The investigation of data voxelization for a three-dimensional volumetric display system

2009 J. Opt. A: Pure Appl. Opt. 11 045707

(http://iopscience.iop.org/1464-4258/11/4/045707)

IOP PUBLISHING JOURNAL OF OPTICS A: PURE AND APPLIED OPTICS

J. Opt. A: Pure Appl. Opt. **11** (2009) 045707 (5pp) doi:10.1088/1464-4258/11/4/045707

The investigation of data voxelization for a three-dimensional volumetric display system*

Xiaoyan Xie, Xu Liu and Yuanfang Lin

State Key Laboratory of Modern Optical Instrumentation, Zhejiang University,
Hangzhou 310027, People's Republic of China

E-mail: liuxu@zju.edu.cn

Received 8 August 2008, accepted for publication 6 January 2009
Published 30 January 2009
Online at stacks.iop.org/JOptA/11/045707

Abstract

A high resolution three-dimensional (3D) volumetric display system utilizing a rotating light-emitting diode (LED) array is presented, which provides viewers with true depth cues, binocular parallax, accommodation and convergence, etc, and can be observed from any direction without the need for any special viewing aids. The data voxelization method for the system is presented. The evaluation of texture distortion due to the deviations of the voxel positions caused in voxelization is introduced. 3D models with two types of texture are built: one in which the gray scale is nearly invariant in the background, and the other in which the gray scale varies in the whole picture. The texture distortion of models with the two types of texture is evaluated and a numerical analysis is given. The relationship between texture distortion and voxelization precision is studied. Voxelization precision can be improved by shortening the voxelization step length. Experiments show that models with textures in which gray scale varies gradually in the whole picture need higher voxelization precision than textures with an invariant gray scale background. In order to obtain similar display quality, the ratio of the voxelization step length of models with the two types of texture is about 5/2.

Keywords: three-dimensional volumetric display system, texture mapping, voxelization, step length, position deviation, texture distortion

(Some figures in this article are in colour only in the electronic version)

1. Introduction

A diverse range of volumetric display systems was first proposed in 1912 [1]. However, due to the restriction of the technologies, these ideas were not practically implemented before the 1990s. In the past decade, with the development of optoelectronic and computer technologies, some 3D volumetric display prototypes have been developed, such as the projection-based 3D display system, laser scanning-based display system, and rotation light-emitting diode (LED) panel system [2–4].

We utilized a color LED with a voxel size of 1.6 mm × 1.6 mm and developed a 3D volumetric display system based on a rotating 2D LED array.

In the 3D volumetric display system, it is important to slice and transform the original 3D data into a voxel format that is proper for system addressing. Since data transformation would cause the original data distorted as it represented in the voxel space. So far many studies have been done on the different kinds of data processing methods for 3D display systems, and a lot of research has been done [5–12]. However, none of them concerns the problem of data distortion. This paper presents a data voxelization method, evaluation of texture distortion, and the relationship between the voxelization precision and texture distortion.

2. Display system

A photograph of the display system we have developed is shown in figure 1. It mainly comprises the LED display panel, driver circuits, and the mechanical system.

* This project was supported by the High-Tech Research and Development Program of China (2007AA01Z339).

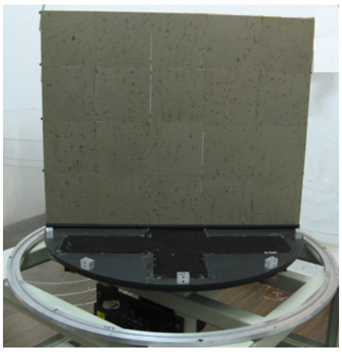

Figure 1. Display system.

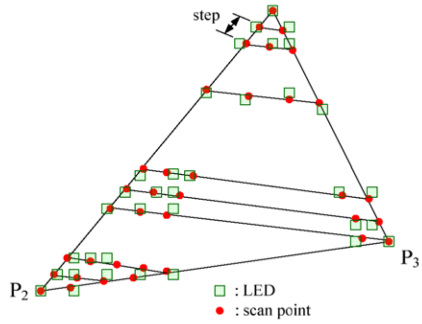

Figure 3. Sketch of a triangle's voxelization.

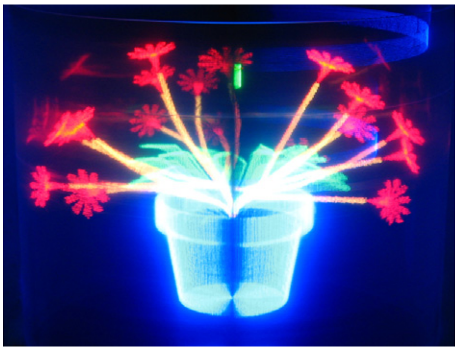

Figure 2. Photograph of a 3D image displayed by the system.

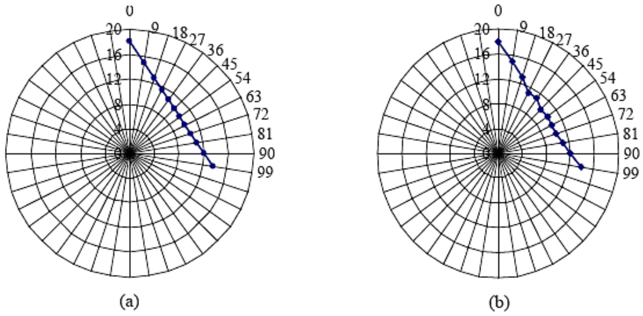

Figure 4. Comparison of a line before and after voxelization; (a) before voxelization, (b) after voxelization.

The LEDs are arranged in a matrix configuration with 256 rows and 320 columns on the panel with a pitch of 2.5 mm × 2.5 mm. The principle of the system is as follows. The LED panel rotates around its central axis, sweeping out a cylindrical voxel space. A 3D model is sliced and preprocessed. The LED panel displays these 2D slices in sequence at a predefined interval when it rotates. When the rotating speed is high enough, these discrete 2D images will be perceived and interpreted as an integrated 3D image due to persistence to the eye.

A photograph of a 3D image displayed by the system is shown in figure 2.

3. Data voxelization

3.1. Texture mapping

The 3D models are built and saved as 3DS format files by 3D Studio Max software. The surfaces of the 3DS models are joined by many small triangles [13]. Textures are mapped to the surfaces of 3DS models using texture mapping, which is a method for adding detail to a computer-generated graphic or 3D model. Textures are applied to objects using texture coordinates, which are associated with each vertex within a model, and indicate what point within the texture is attached to that vertex. Most textures, like bitmaps, are a 2D array of color values. The individual color values are called a texture element, or texel. Each texel has a unique address in the texture. The address can be thought of as a column and row number, which are labeled u and v, respectively. Each texture coordinate is assumed to be a floating-point value. Each set of texture coordinates must be mapped to a position within the texture image. The coordinates of the texture map are in the range [0, 1] in each dimension.

The spatial and texture coordinates of each vertex in the triangles in the 3DS model surfaces are saved in 3DS files, and are loaded and processed by the system's software.

3.2. Voxelization

Since the surfaces of the 3DS models are joined by small triangles, a single triangle is the element in the voxelization. Figure 3 is a sketch of a triangle's voxelization. The process includes the following steps. (1) Setting the step length (this has been labeled in figure 3) according to the model's size and the voxelization precision. The shorter the step length is, the more precise the voxelization is. (2) Discretizing the three sides of the triangle into point sets respectively in the step length. (3) Discretizing the triangle facet into line sets. (4) Discretizing the lines into point sets in the step length. (5) Finding the nearest match LEDs for the points as follows.

Suppose a vertex's coordinates are (x, y, z, u, v); x, y, and z are the spatial coordinates, and u and v are the texture coordinates. Transform x, y, z into cylindrical coordinates r, θ, z. The nearest LED mapped to the vertex is found as follows:

$$\begin{aligned} i &= \mathrm{floor}(r/\mathrm{scale} + 0.5) \\ j &= \mathrm{floor}(z/\mathrm{scale} + 0.5). \end{aligned} \tag{1}$$

(a) (b)

Figure 5. Two types of textures: (a) peony, (b) sea.

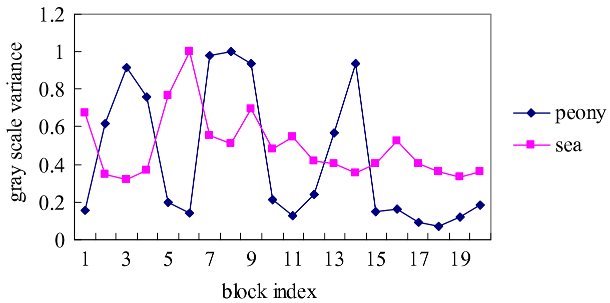

Figure 6. Curves of the block gray scale variance of the two types of texture.

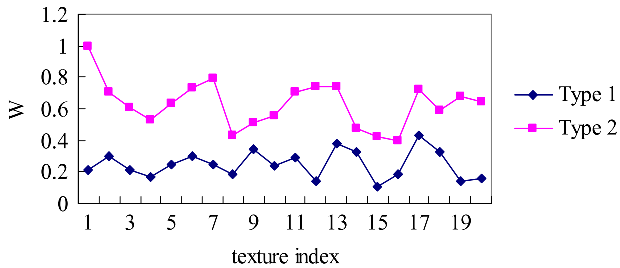

Figure 7. Curves of W in the same step length for the two types of model.

The scaling factor 'scale' can be calculated according to the parameters of LED panel (the pitch of the LEDs, the resolution of LED panel, etc) to change the size of 3DS model to adapt to the size of the display space. The floor function is a function whose value is the largest integer less than or equal to the value in the brackets.

The slice index l for the vertex can be obtained from equation (2):

$$l = \begin{cases} 0, & x \geqslant 0, y = 0 \\ L/2, & x < 0, y = 0 \\ L/4, & x = 0, y > 0 \\ 3L/4, & x = 0, y < 0 \\ \text{floor}\left\{\left[\dfrac{\arctan(y/x)}{2\pi}L\right] + 0.5\right\}, & x \neq 0, y > 0 \\ \text{floor}\left\{\left[\dfrac{\arctan(y/x)}{2\pi}L\right] + L + 0.5\right\}, & x \neq 0, y < 0. \end{cases} \tag{2}$$

L is the total slice number per rotation. So, by the steps above, a vertex with spatial coordinates x, y, z is mapped to the LED in the lth slice, ith column, and jth row, labeled as (i, j, l) in the voxel space. Voxel (i, j, l) has the same texture coordinates u, v as the vertex (x, y, z). Therefore, the point with texture coordinate (u, v) in the texture is attached to voxel (i, j, l). The RGB value of voxel (i, j, l), which equals the RGB value of point (u, v) in the texture, is obtained by loading the data of the texture.

4. Texture distortion

4.1. Cause of texture distortion

The voxelization that finds the nearest matching LED for the vertex causes a position deviation between the voxel and the original vertex. Figure 4 is the top view of a line in voxel space before and after voxelization. It becomes a zigzag line after voxelization.

Since the texture is attached to the surface of the model, it would be distorted if the shape of the model twists due to the voxel position deviations. A single voxel's position deviation is

$$\Delta d_0 = \sqrt{(x_v - x_0)^2 + (y_v - y_0)^2 + (z_v - z_0)^2}. \tag{3}$$

x_v, y_v, z_v are voxel's spatial coordinates, which can be calculated by the indices i, j, l and equations (1) and (2).

In order to simplify the calculation, in this paper we just select models with gray scale textures as objects to study their distortion.

4.2. Effect of voxel position deviation

The voxel position deviation is an important factor of texture distortion, as shown in section 4.1. Suppose n voxels are lightened after voxelization; the evaluation of the effect of the voxel position deviation on texture distortion is

$$D_i = \Delta d_i / \Delta d_{\text{avg}}. \tag{4}$$

Δd_i is a single voxel's position deviation, which can be calculated with equation (3). Δd_{avg} is the average position

(a) (b)

Figure 8. Simulations of the voxelization results with two types of texture in the same step length; (a) box with texture peony, (b) box with texture sea.

deviation of all the lightened voxels.

$$\Delta d_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} \Delta d_i. \tag{5}$$

4.3. Effect of texture detail distribution

Research shows that people usually concentrate on areas where the color or gray scale varies significantly when perceiving an image. These areas have more detail. Therefore, different areas with different detail distribution in an image have different visual effect. We divide an $M \times N$ texture equally into t non-overlapping $P \times Q$ sub-blocks. The detail distribution in each sub-block is described by

$$\sigma_j = \left(\frac{1}{k-1}\sum_{i=1}^{k}\left(g_i - \frac{1}{k}\sum_{i=1}^{k} g_i\right)^2\right)^{1/2}; \tag{6}$$

$k = P \times Q$ $(j = 1, 2, \ldots, t)$. Voxels attached to the points located in the ith sub-block have an effect factor of texture detail distribution σ_j on the texture distortion.

4.4. Effect of voxel gray scale

Weber–Fechner's rule [14] shows that the visually subjective brightness is proportional to the logarithm of the stimulating signal's brightness. Here the stimulating signal's brightness is the relative brightness that the signal has compared to the background. In our display system, a voxel's final display brightness is proportional to its gray scale value. Suppose that a single voxel's gray scale is g_i, and that the average gray scale of the sub-block where the point that the voxel attaches to is $g_{j\text{avg}}$; then the relative gray scale of the voxel is $(g_i/g_{j\text{avg}})$. Voxels with different relative gray scales have a different visual effect on texture distortion even if their position deviations are the same. Therefore, the evaluation of the effect of a single voxel's relative gray scale on the texture distortion is

$$G_i = b \log \frac{g_i}{g_{j_{\text{avg}}}}; \tag{7}$$

b is a constant.

Therefore, the evaluation of a texture's distortion after voxelization is

$$W = \frac{1}{n}\sum_{i=1}^{n} D_i |G_i| \sigma_j. \tag{8}$$

j changes from 1 to t, depending on which sub-block the point in the texture attached to voxel i is located in. The bigger W is, the more serious the texture distortion is.

5. Experiment and analysis

Since the step length affects the voxelization precision (see section 3.2), the texture distortions were evaluated and the relationship between the step length and the texture distortion is studied below.

Two types of texture were utilized. (1) Images with nearly invariant gray scale in the background: the gray scale just varies in small areas; see figure 5(a). (2) The gray scale gradually varies in the whole picture; see figure 5(b). 20 600×800 textures were selected in each type. Each texture was divided into 20 blocks. Taking figures 5(a) and (b) as examples, the gray scale variances of the blocks were calculated by using equation (6); see figure 6. The value on the diagram is the ratio of each sub-block's σ_j $(j = 1, 2, \ldots, t)$ to the maximal value in σ_j. From the diagram we can see that block variances of picture (b) (sea) do not change much (most are in the range 0.4–0.6), while some blocks in the background of picture (a) (the peony) have very low variances, less than 0.2, and some in the pattern part have high variances.

3DS models with the two types of texture were built. Their texture distortion was evaluated by calculating W in the same voxelization step length. The calculated results are shown in figure 7. They are normalized values, each W being divided by the maximal value.

Type 1 and type 2 in figure 7 mean the calculated results of models utilizing the type (1) and (2) textures. From the curves we can see that textures in type (2) distort more seriously than in type (1) in the same step length. Because the gray scale in the background of type (1) textures is nearly invariant, the voxel position deviations do not cause texture distortion in these areas, whereas in type (2) the gray scale gradually varies over the whole texture, and a small position deviation would cause gray scale distortion. Figure 8 shows simulations

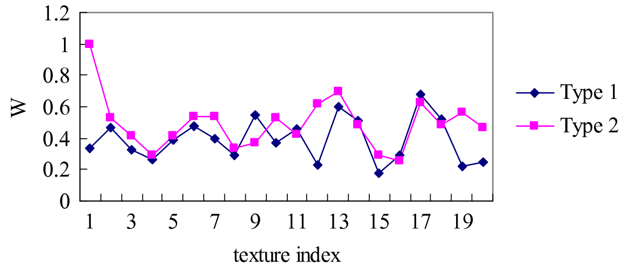

Figure 9. Curves of W for the two types of model, with the step length ratio being 5/2.

Figure 10. Simulation figure of voxelization results of the box with texture sea after shortening the step length.

of the voxelization results of the box models utilizing the peony and the sea as textures. There is some loss of detail in the boundaries where gray scale varies in figure 8(a), but not obviously, whereas in figure 8(b), there are some obvious distortions. For example, the boundary of the sky and the sea, the waves, and the rock are distorted, compared to figure 5(b).

We shortened the voxelization step length for type 2 models while keeping it constant for type 1. Lots of experiments show that when the ratio of type 1's step length to type 2's step length is 5/2, their texture distortion evaluation curves are similar, as shown in figure 9. The simulation figure of the model box with the texture sea after step length was shortened is shown in figure 10, in which the texture distortion is improved. This is because the voxelization precision is improved as step length shortened. Thus the voxel position deviations reduce and the display quality is improved.

6. Conclusion

A data voxelization method for a 3D LED volumetric display system was presented. Evaluation of the distortion of the gray scale textures after voxelization was introduced. Experiments show that models with textures in which the gray scale varies in the whole picture need higher voxelization precision. Further discussion that focuses on the distortion of color textures is required.

References

[1] Luzy E and Dupuis C 1912 Procédé pour Obtenir des Projections en Relief *French Patent Specification* 461,600
[2] Downing E A *et al* 1996 A three-color, solid-state, three-dimensional display *Science* **273** 1185–9
[3] Favalora G E *et al* 2002 100 Million-voxel volumetric display *Proc. SPIE* **4712** 300–12
[4] Yuan-fang L *et al* 2003 Three-dimensional volumetric display system utilizing a rotating two-dimensional LED array *Acta Opt. Sin.* **23** 1158–62
[5] Chen H and Fang S 1999 Fast voxelization of 3d synthetic objects *ACM J. Graph. Tools* **3** 33–45
[6] Zhao D, Wei C, Hujun B, Hongxin Z and Qunsheng P 2004 Real-time Voxelization for Complex Polygonal Models *Proc. 12th Pacific Conf. on Computer Graphics and Applications*
[7] Eisemann E and Décoret X 2006 Fast scene voxelization and applications *Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games* pp 71–8
[8] Downing E A *et al* 1996 A three-color, solid-state, three-dimensional display *Science* **273** 1185–9
[9] Favalora G E *et al* 2002 100 Million-voxel volumetric display *Proc. SPIE* **4712** 300–12
[10] Haumont D and Warzee N 2002 Complete polygonal scene voxelization *ACM J. Graph. Tools* **7** 27–41
[11] Heidelberger B, Teschner M and Gross M 2003 Real-time volumetric intersections of deforming objects *Proc. Vision, Modeling, Visualization (2003, Munich)* pp 461–8
[12] Jian H *et al* 1998 An accurate method for voxelizing polygon meshes *Proc. IEEE Volume Visualization 98* (Piscataway, NJ: IEEE) pp 119–26
[13] Xiaojun P, Yan Li and Hangaen He 2003 3DS models application in virtual reality geometry modeling *Comput. Simul.* **20** 52–5
[14] Xuguang Lu *et al* 2005 A method of image quality assessment based on visual regions of interest *Microcomput. Inf.* **21** 95–6