An Approach to Support Stereoscopic 3D Web

Qinshui Chen, Wenmin Wang*, Ronggang Wang, Jianlong Zhang, Zhongxin Liu

School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University
Lishui Road, Nanshan District, Shenzhen, China, 518055

iwater@sz.pku.edu.cn, *wangwm@ece.pku.edu.cn, rgwang@ece.pku.edu.cn

ABSTRACT

With the stereoscopic technology gradually maturing, realistic 3D devices are more and more popular. However, how to adapt current Web technologies to this booming trend still remains elusive. In this paper, a novel approach to support stereoscopic 3D Web is proposed that is to extend existing HTML, CSS and JavaScript languages. Under our stereoscopic 3D web model, each element in a web page can be assigned with a depth value, which is essential for the perception of 3D depth. WebKit is modified to accomplish a stereoscopic 3D web browser, and as the result turned out, each element of a web page can be seen in a truly 3D space on a stereoscopic 3D device that appears to be floating from the display surface. It is shown that this approach provided a new ground-breaking innovation for user experience.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation – *display algorithms*; I.3.8 [**Computer Graphics**]: Applications; I.4.9 [**Image Processing and Computer Vision**]: Applications.

General Terms

Design, Human Factors, Languages, Theory.

Keywords

Stereoscopic, stereoscopy, 3D, Web, browser, WebKit.

1. INTRODUCTION

Research on the 3D Web has long been investigated, with popular technologies such as VRML, X3D, WebGL and O3D et al [1]. However, these technologies mainly fall into the category of monoscopic 3D (M3D) – traditional 3D computer graphics that create images based on a 3D coordinate system and then display these images onto a flat 2D device. With the stereoscopic technology gradually maturing, what we have been doing since the beginning of 2011 is to apply stereoscopy to the Web based on binocular parallax theory [2]. In this paper we will introduce the basic principles of stereoscopy, and discuss how they can be used in the Web. Based on these principles, we will propose a new set of client-side Web techniques – HTML-S3D, CSS-S3D and JavaScript-S3D for descriptions of stereoscopic web pages. A 3D web browser based on the open-source WebKit project – a popular render engine for browsers – are also implemented for demonstration purposes.

2. STEREOSCOPIC 3D WEB PRINCIPLES

2.1 Binocular Parallax

The brain's ability to appreciate a third dimension using a 3D display is based on the characteristics of the human visual system. The visual system receives two views of the visual scene, i.e., the left- and right-eye views, which largely overlap but differ slightly because they originate from two different perspectives. The human brain extracts the depth information and perceives the stereoscopic scene by fusing these two images acquired by the eyes.

The same object in the scene will have different relative positions in the two views. This difference in relative position is typically called horizontal parallax, which can be divided into three classes – positive, zero and negative parallax. When displayed stereoscopically, those objects with zero parallax will be depicted at the screen plane, those with negative parallax will appear in front of the screen, and those with positive parallax will be perceived behind the display (Figure 1).

2.2 From Object Space to Image Space

This section will describe the mathematical relationship between objects, how they are perceived in the three-dimensional space (object space) by the viewer, and how they are drawn in the left- and right-eye images (image space).

Assume we are in a left-handed coordinate system, and the 3D screen lies on the x-y plane, as shown in Figure 2.

A standard projection with its projection center lies on the z axis at (0, 0, -V) can be described by this matrix M:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{V} & 1 \end{bmatrix} \tag{1}$$

We couldn't use the M matrix for the left- and right-eye projections directly because their projection centers do not locate at (0, 0, -V). But they can still be achieved using a trick of translations. For an arbitrary point P = $(X,Y,Z,1)^T$, $P_l = (X_l,Y_l,Z_l,1)^T$ and $P_r = (X_r,Y_r,Z_r,1)^T$ are the corresponding projected points for the left- and right-eye images (Figure 2).

SAC'14, March 24-28, 2014, Gyeongju, Korea.
Copyright 2014 ACM 978-1-4503-2469-4/14/03…$15.00.
http://dx.doi.org/10.1145/2554850.2555096

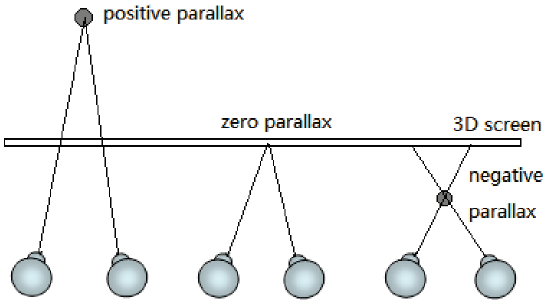


Figure 1. Three types of parallax

If we move P and P_l off the x axis by +e/2 units, then the translated P_l is the projection result of the translated P using matrix M:

$$TP_l = M(TP) \tag{2}$$

Where T is the translation matrix for points to move along the x axis by +e/2 units:

$$T = \begin{bmatrix} 1 & 0 & 0 & \frac{e}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Thus,

$$P_l = T^{-1}MTP \tag{4}$$

Replacing M and T in Equation (4) with Equations (1) and (3) results in:

$$P_l = \begin{bmatrix} X - \frac{eZ}{2V} \\ Y \\ Z \\ 1 + \frac{Z}{V} \end{bmatrix} \tag{5}$$

It's rather similar for point P_r:

$$P_r = TMT^{-1}P = \begin{bmatrix} \mathrm{X} + \frac{eZ}{2V} \\ Y \\ Z \\ 1 + \frac{Z}{V} \end{bmatrix} \tag{6}$$

Scrutiny of Equations 5 and 6 results in two straightforward but valuable observations:

1) Screen Parallax p:

$$p = X_r - X_l = \frac{eZ}{V+Z} \tag{7}$$

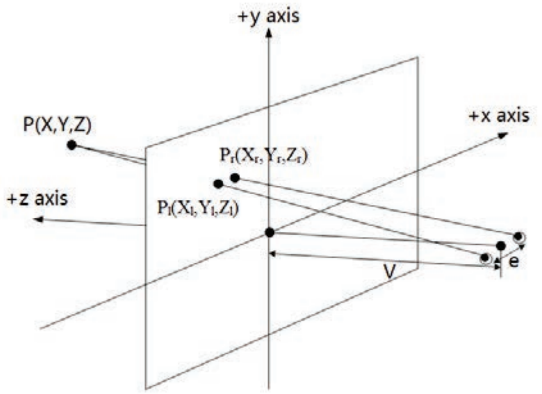


Figure 2. 3D display system with associated parameters

The value of p is essential for the stereoscopic effect. Stereoscopic displays generally exhibit accommodation-convergence (AC) mismatch where the eyes converge at the apparent point of fixation in the image but focus on the screen [3]. If p is too large, this AC mismatch will be too significant, causing visual discomfort; if p is too small, the stereoscopic effect is lost.

2) P_l and Pr have the same y coordinate value, which means we can pan each stereoscopic element in the left-eye view by its screen parallax to get the right-eye view. The size of each stereoscopic element is also scaled, making elements higher on the z axis (further away to the viewer) appear smaller, and those closer to appear larger. The scaling is proportional to:

$$ratio = \frac{V}{V+Z} \tag{8}$$

3. DESIGN AND IMPLEMENTATION

3.1 Extension of W3C Web Standards for Stereoscopic Web Page

Current W3C standards for the Web have not provided a way to describe a stereoscopic 3D scene. Therefore we propose a new set of client-side Web techniques – HTML-S3D, CSS-S3D and JavaScript-S3D, which are supersets of current W3C web standards.

3.1.1 HTML-S3D

A "view" HTML tag is invented to specify a pair of resources for the left- and right-eye view (Table 1). Although we only use this tag for stereoscopic images currently, other resource types could also be included if necessary.

Table 1. <view> HTML tag

Attribute	Value	Description
type	image	Resource type
left-src	URI	The location of the resource for the left-eye view
right-src	URI	The location of the resource for the right-eye view

Table 2. CSS-S3D properties

Attribute	Value	Default	Description
depth	length	0	Stereoscopic depth perceived by the user.
parallax	length	0	Screen parallax.
eye-distance	length	0.035V	The distance between the two eyes.

Table 3. Possible values for Screen.3DFormat

Value	Description
SbS	3D format “side by side”.
TaB	3D format “top and bottom”.
Mono	2D format monoscopic.

3.1.2 CSS-S3D

Three new properties are introduced in CSS-S3D for web authors to control the stereoscopic 3D scene (Table 2).The “depth” property is introduced to assign stereoscopic depth information to elements; the “parallax” property is used to assign a screen parallax; and the “eye-distance” allows the user to choose an appropriate “e” parameter as mentioned in Section 2.2.

The “depth” property is specified in units of the view volume (the V parameter) and should be assigned with a value between -1 and +1. The value of depth (d) is used to compute an element’s screen parallax (Equation 7, Z = dV). It also causes elements that have negative Z coordinates (closer to the viewer) to be scaled up, and those further away (positive Z coordinates) to be scaled down (Equation 8); these effects are applied after elements have been sized and positioned according to the CSS visual formatting model [4]. Figure 3 shows what an element with a positive “depth” property may look like when displayed on a 3D screen. Note that it have some area for the left (or right) eye only, and both its width and height are scaled. We can also see that this stereoscopic element has the same upper-left corner position for the original view and the left-view, and this will be explained latter.

The “parallax” property is useful when the user wants a stereoscopic element without its size being zoomed (with no “depth” property), or when he wants to overwrite a default-computed parallax (working with a “depth” property). A bigger parallax value may result in a more significant stereoscopic effect, but too big a value will cause severe visual fatigue. Negative values, which will make the element appear in front of the screen plane, are also allowed.

The “eye-distance” property is essential to strike a balance between the AC mismatch and the stereoscopic effect. A small value is preferred, and our implementation chooses e = 0.035V, while V = 61cm.

3.1.3 JavaScript-S3D

A “3DFormat” property is added to the “Screen” global object for queries of 3D screen information. It tells the user whether the screen supports stereoscopic 3D, or what kind of 3D display it is.

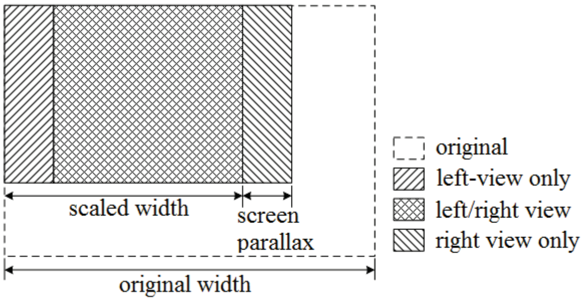


Figure 3. An element with a positive “depth” property

3.2 Implementation on WebKit

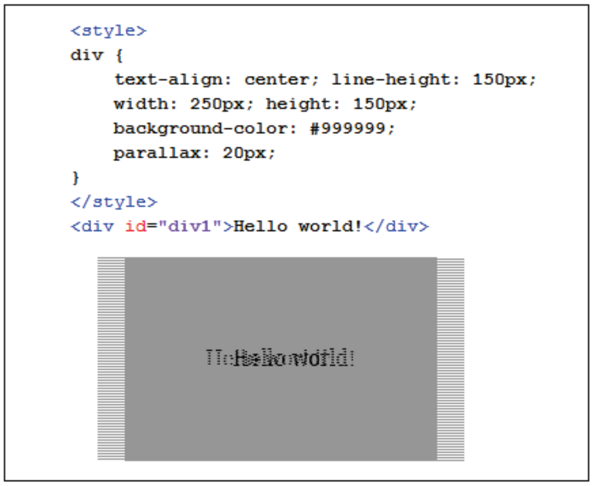
WebKit is one of the most popular browser engines in the world. It is a layout engine software component designed to allow web browsers to render web pages, powering famous browsers like Google’s Chrome and Apple’s Safari [5]. Corresponding browser engines includes the Gecko (used by Mozilla Firefox) and Trident (used by IE). We adopt WebKit for its notable features such as being open-source, high speed, less memory, etc.

The main flow of WebKit mainly includes DOM tree constructing, CSS rules parsing, render tree constructing and painting [6]. Adding HTML-S3D tags and CSS-S3D properties to WebKit doesn’t have to change its original working flow.

The most important change we made to WebKit is a re-painting procedure. As mentioned in section 2.1, the generation of stereoscopic depth requires two views, each for one eye. In other words, two visible frames must be rendered for a single stereoscopic web page. Therefore, we create two frame buffers for the rendering process. While painting the left-eye frame buffer, we don’t change the layout of each element, but stereoscopic elements are properly zoomed; to get the the right-eye frame, we only need to pan those stereoscopic elements in the left frame according to Formula 7. This is the reason why in Figure 3 the left-view and the original view share the same position for its upper-left corner. Finally, these two frame buffers are combined on the 3D screen.

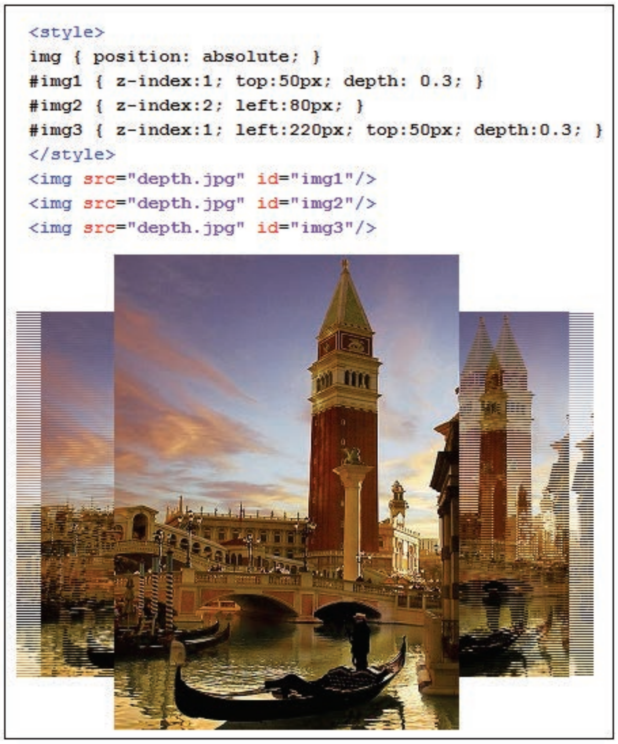
4. EXPERIENTIAL RESULTS

We have implemented a web browser based on the open-source WebKit project to display stereoscopic 3D web pages. Examples are shown in Figure 4 and Figure 5. Figure 4 shows the usage of the “parallax” property. The “div” element in this figure is painted to the left- and right-eye frame using its original size, and a positive screen parallax is added, which makes it floating behind the screen. Figure 5 is an example of the “depth” property. The image in the middle part has a default zero “depth” value, and is painted to a block without a parallax on the screen; the images on the two sides, however, both have a size shrinkage and are observed behind the screen, as they have a positive depth value.



```
<style>
div {
    text-align: center; line-height: 150px;
    width: 250px; height: 150px;
    background-color: #999999;
    parallax: 20px;
}
</style>
<div id="div1">Hello world!</div>
```

Figure 4. Example for the “parallax” property



```
<style>
img { position: absolute; }
#img1 { z-index:1; top:50px; depth: 0.3; }
#img2 { z-index:2; left:80px; }
#img3 { z-index:1; left:220px; top:50px; depth:0.3; }
</style>
<img src="depth.jpg" id="img1"/>
<img src="depth.jpg" id="img2"/>
<img src="depth.jpg" id="img3"/>
```

Figure 5. Example for the “depth” property

On a stereoscopic 3D display, we can see that element in the stereoscopic 3D web page is displayed in a truly 3D space that looks like to be floating from the display surface.

Our browser is compatible with current 2D web pages. All existing websites can be displayed on this browser, but in a planar form.

The hardware used in our implementation is Intel Core i5-3470 3.20GHz, 8.00 GB RAM. Our browser is a modification of the Qt-port WebKit-r117946, built on a 64-bit Linux Mint box.

5. CONCLUSIONS

In this paper, we have presented a novel approach to support stereoscopic 3D Web by extending existing HTML, CSS and JavaScript languages. We discussed the underlying principles of stereoscopic 3D and how these principles can be applied to the Web. We also extended current web languages for the stereoscopic 3D web. With HTML-S3D, CSS-S3D and JavaScript-S3D, users are able to add stereoscopic depth to an element easily.

There are still several issues remaining to be done. First, some parameters such as e and p, play an important role in the stereoscopic view. Appropriate values (or value ranges) should be studied to achieve more desirable visual effects. Second, although using properties in CSS-S3D we can easily turn an element into a stereoscopic one, the front-to-back rendering order of this element won’t be changed so it doesn’t interact with other elements. Take Figure 5 for example, the middle image will be painted behind the image on the left side if it doesn’t have a bigger “z-Index”, although it has a smaller “depth” property (closer to the viewer). Moreover, as we can see from Equation 5 and 6, an element’s painting position is relevant to its (X, Y, Z) position in the 3D coordinate system and the viewer’s observing angle, which we don’t all take into account when dealing with the “depth” property. The web page authors may want to create a stereoscopic 3D rendering context, in which all the elements share the same global 3D rendering parameters and are all ranged in the same 3D scene. Third, all the perceived stereoscopic elements are parallel to the screen plane. It still needs a further investigation about how to apply 3D transforms to the stereoscopic view.

6. ACKNOWLEDGMENT

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.

7. REFERENCES

[1] Ortiz, S. Is 3d finally ready for the web? *Computer*, *43*, 1 (2010), 14-16.

[2] 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 20, 2011.

[3] Urey, H., Chellappan, K. V., Erden, E., and Surman, P. State of the art in stereoscopic and autostereoscopic displays. In *Proceedings of the IEEE, 99, 4 (2011), 540-555*.

[4] W3C. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. http://www.w3.org/TR/2011/REC-CSS2-20110607/.

[5] The WebKit Open Project. http://www.webkit.org/.

[6] Tali, G. and Paul, I. How Browsers Work: Behind the scenes of modern web browsers. http://www.html5rocks.com/en/tutorials/internals/howbrowserswork