Computer Standards & Interfaces 50 (2017) 65–75

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

CSS3 extensions for setting web content in a 3D view volume and its stereoscopic 3D display

Soon-Bum Lim, Hee-Jin Lee, Wen-Yan Jin, Seung-Min Shim*

Dept. of Multimedia Science, Sookmyung Women's University, Cheongpa-ro 47-gil 100, Yongsan-gu, Seoul 04310, Republic of Korea

ARTICLE INFO

Article history:
Received 10 September 2015
Received in revised form
30 April 2016
Accepted 5 June 2016
Available online 7 June 2016

Keywords:
Web contents
3D
View-volume setting
Stereoscopic
Preprocessor
HTML5
CSS3

ABSTRACT

Over recent years, the development of electronic devices such as mobile devices and smart TVs has been remarkable; meanwhile, the emergence of 3D TV has given rise to a rapid increase of the demand for 3D expressions. The objective of this study is the use of a CSS stylesheet for the reconstruction of the web content that is typically expressed in formats like HTML into 3D space. To this end, the basic information that is required for the defining of the 3D graphics of 3D space must first be identified, and the remaining information that cannot be covered by the genuine CSS3 specifications can be set in the view volume through the extension of the 3D-viewpoint and 3D-view-volume specifications; therefore, the specifications were extended so that the contents of the 3D view volumes can be placed at any user-selected location. A preprocessor that can convert the genuine specifications into the extended specifications could then be implemented using the javascript, enabling a viewing of the extended-specifications contents on the current browser. A rendering-engine emulator was also implemented for the checking of the demonstration results and for the evaluation of the stereoscopic-display expression that is one of the various 3D-content-application fields. If the extended CSS3 specifications that are proposed by this study become the W3C standards, the 3D expression of web content will become more unconstrained and more convenient.

1. Introduction

Thanks to phenomenal technology advances over recent years, 3D-related studies are being actively carried out in a variety of fields including 3D video/imaging, 3D display, and 3D printing. The ongoing emergence of 3D TV, HMD etc. is especially significant, and is accompanied by an increasing number of studies regarding the way that additionally-provided data is expressed with browser-based declarative contents. Languages like XML, HTML, and CSS are often used as browser-based declarative contents that can be described independently of the platform and regardless of the device type if the Web standard is used.

HTML5 and CSS3 are browser-based and declarative-content technologies that are the most visually arresting; in particular, CSS supports various profiles in consideration of different device environments. For example, the CSS TV Profile is defined in consideration of the operational requirements and the limitations of TV devices, and it is used for the expression of browser-based TV content. The current CSS TV-profile specifications were, however, defined before the emergence of CSS3, so the 3D-space concept was not included; also, the HTML5- and CSS-related specifications that have been or are being developed do not contain contents that can be projected in 3D space and expressed as a stereoscopic 3D image. Therefore, the research to express the received contents, including not only the declarative contents but also web contents from TV, HMD and various 3D devices and express them into 3D stereoscopic by placing them on the 3D space is required.

Given the existing gaps, the genuine CSS3 specifications were extended in previous studies for the defining of the 3D space, whereby the projection of HTML5 contents in 3D space offers a way to express the contents in a stereoscopic-3D-display Web environment. An emulator for a stereo-projection mode was also implemented along with these studies, followed by an evaluation of the demonstration results.

The following limitations, however, hamper the results of the previous studies: An in-depth study regarding the extended view-volume specifications was not conducted, and a verification of the preprocessors and emulators was not sufficiently completed. This study therefore extended the CSS3 specifications with a focus on the view-volume setting and implemented a preprocessor and an emulator, so that analysis results that compensate for the shortcomings of the previous studies could be produced. The specific goals of the present study are as follows:

* Corresponding author.
E-mail address: undervine@naver.com (S.-M. Shim).

http://dx.doi.org/10.1016/j.csi.2016.06.001
0920-5489/

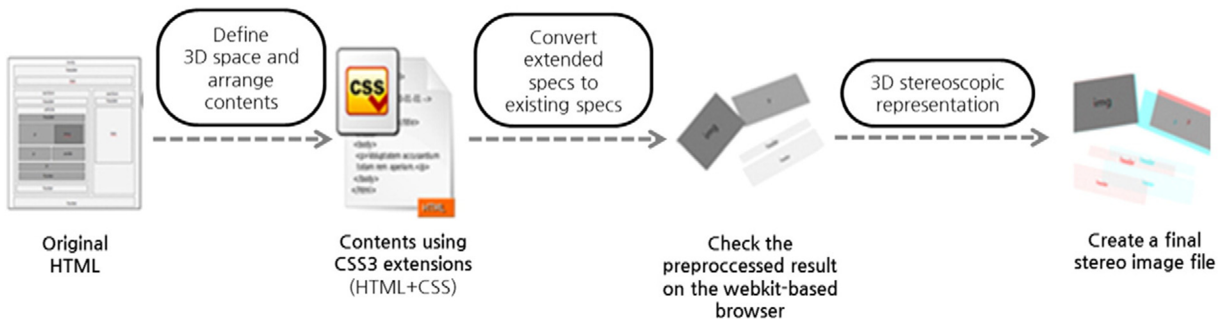

Fig. 1. Research objective.

- Design of CSS3 specifications: The extended CSS3 specifications that are necessary to set the view volumes were proposed so that users could freely define the 3D space. Also, the properties that are used for the placement of the contents in the view volume and the expression of the contents in the stereo mode are suggested.
- Implementation of preprocessor and analysis of execution results: The preprocessor was implemented to translate the extended properties into the genuine properties, making them visible on the current browser. The sample contents were written and used to evaluate the accuracy of the preprocessor-execution results.
- Implementation of emulator and analysis of execution results: When the projection mode was set as "stereo," the rendering-engine emulator was implemented for the checking of stereoscopic images. Also, to verify the system, users were shown stereo images that were displayed on a 3D TV for the evaluation of the execution results (Fig. 1).

2. Related research works

Although a variety of 3D graphic-file formats such as VRML and X3D are currently available, they are not fully utilized due to a lack of 3D interaction-related technology for hypertext-based web interfaces. To solve this problem, Jankowski and Decker [1] introduced a 3D dual-mode user interface that enables users to switch from hypertext to graphic data, and vice versa; here, users can choose one of two interface modes, and Fig. 2 shows that the dual-mode user interface allows the same information to be expressed in two modes.

Next, declarative languages such as X3DOM and XML3D, which use major web technologies including HTML, CSS, and Javascript to describe interactive 3D content, are used to mix 3D-scene definitions and standard HTML markup so that 3D content becomes part of the document object model (DOM); however, both XML3D and X3DOM support CSS only in a 3D-transformation module. But, without the support of full CSS integration, the styles and structures of the DOM can become entangled. Sutter, Sons, and Slusallek [2] conducted a study on a CSS-integration model for the 3D expression of declarative contents, and described how CSS is utilized for the definition of materials, lights, and cameras. Fig. 3 shows an example where a camera is defined by CSS.

With the development of a variety of technologies for 3D expressions on the Web, it has become possible to see 3D web content on web browsers through the application of Web3D and WebGL; however, as the standard web browser is basically a 2D display-based device, only a single view is offered. Wang, Dong, et al. (4 persons) [3] conducted a study regarding the utilization of the stereoscopy principle on the Web, whereby new languages such as HTML-S3D, CSS-S3D, and Javascript-S3D are introduced for the representation of stereoscopic web pages; furthermore, they implemented a stereoscopic 3D web browser for the rendering of written web pages. Fig. 4 shows the demonstration results of the sample web pages that were coded with HTML-S3D, CSS-S3D, and JavaScript-S3D.

Apart from these research developments, the W3C is currently drafting standard extensions that support stereoscopic 3D expression [4]. Table 1 shows a summary of the CSS extensions that were written for such a purpose.

Although there are a variety of 3D-related studies, as examined above, other studies have concerned only a part of the 3D. However, this study includes overall process from setting View Volume to creating an overall 3D concept on the web to make it available

Fig. 2. Integration of hypertext and 3D-data information.

```
#camera {
    camera-model: url("urn:camera:perspective");
    --vfov: 61deg;
    --clip-planes: 0.1 auto;
}
```

Fig. 3. Using CSS for the defining of a camera.

Fig. 4. Web page for stereoscopic 3D-TV GUI for which HTML-S3D, CSS-S3D, and JavaScript-S3D are used.

Table 1
Supporting extensions for stereoscopic 3D.

Contents	CSS properties
Extension of CSS 3D transforms	Perspective-baseline
Stereo properties for the specification of stereo-3D content	Stereo-content, stereo-render-option, stereo, stereo-size-type, stereo-order-type, stereo-format
Responsive contents	Non-normative
Media type "3D display" for media query	Non-normative
Display interface (device API for stereo rendering)	Non-normative
3D graphics such as WebGL, 3D SVG	Non-normative

from upcoming 3D devices.

3. Extended CSS3 specifications for 3D expression of content on the web

The CSS 3D-transforms module is used for the placement of elements in 2D or 3D space, and it encompasses the specifications of CSS 2D transforms, CSS 3D transforms, and SVG transforms, all of which are necessary for the editing of web content. The CSS 3D-transforms module currently offers specifications for the definition of the 3D transforms of objects, but because it has no 3D-space concept, specifications do not exist for the defining of 3D space; therefore, this study proposes the extended specifications for the definition of 3D space. In addition, the study suggests properties for the placement of the defined contents in 3D space and to enable the selection of the stereo mode.

3.1. Extended specifications for view-volume setting

To set the view volume, information regarding the camera setting, lens setting, and clipping setting is required [7]. The view-volume setting can be divided into the view-point setting and the view-space setting.

3.1.1. 3D-view-point setting

The view point in 3D graphics indicates the position from which an object is being observed. For the 3D-view-point setting, values must be assigned to the following variables: view point, which is the camera position; view direction, which is the direction the camera is pointed at; and the up-vector variable, which is the gradient of the camera [7].

Table 2
CSS3 extension for the setting of the 3D view point and related CSS3 properties.

Basic information for the setting of the 3D view point	Related CSS3 properties	Extended CSS3 properties
View point (x, y, z)	Perspective-origin: x y; perspective: <length>;	-3dLayout-perspective-origin: x,y,z
View direction (x, y, z)	fixed (0,0,1)	-3dLayout-view-direction: x,y,z
Up-vector (x, y)	fixed (0,1,0)	-3dLayout-up-vector: x,y

In the current CSS3, the camera's position is established by the setting of the x and y coordinates with the perspective-origin property of the 3D-transforms module, and by setting the z coordinate with the perspective property; however, the problem here is that the user is not allowed to control the pointing direction and the gradient of the camera because they are artificially fixed. This study therefore proposes extended CSS3 specifications so that users can freely control the view point, as shown in Table 2.

As users were previously allowed to set only the x and y coordinates, only the vertical and horizontal points of a position where the viewpoints were to be placed could be set. The extensions, however, allow users to set the x and y coordinates that are of the -3dLayout-perspective-origin property through the use of the vertical and horizontal values of the camera position, and the z coordinate can be set with the use of the distance between the camera and the view window. Also, the -3dLayout-view-direction allows users to freely set the x, y, and z coordinates of the camera's view direction that were previously fixed to the z direction. The availability of a property that can set the x and y coordinates of the up-vector here means that it is possible to express the effects of images that are taken with a camera lying on the back.

3.1.2. 3D-view-space setting

The setup of a view space involves the establishment of the space domain within the window of a view plane wherein the projection of an object is based on view-point-setting information such as the fovy, aspect, near clipping plane, and far clipping plane. In the current CSS3, the distance between the camera and the view window can be expressed using the perspective property; however, as the other basic information is fixed to the screen size, users cannot control the view space. To solve this problem, the present study added the extended specifications that are shown in Table 3.

The fovy parameter indicates a viewing angle that is expressed in the direction of the vertical y axis within a range of 0° to 180°; using this angle, a user can zoom in and out in relation to an object image. The aspect is a value that can be calculated through the

Table 3
CSS3 extension for the setting of the 3D view space and related CSS3 properties.

Basic information for the setting of the 3D view space	Related CSS3 properties	Extended CSS3 properties
View angle (a_w, a_h) near, far clipping distance(d_n, d_f)	Fixed(screen size)	-3dLayout-perspectivevolume: <fovy> <aspect> <near> <far>;
Convergence distance (d_c)	Perspective: <length>;	–

Table 4
CSS3 extension for placing web content in 3D view volume.

Category	Related CSS3 properties	Extended CSS3 properties
Absolute positioning	TranslateZ (< translation-value >)	-3dLayout-depth: < length > ;
	None	-3dLayout-relative-positioning: x y z;
Relative positioning	z-index:	-3dLayout-z-depth:
	None	-3dLayout-SetbasePosition: x y z;
	None	-3dLayout-positioning: x y z;

division of the height into the width of the view window. As the near- and far-clipping distances are limited to the range of the view space, only those objects that are positioned within this limited space are projected into the view plane; consequently, users can freely control the view space that was initially fixed to the size of the previous screen.

3.2. Placement of contents in 3D view volume

As the multimedia content of the current web pages are designed to match the same z value, the setting of the z value, which is required for the placement of 2D objects in a 3D view volume, cannot occur. To solve this problem, this study proposes the following two input methods: the input of z values in the "real value + unit" form, and the z-value input that is required for the placement of an object at a relative distance from the previously positioned layer.

Regarding these proposed methods, -3dLayout-depth can input z values in the "real value + unit" form, while, in contrast, -3dLayout-z-depth is a property that is used to place an object at a relative distance from the previously positioned layer. After the selection and setting of a layer as the standard among the parent layers, each object must be set with the -3dLayout-relative-positioning property to set the relative position from the previously positioned 3dLayout-SetbasePosition, and the object will then be moved in the z direction as far as the input value through the use of the 3dLayout-z-depth property. Also, users can simultaneously and easily input the x, y, and z coordinates of the position where an object is placed through the -3dLayout-positioning property, and they can also easily set the horizontal (x), vertical (y), and z coordinates of a position where the concerned object is placed (Table 4).

3.3. Setting of stereoscopic mode

The value of the genuine "transform-style" property was added so that users can choose between the following two projection options for the web content that has been written with the above-mentioned extensions: original format or stereoscopic space. If users set the property value of the transform style as "3dstereo," they can see the content in either 3D space or a stereoscopic format (Table 5).

Table 5
CSS3 extension for projection-mode setup in 3D view volume.

Related CSS3 properties	Extended CSS3 properties
Transform-style: flat/preserve-3d;	transform-style: 3dstereo;

4. Implementation of preprocessor for the translation of the genuine properties into the extended CSS3 properties

As the extended CSS3 specifications that are proposed in this study are not of the W3C standards, they cannot be represented by the existing browser; therefore, a preprocessor that allows users to see the setting results of the sample-content view volumes that were prepared using the extended CSS3 properties was implemented for this study. The system requirements of the preprocessor are shown in Fig. 5. If users add and execute the preprocessor file for the content that is written with HTML and CSS3 extensions, the preprocessor will translate them into CSS 3D-transforms modules to create a new CSS file that is executable on the browser.

4.1. Translation process of extended properties

Fig. 6 shows a diagram of the translation process of the preprocessor. First, after reading a CSS-file line that is linked by a user, the preprocessor checks whether the extensions exist; if the line belongs to the translation process of multiple properties (4.2.2), the preprocessor stores it and then translates it together with the previously stored values. If, however, the CSS-file line does not belong to the translation processes of multiple properties, it is directly translated (4.2.1). If duplicated extension properties exist

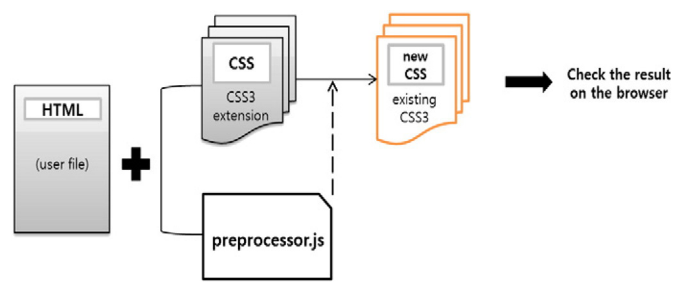

Fig. 5. System overview of preprocessor.

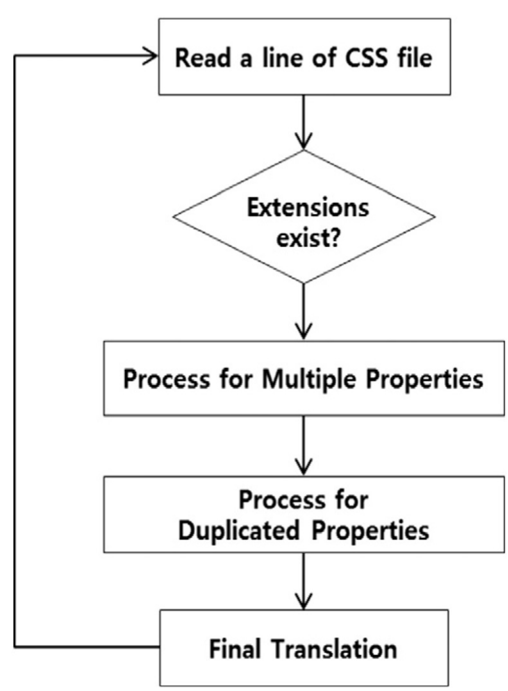

Fig. 6. Flowchart of preprocessor.

Input (extended CSS3 properties)	Output (genuine CSS3 properties)
-3dLayout-perspective-origin : x y z;	-webkit-perspective-origin : x y; -webkit-perspective : z;
-3dLayout-view-direction : x, y, z;	-webkit-transform: translateZ(<perspective-d>) rotateX(-θ_x) rotateY(-θ_y) translateZ(-<perspective-d>);
-3dLayout-up-vector : x, y;	-webkit-transform: translateZ(<perspective-d>) rotateZ(-θ_{up}) translateZ(-<perspective-d>);

Fig. 7. Translation of view-point setting.

before the final translation, the latest input-property values are used for the line translation (4.2.3).

4.1.1. Translation process of single property

The current CSS3 can be applied to objects, but it cannot be directly represented in 3D space. By using the translations of objects, it is possible to obtain the same effect as that from the setup of view volumes.

(1) View-point setting

The view-point setting can be divided into the view-direction setup and the up-vector setup. These extended properties are used for the designation of the position of a viewpoint that has been translated into the perspective origin and the perspective property of the CSS3 3D-transforms module; the former is comprised of the x and y coordinates of the view point, while the latter is the z coordinate (Fig. 7).

Next, the extended properties that were used to set the view direction are translated into the rotate X(), rotate Y(), and translate Z() properties. First, the up-and-down vertical rotation of a camera is expressed as rotate X(); herein, indicates the vertical rotation angle on the x axis.

$$\text{Vertical rotation:}\theta_x = \arctan\left(\frac{y}{z}\right) \tag{1}$$

$$\text{Horizontal rotation:}\theta_y = \arctan\left(\frac{x}{z}\right) \tag{2}$$

The left-and-right horizontal rotation of the camera is then expressed as rotate Y(), and θ_y indicates the horizontal rotation angle of the camera on the y axis. For your reference, as the camera and objects were moving in opposite directions, the values of θ_x and θ_y are negative; however, rotate X() and rotate Y() rotate on their own axes. To rotate an object according to a specific angle, translate Z() was therefore used in this study to rotate the object around the radius, which is the distance between the camera and the view plane, by an angle of < perspective-d > that is entered by the user; the object is consequently reverted to the original position. Fig 8 (a) shows the moving path of an object that is a simulation of the effect of the rotation of the camera to the left or right.

Lastly, the extended properties for the up-vector setup are translated into the rotate Z(< deg >) properties of the CSS3 3D-transforms module.

$$\theta_{up} = \arctan\left(\frac{x}{y}\right) \tag{3}$$

Herein, θ_{up} is the property value of rotate Z(), which is used to set the rotation angle on the z axis, or the up-direction of the camera. As the camera and objects are moving in opposite directions, the value of θ_{up} is negative.

(2) View-space setting

The view space was set through the use of the scale property and the visibility: hidden property value, as follows:

$$\text{scale}_y = \cot\left(\frac{<fovy>}{2}\right) \tag{4}$$

$$\text{scale}_x = \frac{\text{scale}_y}{<aspect>} \tag{5}$$

If a person remains in the same position, the value of < fovy > is 90°. If the value of either < near >, a distance to the near clipping plane, or < far >, a distance to the far clipping plane, is negative, an object deviates from the directions of the default camera lens so that nothing is portrayed in the view; therefore, the set values of both < near > and < far > must be positive. Also, the value of translate Z, which was used to express the z values, is out of the range of < perspective-d > – < far > < translateZ > < perspective-d > – < near >, the hidden visibility property that is

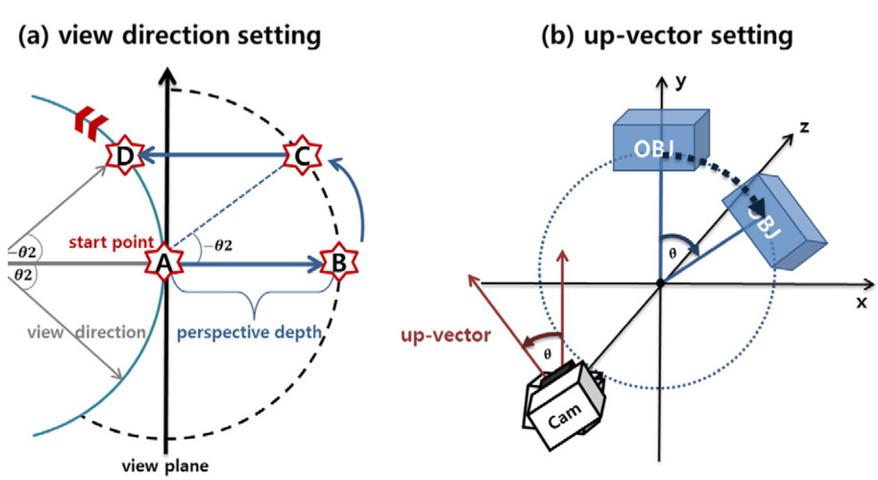

Fig. 8. Moving path of the object.

Input (extended CSS3 properties)		Output (genuine CSS3 properties)
-3dLayout-perspectiveVolume: <fovy> <aspect> <near> <far>;	⇨	scale(scale$_x$, scale$_y$); visibility:hidden;

Fig. 9. Translation of view-space setting.

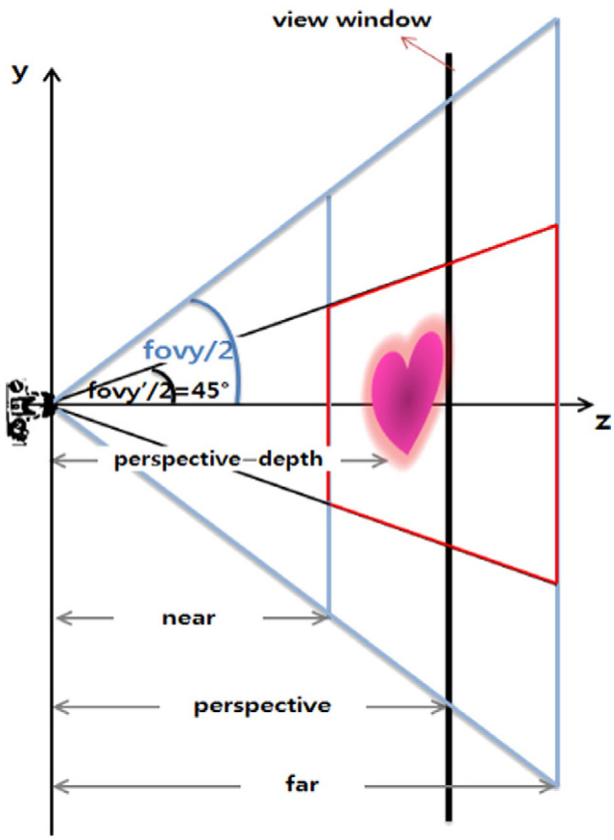

Fig. 10. CSS3-based view-volume setting.

Input (extended CSS3 properties)			Output (genuine CSS3 properties)
Absolute	-3dLayout-depth:<length>;	⇨	-webkit-transform: translateZ(z);
	-3dLayout-relative-positioning: x y z;		left:x; top:y; -webkit-transform: translateZ(z);
Relative	-3dLayout-z-depth:		z-index:
	-3dLayout-SetbasePosition: x y z;		left:x; top:y; -webkit-transform: translateZ(z);
	-3dLayout-positioning: x y z;		left:x; top:y; -webkit-transform: translateZ(z);

Fig. 11. Translation of content arrangement in 3D view volume.

used to hide an object (Figs. 9–10).

(3) Translation of content arrangement in 3D view volume

Input (extended CSS3 properties)		mid-process		Output (genuine CSS3 properties)
-3dLayout-view-direction: x y z; -3dLayout-up-vector: x y; -3dLayout-PerspectiveVolume: <fovy> <aspect> <near> <far>;	⇨	-webkit-transform: translateZ(<perspective-d>) rotateX(-θ_x) rotateY(-θ_y) translateZ(-<perspective-d>); -webkit-transform: translateZ(<perspective-d>) rotateZ(-θ_{up}) translateZ(-<perspective-d>); -webkit-transform: scale(scale$_x$, scale$_y$);	⇨	-webkit-transform: translateZ(<perspective-d>) rotateZ(-θ_{up}) rotateX(-θ_x) rotateY(-θ_y) translateZ(-<perspective-d>) scale(scale$_x$, scale$_y$);

Fig. 12. Translation of view-volume setting.

In the genuine CSS3, the contents can be arranged in 2D space, but any additional properties are not supported for their expression in 3D space. To rearrange 2D contents in 3D space, the positions of the contents must be extended along the z axis. The extended CSS3 used by this study translates the positions of an object along the z axis by as far as the z value, which is set as an extended specification (Fig. 11).

4.1.2. Translation process of multiple properties

If more than two view-volume-related properties are used, the translation process of multiple properties needs to occur. First, transform functions are used for the translation process of a single property, but when more than two properties are used together, it is necessary to integrate those functions that were used in the CSS3-transforms module into a single transform property [8]; therefore, in the translation process of multiple properties, numerous functions are integrated into a single transform property. Also, all of the transform functions are expressed in 4 × 4 transformation matrices [9], and because the commutative law does not hold true [10], the transform functions must follow a certain order in accordance with the way that they are described within the transform attributes. This study followed the order of rotate Z, rotate X, rotate Y, and scale. As shown in Fig. 12, the mid-process was similar to the one used by the single-property translation process, but the outputs were produced in the sequence of rotate Z → rotate X → rotate Y → scale (regardless of the input order).

4.1.3. Translation process for duplicated settings of the same properties

When the same properties are duplicated in the 3D-transforms module, they are expressed as the last property values [11]. The preprocessor used by this study replaced the genuine properties with the last values, removing all of the other overlapping properties when the duplicated properties are used (Fig. 13).

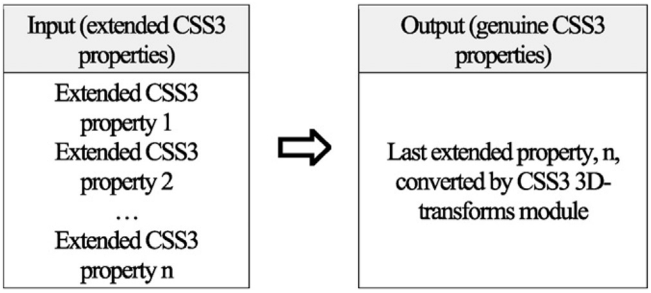

Fig. 13. Translation of duplicated properties.

4.2. Results and analysis of preprocessor execution

4.2.1. Preparation of sample contents and conformance testing

In this study, to check whether the extensions are designed adequately for the serving of the intended purpose, sample web pages were placed within 3D view volumes that are defined by the extended CSS3 specifications, and the expression results were checked on the kit-based web browser. The expression results were then compared with those produced by Maya, a 3D-graphic software program, to evaluate the accuracy of the preprocessor-execution results. Further, in order to verify the efficiency of CSS3 extended specification, we analyzed the reduction rate by measuring the number of code characters of the same sample written in the CSS3 extended specification and the number of code characters in genuine CSS3. The calculation formula is $\frac{\mathrm{genuineCSS3} - \mathrm{extendedCSS3}}{\mathrm{genuineCSS3}}$ (unit: number of code characters). The calculation is rounded off to three decimal places.

The title of the sample contents prepared by this study is “News web page” [12]. First, a certain depth value was assigned to each element of the sample contents, and all of the setting information that is related to the view volume is described in <body>. The default position of the camera was set as the -3dLayout-perspective-origin property. The coding of the sample contents was completed in the following order: view-point setting → view-space setting → view-volume setting (Fig. 14).

(1) View-point setting

To check the view-point results after the setting of the view direction, the camera was rotated up and down and a certain value was entered into the up-vector to tilt it in Maya. Fig. 15 (a) shows the results in Maya when the camera was rotated at a horizontal angle of 18.435° to the right and at a vertical angle of 9.462° to the upside, and when it was rotated at a horizontal angle of 26.565° to the right. Fig. 15 (b) shows the expression results of the gradient of the camera according to the -3dLayout-view-direction, when the extended specification -3dLayout-view-direction is used in the preprocessor to express the horizontal and vertical rotations of the camera (Fig. 16).

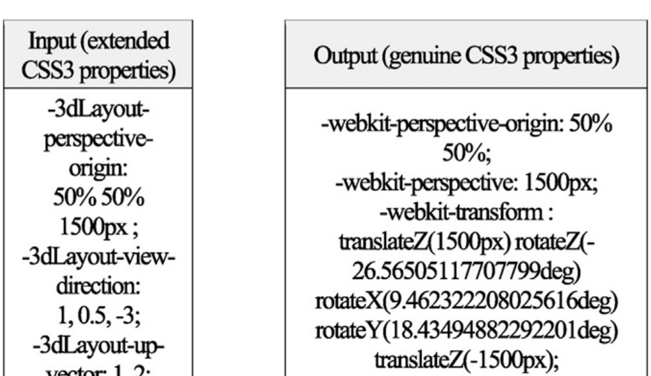

Fig. 14. Translation of settings of vertical rotation and camera gradient.

(2) View-space setting

To check the view-space results, the camera's fovy was set. Fig. 17 (a) shows the results in Maya when the camera's fovy is set as 60°. Fig. 17 (b) shows the results when the camera's fovy is set as 60° by the extended CSS3 property -3dLayout-PerspectiveVolume.

(3) View-volume setting

Next, the creation results of the view volume were checked after the view point and view space were set (Fig. 18); to this end, the rotation, gradient, and fovy of the camera were also set. Fig. 19 (a) shows the execution results in Maya when the camera was rotated at 18.435° to the right and at 9.4629.462 to the upside, and when it was rotated 26.565° to the right with the fovy set at 60°. Fig. 19 (b) shows the execution results when the horizontal and vertical rotations of the camera are set with the use of the extended specification -3dLayout-view-direction in the preprocessor, and when the fovy of the camera is set as 60°.

4.2.2. Analysis of results

Including the three contents mentioned in the previous chapter, the results of a total of 14 sample contents that were created with the extensions and Maya were compared. According to the evaluation results, the execution results of the extended CSS3 specifications are consistent with those of Maya, indicating that it is possible to efficiently reconstruct web content in 3D space by freely setting the view volume. In addition, the accuracy of the translation results are confirmed; therefore, users can control the view-volume setting by using the extended CSS specifications and the preprocessor.

We examined the advantages of the extended specifications in comparison with those of the genuine CSS3, and the results are as follows:

- While the genuine CSS3 specifications allow users to set basic locational information when they set the view point, the extended CSS3 specifications allow users to control the location, pointing-direction, and up-vector settings.
- In the genuine CSS3 specifications, as the view space is fixed to the size of the view plane, users cannot control the view space; however, the extended specifications allow users to freely control the view space.

Table 6 shows the results of the comparison between the view-volume settings of the genuine CSS3 specifications and those of the extended CSS3 specifications.

Next, we evaluated the number of the source code by separating the genuine CSS3 source code and extended CSS3 source code into a character unit. We conducted the test and obtained reduction ratio by classifying into View direction, Up-vector, Up-Vector & View direction, View space and View volume. View direction is classified once again by direction. ‘View volume’ is a classification of the samples created using functionality of the View direction, up-vector and view space.

As a result, we confirmed that every examined 14 samples had result of significant decreasing of character numbers using extended CSS3 than using genuine CSS3, as shown in Table 7. In particular, the reduction ratio when setting the view point is showed the greatest efficiency decreased by more than 0.5 times higher than the number of characters in genuine specifications.

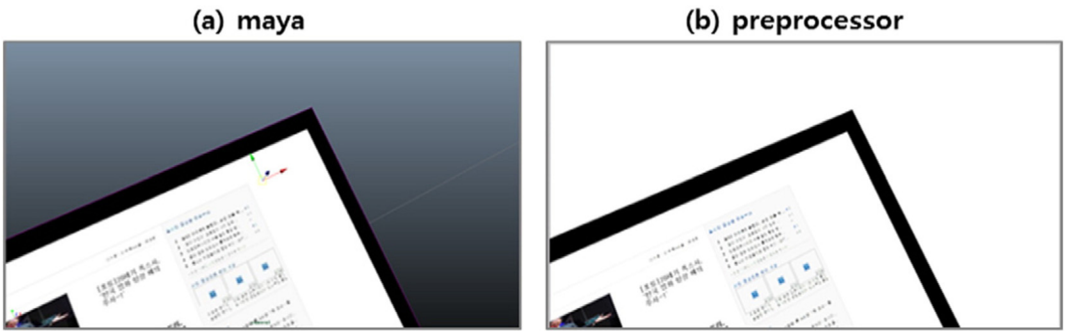

Fig. 15. Comparison between setting results of vertical rotation and camera gradient.

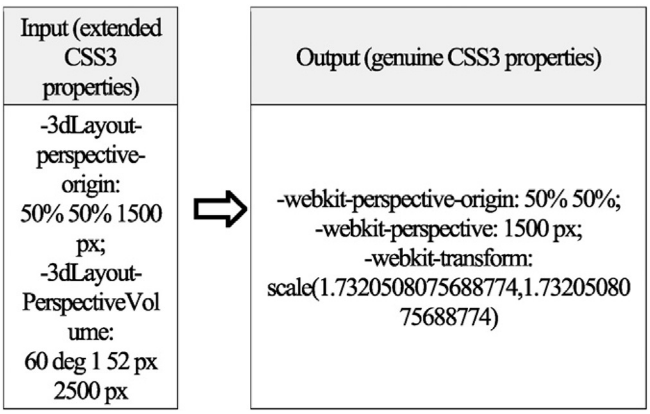

Fig. 16. Translation of settings of camera's fovy.

Also, reduction ratio showed high when setting View volume, which decreased more than 0.4 times. On the other hand, reduction ratio in setting View space was not significant, resulting 0.3 times than genuine CSS3, but it appeared to be very effective than traditional methods.

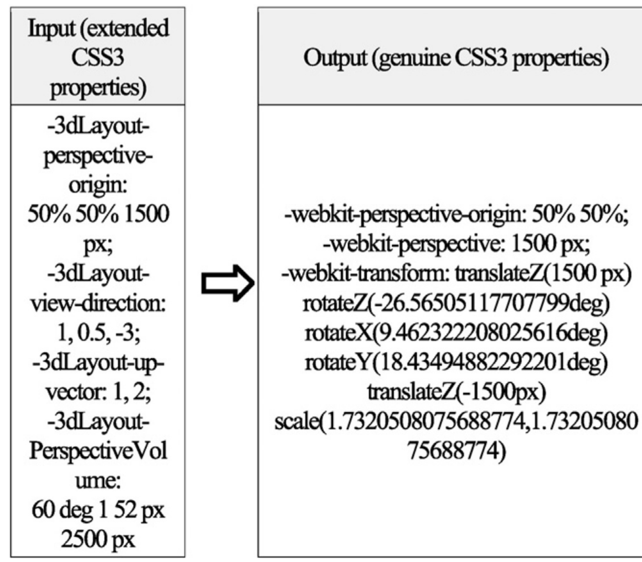

Fig. 18. Comparison results of camera's fovy.

5. Implementation of rendering-engine emulator for stereoscopic 3D expression of web contents

5.1. Implementation and execution results of rendering-engine emulator

This study implemented a rendering-engine emulator so that the results of the stereoscopic expression of the web contents that were written according to the setting of the transform-style property as the extended value 3dstereo could be checked.

The emulator automatically set the stereo view volume by moving the viewpoint location of the mono view volume by a certain distance to the left and right. To create two view volumes at a distance of 65 mm, which is the average distance between the left and right eyes of an adult, the perspective-origin property was used to set a user's location on two view volumes.

- Image output from left-eye view volume

perspective-origin (< x coordinate of view volume set by the user: -32.5 mm > , < y coordinate of view volume set by the user >)

Fig. 17. Comparison of setting results of camera's fovy.

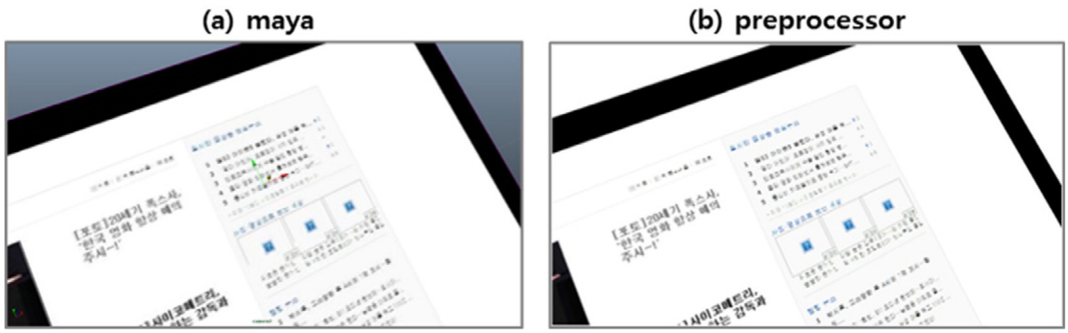

Fig. 19. Comparison results of view-volume setting.

Table 6
Comparative analysis of functions of genuine CSS3 with those of extended CSS3.

Class	Basic information of 3D view volume	Genuine CSS3 specifications	Extended CSS3 specifications
View point setting	Viewpoint(x, y, z)	○	○
	Up-vector(x, y, z)	×	○
	View direction(x, y, z)	×	○
View space setting	View angle (a_w, a_h)	×	○
	near, far clipping distance (d_n, d_f)		
	Convergence distance (d_c)	○	○
Conclusion			all are possible

Table 7
Reduction ratio of characters when using extended CSS3 specification.

Category		Extended CSS3 code (token)	Genuine CSS3 code (token)	Reduction ratio (%)
View point	View direction	75.3	164.3	0.54
	Up-vector	66	147	0.55
	Up-vector & view direction	100	205	0.51
View space		87	124	0.30
View volume		145	249	0.42

- Image output from right-eye view volume

perspective-origin (< x coordinate of view volume set by the user: +32.5 mm > , < y coordinate of view volume set by the user >)

After acquiring the left- and right-eye images from the two-eyed virtual-camera set described above, the images were combined to reconstruct a stereo image. The created stereo image was emulated to fit into the display features of a 3D TV.

The sample contents used in this study consist of Sports News [14], Shopping Mall [15], and History Special [16]. The basic user interface and stereoscopic-expression results of the sample contents can be seen in Figs. 20 to 22.

5.2. Result analysis of emulator execution

To verify the proposed system, this study showed users stereo images that were projected on an actual 3D TV so that they could evaluate the final results. The evaluation participants did not experience any problems recognizing the stereoscopic expressions, as they were not hampered by any major visual impairments or the strabismus condition that is a misalignment of the eyes. From among the participants, this study finally selected nine people with experiences of watching 3D content on a 3D TV. After the stereoscopic concept and the overall creation process of the stereo images were explained to the nine selected participants, three kinds of the sample contents that were created by this study were shown to them. The participants were instructed to score six question items on a scale of 1 to 5 with the following values: Very unreal (1 point), Unreal (2 points), Average (3 points), Real (4 points), and Very real (5 points). The question items are shown in Table 8.

Table 9 shows the averages of the total scores of each of the three sample contents. Overall, the scores of all of the sample contents are high; especially the scores of the 3D shopping mall that are higher than the total average, leading to the consideration that it is the most suitable for 3D. Also, the average scores of

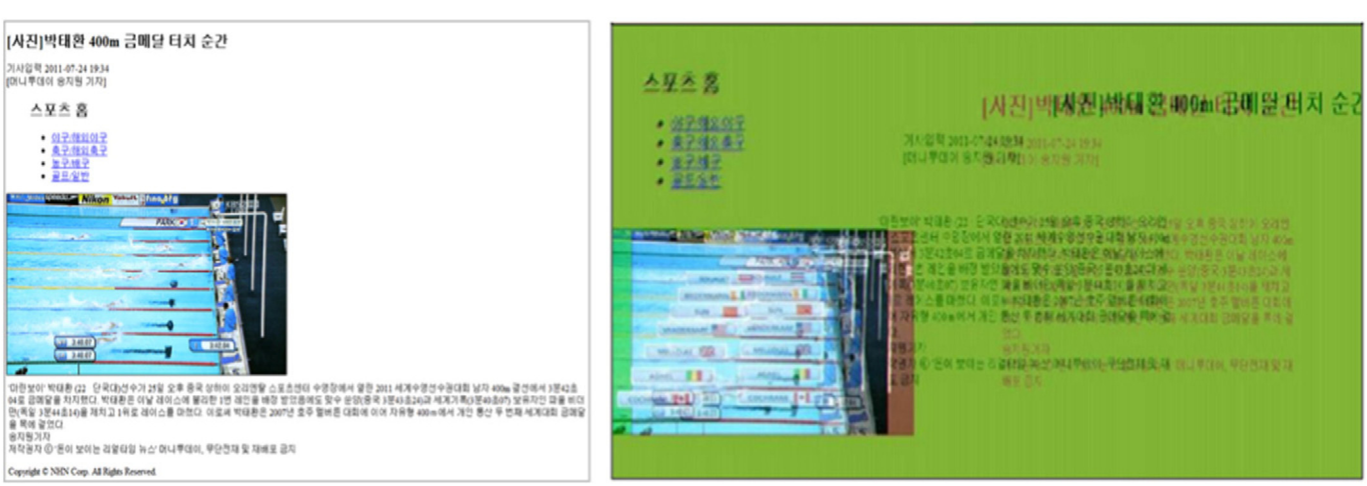

Fig. 20. Basic UI of Sports News page and result of created stereo image.

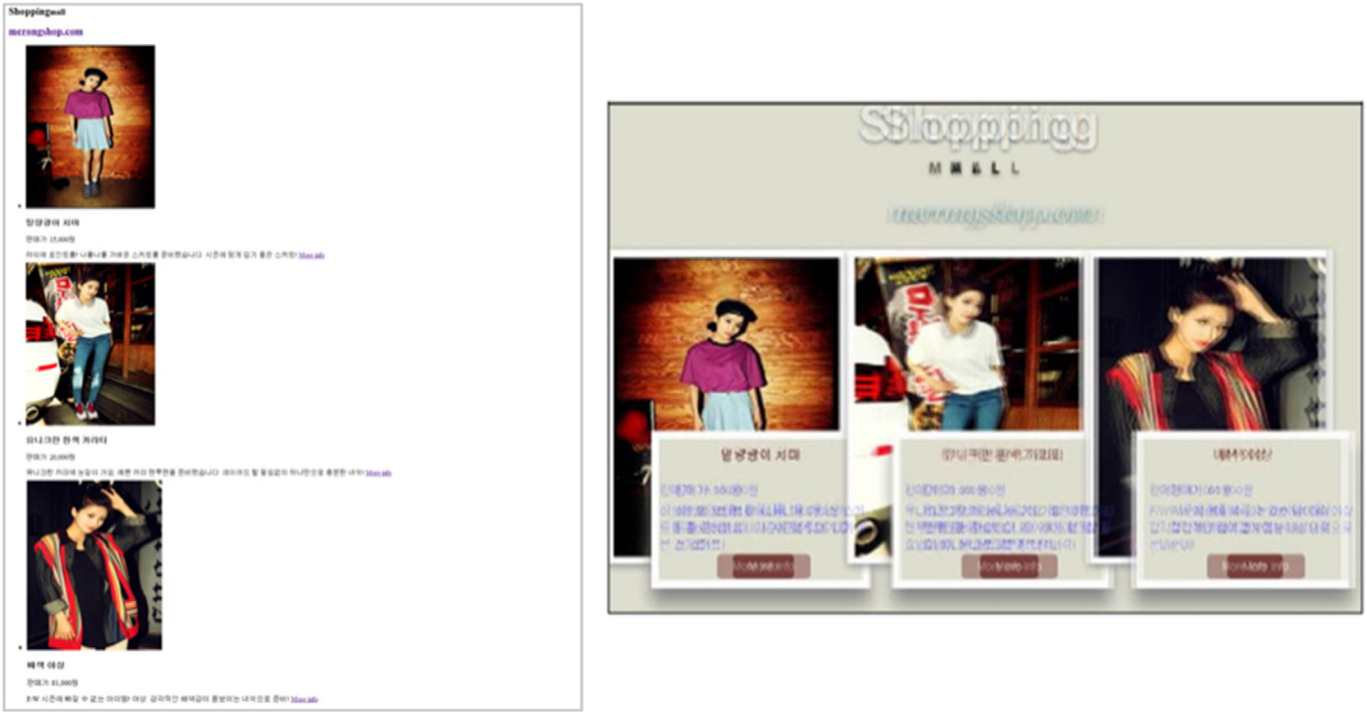

Fig. 21. Basic UI of Shopping Mall page and result of created stereo image.

Questions 1 to 6 are very close to the perfect score, indicating that the extended CSS3 specifications that are proposed by this study can effectively produce 3D effects, and that 3D contents are more interesting than 2D content. Given these findings, further studies must be conducted. In contrast, Question 2 received the lowest scores, but a solution lies in a more-sophisticated creation process regarding the sample contents.

The participants were also asked whether they are willing to use a 3D-content web service that is provided on a 3D TV; all of the evaluators expressed a willingness that is due to the novel nature of 3D content. The participants showed a greater interest in the entertainment contents than the other types, and they said they would primarily use 3D content for this purpose. Some of the evaluators also made suggestions regarding the screen

Fig. 22. Basic UI of History Special page and result of created stereo image.

Table 8
Six question items of the 3D contents written with extended CSS3 extensions.

Question 1	Can you see 3D effects in the 3D contents?
Question 2	Is the substance of the 3D contents effectively delivered to you?
Question 3	Is the screen composition of the reconstructed 3D images appropriate?
Question 4	Are 3D contents better composed than 2D contents?
Question 5	Is the substance of the 3D contents more effectively delivered to you than the 3D contents?
Question 6	Are 3D contents more interesting than 2D contents?

Table 9
Evaluation results of the 3D contents written with extended CSS3 specifications.

Question	Sports news	3D shopping mall	History special	Average
1	4.8	5.0	4.8	4.9
2	4.2	4.4	4.2	4.3
3	3.6	4.0	3.7	3.8
4	4.4	4.6	4.3	4.4
5	4.6	4.7	4.6	4.6
6	4.9	4.9	5.0	4.9
Average	4.4	4.6	4.4	4.5

composition, with one saying that 3D content that allows users to freely control or add dynamic motions would be more interesting.

The evaluation determined that web content must be more effectively composed to be suitable for 3D-display broadcasting by a data-broadcasting service; accordingly, the production process regarding web content needs to be further improved to serve the respective purpose of each type of web page.

6. Conclusion

In this study, the 3D spaces on HTML5 web pages, for which the CSS style and browser-based declarative-content technology are used, are defined; furthermore, the required view volume was set for the reconstruction of the spaces into 3D contents, and the technique for the expression of the spaces on a stereoscopic 3D display is explained. Based on the proposed method, it is possible to effectively reconstruct 2D web content in 3D space using only a CSS style, which belongs to the HTML web standards, and additional programs are unnecessary; furthermore, it is possible to express the 2D content in stereoscopic 3D images.

Future studies including one in which several types of the graphical information that are required for the walk-mode setting of a camera are additionally described need to be conducted. Also, more studies need to be carried out regarding the ways that users can freely obtain camera images after their declarative contents are classified according to their meanings and element types and they are placed on the view volume.

If the extended CSS3 specifications that are proposed by this study are accepted as the W3C standards, it will be possible for users to create 3D contents that can be translated on a web browser without the need to resort to additional compilers or professional graphics-editing software programs. We believe that it will become possible for users to freely express web content in 3D space, but only if CSS3 is used; furthermore, if web content is created in such a way, it can possibly be used on a hybrid TV [17]. But for the standardization, the points of "implemented already and widely" and "stabilized sufficiently" are important. That is, the standard does not guarantee the quality of the implementation. It is each browser vendor's work to implement the extended specifications, which may not be properly implemented. This study implemented a preprocessor and an emulator to check whether the specifications do work or not, and if it is 'widely' implemented by various browser vendors and 'stabilized sufficiently', it would be nice to be expect W3C standardization.

We hope that this study will become a foundation for the development of the free expression of 3D content, and that Web-based 3D-related studies that offer extended specifications for the free-movement camera function will be actively carried out in the near future.

References

[1] J. Jankowski, S. Decker, On the design of a dual-mode user interface for accessing 3D content on the world wide web, Int. J. Hum. Comput. Stud. 71 (7) (2013) 838–857.
[2] J. Sutter, P. Slusallek, A CSS integration model for declarative 3D, in: Proceedings of the 20th International Conference on 3D Web Technology, 2015, June, pp. 209–217 (ACM).
[3] Wang, W., Dong, S., Wang, R., Cheng, Q., Zhang, J., & Liu, Z. (2014, April). Stereoscopic 3D web: from idea to implementation. Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on IEEE Vol. 3, pp. 1440–1444).
[4] Soonbo Han, Dong-Young Lee, Extensions for Stereoscopic 3D Support (W3C), 30 November 2012 Available: http://www.w3.org/2011/webtv/3dweb/3dweb_proposal_121130.html.
[7] Laboratorio di grafica al Calcolatore, downloaded 2014, Apr. 20 Available: http://www.toldo.info/roberto/LaboratorioGrafica/Slides/lesson6-Printable.html.
[8] Css Transforms Tutorials WPD WebPlatform, 2014, Apr. 20 Available: http://docs.webplatform.org/wiki/tutorials/css_transforms.
[9] Simon Fraser, Dean Jackson, Edward O'Connor, Dirk Schulze, CSS Transforms Module Level 1(W3C), 26 November 2013 Available: http://www.w3.org/TR/css-transforms-1/#mathematical-description.
[10] Wikipedia, Matrix multiplication, 2015, July 24 Available: https://en.wikipedia.org/wiki/Matrix_multiplication.
[11] CMSFactory.NET CSS Basic Grammar 2014, Apr. 20 Available: http://www.cmsfactory.net/node/10508.
[12] NATE News, 2014, Apr. 20 Available: http://news.nate.com.
[14] Song Ji-Won (2011, Jul. 24), Park Tae-Hwan, Finishing Touch for Gold Medal in Men's 400 m Freestyle (Money Today News), Available: http://www.mt.co.kr/view/mtview.php?type=1&no=2011072419315450052&outlink=1 (downloaded 2012, Nov. 30).
[15] Merongshop, Photo (Image), 2012, Nov. 30 Available: http://www.merongshop.com.
[16] History Special, Mysterious Ancient Kingdom, Shilla Episode 2, Why Did the Gold Crown Vanish? 2012, Nov. 30 Available: http://www.kbs.co.kr/end_program/1tv/sisa/history.
[17] H.-J. Yim, et al., A study on 3D representation of declarative content for web platform based hybrid TV, in: Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on IEEE, 2014.