

LenticleObjects: 3D Printed Objects with Lenticular Lens Surfaces That Can Change their Appearance Depending on the Viewing Angle

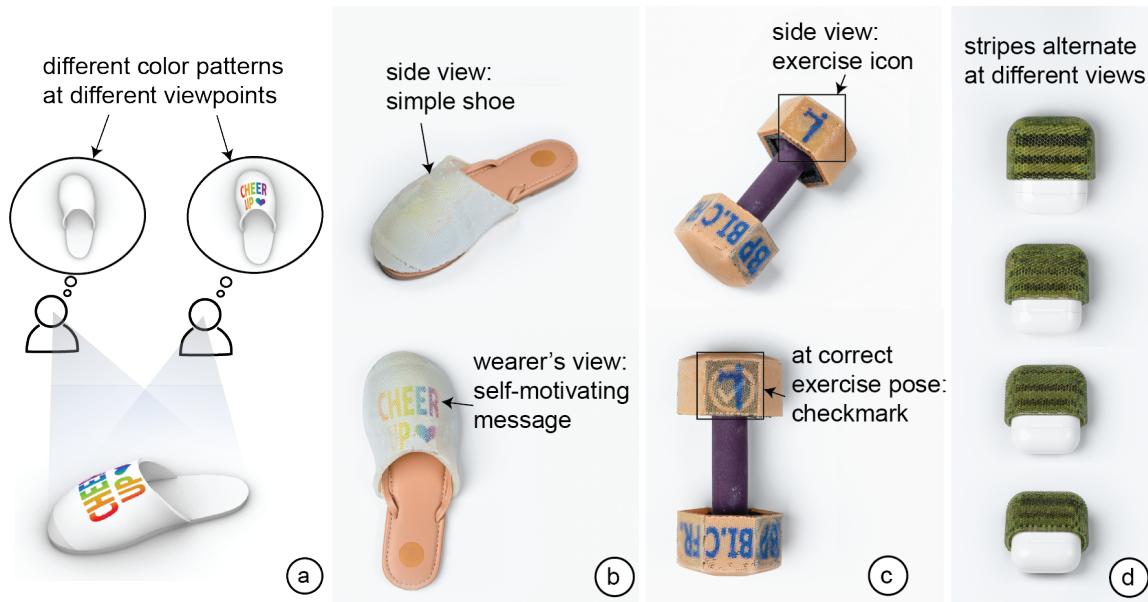


Fig. 1. Our 3D printed objects with lenticular surfaces enable viewers to see different appearances from different viewing angles. To accomplish this, we 3D print the object geometry, lenses, and underlying color patterns in a single pass with a multi-material 3D printer. Our editor supports users in setting up the different viewpoints and assigning the corresponding textures. On export, it automatically generates the files for fabrication.

In this paper, we present a method that makes 3D objects appear differently under different viewing angles. We accomplish this by 3D printing lenticular lenses across the curved surface of an object. By calculating the lens distribution and the corresponding surface color patterns, we can determine which appearance is shown to the user at each angle. We built a 3D editor that takes as input the 3D model and the visual appearances, i.e. images, to show on its surface at different viewing angles. Our editor then calculates the corresponding lens placement and the underlying color patterns. The 3D model, color pattern, and lenses are then 3D printed in one pass on a multi-material 3D printer to create the final 3D object. We evaluate the result by conducting a technical evaluation of the optimal lens geometry and lens size as well as the minimum viewing angle and maximum number of viewpoints that can be supported. We then demonstrate our system in practice with a range of use cases.

CCS Concepts: • Human-centered computing → Human computer interaction (HCI).

Author's address:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

Additional Key Words and Phrases: multi-material 3D printing; optical effects; lenticular lenses; design tools.

ACM Reference Format:

. 2018. LenticleObjects: 3D Printed Objects with Lenticular Lens Surfaces That Can Change their Appearance Depending on the Viewing Angle. 1, 1 (September 2018), 20 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Lenticular printing refers to the method of using lenticular lenses to show different images from different viewing angles [12]. It is commonly applied in arts and advertisement to achieve special purpose effects, such as showing various images on one card in commercial motion cards [7, 30] or creating the illusion of depth in glasses-free 3D displays [4, 10]. Traditionally, lenticular lens effects are created by placing flat lenticular lens sheets onto 2D color patterns, which are made from multiple images. Depending on the viewing angle, light rays enter the lens at different angles and thus reflect off different parts of the underlying color pattern. Therefore, under different viewing angles, a different part of the color pattern is visible to the viewer's eye and as a result the viewer sees only a particular image that is contained in the color pattern.

To date, lenticular displays have been restricted to 2D surfaces and have not yet been explored in the context of 3D objects and their curved surfaces. One reason for this is that up until recently, no fabrication process existed that was able to manufacture the object geometry, lenticular lenses, and high resolution color patterns in one integrated process. Being able to manufacture such a 3D print would allow not only 2D flat sheets, but also 3D objects to appear differently from different viewing angles thereby enabling new applications in product design and HCI.

In this paper, we explore how to use recent advancements in multi-material and high-resolution color 3D printing to fabricate 3D lenticular displays in the form of 3D objects that look different under different viewing angles (Figure 1). The 3D printer we use (Stratasys J750 [21]) can print with optical clear material [24], which we use for the lenses, as well as material that can be colored by mixing CMYK on demand [23], which we use for the color pattern and geometry. To facilitate the creation of objects that look different from different viewing angles, we provide designers with a 3D editor that takes as input the desired viewing angles and corresponding images and then computes the lens placement and color pattern distribution across the object's surface. In our technical evaluation, we show that our conversion and fabrication pipeline can create 3D printed lenticular surfaces with up to 22 different viewing angles.

Our method of printing lenticular lenses across a 3D object's geometry allows for new applications. For instance, objects that can display different appearances under different viewing angles can be used to guide users into correct poses, grab user's attention by showing color highlights at specific viewing angles, and provide private messages only seen by the user.

In summary, we contribute:

- a method to extend lenticular lens displays to 3D object geometries, enabling 3D objects to look different under different viewing angles;
- an interactive editor that enables designers to specify viewing angles and the corresponding visual designs, and that subsequently outputs the fabrication files for 3D printing the object with the lenses and color pattern;
- an algorithm that distributes the lenses across the surface of the 3D model, assigns each lens the corresponding color pixels from the input texture and arranges them under the lens according to the desired viewing angle;
- a technical evaluation of the optimal lens geometry and lens size as well as the minimum viewing angle and maximum number of viewpoints that can be supported;

- a range of scenarios that demonstrate our system in practice, including objects that guide user's into the correct poses, grab user's attention, or provide private messages only seen by the viewer.

The remainder of this paper is structured as follows: We first review related work and then discuss each of the contributions listed above in order.

2 RELATED WORK

Our work is related to the fabrication of optics, in particular lenses, as well as work on lenticular displays.

2.1 Fabricating Optics

Recent developments in fabrication have enabled researchers to create objects with precisely tuned optical effects. For instance, to control the appearance of objects, Weyrich et al. [31] mill small microstructures into the surface of an object to create a homogeneous reflectance. Rouiller et al. [19] extend this concept to 3D printing and expand it to spatially varying reflectance by printing both matt diffuse and highly reflective materials onto a single object surface. Researchers also explored how to control the scattering of light by either stacking pre-existing layers of materials with different scattering properties [9] [5] or by using a 3D printer's capability to vary materials spatially to change scattering behavior across an object's volume [16].

To go beyond optical effects on the surface of objects, researchers also experimented with the fabrication of optical clear material. While early approaches used pre-fabricated optically clear layers [1] or pre-fabricated optically clear sticks [34], more recently researchers investigated the use of 3D printing for creating custom optics. Printed Optics [32], for instance, showed how to 3D print custom optical elements, such as light fibers, using optically clear 3D printing material. Papillon [2] extended this work by optimizing the routing of printed light fibers for curved surfaces. Finally, Computational Light Routing [17] provided a generalizable computational solution to perform the routing of 3D printed fibers on arbitrary geometries. Another way to fabricate with optically clear material is to use CNC milling. Researchers, for instance, showed how to create custom caustics [15] by milling small reflective microfacets onto the surface of a geometry. The microfacets change the direction of the incoming light rays and concentrate in specific areas to match a target brightness distribution previously specified by the designer. Since milling the microfacets can create artifacts when the resolution is high, several optimization approaches have been developed to smooth the resulting light distribution [20]. In our work, we fabricate a specific type of optimal element: lenses, and thus optimize our fabrication process for the specifics of the lens geometry.

2.2 Fabricating Lenses

Several researchers have explored how to fabricate lenses using either 3D printing or milling. In Magic Lens [14], Papas et al. describe how to fabricate a 2D lens array that is capable of revealing multiple images with both 3D printing and milling. They evaluate both fabrication methods and conclude that current manufacturing techniques cannot yet fabricate lenses at a high enough resolution to match simulation results. In particular, they point out that the surface roughness, which results from the fabrication process, poses a particular challenge and that further post-processing is needed to make the lens smooth enough for refraction. After evaluating lens results from both milling and 3D printing, they conclude that the milled results are of lower quality than the results from the 3D printer. Recently, researchers also explored ways to further enhance the quality of 3D printed optics to improve their reflectivity (Chen et al. [3]) and surface roughness (Vaidya et al. [29]). To further improve the optical quality of the 3D printing process,

researchers investigated several post-processing techniques, such as using the resin of a UV 3D printer to smooth our existing surface roughness on a previously fabricated lens surface [26]. However, in all of the above works, the lenses were fabricated onto flat object geometries. In our work, we fabricate lenses on curved geometries, which poses new challenges, such as how well the lenses fabricate at different print orientations, for which we provide data in our technical evaluation.

2.3 Lenticular Displays

Several imaging systems have been developed that enable multiple views for stereoscopic effects or brighter screens that redirect light only into the viewer's direction. Matusik and Pfister [13], for instance, use lenticular sheets in two multi-view screens illuminated by multiple projectors to achieve a stereoscopic effect in a cinema setup. Each of these projectors provides one view which is then reflected or refracted in the direction of the audience by an optical layer composed of lenticular sheets, a diffuser, and a retro-reflector. Takaki et al. [27] extend this idea by creating a multi-view display capable of producing 256 different views. To accomplish this, they replace the projection system with flat-panel displays that consist of lenticular lenses. A final image is composed on a common diffuser by a set of projection lenses placed between the diffuser and the flat panels. While previous work on 3D screens focused on achieving a large field of view at the cost of low angular resolution, Efrat et al. [6] developed a back-projection screen that uses lenticular lenses that create narrow range views to multiple locations at high angular resolution, e.g. each seat in a cinema. Instead of using projectors behind a lenticular lens screen, Piovarci et al. [18] developed a reflective surface consisting of a matrix of micro-mirrors that reflect the light of a front-projector to multiple locations in front of the screen. While all of this work enables redirecting light to multiple locations creating different images at different viewing angles, they only developed systems on flat or 1D curved screens. Our work, in contrast, enables multiple views from various view points on highly curved geometries and without the need of projectors. Another stream of research uses lenticular lenses on printed images to create a stereoscopic effect. Yamazaki et al. [33], for instance, show how to optimize the color printing for lenticular lenses. In addition, content-adaptive lenticular prints [28] improve both spatial and angular resolution by fabricating lens arrays of custom lens size, shape, and arrangement using 3D-printing to best match a given input light field. However, their work focuses on 2D lenticular prints and was also limited to printing with only two different materials, which allowed for only black and white prints.

3 BACKGROUND

Figure 2 shows a simplified illustration of a lenticular lens with two different visual appearances under two viewing angles. To create the two different visual appearances, each lens of the lenticular lens array has two colored areas underneath. Because of the magnifying effect of a lenticular lens (Figure 2a), each lens displays only a small portion of its underlying color pattern. Which portion of the color pattern the viewer sees depends on the viewing angle of the viewer, i.e. when the viewer looks at this lenticular lens arrangement from different angles, they will see only one of the two colors through each lens (Figure 2b).

3.1 Image Quality: Spatial and Angular Resolution

The quality of a lenticular display is determined by two resolutions, i.e., its spatial and its angular resolution.

Spatial resolution: Similar to the concept of spatial resolution of digital displays, the spatial resolution for lenticular displays refers to how sharp or blurry the images are. If the spatial resolution is high, the image looks high-quality and individual pixels are not visible to the human eyes. If the spatial resolution is low, individual pixels do not blend as

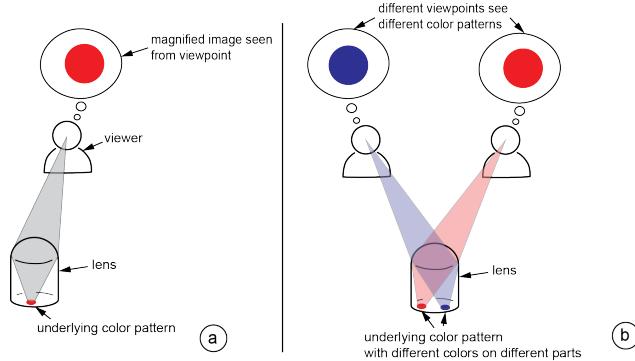


Fig. 2. (a) Magnifying effect of a lenticular lens: each viewpoint corresponds to only a small area underneath the lens. (b) Different viewpoints corresponds to different areas under the lens, thus from each viewpoint the viewers only sees a portion of the color pattern underneath the lens.

well into the overall image. The spatial resolution of a lenticular display can be measured by determining the spatial resolution of the image that the viewer sees from a certain viewing angle. In other words, the spatial resolution of the lenticular display is equal to the spatial resolution of a photo taken of the lenticular display from the angle. Since each color pixel that the viewer sees corresponds to one lens, we can measure the spatial resolution by counting the number of lenses that fit on a particular area on the 3D geometry. As we will show in our technical evaluation, we can support a spatial resolution of up to 12 dots per inch, with each dot being created by a 2mm lens.

Angular resolution: Angular resolution determines how smoothly the pattern changes when the viewer's head moves. If the angular resolution is high, changing the head position or object position results in a smooth transition from one image to the next, forming a video effect. If the angular resolution is low, the images jump from one to the next abruptly, similar to a stop-motion animation with only a few frames. Thus, the more angles a lenticular display supports, the better the angular resolution (Figure 3). Angular resolution of a lenticular display can be determined by measuring the minimum required viewing angle distance between two viewpoints from which the viewer can see different images. We found in our technical evaluation that our system can support up to 22 viewing angles.

3.2 Geometry Constraints: Lens Size and Number of Pixels That Fit Under the Lens

An ideal lenticular display has both high spatial and high angular resolution to create a high-resolution image with smooth viewing angle transitions. However, how high each of the resolution can be is affected by the size of the lenses and the size of the image area underneath the lens.

Size of the lenses: The smaller the lens, the higher the spatial resolution of the resulting image. Since each lens has image color areas underneath it, only one of them is magnified and thus seen by the viewer. Therefore, the size of one lenticular lens is equivalent to the size of one magnified area. Thus, the smaller each lens is, the more lenses will fit in a given area, and the better the spatial resolution will be. We found in our technical evaluation that we can 3D print lenses as small as 2mm with the 3D printer we use, resulting in a resolution of 12 lenses per inch.

Number of image areas under each lens: The number of image areas that fit under each lens determines the angular resolution of the lenticular display. When magnified, each viewing angle sees one of the image areas underneath the lens. Therefore, the more underlying image areas each lens has, the more viewing angles can be supported with different

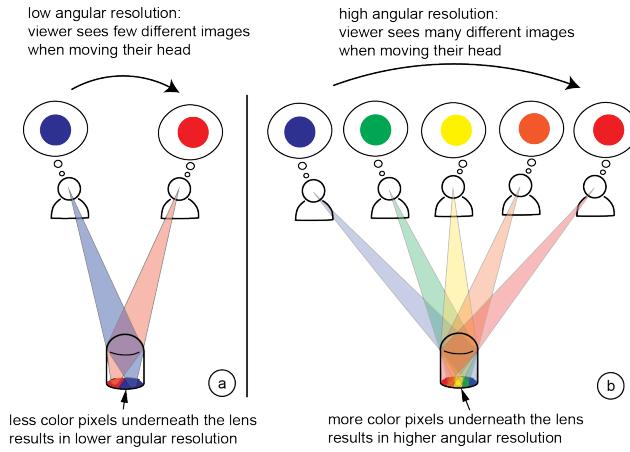


Fig. 3. (a) Lower angular resolution: two viewpoints, vs. (b) higher angular resolution: 5 viewpoints.

images, and the better the angular resolution of the lenticular display will be. However, more image areas also require larger lenses since they have to fit underneath the lens. Thus, spatial and angular resolution cannot be treated separately but depend on each other: the higher the angular resolution, the more image areas need to fit under a lens, and the larger the lens has to be. We found in our technical evaluation that with the 2mm lens size we use each image area has a size of 0.404mm, and thus we can fit up to 22 image areas under the 2mm lens size we use, enabling 22 viewpoints.

4 DESIGN TOOL FOR 3D LENTICULAR DISPLAYS

To support designers in creating 3D lenticular lens displays, we developed a design tool integrated into an existing 3D editor. The designer starts by loading the 3D model of the object and then defines a set of view points and corresponding visual appearances. Our editor then automatically places the lenses on the 3D geometry and assigns the corresponding color patterns to each lens. Afterwards, it provides designers with a set of fabrication files ready for 3D printing. In the next section, we demonstrate the functionality of our editor at the example of a shoe with self-motivating message that can only be seen by the wearer.

4.1 Defining View Points by Placing Virtual Cameras

After loading the 3D model of the object, designers define the view points by placing virtual cameras in the view port at the desired 3D positions (Figure 4a). To do this, designers first click the 'add viewpoint' button, which creates a 3D model of a camera in the view port. Designers can then either manually drag the camera to position it or alternatively enter a 3D coordinate to position the camera more precisely. While the designer is moving the camera, our editor automatically orients the camera to always face towards the object, which creates the specific viewing angle. Designers can verify the viewpoint is correctly positioned by viewing the object through the virtual camera. The designer can add up to 22 viewpoints, which is the number of viewpoints our technical evaluation showed is feasible with the lens size we use.

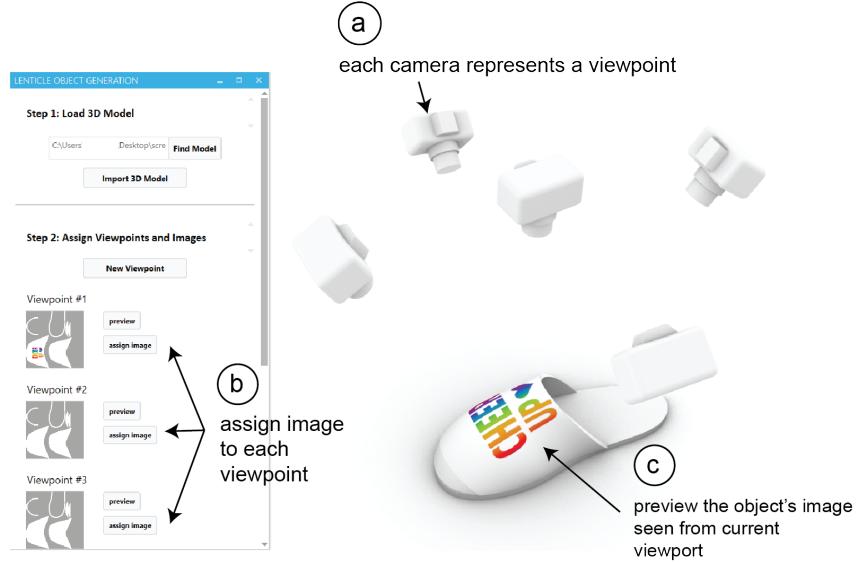


Fig. 4. Defining Viewpoints: After loading the 3D model, designers (a) add viewpoints by placing virtual cameras at the desired positions, our editor always orients the cameras towards the object, thereby creating the specific viewing angle. (b) After positioning a viewpoint, designers (b) load the corresponding visual appearances. (c) Designers can verify if each viewpoint sees the correct appearance by looking through the virtual camera.

4.2 Specifying Appearances for Each View Point by Loading Corresponding 3D Textures

After specifying a viewpoint, designers can define which appearance the object should have from the viewpoint. To do this, designers load one 3D texture for each of the specified viewpoints using the 'assign image' button (Figure 4b). Once a texture is added to a viewpoint, the corresponding camera has the texture associated with it (Figure 4c).

4.3 Previewing the Lenticular Object: Generating Lenses and Color Patterns

After assigning the textures to viewpoints, designers press the "generate lenses and pattern" button to create the lenses and underlying color patterns from the previously defined viewpoints and the assigned 3D color textures (Figure 5).

4.4 Exporting the Fabrication Files

When designers are satisfied with the resulting object, they click the "export" button to generate the fabrication files for 3D printing. The fabrication files consist of a geometry file (.vrml) and an image file for the color pattern (.png). The geometry file includes two parts: the lenses and the object geometry. In addition, the geometry file references the color pattern image. In the next section, we describe the slicing and fabrication process as well as the post-processing techniques we used.

5 FABRICATION

We first describe the 3D printer and 3D printing materials we use and then detail the slicing, fabrication, and post-processing procedure for creating the physical results.

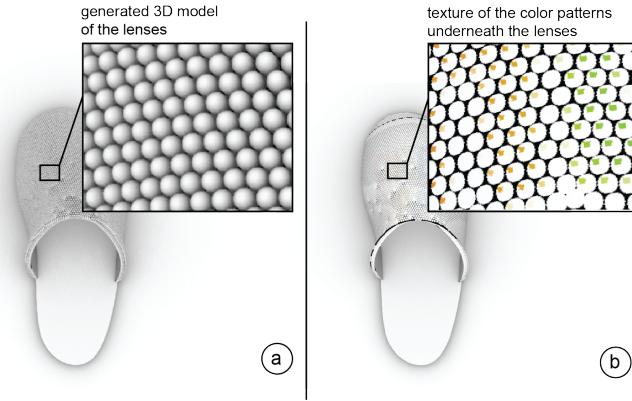


Fig. 5. Previewing the resulting object: (a) Generating the lens geometry and underlying color pattern. (b) Simulating the optical result using ray tracing to provide a more accurate preview of the object before fabrication.

5.1 3D Printer: Printing with Optically Clear Material and High Color Resolution

We fabricate the 3D lenticular objects using a multi-material 3D printer that can print the object geometry, color patterns, and lenses in one pass. We fabricated all the objects in this paper using the Stratasys J750 3D printer [22], which uses polyjet technology. We use the Stratasys polyjet material VeroClear, a type of clear 3D printable acrylic, to fabricate the lenses [24]. We use the Stratasys VeroBlackPlus, VeroPureWhite, and VeroVivid family (VeroCyan-V, VeroMagenta-V, and VeroYellow-V) [23], a collection of rigid opaque 3D printing materials that can be colored on a per-voxel basis using the Stratasys' capability to mix colors from CMY and black/white into the jet-able material during fabrication.

5.2 Slicing the Lenticular 3D Model for Printing

Before printing on the Stratasys J750 3D printer, designers need to slice the geometry fabrication file (.vrml) that was exported from the 3D editor. To do this, designers load the fabrication file into the slicer GrabCAD that can be used with the Stratasys printer (Figure 6a). After loading the .vrml file, which contains two parts: the lenses and the object geometry, designers go to the print settings and associate the corresponding print materials to the parts. Finally, designers click the "slice" button, which generates a file in .print file format, which designers can subsequently load into the 3D printer for fabrication (Figure 6b).

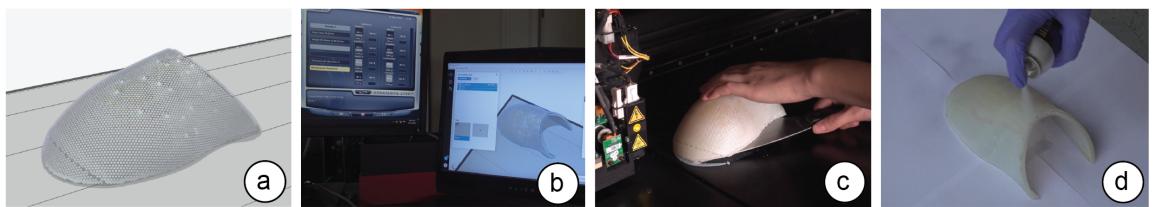


Fig. 6. Fabricating the resulting object: (a) In the slicer, we assign the print settings for the optical clear material to the lenses and the print settings for the colored opaque material to the object geometry with color pattern. (b) Next, we send the fabrication file to the 3D printer. (c) The 3D printed result. (d) After printing, we post-process the model by spraying a clear headlight coating to further improve the optical quality of the lenses.

5.3 3D Printing the Lenticular Object

Fabricating the objects shown in this paper, took 12 hours on average with the objects being of a size of 112cm^3 on average. The object in the walkthrough, which is 155cm^3 printed for 16.5 hours. The printed object is shown in Figure 6c.

5.4 Post-Processing to Improve Lens Quality

After 3D printing finished, designers need to remove all the support material that may surround the lenses depending on the orientation they were printed with. To remove the support material (SUP706 [25]) that the Stratasys J750 3D printer uses, we used a Balco WaterJet machine, which washes off the support materials in less than 15 minutes. The 3D lenticular display already has good quality at this step (Figure 6c), but designers can further improve the quality of the 3D printed lenticular display using simple post-processing techniques. For instance, by spraying clear headlight coating on the surface of the 3D lenticular display, we extend lens clarity and thereby achieve better optical quality (Figure 6d). Post-processing via polishing the lenticular lens display was not possible due to the small size of the lenses.

6 APPLICATION SCENARIOS

Being able to design objects that look different from different viewing angles enables a wide range of novel applications in product design and HCI. In the following section, we highlight several application scenarios that focus on using 3D lenticular objects for pose and attention guidance as well as for displaying personalized messages only visible from the user's unique viewpoint.

6.1 Aligning Objects with Specific Viewing Angles to Inform the Correct Body Pose

Body pose is an important part of activities, such as sports and dancing, and also has a major impact on long-term health when working at a desk in an office environment. However, assessing one's body pose can be hard even with the help of a mirror since one has to self-evaluate to determine when the correct body pose is reached. By placing lenticular lenses on a device that moves with respect to the user, designers can show different surface images to the user when they go through different body poses to signal which one is correct. Figure 7 illustrates this at the example of a dumbbell. Each side of the dumbbell marks a specific exercise pose, and displays a check mark when the user is at the pose. For instance, when the user performs a front raise, the "front raise" face of a correctly angled dumbbell displays the color blue with a check mark. This can prevent the user from exercising at the incorrect angle, which has been shown to lead to injuries [8].

6.2 Moving Objects with Constantly Changing Viewing Angles

When objects are moving with respect to the viewer, they are constantly changing their viewing angle. We can leverage this fact to enable a range of new use cases. For instance, it is often hard for a user to find small objects that have fallen onto the ground. However, by converting the object into a lenticular lens display, we can display a constantly changing color code both as the object falls out of the user's pocket and as the user walks passed the object while looking for it, which helps grab a user's attention. Figure 8a shows this at the example of an ear pod case that flickers with different black-yellow strip colors when the user walks by, grabbing the user's attention.

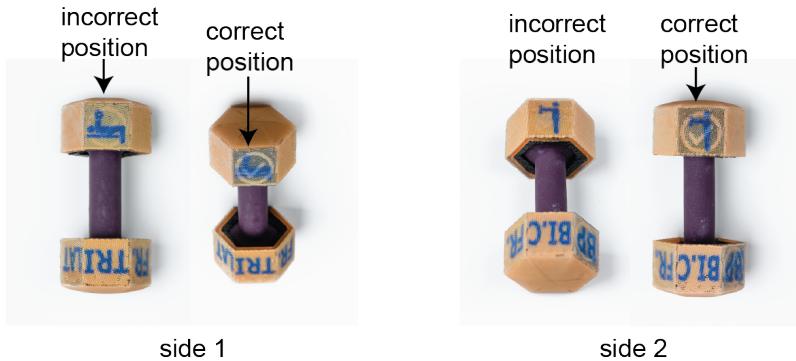


Fig. 7. Pose Guidance: This dumbbell shows specific colors when the user holds it in the correct workout pose (bench press, front raise or lateral raise).

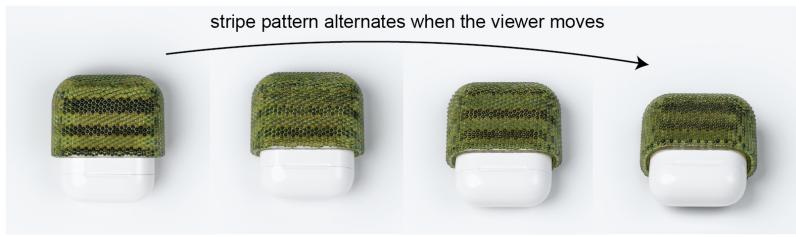


Fig. 8. This lenticular ear pod case flickers between two different yellow-black stripe patterns as the user walks by, grabbing the user's attention.

6.3 Private Messages that are Only Visible from a User's Unique Viewpoint

Lenticular lens displays allow designers to define appearances that are only visible from a specific viewpoint. This can be used to create appearances that are only accessible by a user themselves rather than by others. Figure 9 shows this at the example of showing a private message that is only visible from the user's unique viewpoint. When the user looks down on their pair of shoes, the shoes display a 'Cheer Up' message, however, other people in the surrounding cannot see the message.



Fig. 9. Private messages that are only visible from a user's unique view point. Here, we show a pair of shoes that shows a supportive message to the wearer, which people in the surrounding cannot access.

7 IMPLEMENTATION

Our 3D editor is implemented in the 3D model software Rhino3D as a Grasshopper plugin. Our software pipeline first distributes the lenses across the surface of the 3D model. It then maps the pixels of the different 3D color textures to the lens positions on the 3D model. Next, it computes for each lens how the color pixels of the different 3D textures should be arranged underneath the lens to show the correct appearance for each viewpoint. Finally, it generates the fabrication files and exports them.

7.1 Distributing Lenses Across the Surface of the 3D model

Our goal when distributing the lenses across the 3D object surface is to pack them as closely together as possible while not colliding with each other. Two lenses are not colliding when the distance between the two lens centers is equal or greater than the diameter of the lenses (i.e., in our case $2mm$).

Since lenses are uniform circles, the most efficient way to pack them is hexagonal packing [35]. Hexagonal packing can be achieved by first dividing the surface into equilateral triangles, i.e. triangles with all edges having the same length, and then placing a lens at each corner of the triangle. Since lenses should not collide, we set the edge length of the equilateral triangle to the lens diameter of $2mm$.

To implement this, our system first converts the 3D model into a triangular mesh using the instant meshes open source library [11] (Figure 10a/b). Since instant meshes requires an .obj file as input, our system temporarily exports the 3D model from our editor and uses the exported file for the instant meshes conversion. In addition to the 3D model geometry in .obj format, instant meshes requires several parameters: First, instant meshes requires the target face count, i.e. the number of faces into which the mesh should be converted. Our system determines the number of faces by dividing the surface area of the 3D model by the size of the triangle required to space the lenses without collision. Since our triangle has a side length of $2mm$ to accommodate our lens diameter of $2mm$, the total area of the triangle is $1.732mm^2$. When our system divides the total surface area of the 3D model by this number, it can compute the final target face count. Next, instant meshes also requires as input the rotational and position symmetry. Setting them to a value of 6 results in a hexagonal-directional field suitable for triangular packing.

Once instant meshes finished the conversion, our system imports the converted triangular mesh back into our design tool in the 3D editor Rhino3D. Our system then uses the corners of the triangles, which represent all the vertices of the mesh, as the centers for the lenses to place the lenses across the surface (Figure 10c/d).

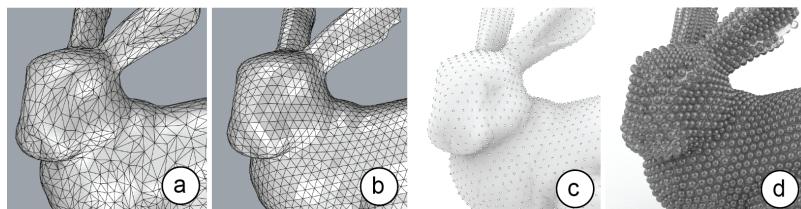


Fig. 10. Placing the lenses: (a) original 3D model, (b) triangular mesh with equilateral triangles, (c) each triangle corner (vertex) is a location for a lens center, (d) lenses placed.

7.2 Taking the Lens Positions on the 3D Model and Mapping the Color Pixels of the 3D Textures to them

Once our system positioned the lenses across the 3D surface, our system next maps the color pixels of the 3D textures to the lens positions on the 3D model (Figure 11a). To know which 3D texture pixels belongs to which lens, our system uses UV mapping. When a 3D object with texture mapping is imported, a list of vertex-to-UV coordinates is already included in the 3D model. Therefore, our system already knows the UV-texture coordinates for each vertex. For points on the surface of the 3D model that are not vertices, our system can further compute their UV-coordinates by finding which face they are on and then interpolate them as the weighted average of the vertices on the face. After this, our system averages the corresponding UV-coordinates of the vertices using the same weight and uses the result as the UV-coordinate of the point.

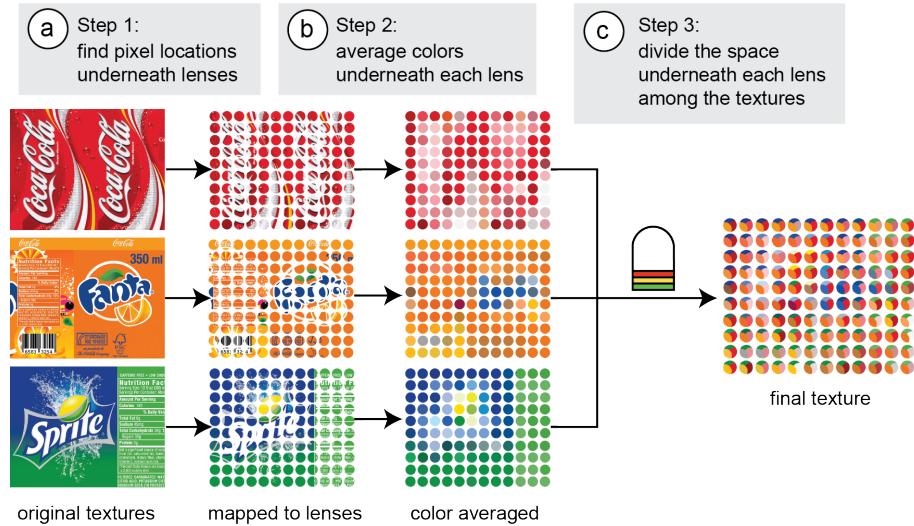


Fig. 11. Mapping 3D color texture pixels onto lenses for each individual 3D texture. (a) Find which pixels are underneath with lens for each texture. (b) For each texture, take the average color underneath each lens as the representative of the overall color underneath the lens. (c) Divide the space underneath each lens using ray-tracing and assign the signature color of the corresponding texture.

After computing all UV coordinates, our system maps each color pixel to a lens. To do this, our system first finds the lens center's pixel location in the color texture by multiplying its UV-coordinate with the length and width of the texture image respectively. Assuming that the input geometry has an even UV-mapping, our system then finds the pixel-radius of each lens on the texture by sampling a point on the circumference of the lens and then finding the distance between the its corresponding pixel and the lens center's corresponding pixel. Therefore, every pixel that is within the pixel-radius distance to the lens center's corresponding pixel are mapped to the lens.

We repeat this step for every input texture but do not have to repeat the UV-mapping since the UV-coordinate of a point on a mesh is the same for all textures. After finishing the processing step, each pixel of each 3D color texture knows which lens it belongs to.

7.3 Averaging the Colors Underneath Each Lens to Create the Representative Color

Next, our system averages the colors of all the pixels assigned to each lens to create the representative color for the lens on a texture as shown in Figure 11b. Our system averages the color for two reasons. First, if our system uses the

Manuscript submitted to ACM

high-resolution textures underneath the lens, it needs to fit all the pixels that fit under the lens into the image area of that lens, which is significantly smaller. As a result, the resolution of the 3D printed color pattern needs to be very high. According to our technical evaluation, the smallest pixel size that our 3D printer supports is 200 microns, which translates to 3 pixels for the image area underneath each of our lenses. Therefore, our current 3D printer setup does not support the resolution for displaying the full-resolution image at a viewpoint.

The second reason is that because of the magnifying effect of the lenses, changing the viewpoint slightly results in looking at two different parts of the underlying pattern. Since users do not keep their head static at all times, a slight movement should not result in a different image. When using high-resolution textures, the user will therefore either see a distorted version of the assigned image or unrelated patterns when they move their head slightly. Because of this, our system averages the color and assigns all pixels underneath a lens the average color of all pixels that fit under the lens.

7.4 Arranging Color Pixels Underneath a Lens According to a Desired Viewpoint

The previous processing steps determined which representative color is underneath which lens. Thus, the output of the previous step was a color to lens mapping for each of the loaded color textures. The question remains how the representative colors from the different textures should be arranged underneath each lens to show the correct color for the corresponding viewing angle (Figure 11c).

Before our system can distribute the color pixels underneath the lens, it first has to determine which area underneath the lens is visible from which viewpoint. To accomplish this, our system casts rays from different viewpoints to the top surface of the lens. When the various rays hit different points on the top surface of the lens, they enter the lens at different angles and thus reflect off different positions at the bottom of the lens (called backplane). Since the color pattern is placed at the bottom of the lens, we know that the positions where the rays reflected off the backplane will be visible from the specific view point from which the ray was cast. We thus only keep the color pixels of the desired texture that should be visible from this viewpoint in these places. We delete all the remaining color pixels from other textures in this area. Because of the small lens size, our system approximates the backplane of the lens to be a flat plane to improve its performance.

7.5 Exporting the Fabrication Files

Finally, once the lenses are generated and the color pattern is correctly distributed across all the lenses, our system exports the fabrication files, i.e. the geometry file (.vrml) with the lenses and object and the color pattern image (.png) that is referenced in the .vrml file.

8 TECHNICAL EVALUATION

In this section, we evaluate the display quality through several key factors concerning the printed lenses (optimal lens geometry, printable lens size), the color pattern (printable resolution, number of pixels that fit under a lens), and the view points (minimum distance between viewpoints, maximum number of viewpoints).

8.1 Optimal Lens Geometry

The basic geometry for a spherical lenticular lens is shown in Figure 12. It has a partial-sphere lens on top and a substrate at the bottom. We define r as the radius of the top spherical lens, d as the diameter of the lens, and $h_{substrate}$ as the height of the substrate underneath the top spherical lens.

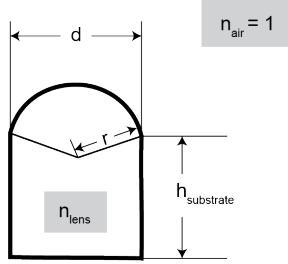


Fig. 12. Geometry of a spherical lenticular lens. To determine the optimal geometry, we aim at minimizing the overall lens height, minimizing the image area of the viewpoints, and maximizing the observation angle of each lens.

To determine the optimal lens geometry, we take into account the refractive index of the lens n_{lens} and the refractive index of the surrounding air n_{air} . For the refractive index of the lens n_{lens} we use 1.52 according to the data sheet from the 3D printable optical clear material we use [24]. We approximate the refractive index of the air n_{air} to be 1. The remaining parameters, i.e. radius r , diameter d , and substrate height $h_{\text{substrate}}$ need to be determined.

Important for our calculation is that independent of the size of the lens, as long as the ratio of $d : r : h_{\text{substrate}}$, is the same, the refractive behavior of the light inside the lens is the same. Therefore, we can choose an arbitrary lens diameter, and then calculate the optimal radius r and substrate height $h_{\text{substrate}}$, which will give a valid result for any lens size that has the same ratio of $d : r : h_{\text{substrate}}$.

We optimize the lens geometry, i.e. the ratio between $d : r : h_{\text{substrate}}$, according to the following goals. First, we want to minimize the overall lens height, which consists of the bottom substrate height and part of the top spherical lens radius, to prevent the lens from protruding too far out from the object geometry. Second, we want to minimize the image area, i.e. the area underneath the lens seen from each viewpoint to be able to fit a large number of viewpoints under a single lens. Third, we want to maximize the overall observation angle of each lens, so that the viewer still sees color pixels even when looking at the lens from a very low angle, i.e. from the side, and not just from the top. Our three overall goals are trade offs with each other. For example, a small image area indicates a large magnifying effect, which typically requires thicker lenses. In order to determine the best trade-off among the three goals, we wrote a script that computes the relation between the three goals (Figure 13a).

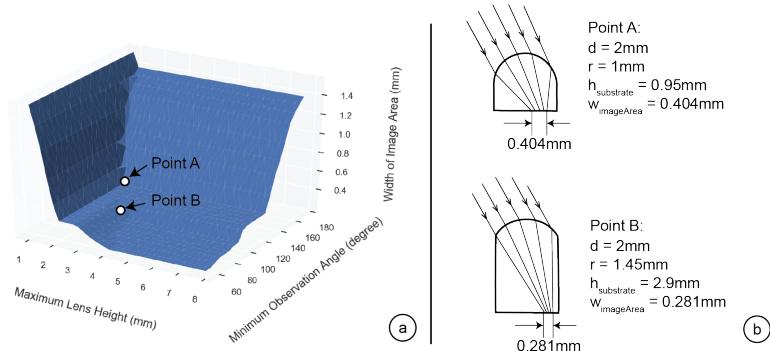


Fig. 13. (a) Selecting points that have small image area, small lens height and large observation angle, (b) corresponding lens geometry of two optimal points.

The script sets various upper bounds for the lens thickness and lower bounds for the observation angles. It then compute the minimum image area viable with those bounds and the ratio between $d, r, h_{substrate}$ that yield such an image area. The result of this computation is plotted in Figure 13a.

Given the relation demonstrated in Figure 13, we can see that point A and point B both have small image area sizes while still maintaining relatively small lens height and large observation angle. At point A, the corresponding lens geometry with diameter $d = 2\text{mm}$ has radius $r = 1\text{mm}$ and $h_{substrate} = 0.95\text{mm}$. Such as lens has an image area size of 0.404mm , lens height of 1.95mm and observation angle of 131° (Figure 13b). Making the image area even smaller, results in thicker lenses and sacrifices more observation angles. In particular, at point B, the lens geometry has $d = 2\text{mm}$, $r = 1.45\text{mm}$ and $h_{substrate} = 2.9\text{mm}$, which yields an image area of 0.281mm , together with the lens height of 3.4mm and observation angle of 97.4° .

For our purposes, we prefer a smaller lens height to not protrude too far from the object geometry, but at the expense of a smaller observation angle. Therefore, the lens geometry at point A is preferable for our use case. Thus, the resulting ratio for the lens geometry is $d : r : h_{substrate} = 2 : 1 : 0.95$.

8.2 Resolution of 3D Printable Color Pattern

While the previous section provides insight into the optimal ratio of $d : r : h_{substrate}$ for the lens geometry, it does not make a recommendation on the best size of the lens. The next question that arises is thus what values we should choose for the parameters. One important factor that we need to consider is that we cannot have lenses that have an image area smaller than the color pixel resolution our 3D printer can produce.

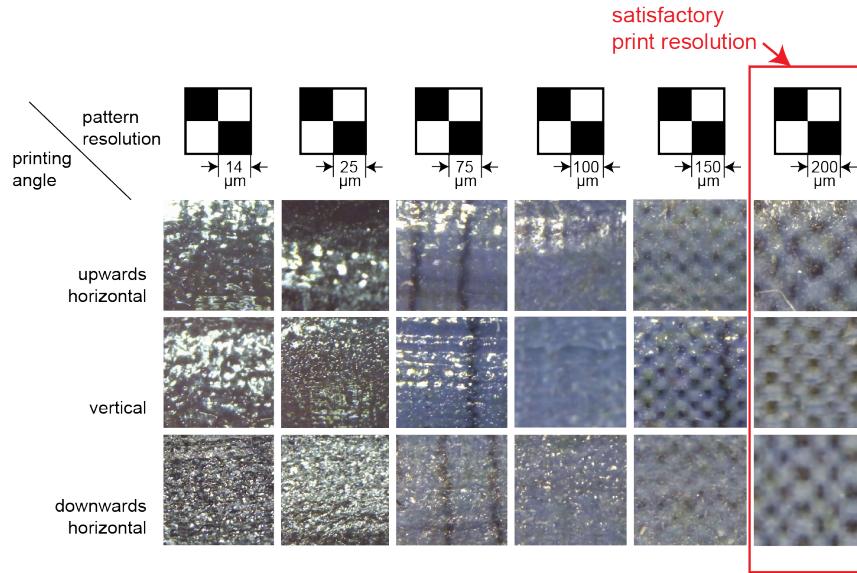


Fig. 14. 3D printed color pattern: checkerboards of different resolutions show that the minimum color resolution is 200 microns per pixel length. Smaller color pixels blur together and are not distinguishable anymore.

To determine the maximum resolution of the color pattern that can be fabricated on our 3D printer, we ran an experiment in which we 3D printed black and white checkerboard patterns of different resolutions and then

analyzed their quality under a microscope (inskam-316 microscope). Our checker board sizes had an edge length of 14, 25, 75, 100, 150, 200 microns (Figure 14), which cover values around the color resolution as indicated in the 3D printer data sheet, i.e. 14 – 200 microns. We printed the color patterns on a cylindrical shape to also take into account that different print orientations may influence the print quality. We found that the quality of the color pattern was lower when the color pattern was at the bottom of the cylindrical geometry. Overall, we found that for the color pattern of 200 microns (0.2mm), the visual quality was most consistent. For smaller values, we found that the printed checkerboards blurred together. Based on this result, we know that we can support lenses with an image area of 0.2mm or larger. Lenses that have such an image area need to be at least 1mm in diameter.

8.3 Smallest 3D Printable Lens Size

After confirming that the 3D printable color resolution can support lenses as small as 1mm, the question remains if such a small lens geometry can actually be printed on our 3D printer.

To determine the minimum 3D printable lens size, we ran the following experiment: Using the lens geometry ratio 2:1:0.95 as determined in section 8.1, we generated six different lens sizes, ranging from a diameter d of 0.5mm to 3mm in 0.5mm increments. We chose these values based on preliminary experiments we had done with our first 3D prints. Since in addition to size, the print orientation may also impact the optical quality of the lenses, we placed the lenses on the circumference of a cylindrical shape (diameter: 2.25cm, thickness: 5mm). We used one cylinder per lens size and placed one lens each at angles ranging from 0° to 345° in 15° increments, 24 lenses total per size).

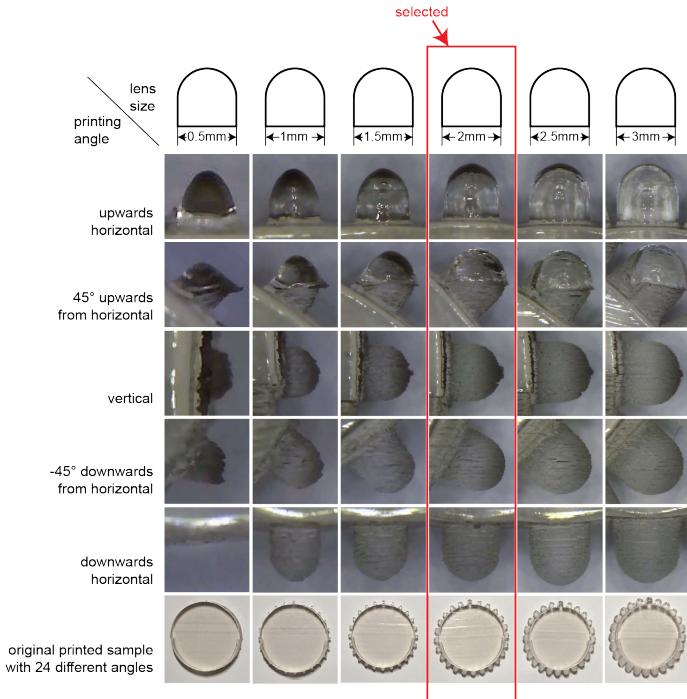


Fig. 15. Printed Lenses of 0.5mm – 3mm diameter. Lenses with smaller than 2mm diameter have distorted geometries that result in inaccurate light behavior.

After printing the six cylindrical geometries, we used a microscope (model: inskam-316) to analyze each lens for the following factors: (1) is the surface of the lens curved correctly or is it deformed? and (2) Is the surface of the lens smooth or rough? Figure 15 shows close-ups of the different lens sizes when printed at the different angles on the cylindrical geometry. We found that the lens with a diameter of 2mm was the smallest acceptable. The lenses that have diameters smaller than 2mm result in distorted geometries. For example, the top surface of the 1.5mm -diameter lens facing upwards is not a half-sphere. All lenses facing downwards have rough surfaces because they are attached to the support materials and can only be printed with matte finishing. However, since our lens only displays one pixel for each lens at a certain viewpoint, surface roughness that does not affect the overall lens geometry is acceptable. We thus conclude that our final lens geometry has a diameter of 2mm , radius of 1mm , and substrate height of 0.95mm .

8.4 Maximum Number of Viewpoints

To calculate the maximum number of viewpoints, we compute the number of image areas that can fit onto the backplane of a lens. Since the width of the image area with our lens geometry is 0.404mm and the diameter of the lens is 2mm , we can calculate that the maximum number of viewpoints in one direction is $5 (\frac{2}{0.404})$. To calculate the maximum number of viewpoints in all three dimensions, we perform a circle packing, packing the image areas from different viewing angles onto the backplane of the lens. Our results show that we can fit up to 22 image areas onto the backplane of a lens. Therefore, we can have at most 22 viewpoints for a lens.

To calculate the smallest angular difference between two viewpoints, we sample pairs of image areas with distances of 0.404mm (first in pair range from 0 to 2mm from the leftmost side, with stepsize of 0.01mm) at different locations on the backplane and trace rays backwards at different incident points (range from 0 to 2mm from the leftmost side, with stepsize of 0.01mm). The difference between the refracted angles from the two points represents the smallest angle a viewpoint has to move in order for its reflected pixel to get 0.404mm away from the original pixel that it hits. The angle ranges from 5.383° to 23.038° for view points at different locations of the lens.

To conclude, with the current 3D printing technology, the smallest lenses we can print are 2mm in diameter, which results in a spatial resolution of 12 pixels (lenses) per inch. Our printed lenticular displays can support up to 22 viewing angles, with a maximum observation angle of 131° , and a minimum angular distance between viewpoints of ca. 23° . In the future, as 3D printing technology improves, the printed results will further increase in quality and smaller lens sizes may become possible, which better integrate with the object geometry.

9 DISCUSSION AND LIMITATIONS

Our current implementation is subject to several limitations, which can be addressed in future work:

One Viewpoint, Multiple Viewing Angles: Our current implementation assumes that the head of the viewer is facing towards the center of the object. Thus, when the designer places a new viewpoint in our editor, the camera representing the viewpoint always faces towards the center of the object. However, a single viewpoint may be used to look at the object at different angles. This is, for instance, the case when the viewer is standing in one position, but rotates the head to look at different parts of the object. Since we mainly print small objects due to the limited build plate size of our 3D printer ($49\text{cm} \times 39\text{ cm} \times 20\text{cm}$ maximum print size), we can assume a fixed head position focused towards the center of the object since when rotating the head, the viewer would no longer look at the object. However, for larger objects where rotating the head is more common to look at different parts, our implementation needs to be extended.

Impact of Lenses on Surface Geometry: Since the lenses are placed on top of the surface of the 3D model, the 3D model geometry is extended by a lens layer of 1.95mm height. While a layer of such lenses may not be an issue for larger

objects, smaller fragile objects will substantially increase in thickness. In addition, placing the lenses on the surface also changes the tactile qualities of the object. We hope that with future developments in 3D printing technology, especially the 3D printing of optical clear materials, we will be able to solve this problem by printing lenses that are so small that they better blend with the object geometry.

Colliding Lenses on Highly Curved Geometries: When geometries are highly curved (i.e. curvature of the concave geometry faces is larger than 0.513mm^{-1}) and we place lenses on them, it can happen that the top of the lenses collide and thus merge together during printing, resulting in an incorrect visual appearance at this point. For future work, we plan to both change the way we place the lenses to prevent collision and to put smaller lenses in highly concave parts of the geometry in order to reduce collision.

Other Lens Types (Cylindrical Lenses): While in this paper, we have focused on spherical lenses, other lens types, such as cylindrical lenses that have the shape of a long and thin partial cylinder, exist. Unlike spherical lenses, which magnify the color pattern in both x and y directions, cylindrical lenses provide only a 1D magnification, i.e. they magnify the underlying color pattern only in the direction of their cross section, but not along the lens. Since cylindrical lenses only have a magnifying effect in one direction, they show an entire line of the underlying color pattern at once. This has two benefits: First, it allows for higher spatial resolution since the underlying color pattern is not broken up into individual pixels, with one pixel under each lens, but only into individual lines. In addition, when a viewer looks at an object covered with cylindrical lenses, the appearance will not change if the viewer moves the head along the direction of the lens (e.g., from left to right) and will only change if the viewer moves the head in perpendicular direction (e.g., up-down). Figure 16 shows an application of this that we created manually: a minimalist product that only shows text instructions when viewed at a particular angle and is otherwise clear. The text of the instruction is high resolution even at this small size due to the cylindrical lenses. In addition, the viewer can hold the product at any horizontal position at the height of their eyes and will still be able to access the text. Only when the product is held at a lower or higher angle, the text disappears.



Fig. 16. Cylindrical Lenses: A case with minimalist design that only shows functional texts when being held at eye level.

10 CONCLUSION

In this paper, we demonstrated how we can leverage recent developments in multi-material and multi-color 3D printing to fabricate 3D lenticular lens displays. We showed how our design tool supports designers in creating 3D lenticular

Manuscript submitted to ACM

lens displays by enabling them to define custom view points and assign desired color textures to them. We discussed our implementation pipeline that automatically computes the lens distribution across the object geometry and assigns the correct color pixels to each lens to achieve the desired appearance from each view point. We also provided information on the slicing parameters as well as the printing and post-processing procedure to fabricate the resulting objects. We then demonstrated various example applications, ranging from pose-guiding tools to private messages. Finally, we provided a technical evaluation of the lens geometry, printable lens size, printable color patterns, and number of viewing angles possible. For future work, we plan to further improve our implementation by taking into account the geometry of the object when generating the lenses and also support multiple viewing angles from one view point.

REFERENCES

- [1] Amit Bermano, Ilya Baran, Marc Alexa, and Wojciech Matusik. 2012. ShadowPix: Multiple Images from Self Shadowing. *Computer Graphics Forum* 31, 2pt3 (2012), 593–602.
- [2] Eric Brockmeyer, Ivan Poupyrev, and Scott Hudson. 2013. PAPILLON: Designing Curved Display Surfaces with Printed Optics. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. 457–462.
- [3] Yu-Fan Chen, Yen-Hung Wang, and Jui che Tsai. 2019. Enhancement of surface reflectivity of fused deposition modeling parts by post-processing. *Optics Communications* 430 (2019), 479–485.
- [4] Neil Dodgson. 2005. Autostereoscopic 3D Displays. *Computer* 38, 8 (Aug. 2005), 31–36.
- [5] Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. 2010. Fabricating Spatially-Varying Subsurface Scattering. *ACM Transactions on Graphics* 29, 4, Article 62 (2010).
- [6] Netalee Efrat, Piotr Didyk, Mike Foshey, Wojciech Matusik, and Anat Levin. 2016. Cinema 3D: Large Scale Automultiscopic Display. *ACM Transactions on Graphics* 35, 4, Article 59 (2016).
- [7] Gregory Ferenstein. 2013. *An Anti-Abuse Ad with a Secret Message Only Children Can See*. <https://techcrunch.com/2013/05/06/an-anti-abuse-ad-with-a-secret-message-only-children-can-see/>
- [8] M Gallagher. 1996. Ten most common causes of training injury. *Muscle & Fitness* (1996).
- [9] Miłos Haśan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. 2010. Physical Reproduction of Materials with Specified Subsurface Scattering. *ACM Transactions on Graphics* 29, 4, Article 61 (2010).
- [10] Tunglu Hung. 2000. *Fan Lihua*. https://library.arstori.org/asset/ASIAARTIG_10313977571
- [11] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant Field-Aligned Meshes. *ACM Transactions on Graphics* 34, 6, Article 189 (2015).
- [12] Gabriel Lippmann. 1908. La Photographie Integrale. *Comptes-Rendus Academie des Science* 146 (1908), 446–451.
- [13] Wojciech Matusik and Hanspeter Pfister. 2004. 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes. 23, 3 (2004), 814–824.
- [14] Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz. 2012. The Magic Lens: Refractive Steganography. *ACM Transactions on Graphics* 31, 6, Article 186 (2012).
- [15] Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich. 2011. Goal-based Caustics. *Computer Graphics Forum* 30, 2 (2011), 503–511.
- [16] Pieter Peers, Karl vom Berge, Wojciech Matusik, Ravi Ramamoorthi, Jason Lawrence, Szymon Rusinkiewicz, and Philip Dutré. 2006. A Compact Factored Representation of Heterogeneous Subsurface Scattering. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. 746–753.
- [17] Thiago Pereira, Szymon Rusinkiewicz, and Wojciech Matusik. 2014. Computational Light Routing: 3D Printed Fiber Optics for Sensing and Display. *ACM Transactions on Graphics* 33, 3 (May 2014).
- [18] Michal Piovarčí, Michael Wessely, Michal Jagielski, Marc Alexa, Wojciech Matusik, and Piotr Didyk. 2017. Directional Screens. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication (SCF '17)*. Article 1.
- [19] Olivier Rouiller, Bernd Bickel, Wojciech Matusik, Marc Alexa, and Jan Kautz. 2013. 3D-Printing Spatially Varying BRDFs. *IEEE Computer Graphics and Applications* 33, 6 (2013), 48–57.
- [20] Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-Contrast Computational Caustic Design. *ACM Transactions on Graphics* 33, 4, Article 74 (2014).
- [21] Stratasys. 2020. *Digital Materials*. <http://www.stratasys.com/materials/polyjet/digital-material>
- [22] Stratasys. 2020. *Stratasys J750 Digital Anatomy 3D Printer*. <https://www.stratasys.com/3d-printers/j750-digital-anatomy>
- [23] Stratasys. 2020. *Stratasys Vero*. <https://www.stratasys.com/materials/search/vero>
- [24] Stratasys. 2020. *Stratasys VeroClear*. <https://www.stratasys.com/materials/search/veroclear>
- [25] Stratasys. 2020. *SUP706 Support Material*. <https://support.stratasys.com/materials/polyjet-materials/polyjet-support-materials>

- [26] Kei Sugiyama and Koji Tsukada. 2019. Proposal of Lens Shaping Method Using UV Printer. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. 125–127.
- [27] Yasuhiro Takaki and Nichiyo Nago. 2010. Multi-projection of lenticular displays to construct a 256-view super multi-view display. *Optics Express* 18, 9 (2010), 8824–8835.
- [28] James Tompkin, Simon Heinzel, Jan Kautz, and Wojciech Matusik. 2013. Content-Adaptive Lenticular Prints. *ACM Transactions on Graphics* 32, 4, Article 133 (2013).
- [29] Nina Vaidya and Olav Solgaard. 2018. 3D Printed Optics with Na-nometer Scale Surface Roughness. *Microsystems and Nanoengineering* 4, 18 (2018).
- [30] Walgreens. 2020. *Lenticular Prints*. <https://photo.walgreens.com/store/lenticular-print-details>
- [31] Tim Weyrich, Pieter Peers, Wojciech Matusik, and Szymon Rusinkiewicz. 2009. Fabricating Microgeometry for Custom Surface Reflectance. *ACM Transactions on Graphics* 28, 3, Article 32 (2009).
- [32] Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. 589–598.
- [33] Hideki Yamazaki and Yasuhiro Takaki. 2012. Printing 3D Light Field with 1D Halftone Screening. In *ACM SIGGRAPH 2012 Posters (SIGGRAPH '12)*. Article 43.
- [34] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. 2012. Pixel Art with Refracted Light by Rearrangeable Sticks. *Computer Graphics Forum* 31, 2pt3 (2012), 575–582.
- [35] Chuanming Zong. 2014. Packing, covering and tiling in two-dimensional spaces. *Expositiones Mathematicae* 32, 4 (2014), 297–364.