Supplementary Material:
Eyeglasses-free Display: Towards Correcting Visual Aberrations with Computational Light Field Displays

Fu-Chung Huang[1,4] Gordon Wetzstein[2] Brian A. Barsky[1,3] Ramesh Raskar[2]

[1]Computer Science Division, UC Berkeley [2]MIT Media Lab
[3]School of Optometry, UC Berkeley [4]Microsoft Corp.

This document contains additional implementation results, light field raw data not shown in the papers, and details on evaluations. Appendix A derives the coordinate transforms in the paper. Appendix B extend the discussion of the light field prefiltering in flatland. Appendix C explains the details in constructing the light field display prototype. Appendix D shows the implementation details about the prefiltering of the light field image. Appendix E illustrates the experiment set up and photographs. Appendix F gives the complete evaluation figure in the paper. Appendix G compares the higher order aberrations and their corrections. Appendix H shows all the generated light fields raw data used in the paper.

A Derivation of the Light Field Transport

We refer the readers to [Liang et al. 2011], [Lanman 2011], or [Huang 2013] for a comprehensive derivations and analysis of the general light field transport; in this section, we derive a much simpler geometric mapping between the display light field l^d with the coordinates (x^d, u^d) and the retina light field l with the coordinates (x, u). Note that as with the main paper, the display light field shares the same angular plane with the retinal light field, i.e. $u^d = u$.

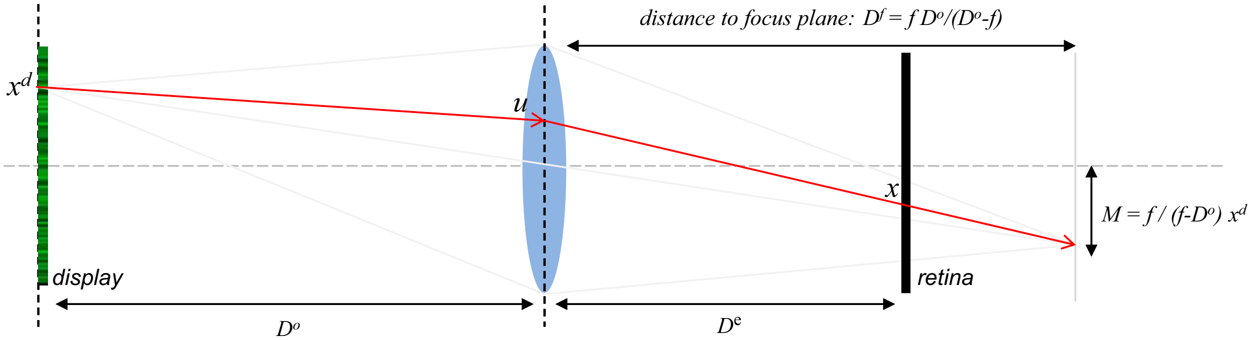

distance to focus plane: $D^f = f D^o/(D^o\text{-}f)$
x^d
u
x
$M = f/(f\text{-}D^o)\ x^d$
display
retina
D^o
D^e

Figure S.1: *Derivation of the geometric mapping $x^d = \phi(x, u)$ between the display light field and the retinal light field. For the light ray in red, it is refracted by the lens and intersect a retinal location x.*

When the display is located outside the focal range of the eye, as shown in Figure S.1, its focus plane is behind the retina. Following the thin lens equation, the distance D^f from the pupil plane to the focus plane is given by $D^f = fD^o/(D^o - f)$. Using the similar triangles, we can easily find the magnification and the corresponding location of x^d on the focus plane:

$$\frac{x^d}{D^o} = \frac{-M}{D^f} \quad \Longrightarrow \quad M = \frac{f}{(f - D^o)} x^d \tag{S.1}$$

Finally, we want to know where does the ray (x^d, u) (in red) intersects the retinal plane. Again, we can use the similar triangles on the right hand side:

$$\frac{(u - M)}{D^f} = \frac{(u - x)}{D^e} \quad \Longrightarrow \quad x^d \quad = \quad -\frac{D^o}{D^e}x + u\left(1 + \frac{D^o}{D^e} - \frac{D^o}{f}\right) \quad = \quad -\frac{D^o}{D^e}x + D^o\Delta u, \tag{S.2}$$

where $\Delta = \frac{1}{D^o} + \frac{1}{D^e} - \frac{1}{f}$, and this equation gives the mapping $x^d = \phi(x, u)$. Since the angular parameterization does not change, we obtain the final transform matrix:

$$\begin{pmatrix} x^d \\ u^d \end{pmatrix} = \begin{pmatrix} -\frac{D^o}{D^e} & D^o\Delta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ u \end{pmatrix}. \tag{S.3}$$

Finally, it is also easy to derive the corresponding transform in the frequency domain. According to the Fourier linear transform theorem [Ramamoorthi et al. 2007], the coordinate transform is the inverse transpose of the original matrix, i.e.

$$\mathbf{T}^{-T} = \begin{pmatrix} -\frac{D^e}{D^o} & 0 \\ D^e\Delta & 1 \end{pmatrix} \tag{S.4}$$

Note we drop the inverse discriminant term since it is only a scaling constant to the spectrum of the light field. It is straight forward to understand the two transforms: in the spatial domain, a defocused eye generates a scaled shearing when $\Delta \neq 0$. Since the defocus of the eye causes a shearing in x, in the frequency domain the light field will be sheared in the other direction ω_u, as indicated by the second transform matrix.

Given this coordinates transformation, it is easy to model the formation of the retinal image $i\left(x\right)$ from the retinal light field l by integrating rays along the angular domain Ω_u capped by the aperture function $A\left(u\right)$:

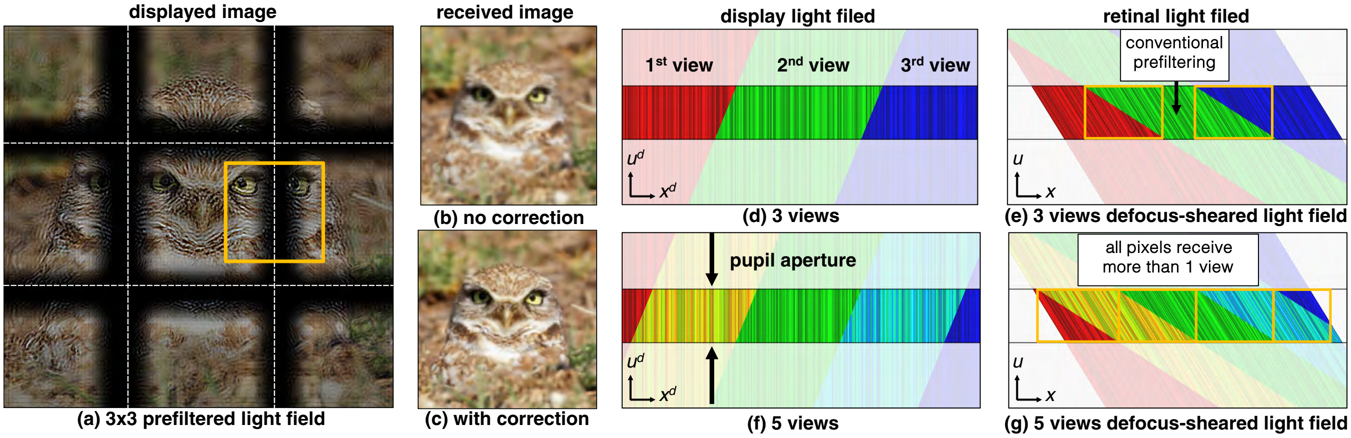
$$\begin{aligned} i\left(x\right) &= \int_{\Omega_u} l\left(x, u\right) du \\ &= \int_{-\infty}^{\infty} l^d\left(\phi\left(x, u\right), u\right) A\left(u\right) du. \end{aligned} \tag{S.5}$$

B Light Field Analysis

Figure S.2 shows an example of a prefiltered light field with 3 × 3 views for an sample scene. In this example, the different views contain overlapping parts of the target image, allowing for increased degrees of freedom for aberration compensation. Within each view, the image frequencies are amplified. However, the proposed inverse method operates as a 4D prefilter on the full light field, not just each view separately. When optically projected onto the retina of an observer, all views are integrated, which results in a perceived image that has significantly improved sharpness (c) as compared to an image observed on a conventional 2D display (b).

Figures S.2(d-g) illustrate the corresponding image formation on the retina using a flatland light field with the x-axis being lateral location and the y-axis the location on the pupil. Figures S.2(d and f) show the display light field with 3 and 5 views respectively; because the angular domain is parameterized on the pupil plane, some pixels will have neighboring views entering the pupil.

Incident on the retina is a sheared light field due to the defocus of the eye, and the perceived image is formed by vertically integrating the retinal light field (e and g) in the angular domain modulated by the white-shaded pupil function. The inverse problem of prefiltering light field views is only well-posed in regions where more than one light field view is observed simultaneously at any spatial location, indicated by the yellow boxes. If only a single light field view is observed at some location (e), the inverse problem is equal to that of vision-correction with a conventional 2D display: ill-posed. Increasing angular sampling with more views (g) removes the singularities by packing the well-posed regions tightly. In order to evaluate how many views exactly are necessary for successful vision-correction, we perform the condition analysis in the main paper.



(a) 3x3 prefiltered light field (b) no correction (c) with correction (d) 3 views (e) 3 views defocus-sheared light field (f) 5 views (g) 5 views defocus-sheared light field

Figure S.2: *Extended version of Figure 4 of the primary text. We show a prefiltered light field with* 3×3 *views on the left that results in a perceived image (c), which has a significantly higher sharpness than what would be observed on a conventional screen (b). We also show a 1D spatio-angular slice of the light field illustrating the principle of vision correction (d-g). In display space, the light field is parameterized on the display surface* x^d *and on the pupil* u^d *(d,f). The finite pupil size (translucent, white region) limits the emitted light field to the part that actually enters the eye. The same light field can be rendered in eye space, where* x *and* u *are the coordinates on the retina and the pupil (e,g). Here, the light field in eye space is a sheared version of its corresponding light field in display space. We plot two different examples: three views entering the pupil (d,e) and five views entering the pupil (f,g). Individual views are drawn in different colors. The problem of vision-correction is ill-posed if a position* x *on the retina only receives contributions from one light field view, whereas the same problem becomes well-posed as more than one light field views contribute to it (indicated by the yellow boxes in e, g).*

C Prototype Construction

The prototype is made of 2 spacing transparent acrylic sheets[1], as shown in Figure S.3 (a), and a printed pinhole mask, which we will attach in another file “pinhole_mask_x9.zip” as a reference for printing. The composited pinhole-based parallax barrier has a thin and small form factor, as shown in Figure S.3 (b) and (c), and it fits nicely on an iPod touch (b and d). The pitch of the pinhole is $75\mu m$ wide, and the array of pinholes are separated by $390\mu m$; the iPod Touch 4 has a $78\mu m$ pixel pitch (326DPI), and thus the pinhole array gives a 5-to-1 spatial-angular trade-off ratio. The mask to screen alignment requires some rotations and shift tweaking[2], and in Figure S.3 (e1 to e5), we show the alignment process of the mask on a prefiltered light field image under a defocused camera.

[1] 2”(W)x3”(H)x3/32”(T), Optically Clear Cast Acrylic (part # 8560K182) from McMaster: http://www.mcmaster.com/#acrylic/=q31oar

[2] The reader can find more detailed instruction at http://displayblocks.org/diycompressivedisplays/parallax-barrier-display/

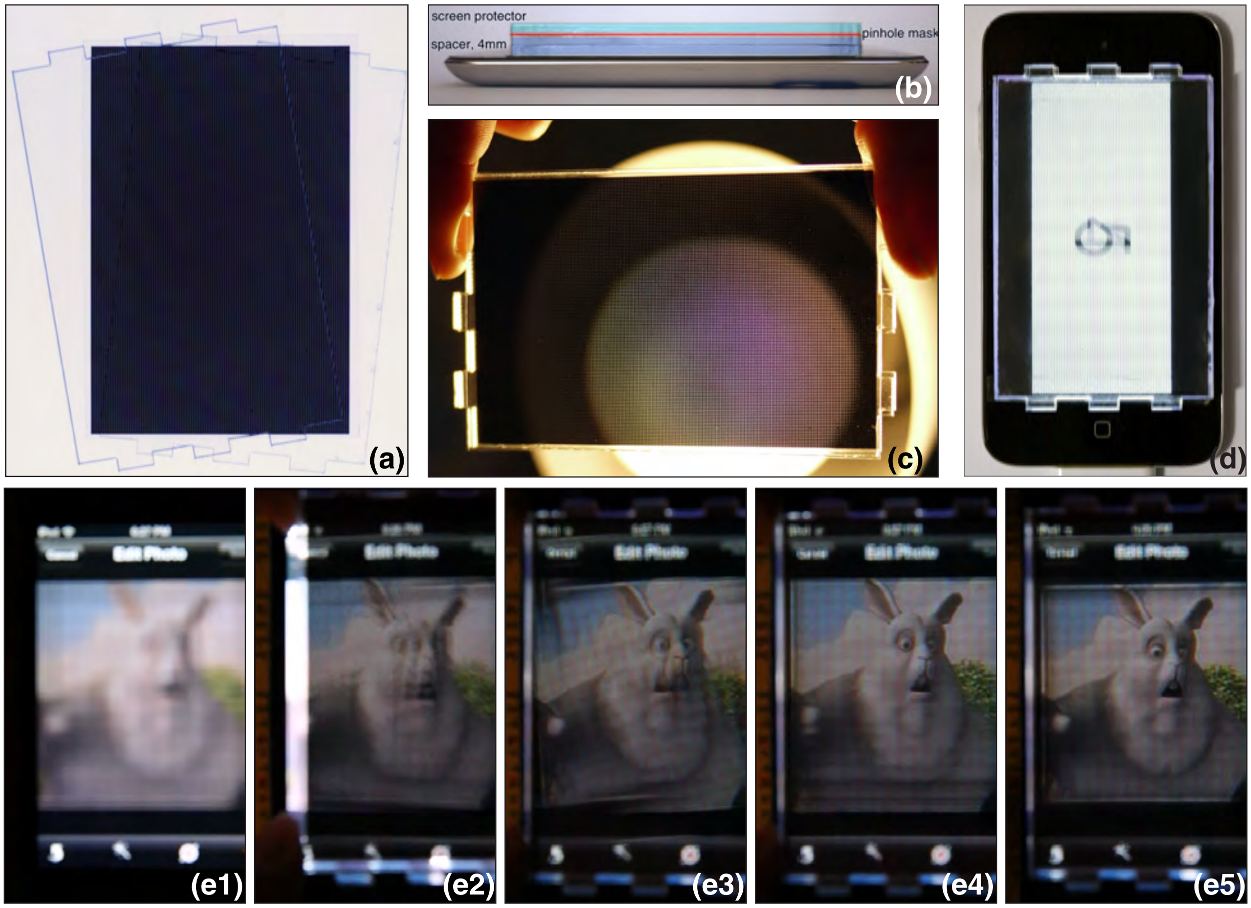


Figure S.3: *Prototype construction.*

D Software Implementation

In this section, we describe the core implementation that gives the light field transport matrix. The idea is to generate many ray-samples from the sensor, and then count the samples fall on the screen coordinates. We first assume a circular symmetric lens, so that X and Y are separable; the following function $Screen1DLightField()$ generates the light field on the screen side. First, given the sensor side light field formed by the spatial and angular samplings (line 3 and 4), we propagate the rays from the sensor to the screen (line 7). The screen side light field $[Y_o, V_o]$ is then converted to an index-based format (line 8 and 9).

```
 1 function [Yo, Vo] = Screen1DLightField() %gives the flatland light field on the screen side
 2 Screen_Sampling = linspace(Screen_Size/2, −Screen_Size/2, Screen_Res+1);
 3 Sensor_Sampling = linspace(Sensor_Width/2, −Sensor_Width/2, Sensor_Res+1);
 4 Angles_Sampling = −Aperture/2:(1/Sampling):Aperture/2;  %angular sampling on the aperture
 5 for(j = 1:Sensor_Res)
 6   %Do: camera to screen   f: camera focal length   Di: sensor to camera lens
 7   [Yo(:,j),Vo(:,j)] = Camera2ScreenTransport(Sensor_Sampling(j), Angles_Sampling, Do, f, Di);
 8   Yo(:,j) = dsearchn(Screen_Sampling, Yo(:,j));  %find the spatial pixel index on screen
 9   Vo(:,j) = convertViewIndex(Vo(:,j), Angular_Res);%find the angular view index on screen
10 end
```

With the above function giving the transported screen side light field, the following code counts the samples and generates the light field transport matrix $\mathbf{P}$ that relates the screen side light to the received image on the sensor. We ignore the code for aperture blocking, and the *samples* (line 9) gives samples of the screen side light field $[x_o, y_o, u_o, v_o]$ originated from sensor location $[i, j]$. Matlab built-in functions (line 10 and 11) accumulates and counts the repetitive samples that fall on the same discrete screen light field coordinate; so that $counts$ represents the 4D coordinates and its number of samples in $[x_o, y_o, u_o, v_o, \#]$. These records are then used to set the entries of the transport matrix $\mathbf{P}$ (line 14 to 17). Note that the transport matrix is a 2D-to-4D mapping such that $\mathbf{Pl} = \mathbf{i}$

```
 1 function P = BuildTransportMatrix()
 2 [Xo,Uo] = Screen1DLightField();   [Yo,Vo] = Screen1DLightField(); %separable 4D light field
 3 for(j = 1:Sensor_Res)                        %at each sensor location (i and j)
 4   for(i = 1:Sensor_Res)                      % retrieve the previously transported ray samples
 5     [xo,yo] = meshgrid(Yo(:,j), Xo(:,i));   %transposed for image x-y convention
 6     [uo,vo] = meshgrid(Uo(:,j), Vo(:,i));
 7     pixels  = [reshape(xo, [], 1) reshape(yo, [] ,1)];
 8     angles  = [reshape(uo, [], 1) reshape(vo, [] ,1)];
 9     samples = [pixels angles];  %each row is a record of 4D light ray; all starts from (i,j)
10     [b,m,n] = unique(samples, 'rows');      %matlab function accumulate the sampling
11     counts  = [b accumarray(n,1)];          % and counts the number of repetition.
12     %Now, each row of counts gives the 4D coordinates, and the number of samples fall into it
13     for(r = 1:size(b,1))          %for each coordinates and the # of samples, fill the matrix
14       index0 = (i-1)*Sensor_Res + (j-1);                                          %row index
15       index1 = ((counts(r,2)-1)*Screen_Res  + (counts(r,1)-1)  )*Angular_Res^2;   %col index
16       index2 = ((counts(r,4)-1)*Angular_Res + (counts(r,3)-1)  );    %sub-col index (angular)
17       P(index0+1, index1+index2+1 ) = counts(r,5); %set the # of samples to the 4D coord
18     end
19   end
20 end
```

Finally, to compute the desired screen side light field, we simply need to solve the inverse projection. As described in the paper, we use the solver package *LBFGSB* [Byrd et al. 1995][3] to solve the inverse problem.

```
img = reshape(im2double(imread(filename)), [],1); %vectorize the image
lf  = lbfgsb(P, img); %fast nonnegative inverse least square solver
% ... and then reshape lf to desired light field image format
```

D.1 Solving Over-Constrained Inverse Projections

In the paper, we describe the mechanism for handling changes in defocus and off-axis viewing; these conditions are straightforward to incorporate in our spatial domain solver. The linear system described in the previous section is depicted in the top left of Figure S.4; a single light field projection matrix is used, and the computed light field prefiltered image is shown in the top right. For the purpose of changing defocus or off-axis viewing, an over-constrained linear system considers differently scaled or shifted perceived image, as shown in the lower left of Figure S.4; an example of the prefiltered light field image for off-axis viewing is shown in the lower right.

[3] Matlab wrapper code: http://www.mathworks.com/matlabcentral/fileexchange/35104-lbfgsb-l-bfgs-b-mex-wrapper/

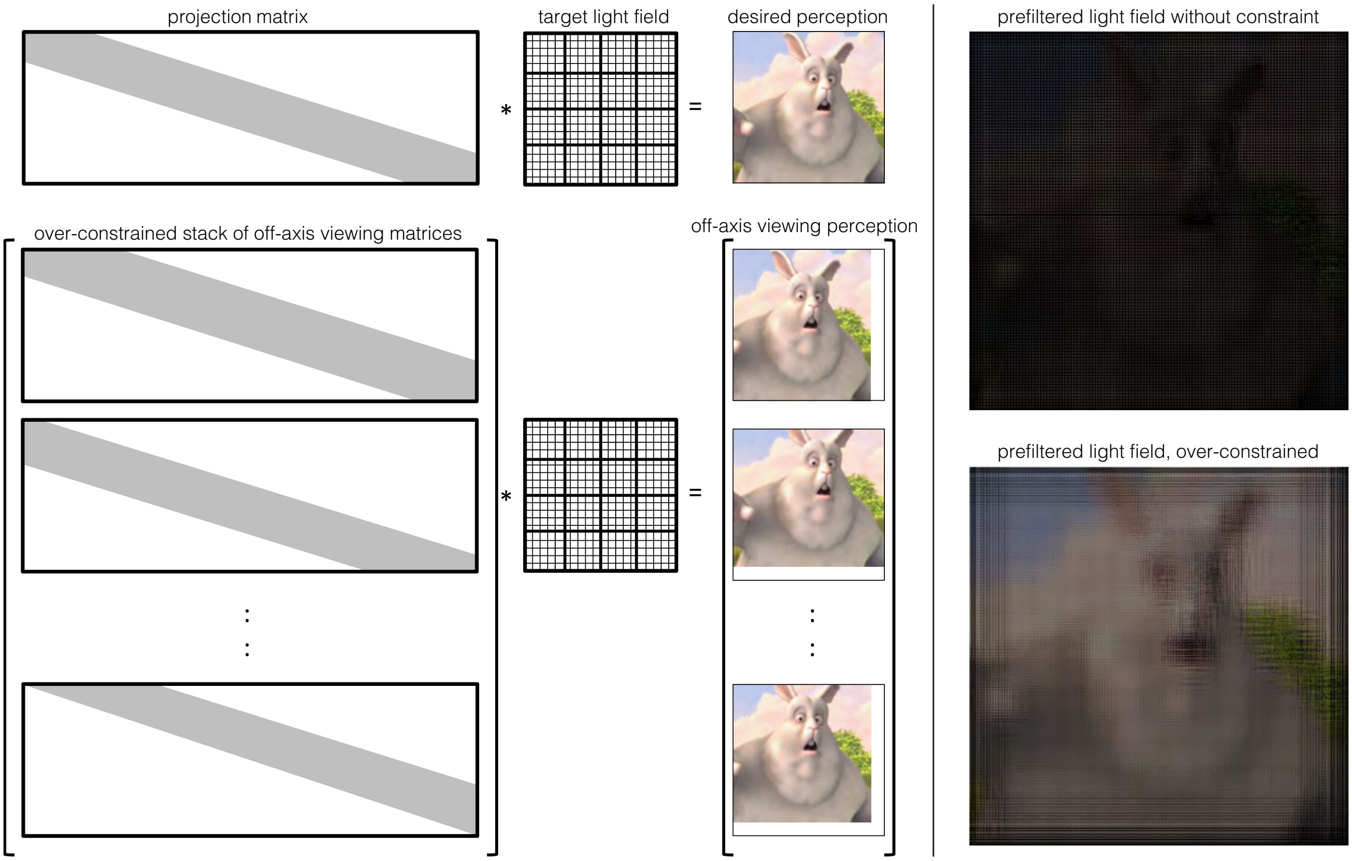
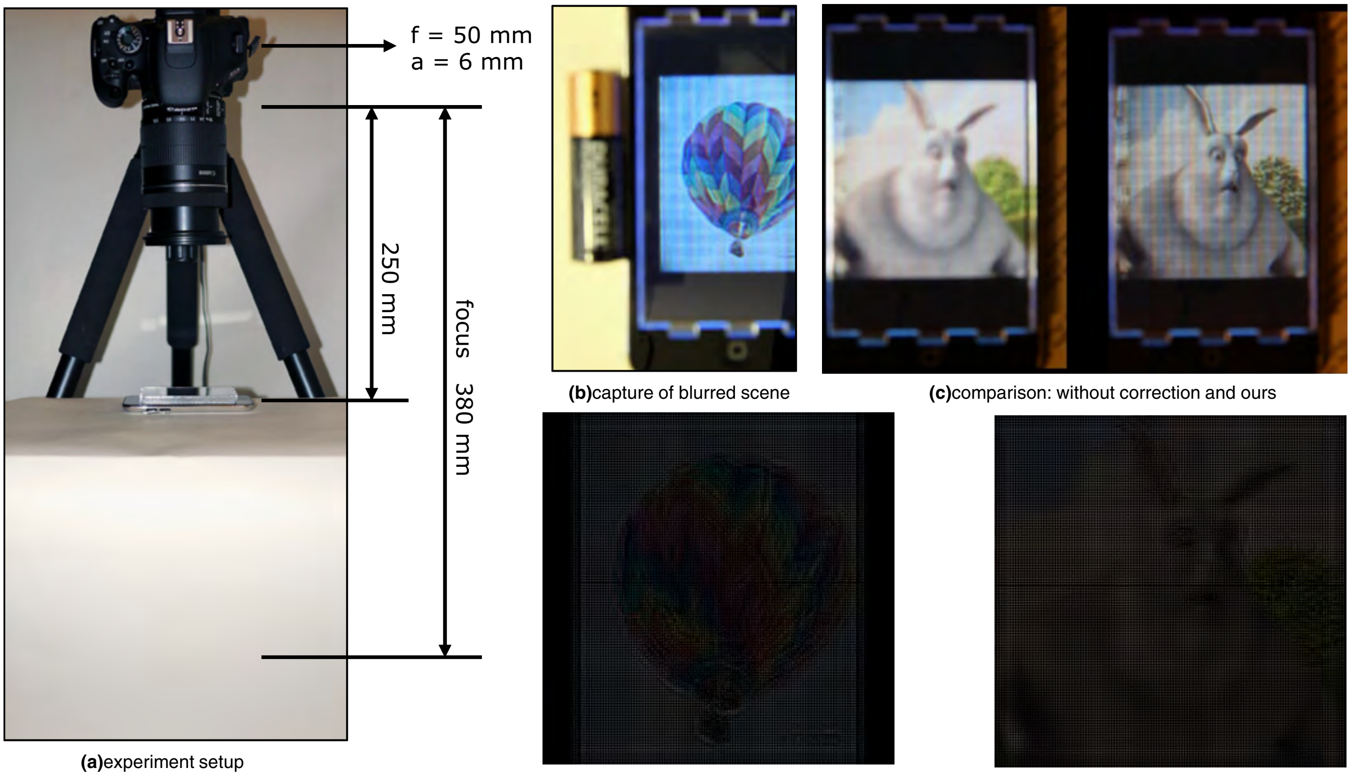

projection matrix
target light field
desired perception
prefiltered light field without constraint
over-constrained stack of off-axis viewing matrices
off-axis viewing perception
prefiltered light field, over-constrained
*
=
*
=

Figure S.4: *Illustrations of system building.*

E Experiment Setup

We test our prototype with a Canon T3i DSLR camera, and the 50mm lens is stopped at f/8. The focus plane is 380mm away, and the optical system emulates a -6D hyperopic eye reading at 250mm cellphone. Photographs of the system setup is in Figure S.5(a). The receiving blurred scene is in (b), where a battery is put beside the display to show how blurred the scene is; in the meantime the imagery shown on the screen is corrected with our method. We demonstrate the short animation used by Huang et al.[2012], as shown in Fig.S.5(c). Direct comparison is not shown since we are able to recover the full contrast. The sharpness is directly compared with that without any correction: the face and the background are all sharp using our light field prefiltering. Finally, the prefiltered light field raw data is shown in (d). Note that even the light field display we constructed has angular resolution 5-to-1, our method only uses 3 out of the 5 views; hence some pixels are left black and unused.



(a)experiment setup

(b)capture of blurred scene

(c)comparison: without correction and ours

(d)prefiltered light field for examples shown in (b) and (c)

Figure S.5: *(a) Experiment setup emulated a -6D hyperopic eye. (b) and (c) The blurred scene and the captured photographs. (d) The prefiltered light field raw data used to test the experiments in (b) and (c).*

F Evaluations

In Figure. S.6 we show the complete version of the simulated evaluations in the paper with HDR-VDP2[Mantiuk et al. 2011]. Our method generally outperforms previous work in both the numerical and the visual appearances.

G Correcting for higher order aberrations.

In Figure. S.7 we show the point spread functions for different wavefront geometries using single termed Zernike polynomials and the randomly combined wavefront on the last column. Comparing the lower order defocus term in (a) and higher order spherical term in (c), they are similar in shapes but quite different in the results, and are both difficult to correct using conventional display. In practice, we found light field prefiltering corrects all aberrations quite well, with slight degradation for the coma case in (d).

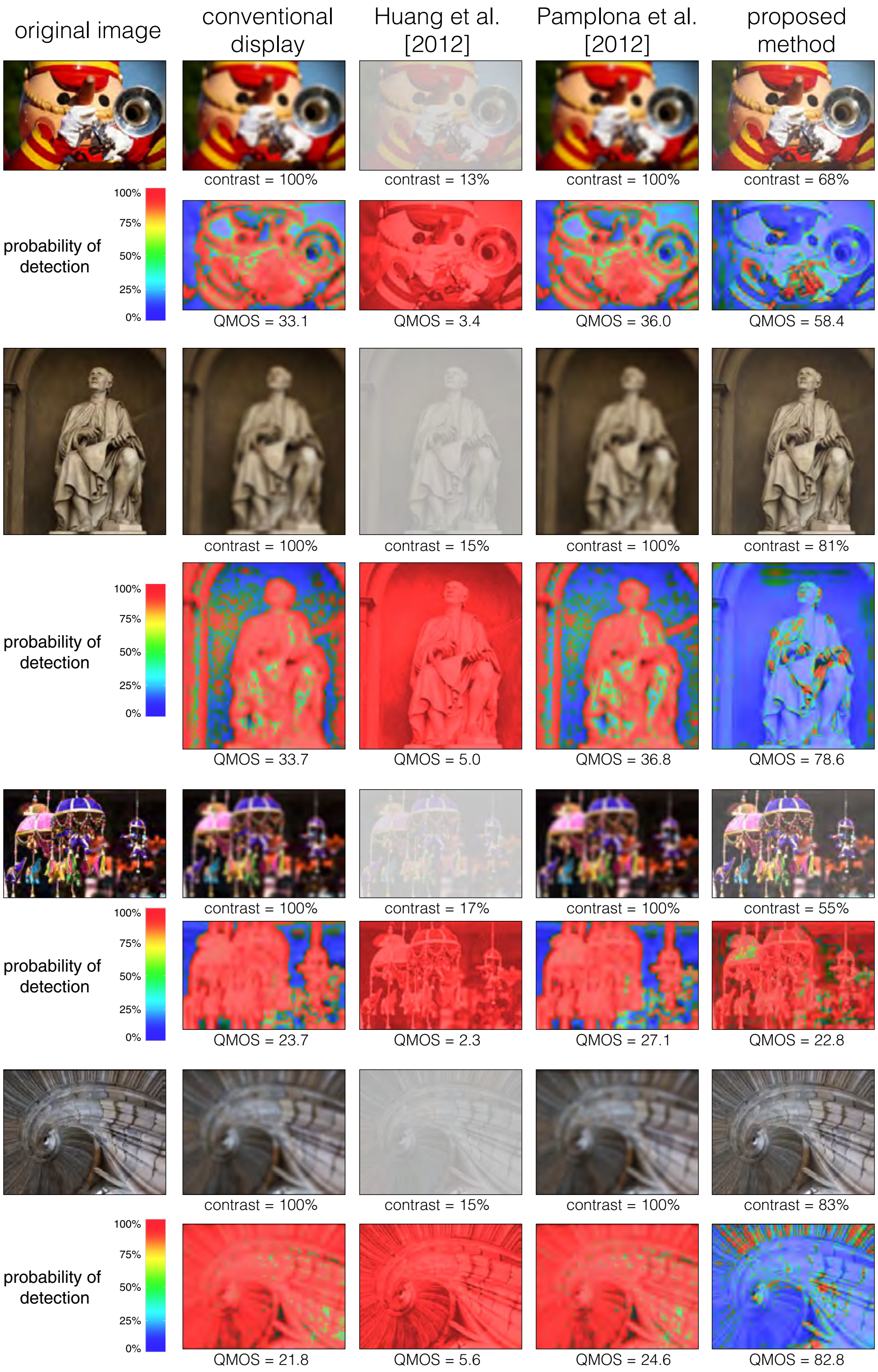


Figure S.6: *Complete evaluations.*

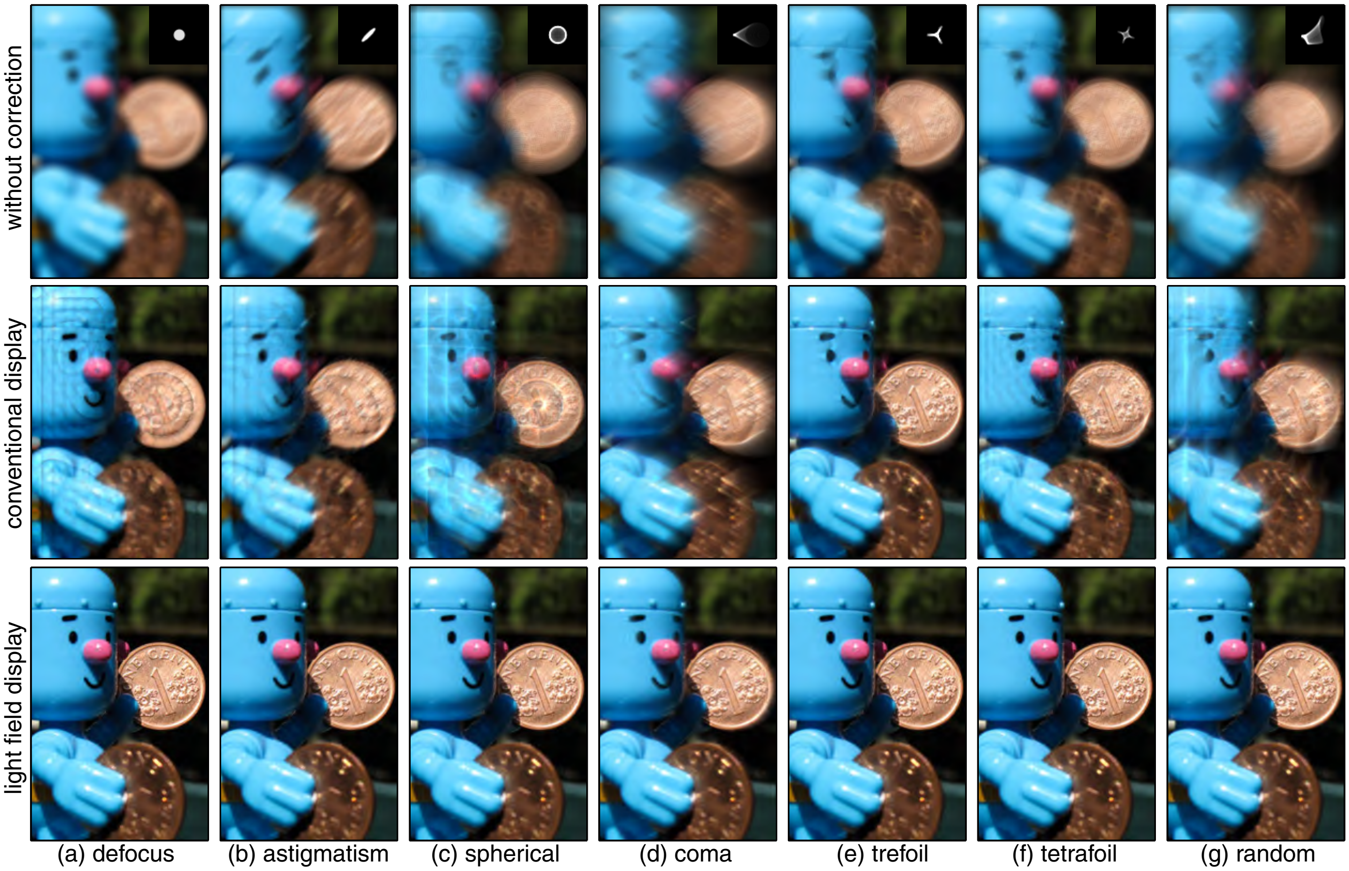


Figure S.7: *Corrections for different aberrations. The lower order terms includes (a) defocus and (b) astigmatism, and the rest are all higher order terms.*

H Raw Light Field Data

H.1 Teaser

The raw data used to generate the teaser is shown in Fig. S.8. The underlying block resolution for the light field predistortion is 8-by-8, and the received image has spatial resolution of only 96-by-96; the block resolution is 3-by-3 in our method, and the received resolution is 256-by-256. This data is designed for 4.5 myopic eye looking at display 350mm away.

H.2 Implementation and Results

The implementation data is designed to be shown on our prototype, and only a 3-by-3 out of the 5-by-5 block will enter the pupil as designed. Raw data are shown in Figure S.9, and since the light field prefiltering only uses 36% of the pixels, the raw images look darker. Our implementation of light field predistortion[Pamplona et al. 2012] re-samples the sheared light; for simplicity, all 5-by-5 angles are filled with data.

The printing resolution constrains the angular resolution to be a fixed 5-by-5 block on an ipod touch 4; and the spatial resolution is 128-by-128 over a 4.9cm-by-4.9cm square region on the display. And since the same hardware is shared for both light field distortion and light field prefiltering, their resolution will be the same. The preprocessed images are also gamma corrected. The data is designed for 6D hyperopic eye looking at display 250mm away.

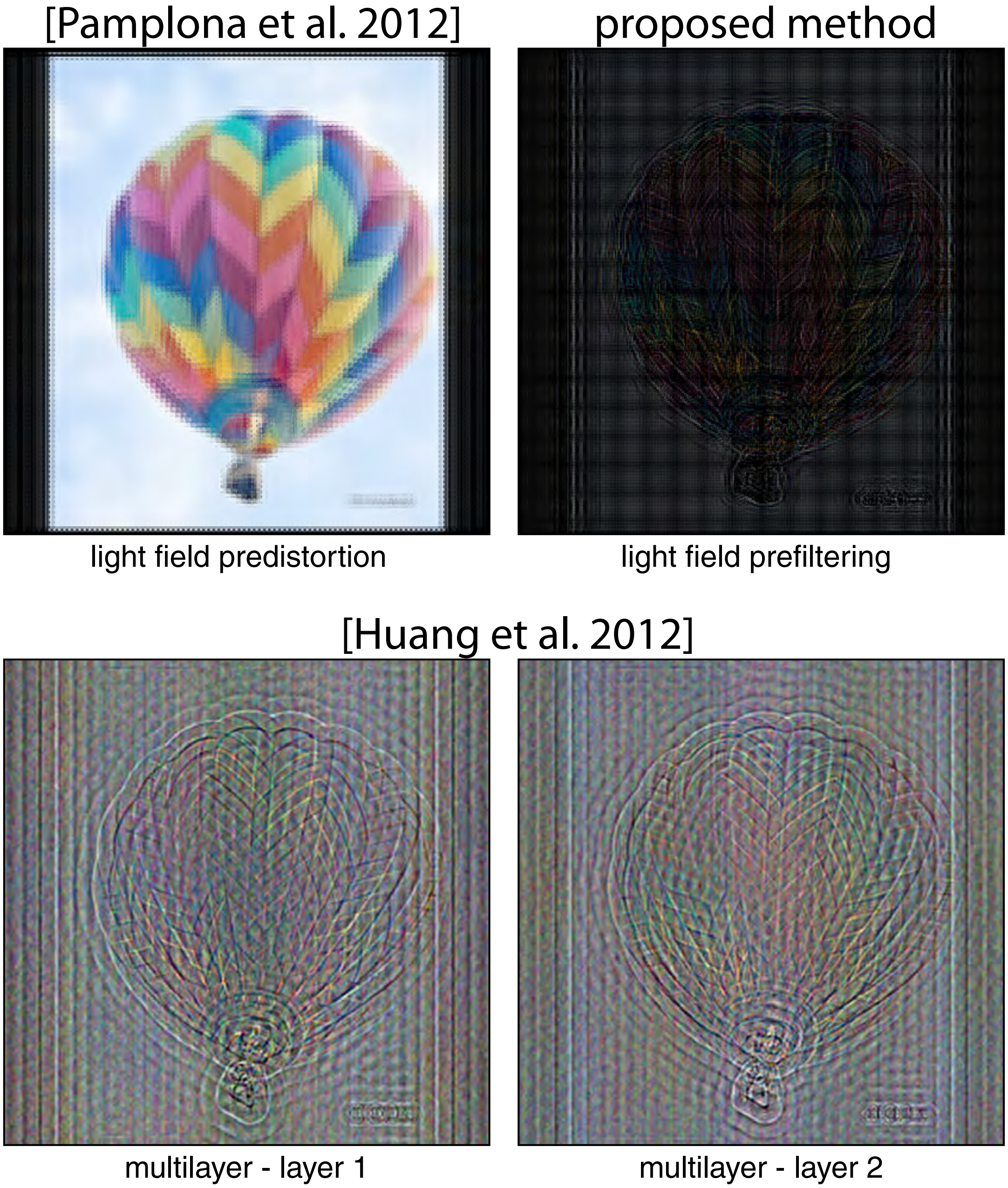


Figure S.8: *Teaser raw data.*

H.3 Evaluations

Finally, the data in evaluation implemented for Google Nexus 10 with 300PPI(84.6 micron pitch). Similar to the iPod Touch implementation, on the 3-by-3 angular block out of the underneath 5-by-5 group is used (36% brightness). The spatial resolution is 256-by-256 over a 10cm-by-10cm square region on the display. The data is designed for 6.75D hyperopic eye looking at display 300mm away.

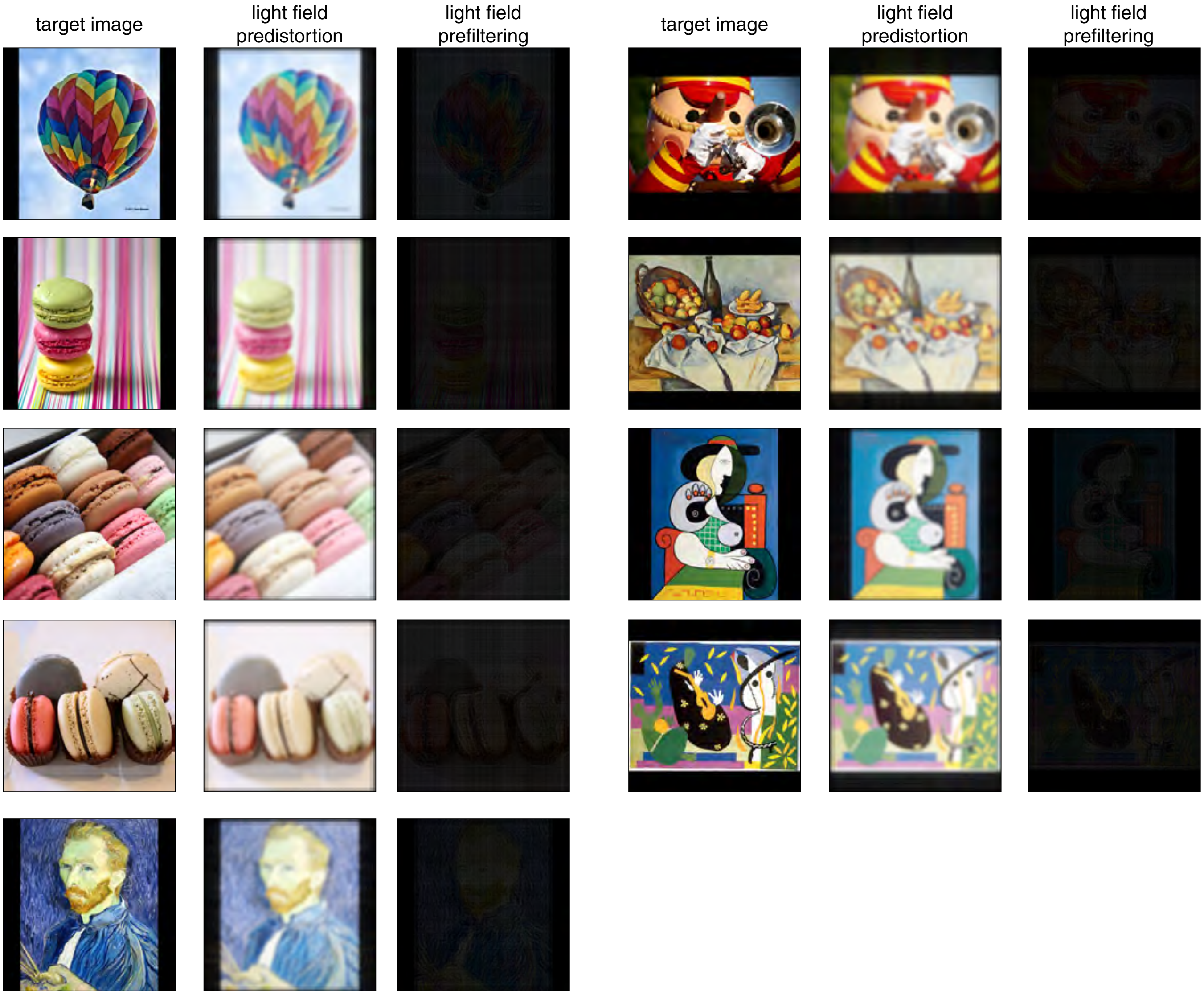


Figure S.9: *Raw data for the implementation and results section.*

Figure S.10: *Raw data for the evaluation section.*

Supplementary References

BYRD, R. H., LU, P., NOCEDAL, J., AND ZHU, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput. 16*, 5 (Sept.), 1190–1208.

HUANG, F.-C., LANMAN, D., BARSKY, B. A., AND RASKAR, R. 2012. Correcting for optical aberrations using multilayer displays. *ACM Trans. Graph. (SIGGRAPH Asia) 31*, 6, 185:1–185:12.

HUANG, F.-C. 2013. *A Computational Light Field Display for Correcting Visual Aberrations*. PhD thesis, EECS Department, University of California, Berkeley.

LANMAN, D. 2011. *Mask-based Light Field Capture and Display*. PhD thesis, Providence, RI, USA. AAI3479705.

LIANG, C.-K., SHIH, Y.-C., AND CHEN, H. 2011. Light field analysis for modeling image formation. *Image Processing, IEEE Transactions on 20*, 2 (feb.), 446 –460.

MANTIUK, R., KIM, K. J., REMPEL, A. G., AND HEIDRICH, W. 2011. Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *Proc. ACM SIGGRAPH*, 40:1–40:14.

PAMPLONA, V., OLIVEIRA, M., ALIAGA, D., AND RASKAR, R. 2012. Tailored displays to compensate for visual aberrations. *ACM Trans. Graph. (SIGGRAPH) 31*.

RAMAMOORTHI, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph. 26*, 1.