View Selection for Virtual View Synthesis in Free Navigation Systems

Adrian Dziembowski
Chair of Multimedia Telecommunications and Microelectronics
Poznan University of Technology
Poznań, Poland
adziembowski@multimedia.edu.pl

Jarosław Samelak
Chair of Multimedia Telecommunications and Microelectronics
Poznan University of Technology
Poznań, Poland
jsamelak@multimedia.edu.pl

Marek Domański
Chair of Multimedia Telecommunications and Microelectronics
Poznan University of Technology
Poznań, Poland
marek.domanski@put.poznan.pl

***Abstract*—The paper deals with the problem of the optimal selection of the real views used for the virtual view synthesis in the free navigation systems. The practical free navigation system requires the real-time view synthesis. Therefore, the virtual view cannot be synthesized using the information from all the real cameras, but only few, most proper ones. In the paper the method of fast selection of two proper real views is presented. The proposed approach allows to select the real views for different camera arrangements, thus can be applied to the practical free navigation systems.**

***Keywords*—free navigation, free-viewpoint television, virtual view synthesis**

I. Introduction

The free navigation (Free Viewpoint Television, FTV) systems [1], [2], [3] allow the user to freely, virtually navigate within a scene captured by a number of cameras [4], [5], [6]. In order to provide smooth movement between different viewpoints, the virtual viewpoints have to be synthesized [7] using the information from the real views, corresponding depth maps [8], [9] and camera parameters [10].

The paper deals with the problem of the optimal selection of two real views used for the synthesis of the virtual view. This problem is usually skipped in the literature. For example, in the state-of-the-art virtual view synthesis method – VSRS (View Synthesis Reference Software) [11] the virtual view is synthesized from two manually selected real views.

Of course, the synthesis from the optimal pair of the real views should result in the virtual view with as little non-synthesized areas as possible. In the most straightforward approach, an algorithm could perform the synthesis using all the pairs of the real views and check the percentage of area directly projected from the real views. However, such an approach requires processing of many real views, thus cannot be used in the practical, real-time free navigation system [12].

II. Motivation

In order to allow a user to freely navigate around the acquired scene, a view synthesis algorithm should work in the real-time. Adding more real views for virtual view synthesis [13], [14], [15] increases the quality of synthesized views but also increases the computational time, which is critical in the practical free navigation systems [3], [12]. Therefore, in the real-time systems view synthesis based on only two real views is preferable.

This paper deals with the problem of choosing the best two real views in order to achieve the highest quality of the synthesized virtual views. Obviously, presented approach can be also used in the multiview synthesis, MVS [13], where large part of the virtual view is synthesized using information from two real views and the remaining views are used only for the disocclusion filling.

III. View Synthesis Problems

In the multicamera systems three main view synthesis problems can be distinguished. These problems are described in subsections A, B and C. All of them reduce the quality of the synthesized virtual view.

A. Common Scene Area

The first problem in virtual view synthesis (especially for the sparse multicamera systems, where the cameras are located arbitrarily) is the area of the virtual view that can be directly projected from the real views. Some parts of the virtual view cannot be projected because they were occluded by the foreground objects in the real views. The problem is the bigger, the greater is the distance between the virtual camera and real ones.

B. Finite Image Resolution

The second problem is the finite resolution of the real views. Let us assume a simple, exemplary scene, where four real cameras acquire one simple object (Fig. 1).

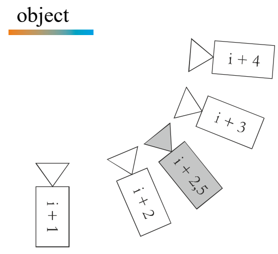

Fig. 1. Finite image resolution problem: a simple scene.

The cameras see the object from different angles, so in the different real views the width of the object would vary (the top

The presented work has been funded by the Polish Ministry of Science and Higher Education for the status activity consisting of research and development and associated tasks supporting development of young scientists and doctoral students in 2018 in Chair of Multimedia Telecommunications and Microelectronics

row of Fig. 2). In the view (i + 4), the object is noticeably lower than in the view (i + 1) and, more importantly, in the virtual view (i + 2.5). Therefore, while projecting the points from the view (i + 4) to the virtual view (i + 2.5), the discontinuities in the object's surface would appear (last column of Fig. 2).

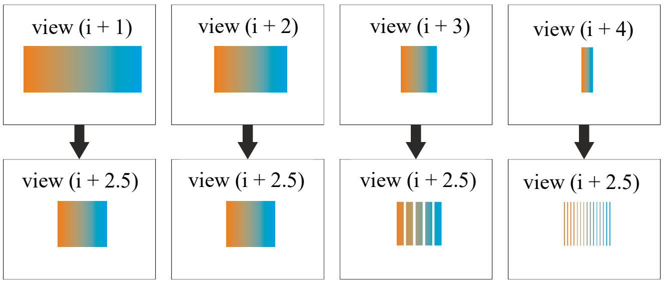

Fig. 2. Finite image resolution problem: projection of the same object from different real views into one virtual view.

According to the presented considerations, in order to preserve objects' continuity, the objects in the scene should be projected from these real views, where the object has the highest width. In the example presented in Fig. 2, the best quality would be acquired when using views (i + 1) and (i + 2).

However, there is a third main problem in the virtual view synthesis, described in the following subsection – non-lambertian surfaces of the objects in the scene.

C. *Non-Lambertian Reflectances*

The real, natural objects placed in the acquired scene have the non-lambertian surfaces, which means they reflect the light differently in different angles – depending on the direction of the light fall. This phenomenon is schematically presented in Fig. 3. The light source is symbolized as a sun. The luminance of the light reflected in the different directions is proportional to the length of the arrows.

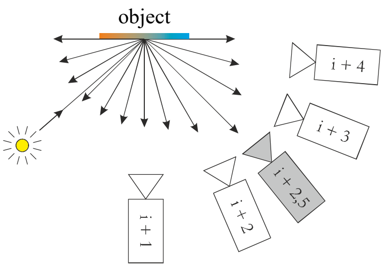

Fig. 3. Non-lambertian reflectances problem: a simple scene.

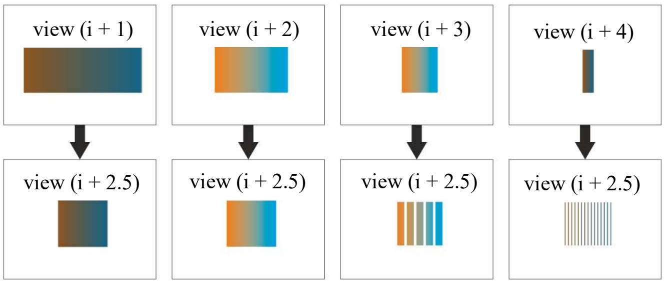

Fig. 4. Non-lambertian reflectances problem: projection of the same object from different real views into one virtual view.

The views captured by the cameras are presented in Fig. 4. As it is shown, the same object captured by different cameras has different brightness. For the views (i + 1) and (i + 4) it is visibly darker, than for views (i + 2) and (i + 3). Of course, presented example is exaggerated, but such an effect can be observed in the views captured by the real multicamera systems.

According to the Fig. 3, the object seen by the virtual camera (i + 2.5) should be even brighter than in the real views (i + 2) and (i + 3). However, the synthesis using these views would provide higher quality than for the further real views.

D. *Conclusion*

Presented considerations imply, that the best quality of the synthesized virtual view can be obtained when using two neighboring real views – the nearest left and the nearest right real view. In Section IV we presented the results of the experiment that proves that assumption.

IV. Quality of the Virtual View Synthesized using Different Real Views

A. *Experiment*

In order to examine, which real views should be chosen, the testset containing 12 test sequences was used (Table I). The testset contains diverse sequences (different resolutions, linear and non-linear camera arrangements with cameras arranged uniformly and in the stereopairs, synthetic and natural content, different scene characteristics, etc.).

TABLE I. Test Sequences

Test sequence	**Source**
Big Buck Bunny Butterfly	Holografika [16]
Big Buck Bunny Flowers	Holografika [16]
Poznan_Blocks	Poznan Univ. of Technology [17]
Poznan_Blocks2	Poznan Univ. of Technology [18]
Poznan_Fencing2	Poznan Univ. of Technology [18]
Poznan_Service2	Poznan Univ. of Technology [18]
Ballet	Microsoft Research [19]
Breakdancers	Microsoft Research [19]
Soccer Arc	Hasselt University [20]
Soccer Linear	Hasselt Univeristy [20]
Poznan_Carpark	Poznan Univ. of Technology [21]
Poznan_Street	Poznan Univ. of Technology [21]

For all of the test sequences we performed the same experiment: we synthesized the virtual view in the position of the middle real view (view 45 for both *Big Buck Bunny* sequences and view 4 for all the other sequences). The virtual view synthesis was performed using MVS algorithm [13].

The virtual view was synthesized using all the combinations of two real views (obviously, excluding the view at the position of the virtual one). For example, for *Poznan_Street* sequence the virtual view 4 was synthesized using the following pairs of the real views: 1-1, 1-2, 1-3, 1-5, 1-6, 1-7, 2-2, 2-3, 2-5, 2-6, 2-7, 3-3, 3-5, 3-6, 3-7, 5-5, 5-6, 5-7, 6-6, 6-7 and 7-7, where the notation 1-2 means, that the virtual view was synthesized using real views 1 and 2. Used synthesis algorithm is not sensitive to the order of used real views, therefore the view synthesized using views 1-2 would be the same as for the real views 2-1.

Obviously, considering mentioned in the subsection A smaller common scene area for further real views, the results of the PSNR estimation would be very predictable – the greater is the distance between the virtual camera and the real one, the lower is the quality of the synthesized view. Therefore, in order to present the influence of two problems described in the subsections B and C, the PSNR value was calculated only for the areas directly projected from the real views (the inpainted points of the virtual view were omitted in the PSNR evaluation).

B. Experimental Results

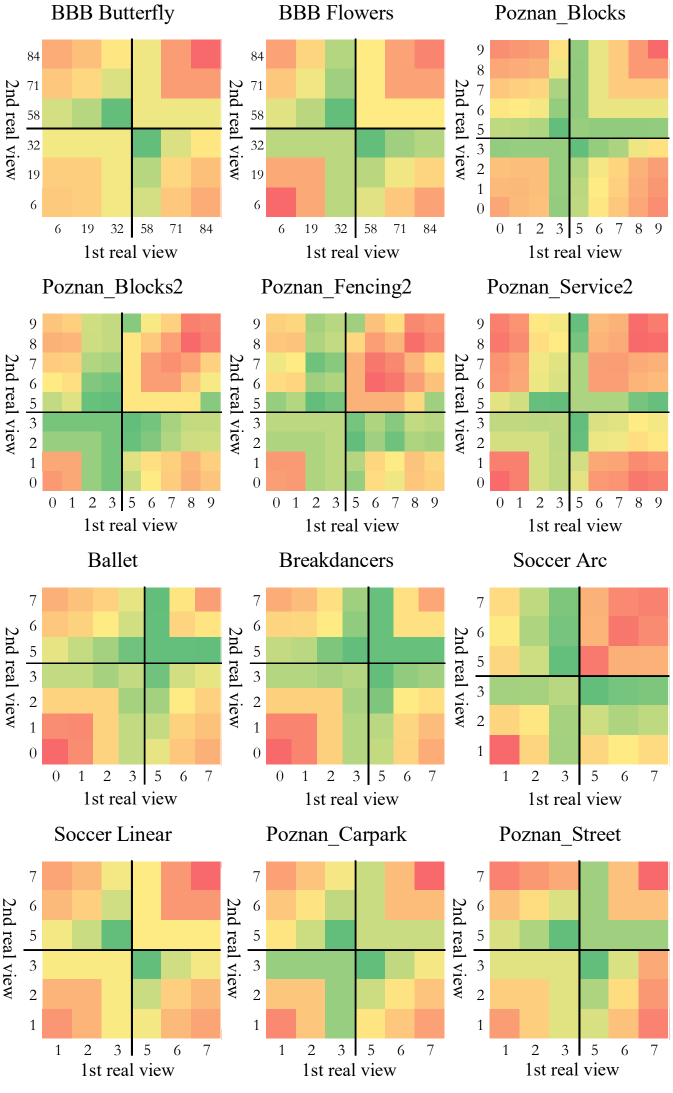

Fig. 5. The quality of the virtual views synthesized from different pairs of real views; PSNR value was estimated **only for areas directly projected** from the real views (inpainted areas were omitted); for each test sequence on the horizontal axis there is a first real view's index, on the vertical axis – index of the second real view.

In Fig. 5 the experimental results are shown. For each sequence, on the horizontal axis the index of the first real view used for view synthesis is presented, on the vertical axis – the index of the second real view. The virtual view position is marked with black solid line (both horizontally and vertically).

Green points mean higher quality of the synthesized view, the red ones – lower PSNR value. Because of the very different quality of views synthesized for different test sequences (e.g. 22 dB for *Soccer Arc* and 35 dB for *Poznan Street*), the scale was set separately for each test sequence.

Presented results indicate, that the quality of the virtual view synthesized using the nearest left and the nearest right real view is the highest (green points for pairs 32-58 and 58-32 for *Big Buck Bunny* sequences and pairs 3-5 and 5-3 for the other ones).

Of course, for some sequences (i.e. *Ballet* and *Poznan_Service2*) usage of different real views provides similar results, but – according to the subsection A – the common area is smaller for further real views, so the selection of the nearest real views would provide the best quality.

C. Conclusion

Presented considerations and experimental results imply, that the best quality of the synthesized virtual view can be obtained when using two neighboring real views – the nearest left and the nearest right real view.

V. NEIGHBORING REAL VIEWS CHOSING

In the previous section it was stated, that the virtual view should be synthesized from the left and right neighboring real views. However, the information about the proximity between virtual and real views is not directly given and it should be extracted from the camera parameters [10]. This section deals with the problem, how to decide, which views are the nearest ones.

The simplest way for neighboring views selecting would be the calculation of an Euclidean distance between the virtual camera and all the real cameras:

$$d(i,V) = \sqrt{\left(t_{x,i} - t_{x,V}\right)^2 + \left(t_{y,i} - t_{y,V}\right)^2 + \left(t_{z,i} - t_{z,V}\right)^2}, \quad (1)$$

where $t_{x,V}$, $t_{y,V}$ and $t_{z,V}$ are the elements of the virtual camera's translation vector $\mathbf{T_V}$ and $t_{x,i}$, $t_{y,i}$ and $t_{z,i}$ define the translation $\mathbf{T_i}$ of the i-th real camera. Then, two cameras with smallest $d(i,V)$ would be chosen.

Such an approach could be used for linearly or non-linearly arranged cameras, but only for systems with cameras distributed evenly (Fig. 6).

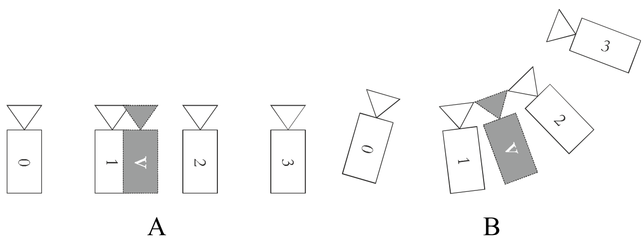

Fig. 6. Multicamera systems with evenly distributed cameras.

For arbitrarily located cameras (e.g. for cameras arranged in the stereo pairs [18], [22]) such an approach can lead to wrong view selection. In the example presented in Fig. 7A, views chosen using (1) would be view 0 and 1 – two real views located at the same side of the virtual view.

In order to properly select view 1 and 2, the triangle inequality equation can be used. In that approach, one nearest real view is searched using (1), but the second one should also satisfy the following condition:

$$d(j,V) \leq d(j,i). \tag{2}$$

The above condition requires the distance between virtual camera V and real camera j to be smaller than the distance between both the real cameras i and j.

This approach allows to properly choose the neighboring left and right view regardless of the real cameras arrangement. However, in the free navigation systems a user should also be able to step into the scene (Fig. 7B).

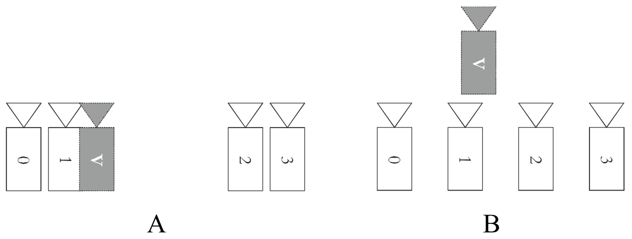

Fig. 7. Multicamera system with cameras arranged in stereo pairs (A) and virtual navigation "into the scene" (B).

In this scenario, presented view selection method would be insufficient: views 1 and 2 should be chosen. The nearest view – 1 would be selected properly but $d(2,V) > d(2,1)$, so not a view 2 but view 3 would be selected as the second neighbor.

Assuming disadvantages of presented methods we propose the method allowing proper view selection for any camera arrangement. In the proposed approach the view selection is processed in the transformed coordinate system (Fig. 8), with the origin in the center of the virtual camera. Its axes X' and Y' create the plane of camera converter and Z' is parallel to the optical axis of the virtual camera.

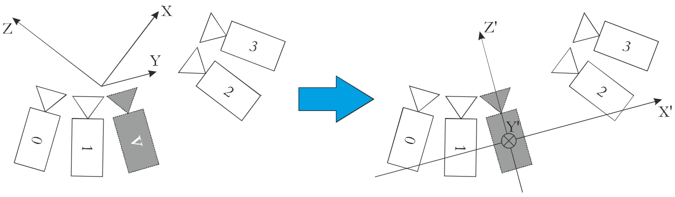

Fig. 8. Transformation of the global coordinates system into the virtual camera's coordinate system.

The position (translation vector) of the real cameras in the transformed coordinate system are calculated as:

$$\mathbf{T}'_{\mathbf{i}} = \mathbf{R}_{\mathbf{V}} \cdot (\mathbf{T}_{\mathbf{i}} - \mathbf{T}_{\mathbf{V}}), \tag{3}$$

where $\mathbf{R}_{\mathbf{V}}$ is the rotation matrix of the virtual camera, $\mathbf{T}_{\mathbf{i}}$ – translation vector of the i-th real camera in the global coordinate system and $\mathbf{T}_{\mathbf{V}}$ is the translation vector of the virtual camera in the global coordinate system.

Then, separately for cameras with positive and negative t_x value, the Euclidean distance between the camera and the origin of the modified coordinate system is calculated.

The left nearest view is the one with the smallest Euclidean distance among the views with negative value of t_x; the right neighboring view – the view with the smallest distance among the views with t_x greater than 0.

Presented approach results in the best achievable quality of the synthesized views, therefore it was used for real view selection in the view synthesis algorithms [13], [23], allowing to properly choose the real views in the practical free-viewpoint television system [3], [12].

VI. Conclusions

The quality of the virtual view synthesized using different pairs of the real views was discussed. Performed experiments prove, that in order to achieve the best possible quality of the virtual view, the nearest left and the nearest right view should be used for the view synthesis.

In the paper an efficient method of selecting optimal real views is presented. It allows for proper selection for the real views independently on camera arrangements. The proposed method does not require projecting points from all the real views in order to choose the best real views. Therefore, the presented approach is fast, thereby it can be applied in the practical free navigation systems.

References

[1] M. Tanimoto, M.P. Tehrani, T. Fujii, T. Yendo, "FTV for 3-D spatial communication," Proc. IEEE, vol. 100, no. 4, pp. 905-917, 2012.

[2] C.C. Lee, A. Tabatabai, K. Tashiro, "Free viewpoint video (FVV) survey and future research direction," APSIPA Tr. on Signal and Information Processing, vol. 4, 10.2015.

[3] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, K. Klimaszewski, D. Mieloch, R. Ratajczak, O. Stankiewicz, J. Siast, J. Stankowski, K. Wegner, "Demonstration of a simple free viewpoint television system," in 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017, pp. 4589-4591.

[4] P. Goorts, M. Dumont, S. Rogmans, P. Bekaert, "An end-to-end system for free viewpoint video for smooth camera transitions," IC3D 2012, Liege, Belgium, 3-5.12.2012.

[5] M. Tanimoto, "Overview of FTV (free-viewpoint television)," ICME 2009, New York, USA, 28.06-3.07.2009.

[6] M. Domański, M. Bartkowiak, A. Dziembowski, T. Grajek, A. Grzelka, A. Łuczak, D. Mieloch, J. Samelak, O. Stankiewicz, J. Stankowski, K. Wegner, "New results in free-viewpoint television systems for horizontal virtual navigation," in 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, 2016, pp. 1-6.

[7] M. Domański, M. Gotfryd, K. Wegner, "View synthesis for multiview video transmission," IPCV 2009, USA, 2009.

[8] S. Zinger, L. Do, P.H.N. de With, "Free-viewpoint depth image based rendering," J. Vis Comm. & Im. Repr., vol. 21, pp. 533-541, 2010.
[9] K. Muller, P. Merkle and T. Wiegand, "3-D Video Representation Using Depth Maps," Proceedings of the IEEE, vol. 99, no. 4, pp. 643–656, April 2011.
[10] A. Heyden, M. Pollefeys, "Multiple view geometry," in: Emerging Topics in Comp. Vis., Prentice Hall, 63-75, 2008.
[11] T. Senoh, K. Yamamoto, N. Tetsutani, H. Yasuda, K. Wegner, "View Synthesis Reference Software (VSRS) 4.2 with improved inpainting and hole filling," ISO/IEC JTC1/SC29/WG11 MPEG2013, M40657, Hobart, Australia, 3-7.04.2017.
[12] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, D. Mieloch, R. Ratajczak, J. Samelak, O. Stankiewicz, J. Stankowski, K. Wegner, "Free-viewpoint television demonstration for sports events," ISO/IEC JTC1/SC29/WG11, Doc. MPEG M41994, Gwangju, Korea, 22-26 January 2018.
[13] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domański, "Multiview Synthesis – improved view synthesis for virtual navigation," 32nd Picture Coding Symposium, PCS 2016, Nuremberg, Germany, 4-7.12.2016.
[14] B. Ceulemans, S.P. Lu, G. Lafruit, A. Munteanu, "Robust multiview synthesis for wide-baseline camera arrays," IEEE Transactions on Multimedia, early access, 2018.
[15] S. Li, C. Zhu, M.T. Sun, "Hole filling with multiple reference views in DIBR view synthesis," IEEE Transactions on Multimedia, early access, 2018.
[16] P.T. Kovacs, A. Fekete, K. Lackner, V.K. Adhikarla, A. Zare, T. Balogh, "[FTV AHG] Big Buck Bunny light-field test sequences," ISO/IEC JTC1/SC29/WG11 MPEG2013, M36500, Warsaw, Poland, 22-26.06.2015
[17] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, "Poznan Blocks - a multiview video test sequence and camera parameters for free viewpoint television," ISO/IEC JTC1/SC29/WG11 MPEG2014, M32243, San Jose, United States, 13-17.01.2014.
[18] M. Domański, A. Dziembowski, A. Grzelka, A. Łuczak, D. Mieloch, O. Stankiewicz, K. Wegner, "Multiview test video sequences for free navigation exploration obtained using pairs of cameras," ISO/IEC JTC1/SC29/WG11 MPEG2016, M38247, Geneva, Switzerland, 30.05-3.06.2016.
[19] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, "High-quality video view interpolation using a layered representation," ACM TOG, vol. 3, s. 600-608, 08.2004.
[20] P. Goorts, "Real-time adaptive plane sweeping for free viewpoint navigation in soccer scenes," doctoral dissertation, Hasselt University, Belgium, pp. 175-186, 16.06.2014.
[21] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner, "Poznań multiview video test sequences and camera parameters," ISO/IEC JTC1/SC29/WG11 MPEG2009, M17050, Xi'an, China, 26-30.10.2009.
[22] O. Stankiewicz, M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, J. Samelak, "A free-viewpoint television system for horizontal virtual navigation," IEEE Tr. on Mult, 2018.
[23] A. Dziembowski, J. Stankowski, "Real-time CPU-based virtual view synthesis," submitted to International Conference on Signals and Electronic Systems, ICSES 2018, 10-12 September 2018, Kraków, Poland.