

Interactive Applications for Studying Mathematics with the Use of an Autostereoscopic Display

Monica Ciobanu, Antoanela Naaji, Ioan Virag and Ioan Dascal

Abstract The abstract character of mathematical concepts can often discourage students who need to assimilate new materials. In order to help them overcome this problem, we developed a set of interactive teaching tool packages (ITTPs) consisting of theoretical modules and applications. A pseudo-holographic display was used for visualizing 3D objects and the outcome of modifications of the parameters inside mathematical formulas. One of the starting ideas was to develop the software using open-source libraries capable of manipulating 3D models like Java 3D. We extended the concept to work inside a web browser as well, in order to make the applications available on mobile devices and eliminate the need to install any other software.

Keywords Autostereoscopic display · Teaching tools · Pseudo-holographic images

1 Introduction

In today's scientific world, the use of holographic-like images for developing applications in various fields such as medical imaging, industry, military applications, entertainment, advertising, or computer game development represents a cutting-edge technological approach.

The main goal of our research was to use this technology to improve the quality of education by developing new visual-interactive methods for teaching mathematics.

Our team further developed an idea started by Wolfgang Schlaak from the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute [1]. In their research, they built an application that allowed physicians to study the patients' CT scans by visualizing them in 3D, with the possibility of rotating the image by means of hand gestures (called noncontact image control). The display they used was an

M. Ciobanu · A. Naaji (✉) · I. Virag · I. Dascal
Department of Computer Science, "Vasile Goldis" Western University of Arad,
96 Liviu Rebreanu St. Arad, Satu Mare, Romania
e-mail: anaaji@uvvg.ro

autostereoscopic multiview display, which involved tracking the user's eyes by means of a set of cameras mounted on top of the display and adjusting the image accordingly, to produce a sensation of depth.

In our research, we used the latest technology in the field of image visualization consisting of HoloVision aerial image display, produced by the Provision company [2]. This new type of autostereoscopic display does not require any cameras for tracking the viewers' eyes, being based on Pepper's Ghost technique, and allows a larger number of persons to visualize the projected 3D models without the use of special glasses.

For the time being, the HoloVision display was only used for marketing purposes; to our knowledge, we are the first scientific team that used the aerial image display for educational purposes.

The main differences between the two technologies is that the monitor we used does not have the 3D image attached to the display, but rather exhibits similarity to an actual hologram. The monitor uses a special mirror in order to focus and display the pseudo-3D holographic image.

Another key objective was to use available open-source libraries in order to allow other teams to further develop the applications or to adapt the existing ones to different fields of study.

The applications were developed to run on both classical computer architecture and holographic display for an enhanced user experience.

In Fig. 1, we can see that the user interface developed for the 2D monitors is different from the one developed to display the images on the autostereoscopic

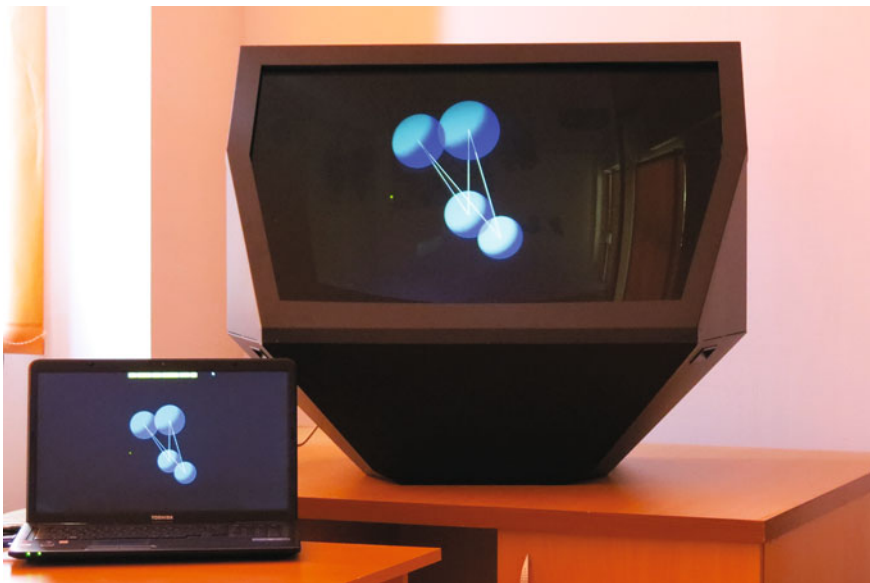


Fig. 1 Interfaces developed for laptop and holographic display

device. It was necessary to create two types of interfaces because, the HoloVision display is optimized to render 3D models focusing the image outside of the display area, providing the illusion that the image floats in midair in front of the monitor. The 3D image cannot be rendered in the same way on a 2D monitor, but the 3D model keeps its details on the HoloVision display even if it is scaled.

The two versions of the software were needed for individual study on a 2D monitor and for presenting the subjects to a group of students using the 3D display, as the viewing angle and distance allow this. The HoloVision holographic display that we used is able to display a 30 cm image focused at approximately 90 cm in front of the monitor, viewable at a 60-degree angle from the center and does not require the user to wear any kind of special glasses.

2 The Software Developed for Interactive Teaching Tool Packages Using Open-Source Libraries

The autostereoscopic display was used for ITTPs related to linear algebra and analytical geometry. We developed the programs based on Java 3D API and JavaScript because of their open-source nature. This was necessary in order to allow others to contribute with similar applications for different fields of study where visualization plays an important role such as architecture or medicine.

2.1 The Applications Developed by Using Java 3D API

The Java 3D API is an application programming interface used for creating portable 3D applications and applets. It provides a library for creating and manipulating 3D models and for constructing the structures used in rendering the scene. A scene graph is used to construct the scene, which is structured as a tree that contains the elements necessary to display the 3D objects. Developers can easily describe very large virtual worlds using this approach. Java 3D delivers Java's famous "write once, run anywhere" benefit to developers. Reference [3] presents a very detailed description regarding the mathematical concepts used behind the scenes in Java 3D.

Using Java 3D API, we developed applications for vectorial spaces, straight lines, and planes in three-dimensional Euclidean space, sphere and circle in space and conics in Euclidian plane.

The first application refers to the calculus of the expression of a vector with the help of its coordinates with respect to a base, solved using elements of matrix calculation. The graphical interface of the developed software can be seen in Fig. 2.

The second application solves problems related to elements of vector calculation and implement basic algebraic operations such as scalar multiplication, vector addition, and vector product. The created program allows the representation and rotation around axis of the vectors obtained by the addition of vectors using the

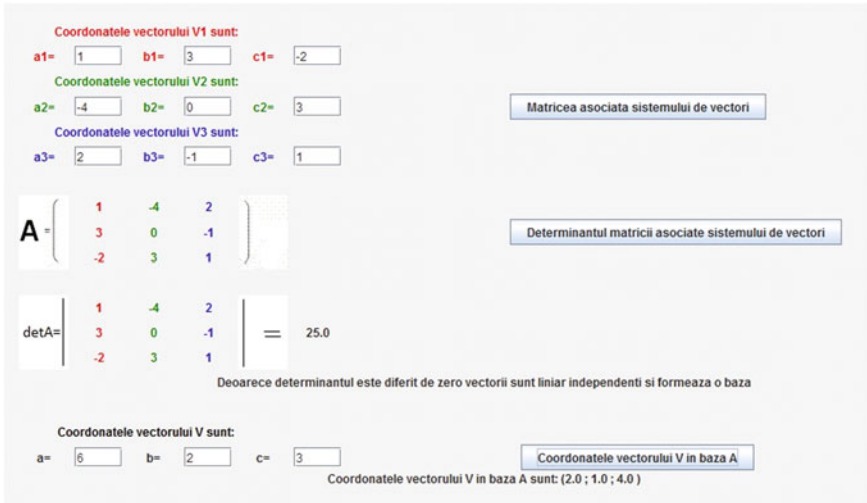


Fig. 2 Interface for the calculus of coordinates of a vector

parallelogram or the triangle rule, by multiplication of vectors by scalars, or by vector product calculation (Fig. 3).

Other applications are used to determine the intersection between two lines, two planes, a plane and a line, a line and a sphere, and a plane and a sphere (Fig. 4). They allow the calculus of the angle between lines and planes, the power of a point with respect to a sphere and draw the radical plane of two spheres, the radical axis of three spheres, and the radical center of four spheres.

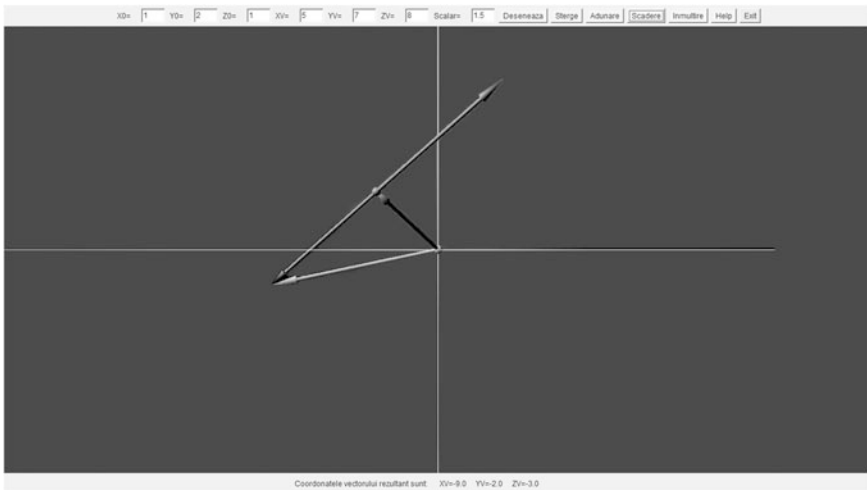


Fig. 3 Interface for vector calculus

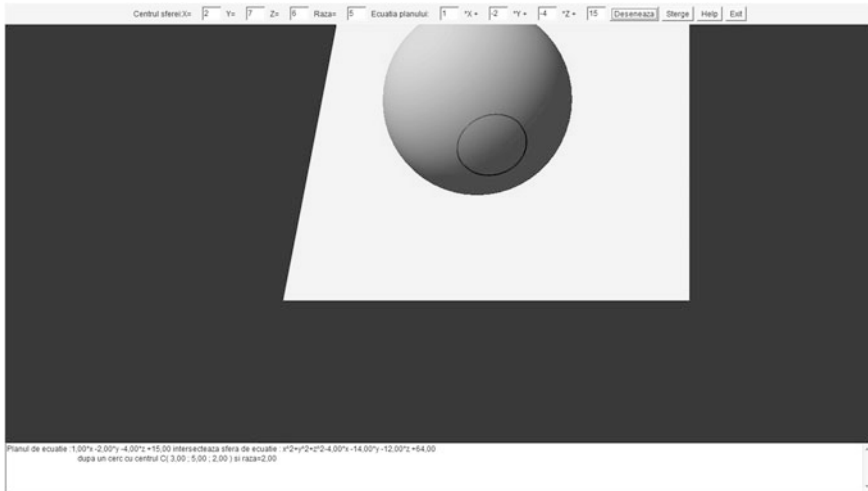


Fig. 4 Interface for visualizing the intersection between a sphere and a plane

In Fig. 4 is shown the circle obtained as the intersection of a sphere and a plane. The application also can illustrate the circle as intersection of two spheres and analyze all possible situations regarding the distance between the centers of the two spheres or from the sphere to the plane.

Another application draws conics starting from their general equation (Fig. 5). For the graphic representation of the nondegenerate conic, with or without center, the canonical equation of the conic, the center of the conic, and the axis of symmetry are determined. Thus, the invariant orthogonals are calculated and, based on their values, different types of conics are represented (ellipses, hyperbolas, parabolas as well as degenerate conics), along with complete solutions.

2.2 The Applications Developed by Using JavaScript

For the visualization part of the conic sections, we decided to use a JavaScript-based solution [4]. This approach, besides the fact that can be run inside any up to date browser, will also allow, if there is needed, to extend the applications to be run on mobile devices.

Using JavaScript, we developed applications for representations of conics as intersection of a cone surface and a plane. The user interface is a simple one and allows the movement of the plane on the Ox and Oy axes or the rotation of it around the Ox axis. It is also possible to rotate the whole 3D model for viewing the intersection of the plane with the cones from different angles (Figs. 6 and 7).

Conic sections, as the circle, ellipsis, parabola, or hyperbola, can be viewed in real time by adjusting the cursors.

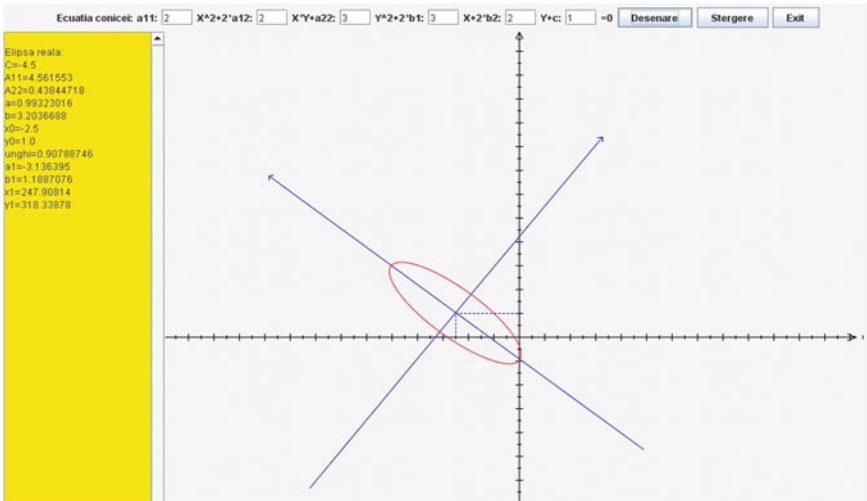


Fig. 5 Interface for the representation of conics in euclidian plane

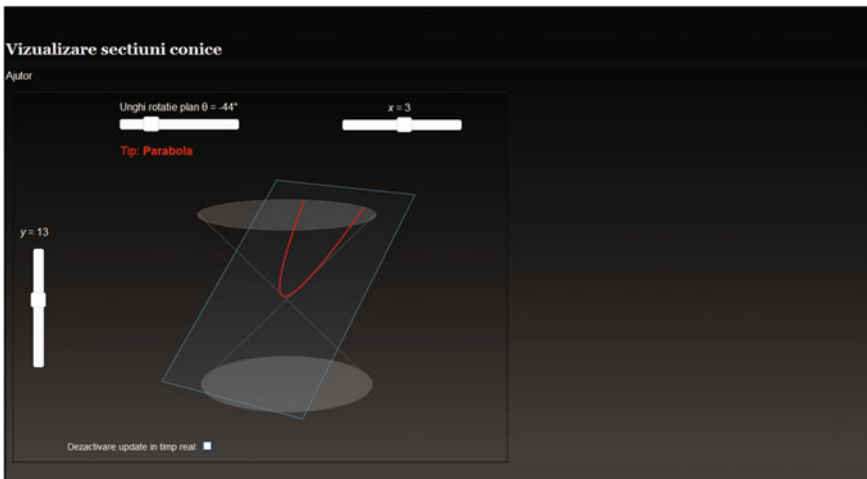


Fig. 6 Interface for visualizing a parabola

The source code for all the applications is based on standards like HTML5, CSS, the jQuery API [5] for the user interface, the Raphael library [6] for working with vector graphics, and the conics3D JavaScript extension written by Lodewijk Bogaards [7] for the visualization of the conic sections.

It is also possible to deactivate the real-time update of the image if the application runs on older browsers.

The applications also have a help section where the different options are explained to the user.

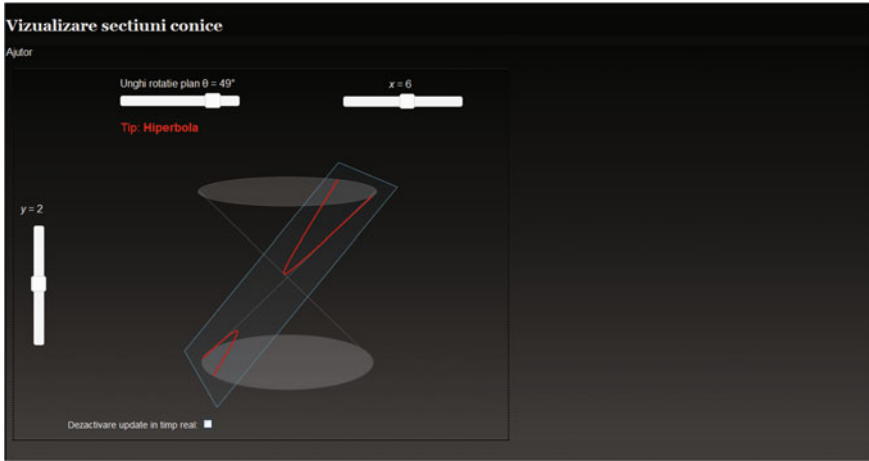


Fig. 7 Interface for visualizing a hyperbola

3 Conclusions

The ITTPs designed with the help of software are an illustration of the ability to study, research, and develop, facilitating the learning of mathematical subjects.

The interactive applications conceived for the topics of ITTPs facilitate the understanding of abstract notions, following the perception gained by the student from direct interaction with the three-dimensional geometric object. The programs created within our research provide the possibility to regard this object from different angles.

The use of autostereoscopic display provides a new approach, both from the technical point of view and teaching methods.

Acknowledgments The presented results were financially supported by the Hungary-Romania Cross-Border Co-operation Program 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>).

References

1. Fraunhofer Institute of Research webpage (<http://www.fraunhofer.de/archiv/pi-en-2004-2008/EN/press/pi/2007/11/ResearchNews112007Topic3.html>)
2. Provision Company (<http://www.provision.tv/>)
3. Ammeraal L, Zhang K (2007) Computer graphics for java programmers, Wiley, Hoboken

4. Osmani A (2014) Learning JavaScript design patterns, available online at <http://addyosmani.com/resources/essentialjsdesignpatterns/book/>
5. jQuery API Documentation <http://api.jquery.com/>
6. The Raphaël JavaScript library home page <http://raphaeljs.com/>
7. Bogaards L, Conic Sections app, <http://www.mrhobo.nl/>