

Web Based Reality With Structure Sensor

¹Georgi Krastev, ²Valentina Voinohovska, ³Svetlozar Tsankov

¹Dept. of Computer Science, University of Ruse

^{2,3}Dept. of IIT, University of Ruse

Abstract

The article describes a 3D-sensor (scanner) of close (to 3.5 m) virtual and augmented reality Structure Sensor. The WEB based technologies for visualization of three-dimensional objects and especially WebGL and Three.js library of JavaScript are the point of discussion. An architectural template of design in programming Model-View-Controller or MVC, based on the separation of business logic from the GUI and data in an application, is attached to the paper. Attention is paid to photorealism in three-dimensional visualization and the results have been shown.

Keywords

Virtual Reality, Augmented Reality, 3D Sensor, Web Based

I. Introduction

The 3D-scanner Structure Sensor allows the nearby surroundings to be scanned and moved to virtual and augmented reality [7]. It allows the creation of digital models of objects and the environment in movement. Such opportunities open up new horizons for computer games, training programs, marketing, architecture, design and many other applications.

The scanner operates through a process known as “structured light.”

An infrared laser source applies an invisible dot pattern on objects located in front of the scanner. Simultaneously, an infrared sensor registers the distortions in the point model while moving the sensor around objects with a VGA resolution. This type of observing points and distortion creates a depth map of the 3D scene and the photographed objects. The software uses the camera of iPad, to which it is attached in order to obtain information about the colors of the environment and combining it with the rest of the collected data, displays a realistic 3D reality.

The Structure sensor is attached to the iPad device (adapted for use with devices of the fourth generation iPads and iPad mini) (Fig. 1).

The technical data of Structure sensor are:

- Recommended Range (Operating distance): 0,4 m-3,5 m.
- Precision: 0.5 mm at 40cm (0.15%), 30 mm at 3m (1%).
- Accuracy: 1% of the working distance.
- Resolution: VGA (640x480), QVGA (320x240).
- Field of View (viewing angle): Horizontal: 58 degrees, Vertical: 45 degrees.
- Battery Life: 3-4 hours of active sensing, 1000 + hours of standby.
- Operating Temperature: 0° to 35°C.
- Dimensions, Weight: 119, 2x27, 9x29 mm, 95 g.

When working with Structure sensor there are three types of information flows: IR, Depth, and Depth + Color. The IR flow is the only one with infrared data. The flow of Depth offers a stream of false color to distinguish between near and far objects using hot to cold color schemes to show close objects (red) compared with distant objects (blue). The flow Depth + Color creates data using both real and artificial colors based on depth. The information can be stored in iPad, or can be sent to another computer via Wi-Fi.

By using a special cable (USB Hacker Cable) the sensor can be connected to other drives, can be used with platforms like Android and Windows, or can be used for other functions for 3D printing.

This new technology allows 3D images in motion to be scanned with some limitations.



Fig. 1: Securing Structure Sensor To iPad

II. Visualization Technologies of 3D objects

The variety of programming languages in the late 90s of the 20th century and early 21st century favor the creation of semi-finished products and libraries to work with three-dimensional objects. Most of them are based on primitives built by graphics libraries such as WebGL (Web Graphics Library) [1, 5] and OpenGL (Open Graphics Library). These libraries provide direct access to the functionality of GPUs and thus provide hardware acceleration in the processing of two-dimensional images and three-dimensional scenes.

Most technologies used for visualization of three-dimensional objects are made in JavaScript, but there are solutions developed in high-level languages such as Java. Among the most used libraries are Google O3D [8], Google Indoor View [9], Three JavaScript [10], X3DOM [11], JSC3D [12] and Java applet [13]. The emphasis in technology development is on simplifying the functionality of basic primitives and optimization of performance and memory usage. An essential criterion is also the supported operating systems and browsers of the working technologies. The support of mobile versions also contributes to the widespread use of these libraries.

III. Three JavaScript

Among some of the developments related to the HTML 5 and WebGL technologies, due consideration is given to Three.js as a library for the presentation of three-dimensional objects [1, 5, 14].

The focus of development is on the complexity of Three.js and the opportunities it provides as the core of applications including three-dimensional scenes or 3D games.

Three.js library is written in JavaScript and is built around WebGL, using the functionality of the library with graphic primitives for the needs of developers using scripting languages [10]. This library is an often used technology due to its flexibility. During the development, various elements supporting the building of a realistic three-dimensional scene can be used.

When objects are illuminated one or several points of illumination can be used. Types:

- AmbientLight - global illumination;
- AreaLight - illumination in a limited area;
- DirectionalLight - targeted illumination where the rays are parallel to each other;
- HemisphereLight - lighting identical to DirectionalLight, but positioned above the three-dimensional scene;
- PointLight - illumination from a point and with certain intensity - suitable for simulation of a light bulb;
- SpotLight - identical to PointLight, but oriented at a specific area.

The library provides a mechanism for loading objects dynamically using AJAX (Asynchronous JavaScript and XML) applications to the resources of the server. For the convenience of developers, several objects with which to load parts of the three-dimensional scene are provided:

- JSONLoader - to load three-dimensional objects from data in JSON format;
- TextureLoader - to load images that will be used for visualizing surfaces;
- SceneLoader - to load whole 3dimensional scenes.

There are three options when starting the initialization of a scene: a normal scene without effects (Scene), a scene with fog linear dependence of distribution attached to it (Fog) and a scene fog with exponential dependence of distribution attached to it (FogExp2). The animations Animation MorphTarget or KeyFrameAnimation can be applied to each scene and one of the following cameras can be used: CombinedCamera or CubeCamera

Among the primitives to visualize three-dimensional objects involved in the library are:

CircleGeometry,	ConvexGeometry,
CubeGeometry,	CylinderGeometry,
ExtrudeGeometry,	IcosahedronGeometry,
LatheGeometry,	OctahedronGeometry,
ParametricGeometry,	PlaneGeometry,
PolyhedronGeometry,	ShapeGeometry,
SphereGeometry,	TetrahedronGeometry,
TextGeometry,	TorusGeometry,
TorusKnotGeometry,	TubeGeometry.

For more accurate visualization of complex objects, small subprograms are used. They are submitted directly to the GPU in visualization or the so-called shaders. In Three.js following possible shaders can be used: ShaderChunk, UniformsUtils, UniformsLib or ShaderLib.

The disadvantage of Three.js software developers point out [6] is the complexity and the smooth curve of studying the technology. Therefore, the implementation of applications shifts in time and in spite of the rich set of primitives the library requires too much

time for absorption and penetration and is often overlooked. The advantage of the library over other technologies for visualization is the large number of software developers using it

IV. Model-View-Controller

An architectural template for design in programming - Model-View-Controller - is attached to the paper (Model-View-Controller or MVC). It is based on the separation of business logic from the GUI and the data in an application (Fig. 2).

The model is the core of the application of predetermined area that is being developed. Generally, it is the data from the real world that are modeled and that have to be worked on - to enter, modify, display, etc. We need to distinguish between the real world around us and the imaginary abstract model world that is the product of reason, which is perceived as statements, formulas, mathematical symbols, schemes, diagrams and other aids.

The view is this part of the application source code responsible for displaying the data of the model. For example, the view may consist of JSP pages, ASP pages, JFrame following form Swing application. It depends on what GUI is done and what platform is used.

The Controller is the part of the source code (or class, or library) that takes the data from the model or calls additional methods on the model, pre-processes data, and only then submits them to the view. When creating a web GUI, the HTML code can be easily modified even by non-programmers - they will look upon the template as an ordinary HTML page.

The advantages of MVC are:

- The model is independent of the controller and the view;
- The model can be planned and implemented independently of other parts of the system;
- Various views (interfaces) can be implemented for one and the same pattern- such as web interface and native interface to Facebook;
- The controller and the view can be changed without changing the model;

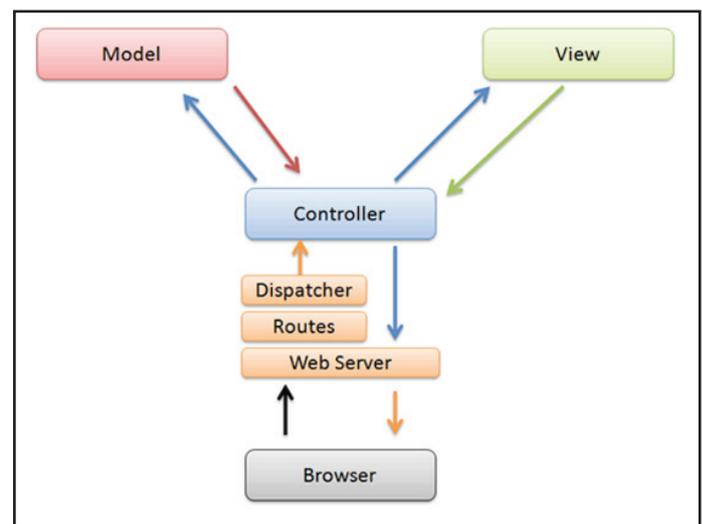


Fig. 2: MVC Model

The logical sequence of the way the developed application works includes the following steps: declaring the necessary variables; creating a scene in which the object will be displayed; tuning the operation with a camera; adding lighting that will improve the contrast and the visualization of the object; additional loading of an object; adding controllers for managing the object; visualization of the object.

As the process of the computer visualization seeks to mimic the process of shooting with a real camera, one of the main tasks is to create a synthetic computer camera. Each product of three-dimensional visualization and animation has such a camera, all of which share many characteristics, but also have some specific features, which requires examining them in particular [158].

The synthetic computer camera has the following features (Fig. 3):

- The near limit is used as a projection plane (screen);
- A direction of observation is introduced too, starting from the chamber (center of projection) and passing through the center of the near and the far limit of the apparent volume;
- Out of the linear parallel projections, most commonly an orthogonal and less often axonometric projection appears. It is rare to come across cameras that allow setting a direction of observation which is not perpendicular to the plane of the camera, i.e., that have the ability to create oblique projections;
- With perspective projections, the single-point perspective projection is basically used.

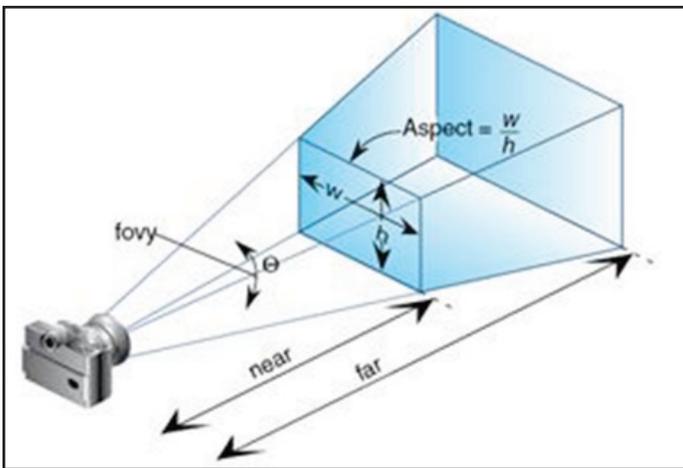


Fig. 3: Main Characteristics of the Synthetic Camera: Frustum and its Parameters

The main characteristics of the synthetic camera that need to be set in computer visualization programs are:

- Set the frustum - this includes setting the visible angle and depth of the volume.
- aspect ratio - set the ratio of width to height of the visible frustum. Modern products allow setting different ratios according to the specific needs of the visualization and its application;
- visible angle - the parameter is used to simulate normal, wide angle and telephoto cameras. The Field of View - fov- is set, while the horizontal angle is calculated from it using the ratio "width / height". In perspective projection, the parameter determines the degree of perspective distortion
- depth of volume - the parameter is used to remove objects from the area of the focusing of the camera. For this purpose, the near and far limits, which are parallel to one another, are used. The reason for setting the far limit is that objects behind it will be viewed as very small and hard to see or the influence of these objects needs to be removed and the observer needs to focus on a particular part of the scene. There are a few reasons that cause the near limit to appear. The most important reasons are: nearby objects become very large and the scene cannot be seen; nearby objects get distorted too much and their form cannot be perceived; the possibility to visualize objects located behind the camera is lost.

- It is less common to visualize objects unfocused, i.e. the receipt of obscure objects that are beyond the visible and focused volume

- The position of the camera and viewing direction - the camera can be located anywhere in the world coordinate system and can have any direction of observation.
- The orientation of the camera - it sets the rotation of the camera around the direction of observation, i.e. the direction 'vertical' of the camera is not parallel to the axis 'y' of the global coordinate system.
- The Coordinate System camera is a right-oriented coordinate system.
- Projection plane:
 - What is used for perspective projection is either the front limit of the visible volume or a plane parallel to the front and rear limit of the visible volume and is at a distance F from the front limit and in the rear limit;
 - For parallel projection, it doesn't matter where the plane is located because there is no perspective distortion, which could be influenced by the distance between the object, the projection plane and the center of the projection.

V. Photorealism in 3 Dimensional Visualization

Photorealism emerged as a trend in painting, which seeks to recreate all the elements of the painted scene as accurately as possible, i.e. full correspondence with the smallest details in the shape and texture of objects, lighting, and perspective observation. This concept has now transferred in computer visualization too, i.e. photorealistic computer visualization seeks to recreate "credible photographic details of the displayed objects". This requires specialists in computer visualizations to pay particular attention to four key elements for each scene: the form of objects, the type and quality of textures and materials of objects, realistic lighting and camera position - choice of perspective.

One of the most discussed features of any three-dimensional visualization is its realism. This concept has many different interpretations and elements most often discussed of which are the two types: physical realism and visual realism [2-3].

In literature, physical realism means full correspondence with the physical nature of the environment that is modeled and visualized. This includes four basic elements of any environment: accurate models of objects, precision in lighting, precision of movement of objects and the accuracy of the behavior of objects (their reaction to external influences). In general, this cannot be achieved because there are no precise mathematical and physical models of all the types of matter and all types of behavior. So physical realism is most often sought only in one of the elements of the scene, for example, the realism of the model, light or movement.

Visual Realism is associated with the observer's assessment of the realism of the displayed objects and scenes. The flexibility of this definition depends on the viewer's ability to assess the realism.

The three most common factors influencing the assessment of 'visual authenticity' are:

- Whether the viewer sees the object / s 'well' - in this case factors are the size of the object, the distance of the object, the lighting in the scene, the presence of other objects in the scene that 'attract' the viewer's eye (in this case the way of looking is changed from focused to peripheral, ie the perception of color is reduced) and the rate of passage through the scene (with computer animations and simulations).
- What is the viewer's understanding of 'realistic' for this type of objects - that is related to the stereotypes that one has a understanding and response.

- ‘Awareness’ of the visual objects is the third factor - of objects that the viewer sees every day, there is one understanding of realism, while things one has never seen, one can only imagine them what they should be like (this depends heavily on the stereotypes of the viewer).

For a better user experience, the opportunity to manage the object through mouse is included- a new library OrbitController is added, allowing a change of coordinates of the object in the scene through mouse. The results obtained are shown in fig. 4.

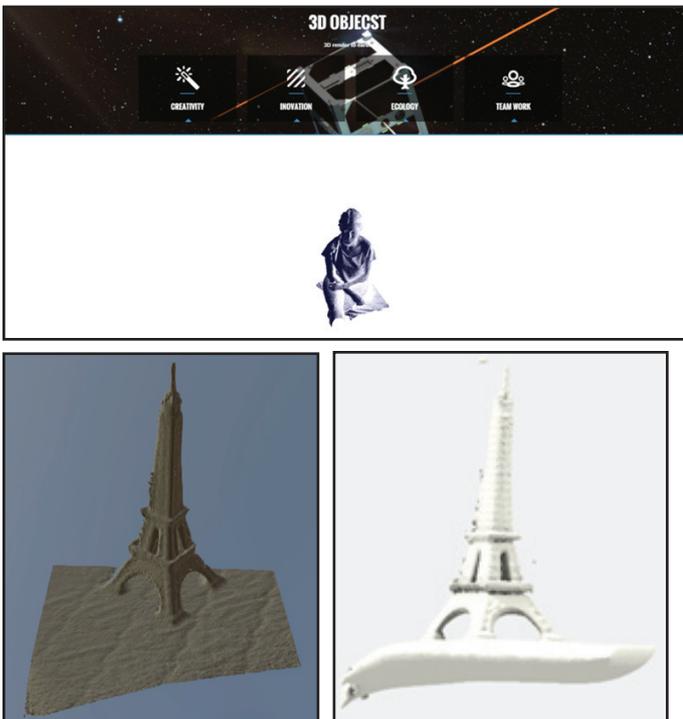


Fig. 4: Achieved Results

VI. Insurance Requirements to Hardware

In hosting (installing) the system of a web-server we should take into account the expected increase in disk space that the database occupies. The expected monthly traffic (bandwidth), depending on the daily impressions of the site, should also be considered so that an appropriate hosting plan should be selected providing the required disk space and traffic satisfying the needs of the site.

Testing of the development

Testing the development includes:

- WEB application is opened by different browsers: Internet Explorer; Mozilla Firefox; Google Chrome.
- Tests were conducted on different mobile devices with different resolutions: Galaxy Nexus - 720 × 1280; Samsung galaxy S3 - 720 × 1280; Samsung Galaxy S5 - 1080 x 1920; HTC sensation - 540 × 960.

It should be noted that in some cases a mismatch in the display of the content occurs.

VII. Conclusion

The established product displays 3D objects captured through Structure sensor by different users, giving them easy access. The creation of three-dimensional visualization requires of those working in the field of photorealistic visualizations to take into account the impact of many additional physical phenomena because otherwise artifacts are created that distort the sense of ‘photorealism’.

As with each new product and technology, especially considering the fact that this is the first version of Structure Sensor, some errors

occur in operation. One of the biggest drawbacks is the inability of the scanner to “see” objects with a solid color of every detail. Another problem that arises in the monitoring of materials is that sometimes the IR signal is not registered back to the sensor, as is the case with some plastics or materials such as liquids that distort IR signals. These are issues that will undoubtedly be resolved in subsequent versions of the product. It could be said that at the moment Structure Sensor is a prototype of what awaits us in the near future.

VIII. Acknowledgements

The present document has been produced with the financial assistance of the European Social Fund: project “Research of advanced audio-visual and information technologies in education”. The main aims and tasks of the project are: creating preconditions and introduction of innovative educational technologies in education and in particular in e-learning; analyzing the conditions, opportunities and constraints for implementation of modern educational technologies through wide use of audio-visual tools; offering specific advanced technological tools for specific group of disciplines; experimental verification of the proposed technological tools; further development of the e-learning software platform.

References

- [1] Danchilla B., Beginning WebGL for HTML5, 2012
- [2] Fleming, B. 3D Photorealism Toolkit. Wiley, 1999
- [3] Fleming, B. Advanced 3D Photorealism Techniques. Wiley, 1999
- [4] Gallardo, A. 3D Lighting: History, Concepts, and Techniques, Charles River Media, 2000
- [5] Parisi T., WebGL: Up and Running, 2012
- [6] Raasch J., JavaScript Programming: Pushing the Limits, 2013
- [7] Specification of Structure sensor, <http://structure.io/>
- [8] Specification of O3D, 2010, <http://code.google.com/p/o3d>
- [9] Specification of Google Street View, 2014, <http://indoorstreetview.com>
- [10] Specification of Three JavaScript, 2 014, <http://threejs.org>
- [11] Specification X3DOM, 2014, <http://www.x3dom.org>
- [12] Specification of JSC3D, 2014, <http://code.google.com/p/jsc3d>
- [13] Specification of Java applet, 2014, <http://docs.oracle.com/javase/tutorial/deployment/applet>
- [14] Williams J., Learning HTML5 Game Programming: Build Online Games with Canvas, SVG and WebGL, 2011.