Supplementary Material:

A Compressive Light Field Projection System

Matthew Hirsch*   Gordon Wetzstein*   Ramesh Raskar

MIT Media Lab

A   Additional Details on Optical Configuration
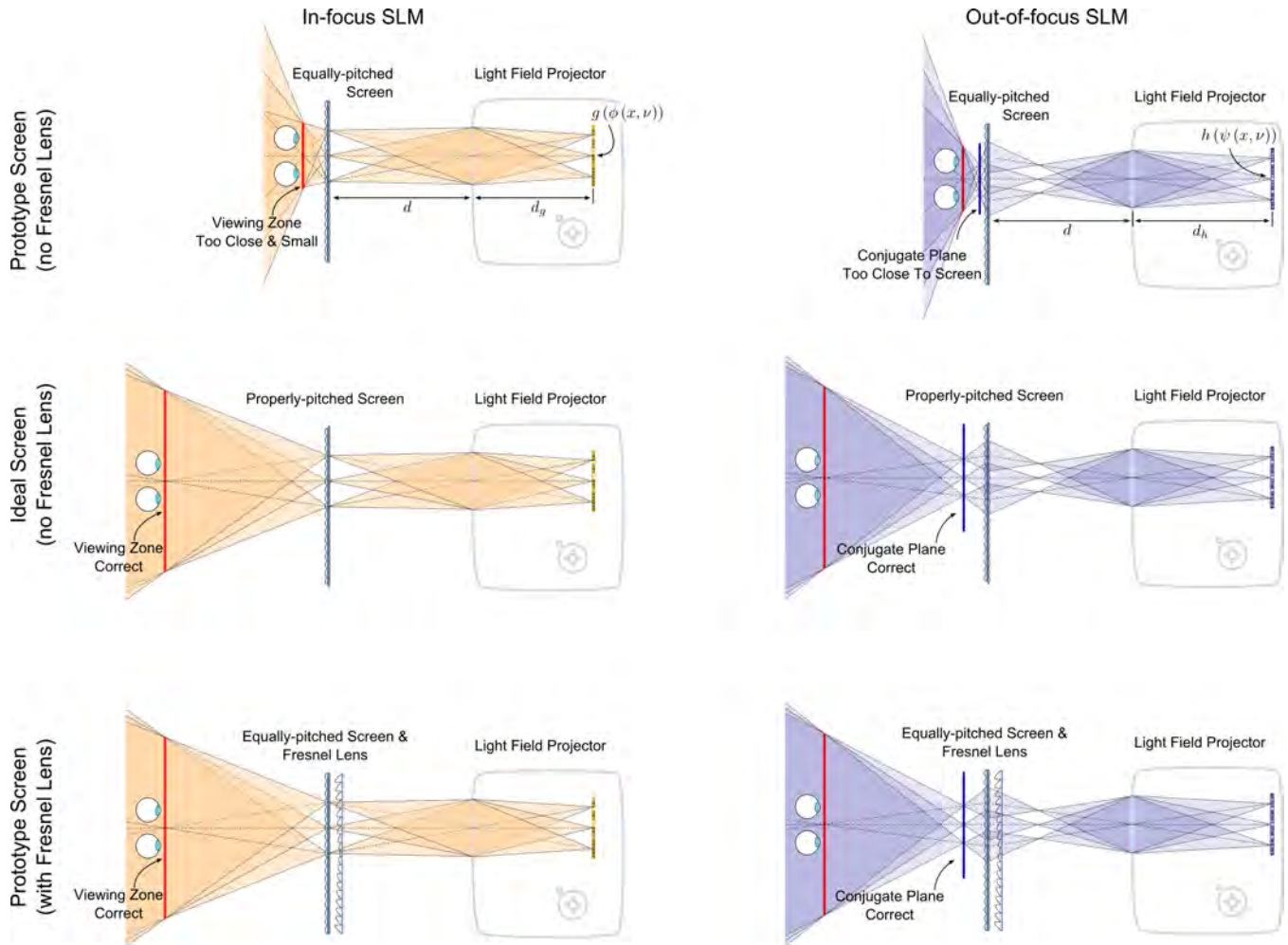

Figure S.1: *Illustration of ray paths for in-focus and out-of-focus SLMs through the screen. The ideal screen (center row) is a super-lens: angles are not only expanded by lenticulars with different focal lengths but also focused though a different pitch of the two lenticular sheets. Our prototype screen has the same lenticular pitch (top), which shifts the viewing zone too close to the screen. We compensate for this effect using an additional Fresnel lens (bottom), which would not be necessary in a commercial implementation.*

In Figure S.1, we illustrate the optical paths in more detail than in the primary text. As described in the paper, the in-focus SLM g is focused on the screen. Angular variation for the illumination cones of each pixel are expanded by the screen. If the lenticular of the screen have the same pitch, as is the case in the current prototype, this will create a viewing zone that is very close to the screen (top). The ideal screen (center) would employ lenticulars that have a different pitch, creating a viewing zone (red bar) at a desired distance. We can achieve the same effect

with an additional Fresnel lens (bottom), but introduce slight image distortions that would not occur for the "ideal" screen. For the research prototype, the setup on the bottom, however, gives us more flexibility to test different optical configurations.

The optical paths of the second SLM h are shown in the right column. An image is created in mid-air in front of the screen and re-imaged to the viewer side. The illumination paths intersect those of the in-focus SLM on the viewer plane, which means the observer can freely move inside the red area and see both images. The two SLM images are created with a distance in between them and they interact in a multiplicative fashion. Again, the "ideal" screen is approximated by the prototype and an additional Fresnel lens, which gives us more flexibility in the display prototyping stage.

B Derivation of Nonnegative Update Rules

In this section, we provide a derivation of the multiplicative update rules listed in the paper. We start by reviewing the objective function $J(\mathbf{g}, \mathbf{h})$:

$$
\begin{array}{ll}
\underset{\{\mathbf{g},\mathbf{h}\}}{\text{minimize}} & J(\mathbf{g}, \mathbf{h}) = \frac{1}{2} \left\| \beta \mathbf{l} - \sum_{t=1}^{T} (\boldsymbol{\Phi} \mathbf{g}_t) \circ (\boldsymbol{\Psi} \mathbf{h}_t) \right\|_2^2 \\
\text{subject to} & 0 \le g_{ik}, h_{jk} \le 1, \ \forall i, j, k
\end{array}
\tag{S.1}
$$

Whereas J is a biconvex, nonlinear function. Most update rules fix on all parameters other than one, e.g. $\mathbf{g}_t$, and perform an update on the free variable. Such an update is done in a steepest descent fashion, i.e.

$$
\mathbf{g}_t \leftarrow \mathbf{g}_t - \alpha \nabla_{\mathbf{g}_t} J(\mathbf{g}, \mathbf{h}),
\tag{S.2}
$$

where α is the step length and the descent direction is given by the gradient of the objective function $\alpha \nabla_{\mathbf{g}_t} J(\mathbf{g}, \mathbf{h})$.

The gradient w.r.t. $\mathbf{g}_t$ can be calculated with matrix algebra and follows the form $\nabla_x \frac{1}{2} (\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{A}^T \mathbf{b} - \mathbf{A}^T \mathbf{A}\mathbf{x}$:

$$
\begin{aligned}
\nabla_{\mathbf{g}_t} J(\mathbf{g}, \mathbf{h}) &= \widetilde{\boldsymbol{\Phi}}^T \left(\beta \mathbf{l} - \sum_{j=1, j\neq t}^{T} (\boldsymbol{\Phi} \mathbf{g}_j) \circ (\boldsymbol{\Psi} \mathbf{h}_t) \right) - \widetilde{\boldsymbol{\Phi}}^T \widetilde{\boldsymbol{\Phi}} \mathbf{g}_t \\
&= \beta \widetilde{\boldsymbol{\Phi}}^T \mathbf{l} - \widetilde{\boldsymbol{\Phi}}^T \left(\widetilde{\boldsymbol{\Phi}} \mathbf{g}_t + \sum_{j=1, j\neq t}^{T} (\boldsymbol{\Phi} \mathbf{g}_j) \circ (\boldsymbol{\Psi} \mathbf{h}_t) \right) = \beta \widetilde{\boldsymbol{\Phi}}^T \mathbf{l} - \widetilde{\boldsymbol{\Phi}}^T \widetilde{\mathbf{l}}
\end{aligned}
\tag{S.3}
$$

where $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi} \operatorname{diag}(\boldsymbol{\Psi} \mathbf{h}_t)$, $\operatorname{diag}(\cdot) : \mathbb{R}^M \to \mathbb{R}^{M \times M}$ creates a diagonal matrix from a vector, and $\widetilde{\mathbf{l}} = \sum_{t=1}^{T} (\boldsymbol{\Phi} \mathbf{g}_t) \circ (\boldsymbol{\Psi} \mathbf{h}_t)$ is the current estimate of the light field given patterns on both displays.

For one particular choice of step length α, it will be possible to reformulate Equation S.2 such that all sums and differences disappear and only multiplications and divisions remain in the update. For an initial guess of the unknowns that is purely positive and a positive target light field $\mathbf{l}$, multiplicative update rules guarantee that no subsequent update will become negative. That step length is

$$
\alpha = -\frac{\mathbf{g}_t}{\widetilde{\boldsymbol{\Phi}}^T \widetilde{\mathbf{l}}}.
\tag{S.4}
$$

Combining and simplifying Equations S.2-S.4 results in the following multiplicative update rule for $\mathbf{g}_t$:

$$
\mathbf{g}_t \leftarrow \mathbf{g}_t \circ \frac{\boldsymbol{\Phi}^T (\beta \mathbf{l} \circ (\boldsymbol{\Psi} \mathbf{h}_t))}{\boldsymbol{\Phi}^T \left(\widetilde{\mathbf{l}} \circ (\boldsymbol{\Psi} \mathbf{h}_t) \right) + \epsilon}
\tag{S.5}
$$

A multiplicative rule for $\mathbf{h}_t$ can be derived in the same fashion as

$$\mathbf{h}_t \leftarrow \mathbf{h}_t \circ \frac{\mathbf{\Psi}^T \left(\beta \mathbf{l} \circ \left(\mathbf{\Phi} \mathbf{g}_t \right) \right)}{\mathbf{\Psi}^T \left(\widetilde{\mathbf{l}} \circ \left(\mathbf{\Phi} \mathbf{g}_t \right) \right) + \epsilon} \tag{S.6}$$

The value $\epsilon = 10^{-12}$ is added to prevent division by zero.

C Pseudo-Code

Here we outline pseudo-code for efficient implementations in OpenGL or Matlab and compare it to the mathematical formulations outline above. In general, the matrix-vector multiplications with $\mathbf{\Phi}$, $\mathbf{\Psi}$, and their transposes are implemented in a matrix-free fashion using multiview rendering and projective texture mapping.

Algorithm 1 OpenGL implementation of multiplicative update rules.

1: **init** $\mathbf{g}_t = rand(0,1)$, $\mathbf{h}_t = rand(0,1)$, for $t = 1 \dots T$
2: **for each** *iteration i*
3:    $\widetilde{\mathbf{l}} \leftarrow renderLightField\left(\mathbf{g}_{1\dots T}, \mathbf{h}_{1\dots T}\right)$
4:     **for each** *frame t*
5:        $\bar{\mathbf{l}} \leftarrow renderLightField\left(\mathbf{h}_t\right)$
6:        $\mathbf{g}_t \leftarrow \mathbf{g}_t \circ projectiveTextureMap\left(\beta \mathbf{l} \circ \bar{\mathbf{l}}, g\right) / \left(projectiveTextureMap\left(\widetilde{\mathbf{l}} \circ \bar{\mathbf{l}}, g\right) + \epsilon\right)$
7:     **end**
8:    $\widetilde{\mathbf{l}} \leftarrow renderLightField\left(\mathbf{g}_{1\dots T}, \mathbf{h}_{1\dots T}\right)$
9:     **for each** *frame t*
10:       $\bar{\mathbf{l}} \leftarrow renderLightField\left(\mathbf{g}_t\right)$
11:       $\mathbf{h}_t \leftarrow \mathbf{h}_t \circ projectiveTextureMap\left(\beta \mathbf{l} \circ \bar{\mathbf{l}}, h\right) / \left(projectiveTextureMap\left(\widetilde{\mathbf{l}} \circ \bar{\mathbf{l}}, h\right) + \epsilon\right)$
12:     **end**
13: **end**

In this algorithm, the function $renderLightField\left(\mathbf{g}_{1\dots T}, \mathbf{h}_{1\dots T}\right)$ renders a multiview light field of all layers and time frames, implementing $\widetilde{\mathbf{l}} = \sum_{t=1}^{T} \left(\mathbf{\Phi} \mathbf{g}_t\right) \circ \left(\mathbf{\Psi} \mathbf{h}_t\right)$. The function $renderLightField\left(\mathbf{h}_t\right)$ also renders a light field from the same perspective but only draws one of the layers at frame t.

The function $projectiveTextureMap\left(\beta \mathbf{l} \circ \bar{\mathbf{l}}, g\right)$ performs projective texture mapping of multiple images from the perspectives of the light field cameras onto a geometric representation of g, implementing $\mathbf{\Phi}^T \left(\beta \mathbf{l} \circ \bar{\mathbf{l}}\right)$. Similarly, $projectiveTextureMap\left(\beta \mathbf{l} \circ \bar{\mathbf{l}}, h\right)$ implements $\mathbf{\Psi}^T \left(\beta \mathbf{l} \circ \bar{\mathbf{l}}\right)$.

For the 2D superresolution image synthesis described in the paper, we replace the rendering and projective texture mapping functions with image convolutions. Additional Matlab scripts implementing these update rules for light field factorization and also for superresolution image synthesis are included as supplemental material.

D Additional Implementation Details

Figure S.2 shows top and side views of ideal optical paths and aberrated optical paths in the prototype. In the ideal case, a single light ray intersects a single pixel on the front layer but it is vertically diffused, which means the ray is affected by multiple pixels on the rear screen – even for perfect optics. The prototype observes scattering, interreflections, aberrations, and other non-ideal effects in addition to the vertical blur. We can partially compensate for these aberrations by capturing the point spread functions of a single pixel on front and rear screen, model the aberrations as convolutions with those PSFs, and incorporate them into the factorization algorithm. Therefore, the
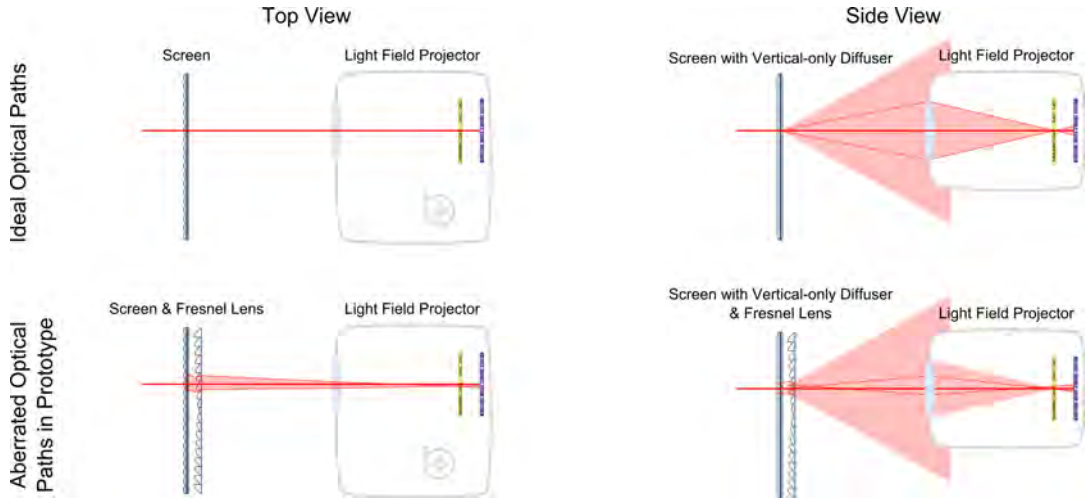
Figure S.2: *Illustration of optical path of a single ray of the light field on the observer side of the prototype horizontal-only parallax screen. When seen from above, the ray conceptually travels along a straight line through the screen and intersects each SLM on exactly one pixel (top right). A vertical diffuser in the screen spreads the ray out in that dimension (top right), eliminating possible vertical parallax but allowing the observer to freely move his head up and down. In the prototype screen (bottom), scattering and interreflections within the screen and between Fresnel lens and screen are observed in addition to the effects of the ideal optics.*

solver not only achieves light efficient image synthesis, but also partially corrects for aberrations and also limited dynamic range of the SLMs. All of these benefits cannot be achieved with purely optical designs, for instance using time-sequential parallax barriers.

The update rules for light field synthesis with aberrations include an additional convolution into Equations S.5 and S.6:

$$\mathbf{g}_t \leftarrow \mathbf{g}_t \circ \frac{\overline{\mathbf{\Phi}}^T \left(\beta \mathbf{l} \circ \left(\overline{\mathbf{\Psi}} \mathbf{h}_t \right) \right)}{\overline{\mathbf{\Phi}}^T \left(\widetilde{\mathbf{l}} \circ \left(\overline{\mathbf{\Psi}} \mathbf{h}_t \right) \right) + \epsilon} \tag{S.7}$$

$$\mathbf{h}_t \leftarrow \mathbf{h}_t \circ \frac{\overline{\mathbf{\Psi}}^T \left(\beta \mathbf{l} \circ \left(\overline{\mathbf{\Phi}} \mathbf{g}_t \right) \right)}{\overline{\mathbf{\Psi}}^T \left(\widetilde{\mathbf{l}} \circ \left(\overline{\mathbf{\Phi}} \mathbf{g}_t \right) \right) + \epsilon} \tag{S.8}$$

where $\overline{\mathbf{\Phi}} = \mathbf{\Phi} \mathbf{\Phi}_{blur}$ and $\overline{\mathbf{\Psi}} = \mathbf{\Psi} \mathbf{\Psi}_{blur}$. The matrices $\mathbf{\Phi}_{blur}$ and $\mathbf{\Psi}_{blur}$ encode a convolution with the measured point spread functions. In the OpenGL solver (Alg. 1), this is implemented as a matrix-free blur operation using a Cg shader.

E Additional Results for Superresolution and HDR Projection

Figure S.3 shows additional results for two-dimensional superresolution and high dynamic range projection. We show multiple frames from the "Big Buck Bunny" movie and two resolution targets. Improvements in contrast and resolution are observed for all simulated and captured results. The differences may be most obvious when zooming into the electronic copy of this figure and may be washed out in a printed version. In the physical experiments, the difference is obvious for a human observer, although we did not perform formal user experiments to formally verify this.

We would like to point out that we captured the point spread functions (PSFs) of both SLMs on the diffuse screen and used those for the image synthesis. For this purpose, we capture a single PSF in the center of the screen for each

SLM and assume a shift-invariant kernel over the screen area. In practice, the PSFs slightly vary over the screen. Slight artifacts in the captured results, for instance in the black horizontal bar just below the center of the resolution target (second last row), are due to the approximation of the spatially-varying PSF with a shift-invariant PSF. This could be accounted for in the solver and is not a fundamental limitation of the proposed system.

F   Additional Details on Possible Multi-device Configurations

In this section, we discuss possible extensions of the proposed light field projection systems to multi-device configurations. An illustration of possible system configurations via different combinations of projector and screens is shown in Figure S.4.

Horizontal and Vertical Parallax   Combined with 2D lenslet arrays in the angle-expanding screen, multiple light field projectors can be employed. For this purpose, a regular grid of projectors can be mounted such that their apertures are densely spaced. Note that lenslets produce periodic viewing zones; this fact can be exploited by spacing projectors further apart yet at the virtual position of the dense setup in the periodic viewing zones. The screen lenslets can even be fabricated with astigmatism, that is with different focal lengths for the horizontal and vertical dimension. The main benefit of multi-device full parallax systems are *flexible and extended viewing zones or fields of view*.

Horizontal-only Parallax   As with single-device setups, a "screen sandwich" of semi-cylindrical lenticulars and a vertical-only diffuser can be used for horizontal-only parallax configurations. Placing multiple projectors next to each other allows for extended fields of view, whereas stacking them vertically overlays their contributions and facilitates higher-rank light field synthesis (see discussion in the primary text). The image formation for D vertically stacked devices, each with a 1D aperture SLM, is

$$\widetilde{\mathbf{l}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{d=1}^{D} \left(\mathbf{\Phi}^{(d)} \mathbf{g}_t^{(d)} \right) \circ \left(\mathbf{\Psi}^{(d)} \mathbf{h}_t^{(d)} \right), \tag{S.9}$$

where $\mathbf{\Phi}^{(d)}$ and $\mathbf{\Psi}^{(d)}$ map the SLM images of device d to the emitted light field. Straightforward modifications of the multiplicative update rules can be employed to compute the light field decompositions for all devices. The combined formulation for a horizontal-only parallax screen with multiple vertically-arranged projectors is

$$\mathbf{g}_t^{(d)} \quad \leftarrow \quad \mathbf{g}_t^{(d)} \circ \frac{\mathbf{\Phi}^{(d)T} \left(\beta \mathbf{l} \circ \left(\mathbf{\Psi} \mathbf{h}_t^{(d)} \right) \right)}{\mathbf{\Phi}^{(d)T} \left(\widetilde{\mathbf{l}} \circ \left(\mathbf{\Psi} \mathbf{h}_t^{(d)} \right) \right) + \epsilon}$$

$$\mathbf{h}_t^{(d)} \quad \leftarrow \quad \mathbf{h}_t^{(d)} \circ \frac{\mathbf{\Psi}^{(d)T} \left(\beta \mathbf{l} \circ \left(\mathbf{\Phi} \mathbf{g}_t^{(d)} \right) \right)}{\mathbf{\Psi}^{(d)T} \left(\widetilde{\mathbf{l}} \circ \left(\mathbf{\Phi} \mathbf{g}_t^{(d)} \right) \right) + \epsilon} \tag{S.10}$$

We evaluate light field compressibility for a varying number of devices in Figure S.5. We note that adding more devices to this particular setup has the same effect as increasing the speed of SLMs in a single device but the additional benefit of increased brightness. With this experiment, we reveal another important insights of this paper: the proposed compressive display architecture allows for a direct tradeoff between speed of employed SLMs and number of devices. While previously light field projection was possible with multi-device setups, the proposed compressive system not only makes these more efficiently but also opens a wide system parameter design space and facilitates single-device light field projection as one possible option. However, we did not build such a multi-device system in practice.
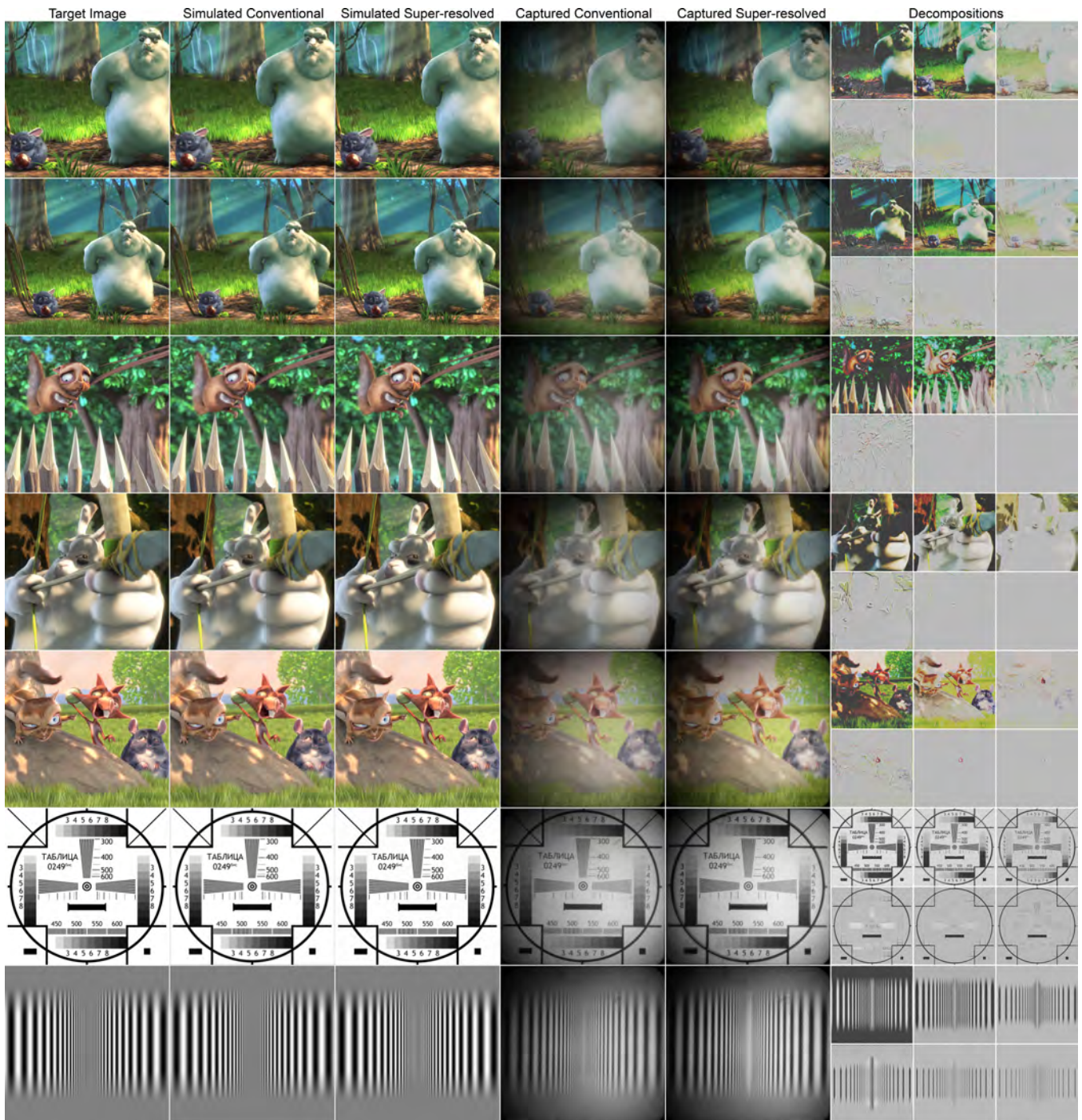
Figure S.3: *Additional results for superresolution and high dynamic range projection. We show (from left): target high-resolution image, simulated projection with a single SLM that has a lower contrast and resolution than the target, simulated result achieved with proposed method, captured result for a single physical SLM, captured result with the proposed method, and decomposed frames for both SLMs and a rank-3 image formation. The differences between conventional and super-resolved method become more obvious when zooming in.*
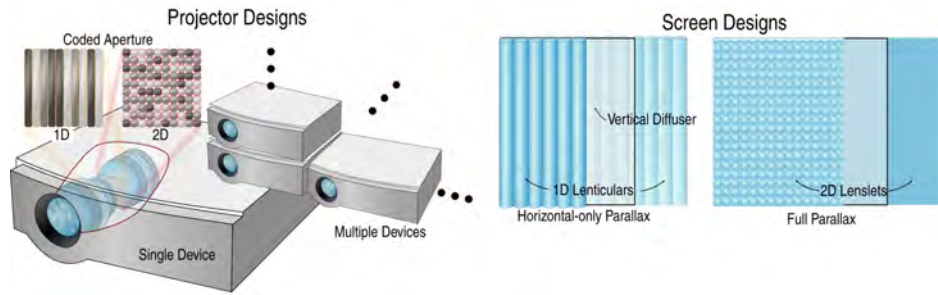
Figure S.4: *Illustration of different screen and projector configurations. Horizontal-only and full parallax can be achieved with a single device using screens composed of arrays of 1D lenticular or 2D lenslet, respectively. Multiple devices allow for wider fields of view and higher-rank light field decompositions.*
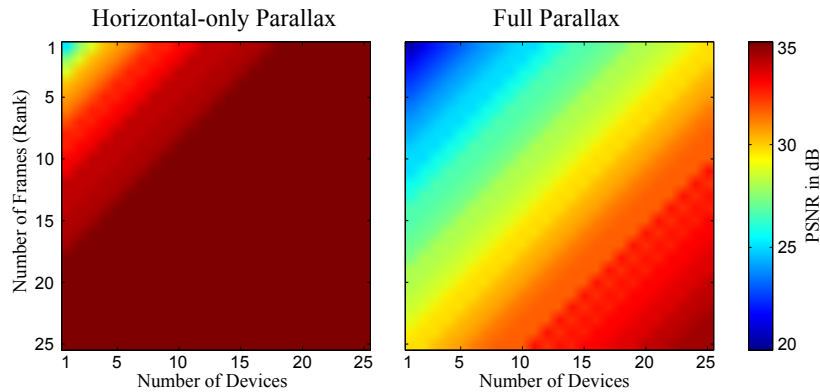

Figure S.5: *Light field compressibility. We simulate reconstructions of the "t-rex" scene within $10°$ field of view for a varying number of devices and rank of each device. While full-parallax systems may require either higher-speed modulators or many devices, moderate fields of view can be achieved equally well with a single high-speed projector or a few lower-speed devices. The proposed display architecture facilitates a direct tradeoff between the number of devices and the speed of employed SLMs.*

G   Experiments with Coded Aperture Light Field Projector Prototype

In this section, we document our efforts on implementing the proposed system with a coded aperture light field projection instead of a duallayer projector. Using off-the-shelf optics, we could not achieve a high-enough image quality with this prototype. We experimented mostly with the reflective LCoS modules and disassembled a Canon EF 50mm f/1.8 camera lens, which was subsequently used as a projection lens. An illustration of the setup is shown in Figure S.6 and a photograph of the setup in Figure S.7. Adding the LCoS into the aperture of the projection lens
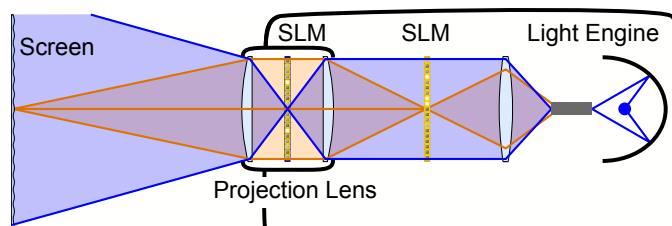

Figure S.6: *Coded aperture module and projector illumination. While the image SLM is conjugate to the screen (orange paths), the aperture SLM is conjugate to the output of the light engine, which produces a clean area source (blue paths).*

required us to disassemble the compound lens and increase the spacing between individual elements to fit the LCoS and a polarizing beam splitter cube. This modification results in optical aberrations that severely reduce observed image quality. Custom-design optics or transmissive SLMs could overcome these limitations.
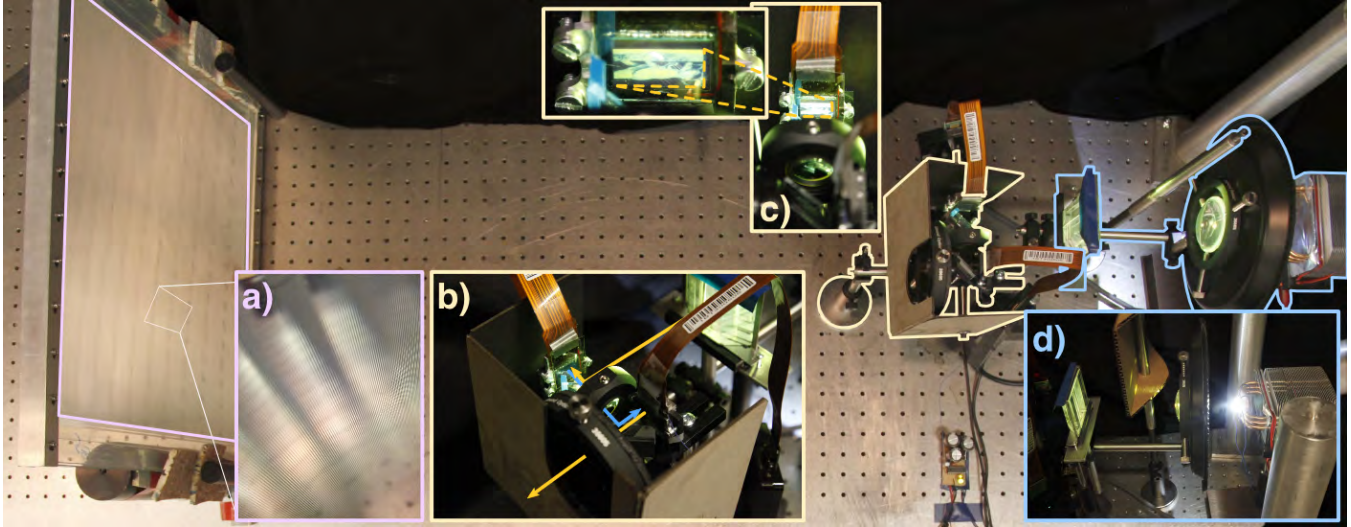

Figure S.7: *Angle-expanding screen (a), composed of back-to-back lenticular sheets, and a collimating Fresnel lens. Modified projection lens (b), with an SLM at the image and aperture planes of the projector. (c) Spatial light modulator at the image plane of the coded aperture projector, showing one frame of the "t-rex" scene. (d) 10W LED, collimating lens, and minifying and flattening prism.*

We show a quantitative evaluation of light field compressibility for a single projector in Figure S.8. Both horizontal-only and full parallax light fields are considered for decompositions with rank 6, 12, and 18. In all cases, the target light field has 25 views equally spaced over the entire field of view (FOV). Intuitively, light fields containing only horizontal parallax are much more compressible, which is confirmed by higher peak-signal-to-noise (PSNR) values. As the FOV increases, compressibility of the light field decreases due to a larger amount of parallax.
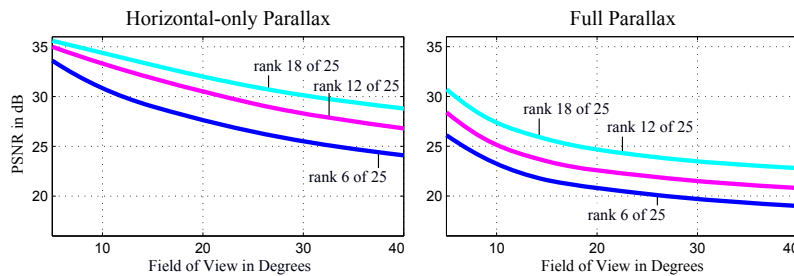

Figure S.8: *Light field compressibility. We simulate reconstructions of the "t-rex" scene for a varying field of view. The target light field has either 25 or 5 × 5 views equally distributed in a horizontal-only (left) or horizontal and vertical (right) viewing zone, respectively. Horizontal-only parallax light field are much more compressible; higher-rank decompositions achieve a better quality. We observe that even a rank-6 decomposition with fields of view up to 15° achieve high-quality reconstructions.*

Figure S.9 shows the prototype from several positions. The scene contains a teapot. Slight parallax is observed in the individual images, but the 2D quality is low. Variations in intensity are mainly due to imperfections in the illumination path of the projection system. This result is created by decomposing a target light field with six views into a rank-6 representation but at double the image brightness than what could be achieved with a conventional sweeping slit in the aperture. Both image and aperture SLM are synchronized at 240 Hz.
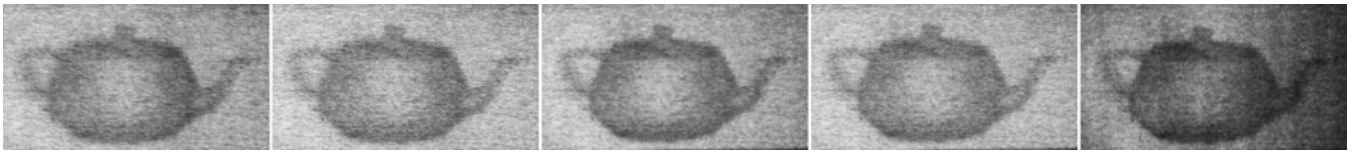
Figure S.9: *Photographs of early coded aperture prototype showing a scene with a teapot from several viewpoints.*

In Figure S.10, we compare the "direct" sweeping slit approach with the proposed compressive approach to light field synthesis. A target light field with six horizontal views is decomposed into a set of six pairs of patterns. Each pair has one pattern for the image SLM and one for the aperture SLM. While this decomposition does not achieve any compression the the number of viewpoints, we increase the brightness of the target light field by a factor of two (Fig. S.10, top) with respect to what is achieved by a sliding slit (Fig. S.10, bottom). Photographs with similar camera exposure times (Fig. S.10, center and bottom) reveal how the proposed approach allows for a significantly increased brightness of the displayed images.
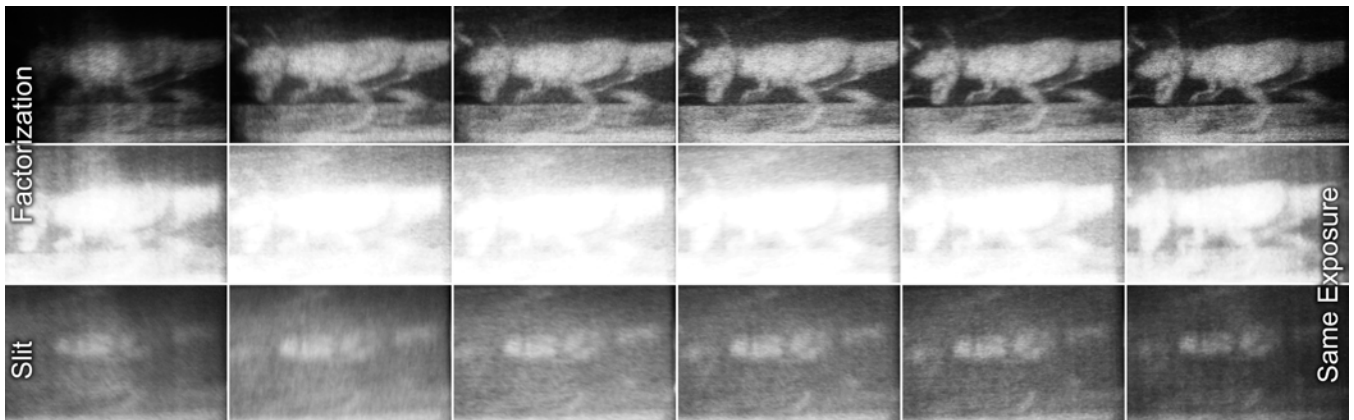

Figure S.10: *Photographs of early coded aperture prototype showing the "t-rex" scene. We show the brightness benefits achieved with compressed light field display (top row) as compared to displaying the target light field views in a time-sequential manner with a synchronized transparent slit in the projector aperture (bottom row). The center row shows the factorized projection at the same exposure time as the sweeping slits. Low image quality for the latter is due to the limited LCoS contrast.*