Visualization and Computer Graphics on Isotropically Emissive Volumetric Displays

Benjamin Mora, Ross Maciejewski, Min Chen, Member, IEEE, and David S. Ebert, Senior Member, IEEE

Abstract—The availability of commodity volumetric displays provides ordinary users with a new means of visualizing 3D data. Many of these displays are in the class of isotropically emissive light devices, which are designed to directly illuminate voxels in a 3D frame buffer, producing x-ray-like visualizations. While this technology can offer intuitive insight into a 3D object, the visualizations are perceptually different from what a computer graphics or visualization system would render on a 2D screen. This paper formalizes rendering on isotropically emissive displays and introduces a novel technique that emulates traditional rendering effects on isotropically emissive volumetric displays, delivering results that are much closer to what is traditionally rendered on regular 2D screens. Such a technique can significantly broaden the capability and usage of isotropically emissive volumetric displays. Our method takes a 3D data set or object as the input, creates an intermediate light field, and outputs a special 3D volume data set called a lumi-volume. This lumi-volume encodes approximated rendering effects in a form suitable for display with accumulative integrals along unobtrusive rays. When a lumi-volume is fed directly into an isotropically emissive volumetric display, it creates a 3D visualization with surface shading effects that are familiar to the users. The key to this technique is an algorithm for creating a 3D lumi-volume from a 4D light field. In this paper, we discuss a number of technical issues, including transparency effects due to the dimension reduction and sampling rates for light fields and lumivolumes. We show the effectiveness and usability of this technique with a selection of experimental results captured from an isotropically emissive volumetric display, and we demonstrate its potential capability and scalability with computer-simulated high-resolution results.

Index Terms—Three-dimensional displays, volume visualization, display algorithm, expectation maximization.

1 INTRODUCTION

The ultimate display may be thought of as a device that can reproduce any given light field (LF) or plenoptic function [1], [19], [26]. However, in order for such a hypothetical 4D LF device to display images at an adequate resolution, it would need a tremendous amount of bandwidth and processing power—capabilities that may not be available for many years [33]. In the meantime, 3D displays, which have recently become more affordable, can provide users with a more immersive visualization experience than compared with traditional 2D displays. While not the final goal, these new 3D displays are an important step toward the ultimate display.

Our work is concerned with a technique for improving the shortcomings of such systems, focusing on the Perspecta Display System [15] as a typical isotropically emissive volumetric display (IEVD) [9]. This system is based on a sweeping plane that performs 24 rotations per second, projecting volume slices at a high refresh rate.

Due to the absence of absorbing materials on IEVDs that would facilitate occlusion effects, the final image perceived by the viewer consists of a set of ray integrals, each of which is the sum of all emitted light along the ray. Hence, there is no physical means for displaying a completely opaque object in a conventional manner such that the lumiance of the occluded part is not perceived by the viewer [9]. Given a closed surface object, without occlusion, the ray that passes through the object will normally intersect with the surface at two or more points. The perception of each surface point is view dependent (see Section 3), even before taking into account coloring and shading. Thus, the creation of shading effects in an IEVD is a nontrivial problem, for which no existing solution can be found yet.

Unfortunately, occlusion and shading of object surfaces are common in real-world situations and provide optical cues for the shape of an object and its surface properties. As such, visualization on these IEVDs can be perceptually less intuitive. Fig. 1a shows a typical computer-synthesized isosurface of the UNC head data set as it would be displayed on a 2D screen. This work aims at reproducing such 3D rendering on IEVDs with viewpoint dependencies. When directly displaying the original data set on an isotropically emissive display, it results in an x-ray-like visualization, as shown in Fig. 1b. A naive attempt to display an isosurface by using a segmented data set would produce Fig. 1c. Clearly, it is still difficult to perceive surfaces in Fig. 1c. One imperative reason why Figs. 1b and 1c are less intuitive than Fig. 1a is that there is no surface shading nor occlusion present in Figs. 1b and 1c.

One suggestion for improving renderings is to pass a voxelized representation of a shaded surface to an isotropically emissive display. Unfortunately, as shown in Fig. 1d,
this does not alleviate the problem. Instead, it reduces the surface perception because in such a representation the shading effects are intended to be perceived correctly for an opaque surface. When removing the opacity-based occlusion, the accumulative ray integrals aggregate the light emitted from the shaded front surface, as well as all emitted light behind the surface. The shading effects that capture the continuity and curvature of the surface can no longer be correctly preserved and perceived.

To overcome these problems, we have developed a novel technique that enables users to visualize an isosurface with shading effects similar to Fig. 1a on an IEVD. As a result, we are able to generate, for the first time, a true 3D experience of an isosurface visualization with emulated shading effects on an IEVD, as shown in Fig. 1e, which is close to our original simulated results (Figs. 1a and 1f).

Fig. 2 illustrates the pipeline required to solve this rendering problem on IEVDs. We first render a 3D object or data set into an LF using a graphics or visualization system. This LF is then encoded inside a specific volumetric data set with the help of a reconstruction algorithm. We call this special volume data set a *lumi-volume*. When such a lumi-volume is fed into an IEVD, the accumulative ray integrals reproduce most shading effects on the viewer’s retinal image, as exemplified in Fig. 1e. In other words, the lumi-volume recreates an approximation of the original LF on the isotropically emissive display.

This pipeline can also accommodate an arbitrary LF captured by a plenoptic camera. Nevertheless, in this work, we focus on reproducing volume visualization and computer graphics renderings, which is the main application area of IEVDs. We employ an expectation-maximization (EM) algorithm to generate a lumi-volume from an LF, which minimizes the difference between the original LF and the LF rendered by the IEVDs.

To improve current imaging on isotropically emissive displays, we must first formalize the rendering principles behind IEVDs. Then, we examine various technical issues and limitations of the proposed technique. We also show that the loss of one dimension of information in approximating a 4D LF with a 3D lumi-volume will result in a more continuous visualization of partially visible objects through transparent surfaces. We examine the visualization quality in relation to LFs generated using both direct volume rendering (DVR) and nonrealistic maximum intensity projection (MIP). Our experimentation with a real IEVD system shows that this technique is effective and deployable, and our simulation study suggests that high-quality results are attainable.

2 RELATED WORK
2.1 3D Displays

Two-dimensional displays have a limited capacity in which to provide depth cues. Many 3D or stereo display technologies have been developed for enriching the users’ experience of depth perception. An overview of these developments can be found in the August 2005 issue of Computer.

The most widely used class of techniques for improving depth perception is that of stereo parallax, which provides each eye with separate images. This is a low-cost technique, only requiring the creation of two images per frame. However, wearing special glasses or goggles is often inconvenient and may lead to the users’ fatigue. Autostereoscopic displays [13], [28] were developed to circumvent these issues, and some multiview systems are able to horizontally multiplex many images. However, with such displays the object is always at a fixed focal depth, and the aspect ratio is only valid for a given depth.

On the other hand, volumetric displays [4], [5], [15], which use 3D pixels that absorb or emit light, do not suffer from the limitations of stereo parallax and autostereoscopic displays. One such device is the Perspecta Display System [15] (Fig. 3). This system uses a fast sweeping plane where
slices of a 3D volume are projected onto a rotating screen inside a hemisphere construct, providing a nearly \(4\pi\) steradian field of view (FOV). Since every emission of light on the sweeping plane will always be visible to the user, this system is classified as an IEVD, which provides x-ray-like renderings of the input data. This particular display has already been deployed in several applications [20], [35], [36].

Other types of IEVDs also exist, mostly based on fluorescent material that is excited by electron beams or lasers. Recently, some variants of the sweeping plane technique, using a mirror instead of a diffuse surface, have been proposed [9], [25]. The main advantage here is that the imaging is not limited to the isotropically emissive model. In the transpost system [30], the mirror is replaced by a screen with a limited viewing angle.

Another well-known way to produce 3D displays is holography [33]. This is an old technique that uses Fourier optics to fully display a 4D LF. However, the technique is limited by the huge amount of computation required to process the data and can only be handled by today’s supercomputers when high resolution is needed.

### 2.2 Light Field

An LF is a 4D function that associates a color to every unobtrusive ray of a 3D space. The concept was introduced in 1996 independently by Levoy and Hanrahan [26] and by Gortler et al. [19] to facilitate interactive browsing of an object without the need for rendering the object. For example, the lumigraph representation [19] uses two facing slabs to discretize the 4D LF space. A color is then associated with every possible pair of grid points chosen from both slabs, which actually defines a ray inside the LF space. Fundamentally, the LF notion is similar to the plenoptic function proposed in 1991 by Adelson and Bergen [1].

In our work, the color of a ray of light will be obtained very differently by combining all the voxels along the ray path according to a rendering model. Due to the fact that an LF data set consists of samples taken from a 4D domain, a huge amount of data is produced. Such data cannot yet be displayed by current technologies, nor is it compatible with the input representation required by IEVDs. As such, our technique approximates a 4D LF through a new 3D light-emitting volume representation referred to as a lumi-volume.

### 2.3 Algorithms for Volume Reconstruction

Since IEVDs are not capable of accurately reproducing all user-defined images, our goal is to create 3D renderings that approximate any arbitrary LF data that the user wishes to display. There are numerous voxel-based methods that reconstruct a 3D volume from an LF [11], [14], [21], [31], sometimes with the help of specific heuristics. In contrast, our method achieves the opposite, recreating an approximation of the original LF through volume rendering (using an isotropically emissive model in our case). Here, it is important to note that the resultant lumi-volume is not intended for general use and is only to be used on IEVDs that generate x-ray-like renderings.

To create a lumi-volume, we need to define a volume with projections that will match the given LF input as closely as possible. This parallels medical techniques that reconstruct 3D volumes from projections. The most prominent techniques are the filtered back-projection (FBP) reconstruction technique, the algebraic reconstruction technique (ART), and the EM algorithm. The first technique uses a Fourier transform, while the two others are iterative techniques that solve a linear system.

The FBP reconstruction method was the first technique used in computed tomography to reconstruct a volumetric representation of an object. However, the reconstruction equation is known only for specific cases, for example, cone beam reconstruction [16], and cannot be used for any arbitrarily positioned projection. The ARTs [18] utilize an iterative process, and at every iteration, an algorithm evaluates the difference between the synthesized projections of the current volume and the actual projections. The algorithm then uses this difference to produce a new refinement. The algorithm usually converges to a solution, provided that it exists (e.g., the data is not corrupt). However, the ART algorithm may find negative solutions, which are not suitable for this work since the voxels in a volumetric display cannot emit a negative amount of light. Therefore, this work is built on the EM algorithm [12], [32], which also employs an iterative process to solve a linear system. For self-containment, this algorithm will be briefly described in Section 4.

There has been a series of efforts implementing these algorithms on graphics hardware [37], including a hardware-assisted FBP implementation by Cabral et al. [6], an ART implementation by Mueller and Yagel [29] and Trifonov et al. [34], and a hardware-based EM algorithm by Chidlow and Moller [8]. More recently, there has been interest in applying tomography reconstruction techniques to handling LFs that capture transparent phenomena [24]. In [3] and [23], Ihrke et al. used tomography reconstruction to reproduce fire. More closely related to our work, [17] used tomography to model data sets. Their work, though limited to convex surfaces, showed that this approach is useful. Finally, Dachille et al. [10] applied the simultaneous ART (SART) method to recreate an LF. While the SART method may produce negative solutions (which is undesirable in
rendering displays in computer graphics and visualization. we shall first examine how IEVDs differ from traditional rendering displays in computer graphics and visualization.

Absorption and occlusion. IEVDs like the Perspecta have commonly been used to display voxelized 3D models, where the voxel values typically store the material properties of a model (e.g., a computed tomography data set) or a discrete boundary representation of an object (e.g., an isosurface). When such voxel values are mapped to luminance by the IEVD device, the light emitted from voxels arrive at the viewer’s location in a summative manner without any absorption or occlusion events. Although the resulting x-ray-like visualizations, as seen in Figs. 1b and 1c, can be interpreted by viewers, it is much different from the visualization of a real-life object on a traditional 2D rendering system, such as Fig. 1a. As such, it is highly desirable for a lumi-volume to encode voxel luminance in a specific way to compensate for the lack of absorption and occlusion. This requirement was recognized and conjectured as nonsolvable without physical modification of the display in [9].

View-dependent rendering. With traditional rendering of a 3D model on a 2D display, the illumination of each visible surface element of the model can be either view dependent (e.g., Phong, Blinn, and BRDF) or view independent (e.g., diffuse-only and precomputed radiosity). For view-dependent shading effects (e.g., specular highlight), the color of each visible surface element of the model is normally recomputed every time the viewpoint is modified. However, on an IEVD, the voxel luminance, if any, will always be view independent because of the lack of occlusion. The viewpoint modification occurs when the viewer moves around; however, the 3D data set passed to the display remains the same.

Hence, as shown in Fig. 1d, when a data set that encodes view-independent shading effects is passed to an IEVD, the results at different viewing positions can be very different from what is expected. In fact, at many viewing positions, the visualization can be quite incomprehensible. Thus, it is necessary for a view-independent lumi-volume to encode shading effects in an appropriate way such that the desired view-dependent shading would be partly maintained through the different view-dependent rendering integrals of an isotropically emissive display (Fig. 5).

Consistent brightness. A voxelized surface has a physical thickness. As illustrated in Fig. 4, the length of a ray-surface intersection $d/\cos(\alpha)$ is related to the angle of incidence, as is the brightness of the surface due to the accumulated light being proportional to the length of the intersection. Therefore, the wider the angle is, the brighter the surface appearance is, as demonstrated in Figs. 1c and 12a. This visual effect is a major obstacle when attempting to reproduce computer graphics on the volumetric display with voxelized scenes. Lumi-volumes address this difficulty by considering the most correct rendering solution for this task. To circumvent the above issues on a display with a fixed input data set, the ideal solution would be to output a real 4D LF that captures the visualization for all viewpoints. Nevertheless, the practical restriction of the 3D input data set used by IEVDs obliges us to seek an alternative solution, that is, to approximate a 4D LF with a lumi-volume.

In order to match the lumi-volume as closely as possible to the original LF, the problem must be first formalized in terms of viewpoint, LF, and line integrals. The algorithm for constructing a lumi-volume must then make use of all the possible volume samples, not just the ones across the surface, and work in a line integral domain, similar to the process of volume reconstruction in medical imaging.

For instance, if a surface element is represented as a voxel (which can only have a unique color), it is not possible to obtain different shading properties as the viewpoint changes (Fig. 5). If one now encodes the luminance properties along the entire light path instead of just considering the surface elements, which results in different line integrals, it is possible to create more independently shaded surface elements. When the lumi-volume is passed to an IEVD, it results in different visualizations for different viewpoints, collectively offering the best approximation of the wanted 4D LF.

There are, however, restrictions on the reproduced LF. For instance, the projection made according to one
Fig. 6. Projection and partial volume effect. \( b_i \) represents an LF element, \( x_j \) represents a voxel, and \( A_{ij} \) represents how much of \( x_j \) is intersected by \( b_i \).

viewpoint must be the same as the one made for the viewpoint in the opposite direction.

4 Expectation Maximization

The EM algorithm was proposed by Dempster et al. [12] and applied to medical imaging by Shepp and Vardi [32]. The main idea is to solve a linear system in an iterative way. Let \( b \) be an input LF (see Fig. 6), with \( b = (b_1, b_2, \ldots, b_M) \in \mathbb{R}^M \) being the set of observed variables inside the 4D LF, and let \( x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N \) be the unknown voxels of a lumi-volume that must be constructed as an approximation of \( b \). Due to the isotropic emission of the target displays, every projection \( b_i \) can be considered as a linear combination of the voxel values in the lumi-volume. Therefore, we can write our system as \( A \cdot x = b \), where \( (A_{ij}) \) is an \( M \times N \) matrix representing the partial volume effects. Each weight \( A_{ij} \) represents the volume of the intersection between \( b_i \) and \( x_j \). The EM algorithm, which iteratively converges to the solution of our linear system, can be expressed as follows:

\[
x_j^{n+1} = x_j^n \frac{\sum_{i=1}^{M} A_{ij} \Delta_i}{\sum_{i=1}^{M} A_{ij}},
\]

where \( n \) is the iteration number.

Each iteration of the algorithm can be decomposed into three main steps. The first step is to project the current lumi-volume onto a set of the viewing planes, resulting in a synthesized intermediate LF \( r = (r_1, r_2, \ldots, r_M) \in \mathbb{R}^M \), where \( r_i = \sum_{j=1}^{N} A_{ij} x_j^p \), and \( r_i \) is initialized to one prior to the first iteration.

The second step is to evaluate the difference between the input LF and the synthesized intermediate LF by calculating the ratios between \( b_i \) and \( r_i \) for all \( i = 1, 2, \ldots, M \).

The final step is to update every voxel \( x_j \) of the lumi-volume by multiplying its value with the mean deviation ratio of all the LF rays (or pixels) that intersect \( x_j \). The average must also be weighted by the partial volume effects.

Unfortunately, this algorithm is rather slow, with a complexity of \( O(m^3 \cdot p \cdot l) \), where \( N = m^3 \) is the total number of voxels in the lumi-volume, \( p \) is the total number of projections in the LF, and \( l \) is the total number of iteration steps. However, the convergence of this algorithm is linear, and a relatively low number of steps (\( \sim 20-80 \)) will usually result in an adequate approximation. To compensate for the high complexity level, numerous methods have been proposed to improve the speed of the algorithm. One such method is the ordered subset maximization method (OSEM) [22] that runs the algorithm on subsets, thereby improving the speed of convergence.

Converging to a solution usually makes sense only if there exists a solution. In our case, we know that there is no solution to our problem, since the input LF is a full 4D representation in a 4D space, and a bijection cannot be established with 3D space where the lumi-volume is defined. However, the EM algorithm has been used in statistics to find maximum-likelihood estimates, indicating that the algorithm may converge to a good estimate of the input LF by finding a minimum to \( (A \cdot x - b)^2 \).

5 Experimental Environment

In addition to the development of the pipeline shown in Fig. 2, we have conducted a series of experiments to study a number of technical issues in deploying our system. The main results of these experiments will be presented and analyzed in Section 6. In this section, we concentrate on the data flow depicted by the solid lines in Fig. 2, describing the environment and conditions for the experimental study, and presenting the models and target rendering results (Fig. 7) for benchmarking our approach using lumi-volumes.
5.1 Models and Rendering Methods

Fig. 7a was rendered using the DVR method, Fig. 7b was rendered using the MIP method, and Figs. 7c and 7d was rendered using the projection-based mesh rendering method. We utilized our own implementation of the DVR and MIP methods and OpenGL for mesh rendering. These three methods capture the characteristics of some of the most commonly used rendering algorithms in visualization and computer graphics.

MIP is a widely used volume rendering method in medical visualization. Since only the maximum value along a ray is kept, it provides selective visualization of the internal structure of data sets. This is often used in combination with contrast agents that aim to enhance the signal for the specific part of the data set under investigation. Although MIP does not provide realistic renderings, it presents an interesting challenge since it differs from the other two methods in many respects. On one hand, an LF resulting from MIP is more coherent since local maximums are likely to be distributed over many projections of the LF. For instance, the projection will always be the same for a specific direction and the opposite direction. On the other hand, MIP is fundamentally different from traditional surface or x-ray-like renderings.

Our LFs are computed from different rendering methods, and the lumi-volumes constructed in this work approximate a wide variety of rendering effects on an IEVD. Collectively, the results can offer more conclusive observations as to the usability of this approach and provide indicative conjecture as to its extensibility to other rendering methods.

5.2 Light Field Sampling

While the construction of a lumi-volume that would encode all information in an LF is fundamentally unattainable due to the dimension reduction, a careful selection of sampling parameters can significantly improve the performance of the EM algorithm in terms of both speed and accuracy. The main unknown attribute before starting this work was how much the RMS error will evolve with regard to the number of projections used. Chai et al. [7] and Lin and Shum [27] considered the efficient sampling of LFs, and their work offered useful guidance as to the optimal sampling rate.

To generate a coherent LF, one must evenly sample the 4D LF space \((\theta, \varphi, u, v)\), as illustrated in Fig. 8. This allows for the possibility of restricting the FOV of the LF to a smaller portion of the 4D space, enclosed by the imagery projections in the LF. We utilize polar coordinates to define the projection planes in the LF. Each projection direction is defined by two angles, \(\theta\) and \(\varphi\), that fall respectively in the ranges of \([-\max_{\theta}, +\max_{\theta}]\) and \([-\max_{\varphi}, +\max_{\varphi}]\). These two intervals are uniformly sampled by taking \(n\) evenly spaced samples for the angle \(\varphi\) and then taking \(n \cdot \cos(\varphi)\) evenly spaced samples for the angle \(\theta\). Objects are then rendered using an orthogonal projection for every sampled direction. The cosine term is required to avoid a biased higher sampling density at the poles.

5.3 Implementation of the EM Algorithm

Although the OSEM version can improve computation time by up to an order of magnitude and is relatively easy to implement, it makes the algorithm oscillate around the ideal solution. Therefore, this speed acceleration technique was not adopted by our experimental environment, and all results presented in this paper involve only lumi-volumes produced by the original EM algorithm. To improve the performance, we utilized a CPU-based implementation that has been optimized using SSE instructions. This was preferred to a hardware-accelerated implementation because we wanted some flexibility on the implementation side. As demonstrated by previous hardware-based implementations [6], [8], [37], it is feasible to further improve the speed of a volume reconstruction algorithm using hardware-based algorithms, which is not the focus of this work.

5.4 Perspecta Display

We conducted extensive experimentation of displaying constructed lumi-volume data sets on the Perspecta Display System version 1.7 (Fig. 3). The results, as shown in Fig. 1e and Section 6, are very encouraging when compared with other forms of volume data sets, such as those shown in Figs. 1b, 1c, and 1d.

The Perspecta system represents the current commercially available technology. It has a limited dynamic range. The sweeping plane does 24 revolutions per second, and 198 images (of resolution 768\(^2\) pixels) are displayed for each revolution, representing a total number of 117 million voxels. Since the Texas Instruments digital light processor can only perform approximately 8,000 binary state changes per second for each pixel, a color depth of approximately 2 bits per voxel is possible. The system makes use of dithering and halfoning to improve the color depth, which could improve the rendering capability but also may change the properties of our data sets. Unfortunately, we have no control over this feature. On the API side, an 8-bit voxel color depth is used, allowing 256 colors per voxel through the use of a palette. With only a few distinct intensities, the display can still produce useful visualizations (Figs. 1b and 1c) and can support lumi-volume visualization (Fig. 1e).

On the other hand, the lumi-volumes created with our method could require floating-point precision, with voxels having a high dynamic range. We expect that the effectiveness and usability of our method will improve with the
gradual introduction of high-resolution IEVDs in the near future. In addition, taking photographic pictures of the Perspecta system was challenging because a long exposure time must be used, which results in fuzzy images due to the internal vibrations of the device.

We have also provided high-resolution results produced from simulations in our evaluation in conjunction with the direct experimentation on a Perspecta display. The simulated results offer a scalable evaluation of the capabilities of our method.

5.5 Voxelization of 3D Meshes

In order to have a better idea of the rendering improvements brought to isotropically emissive displays, a simple voxelization of the mesh-based models used in this paper has been implemented. The voxelization was made with OpenGL by rendering a specific image of our LF (e.g., \(C_{18} = 0\) and \(C_{30} = 0\)) as many times as there are slices in the final voxelization. At each step, the front and back values of the frustum are changed in order to consider only the part of the model inside the given slice. The collection of all obtained images gives us the voxelization. Note here that voxel shading is always dependent on our reference viewpoint.

6 RESULTS

6.1 Quantitative Analysis of the Results

The different tests for assessing our method and its parameters are summarized in Table 1, including the following:

- the number of imagery projections in the rendered LF,
- the resolution of each imagery projection in the rendered LF,
- the size of the corresponding lumi-volume constructed,
- the FOV attributes, \(\max{\theta}\) and \(\max{\phi}\),
- the sampling rate,
- the runtime in seconds per iteration, and
- the number of iterations used to construct the lumi-volume.

An AMD athlon 64 X2 3800+ (2 GHz) was used for the construction of the lumi-volumes, but only one core was used. The UNC head data set has undergone several tests to analyze the effect of the FOV in generating LFs. In various “Head 1D” tests, the projections in an LF were generated by varying the horizontal FOV (\(\max{\theta}\)), while fixing the vertical FOV (\(\max{\phi}\)) to zero. The “Head 2D” tests involved varying both polar angles.

We evaluate the quality of a lumi-volume by measuring the RMS error of the LF reconstructed by our lumi-volume. The RMS error considers all the samples of the original LF \(b_i\) and computes the difference between the reconstructed LF \(r_i\) at the original sampling locations using the following formula:

\[
\sqrt{\frac{1}{M} \sum_{i=1}^{M} (r_i - b_i)^2}
\]

The measurements are based on a 256-gray-level integer scale. Fig. 9 shows the progressive changes of the RMS error during the iterations of the EM algorithm.

The head data set has undergone several tests to analyze the effect of the FOV on the reconstruction. The Head 1D tests (Fig. 9a) have been conducted by adjusting the horizontal FOV (\(\max{\theta}\)), while the vertical FOV (\(\max{\phi}\)) was set to zero. Note that even with a null vertical FOV, our problem still remains unsolvable. The Head 2D and bunny tests (Fig. 9b) have been realized by adjusting both angles.

First, in Fig. 9, one can easily observe that the algorithm appears to converge to a solution as expected. Here, we see that lumi-volumes produced using smaller FOVs tend to result in decreasing RMS errors after each iteration of the EM algorithm. However, for some of the input LFs created with a wider FOV (e.g., Head 1D 70-90 degrees, Head 2D 40 degrees, Head 2D MIP 90 degrees, Aneurism MIP), the RMS error reaches a local minimum and then begins to increase.

In practice, halting the iteration process right after the local minimum does not necessarily produce a lumi-volume that will lead to the best visual results. More iteration steps always seem to lead to sharper reconstructed LFs in our experiments (Fig. 16, images 4e, 5a, and 5b).

Second, one can also observe that the RMS error is lower with lumi-volumes for MIP-rendered LFs (Fig. 9c). This is because MIP provides a more coherent rendering, as

---

**TABLE 1**

<table>
<thead>
<tr>
<th>Test</th>
<th>Dataset</th>
<th>Original Light Field Image Size</th>
<th>Light Field Image Size</th>
<th>(\max{\theta})</th>
<th>(\max{\phi})</th>
<th>Sampling ((n))</th>
<th>Time (s) per OSEM iteration</th>
<th>Number of OSEM iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bunny RGB</td>
<td>804</td>
<td>320(^2)</td>
<td>416(^3)</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>9600</td>
</tr>
<tr>
<td>2</td>
<td>Aneurism MIP</td>
<td>536</td>
<td>192(^2)</td>
<td>320(^3)</td>
<td>90</td>
<td>90</td>
<td>30</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>IEEE TVCG</td>
<td>804</td>
<td>320(^2)</td>
<td>384(^3)</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>4800</td>
</tr>
<tr>
<td>4</td>
<td>Head 1D, (\max{\theta} = x) degrees</td>
<td>(x)</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>130</td>
</tr>
<tr>
<td>5</td>
<td>Head 2D, 10 degrees</td>
<td>90</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>746</td>
</tr>
<tr>
<td>6</td>
<td>Head 2D, 20 degrees</td>
<td>378</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>2600</td>
</tr>
<tr>
<td>7</td>
<td>Head 2D, 40 degrees</td>
<td>1444</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>2200</td>
</tr>
<tr>
<td>8</td>
<td>Head 1D MIP, (\max{\theta} = y) degrees</td>
<td>(x)</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>130</td>
</tr>
<tr>
<td>9</td>
<td>Head 2D MIP, 90 degrees</td>
<td>1237</td>
<td>256(^2)</td>
<td>320(^3)</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>2600</td>
</tr>
</tbody>
</table>

For tests 4 and 8 (Head 1D), \(\max{\theta}\) varies in the range [10..90] with a 20-degree increment between each configuration.
discussed previously. For example, comparing the RMS errors for the UNC Head 1D tests in the top and bottom graphs, we can see clearly that MIP-rendered LFs are better approximated by lumi-volumes than DVR-rendered LFs. Third, we can also observe that the quality is affected by the scene complexity of the original LF. For example, the aneurism MIP data set produces a more complex LF than those LFs of the MIP-rendered UNC Head, thus resulting in a less accurate lumi-volume. The UNC Head 2D test has a higher degree of freedom than those UNC Head 1D tests in producing an LF, which also results in higher RMS errors.

6.2 Qualitative Assessment of the Results

This section provides a visual assessment of the results in several different aspects. The videos that accompany this paper also support the qualitative visual analysis by juxtaposing some of the original input and the corresponding reconstructed LFs. Some screen captures of the displays showing the effectiveness of our technique on different viewpoints are represented in Fig. 10. However, due to the limitation of the display, we will mainly discuss the synthetic results.

6.2.1 The Transparency Effect

Because of the lack of occlusion on IEVDs, real or simulated, one inevitable artifact is the transparency effect. A surface that is opaque in the original input LF is likely to become translucent in the reconstructed LF. This is noticeable with the Head 1D example, where varying the FOV parameter \( \text{max}_{\phi} \) from 10 to 180 degrees clearly increases the transparency of the isosurface (Fig. 16). This is also the case with the MIP LFs, and, especially, for the aneurism MIP data set (Fig. 16, image 5d). Note that this is more clearly visible in the comparative videos.

To emphasize this effect, we have created a 3D scene (Fig. 7d), where the 3D text is partially hidden by a set vertical columns. In the reconstructed LF (Fig. 11b) obtained from a lumi-volume, the entire text is now visible, because the columns have lost their capacity for occlusion and appear to be translucent. In simulation, the summation of all luminance along a ray may also result in a saturated pixel color, due to the color quantization (e.g., the 256 gray scales). Nevertheless, this should not be the case with isotropically emissive displays since the human eyes have a higher dynamic range. Although one may solve the saturation problem by scaling down the pixel intensities, we have chosen not to reduce the brightness of any images in this paper in order to make a consistent visual comparison of the results.

The apparent translucency of an opaque surface is an inevitable artifact of IEVDs. As demonstrated in Fig. 1, in comparison with the naive approach for passing raw or segmented volume data sets to an IEVD, our results have actually shown the removal of a significant amount of transparency. As such, the use of lumi-volumes offers a tangible solution to applications where excessive levels of apparent translucency is highly undesirable. This is also supported by the direct comparison of the lumi-volume-reconstructed LFs (Figs. 11 and 12) with the LFs produced from an OpenGL voxelization of the same 3D models. In these two cases, transparency is clearly reduced with our technique, though not completely removed.

The MIP case is also very interesting (Fig. 16). The main MIP structures are now transparent, which is useful for distinguishing several overlapping objects. This could represent a nice extension to MIP, but a user study will be required to determine the practicality of such an extension.

Figs. 12d and 12e show two cross sections of our bunny lumi-volume and illustrate the main difference between our method and former volumetric reconstruction methods [29] that produce a voxelization of the scene. Here, the lumi-volume stores the required information inside the entire
data set, not only on the surface of the model. As such, the data set is of no use in regular rendering applications.

In many visualization applications, a small amount of transparency effects can actually assist viewers in their visual interpretation of the internal structures or occluded parts of the models being observed on IEVDs. The advantage of our method is that the reconstructed rendering may allow users to interpret the occluded parts from the front surface. For instance, Fig. 11b shows the clearly discernable text, *IEEE TVCG*, which is not distinguishable in the original LF. We could, for instance, further imagine an LF acquired from an airplane that would be processed with our technique to discover what is hidden below trees.

6.2.2 Rendering Fidelity
The dimension reduction from a 4D LF to a 3D lumi-volume also leads to some loss of high-frequency details, as shown on the enlarged images in Fig. 12. In this example, a maximum FOV of ±40 degrees was chosen for both \( \theta \) and \( \phi \) angles, which corresponds to an approximate solid angle of 1.7 steradian. In this case, some high-frequency texture details, as well as some specular lighting effects, have partially faded. This attenuation also tends to get worse if we increase the FOV of the LF, especially for opaque surface renderings. Increasing the size of the lumi-volume does not seem to alleviate this problem (Fig. 14). Color fading is also visible with MIP-rendered LFs but to a lesser extent.

Despite the loss of some high-frequency details in Fig. 12c, we can observe that the contours of the textures are actually quite well preserved, and the overall perception of the surface with lumi-volumes is significantly superior to a raw voxelization, as shown in Fig. 12a. To be more general, all our examples clearly show that the reproduced LFs create an adequate visual approximation of the original LFs, despite the inherent limitations of such a technique. In other words, unlike results that would be obtained from a simple object voxelization passed to the display, results produced from lumi-volumes are relatively close to what would be displayed on a 2D screen, which is the main issue this paper intended to address.

However, for realistic renderings, the FOV needs to be restricted to a given solid angle, and it seems that a ±40-degree FOV is a reasonable choice. Since these lumi-volumes are created with a particular FOV, data can only be interpreted on the Perspecta Display when users stand at the correct viewing angle. If users move outside of the appropriate viewing location, the volume becomes cloudy, and the visualization becomes incoherent (Fig. 13).

6.2.3 Sampling Rate Analysis
To evaluate the sampling quality of an original LF, we rendered the UNC head data set at different sampling rates on \( \theta \) (5, 10, 20, 40, and 80 images of 192 pixels), with \( \text{max}_\theta \) set to 40 degrees. Through simulation, we observed the reconstructed LFs at two consecutive sampling angles, as well as at an intermediate angle. As shown in Fig. 15 for the lowest sampling rate (with five imagery projections, two consecutive images differ from an angle of 20 degrees), one can see that the reconstruction from lumi-volumes is very good at the original sampling locations, with no visible transparency effect. However, the reconstruction at intermediate locations is far from satisfactory. By increasing the sampling rate, the reconstruction at intermediate locations becomes better and better. When this phenomenon is transferred to real displays, it means that viewers can experience a more continuous visualization between two consecutive sampling angles.

We also noticed that ringing artifacts appear if the sampling rate is too low. We have found that using 80 samples (when images differ by an angle of 1 degree) appears to be good enough in this particular case. Finally, the reconstructed LF converges to a more transparent rendering solution when the sampling rate is increased.

We have made similar observations with other data sets. From our experience, increasing the sampling rate always seems to result in a better visualization, and the EM algorithm also converges more consistently to a
specific lumi-volume. However, a high sampling rate does require more time in the construction of a lumi-volume.

6.2.4 Lumi-Volume Sampling Rate
We have also evaluated the possible impact of using larger lumi-volumes on the rendering by varying the size from $256^3$ to $512^3$ voxels. The same input LF (Head 1D, $max_\theta = 40$ degrees, 80 projections) was used, and the reconstructed LFs were observed. Four selected projections were shown in Fig. 14.

Our observation indicates that the lumi-volume size has relatively little impact on the rendering fidelity and the transparency effect. The intrinsic properties of our lumi-volume are still the same. The only effect we have noticed is that the reconstructed LF appears to become smoother when the size increases, which is as expected.

6.2.5 Overall Remarks on the Results
Our experimental study shows that our new technique has managed to simulate traditional rendering effects on isotropically emissive displays, which was once thought to be impossible [30], [9]. Computer graphics images can now be shown on emissive displays with almost the same appearance as images found on a basic 2D screen.

More importantly, surface shading effects can now be produced on IEVDs, and the apparent translucency effects of supposedly opaque surfaces can be controlled at a reasonable level, provided that an appropriate FOV is chosen. As long as a lumi-volume is generated from an LF consisting of a sufficiently large number of imagery projections, viewers can have a seamless 3D experience with full depth perception inside the FOV. This result can be compared with the very recent work of Jones et al. [25] where the diffuse surface of the display has been replaced.
by a mirror covered with a holographic diffuser, similar to [9]. Both methods have specific advantages and disadvantages. For instance, the user can experience a full 360-degree horizontal FOV in [25] but at the cost of positioning the user at a predefined distance $d$ of the display and using head tracking to ensure correct vertical parallax when required. Shading is also very limited since this new display can only work with binary images. In our case, the contribution of several voxels to a ray allows having more gray levels, even if the Perspecta display is also limited with respect to color quantization.

Hardware modification will be needed to further improve rendering quality, increase the FOV, and reduce transparency effects. While some recent approaches use anisotropic spinning surfaces [30], [25], [9] to achieve this goal, we plan to study the use of multiple spinning diffuse surfaces. For instance, a different image could be projected on both sides of our Perspecta Display’s spinning plane, leading to the use of two different lumi-volumes.

7 Conclusion

IEVDs offer an intuitive means for data visualization and exploration. However, computer graphics on these displays has been limited in the past to sending either voxelized 3D meshes or medical data sets to the display, resulting in low-fidelity renderings. This paper has offered a way to alleviate this issue by formalizing, for the first time, the rendering problem on isotropically emissive displays and introducing the correct rendering pipeline to produce lumi-volumes.

We have provided the first set of experimental evidence to support the usability and scalability of this new technique. This finding significantly broadens the application domain of isotropically emissive displays and is expected to simulate further research and deployment of the volumetric display technology.

Our simulation indicates that the lumi-volume technique can scale as the resolution and bandwidth of volumetric display technology continues to increase. We are looking forward to the continued enhancement of commercial volumetric display systems such as the Perspecta system.

Our future work will include investigation into faster reconstruction algorithms, investigation into reconstruction algorithms that are not EM based, multiplane display simulations, using head tracking, compression methods for lumi-volumes, and perceptual merits of enhanced transparency in MIP-based visualization on IEVDs.

Acknowledgments

The authors wish to thank Gregg Favalora and Joshua Napoli (Actuality Systems) for helping them with the use of the Perpecta Display SDK and the Purdue University Envision Center for giving them access to a Perspecta display. This work was supported in part by a grant from EPSRC (EP/E001750/1).
REFERENCES


