Projection Volumetric Display using Passive Optical Scatterers

Shree K. Nayar and Vijay N. Anand
Department of Computer Science, Columbia University

Technical Report: CUCS-030-06
July, 2006

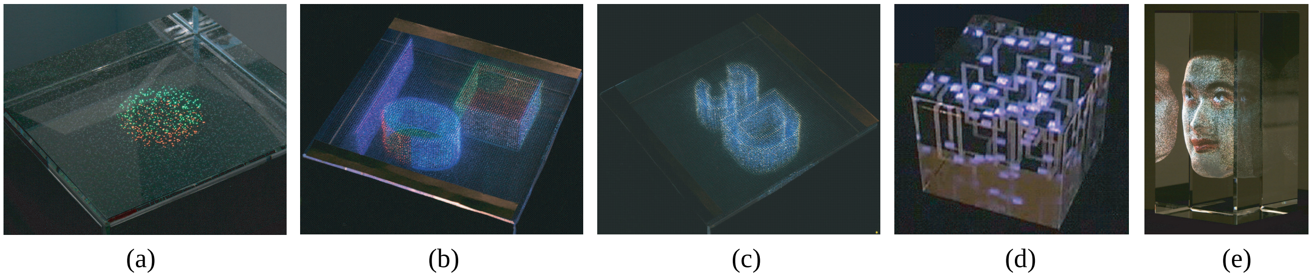


Figure 1: A class of low-cost volumetric displays are proposed that use a simple light engine and an indexable 3D point cloud made of passive optical scatterers. The point clouds are etched into a glass block using Laser Induced Damage (LID). Displays with different resolution characteristics have been developed that can render (a) true 3D objects, (b) extruded objects with arbitrary top layers and (c) purely extruded objects. The same approach has also been used to (d) extend the game Pac-Man to 3D and (e) create a 3D avatar.

Abstract

In this paper, we present a new class of volumetric displays that can be used to display 3D objects. The basic approach is to trade off the spatial resolution of a digital projector (or any light engine) to gain resolution in the third dimension. Rather than projecting an image onto a 2D screen, a depth-coded image is projected onto a 3D cloud of passive optical scatterers. The 3D point cloud is realized using a technique called Laser Induced Damage (LID), where each scatterer is a physical crack embedded in a block of glass or plastic. We show that when the point cloud is randomized in a specific manner, a very large fraction of the points are visible to the viewer irrespective of his/her viewing direction. We have developed an orthographic projection system that serves as the light engine for our volumetric displays. We have implemented several types of point clouds, each one designed to display a specific class of objects. These include a cloud with uniquely indexable points for the display of true 3D objects, a cloud with an independently indexable top layer and a dense extrusion volume to display extruded objects with arbitrarily textured top planes and a dense cloud for the display of purely extruded objects. In addition, we show how our approach can be used to extend simple video games to 3D. Finally, we have developed a 3D avatar in which videos of a face with expression changes are projected onto a static surface point cloud of the face.

Keywords: volumetric display, 3D display, projection, passive scatterers, point clouds, laser induced damage, visibility, rendering, 3D video games, 3D avatars.

1 Introduction

Systems for displaying images and videos have become a ubiquitous part of our everyday lives. Televisions provide viewers with news and entertainment. Home entertainment systems with large projection displays allow home viewers to enjoy theater-like experiences. Displays used in computers and cellular phones enable users to interact with a variety of devices and access various forms of information. High quality digital displays have also emerged as possible replacements for physical media such as photographs and paintings.

Despite significant advancements in display technologies, most conventional systems only display 2D images. A viewer, however, experiences the real world in three dimensions by perceiving depth in addition to the horizontal and vertical dimensions. Because 2D images do not contain depth information, they appear flat and less realistic to a viewer. A system that can display static and dynamic 3D images is therefore of great value. In many applications, it is also desirable that the display system simultaneously provide different perspectives of a 3D scene to viewers who are at different locations with respect to the display. Such a system would also allow a viewer to move around the display and gain different perspectives of the scene.

The creation of a device that can display photorealistic 3D content at very high resolution may be considered a holy grail problem that has been pursued with great vigor over the past century. The goal of our work is less ambitious. It is to develop a very inexpensive class of volumetric displays that can present at relatively low resolution, and yet in a compelling way, certain types of 3D content. The types of content we wish to display are simple 3D objects, extruded objects and 3D surfaces that appear dynamic when projected with time-varying images. Our displays use a simple light engine and a dense cloud of passive optical scatterers. The basic idea is to trade off the (2D) spatial resolution of the light engine to gain resolution in the third dimension. The simplest way to achieve such a trade-off is to use a stack of planar grids of scatterers where no two stacks overlap each other with respect to the projection rays of the light engine. However, such a semi-regular 3D grid suffers from poor visibility; as the viewer moves around the point cloud the fraction of points visible to the viewer varies dramatically and is very small for some of the viewing directions. Our key insight is to randomize the point cloud in a manner that is consistent with the projection geometry. We show that when a point cloud is randomized in a specific manner it produces a remarkably stable visibility function.

We have explored several ways of creating dense clouds of passive scatterers. We have chosen to use a technology called Laser Induced Damage (LID) that can very efficiently, precisely, and at a very low cost embed the desired point clouds in a solid block of glass. Each scatterer is a physical crack in the glass that is created by focusing a laser beam at the point. When a point in the cloud is lit by ambient light it is barely visible, but when it is lit by a focused source it glows brightly. We have studied the radiometric and spectral characteristics of LID scatterers and found that they have the properties needed to display objects in color and with sufficient brightness to be viewed within a 120 degree cone that is aligned with the projection direction. For illuminating the scatterers, we have developed an orthographic light engine that uses an off-the-shelf projector and inexpensive optics to create parallel rays with a large footprint. Orthographic projection enables us to use point clouds without resolution biases and makes the calibration of the display relatively straightforward.

We have developed several versions of our volumetric display, each one designed to meet the needs of a specific class of objects or a specific application. We have implemented point clouds with $10,000$ points for the display of true 3D objects (see Figure 1(a)), $190,500$ points for the display of extruded objects with arbitrarily textured top surfaces (see Figure 1(b)), $180,500$ points for the display of purely extruded objects (see Figure 1(c)), $83,866$ points for the extension of the game Pac-Man to 3D (see Figure 1(d)), and $127,223$ points for the face model used to create a 3D avatar (see Figure 1(e)).

2 Related Work

We are interested in developing 3D displays that enable multiple viewers to view the content simultaneously from different locations in space. In addition, we do not wish to require the viewers to wear glasses (spectral or polarization) as this is not desirable in most applications. Several display technologies have been developed that share the same goals. These include autostereoscopic displays that use lenticular lenses and parallax barriers, holographic displays that generate 3D or 4D light fields and volumetric displays with and without moving parts. Excellent surveys that discuss these different approaches as well as their merits and limitations can be found in books [Okoshi 1976; Blundell and Schwartz 2000] and recent articles [Perlin et al. 2000; Matusik and Pfister 2004]. Here, we will only compare our approach to previous volumetric displays, with emphasis on those that do not use moving parts.

The benefit of volumetric displays is that, since they use a physical volume to render the content, the three-dimensionality of the content is naturally perceived with all its cues (parallax, focus, etc.). The use of a physical volume also allows simultaneous viewing of the content by multiple observers from a wide range of directions. A key limitation of volumetric displays is that the volume elements do not block each other and hence occlusion effects cannot be rendered. Since our approach is a volumetric one, this limitation is inherent to it as well.

A comprehensive survey of volumetric displays is presented in [Blundell and Schwartz 2000]. In the case of a swept volume display, the display volume is created by the mechanical (rotational or vibrational) motion of a target screen. The speed of screen motion is kept high enough to not be perceived by the observer. Early versions of this approach have used a spinning mirror that reflected images displayed on cathode ray tubes [Parker and Wallis 1948] and a rotating phosphor-coated screen [Blundell et al. 1994]. A recent commercial product called Perspecta [Actuality Systems] uses a diffuse spinning screen and a digital projector. Swept volume displays produce impressive results but suffer from the drawback of using large moving parts (screen and/or sources) that makes them hard to scale. A static volumetric display can create a display volume without using mechanical motion. Previous implementations have used fluorescence excitation of gases [Schwarz and Blundell 1993] and infrared laser excitation of fluorescent metallic particles [Downing et al. 1994]. These systems are truly novel but at the present stage are expensive and do not produce the scale and resolution needed in many applications. In comparison, our displays are static and yet are very easy and inexpensive to implement as they do not use special gases, materials or sources.

In spirit, the closest work to ours is perhaps the static volumetric display developed by MacFarlane [MacFarlane 1994], which uses a controllable light source and a 3D array of voxels doped with a fluorescent dye to create a 3D image. Optical fibers are used to guide light from the source elements to the voxels. The voxels and fibers are immersed in a refractive index matching liquid to avoid refraction artifacts. Systems with 76,000 and 200,000 voxels have been implemented. While this is an interesting system, it is cumbersome to implement and calibrate. Using just passive scatterers, our displays achieve similar resolutions and are easy to implement. A recent commercial product called the DepthCube [LightSpace Technologies] is a static volumetric display that uses a stack of 20 scattering LCD sheets that are illuminated in sequence by a high-speed digital projector. This display produces very compelling 3D content. However, it is expensive as it uses a high-speed light engine and electrically controlled LCD scatterers. Our displays have similar or better depth resolution but lower resolution in the other two dimensions. However, they are much less expensive as they use passive scatterers.

Finally, an interesting proposal has been made in [Perlin 2004] to use dust particles suspended in air to project 3D images. The idea is to scan the dust using an infrared beam and a detector to find the locations of the particles and then use a visible light scanning beam to light up the appropriate particles. This idea is similar to ours in that it uses passive scatterers. However, since the locations of dust particles are unknown at any given time, they must be first detected and then illuminated using a very fast and precise scanning process.

3 Indexable Point Clouds and Visibility

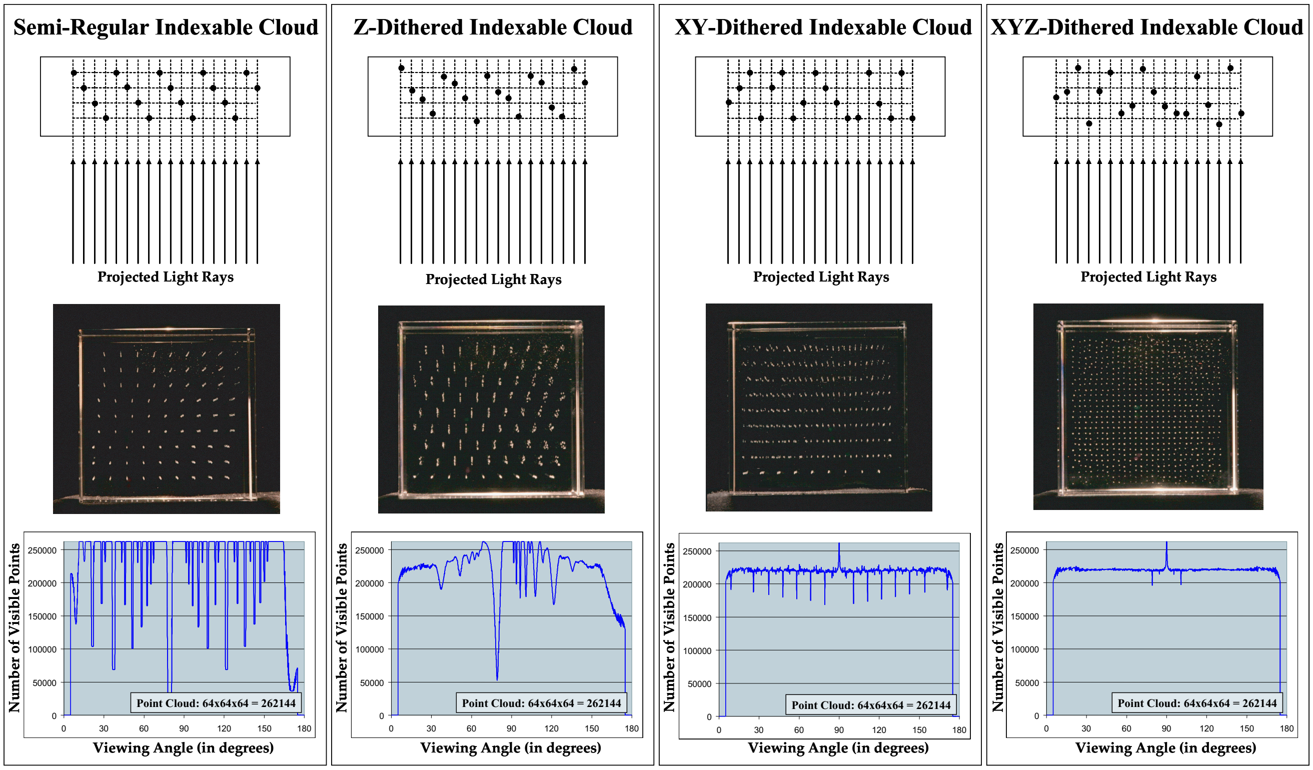
Let us assume that our projector is orthographic, in that, all its light rays are parallel to one another. We have used such a projector in most of our implementations. The advantage of orthographic projection, over the commonly used (perspective) projector, is that it makes the display easy to calibrate and the resolution of the 3D point cloud does not vary along the projection direction. Furthermore, the display volume can be a cuboid rather than section of a pyramid.

An indexable point cloud is one where each of the points (passive scatterers) can be lit up by a light engine without lighting up other points. Such a configuration is essential for displaying arbitrary 3D objects. There are many ways to create an indexable point cloud. However, critical to the choice of the point cloud is the notion of visibility. As the viewer moves around the display, we wish to ensure that the percentage of points that are (at least in part) visible to the viewer does not change dramatically with the viewing direction. Ideally, we would like the visibility to be high and constant with respect to the viewing direction.

To compare the visibilities of different point cloud configurations, we have conducted OpenGL simulations[1] using n^3 points where $n = 64$. Each point cloud was assumed to be within a glass cube (refractive index of 1.5) and each point was modeled as a sphere of diameter 0.5mm. We denote the planes parallel to the projector's image plane as XY planes and the third dimension, which is the direction of light projection, as Z. When all the spheres in a cloud are projected (orthographically) onto a single XY plane, the distances between adjacent spheres in the X and Y directions is 0.6mm. Therefore, the average[2] distance p between adjacent spheres on any given XY plane is $0.6\sqrt{n}$ mm. This was also used as the distance q (along Z) between adjacent XY planes. The viewer was assumed to be at a very large distance from the display and moving along a single plane that was perpendicular to the XY planes. The viewing angle was varied from 5° to

[1] It is hard to analytically derive the visibility function of a point cloud since we are interested in knowing if a point is even partially visible. In fact, even for the simplest point distributions and extreme visibility assumptions, the derivation can be daunting (for example, see [Kostinski 2001]).

[2] As we will see, p is constant in the case of a semi-regular point cloud but can vary for randomized point clouds.



(a) (b) (c) (d)

Figure 2: Visibility functions (bottom row) for point clouds (top row) with (a) semi-regular, (b) Z-dithered, (c) XY-dithered and (d) XYZ-dithered grids. These functions were computed via simulations. The clouds have 64^3 points and for any given viewing angle a point is deemed visible if 25% of its projected area is visible to the viewer. The middle row shows real $9\times9\times9$ point clouds etched in glass cubes using LID.

175° in steps of 0.2°, where at 90° the viewing direction is exactly opposite to the direction of light projection. For each viewing angle, an image of the point cloud was rendered with each sphere assigned a unique color and a sphere was deemed to be visible if more than f percent of its entire unobstructed area[3] was visible.

Semi-Regular Indexable Cloud: The simplest indexable cloud is shown (in 2D) at the top of Figure 2(a). The cloud is illuminated from below by the orthographic projector and is viewed from the upper hemisphere. In this cloud, the XY planes have regular grids of points and the grid for one plane is simply offset in X and Y by a fixed distance with respect to the next plane. The offsets ensures that each point in the cloud can be uniquely indexed by the projector. In the middle of Figure 2(a) is shown a physical glass cube with a $9\times9\times9$ semi-regular indexable cloud. The visibility plot (with $f = 25\%$) for this cloud is shown at the bottom and clearly illustrates the problem with using such a semi-regular cloud. As the viewing angle changes, a single XY grid can occlude one or several other XY grids beneath it. Therefore, the visibility fluctuates dramatically with viewing direction, indicating poor display performance[4].

Z-Dithered Indexable Cloud: This point cloud, illustrated in Figure 2(b), is identical to the semi-regular one, except that the Z coordinate of each point is dithered using a random distribution. In our simulation, we have used a uniform distribution over ±1mm[5]. This simple modification to the regular grid greatly improves the visibility function. For instance, one can see many more points in the image of the glass cube with $9\times9\times9$ points. Overall, the visibility function is much smoother but for some viewing angles the visibility is very low, for example, at 80°.

XY-Dithered Indexable Cloud: Consider the first 4 (from left) points in the semi-regular cloud in Figure 2(a). These points lie within a vertical tube through the point cloud. In the case of an n^3 cloud, the tube would have an XY cross-section of $p\times p$ and a length along Z of $(n-1)q$. Again, p is the distance between adjacent points in a single XY grid and q is the distance between adjacent XY grids in a semi-regular cloud. Within this tube, there is no reason why the points must be offset in XY in a regular fashion as in Figure 2(a). In fact, such an offset gives the point cloud a strong structure which is visible and distracting for the viewer. We only need to ensure that the full range of Z values are included within the tube. Therefore, within each tube, for each Z value we can randomly assign (without repetition) the XY coordinate from the ones in the semi-regular grid. Since there are n points in the tube, there are $n!$ possible XY assignments (permutations). Therefore, for any reasonable value of n, it is very unlikely that any two tubes in the cloud would have the same assignment. Note that such a random assignment of XY coordinates is not the same as the pure randomization of the Z coordinate in the Z-dithered case. A pure randomization of XY coordinates would place points outside the regular structure of the light rays produced by the light engine, while a random assignment does not.

Simulation results for an XY-dithered point cloud are shown in Figure 2(c). We see a significant improvement in the visibility function, which is almost constant over the entire range of viewing angles except for small, sharp dips that are more or less at regular intervals.

XYZ-Dithered Indexable Cloud: This point cloud, shown in Figure 2(d), simply combines the ideas behind the previous two clouds. It uses randomization of the Z coordinate and random assignment

[3] Since the glass cube has a different refractive index from that of air, the unobstructed area varies with viewing angle due to refraction. This effect was taken into account in our calculation of point visibility.

[4] Note that for any indexable point cloud, the visibility is 100% for $\theta = 90°$ as this direction is just the opposite of the projection direction for which the cloud is designed to be fully indexable.

[5] In our implementations, we have chosen the range such that the Z coordinates of adjacent XY grids do not interfere with each other.

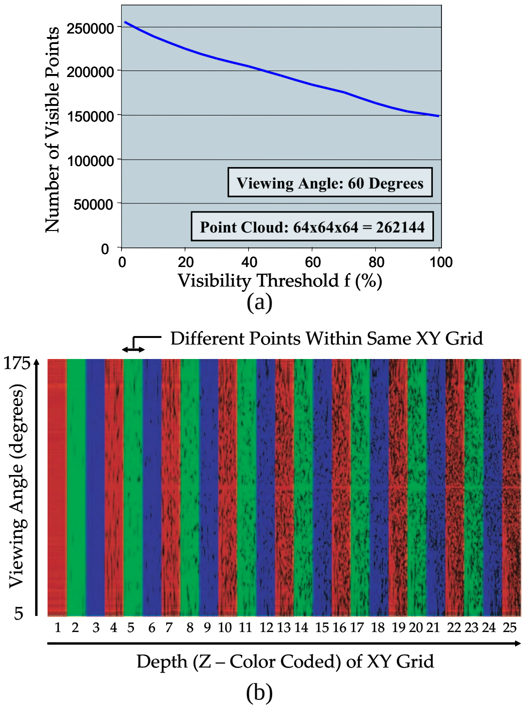


Figure 3: (a) Visibility plotted as a function of the threshold f used for determining point visibility. (b) A color-coded image that illustrates how visibility varies with the depth Z of the XY grid.

for the XY coordinates of each point in the cloud. When this is done, the small dips in the visibility function of the XY-dithered cloud disappear and we have a remarkably stable visibility function where roughly 85% of the points are always visible to the viewer.

We have conducted several other simulations as a part of our visibility analysis. The above visibility functions were computed for viewing angles in a single plane that is aligned with the projection direction. We computed visibility functions for several different rotations of this plane about the projection direction and verified that the above plots remain essentially the same irrespective of the chosen plane orientation. In addition, we repeated the simulations for point clouds with $n = 25$ and $n = 100$ and found that the plots are very similar to those for $n = 64$ except that the high frequency jitter (noise) in all the plots was greater for $n = 25$ and lesser for $n = 100$, as would be expected. We also studied visibility as a function of the percent f of the projected area of a sphere that is used as the threshold for visibility. In Figure 3(a), we see that the visibility falls almost linearly as f increases. Here, $n = 64$ and viewing angle = 60°.

Thus far, we have defined visibility as the fraction of points in a cloud that are visible. Clearly, the points in the XY layers that are deeper in the cloud are more likely to be occluded. Figure 3(b) shows as an image the visibility of each point in a cube with $n = 25$. The colored stripes correspond to individual XY grids at different depths Z, where $Z = 1$ is the top grid that is closest to the viewer. The columns within each colored stripe correspond to different points in a thin slab of width p that is aligned with the plane of viewer motion. Note that all such slabs will have the same visibility properties. The vertical axis represents the viewing angle. A dark point in the image represents a cloud point that is not visible. As expected, all points in the top XY grid are visible over all viewing angles, while the deeper XY grids have more obstruction (dark points). Again, at 90° all points in the cloud are visible as illustrated by the thin horizontal bright line that runs through the middle of the image.

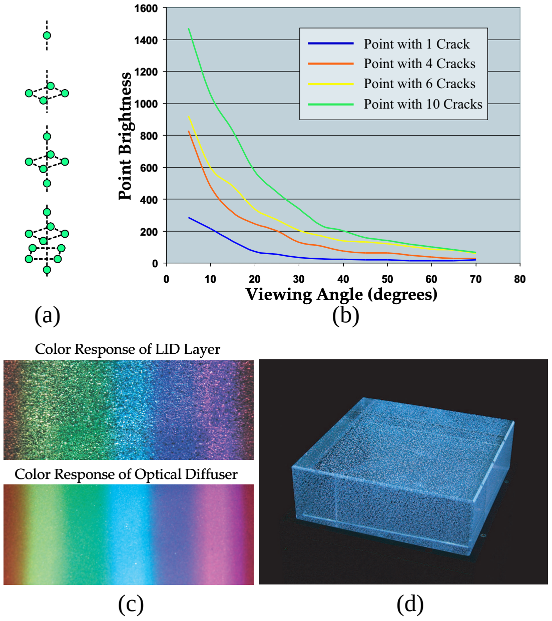


Figure 4: (a) Each point in a cloud can include multiple LID cracks to increase its brightness. (b) The brightnesses of the four points with different crack configurations in (a) plotted as a function of the viewing angle. (c) Color response of a dense layer of LID cracks and an optical diffuser. (d) An XYZ-dithered cloud of $48 \times 48 \times 25$ points in a $200 \times 200 \times 70$mm glass cube, uniformly lit by a projector.

4 Point Clouds with Laser Induced Damage

There are many ways to fabricate a point cloud with passive scatterers. One approach is to embed or etch the scatterers on glass or plastic sheets and then stack the sheets to create a volume. Each sheet serves as a single XY grid and the number of sheets determines the resolution in depth Z. The scatterers can be specular balls, diffuse painted patches or etched patches. With specular balls and painted patches, the projection of light rays must be from the top (opposite to that in Figure 2) as the scatterers are opaque. On the other hand, etched patches act like diffusers and hence the projection can be from the bottom. Our initial intent was to use a stack of glass sheets with grids etched on them. Unfortunately, it is cumbersome to align the sheets and they must have strong anti-reflection coatings to avoid glows due to interreflections between the sheets. More importantly, the use of planar sheets does not lend itself to Z-dithering of points.

A precise, efficient and yet inexpensive way to create a complex point cloud is based on Laser Induced Damage (LID), which uses a focused and pulsed laser beam to produce small cracks in materials such as Plexiglas or crystal glass [Wood 2003]. Each crack then serves as a scatterer that glows when it is lit. LID has become popular in recent years and is used to create 2D images and 3D models in ornaments and souvenirs. In our context, LID is very attractive as it imposes no constraints on the nature (regular or random) of the point cloud. It can create a cloud with hundreds of thousands of points in a matter of minutes at a very low cost[6].

We have empirically studied the photometric properties of LID cracks. In Figure 4(b), the blue line shows the measured brightness of a single crack plotted as a function of the viewing angle. The viewing angle varies from 0° to 70° degrees, where 0° is exactly opposite to the direction of illumination. As can be seen, the brightness of the crack falls as one moves away from the illumination direction, but we have found that in a dimly lit environment the brightness is clearly perceived even at 70°. One can increase the brightness of

[6] Each of the glass crystals with LID point clouds we used was fabricated by Lazart Inc. (www.lazart.com) for a retail price between $50 and $200.

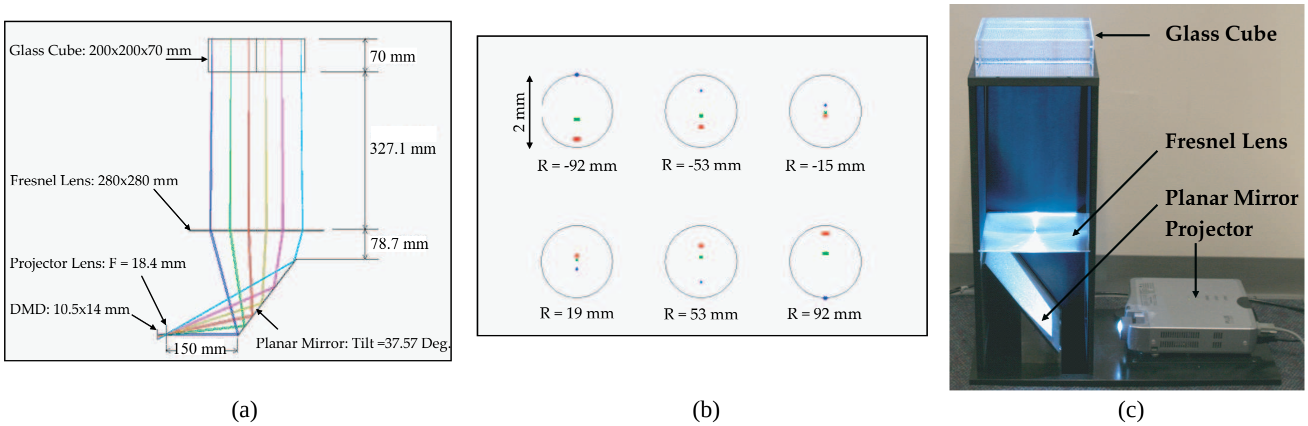


Figure 5: (a) Optical layout of the orthographic light engine. (b) Spot diagrams that quantify the optical resolution of the light engine. (c) The complete display system including the light engine and an LID point cloud.

each point by including multiple cracks within it. In Figure 4(a), the four crack configurations we experimented with are shown, including the single crack. Here, each crack is shown as a sphere although it generally has a complex structure. As expected, the brightness of the point increases with the number of cracks. We have used the third configuration (with 6 cracks) in most of our implementations.

Figure 4(c) shows the spectral characteristics of LID. We created a dense planar grid of cracks and illuminated it with all the colors in the visible spectrum using a projector. The bottom image shows for comparison the same illumination pattern projected onto an optical diffuser. We see that the LID cracks do reasonably well at reproducing the color of the incident light. Figure 4(d) shows a $200 \times 200 \times 70$ glass crystal with a $48 \times 48 \times 25$ point cloud that is XYZ-dithered. In this case, the cloud is uniformly lit with white light.

5 Orthographic Projection System

In theory, our point clouds can be illuminated using any type of projection system that generates a 2D set of light rays. In the case of an off-the-shelf projector, the light rays diverge from a single point requiring XY grids that are farther from the projector to have lower resolution, which is undesirable from the perspective of the viewer. The non-uniform resolution also makes the alignment and calibration of the point cloud with respect to the projector more cumbersome. For these reasons, we have developed an orthographic light engine that produces a parallel set of light rays, as illustrated in Figure 2.

The optical layout of the light engine is shown in Figure 5(a). The light source is a Plus Vision U232 DLP projector with a resolution of 1024x768 pixels. This projector was chosen as it has a wide field of view that enables us to reduce the physical dimensions of the light engine. A planar mirror is used to fold the projection optics to further reduce the size of the system. A Fresnel lens (No. 37 from Fresnel Technologies) was used to convert the projector into an orthographic one. The Fresnel lens converts the diverging rays from the projector into parallel rays that are focused in the middle of the glass crystal that has the LID point cloud. A Fresnel lens was used instead of a conventional lens as a large projection footprint is needed to illuminate the $200 \times 200 \times 70$mm glass crystals we have used for many of our point clouds.

The zoom setting of the projector, the position and tilt of the planar mirror and the focal length of the Fresnel lens were chosen via optical simulation done using the commercially available ZEMAX package. For the simulation, the glass crystal was assumed to have a thickness of 70mm and a refractive index of 1.52. Figure 5(b) shows spot diagrams for the plane of focus, which can be viewed as the point blur functions of the system. Three spots (corresponding to red, green and blue light) are shown for different radial distances from the optical axis in the plane of focus. For small distances from the optical axis ($R = -15$mm and $R = 19$mm) the three spots are very compact and close together. For larger distances, the distances between the spots increases because the Fresnel lens we have used has strong chromatic aberrations as it was not designed for high resolution applications such as ours. Due to these aberrations, as well as defocusing in other depths within the $200 \times 200 \times 70$mm glass cube, the worst-case spot size was determined to be about 1.5mm. This enables us to use a fully indexable point cloud with about about $(200/1.5)^2 = 17,777$ points in our current implementation. It is important to note that higher resolutions, and hence denser point clouds, can be achieved by custom designing a Fresnel lens for our system[7]. Furthermore, some of the applications we show only require partially indexable point clouds (as in the case of extruded objects) and in such cases denser point clouds can be used. Figure 5(c) shows the complete optical system, inclusive of a LID point cloud.

6 Calibration and Rendering

Each LID crack is about 0.18mm in width and about 0.21mm in length. The length is measured along the direction of the laser beam used to create the crack. The positions of the cracks are accurate within 0.05mm. If each point in the cloud has the 6-crack configuration shown in Figure 4(a), it has a width of about 0.5mm and a length of about 0.6mm. If the light engine is perfectly orthographic, it is straightforward to calibrate the display. One can simply use the (X,Y,Z) coordinates of the corner points of the cloud in a coordinate frame attached to the glass crystal and the (i,j) coordinates of pixels in the projector that light up the chosen corner points to determine the projector pixels that light up each and every point in the point cloud. However, due to lens distortions and misalignments in the optical system, such a simple approach is not precise enough.

In our current implementation, we have taken a semi-manual approach to the calibration problem. We calibrate each XY grid (layer) of the cloud separately. The mapping between projector pixels and points in a single grid is modeled as $i = f(X,Y,Z)$ and $j = g(X,Y,Z)$, where f and g are low-order (3 in our case) polynomials. We have developed an interface that allows the user to manually find the projector pixels that light up a small number of points on the given XY grid. These correspondences are used to find the polynomials f and g. Then, all the points on the XY grid are illuminated using this computed mapping. We found that only a few points are not adequately lit by the mapping and the projector pixels corresponding to these points are manually refined using our interface. This process is repeated for all the XY grids in the cloud. This calibration has to be done for each of the three color channels separately since our system

[7] The Fresnel lens we have used costs $70. A Fresnel lens with lower aberrations can be custom-designed and produced in large quantities for a similar price.

has chromatic shifts (see Figure 5(b)). For the most complex clouds we have used, this process takes about 2 hours per color channel and hence a total of 6 hours for the complete calibration. The fully automatic calibration of such a 3D display is a challenging problem. In the future, we plan to develop a method that uses an array of cameras and coded projection. More that one camera is needed as no single viewpoint has 100% visibility[8].

The algorithm we use for rendering objects is simple and efficient. Given a 3D scene, our goal is to determine the brightness (for each color channel) that each point in the cloud needs to be illuminated by. For this, we use an intermediate representation that is a regular 3D grid that has higher resolution than the point cloud but is aligned with it and occupies the same volume. That is, if a point cloud P has n^3 points, we choose a regular grid Q that has m^3 points, where $m > n$. For each point u in the cloud P, we find all its neighboring points $\{v_w | w = 1, 2, ...W\}$ in Q that are less than a small distance d from u. We also find the distances of these points from u, $\{r_w | w = 1, 2, ...W\}$. For each point in the cloud the lists v_w and r_w are computed and stored prior to rendering.

During rendering, each color channel of the input 3D scene is resampled to the regular 3D grid Q to obtain the brightness values I_v. Then, for each color channel, the brightness I_u of each cloud point u is determined as a weighted sum of the brightnesses of its precomputed neighboring points in Q: $I_u = \sum_{w=1}^{W} r_w I_{v_w} / \sum_{w=1}^{W} r_w$. As an example, for $n = 64$, we may use $m = 255$ and $d = 10$ (in the units of the grid Q). Since the lists v_w and r_w are precomputed, the rendering computations are linear in n^3, the number of cloud points. In our applications, we have used clouds with $n < 64$ and hence rendering is easily done at frame-rate for a dynamic 3D scene. The calibration and rendering algorithms described above are implemented on a 2.66GHz Dell workstation.

7 Applications

We now present the different volumetric displays we have developed, where each one is geared towards a specific class of objects.

True 3D Objects

For the display of simple 3D objects, we implemented a fully indexable, XYZ-dithered cloud with $25 \times 25 \times 16 = 10,000$ points in a $200 \times 200 \times 70$mm glass cube. When all the points are projected onto the same XY plane, the distance between adjacent points is 1.9mm in each of the two dimensions. The distance between adjacent XY planes is 4mm and the Z-dithering is in the range ± 1.5mm. Figure 7(a) shows a ball that is half red and half green. Three different views of the display are shown to convey the 3D nature of the content.

This display was implemented only to demonstrate the feasibility of a fully indexable, XYZ-dithered cloud. Its resolution was kept low to make the calibration easy and because of the limited optical resolution of the light engine. With a custom-designed orthographic projector, we believe significantly higher resolutions are achievable.

Extruded Objects with Top Surfaces

Higher resolutions can be achieved for the class of extruded objects with arbitrary top surfaces. These are objects that have translational symmetry in their geometry and texture along the Z direction. In addition, the objects can have "lids" with arbitrary textures. For this, we implemented a point cloud with a fully indexable top layer (the XY grid that is farthest from the projector) with $100 \times 100 = 10,000$ points and a very dense extrusion volume with $180,500$ points. These extrusion points are Z-dithered and lie along projection rays that are different from the rays corresponding to the top layer points. Figure 7(b) shows several extruded objects displayed using this point cloud. Note that the objects with oval and square cross-sections have green and red top lids with holes.

Purely Extruded Objects

In the case of dynamic text used in advertisements or to convey any form of information, the characters can be given a 3D appearance by displaying them as purely extruded objects. For this we have used a dense extrusion volume with $180,500$ Z-dithered points in a $200 \times 200 \times 70$mm glass cube. Figure 7(c) shows different views of the characters "3D" displayed. The characters are blue in color but have white extruded outlines.

Games in Three Dimensions: 3D Pac-Man

Traditionally, video games are played using a 2D display. An inexpensive approach to volumetric display such as ours opens the possibility of designing 3D games. This not only makes the gaming experience more compelling from a visual standpoint but also challenges the player to reason in 3D. As an initial step in this direction, we have implemented an extension of the popular game Pac-Man. The point cloud used to implement the game was etched in a $80 \times 80 \times 80$mm glass cube and is shown in Figure 6(b). The paths and targets used in the design are shown in Figure 6(a), where the four different colors of the path segments correspond to different depths in the cube. The game is really a maze of path segments that are either horizontal or vertical and the targets are small patches at the ends of some of the horizontal path segments. This point cloud can be viewed as a 2.5D one as any given projector pixel may only illuminate a dot on a single horizontal path segment or a full vertical path segment. That is, the paths do not overlap each other in the Z dimension.

Figure 6(c) shows a person playing the game. The player's location is displayed as a green blinking dot and this location is controlled by the player using a joystick. All the path segments are displayed in blue and the targets are displayed in bright pink. The player can travel in either direction along a horizontal path segment but must initiate a jump when he/she arrives at a vertical path segment. When a jump takes place, the player's dot automatically sweeps past a vertical path segment and this motion gives the impression of sliding or climbing. When a player gets close to a target, he/she needs to initiate an explosion of the target, which is displayed as the fizzling out of the bright yellow target color. Audio clips are used to convey the motion of the player along horizontal path segments, the sliding up and down vertical segments and the explosions of targets. We have found that even this simple extension of Pac-Man to 3D truly challenges the player and makes the gaming experience more engaging.

Static Surface with Dynamic Texture: 3D Avatar

It is well-known that in the case of certain objects a 3D visual effect is created by simply projecting a video of the object on a static 3D surface. This is especially the case with human faces where a 3D avatar can be created by simply projecting a video of a speaking person onto a projection surface with the shape of the person's face. Even though misalignments between facial features in the video and projection surface occur they are not easily noticed by an observer. This idea has been around for a long time and is referred to as relief projection [Walt Disney Company 1980][Negroponte 1981]. In these previous systems, the face model was made by creating a physical bust of the face, which is a tedious process. In our case, the face model can be simply etched into a glass cube using LID, enabling us to create 3D avatars for any given face model with ease.

Figure 6(d) shows the 3D avatar we have developed. The face mesh is etched in a $100 \times 100 \times 200$mm glass cube and has $127,223$ points[9]. In this case, we used a conventional (perspective) projector whose

[8] 100% visibility can be achieved by placing an orthographic camera at a viewing angle of 90° degrees. However, in this case, the camera will directly receive the rays projected by the light engine and hence it will be "blinded."

[9] This face model and the projected video were kindly provided to us by Li Zhang at Columbia University.

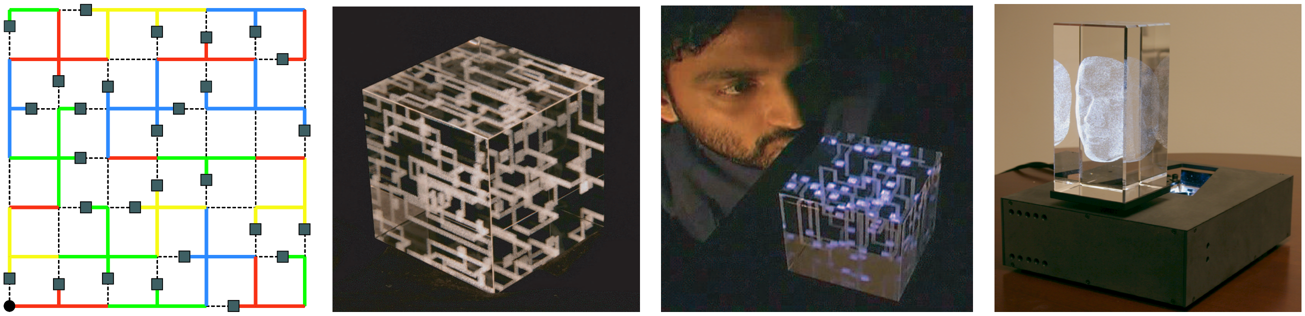

(a) (b) (c) (d)

Figure 6: (a) Design of the 3D Pac-Man where the different colors of the path segments correspond to different depths. (b) The LID point cloud used to implement the 3D Pac-Man. (c) A person playing the 3D Pac-Man. (d) The point cloud of the face and the projection system used to develop the 3D avatar.

throw distance is folded using two planar mirrors to make the system compact. The entire projection system resides within the black box beneath the glass cube (see Figure 6(d)) and the images are projected up at the face model at an angle of 30 degrees. Since the video of the face was captured from a different viewpoint from that of the projection system, a simple user interface was developed to calibrate the system and warp each frame of the video such that it is better aligned with the face model. The calibration is done by simply clicking on points (prominent features such as eye corners, lip corners, etc.) in a single video frame and the projector pixels that illuminate the corresponding points on the face model. These correspondences are used to compute a piecewise linear mapping between the video frames and the face model. Figure 7(d) shows five different views of the 3D avatar. As expected, the misalignments caused by the expression changes are not easily visible and the model truly appears to have a life of its own.

8 Discussion

In this paper, we have presented a class of volumetric displays that are easy and inexpensive to implement and yet are effective in conveying 3D information. We have demonstrated displays with different resolution properties that are geared towards specific classes of objects and contents. The main limitation of the displays is resolution, which is due to the inherent trade-off between resolutions in the three dimensions. However, we believe the resolution of our fully indexable display can be improved substantially. In future work, we plan to custom-design the components of our light engine to achieve significantly greater optical performance. Once this is done, we believe we will be able to implement a fully indexable display with about $250,000$ points using a projector with 1 million pixels.

In addition to increasing the resolution of our indexable displays, we would like to extend our work on 3D games and avatars. We are interested in working with game designers to explore how the third dimension can be used to design new games that engage a player in ways that are not possible with 2D games. With the 3D avatar, we are interested in adding audio and video intelligence to the system so that people can interact with the avatar. A video camera and a microphone will be embedded in the avatar system and used to detect (and maybe even recognize) faces. The captured audio signal will be processed using a commercial speech recognition system. We plan to use these two components to enable the system to respond to questions with speech, expressions and realistic lip movements. Such an intelligent avatar can be used as a guide in a museum or an information source in a home environment.

References

Actuality Systems. Perspecta. *http://www.actuality-systems.com*.

Blundell, B. G., and Schwartz, A. J. 2000. *Volumetric Three Timensional Display Systems*. John Wiley and Sons, Inc.

Blundell, B. G., Schwarz, A. J., and Horrell, D. K. 1994. Cathode Ray Sphere: A Prototype System to Display Volumetric Three-Dimensional Images. *Optical Engineering 33*, 180–186.

Downing, E. A., Hesselink, L., Ralston, J., and MacFarlane, R. M. 1994. A Three-Color, Solid-State Three-Dimensional Display. *Science 273*, 1185–1189.

Kostinski, A. B. 2001. On the extinction of radiation by a homogeneous but spatially correlated random medium. *JOSA 18*, 8, 1929–1933.

LightSpace Technologies. Depthcube. *http://www.lightspacetech.com*.

MacFarlane, D. L. 1994. A Volumetric Three Dimensional Display. *Applied Optics 33*, 31, 7453–7457.

Matusik, W., and Pfister, H. 2004. 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes. *ACM Transactions on Graphics 23*, 3, 814–824.

Negroponte, N. 1981. Media Room. *Proceedings of Symposium of Society for Information Display 22*, 2, 109–113.

Okoshi, T. 1976. *Three-Dimensional Imaging Techniques*. Academic Press, San Diego, CA.

Parker, E. J., and Wallis, P. A. 1948. Three-Dimensional Cathode-Ray Tube Displays. *Journal of IEE 95*, 371–390.

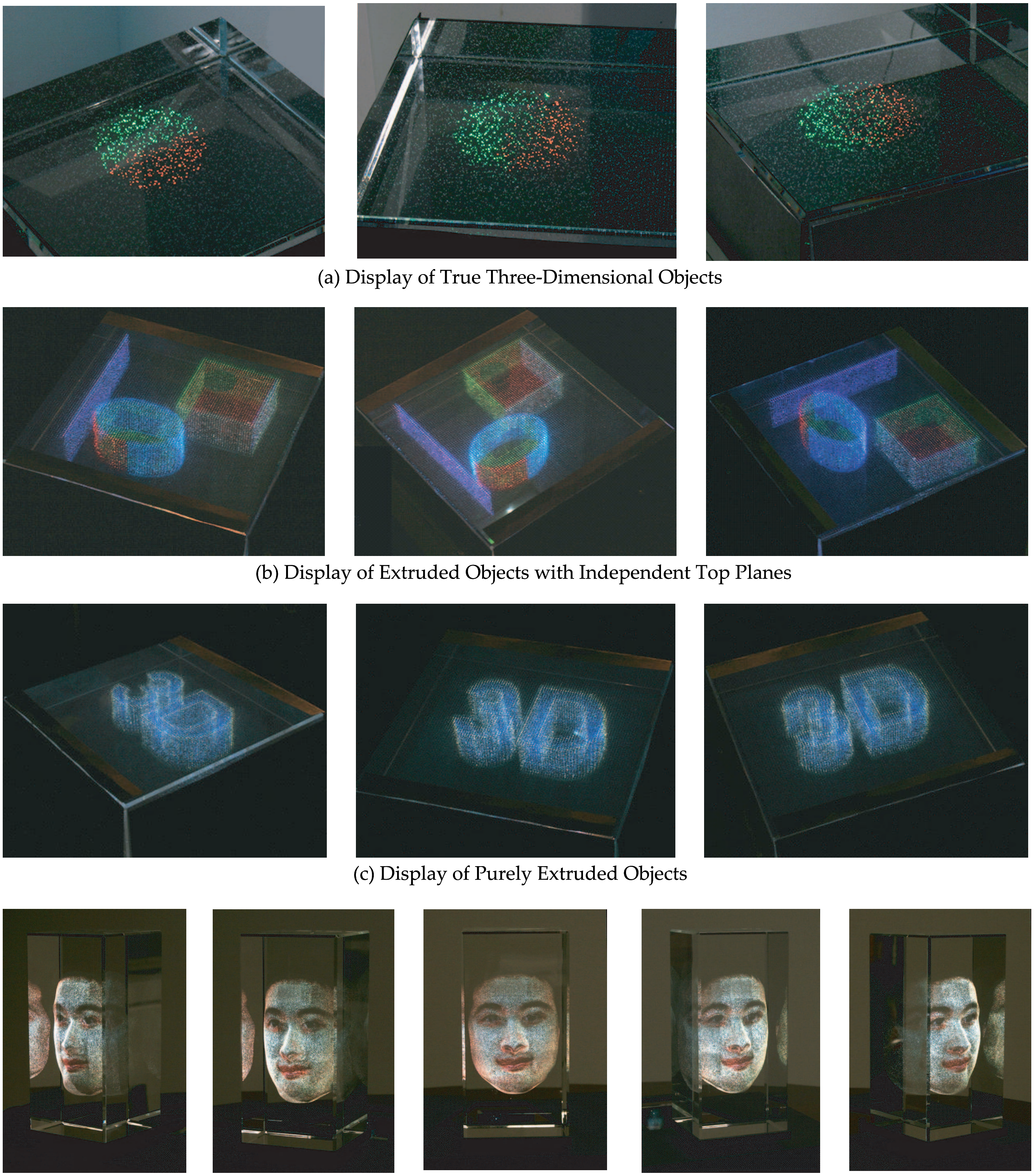
Perlin, K., Paxia, S., and Kollin, J. S. 2000. An Autostereoscopic Display. In *Proceedings of ACM SIGGRAPH*, 319–326.

Perlin, K. 2004. Volumetric Display with Dust as the Participating Medium. *United States Patent Application*, 20040218148 (November).

Schwarz, A., and Blundell, B. 1993. Considerations Regarding Voxel Brightness in Volumetric Displays Utilizing Two-Step Excitation Processes. *Optical Engineering 32*, 11, 2818–2823.

Walt Disney Company. 1980. Talking Heads. *Haunted Mansion, Dinsey World*.

Wood, R. M. 2003. *Laser Induced Damage in Optical Materials*. Institute of Physics.



(a) Display of True Three-Dimensional Objects

(b) Display of Extruded Objects with Independent Top Planes

(c) Display of Purely Extruded Objects

(d) Display of a Face using Relief Projection

Figure 7: (a) A fully indexable display with 10, 000 XYZ-dithered points developed for the rendering of simple 3D objects. (b) A display with 190, 500 points including a dense extrusion volume and an indexable top layer for the rendering of extruded objects with arbitrary top-layer textures. (c) A display with 180, 500 Z-dithered points for the rendering of purely extruded objects. (d) A 3D avatar display with a face model made of 127, 223 points. In each case multiple views of the display are shown to convey the 3D nature of the content.