l+1} bit value is set to zero, which corresponds to the coding of the l parameter. The other code bits are assigned a value of one. In both cases, the stop bit always assumes a value of one and the frame ends the frame end bit of $D_E=0$. Giving a logic zero to the end of P1.2 of the microcontroller disables the laser module. An example of symbol encoding: 6, 24 and 35 is shown in Fig. 6. Taking into account the start and stop bits values (always equal to one), it can be noticed that the width of the first generated pulse is vT_W . For symbols from level $l=0$ the second pulse width is equal $(L+2-v)T_W$, and for the remaining levels lT_W . The third pulse is generated only for symbols from levels $0 < l < L$. Its width is equal to $(L+1-v-l)T_W$.

V. METHOD OF DECODING OF LASER BEAM

Decoding the symbol is carried out on the basis of analysis of the sequence of images obtained from the camera. Firstly, the function of laser module activation $u(t)$ is determined. Decoding the symbol takes place using the extended finite-state machine.

A. CALCULATION OF LASER ACTIVATION FUNCTION

Due to the use of the banded optical filter, only the red R component of the image is used for further analysis. The luminance component is analyzed when using a monochrome camera. It is represented in matrix form in calculations (R matrix). The value of the laser module activation function $u(t)$ is determined in discrete moments of the time nT_S , $n \in N^+$ based on the values of the elements of the R matrix. The pseudo code appropriate for the algorithm determining the values of $u(t)$ is presented in Fig. 7. Firstly, maximizing

```

Set (p);
while (isRunning)
    Delay (Ts);
    frame = GetLatestFrame();
    R = MaximizeContrast (frame.R);
    E = (R - Average (R)) / Maximum (R);
    if (Maximum (E) > p)
        B = Binarization (E, p);
        M = BiggestAreaImageMoment (B);
        RaiseDecoded (frame.timeStamp, 1, M10/M00,
                    M01/M00);
    else
        RaiseDecoded (frame.timeStamp, 0, 0, 0);

```

Fig. 7 Pseudo code to determine the value of the activation function

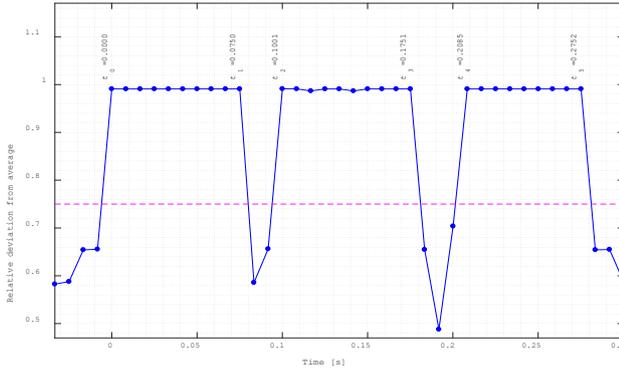
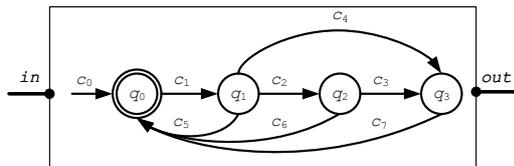


Fig. 8 Maximum value of the relative deviation from the mean in the function of time for $S = 24$ and $T_W = 0.025$ s

the contrast of the color red is performed. Then, for matrix R , the matrix of relative deviations from the mean R is determined, the values of which are determined from formula $e_{ij} = (r_{ij} - r_{avg}) / r_{max}$, where: r_{ij} – is the element of matrix R with coordinates i, j obtained by the contrast maximization operation, i, j – line and column numbers respectively of matrix R , r_{avg} – average value of matrix R (red color component after the contrast maximization operation), r_{max} – maximum value of the R component of the image obtained after the image contrast maximization operation.

In the case of experimenting with a single shooter, at least one laser spot may be observed at a time in the image. This spot is determined based on the values of matrix E . When the maximum value of an element of matrix E exceeds the value of the set threshold p , then the laser is considered to be enabled. At that time the activation function $u(t)$ assumes the value of one, or otherwise zero. The graph of the maximum relative deviation from the mean in the function of time recorded for symbol $S=24$ and the time $T_W=0.025$ s is shown in Fig. 8. The detection of the laser spot forces the determination of matrix B obtained during the binarization of the matrix E . Binarization is carried out in accordance with formula $b_{ij} = \{255, \text{for } e_{ij} > p; 0, \text{for } e_{ij} \leq p\}$, where p – the



$c_0: (-, -, -, v:=0 \wedge l:=0 \wedge T_E:=0, \phi, q_0)$
 $c_1: (q_0, in=[T, x, y], 0 < f(T) \leq L, v:=f(T) \wedge T_E:=time, \phi, q_1)$
 $c_2: (q_1, in=[T, x, y], h(T) \leq L \wedge f(time - T_E) = 1, l:=f(T) \wedge T_E:=time, \phi, q_2)$
 $c_3: (q_2, in=[T, x, y], h(T) = L + 1 \wedge f(time - T_E) = 1, T_E:=time, out:=g(), q_3)$
 $c_4: (q_1, in=[T, x, y], h(T) = L + 2 \wedge f(time - T_E) = 1, T_E:=time, out:=g(), q_3)$
 $c_5: (q_1, -, h(time - T_E) + 2 > L, v:=0 \wedge T_E:=0, \phi, q_0)$
 $c_6: (q_2, -, h(time - T_E) + 1 > L, v:=0 \wedge l:=0 \wedge T_E:=0, \phi, q_0)$
 $c_7: (q_3, -, -, v:=0 \wedge l:=0 \wedge T_E:=0, \phi, q_0)$
 where:
 $f(t) = \text{round}(t/T_W)$
 $h(t) = f(t) + v + 1$
 $g() = v + 0.51(2L + 1 - l)$

Fig. 9 Extended finite state machine

given binarization threshold. The results of binarization are areas in matrix B (spots), the largest of which determines the location of the laser pointer. It is for it image moments M_{00} , M_{01} , M_{10} and the center of gravity is calculated. In pseudo-code, Fig. 7, the *RaiseDecoded* has been proposed to be used and transferred to the function of handling: the time stamp, the value of the activation function, and the coordinates of the center of gravity of the area identified with the laser pointer.

B. EXTENDED FINITE STATE MACHINE

Decoding the symbol is performed by the extended finite-state machine *EFSM*. At the *EFSM* input matrix G with a structure of $[T, x, y]$ is given, where T – pulse width, x and y – laser spot coordinates. In the accepted solution, matrix G represents a single pulse extracted from the activation function of the laser module $u(t)$. According to the assumptions given in point IV, each symbol is encoded by several pulses (two or three pulses) depending on level l . Thus, based on state q of the *EFSM* machine and the determined G matrices, the symbols $S_i \in \{\phi, S_1, \dots, S_n, \dots, S_N\}$ are indicated, where ϕ means “no symbol”, S_1, \dots, S_N represent decoded symbols, and N specifies the allowed number of decoded symbols.

The extended finite-state machine was defined as 6-tuple $EFSM = (Q, q_0, V, I, O, C)$. The various symbols are: Q – set of *EFSM* states; q_0 – initial state; V – a set of *EFSM* variables; I – pulse matrix described in the form of G ; O – a set of possible *EFSM* symbols; A – a set of allowed transitions between *EFSM* states. The diagram of states and elements of set C are shown in Fig. 9. A set of Q states of the *EFSM* machine is defined as $\{q_0, q_1, q_2, q_3\}$. State q_1 represents the position of the symbol S defined within the coding level, and q_2 determines the level on which the symbol is located. The q_3 state is identified with turning off the laser module, and q_0 is the initial state of the *EFSM*. The set of variables V is defined as $\{l, v, T_E, time, X, Y\}$, where the variables l and v indicate the parameters of symbol S , the variable T_E defines the time of arrival at the input of matrix G , $time$ indicates the elapsed time since appearing at the input of the matrix G , and X and Y denote the coordinates of the laser spot determined for the first pulse. Set C is defined as $\{c_k: (q, in, guard, a, out, q')\}$ and means that if *EFSM* is in state q and matrix $[T, x, y]$ appears at the *in* input, then the transition to the state of q' is possible provided that the logical expression *guard* is true. Only in this case will the searched for symbol S be determined on the *out* output. The output of the machine from its initial state q_0 means putting forward hypothesis H_0 on symbol detection. In subsequent times, if the time limits are not met, the machine automatically returns to q_0 , which is equivalent to rejecting hypothesis H_0 . The assumption of hypothesis H_0 takes place only at the moment the machine goes into state q_3 . Only then will symbol S_i appear on the output, indicating its recognition. Determining the output symbol is based on

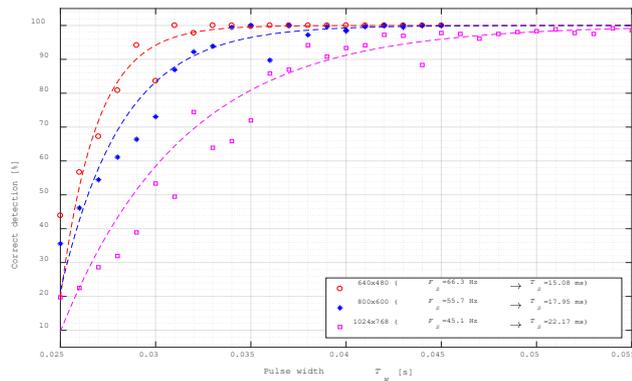


Fig. 10 Detection efficiency in the function of duration of bit T_W and tested image resolutions

variables ν and l . In other cases, the *EFSM* output is given element ϕ .

VI. RESULTS OF RESEARCH

The experiment was carried out for a test set containing $N=36$ symbols, $S \in \{S_n; n=1, \dots, N\}$. Each symbol was encoded with the developed technique (Section IV). In this way, codes were obtained, which individual bits control the laser module (“1” – *ON*, “0” – *OFF*). The test consisted of sending the test set $K=100$ times. Altogether 3,600 symbols were sent and decoded. The experiment was divided into two parts. The first part was the effect of time T_W on the efficiency of detecting S . The study was conducted for three camera resolutions: 640 px x 480 px, 800 px x 600 px and 1024 px x 768 px. In each test, the duration of T_W was changed and accepted values ranging from 0.025 s to 0.055 s with a step of 0.001 s. For each trial, the detection efficiency of the transmitted symbol was determined as well as the average detection efficiency of symbols belonging to the test set, Fig. 10. For the resolution of 640 px x 480 px, images were acquired on an average of 0.015 s ($FPS=66.66$ Hz). For T_W of 0.030 s, the detection efficiency was no worse than 90%. Symbol detection efficiency at a level of 100% was achieved when images were acquired at 0.036 s – FPS equal

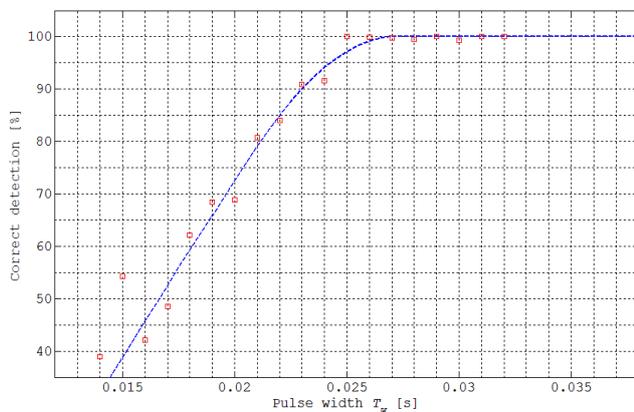


Fig. 11 Detection efficiency in the function of duration of bit T_W for resolution 640 px x 480 px and $FPS=120$ Hz

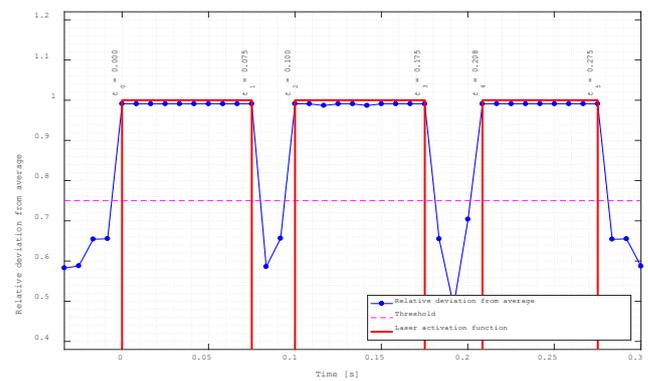


Fig. 12 Detection of symbol S_{24} , $T_W=0.025$ s, image resolution 640 px x 480 px

27.77 Hz. Comparative dependencies were obtained for resolutions: 800 px x 600 px and 1024 px x 768 px.

The second part of the study used the OptiTrack 120 Slim miniature high speed camera for motion capture. The camera was equipped with a lens with a focal length of $F=16$ mm and a bandpass filter of $\lambda_0=650$ nm and $\Delta\lambda=\pm 10$ nm. The study was conducted for a resolution of 640 px x 480 px and $FPS=120$ Hz. The second part of the experiment was divided into two phases. Firstly, it was confirmed that T_W has a similar effect on the effectiveness of the detecting symbols as was the case in the first part of the experiment. The study was performed by changing the duration of the T_W bit from 0.014 s to 0.030 s with a step of 0.001 s. In this case, for the time of T_W greater than 0.025 s, a decoding efficiency of the transmitted symbols was achieved of over 99%. The obtained results are shown in Fig. 12. In the second phase, the effectiveness of symbol recognition for T_W was 0.024 s and 0.025 s. For $T_W=0.024$ s, the efficiency of symbol recognition was no worse than 91.67%, Table I. Detailed results are shown in Table II. The lowest efficiency of 67% was achieved with symbol S_{12} . Efficiency of 100% was obtained for the symbols: $S_1, S_2, S_3, S_7, S_8, S_9, S_{14}, S_{15}, S_{16}, S_{21}, S_{24}, S_{34}, S_{36}$. No symbol has been misrecognized. In three hundred cases, the q_0 status was dropped, because no time limit was met, so no symbol was assigned. For T_W greater than 0.025 s, the detection efficiency was greater than 99%. A detailed result of recognizing symbol S_{24} is shown in Fig. 12. The laser module activation function consisted of three pulses. Parameters ν and l of the symbol were determined based on this. Parameter ν was defined based on formula $\nu = \text{round}((t_1 - t_0)/T_W)$, therefore $l=3$. Parameter l was determined from the formula $l = \text{round}((t_3 - t_2)/T_W)$,

TABLE I.

COLLECTIVE RESULTS OF DEECTING SYMBOLS FOR $TW=0.024$ S AND AN IMAGE RESOLUTION OF 640 PX X 480 PX

| | |
|---------------------------------------|--------|
| Number of symbols tested | 3 600 |
| Number of detections | 3 300 |
| Number of unrecognized symbols | 300 |
| Efficiency | 91,67% |

TABLE II.

DETAILED RESULTS OF DETECTING SELECTED SYMBOLS

| Symbol | Number of symbols in the test set | Number of correct detections | Decoding Efficiency [%] |
|----------|-----------------------------------|------------------------------|-------------------------|
| S_1 | 100 | 100 | 100 |
| S_3 | 100 | 100 | 100 |
| S_5 | 100 | 83 | 83 |
| S_7 | 100 | 100 | 100 |
| S_9 | 100 | 100 | 100 |
| S_{11} | 100 | 96 | 96 |
| S_{13} | 100 | 94 | 94 |
| S_{15} | 100 | 100 | 100 |
| S_{17} | 100 | 96 | 96 |
| S_{19} | 100 | 77 | 77 |
| S_{21} | 100 | 100 | 100 |
| S_{23} | 100 | 84 | 84 |
| S_{25} | 100 | 81 | 81 |
| S_{27} | 100 | 69 | 69 |
| S_{29} | 100 | 82 | 82 |
| S_{31} | 100 | 93 | 93 |
| S_{33} | 100 | 92 | 92 |
| S_{35} | 100 | 98 | 98 |

therefore $l=3$. The detected impulses fulfilled all time limits. On this basis, and based on the proposed coding scheme, Fig. 5, symbol S_{24} was recognized.

VII. CONCLUSION

The article presents methods of: encoding and decoding laser signals developed to control the system of augmented reality. Characteristics of the developed coding method are: constant code frame emission time, one or two intervals between pulses, tight time limits for pulse width and intervals between them. Constant time for frame emission equal to $(3+L)T_W$ to reduce the number of false detection of the transmitted symbol. Adopted strong limitations on pulse width reduce the number of type II errors, but force the use of cameras with a fixed image acquisition time. In turn, a small number of pauses between pulses allows for effective detection when symbols are emitted by multiple encoding devices.

It has been shown that using the proposed methods make it possible to distinguish between commands encoded in a laser beam with an efficiency of 99.9%. In the case of using cameras with similar parameters to the units used, the duration of the T_W bit should be no less than $3T_S$, where T_S represent the interval between the two acquired image frames.

The laser signal encoding algorithm can be successfully implemented in the embedded system. The implementation of the encoding algorithm required 57.1 bytes of data memory (byte and bit addressing memory) and 3,664 bytes of program memory.

REFERENCES

- [1] D. R. Olsen, T. Nielsen, "Laser pointer interaction", in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 17 – 22, 2001. <https://doi.org/10.1145/365024.365030>.
- [2] F. Vogt, J. Wong, S. Fels, D. Cavens, "Tracking Multiple Laser Pointers for Large Screen Interaction", *Extended Abstracts of ACM UIST 2003*, pp. 95 – 96, 2003.
- [3] A. Soetedjo, E. Nurcahyo, "Developing of Low Cost Vision-Based Shooting Range Simulator", in *IJCSNS International Journal of Computer Science and Network Security*, vol. 11, no. 2, 2011.
- [4] H. Ebrahimpour-Komleh, M. Tekiyehband, "Design of an interactive whiteboard system using computer vision techniques", *Proceedings of 6th International Symposium on Mechatronics and its Applications 2009(ISMA '09)*, pp. 423 - 26, 2009.
- [5] J.-t. Wang, C.-N. Shyi, T.-W. Hou, C. P. Fong, "Design and Implementation of Augmented Reality System Collaborating with QR Code", *Computer Symposium (ICS), 2010 International*, pp. 414 – 418, 2010. <https://doi.org/10.1109/COMPSYM.2010.5685477>.
- [6] H. Ukida, S. Kaji, Y. Tanimoto, H. Yamamoto, "Human Motion Capture System Using Color Markers and Silhouette", *Instrumentation and Measurement Technology Conference, IMTC 2006*, proceedings of the IEEE, pp. 151 – 156, 2006. <https://doi.org/10.1109/IMTC.2006.328334>.
- [7] A. Smeragliuolo, N. Hillc, L. Dislad, D. Putrino, "Validation of the Leap Motion Controller using marked motion capture technology", *Journal of Biomechanics*, vol. 49, 9, pp. 1742 – 1750, 2016. <http://doi.org/10.1016/j.jbiomech.2016.04.006>.
- [8] K. Barczewska, A. Drozd, "Comparison of methods for hand gesture recognition based on Dynamic Time Warping algorithm" in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, pp. 207 – 210, 2013.
- [9] J. Lebień, M. Szwoch, "Virtual Sightseeing in Immersive 3D Visualization Lab", in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS*, vol. 8, pp. 1641–1645, 2016. <http://dx.doi.org/10.15439/2016F227>.
- [10] T. Bothe, A. Gesierich, W. Li, C. Kopylow, N. Kopp, W. Juptner, "3D-Camera for Scene Capturing and Augmented Reality Applications", *3DTV Conference*, pp. 1 – 4, 2007. <https://doi.org/10.1109/3DTV.2007.4379469>.
- [11] T. Palys, W. Żorski, "Enhanced movement tracking with Kinect supported by high-precision sensors", in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS*, vol. 5, pp. 883 – 888, 2015. <http://dx.doi.org/10.15439/2015F166>.
- [12] K. Murawski, A. Arciuch, T. Pastelny, "Studying the influence of object size on the range of distance measurement in the new Depth From Defocus method", in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS*, vol. 8, pp. 817–822, 2016. doi: 10.15439/2016F136.
- [13] N. W. Kim, H. Lee, "Developing of vision-based virtual combat simulator", in *Proceedings of International Conference on IT Convergence and Security (ICITCS)*, Macao, China, pp. 1 – 4, 2013.
- [14] GKDesign Engineering: "RS-232 Laser Transceiver", *Electronics Australia*, pp. 56 – 61, 1997.
- [15] K. Murawski, R. Różycki, P. Murawski, A. Matyja, M. Rekas, "An Infrared Sensor for Monitoring Meibomian Gland Dysfunction", in *Acta Physica Polonica A*, vol. 124, no 3, pp. 517 – 520, 2013. doi: 10.12693/APhysPolA.124.517.
- [16] K. Murawski, "Measurement of membrane displacement with a motionless camera equipped with a fixed focus lens", *Metrology and Measurement Systems*, vol. 22, no. 1, pp. 69-78, 2015. doi.: 10.1515/mms-2015-0011
- [17] K. Murawski, "New Vision Sensor to Measure and Monitor Gas Pressure", *Acta Physica Polonica A*, vol. 128, no 1, pp. 6 – 9, 2015. doi: 10.12693/APhysPolA.128.6
- [18] K. Murawski, "Measurement of membrane displacement using a motionless camera", *Acta Physica Polonica A*, vol. 128, no 1, pp. 10 – 14, 2015. doi: 10.12693/APhysPolA.128.10