

Pseudo-holographic Displays as Teaching Tools in Mathematics

Monica Ciobanu

Department of Engineering and Computer Science
Vasile Goldis Western University of Arad,
Bulevardul Revoluției 94, Arad, Romania
Phone: 0257 211 884
cmmciobanu@gmail.com

Antoanela Naaji

Department of Engineering and Computer Science
Vasile Goldis Western University of Arad,
Bulevardul Revoluției 94, Arad, Romania
Phone: 0257 211 884
anaaji@uvvg.ro

Ioan Dascal

SC Antaco Romania SRL, Arad, Romania
Bd. Iuliu Maniu, Nr. 31-33, Sc. B Et. 2 Ap. 10, Arad, Jud. Arad
nelu_dascal1@yahoo.com

Ioan Virag

Automation and Computer Science Faculty
Politehnica University of Timisoara,
Piața Victoriei 2, Timișoara 300006, Romania
Phone: 0256 403 000
ioan.virag@aut.upt.ro

Abstract

Due to the fact that the abstract mathematical notions are often hard to grasp by students, without linking them to a specific physical representation or correlate them with regular facts, our team came up with the idea to create and implement a set of interactive teaching tool packages (ITTPs) to help them overcome such issues. The ITTPs consist of theoretical modules and applications in the framework of a Hungarian - Romanian cross-border project, developed with the aid of existing open source software and technologies. The focus of the Romanian team was on basic elements of linear algebra and analytic geometry, showcased using an auto-stereoscopic display for the purpose of image visualization. This paper presents an overview of similar technologies and the applications developed for representing pseudo-holographic images of geometric objects, namely vectors, straight lines, planes and conics. One of the developed packages was written using web technologies in order to make the applications available on mobile devices, as well as to eliminate the need to install any additional software.

Keywords: Autostereoscopic display, pseudo-holographic images, teaching tools

1. Introduction

The trends in modern education lead to the development of online courses due to their benefits especially in terms of flexibility, knowledge achievement and costs. Getting education by integrating ICT in the learning process motivates learners and offers skills that sometimes lack resources in traditional education. Also, the online environment is changing continuously representing a great opportunity for learning, particularly for individuals who want to advance their career.

In the last years there has been an increasing development of the massive open online courses (MOOC) (Provision Company webpage, 2016), materialized in the form of multiple

platforms, offered to large numbers of students, using technologically enhanced resources and activities, such as virtual laboratories, simulations and cognitive tutors.

Although many people still consider traditional education better in achieving knowledge, online learning proves to be a great alternative for the study many fields and to boost the level of self-motivation. In this way, students have the chance to study at their own pace and in their own time.

In this context, considering the potential and the benefits of e-learning platforms, we created and implemented some courses for mathematical disciplines, adapted to online education (Ammeraal, Zhang, 2007).

The courses were developed as interactive teaching tools packages (ITTPs), containing both theoretical modules and applications (Andrica, Topan, 2004; Atanasiu, Munteanu, Postolache, 2003). The topics related to ITTPs are: mathematical analysis, probability, numerical calculus, computational algebra, linear algebra and analytical geometry. The corresponding applications were made using Maxima or were created using Java (for Android), Java 3D and JavaScript. All of the developed applications are open-source.

These results were obtained in the framework of a Hungarian - Romanian cross-border project (see Acknowledgement).

The aim of our project was to facilitate the understanding of mathematical abstract notions by combining theory with interactive applications. The ITTPs provide students with a new tool in order to make the content more accessible and help them understand and learn mathematical notions (Ciobanu, Naaji, Virag, Dascal, 2016).

For some applications, an autostereoscopic display was used for visualizing 3D objects as pseudo-holographic images, as well as the outcome of parameter changes inside mathematical formulas. For each application we developed a version which can be visualized on a classical 2D display and another one adapted for the 3D pseudo-holographic device (Ciobanu, Ploscar, Dascal, Virag, Naaji, 2015). This new approach (of using a 3D autostereoscopic display for education) differentiates our tools from other existing contributions.

2. Technical Background and Related Work

The proper rendering of three-dimensional (3D) content cannot be achieved using traditional two-dimensional (2D) displays. This lack of the ability to properly display cues related to the depth of the rendered 3D objects makes it very hard for students to grasp the proper spatial representation of the discussed subjects.

The cues needed to understand three-dimensional content displayed on a two-dimensional display are emulated with the help of mathematical algorithms, but the resulting images are still missing some essential pieces that could be properly represented only by using a 3D display device.

According to (Geng, 2013) there are four major physical (physiological) and five psychological depth cues the human brain uses to gain a 3D sensation based on the viewed images. The physical depth cues are:

- *Accommodation* – the measurement of tension in the muscle that changes the focal length of the eyes' lenses in order to bring into focus a specific 3D object in the scene, allowing us to perceive its 3D depth.
- *Convergence* – the difference between the viewing directions of the eyes when looking at the same spot in a 3D scene.
- *Motion parallax* – nearby objects appear to move faster in a 3D scene than those that are far away.
- *Binocular disparity* – refers to the fact that our left and right eyes see slightly different perspectives of the same 3D object. Thus it brings into focus objects at different distances.

Also according to (Geng, 2013) the human visual system can interpret depth by extracting psychological cues from 2D monocular images. These psychological depth cues are:

- *Linear perspective* – the appearance that parallel lines converge at a distant vanishing point on the horizon. This technique was often used by Renaissance painters to depict buildings and scenes that appeared more realistic than those painted in the Middle Ages.
- *Occlusion* – the ability of the human visual system to perceive an overlapped 3D object as laying farther away than the object in front of it.
- *Shading* – when objects are casting shadows upon another object in the 3D scene, this visual cue helps us determine which object is closer to the viewer. It also helps determine the spatial orientation of the object.
- *Texture* – objects with smooth textures are usually interpreted as being farther away since we cannot distinguish too many surface details.
- *Prior knowledge* – 3D shapes and sizes of common geometrical structures can help the viewer infer depth cues due to the fact that the brain already knows how they interact with the surrounding light sources.

Depending on their construction principle, some of the actual 3D displays are able to provide all these depth cues, while others provide only some of them.

The current 3D display technologies fall in two major categories: *stereoscopic displays* (when special eyewear is used to induce the 3D depth sensation) and *autostereoscopic displays* where different techniques provide a sense of 3D depth without the use of any kind of 3D glasses.

Autostereoscopic displays, according to (Geng, 2013), fall into three major construction types: multiview 3D displays, volumetric 3D displays, and holographic displays, although the latest are more commonly referred as pseudo-holographic displays since they do not use real holograms.

The multiview 3D displays usually provide a finite number of different perspectives to a limited number of users by using parallax barriers (Gocheva, Rahnev, 2008; Iltina, Iltins, 2013), or lenticular lenses (Iordan, 2009; Jeon, Park, Gwag, Lee, Kwon, 2014) that block the view of a specific set of pixels for the other users. These displays usually provide a lower resolution because, for a display with N views (perspectives), the actual resolution of each view is $1/N$ of the initial display resolution.

Volumetric 3D displays use individual voxels that provide accurate volumetric information regarding the rendered 3D objects. The major enhancement of these displays is that they provide a 360-degree view of the displayed model, and not only one single perspective for each viewer as the multiview 3D displays. By their construction principle, the volumetric 3D displays fall into two major categories: Static Screen Volumetric 3D Displays as in Figure 1 that have no moving parts and Swept Screen Volumetric 3D Displays as in Figure 2 that usually involves a rotating screen and relies on the persistence of vision.



Figure 1. 3D LED RGB Cube

source: <https://www.youtube.com/watch?v=Pswc-hIvRrM>



Figure 2. *Perspecta volumetric display*

source: http://www.3dcgi.com/cooltech/displays/ActualityMilitary3D_sm.jpg

Pseudo-holographic displays are capable of displaying individual hogels (holographic picture elements) (Kim, Yoon, Yoon, Ju, 2015). Their major advantage is that the displayed 3D image seems to “float away” from the display providing the viewer with an experience close to an actual hologram. Depending on their construction principle, these devices provide a single or several different perspectives of the 3D object to multiple users at the same time.

During project development our team used a HoloVision pseudo-holographic display manufactured by Provision (Lanman, Hirsch, Kim, Raskar, 2010). This autostereoscopic 3D display allows several students to visualize the projected 3D models at the same time without wearing any kind of special glasses.

The HoloVision device is capable of creating the illusion of a 30 cm image focused at approximately 90 cm in front of the monitor, viewable at a 60-degree angle from the center.

The HoloVision device has an in-built minicomputer that allows the storing of 3D content that later can be played (with a pre-installed media player) if the 3D display is used for commercial purposes. Since our team created the teaching tools to be interactive, we could not use the device’s player and connected our laptop to the HoloVision’s VGA port, using it in a similar fashion as a 3D projector. The laptop our team used had a 17.3 inch display, an Intel Core i3 processor at 2.2 GHz, 4 GB DDR 3 memory and an integrated HD 3000 graphics processor that provided enough processing resources to render the interactive 3D content.

3. The Applications Developed for Interactive Teaching Tools

In this paragraph we present some of the applications developed based on Java 3D API and JavaScript (Lucente, 1994; Mustea, Naaji, Herman, 2014). They are related to topics from linear algebra and analytical geometry (Osmani, 2014; Xie, Zhao, Yang, et al. 2012), namely vectors, straight lines, planes, and conics. The software was developed in two versions: one for 2D monitors, suitable for individual study, and the other one for autostereoscopic display. In the first version, the text fields and tags are created to input data, positioned in the upper side of the monitor, with buttons for calculations and 3D representation of the images. Likewise, results of calculations and details regarding geometric objects are displayed in the lower side of the monitor (Figure 3).

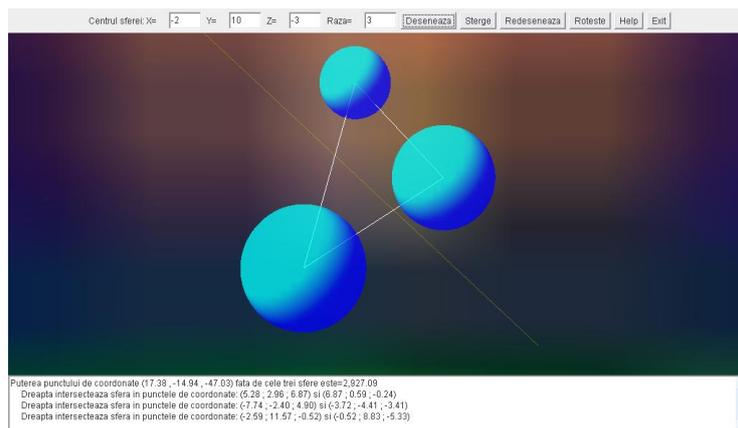


Figure 3. Interface developed for 2D monitor

In the second version for the autostereoscopic device (Figure 4) the area for data input displayed at the beginning is hidden when 3D objects are represented, thus allowing emphasis of the pseudo-holographic image. Two types of interfaces were needed because the 3D image cannot be rendered in the same way on a 2D monitor as on the HoloVision device, where the 3D model keeps its details even if it is scaled.

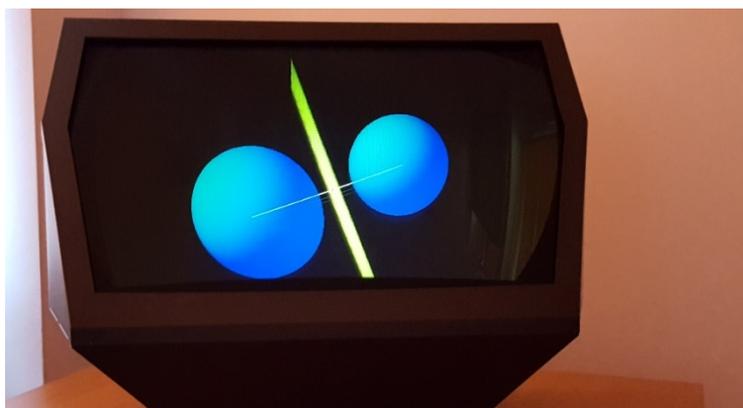


Figure 4. Interface developed for holographic display

3.1. The applications developed for representing vectors and elements of vector calculus

The Vector Spaces ITTP comprises vector spaces, Euclidean vector spaces, and free vectors, emphasizing connections between linear algebra and analytical geometry. Isomorphism between a vector space and a real space of the same dimension leads to a vector being expressed with the help of its coordinates relative to a basis. The notions of scalar product of two vectors, vector norm, and angle between two vectors are introduced, with the aim of constructing an orthonormal vector system. Definitions have a general character, being valid in an n -dimensional space, but for the applications to have an intuitive support, spaces of dimensions 2 and 3 are considered to represent geometric objects. Bounded vectors (oriented segments), equivalence relation on the set of bounded vectors and free vectors (equivalence classes) are defined and elements of vector calculation are presented.

One of the applications created allows establishing whether a vector system forms a basis or not, determining a vector's coordinates in relation to a base.

Another application represents 3D vectors obtained by multiplication with a scalar, vector addition and subtraction, vector product. In the version for individual study, on a 2D monitor (Figure 5), data are introduced in text fields found in the upper side of the monitor. Tags indicate 3D coordinates of vector extremities. Data input for several vectors and their representation is made

by successively clicking the *Draw* button. The *Delete* button allows deleting the last drawn vector, thus making it possible to successively reduce the number of vectors. The *Add* button comprises a submenu which allows selecting the vector addition method, i.e. parallelogram method or triangle method. The Subtract button works in a similar way. The *Multiply* button allows calculating the product between a vector and a scalar and the vector product. Displayed images can be rotated around the Oy axis to better view the result of the performed operation. In the lower side of the monitor there is a text area where the coordinates of the resulting vector are shown.

This application was created using Java 3D API.

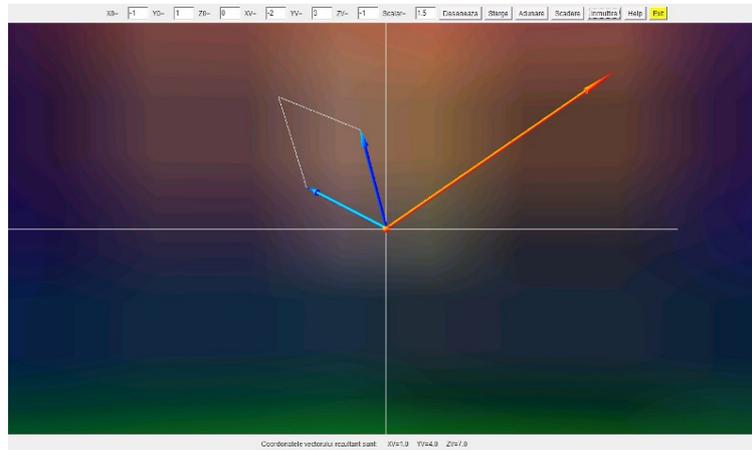


Figure 5. Interface developed for visualizing vector calculus

3.2 The applications developed for representing straight lines and planes

The Straight lines and planes in 3-dimensional Euclidian space ITTP defines the theoretical bases related to these topics and presents the application developed for visualizing them.

The vector equation of a line is given when one point on the line and the directing vector of the line are known or when two points on the line are known. The vector equation of the plane is given when one point of the plane and one basis are known or when the position vectors of three points on the plane are known.

In order to give the Cartesian equations of planes and lines, the notion of general (affine) Cartesian coordinate system is defined. The following are given: the equation of a plane passing through a point and which is perpendicular to a vector; the vector equation of the plane when one point and two of its direction vectors are known; the parametric equations of the plane; the equation of the plane in the form of a determinant; the equation of the plane through cuts. The necessary and sufficient conditions for two planes to be secant, parallel or to coincide are also presented.

The vector equation of a line is re-written in the form of parametric equations of the line when one point on the line and the direction vector are known, i.e. Cartesian or canonic equations. The canonic equation of a line is also given when we know the coordinates of two points on the line.

One of the developed applications allows generating the image of straight lines and planes.

As stated above, two versions of the application were created, the one for individual study being shown in Figure 6. The application has tags and text fields in the upper side for inputting data for two planes and two lines. By using buttons, 2 planes or 2 lines or one line and one plane can be selected for viewing. If the two planes or the two lines are parallel or if the line is parallel to the plane, the application displays this information. If the objects intersect each other, the application displays the angle between them in radians and the coordinates of the point or line of intersection.

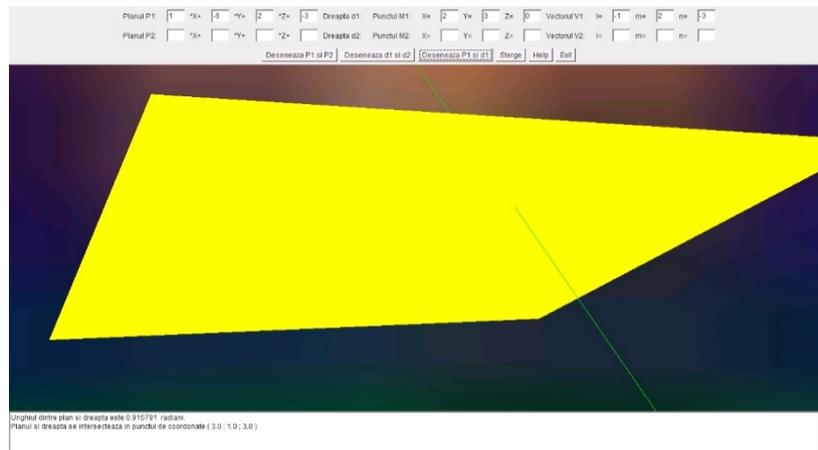


Figure 6. Interface developed for visualizing straight lines and planes

The notion of common perpendicular to two non-coplanar lines is defined in order to give the formula for distance between two non-coplanar lines.

The ITTP was thought out by the authors so as the students would assimilate the formulas presented as easily as possible.

The version for the autostereoscopic device does not display the results of calculations, but creates a pseudo-holographic image for perceiving the presented notions.

The applications were created using Java 3D API.

3.3 The applications developed for representing conics

The Hyperquadrics ITTP is structured into three parts, namely: sphere and circle, conics, and quadrics. With regard to sphere, the definition of a sphere and the implicit Cartesian equation is presented, as well as the general Cartesian equation and parametric equations of the sphere. A classification of points with respect to a sphere, position of a sphere to a straight line, a plane or another sphere is also presented. The circle is defined in 3-dimensional space as the intersection between a sphere and a plane, as well as between two spheres, case which can be reduced to the first one by considering one of the spheres and the radical plane of them.

One application is related to the calculus of the power of a point with respect to a sphere. In the version for individual study (Figure 3), in the upper side of the monitor there are tags and text fields for data input and the *Draw* button displays the image of the corresponding sphere. By clicking on the surface of the monitor, the application randomly draws a half line starting from that point, which intersects the sphere in two points. At the same time, it calculates and displays the power the point with respect to the sphere in the lower side of the monitor.

For two spheres, the data are entered successively and the radical plane is displayed (Figure 4). By clicking on the plane, two half lines are randomly drawn, starting from that point and intersecting the spheres. The application calculates and displays the power of the point with respect to the two spheres.

A radical axis is drawn for three spheres (Figure 3). By clicking on a point on this line, three half lines which intersect the spheres are randomly drawn. The power of the point with respect to the three spheres is calculated and displayed.

A radical point is obtained for four spheres. By clicking on the point, four half lines which intersect the spheres are randomly drawn and the power of the point with respect to the spheres is determined.

For better viewing, the *Rotate* button rotates the entire set of spheres around the Oy axis.

Another application presents the intersection between a sphere and a straight line, a sphere and a plane and between two spheres (Figure 7).

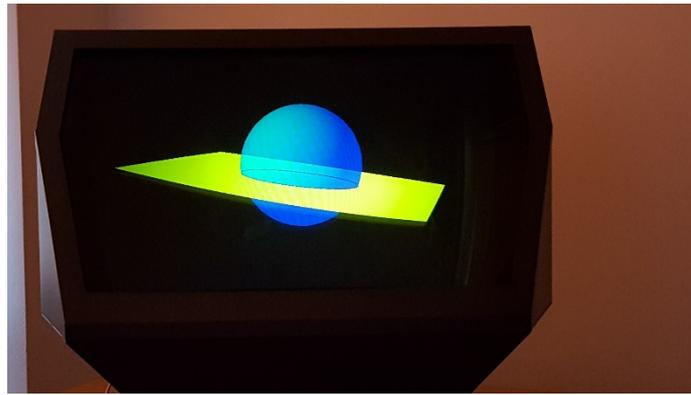


Figure 7. Interface developed for visualizing intersections between a sphere with a line, a plane or a second sphere

For the individual study version, the coordinates of the center and radius of the circle obtained as intersection are calculated and displayed in the lower side of the monitor (Ciobanu, Naaji, Virag, Dascal, 2016).

To have a better image over the 3D model and to see it from different angles, the application allows rotating it.

These applications were also made using Java 3D API.

With regard to conics, the theoretical part of ITTP presents them by canonical and general equations as well as the intersection between a conical surface and a plane.

An application was developed using Java, to represent conics starting from their general equations.

Orthogonal invariants are defined and the center or vertex of the conic and the axes (or axis) of symmetry are determined. In the version for individual study, all results, including the nature and genus of the conic, are displayed in the left side of the monitor (Figure 8).

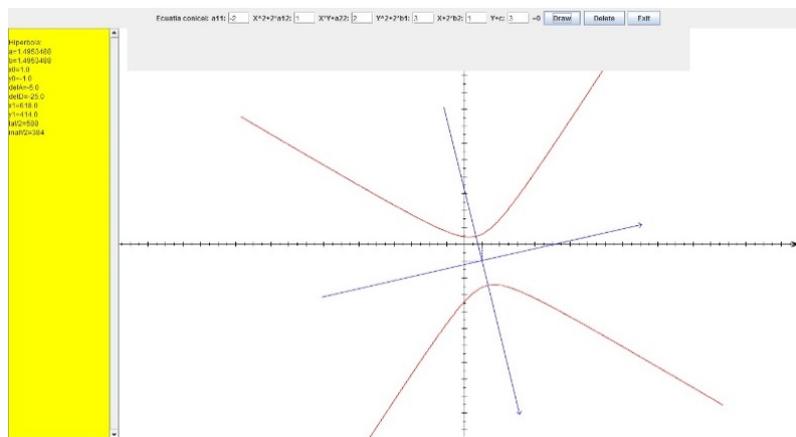


Figure 8. Interface developed for visualizing conics given by their general equations

In another application, the software module used for explaining the conics and the intersection between a cone surface and a plane was written in JavaScript. The 3D model can be rotated with a regular 2D mouse or with a gyromouse that has built-in gyroscopes which allows for three degrees of movement freedom, enhancing the user experience. For a better view of the conic section, the 3D model can be rotated. Different conics such as: circle, ellipse, hyperbola or parabola can be obtained by the movement of the plane on the Ox and Oy axes, or its rotation around the Ox axis. (Figure 9) (Ciobanu, Naaji, Virag, Dascal, 2016; Ciobanu, Ploscar, Dascal, Virag, Naaji, 2015).

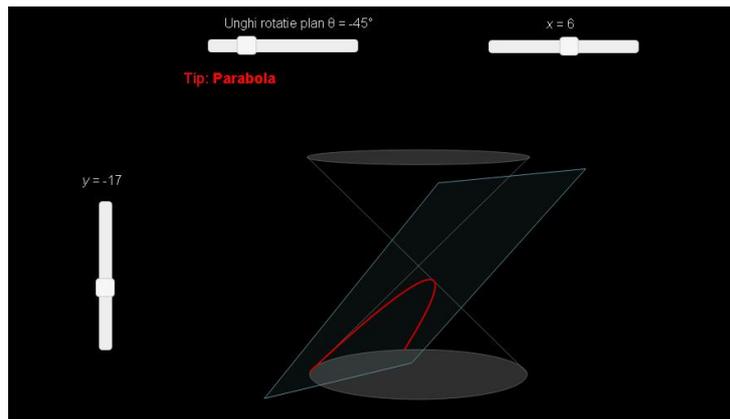


Figure 9. Interface for visualizing a conic section

The source code for the applications is based on standards like HTML5, CSS, the jQuery API (jQuery API Documentation, 2013) for the user interface, the Raphael library (The Raphaël JavaScript library home page) for working with vector graphics and the Conics3D JavaScript extension written by Lodewijk Bogaards (Bogaards, 2016) for the visualization of conic sections.

4. Conclusions

The interactive teaching tools packages (ITTPs) are designed to present some topics from mathematical disciplines, in order to facilitate the understanding of these abstract subjects. Through our applications, for topics from linear algebra and analytical geometry, geometric objects can be visualized with help of an autostereoscopic display where the 3D model can be observed and rotated. The display does not require the use of any kind of special glasses. The applications were developed in two versions, one for the autostereoscopic display, and one for regular 2D flat screen.

The resulting ITTPs are freely available on the involved partner universities e-learning platforms and project web pages.

The preliminary results and feedback from the target groups are promising and future testing is required to gain insights regarding the execution times of the various ITTPs.

Acknowledgment:

The results presented here were financially supported by the Hungary-Romania Cross-Border Co-operation Programme 2007-2013, under the grant no. HURO/1001/040/2.3.1., entitled: Development of Open Source Software and Interactive Teaching Tools for Mathematics – SOFTMAT (<http://softmat.uvvg.ro>). The project was implemented by College of Nyiregyhaza, Hungary as lead partner, and “Vasile Goldis” Western University of Arad, Romania.

References

- Ammeraal, L. & Zhang, K. (2007). Computer Graphics for Java Programmers. John Wiley & Sons Ltd.
- Andrica, D. & Topan, L. (2004). Analytic Geometry. Cluj University Press, Cluj-Napoca.
- Atanasiu, G., Munteanu, G., & Postolache, M. (2003). Algebra liniara, geometrie analitica si diferenciala. *Ecuatii diferentiale*. Bucharest: Fair Partners.
- Bogaards, L. (2016). Conic Sections app. Retrieved from <http://www.mrhobo.nl>
- Ciobanu, M., Naaji, A., Virag, I., & Dascal, I. (2016). Interactive Applications for Studying Mathematics with the use of an Autostereoscopic Display. *Soft Computing Applications, series Advances in Intelligent and Soft Computing*, vol. 356, Springer International Publishing Switzerland, 673-680.
- Ciobanu, M., Ploscar, A., Dascal, I., Virag, I., & Naaji, A. (2015). Interactive Teaching Tools for Visualizing Geometrical 3D Objects Using Pseudo Holographic Images. *Recent Advances in*

Computer Science, Proceedings of the 19th International Conference on Computers, Series Recent Advances in Computer Engineering Series, nr. 34, 215-218.

- Geng, J. (2013). Three-dimensional display technologies. *Advances in optics and photonics*, 5(4), 456–535.
- Gocheva, S. & Rahnev, A. (2008). New Challenges in e-Learning of Mathematics via EVLM and IDeLC projects. *New Aspects of Engineering Education, Proceedings of the 5th WSEAS / IASME International Conference on Engineering Education*, 437-442.
- Iltina, M. & Iltins, I. (2013). Mathematics with Software Mathematica at Riga Technical University. *Recent Advanced in Educational Methods, Proceedings of the 10th International Conference on Engineering Education*, 81-86.
- Iordan, A. (2009). Development of Interactive Software for Teaching Three-Dimensional Analytic Geometry. *Proceedings of the 9th WSEAS International Conference on Distance-learning and Web Engineering*, 23-28.
- Jeon, H. J., Park, G., Gwag, J. S., Lee, J. H., & Kwon, J. H. (2014). Color-Matching Liquid Crystal Display using a Lenticular Lens Array and RGB Light Sources. *Journal of the Optical Society of Korea*, 18(4), 345-349.
- jQuery API Documentation. (2013). Retrieved from <http://api.jquery.com>
- Kim, S. K., Yoon, K. Y., Yoon, S. K., & Ju, H. (2015). Parallax barrier engineering for image quality improvement in an autostereoscopic 3D display. *Optics Express*, vol. 23 (10), p. 13230.
- Lanman, D., Hirsch, M., Kim, Y., & Raskar, R. (2010). Content-Adaptive Parallax Barriers: Optimizing Dual-Layer 3D Displays using Low-Rank Light Field Factorization. *ACM SIGGRAPH Asia 2010 papers (SIGGRAPH ASIA '10)*. ACM, New York, NY, USA, Article 163.
- Lucente, M. (1994). *Diffraction-Specific Fringe Computation for Electro-Holography* (Doctoral Thesis Dissertation). MIT Dept. of Electrical Engineering and Computer Science.
- Mustea, A., Naaji, A., & Herman, C. (2014). Using Moodle for the Development of Massive Open Online Courses. *Let's build the future through learning innovation, Proceedings of the 10th International Scientific Conference "eLearning and Software for Education", Volume 1*, Carol I National Defence University Press.
- Osmani, A. (2014). *Learning JavaScript Design Patterns*. Retrieved from <http://addyosmani.com/resources/essentialjsdesignpatterns/book>
- Provision Company webpage. (2016). Retrieved from <http://www.provision.tv>
- The Raphaël JavaScript library home page. Retrieved from <http://raphaeljs.com>
- Xie, H., Zhao, X., Yang, Y et al. (2012). Cross-lenticular lens array for full parallax 3-D display with crosstalk reduction. *Science China Technological Sciences*, 55 (3), 735–742.



Monica Ciobanu (b. August 15, 1973) received her BSc in Mathematics (1996), MSc in Operator models, variety of applications and optimal control (1997), PhD in Mathematics (2003) from West University of Timisoara, Romania. Now she is lecturer of mathematics in Department of Engineering and Computer Science, Faculty of Economics, Engineering and Computer Science, “Vasile Goldis” Western University of Arad, Romania. Her current research interests include different aspects of linear algebra, differential geometry, dynamical systems, graph’s algorithms and she focused on applications in biology and medical informatics. She has (co-) authored 2 books and more than 25 papers, more than 40 conferences participation. She participated in more than 5 projects, funded by EU, national or international research authorities. Within the Hungarian - Romanian cross-border project, Development of Open Source Software and Interactive Teaching Tools for Mathematics Mathematics – SOFTMAT, she has been project assistant and coordinator of the project for the “Vasile Goldis” Western University of Arad, Romania. She is a member of Tensor Society, Japan, of Balkan Society of Geometers, Romania, and SRIM (Romanian Society for Medical Informatics).



Antoanela Naaji (b. February, 1963) received her MSc in Automation and Computers (1987), PhD in Mechanical Engineering (2005) from “Politehnica” University of Timisoara, Romania. Now she is associate professor of computer science in Department of Engineering and Computer Science, Faculty of Economics, Engineering and Computer Science, “Vasile Goldis” Western University of Arad, Romania. Her current research interests include web technologies (e-learning), biomechanics (modeling of biological systems) and medical informatics. She has (co-) authored and (co-) edited 13 books and more than 90 papers. Her preoccupations in the above-mentioned fields are reflected in a number of over 20 projects, as manager or member, funded by EU, national or international research authorities or county council. She is a senior member of IEEE, member of EAEEIE (European Association for Electrical Engineer and Information Engineering), AGIR (General Association of Romanian Engineers) and SRIM (Romanian Society for Medical Informatics).



Ioan Dascal (b. April 15, 1963) received his BSc in Informatics (2008), MSc in Web Development Technologies (2010) from „Vasile Goldis” Western University of Arad. He also received his MSc in Chemical Engineering (1989) from "Babes-Bolyai" University of Cluj-Napoca. Now he is C++ programmer. His main area of interest is programming in C, C++, Java, JavaScript, F#, HTML, XML, CSS, PHP, MySQL, etc. His current research interests include functional programming. He has (co-) authored 4 papers.



Ioan Virag (b. March 30, 1977) received his BSc in Informatics (2009) and MSc in Web Development Technologies (2012) from „Vasile Goldis” Western University of Arad. Since 2013 he is PhD candidate in Information Technology and Computers at the Politehnica University of Timisoara, Automation and Computer Science Faculty. His current research activities include Web-Based Medical Imaging Applications and Gesture-Based Interaction in Biomedical Interfaces. He has (co-) authored more than 10 papers and attended numerous international conferences.