Real-time Image Generation for Compressive Light Field Displays

G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar
MIT Media Lab

MIT Media Lab, 75 Amherst Street, Cambridge MA 02139, USA

E-mail: gordonw@media.mit.edu

Abstract. With the invention of integral imaging and parallax barriers in the beginning of the 20th century, glasses-free 3D displays have become feasible. Only today—more than a century later—glasses-free 3D displays are finally emerging in the consumer market. The technologies being employed in current-generation devices, however, are fundamentally the same as what was invented 100 years ago. With rapid advances in optical fabrication, digital processing power, and computational perception, a new generation of display technology is emerging: compressive displays exploring the co-design of optical elements and computational processing while taking particular characteristics of the human visual system into account. In this paper, we discuss real-time implementation strategies for emerging compressive light field displays. We consider displays composed of multiple stacked layers of light-attenuating or polarization-rotating layers, such as LCDs. The involved image generation requires iterative tomographic image synthesis. We demonstrate that, for the case of light field display, computed tomographic light field synthesis maps well to operations included in the standard graphics pipeline, facilitating efficient GPU-based implementations with real-time framerates.

1. Introduction

Within recent years, glasses-free 3D displays have started to enter the consumer market; display form factors now range from hand-held devices and tablets to large-scale home theaters. However, the underlying technology has not fundamentally changed in a century. Parallax barriers [1] and integral imaging [2] are at the core of most modern displays. Parallax barriers add a light-blocking barrier mask, consisting of arrays of vertical slits or pinholes, at a slight offset in front of a conventional 2D display. An observer looking at the screen only sees a subset of the rear pixels, the rest is optically blocked. Hence, an interleaved light field is displayed on the rear device so that the viewer always perceives the correct samples of a virtual 3D scene. Integral imaging achieves a similar effect by mounting an array of lenslets in front of a conventional 2D screen. While successful in displaying glasses-free 3D imagery, both parallax barriers and integral imaging significantly reduce the image resolution; parallax barriers are also very dark. Compressive light field displays employing stacks of two [3] or more [4] high-speed LCDs, stacked transparencies [5], or stacks of polarization-rotating liquid crystal panels [6] have been proposed to overcome the limitations of conventional light field display technology by providing brighter and higher-resolution imagery within wider fields of view and larger depths of field. For the purpose of this paper, we consider a light field to be a collection of two-dimensional images, each depicting a 3D scene from slightly different perspectives [7].

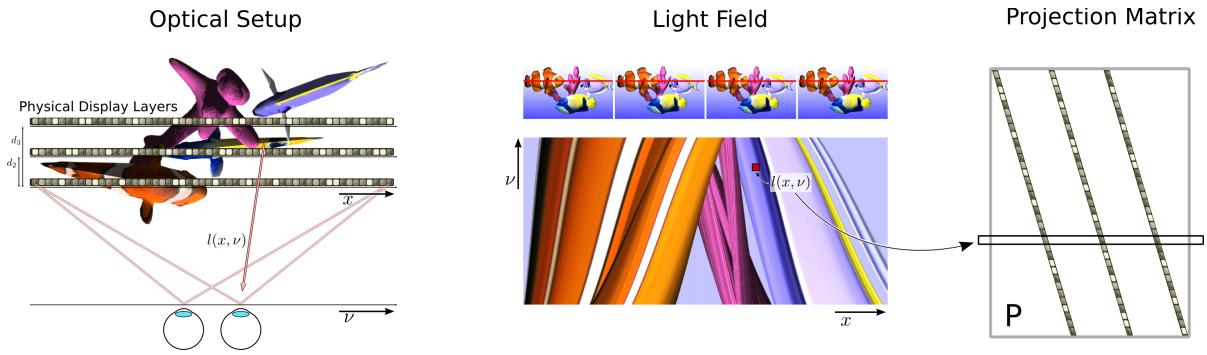

Figure 1. Compressive light field display with a three-layer display. (Left) A virtual 3D scene seen from above with physical LCD layers and viewer indicated. (Center) The corresponding light field is visualized with one spatial and one angular dimension. (Right) Tomographic light field synthesis requires the projection matrix that encodes the physical mapping from display pixels to emitted light field rays.

Compressive light field displays are enabled by the co-design of optical elements and computational processing algorithms. Rather than pursuing a "direct optical" solution (e.g., adding one more pixel to support the emission of one additional light ray), compressive displays aim to create flexible optical systems that have the capability to synthesize a compressed target light field. In effect, each pixel emits a superposition of light rays; through compression and tailored optical designs, fewer display pixels are necessary to emit a given light field, than would be demanded with a "direct optical" solution. The benefits of compressive displays, however, come at the cost of significantly increased computation. The patterns necessary to synthesize a compressed light field require inverse mathematical problems, such as computed tomography, to be solved. In this paper, we investigate efficient implementations of these algorithms and show that, for the specific case of light field synthesis, the underlying abstract mathematical routines directly map to operations supported in hardware on modern graphics processing units (GPUs). The connections between light field, virtual 3D scene, optical display, and mathematical constructs for a three-layer compressive light field display are illustrated in Figure 1.

Related work on 3D displays includes recent advances in parallax barriers, that now support viewer tracking [8] and high-speed panels allowing for time-shifted mask patterns [9]. Integral imaging today is capable of supporting 2D/3D switchable lenses [10] as well as the entire imaging pipeline including image acquisition, compression, and display [11]. Other types of glasses-free 3D displays include volumetric displays [12, 13, 14], stacks of switchable diffusers [15], directional backlighting [16], and multi-focal displays [17]. For a comprehensive review of stereoscopic and automultiscopic display technology, the interested reader is referred to [18].

2. Tomographic Light Field Synthesis

Louis Lumière, a pioneer of cinematographic image capture and display, was the first to propose multilayer display architectures in 1920 [19]. Lumière captured and exposed multiple glass film plates with a camera focused on different depths. A stack of these plates creates a three-dimensional effect, but the underlying problem actually requires an inverse problem to be solved. We show in this section how computed tomographic light field synthesis provides a powerful tool for analyzing multilayer displays and efficiently synthesizing light fields with them.

As shown in Figure 2, multilayer light field displays (e.g., stacks of transparencies or LCDs) are analyzed as general spatial light modulators that act in concert to recreate a light field by attenuating rays emitted by a uniform backlight. Since arbitrary oblique views may be

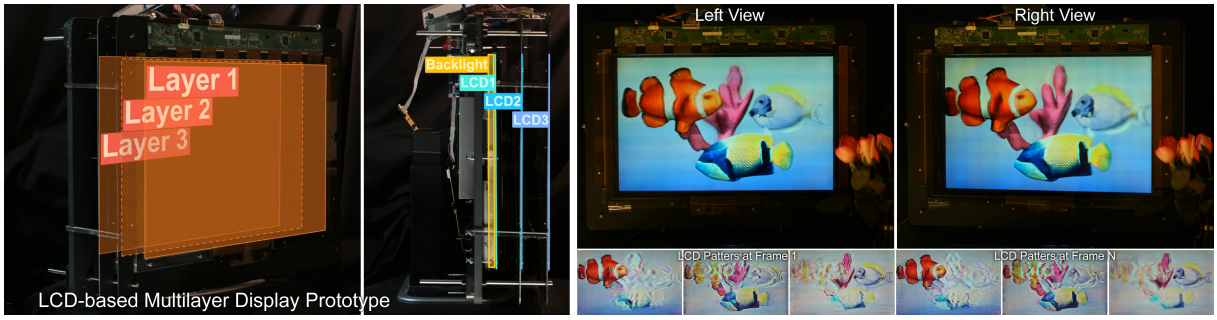

Figure 2. Wide field of view glasses-free 3D display using compressive light field displays. (Left) A prototype configured as a three-layer display; three see-through LCD panels are mounted in front of a uniform backlight. The panels are mounted on custom aluminum frames with driver electronics mounted on the frames. (Right) Photographs showing the display prototypes from two different perspectives. Multilayer light field displays support smooth motion parallax and binocular disparity at a high resolution for a large depth of field over a wide range of viewpoints.

inconsistent with any single attenuator, iterative tomographic reconstruction minimizes the difference between the emitted and target light fields, subject to physical constraints on attenuation. For 3D display, spatial resolution, depth of field, and brightness are increased, compared to conventional parallax barriers.

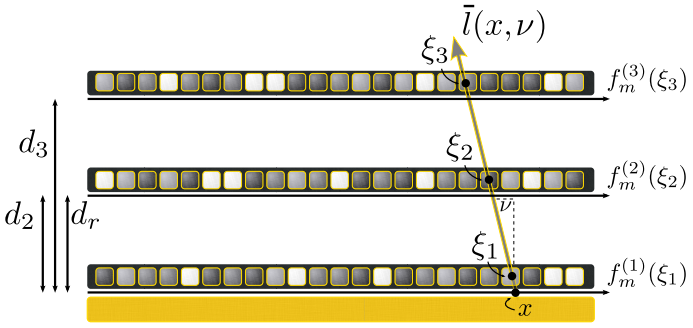

Figure 3. Ray diagram illustrating a multilayer light field display with a single, emitted light ray $\bar{l}\,(x,\nu)$ indicated.

The following analysis adopts an absolute two-plane parameterization of the 4D light field (i.e., the parameterization of rays of light propagating in free space). As shown in Figure 3, an emitted ray is parameterized by its intersection with two planes. Thus, the ray (x,y,ν,υ) intersects the "rear" plane at the point (x,y) and the front plane at the point (ν,υ), with both coordinate systems having an origin in the top-left corner and the relative coordinates $(a=\nu-x, b=\upsilon-y)$. In the following, we analyze two scenarios in "flatland" 2D with straightforward extensions to full 4D light fields and 2D layer patterns: attenuation tomography and tomographic image synthesis for layers of polarization-rotating layers.

2.1. Tomographic Light Field Synthesis with Attenuating Layers

An N-layer stack of optical attenuation functions (i.e., masks) is defined using the absolute light field parameterization, such that the mask on layer n is given by the transmittance function. Following Wetzstein et al. [5], the emitted 2D light field is given by the following expression:

$$l\left(x,\nu\right)=\prod_{n=1}^{N}f^{(n)}\left(x+\left(d_n/d_r\right)\nu\right), \tag{1}$$

where d_n is the distance of layer n from the x-axis. Taking the logarithm gives an additive model:

$$\bar{l}\left(x,\nu\right)=\sum_{n=1}^{N}\ln f^{(n)}\left(x+\left(d_n/d_r\right)\nu\right)=-\sum_{n=1}^{N}\alpha^{(n)}\left(x+\left(d_n/d_r\right)\nu\right), \tag{2}$$

where $\alpha^{(n)}(\xi)=-\ln f^{(n)}(\xi)$ is the absorbance function for layer n. We consider a discrete light field matrix $\bar{l}$ (containing the logarithm of the target intensities for each ray) and discrete attenuation vectors $\alpha^{(n)}$ for each layer n. With this notation, we formulate light field display using a multilayer attenuation-based display as the following constrained linear least-squares optimization problem:

$$\arg\min_{\alpha}\left\|\bar{l}-P\alpha\right\|^2, \quad \text{for } \alpha\geq 0. \tag{3}$$

Here, P is the projection matrix such that the total attenuation along ray is given by the summation $\sum_{j=1}^{J}P_{ij}\alpha_j$. The following section discusses a very related problem: finding an optimal set of polarization-rotation values in a stack of LCDs.

2.2. Tomographic Polarization Field Synthesis

While stacks of light-attenuating LCDs are successful in light field synthesis, they also reduce the light transmission of the display significantly. We explore optically-efficient display implementations using stacks of liquid crystal panels stripped off their polarizers [6]. The resulting *Polarization Field* display contains a stacked set of liquid crystal panels with a single pair of crossed linear polarizers. Each layer is modeled as a spatially-controllable polarization rotator, rather than a spatial light modulator.

Consider a pair of crossed linear polarizers enclosing a single liquid crystal cell; under the polarization rotator model, the transmitted intensity I is given by Malus' law:

$$I=I_0\sin^2\left(\theta\right), \tag{4}$$

where I_0 is the intensity after passing through the first polarizer and θ is the angle of polarization after passing through the liquid crystal cell. The emitted 2D light field $l(x,\nu)$ is given by:

$$l\left(x,\nu\right)=\sin^2\left(\overline{\theta}\left(x,\nu\right)\right)=\sin^2\left(\sum_{n=1}^{N}\phi^{(n)}\left(x+\left(d_n/d_r\right)\nu\right)\right), \tag{5}$$

where $\phi^{(n)}(\xi)$ denotes the polarization state rotation induced at point ξ along layer n. Under this model, ray (x,ν) intersects the N layers, accumulating incremental rotations at each intersection, such that emitted polarization field approximates the target polarization field, given by

$$\theta\left(x,\nu\right)=\pm\sin^{-1}\left(\sqrt{l(u,a)}\right)\bmod\pi, \tag{6}$$

Under these assumptions, the principal value of the arcsine ranges over $[0,\pi/2]$. Note, with full generality, the target polarization field is multi-valued and periodic.

Similar to stacks of attenuating layers, light field display using multilayer polarization rotators can be formulated as a constrained linear least-square optimization problem. We consider a discrete parameterization for which the emitted polarization field is represented as a column vector $\overline{\theta}$ with I elements, each of which corresponds to the angle of polarization for a specific light field ray. Similarly, the polarization state rotations are represented as a column vector ϕ with J elements, each corresponding to a display pixel in a given layer. This yields the following linear model:

$$\overline{\theta}_i = \sum_{j=1}^{J} P_{ij}\phi_j, \tag{7}$$

where $\overline{\theta}_i$ and ϕ_j denote ray i and pixel j, respectively. Using this notation, an optimal set of polarization state rotations is found by solving the following optimization problem:

$$\arg\min_{\phi} \left\|\theta - P\phi\right\|^2, \quad \text{for } \phi_{min} \leq \phi \leq \phi_{max}. \tag{8}$$

As discussed in the following subsections, we use the simultaneous algebraic reconstruction technique (SART) to solve both Equations 3 and 8.

3. Simultaneous Algebraic Reconstruction Technique (SART)

We observe that the tomographic image synthesis problem, i.e. Equations 3 and 8, can be solved by adapting the simultaneous algebraic reconstruction technique (SART). As proposed by Andersen and Kak [20] and further described by Kak and Slaney [21], SART provides an iterative solution wherein the estimate $\phi^{(q)}$ at iteration q is given by

$$\phi^{(q)} = \phi^{(q-1)} + v \circ (P^{\top}(w \circ (\theta - P\phi^{(q-1)}))), \tag{9}$$

where $\circ$ denotes the Hadamard product for element-wise multiplication and elements of the w and v vectors are given by

$$w_m = \frac{1}{\sum_{n=1}^{N} P_{mn}} \quad \text{and} \quad v_n = \frac{1}{\sum_{m=1}^{M} P_{mn}}. \tag{10}$$

After each iteration, additional constraints on $\phi^{(q)}$ are enforced by clamping the result to the feasible rotation range. Note that ϕ can simply be replaced by α to solve the attenuation-based computed tomography problem. Building upon the Kaczmarz method for solving linear systems of equations [22], SART is shown to rapidly converge to a solution approaching the fidelity of that produced by alternative iterative methods, including trust region and conjugate gradient descent techniques [21]. The pseudo-code for the above equation is given in Table 1. In the next section, we discuss an efficient, GPU-based implementation of the SART algorithm that allows for real-time framerates in tomographic light field displays.

4. Efficient Implementation of SART

The SART algorithm introduced in the last section is well-suited for parallel processing on programmable GPUs [23]. Our code is programmed in C++, OpenGL, and Cg. Light fields are rendered in real-time using OpenGL, followed by several iterations of the GPU-based SART implementation.

Tomographic light field synthesis is not restricted to virtual, rendered scenes. Captured multiview footage, either acquired with a camera array, a depth camera such as the MS Kinect, or a single, moved camera can be displayed in the same manner. In fact, recently popularized light field cameras, such as the Lytro, capture exactly the multiview data that is required as

Algorithm SART - Generic

variables P, b, x, w, v, lb, ub

x = zeros();
$w = 1$ / $(P1)$;
$v = 1$ / $(P^T 1)$;

for all iterations k
    $x = x + v \circ (P^T(w \circ (b - Px)))$;
    x = clamp(x, $[lb\ ub]$);
end

Table 1. SART as a generic update routine. An initial guess of the solution x, here initialized as zeros, is iteratively updated until the algorithm is converged or the maximum number of iterations is reached.

the input to our display. In the remainder of this paper, we assume that a target light field is available; in our implementation, we render it interactively using OpenGL. Each view of the light field depicts the same 3D scene from a slightly different perspective. While many possibilities of accelerating the rendering part of the implementation exist, for instance by exploiting inter-view or inter-frame correlations, this paper focuses on the tomographic part of the image generation, which equally applies to captured light fields.

As outlined in Table 1, SART requires an initial guess of the solution, which can either be all zeros or random values, and iteratively updates that. Only very few different operations are actually required: a forward matrix-vector multiplication Ax, a backprojection using the transpose matrix $A^T x$, element-wise multiplications, sums and differences as well as a clamping operation of the updated solution. Any element-wise operations such as sums, differences, Hadamard products, and the clamping operation can easily be implemented as per-pixel operations in any shader language. The critical operations for an efficient implementation are the forward and backward projections that will be discussed in the following.

4.1. The Projection Matrix

For the special case of tomographic light field synthesis with multilayer displays, the projection matrix P encodes the mapping between light field rays and layer pixels. This concept and the connections between light field rays, physical display layers, target light field, and mathematical projection matrix are illustrated in Figure 1. Each ray $l(x, \nu)$ of the target light field corresponds to one row in P, whereas the columns of P correspond to the pixels of the layers. Except for the boundary cases, each light field ray intersects each layer of the stack exactly once; hence, each matrix row has three non-zero elements. While P is extremely large, it is also very sparse. Note that SART does not require the matrix to be formed at any point as long as matrix-vector multiplications can be directly computed.

4.2. Forward Projection as Perspective Rendering

The matrix-vector product between any vector x and P basically simulates the light field that is observed when the layer pixel values encoded in x are displayed on the multilayer display. Instead of precomputing a large, sparse matrix, this operation can efficiently be implemented on the GPU by rendering each individual layer with the corresponding pattern assigned as a texture from all the perspectives of the light field. Instead of rendering each view for each layer, we initialize one offline buffer for each view. During runtime, each view buffer is activated in sequence, the corresponding projection matrix set, the accumulation buffer activated, and all layers are rendered with the depth test disabled. The accumulation buffer allows for the

contribution of all layers, as seen from a single perspective, to be added up in hardware. All buffers use a precision of 16 bits to allow for non-integer values that can be larger than 255.

These operations are outlined in Table 2. Our implementation, tested on both Windows and Linux, is programmed in C++ and uses OpenGL and Cg shaders. The algorithm assumes a light field with N distinct views and K layers positioned at user-defined depths along the optical axis. Intermediate quantities, including the target light field views and temporary variables storing weights and layer patterns, are internally rendered into 16-bit offline framebuffers (i.e., Framebuffer Objects, FBOs) before the optimized patterns are displayed on the individual layers. Only three separate Cg fragment programs are required, each performing the action implied by their names.

4.3. Backprojection as Projective Texture Mapping

Another critical operation is the multiplication of the transpose projection matrix P^T with an arbitrary vector x. In this case, the vector x does not contain layer pixel values but intensity values of the light field. A backprojection smears the intensity value of each light ray back into the layer pixels and accumulates them there. In an unoptimized, sequential implementation, the intensity value of each light ray is simply added to all layer pixels that the ray intersects. An efficient implementation can perform many of these operations in parallel. Projective texture mapping, as supported in hardware by OpenGL, is one way of implementing these operations. Again, 16 bit offline buffers are initialized, this time for each layer. During runtime, each of these is activated in sequence. For each of the layer buffers, the accumulation buffer is enabled and each view of the light field encoded in x is projected using the corresponding perspective matrix set as the current texture matrix. This basically simulates a number of projectors, each placed at one viewpoint of the target light field displaying the corresponding image in x. The offline buffer for the currently active layer simply sums all the projectors' contributions via the enabled accumulation buffer. The same procedure is repeated for each layer in the display stack as outlined in Table 2.

4.4. Temporal Coherence in Dynamic Scenes

SART, as an iterative optimization algorithm, requires multiple iterations to converge to the optimal solution. For the application of tomographic light field synthesis, we experimentally determined five to ten iterations to be sufficient when the solution is initialized with zeros. For dynamic scenes, where the 3D scene or parts of it change from one frame to another, estimates for the previous frame seed the optimization for the current frame. For static scenes, this effectively implements an increasing number of SART iterations over time, while providing a suitable initialization for successive frames in a dynamic environment. In this "seeding" mode, fast moving objects observe a slight motion blur, while the image quality immediately improves when the motion stops. Figure 4 plots the performance of our GPU implementation for three different display configurations.

5. Discussion

Within the past few years, stereoscopic and automultiscopic 3D displays have started to enter the consumer market. Adoption has been slow, however, mainly due to the high bandwidth requirements, low resolution, and dim images provided by conventional multiview, glasses-free 3D displays. Compressive light field displays offer higher-resolution images to be displayed at a significantly reduced bandwidth. As demonstrated in this paper, the increase in required computational processing can be handled by efficient implementations of tomographic image synthesis on programmable graphics hardware. The underlying mathematical operations map directly to standard operations supported on hardware, such as perspective rendering, projective texture mapping, and accumulation buffering.

Algorithm SART - Implementing Light Field Synthesis using OpenGL

```
variables FBO_LF[N], FBO_LF_TMP[N], FBO_LAYERS[K], FBO_W[N], FBO_V[K], DEPTH[K]

function displayLoop()
    if not initialized
        initialize all FBO_W, FBO_V
    end
    drawLightField();
    runSART();
    drawReconstructedLayers();
end

function drawLightField()
    for all light field views i
        activate FBO_LF[i]
        set perspective i
        drawScene(); // render desired 3D scene (e.g., a teapot)
    end
end

function runSART()
    for all iterations k
        // 1. compute Ax^(k)
        for all light field views i
            activate FBO_LF_TMP[i]
            enable BLEND_MODE
            set perspective i
            for all layers l
                draw 2D plane at DEPTH[l] textured with FBO_LAYERS[l]
            end
        end
        // 2. given Ax^(k), compute W (b - Ax^(k))
        for all light field views i
            activate FBO_LF_TMP[i]
            activate CG_SHADER_MULTIPLY_SUBTRACT(FBO_W[i],FBO_LF[i],FBO_LF_TMP[i])
            draw orthographic 2D plane with normalized texcoords
        end
        // 3. given W (b - Ax^(k)), compute V A^T (W (b - Ax^(k)))
        for all layers l
            activate FBO_LAYER[l]
            activate BLEND_MODE
            for all light field views i
                set projective texture to perspective i
                activate CG_SHADER_MULTIPLY(FBO_V[l],FBO_LAYER[l])
                draw orthographic 2D plane with automatic texcoord generation
            end
        end
        // 4. enforce constraints by clamping values outside feasible range
        for all layers l
            activate FBO_LAYER[l]
            activate CG_SHADER_CLAMP(FBO_LAYER[l])
            draw orthographic 2D plane textured with FBO_LAYER[l]
        end
    end
end

function drawReconstructedLayers()
    for all layers l
        set viewport for display l
        draw orthographic 2D plane textured with FBO_LAYERS[l]
    end
end
```

Table 2. Tomographic light field synthesis using SART. In this particular application, the abstract mathematical operations required for a generic tomographic reconstruction map to standard operations such as perspective rendering and projective texture mapping.

We strongly believe that the key for next-generation, glasses-free 3D display technology is the co-design of display optics and computational processing. In all of our prototypes, compressive light field synthesis is optically facilitated by stacks of conventional displays while computationally exploited with efficient implementations of advanced algorithms. While the physical limits of display optics can only be slowly overcome, available processing power has been growing rapidly; in the near future, we expect the computational side of compressive displays to increasingly contribute to advances in display capabilities. Compressive light field displays already achieve a higher resolution, brighter images, wider viewing zones, and larger depth ranges than conventional technology. We are inspired by the promise that future generations of

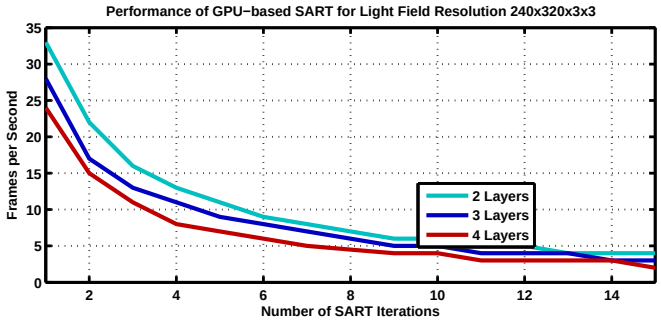

Figure 4. Performance of the GPU-based SART implementation as a function of varying numbers of SART iterations (per frame) and varying numbers of polarization-rotating or light-attenuating layers. The light field resolution is 320×240 spatial samples and 3×3 angular samples; layers have a similar spatial resolution.

compressive displays approach the realism of the physical world with technology that is available today.

Acknowledgments

We thank the Camera Culture Group for their support and also the MIT Media Lab sponsors as well as NVIDIA Research. Gordon Wetzstein was supported by the DARPA SCENICC program and by an NSERC Postdoctoral Fellowship (PDF). Douglas Lanman was supported by NSF Grant IIS-1116452 and by the DARPA MOSAIC program. Ramesh Raskar was supported by an Alfred P. Sloan Research Fellowship and a DARPA Young Faculty Award.

References

[1] Ives F E 1903 Parallax stereogram and process of making same U.S. Patent 725,567
[2] Lippmann G 1908 *Journal of Physics* **7** 821–825
[3] Lanman D, Hirsch M, Kim Y and Raskar R 2010 *ACM Trans. Graph. (SIGGRAPH Asia)* **29**(6) 163:1–163:10
[4] Wetzstein G, Lanman D, Hirsch M and Raskar R 2012 *ACM Trans. Graph. (SIGGRAPH)* **31**(4) 1–11
[5] Wetzstein G, Lanman D, Heidrich W and Raskar R 2011 *ACM Trans. Graph. (SIGGRAPH)* **30**(4) 1–11
[6] Lanman D, Wetzstein G, Hirsch M, Heidrich W and Raskar R 2011 *ACM Trans. Graph. (SIGGRAPH Asia)* **30**(6) 1–9
[7] Levoy M and Hanrahan P 1996 *ACM SGGRAPH* pp 31–42
[8] Perlin K, Paxia S and Kollin J S 2000 *ACM SIGGRAPH* pp 319–326
[9] Kim Y, Kim J, Kang J M, Jung J H, Choi H and Lee B 2007 *Optics Express* **15** 18253–18267
[10] Jacobs A, Mather J, Winlow R, Montgomery D, Jones G, Willis M, Tillin M, Hill L, Khazova M, Stevenson H and Bourhill G 2003 *Sharp Technical Journal* 15–18
[11] Matusik W and Pfister H 2004 *ACM Trans. Graph. (SIGGRAPH)* **23**(3) 814–824
[12] Jones A, McDowall I, Yamada H, Bolas M and Debevec P 2007 *ACM Trans. Graph. (SIGGRAPH)* **26**(3) 40:1–40:10
[13] Cossairt O S, Napoli J, Hill S L, Dorval R K and Favalora G E 2007 *Applied Optics* **46** 1244–1250
[14] Favalora G E 2005 *IEEE Computer* **38** 37–44
[15] Sullivan A 2003 *SID Digest* vol 32 pp 207–211
[16] Travis A, Large T, Emerton N and Bathiche S 2009 *Optics Express* **17** 19714–19719
[17] Akeley K, Watt S J, Girshick A R and Banks M S 2004 *ACM Trans. Graph. (SIGGRAPH)* **23**(3) 804–813
[18] Urey H, Chellappan K V, Erden E and Surman P 2011 *Proc. IEEE* **99** 540–555
[19] Lumière L 1920 *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 891–896
[20] Andersen A and Kak A 1984 *Ultrasonic Imaging* **6** 81–94
[21] Kak A C and Slaney M 2001 *Principles of Computerized Tomographic Imaging* (Society for Industrial Mathematics)
[22] Kaczmarz S 1937 *Bull. Acad. Pol. Sci. Lett. A* **35** 335–357
[23] Keck B, Hofmann H, Scherl H, Kowarschik M and Hornegger J 2009 *SPIE* vol 7258