

# A rendering approach for stereoscopic web pages

Jianlong Zhang, Wenmin Wang\*, Ronggang Wang, Qinshui Chen  
School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University,  
Lishui Road 2199, Nanshan District, Shenzhen, 518055 China

## ABSTRACT

Web technology provides a relatively easy way to generate contents for us to recognize the world, and with the development of stereoscopic display technology, the stereoscopic devices will become much more popular. The combination of web technology and stereoscopic display technology will bring revolutionary visual effect. The Stereoscopic 3D (S3D) web pages, in which text, image and video may have different depth, can be displayed on stereoscopic display devices. This paper presents the approach about how to render two viewing S3D web pages including text, images, widgets: first, an algorithm should be developed in order to display stereoscopic elements like text, widgets by using 2D graphic library; second, a method should be presented to render stereoscopic web page based on current framework of the browser; third, a rough solution is invented to fix the problem that comes out in the method.

**Keywords:** web, stereoscopic, browser, render, matrix

## 1. INTRODUCTION

Web technology provides a relatively easy way to generate contents for us to recognize the world. In recent years, technological developments have made it possible to build 3D content through the Web. 3D web visualization techniques have sprung up in many fields, such as learning, education and training [1], virtual communities, biomedical area [2]. Sixto Ortiz predicts that 3D on the web will do as well as video on the web has done [3] and Neal Leavitt believes 3D web browser will succeed [8]. But all the applications above are just using monocular 3D. Apart from monocular 3D, more specific depth information can be obtained from binocular cues, especially at shorter viewing distances. Based on these cues, viewers may achieve a more natural depth perception. Because our eyes are horizontally separated, each eye has its own viewpoint of the world and thus both eyes receive slightly different retinal images. Stereopsis is the perception of depth that is constructed based on the difference between these two retinal images; the brain fuses the left and right image and from retinal disparity, i.e., the distance between corresponding points in these images, the brain is able to extract depth information [4]. With the development of stereoscopic display technology, the stereoscopic devices are becoming much more popular. There will be many stereoscopic applications in some fields like games [5], films, 3D televisions and 3D mobile devices.

The combination of web technology and stereoscopic display technology will bring revolutionary visual effect. Many conventional applications in web can be written as web pages using HTML, CSS, JavaScript. Through adding new CSS label like “depth”, web pages are interpreted by the browser to make some regions display stereoscopic primitives in the stereoscopic devices. Stereoscopic 3D (S3D) web pages provide with a good supporting of stereoscopic games, media contents or other stereoscopic application. In May of 2011, a patent about S3D browser is presented [9]. On November 30th, 2011, W3C presents a draft about extensions for stereoscopic 3D support [10]. This paper mainly presents a simple method of rendering stereoscopic web pages based on current framework of the browser.

The rest of this paper is organized as follow: Section 2 describes the implementation of rendering S3D web pages, Section 3 presents the result of our experiment, and Section 4 makes our conclusion and points out the work next.

## 2. IMPLEMENTATION OF RENDERING S3D WEB PAGES

### 2.1 Transformation matrix

The stereoscopic display system consists of a single display surface on which left and right images are displayed. The toed-in camera configuration and the parallel camera configuration are compared in [6]. Toed-in camera configuration results

\* wangwm@ece.pku.edu.cn; phone 86 755 2603-5700; fax 86 755 2603-2100; ece.pku.edu.cn

in a curvature of the depth planes, which makes objects at the corners of the image appearing further away from the viewer than objects at the center of the image. And depth plane curvature is closely related to keystone distortion. But the parallel camera configuration results in depth planes which are parallel to the surface of the monitor. This paper adopts the parallel camera configuration in combination with horizontal image translation (HIT) [15].

This paper adopts matrix form because of its advantages. It is a very explicit form of representing orientation, scaling, shifting. This explicit nature provides some benefits [7]: (1) Graphics APIs use matrices to express orientation. When application communicates with the graphics API, you must express your transformations as matrices eventually. (2) Matrix multiplication is used to collapse matrices for nested coordinate spaces into a single matrix. Through matrix multiplication, a simple transformation matrix is deduced for transformation from 2D to S3D.

The geometry of a stereoscopic display process as three separate coordinate transformations: Firstly from the coordinate in the unit of pixel to the coordinate in the unit of millimetre based on the screen configuration, secondly from the coordinate in the unit of millimetre to the coordinates as the positions of left viewing frame and right viewing frame by using the depth information, thirdly from the coordinates in the unit of millimetre back to the coordinates in the unit of pixel for rendering. Because the browser uses pixel as basic unit in the stage of rendering.

- $L_H, L_W$  Screen Height and Width: The vertical and horizontal size of the display screen.
- $e$  Eye Separation: The distance between the observer's eyes. Typically 65mm.
- $R_H, R_W$  Vertical and Horizontal Screen Resolution.
- $V$  Viewing Distance: The distance from the observer's eyes to the display plane.
- $z$  Depth information: Multiples of one thousandth of viewing distance.

This first transformation is shown in the equation (1). It transforms the coordinate system that top left corner of the screen is the original point into the one that the center of screen is the origin. Besides, the graphic APIs uses pixel as basic unit. The new coordinate should be multiplied by the factor  $L_W/R_W$  or  $L_H/R_H$  that indicates horizontal length or vertical length per pixel respectively.

$$\begin{pmatrix} X_1 \\ Y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{L_W}{R_W} & 0 & -\frac{1}{2}L_W \\ 0 & \frac{L_H}{R_H} & -\frac{1}{2}L_H \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (1)$$

The variables of second transformation are shown in Figure 1. The coordinates  $X_{left}$  and  $X_{right}$  can be computed based on the property of similar triangles. Due to no vertical parallax, the coordinates of left view and right view in the y-direction are the same. We don't use absolute depth value but the integer multiple of a ratio of the viewing distance, because we want to avoid the case in which the different of viewing distance with the same absolute depth  $z_0$  leads to the different depth perception. For example, given an absolute depth  $z_0=3$ , the depth perception in the two conditions, with the viewing distance  $V_1=100$  and  $V_2=1000$ , is entirely different. The positive depth  $z$  indicates the pixel is out of the screen. Or else, the pixel is into the screen.

The final transformation is the inverse of the first one. The transformation of right viewing frame is similar to that of left one. Through the last transformation, the coordinates use pixel as its unit again.

The overall coordinate transformation of the three steps is shown in the equation (2)-(3). It just needs scaling and shifting in the x-direction and y-direction. The difference of coordinates between the two viewing frames is the shift in the x-direction. The different between  $X_{left}$  and  $X_{right}$  is easily computed. The two simple matrix is easily applied to graphics APIs to form two different coordinate systems. Then calling the same drawing method with the same parameters will result in left viewing frame and right viewing frame in the different context respectively. This process can be described in the Figure 2.

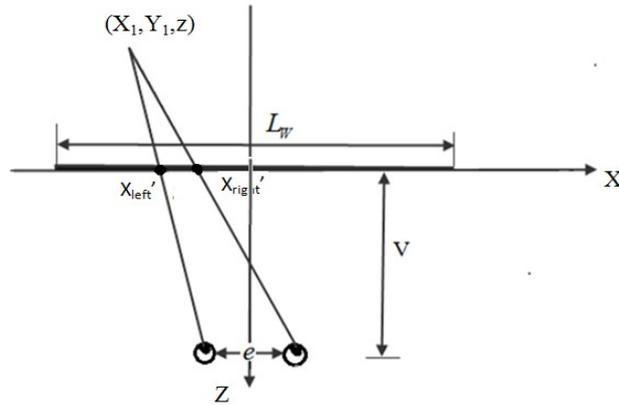


Figure 1. The model of stereoscopic rendering. The point  $(X_1, Y_1, z)$  is projected onto the screen at  $X_{left}'$  and  $X_{right}'$  which are perceived by left eye and right eye respectively.

$$\begin{pmatrix} X_{left} \\ Y_{left} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1000}{1000-z} & 0 & \frac{R_w z(-L_w + e)}{2L_w(1000-z)} \\ 0 & \frac{1000}{1000-z} & -\frac{zR_H}{2(1000-z)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (2)$$

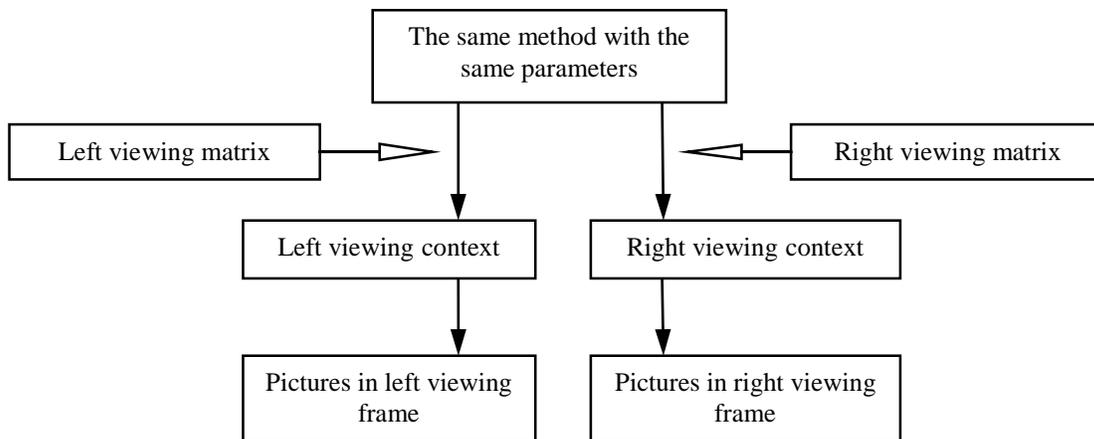


Figure 2. Applying the two matrix to form two viewing context. Different pictures are painted using the same parameters in the different viewing context.

$$\begin{pmatrix} X_{right} \\ Y_{right} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1000}{1000-z} & 0 & \frac{R_w z(-L_w - e)}{2L_w(1000-z)} \\ 0 & \frac{1000}{1000-z} & -\frac{zR_H}{2(1000-z)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (3)$$

## 2.2 Rendering S3D web page

The main function of a browser is to present the web resource by requesting it from the server and displaying it in the browser window. The browser's main components are shown in Figure 3 [11]. The responsibility of the rendering engine is display of the requested contents on the browser screen. The rendering engine will start parsing the HTML document and convert elements to DOM (Document Object Model) nodes in a tree called the "content tree". Then the browser constructs the render tree that is the visual presentation of the document. The purpose of this tree is to enable painting the contents in their correct order. In different browser kernel, the element in the render tree has different name. For example, Gecko calls the element in the render tree "frame" and Webkit call it "render object" or "renderer". We call the element renderer in this paper. After the construction of the render tree, it goes through a layout process that gives each node the exact coordinates where it should appear on the screen and exact size. The CSS 2.1 specification presents the CSS box model that describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model [12]. The last stage is painting—the render tree will be traversed and each renderer will be painted using the UI backend layer. UI backend is used for drawing basic widgets like combo boxes, buttons etc. This UI backend exposes a series of generic interfaces that are not platform specific. The browser engine marshals actions between the UI and the rendering engine. The detail about other components is described in [11].

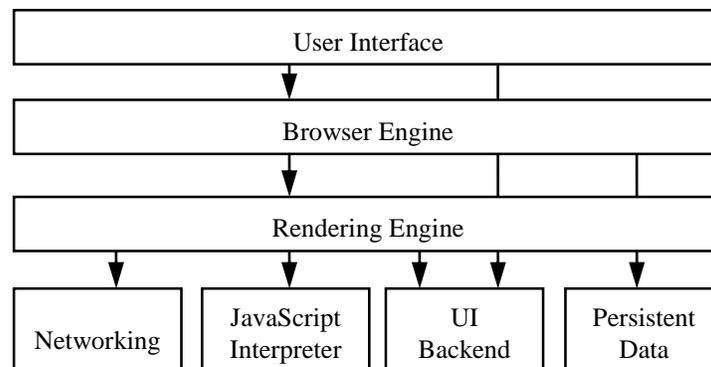


Figure 3. The browser's main components and their relationships.

In order to identifying CSS "depth" label, HTML parsing process should be modified to add depth information to the DOM tree. Because the render tree is constructed based on the DOM tree, depth information will be added to the rendering tree so that very renderer has at least a depth value. The depth information will be used in the painting stage, hit testing etc. In the painting stage, the browser calls the generic interfaces that UI backend exposes with the parameters about information of the rendered primitives. Then it'll call the specific library such as Qt [13], GTK+ [14], etc on a specific platform including Windows, Linux, and Macintosh to paint primitives. At last, the primitives are transmitted onto the monitor from the frame buffer.

For stereoscopic rendering, the two frame buffers for left viewing frame and right one should be created. The left viewing frame is painted into the left frame buffer at first and then the right one into the right frame buffer. So the render tree is traversed twice, first for left viewing frame and second for right one.

The process of stereoscopic rendering is presented below. The browser calls the generic interfaces with some parameters, such as the coordinate, size. Before the primitives are rendered into the different frame buffers, the left viewing matrix and the right viewing matrix deduced above are applied to the transformations. It results in the different context. If we paint a widget or primitive in the different context, the graphic library converts information of coordinate, size etc into new value for stereoscopic rendering. In addition, the matrix need extra information like screen size, resolution, a global data structure including stereoscopic setting parameters such as screen size, screen resolution, eye separation, and viewing flag that indicates which rendering viewing frame is currently painted are defined. The rendering cost of the two viewing frames will be at least double. At last, the two frames in the buffers are fused into one frame according the different property of the monitor. Then the fused frame is transmitted onto the screen.

For example, painting a simple button which is shown on the right in the Figure 4. We want to draw a button at the coordinate  $(x_0, y_0, z_0)$  with the size  $(w_0, h_0)$ . The browser firstly will call the generic interfaces that are not platform specific.

Before Qt in Linux paints the button, transformation performs using different viewing matrix and forms different context. The new coordinates and size in the left and right viewing frame are computed in Qt. Because of the same depth  $z_0$  of the rectangle region of the button, the size of two viewing frames is the same, too. After all the primitives are painted into the two viewing frame buffers, the two frames will be fused into only one for display based the property of the stereoscopic monitor. The last fusion can be implemented in some hardware to improve the performance.

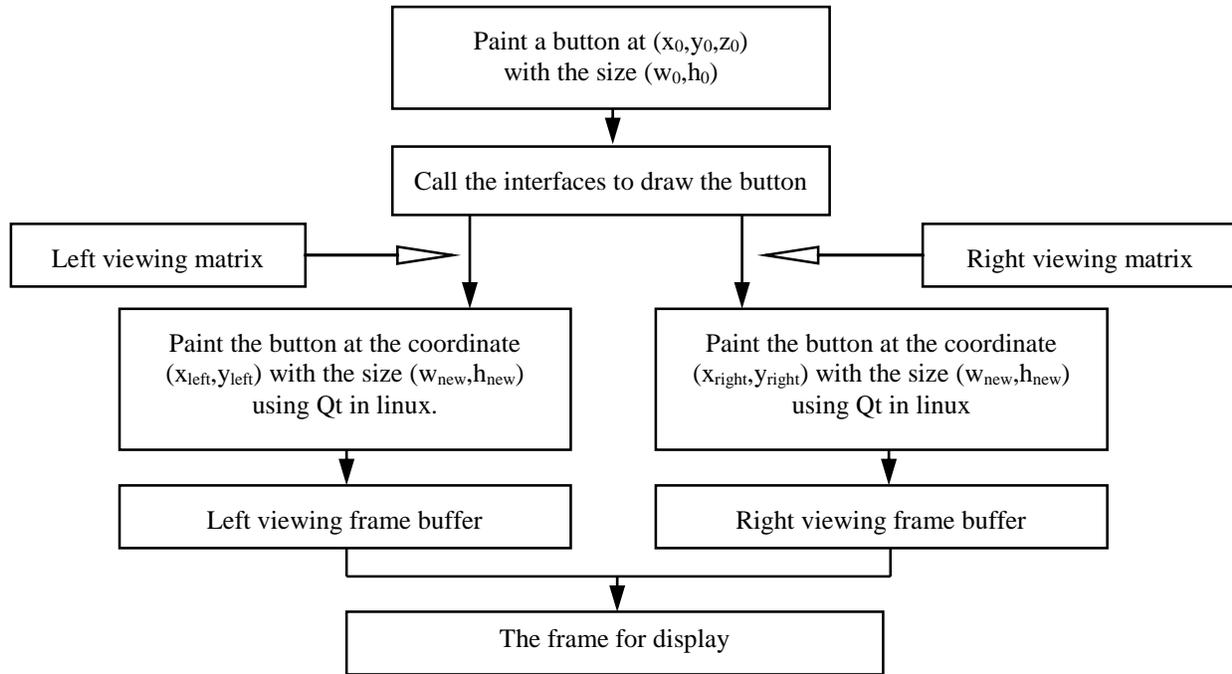


Figure 4. The basic process of painting stereoscopic primitives on the left and the process of painting a simple button on the right.

If a graphic library that can render stereoscopic widgets for display comes out in the future, the stage of transformation can be removed and the depth information is directly passed into the specific graphic library to paint stereoscopic images. It's simple and effective to modify the browser to render stereoscopic HTML web pages based on current framework through applying this method. However, some problems happen in this method. It will be discussed in the next section.

### 2.3 Repainting

The browser adopts event-driven architecture (EDA). An event can be defined as “a significant change in state”. When an event happens, it can be known within the architecture and triggers the message (called the event notification) emission. Then the event consumers have the responsibility of applying a reaction as soon as the event is presented.

In the browser engine, different consumers handle the messages of painting and repainting. When painting event happens, the browser paints the S3D web pages successfully. But if some regions change and should be repainted, the problem comes out. The case is shown in the Figure 5. The current 2D graphic library finds that the rectangle in black needs to be repainted and relevant generic painting interfaces are called. In fact, as shown in the Figure 5, the region in left viewing frame is painted in the rectangle in red and the one in right viewing frame in blue. So the browser doesn't repaint or erase the region in gray. If there is some useless information in the gray region, it'll mess the whole frame. In our method, we simply repaint the whole region of the frame, not only the region where changes happen, which is a waste of time, but fixes the problem. If there exists a graphic library that can paint stereoscopic widget in the future, this problem will not come out.

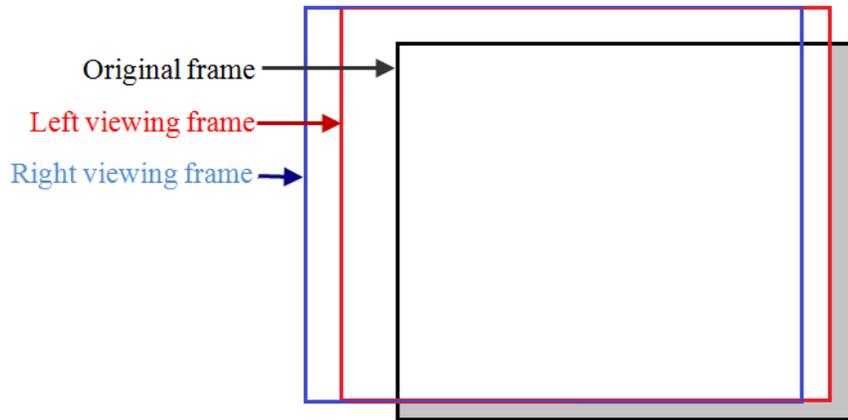


Figure 5. The gray region isn't repainted or erased when repainting happens.

### 3. EXPERIMENTAL RESULT

In this paper we implemented a web browser that supports stereoscopic rendering of pictures, text, widgets through using Qt graphic APIs in the Linux platform. Our experiment is based on Webkit, an open source web browser engine.

The difference of planar objects like pictures, text between left viewing frame and right viewing frame is just scale and shift. In theory, our eyes fuse the two frames and generate the sense of depth. But the eyes can't identify the depth information in some cases. For the web pages in which the contents are on the different layers, the stereoscopic display technology can enhance the effect. A simple demo is shown in the Figure 6 in which the three pictures have respectively depth of 6, 3, 0 from left side to right out of the screen. The screen adopts polarizing techniques. The odd rows are used to display left viewing frame and the even ones right viewing frame. The viewer who wears glasses which contain a pair of different polarizing filters can find the three pictures in different distance from his eyes clearly.

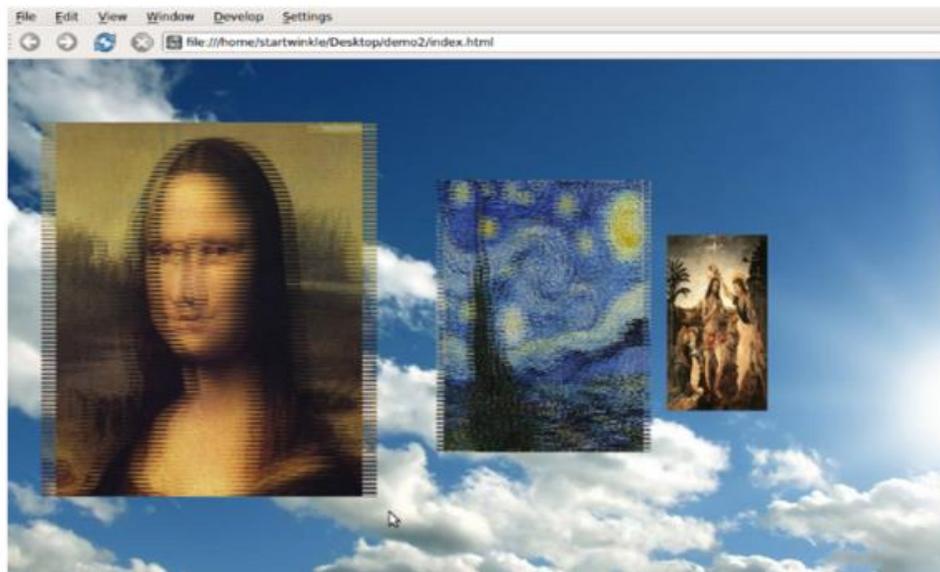


Figure 6. Pictures on the window in different depths that are 6, 3 and 0 from left side to right respectively.

## 4. CONCLUSION

Our experiment demonstrates that the combination of stereoscopic display technology and web technology can bring evolutionary visual effect. But our experiment also demonstrates that stereoscopic display of the planar objects doesn't have obvious sense of depth except the case that the planar objects give us an illusion that they are in different depth. However, the problem does not happen in the stereoscopic display of 3D models in the web. I believe with the development of 3D web visualization techniques, the application of stereoscopic display technology in the web will have large potentials.

On the other side, there are still some other problems to be fixed in this project. For example, when the link, text or button is clicked, the browser engine starts the hit-testing. Given a coordinate in the web page, the browser engine iterates the render tree and finds out the renderer that includes the coordinate. Due to the change of size and location of the renderer, the hit testing in the browser engine should be modified. However, if there exists a strong graphic library, all the problems described above can be effectively solved.

## 5. ACKNOWLEDGEMENTS

This project was supported by Shenzhen Special Foundation for Basic Research (JC201104210117A, JCYJ20120614150236998), by Shenzhen Engineering Lab on Intelligent Perception for Internet of Things, and by Shenzhen Engineering Lab of Three-dimensional Media Technology.

## REFERENCES

- [1] Chittaro, L. and Ranon R., "Web3D technologies in learning, education and training: Motivations, issues, opportunities," *Computers & Education*, 49(1), 3-18(2007).
- [2] Hui, L., Liu, E., Zhao, X., McFarlane, N.J. and Clapworthy, G. J., "Web-Based 3D Visualisation for Biomedical Applications," *Information Visualisation (IV)*, 2011 15th International Conference on, pp. 632-637. IEEE (2011).
- [3] Ortiz, S., "Is 3d finally ready for the web," *Computer*, 43(1) 14-16 (2010).
- [4] Lambooi, M.T, IJsselsteijn, W.A. and Heynderickx, I., "Visual discomfort in stereoscopic displays: a review," in *Stereoscopic Displays and Virtual Reality Systems XIV*, Proceedings of the SPIE-IS&T Electronic Imaging, SPIE Vol. 6490, pp. 64900I-01 to 64900I-13 (2007).
- [5] Bickerstaff, I., "Case study: the introduction of stereoscopic games on the Sony PlayStation 3," in *Stereoscopic Displays and Applications XXIII*, Proceedings of the SPIE-IS&T Electronic Imaging, SPIE Vol. 8288, pp. 828815-1 to 828815-14 (2012).
- [6] Woods, A.J., Docherty, T. and Koch, R., "Image distortions in stereoscopic video systems," in *Stereoscopic Displays and Applications IV*, Proceedings of the SPIE-IS&T Electronic Imaging, SPIE Vol. 1915, pp. 36-48 (1993).
- [7] Dunn, F. and Ian, P., [3D math primer for graphics and game development, 2nd edition], CRC Press, 218-228 (2011).
- [8] Leavitt, N., "Browsing the 3D Web," *Computer*, 9(9), 18-21 (2006).
- [9] Wang, W., Wang, X., Gao, W. and Liang, F. 2011. An Implementation of 3D Web Browser based on Binocular Parallax Theory. China Patent, ZL201110131554.1 (May 2011).
- [10] The World Wide Web Consortium, "Extensions for Stereoscopic 3D support," *Slate*, 30<sup>th</sup> November 2012, <[http://www.w3.org/2011/webtv/3dweb/3dweb\\_proposal\\_121130.html](http://www.w3.org/2011/webtv/3dweb/3dweb_proposal_121130.html)> (October 2013).
- [11] Garsiel, T. and Irish, P., "How browsers work: Behind the scenes of modern web browsers," *Slate*, 5<sup>th</sup> August 2011, <<http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>> (October 2013).
- [12] The World Wide Web Consortium, "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification," *Slate*, 7<sup>th</sup> June 2011. <<http://www.w3.org/TR/CSS2/>> (October 2013).
- [13] Qt Project, "Qt Project" *Slate*, 2013. <<http://qt-project.org/>> (February 2014)
- [14] The GTK+ Project, "The GTK+ Project" *Slate*, 2007. <<http://www.gtk.org/>> (February 2014)
- [15] Broberg, D. "Guidance for horizontal image translation (HIT) of high definition stereoscopic video production" in *Stereoscopic Displays and Applications XXII*, Proceedings of the SPIE-IS&T Electronic Imaging, SPIE Vol. 7863, pp. 78632F-1 to 78632F-11 (2011).