Digital integral cloaking: supplementary material

JOSEPH S. CHOI^{1,*} AND JOHN C. HOWELL^{2,1,3,4}

¹The Institute of Optics, University of Rochester, Rochester, New York 14627, USA

²Department of Physics and Astronomy, University of Rochester, Rochester, New York 14627, USA

³Center for Coherence and Quantum Optics, University of Rochester, Rochester, New York 14627, USA

⁴Institute for Quantum Studies, Chapman University, Orange, California 92866, USA

*Corresponding author: joseph.choi@rochester.edu

Published 19 May 2016

This document provides supplementary information to “Digital integral cloaking,” <http://dx.doi.org/10.1364/optica.3.000536>. Detailed specifications and measurements for our setup and demonstration are provided here. Additional principles and metrics are provided to help clarify our work. © 2016 Optical Society of America

<http://dx.doi.org/10.1364/optica.3.000536.s001>

1. METHODS

Here we describe in detail the experimental setup, measurements, and specifications.

A. Object space

The ‘input plane’ is where the sensor lied for the input camera (Sony DSC-RX10), and is parallel to the mechanical scanner/slider on which the input camera was mounted. The ‘output plane’ is where the display screen (Apple iPad mini 4) lied, along with the lenticular lens mounted on the display. The output plane is parallel to the input plane and the two planes are separated by a distance L (L was 13 cm for our demonstration). The background objects consisted of four sets of colored blocks with the dimensions shown in Table S1 (see also Figures 2(a) and 2(b) of the main paper). For horizontal (x) separations between the blocks, the farthest sides of the dark green blocks and the red blocks were 9 cm apart, and the farthest sides of the lime green blocks and the yellow blocks were 18 cm apart. The blocks, from the back side of the dark green blocks to the front side of the yellow blocks, spanned a total depth of about 51 cm. The distance from the front face of the yellow blocks to the camera sensor at the input plane was 39 cm.

B. Input surface

We provide details of the input surface (at $z = z_i$) that collects the rays for propagation using Equation (1).

B.1. Input scan hardware

To capture the input images at the input plane, we used a digital camera (Sony DSC-RX10) for our ‘input camera.’ The Sony DSC-RX10 had a 1 inch 20.2 MP Exmor sensor for low lighting, and a

Table S1. Object dimensions. Distance is from the input plane to the center of the objects, depth is the longitudinal (z) dimension, width is the horizontal (x) dimension, and height is the vertical (y) dimension. (All length units are in cm.)

Objects	Distance	Depth	Width	Height
Back wall (w / black cover)	98			
Dark green blocks	87.6	3.8	3.8	33
Red blocks	70	5.7	2.5	35
Lime green blocks	47	3.8	3.8	33
Yellow blocks	40.6	3.8	3.8	29

f/2.8 Carl Zeiss Vario-Sonnar T* 24-200 mm lens with 8.3x optical zoom. It was mounted on a mechanical slider (from Varavon) that scans horizontally at a fixed speed, which can be adjusted. Visualization 1 shows ~ 58 cm of travel on the scanner, but only $(W_s + 2L \tan(FOV_l/2)) = 18.5$ cm was necessary for our setup. Here, $W_s = 12$ cm was the active screen width of the display (Apple iPad mini 4), and $FOV_l = 29^\circ$ was the field-of-view (FOV) of the output lenslet array (20 cylindrical lens-per-inch array from Micro Lens Technology).

B.2. Input scan video

Visualization 2 shows the actual scan by the input camera (Sony DSC-RX10), which was used for generating our cloaked image (Figures 2(c)-(f)), but modified to reduce the file size: Same 1920x1080 pixels frame size, but sped up by 500%, frame rate reduced to 30 frames/second from 60 frames/second, and bitrate reduced to 5 Mbps from 26 Mbps, compared to the original. With this input scan in Visualization 2, we captured 60 frames per second, while scanning at 6.35 mm/s. The input camera was focused at infinity, zoomed completely out (giving maximum spatial view), and stopped down to f/10. The field-of-view for the camera was measured to be about 60° and 33° for the horizontal and vertical dimensions, respectively.

C. Ray propagation processing

Without the propagation of the input rays to the output rays for $L \neq 0$, our system would not be a cloaking system, but rather a 3D display system. This propagation of rays over a finite distance is then a simple but important characteristic of a cloak. After collecting the images from the input video scan, the video frames were processed using our code (written in C++) on our laptop (Lenovo Thinkpad W540).

C.1. Steps to propagate rays

Each input video frame was assigned an input ray position x_i , that is given by the position of the input camera at the time the frame was captured. Each ‘view’ represents an input ray angle θ_i , which corresponded to a horizontal pixel position of the input video frame. Each valid input ray was then propagated using Equation (1) with $(z_f - z_i) \rightarrow L$, while assuming ambient air ($n = 1$). If the calculated output ray fell on a valid display pixel (in our case, an RGB subpixel), then the image pixel information for this input ray was mapped to the output display pixel. This then completed the ray propagation for the cloak.

C.2. Inverting propagation

For digital cloaking, it may be useful to invert Equation (1):

$$\begin{cases} x_i = x_f - (z_f - z_i) \tan \theta_f, \\ \theta_i = \theta_f \end{cases} \quad (\text{S1})$$

Equations (S1) show us how to find the detector pixel that should be used for the display pixel positioned at $x = x_f, z = z_f$. For example, we can first select detector pixels (or subpixels) that collect rays of the same angle ($\theta_i = \theta_f$), color, and any other desired attributes as the output ray. Out of these candidate pixels, we then select the one that has transverse (x) position closest to x_i , and longitudinal (z) position closest to z_i , where x_i and z_i satisfy Equations (S1). Graphically, we can extend the dotted arrow in Figure 4(b) backward until it intersects the cloaking surface, and then find the nearest matching pixel.

C.3. Vertical scaling

Since our demonstration was a 2D cloak that used cylindrical lenses, the vertical pixel positions of the background objects were fixed. This implies that the vertical size, or magnification, relative to the screen size will be constant, even if the observer changes position only in the 2D x - z plane. However, we can at least match the vertical magnification for a particular observer distance and object depth combination, which is what we did. Specifically, for our demonstrations shown in Figures 2 and 3, our code matched the vertical size of objects that were 65.4 cm from the input plane (in the $-z$ direction), which corresponded roughly to the middle of all four sets of blocks. This object plane would then theoretically match in vertical size when an observer was 185.4 cm from the output plane (in the $+z$ direction).

D. Output surface

We now describe the display (output) plane of our cloak.

D.1. Output display

The output display monitor we used was an Apple iPad mini 4. The display was an LED-backlit LCD monitor, with an active screen size of 20 cm (≈ 7.9 inch) in the diagonal direction. The display pixel size was 2048 x 1536 resolution, at 326 pixels-per-inch (PPI). We assumed the display pixels were square pixels, which was a fairly good assumption. This gives a width of 12.0 cm, and a height of 16.0 cm for the active screen size, in portrait mode. The 3 RGB (Red, Green, Blue) subpixels that composed 1 display pixel, were long in the vertical direction and short in the horizontal direction when the iPad mini was in portrait mode. This is why we used the display screen in portrait mode with the slanted lenslet array at an angle of 3 pixels vertically down, and 1 pixel horizontally across (See Figure S1).

D.2. Output lenslet array

The 20 cylindrical lens-per-inch array from Micro Lens Technology we used was a very thin (0.41 mm) sheet of plano-convex lenses. This was so we could manually adjust the separation distance between the lenslet array and the display screen. The lenslet array was mounted on a transparent glass that was 2.3 mm thick, using the adhesive layer on the flat side of the lenslet array. The distance between the lenslet array and the screen was adjusted so that the LCD screen pixels lied at the back focal plane of the lenslet array. The field-of-view of the output lenslet array (FOV_l) given by the manufacturer was 29° .

D.3. Slanted RGB subpixel interlacing

The cylindrical lenses of the output lenslet array were slanted at an angle of 3 pixels vertically down, 1 pixel horizontally across, at an angle of $\theta = \arctan(1/3) = 18.435^\circ$ as shown in Figure S1. This then uses the 3 RGB subpixels of the display to gain 3 times the horizontal angular resolution (in number of ‘views’), at the sacrifice of vertical resolution [1]. Our particular combination of slanted lenslet array and display monitor generated 51.5 discrete ‘views’ over 29° of viewing angles (field-of-view).

With the slanted interlaced image, each RGB subpixel gives a unique ray angle (‘view’) due to its relative position under the lenslet (Recall Figure 1(a)). Each subpixel to the right increases the view number, and each subpixel directly below decreases the view number, modulo total view number (for the view number convention in Figure S1). To form a complete and accurate color for a given image point, the slanted setup now requires neighboring red, green, and blue subpixels along the slant to form one ‘effective’ pixel. This ‘effective’ pixel is 1 pixel wide and 3

Fig. S1. Slanted RGB subpixel interlacing example. A cylindrical lenslet array is placed in a slanted angle ($\theta = 18.435^\circ$) on a display panel. This example outputs 9 discrete output ray angles (or ‘views’), instead of 3 (if each lenslet laid vertically along the y-axis). Each RGB subpixel generates a unique ray angle (‘view’), with corresponding view number labeled. Now, each set of neighboring red, green, and blue subpixels along the slant, of the same ‘view,’ form one ‘effective’ pixel, for which one example is highlighted. (We assumed square display pixels that contain 3 RGB subpixels as shown.)

pixels in height, hence the vertical resolution is reduced by $1/3$ as stated earlier. The angle $\theta = 18.435^\circ$ was obtained by assuming that the original display pixels were squares (width=height), all identical in size, and that each display pixel is composed of 3 rectangular subpixels that are ordered red (R), green (G), and blue (B) horizontally.

With this slanted setup, the total number of views ($DimViewX_s$), for a given display monitor and lenslet array combination, is given by:

$$DimViewX_s = \sqrt{10} \cdot \frac{PPI}{LPI}. \quad (S2)$$

PPI (pixels-per-inch) is the pixel density of the display monitor, and LPI (lens-per-inch) is the lens density of the cylindrical lenslet array used. Equation (S2) assumes that the display pixels are uniform and square in dimension.

E. Measurements

Here we describe the details of the “measurements” that demonstrated our 2D digital integral cloak.

E.1. Horizontal (x) demonstration

The details for the horizontal demonstration shown in Figures 2(c)-(f) are now given. The “observer” camera (Nikon D3200) was 260 cm (along +z) from the display screen (cloak), and scanned a total distance of 61.4 cm (along -x) in Visualization 3. The center of the cloak and camera aligned (0° , or $x = 0$) at 13 seconds into the 20 second clip shown in Visualization 3. Viewing angles are positive when the observer camera is to the right ($x > 0$) of the cloak/display, when facing the display screen as shown in Figures 2(c)-(f). Both Visualization 3 and Visualization 4 were sped up by 200%, from the original videos.

E.2. Longitudinal (z) demonstration

Figures 3(a)-(d) showed images from varying observer positions ($z = 272$ cm, 235 cm, 203 cm, and 150 cm, for Figures 3(a)-(d), respectively, for the display screen located at $z = 0$). These positions then collect varying ranges of angles (or ‘views’) from

the light rays emitted by the output cloaking display screen. Assuming that the center of the observer/camera aligns with the center of the cloaking screen, we can calculate the range of horizontal (x) angles collected ($FOV_{x_{observer}}$):

$$FOV_{x_{observer}} = 2 \arctan(W_s/2/z_{obs}), \quad (S3)$$

where W_s is the horizontal width of the output display screen of the cloak, and z_{obs} is the longitudinal distance of the observer from the output screen (the output plane, which is set to $z = 0$). The resulting $FOV_{x_{observer}}$ values were then 2.53° , 2.93° , 3.38° , 4.59° , for Figures 3(a)-(d), respectively.

The images shown in Figures 3(a)-(d) were cropped, but maintained the same pixel size as each other, just like the original images. The observer camera for these images used a tripod to maintain the same height (y), while varying the longitudinal distance z . We attempted to maintain the same horizontal position x of the observer camera, but there was a variance of several cm or so. Note that the centers of the camera and the screen must align for Equation (S3) to be correct. However, for the small $FOV_{x_{observer}}$ values as given in Figures 3(a)-(d), the range of angles should closely match to the calculated values given.

2. CHARACTERIZATION METRICS

We present details of some of the calculations made in the main text, for the metrics that were used to characterize our digital integral cloak. Since our cloak was a 2D demonstration, our analysis is limited to the horizontal (x) and longitudinal (z) dimensions.

A. Output surface metrics

A.1. Output spatial resolution

The limiting spatial resolution of our cloaking system was given by the output lenslet array. The spatial resolution then can simply be calculated by inverting the lens per length value of the lenslet array. So for the 20 lens-per-inch (LPI) lenslet array we used, the spatial resolution is given by 1.27 mm = $1/(20$ lens/inch). This is the width of a single lens of the output lenslet array. However, this is true only if the cylindrical lenslet arrays were not slanted at an angle. Since we used RGB subpixel interlacing, we slanted our lenslet array at an angle of 1 pixel across (horizontally) to 3 pixels down (vertically) (See Figure S1). The corrected horizontal spatial resolution (dX'_s in inches) is then given by

$$dX'_s = \frac{1}{LPI} \cdot \frac{\sqrt{10}}{3}. \quad (S4)$$

The limiting spatial resolution of our cloak, calculated from Equation (S4), is then 0.0527 inch = 1.34 mm.

A.2. Output display pixel spacing

The output display monitor (iPad mini 4) had a pixel density of 326 pixels-per-inch (PPI). So the pixel spacing resolution (horizontal or vertical) of the output display is 0.078 mm. Since we use RGB subpixel interlacing, we actually use the 3 RGB subpixels, giving a finer spatial resolution of $0.078/3$ mm = 0.026 mm per RGB subpixel.

This small pixel resolution was used in calculating the propagated ray positions (Equation (1)) more precisely than the spatial resolution of the lenslet array (Equation (S4)), for our cloaking demonstration. However, using the spatial resolution in Equation (S4) also seemed to provide similar images, likely due to the relatively small spacing of both values compared to the observation distances.

A.3. Output angular resolution

The limiting angular resolution of our cloaking system was given by the output surface (display + lenslet array). The angular resolution ($d\theta_o$) is simply

$$d\theta_o = \frac{FOV_l}{DimViewX_s}. \quad (S5)$$

FOV_l is the field-of-view (FOV) of the lenslet array, and $DimViewX_s$ is the total number of views from Equation (S2). For the digital integral cloak we demonstrated, $d\theta_o = 0.56^\circ$.

B. Input surface metrics

B.1. Ideal input system

First, the quality of input rays captured rely on the ability to separate position and angle (momentum) precisely, with no cross-talk. This is done by focusing the input camera to infinity (see Figure 1(a)) and ‘stopping down’ the aperture. The sum effect of these two steps is that a single input ray (with fixed input position and angle) is mapped to a single detector pixel.

B.2. Input spatial resolution

Recall that for the input scan we used (see Visualization 2), we captured 60 frames per second, while scanning at 6.35 mm/s. This high frame rate and slow scan gave a horizontal spatial resolution (dX_c) of 0.106 mm for the spacing between input camera positions:

$$dX_c = \frac{v_{scan-s}}{FPS_c}. \quad (S6)$$

Here, v_{scan-s} is the speed of the scan in units of (length/second), and FPS_c is the frame-per-second frame rate of the input camera.

B.3. Input angular resolution

Focusing to infinity maps the same ray angles to the same detector pixels of the input camera. Hence the input ray angles are then limited by both the aberrations of the camera and the detector pixels. With a highly aberration-corrected Carl Zeiss lens, we assume aberrations were minimal. Then the horizontal input angular resolution ($d\theta_i$) is given by

$$d\theta_i = \frac{FOV_c}{DimX_f}, \quad (S7)$$

where FOV_c is the horizontal field-of-view of the input camera, and $DimX_f$ is the total number of horizontal pixels for the input video frame. $d\theta_i$ corresponds to the uncertainty in the input ray angles. For our case, $d\theta_i \approx 60^\circ / 1920 = 0.031^\circ$.

B.4. Input aperture

Next, we needed to ensure that each transverse object position was mapped to separate detector pixels as well. This can be achieved by ‘stopping down’ the aperture of the input camera. Doing so decreases the aperture size, reducing the transverse (x, y) spatial extent of the rays that are captured by the input camera. Otherwise, blurring of the input images occur, i.e., a reduced depth-of-field results. It is well known in optics and photography that smaller ‘f-numbers’ (=focal length/aperture size) or reduced aperture size, improves the depth-of-field, which is how clear (‘in-focus’) the objects appear. Ideally, the aperture size should be infinitely small, but then the amount of light entering is infinitely small, too. So we balanced this by using an f-number ($f/\#$) = $f/10$.

Fig. S2. (a)-(b) Input scan video frames. Snapshots at (a) 11 second, and (b) 17 second from Visualization 2. Large depth-of-field of the input scan is demonstrated, with objects from 39 cm to 98 cm (back wall) from the camera (input plane).

With the input $f/\#$, we can calculate the diameter of the input aperture stop (CA_i) used:

$$CA_i = \frac{f}{f/\#}. \quad (S8)$$

Here, f is the focal length used. The specifications for the Sony DSC-RX10 input camera stated the focal length range to be $f = 8.8 - 73.3$ mm. Since we had zoomed all the way out, we used $f = 8.8$ mm. This then gives an input aperture size $CA_i = 0.88$ mm. CA_i gives roughly the range of transverse spatial positions of the objects that are captured for each detector pixel of the input camera. This is particularly true when the lens is focused to objects at infinity, since only a single collimated ray bundle will enter the lens and be focused onto a single detector pixel (ignoring diffraction and aberrations). So CA_i in this way is proportional to the blurring of objects in the input images.

The number to compare to CA_i would be the aperture set by the output surface, which is given by the spacing of a single lens in the output lenslet array (1.34 mm), or the output spatial resolution (Equation (S4)). Again, the output lenslet array can be seen to be the limiting element for spatial resolution and blurring of objects in the cloak. It is worth noting that CA_i is different than the positional uncertainty of the input camera; the latter is the uncertainty for each input ray position, and is given by the input spatial resolution (Equation (S6)) of the camera scan we determined earlier (0.106 mm).

B.5. Input depth of field

Here we discuss the depth-of-field of our input system briefly. Figure S2 shows snapshots of the input scan used for our cloaking system (see Visualization 2). We can see that all the blocks, even the groove lines in the white wall, and the patterns on the back wall are in good focus. Again, the input camera was not the limiting factor for our cloaking system, as we could achieve several meters of depth-of-field easily, but the display (output) surface limited both the spatial and angular resolutions, and hence also what object depths that could be shown clearly.

C. Other system metrics

C.1. Sensitivity to misalignment

Misalignment in hardware setup and imprecise propagation calculations can reduce the cloak quality. These can physically be manifested in skewed angles, shifts in object positions, or mismatching speeds of object movements as seen on the cloak, when compared to the actual background without the cloak. Our system seemed to be most sensitive at the output surface. This is mainly due to the high pixel density (326 pixels-per-inch) of the output display. So the horizontal spatial sensitivity (dx_{mis}) of an unslanted lenslet array to be misaligned on the display is such that a 0.078 mm change in position will shift the ‘view’ seen by

an observer. For slanted lenslet arrays, which is what we used, dx_{mis} is 3 times more sensitive (0.026 mm).

Ideally, the angle of the lenslet array should align exactly along the display pixel line. Even 1 pixel change across the total height of the display screen will begin to cause deviations from the ideal image. So the angular sensitivity ($d\theta'_{mis}$) of the alignment of the lenslet array with respect to the display screen pixels is

$$d\theta'_{mis} = 2 \arctan \left(\frac{(1/2)(3/\sqrt{10})(1/3/PPI)}{Hs_{in}} \right), \quad (S9)$$

where PPI is the pixels-per-inch linear pixel density, and Hs_{in} is the active display screen height in inches. Equation (S9) is for slanted lenslet array mounting on the display. For unslanted lenslet mounting, the $(3/\sqrt{10})$ factor in arctan should be removed. For our demonstrated digital integral cloak with slanted lenslets, $d\theta'_{mis} = (8.8 \times 10^{-3})^\circ$. We can see that increased resolution of displays used require increasingly careful mounting of the lenslet array on the display screen, in both relative positioning and angular alignment.

C.2. Other factors

Other causes of reduced image quality for the cloak include non-uniform bending of the lenslet array plane toward or away from the display screen, and aberrations of the lenslet arrays themselves. Since easily accessible lenslet arrays are spherical or cylindrical in curvature, even spherical aberrations are not corrected. These aberrations effectively increase the depth and transverse size of the focal point of the lenslet arrays. The result is that the display pixels will not be in focus and multiple display pixels will mix, causing blurring of the output image. Even simple lens optimizations of the lenslet array surfaces should improve the cloaking quality.

3. ADDITIONAL DISCRETIZED CLOAKING DESIGNS

We suggest a few other methods for discretized cloaking, than the digital integral cloak we presented in the main paper.

A. Integral cloak with surfaces that are both input and output

In the main text, Figure 1(b) showed a cloak that works with objects on one side and the observer(s) on the other side, i.e., with one surface working as input only, and the other as output only. However, to build an omnidirectional cloak, it needs to work symmetrically. Hence, each plate and lens combination needs two functions- to capture light rays and to display cloaking light rays. This can be done by placing both detector pixels and display pixels next to, or nearby each other, for example. This would then make Figure 1(b) to be front and back symmetric, where an observer can be in front of either side.

B. Passive optical fiber cloak

Optical fibers or other optics, with good anti-reflection coatings, can be used to collect and transmit analog measurement values. Some advantages of this include cloaking in a passive (rather than active) manner, and that its spectrum can be continuously broadband. A disadvantage is that the cloakable volume will be reduced by the fibers now occupying some of this space. The amount of fiber can be reduced by using micro-arrays of objective lenses to couple the input light into the fibers, but at the price of cost and decreased resolution.

Fig. S3. Lensless ‘integral cloaking’ surface. Cross-section view shown. Detector and/or display pixels can be placed on curved ‘superpixels.’ The top layer, which has holes for rays to pass, can be made to absorb light, or also detect incident light to calculate and compensate for irradiance losses.

C. Lensless digital integral cloak

Additionally, a discretized cloak without lenses is possible by using curved surfaces for detection and emission of rays. This is similar to one of the original methods proposed by Lippman for integral photography [S2]. An example surface of such a cloak is shown in Figure S3. This method might be easier to mass-produce and align than integral cloaking with lenses, and its field-of-view will not be limited by lens curvatures (although limited in other ways).

D. Static digital integral cloak

A static cloak for an observer with fixed z -position can also be obtained relatively simply by recording with a camera *from* the planned observer locations. For large observation distances, a finite number of camera images can be used (one image for each ‘view’) for reasonable results. The ray processing becomes simplified, with each ‘view’/angle using a single camera location corresponding to that given view.

REFERENCES

- S1. J. Geng, “Three-dimensional display technologies,” *Advances in Optics and Photonics* **5**, 456–535 (2013).
- S2. “Integral photography,” *Scientific American* **105**, 164 (1911).