Metameric Varifocal Holograms: Supplementary Material

David R. Walton* UCL
Koray Kavaklı† Koç University
Rafael Kuffner dos Anjos‡ University of Leeds
David Swapp§ UCL
Tim Weyrich¶ UCL
Hakan Urey‖ Koç University
Anthony Steed** UCL
Tobias Ritschel†† UCL
Kaan Akşit‡‡ UCL

Index Terms: Computing methodologies—Computer graphics—Graphics systems and interfaces—Perception; Hardware—Communication hardware, interfaces and storage—Displays and imagers

1 Holographic Image Propagation

As depicted in Fig. 1, when such holograms are illuminated with a collimated coherent light (e.g. laser), these holograms can reconstruct an intended optical field at target depth levels. How light travels from a hologram to a parallel image plane is commonly described using Rayleigh-Sommerfeld diffraction integrals [2]. The first solution of the Rayleigh-Sommerfeld integral, also known as the Huygens-Fresnel principle, is expressed as follows:

$$u(x,y) = \frac{1}{j\lambda} \iint u_0(x,y) \frac{e^{jkr}}{r} cos(\theta) dx dy, \tag{1}$$

where the field at a target image plane, $u(x,y)$, is calculated by integrating over every point of the hologram's field, $u_0(x,y)$. Note that, for the above equation, r represents the optical path between a selected point over a hologram and a selected point in the image plane, θ represents the angle between these two points, k represents the wavenumber ($\frac{2\pi}{\lambda}$) and λ represents the wavelength of light. In this described light transport model, optical fields, $u_0(x,y)$ and $u(x,y)$, are represented with a complex value,

$$u_0(x,y) = A(x,y) e^{j\phi(x,y)}, \tag{2}$$

where A represents the spatial distribution of amplitude and ϕ represents the spatial distribution of phase across a hologram plane. The described holographic light transport model is often simplified into a single convolution with a fixed spatially invariant complex kernel, $h(x,y)$ [5].

$$u(x,y) = u_0(x,y) * h(x,y) = \mathscr{F}^{-1}(\mathscr{F}(u_0(x,y))\mathscr{F}(h(x,y))) \tag{3}$$

There are multiple variants of this simplified approach [3,7,8]. In our case, we choose to use the most common form of h described as

$$h(x,y) = \frac{e^{jkz}}{j\lambda z} e^{\frac{jk}{2z}(x^2+y^2)}, \tag{4}$$

where z represents the distance between a hologram plane and a target image plane.

*e-mail: david.walton.13@ucl.ac.uk
†e-mail: kkavakli@ku.edu.tr
‡e-mail: r.kuffnerdosanjos@leeds.ac.uk
§e-mail: d.swapp@ucl.ac.uk
¶e-mail: t.weyrich@cs.ucl.ac.uk
‖e-mail: hurey@ku.edu.tr
**e-mail: a.steed@ucl.ac.uk
††e-mail: t.ritschel@ucl.ac.uk
‡‡e-mail: k.aksit@ucl.ac.uk

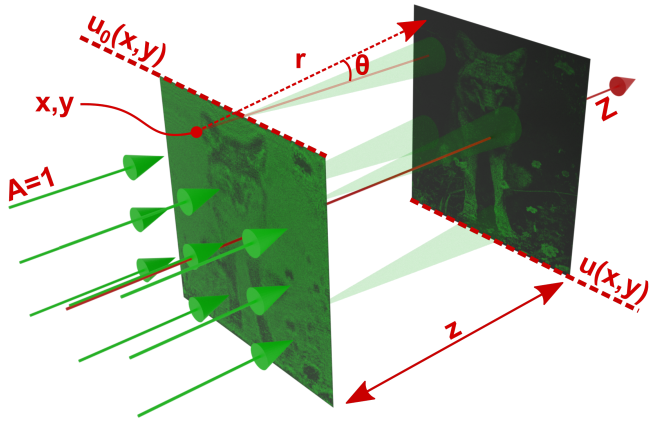

Figure 1: Holographic image reconstruction. A collimated beam with a homogenous amplitude distribution (A=1) illuminates a phase-only hologram $u_0(x,y)$. Light from this hologram diffracts and arrive at an image plane $u(x,y)$ at a distance of z. Diffracted beams from each hologram pixel interfere at the image plane and, finally, reconstruct a target image (Refer to Equation 3 for details).

2 Implementation Details

Here we discuss in detail some aspects of the implementation we were unable to fully cover in the main manuscript.

2.1 Downsampling

Whilst our implementation of the metameric loss function mainly follows the analysis procedure of Walton et al. [6], we found it was important to change the downsampling method used to compute the steerable pyramid. In [6] the steerable pyramid construction is accelerated by replacing the original downsampling approach used in [4] with a more efficient 2x2 averaging approach. This decision appears to have been made in [6] as it allows the steerable pyramid construction to use efficient mipmap generation methods on GPUs for improved efficiency in real-time applications.

Unfortunately, we found that when using this approach, optimised images would contain large blocky artefacts as shown in the upper example in figure 2. As a result we reverted back to the original downsampling approach of [4], which involves applying a lowpass filter with a stride of 2 pixels. As seen in the upper half of figure 2, this eliminated these blocky artefacts.

We hypothesise that this occurs because the simple 2×2 averaging approach is not well-suited to backpropagation. For example, suppose we used a pyramid of depth 5, and a single pixel in the lowpass residual was incorrect. This pixel was originally generated by averaging pixels over a $2^5 \times 2^5 = 32 \times 32$ block of pixels in the input, so when taking a step in the optimisation, the same update will be applied to all pixels in this 32×32 block. Since these blocks do not overlap, they will remain visible in the output, as can be seen in the lower example in figure 2.

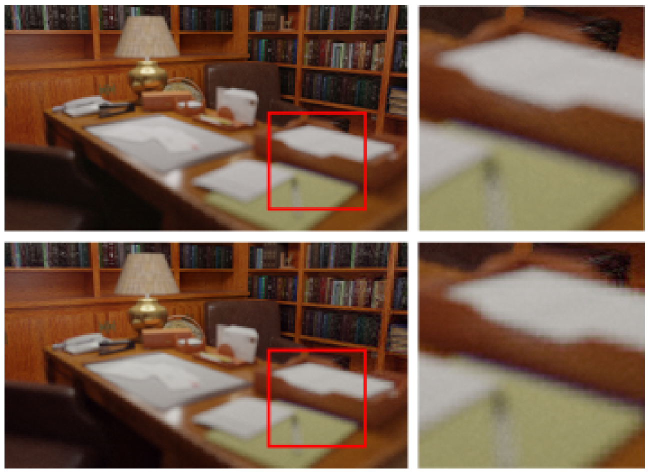

Figure 2: Images optimised to directly match a target image under our metameric loss function. Below: using 2x2 averaging downsampling. Above: using the original downsampling method of [4].

2.2 Foveation

This section will describe the way in which the pooling LoD map is calculated. In the main paper, this is referred to as `calc_lod_map()` in the pseudocode.

We define the pooling regions in a similar way to [1]. A main goal of the pooling approach is to foveate in a consistent way for different image sizes and viewing distances.

For each pixel, we compute a pooling size in radians, based on its eccentricity. The pooling region itself covers all viewing directions with angles less than the pooling size from the pixel. Pooling regions are thus conical in 3D space.

We assume the image will be displayed to the user at a plane a fixed distance away from them. As a result the pooling regions we compute will be intersections of this plane with each pooling cone as visualised in figure 3.

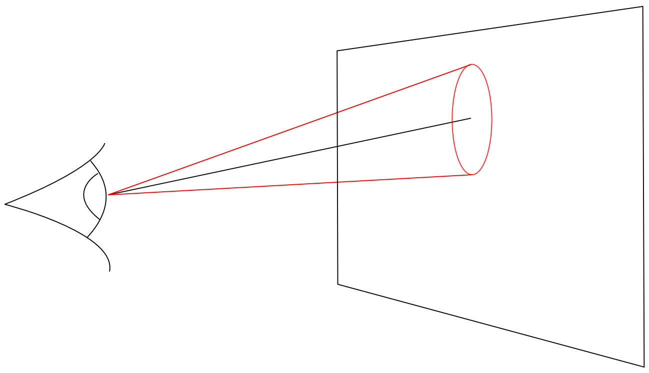

Figure 3: Visualisation of a conical pooling region in red intersecting with a planar image.

We make the following simplifying assumptions:

1. The user has a single viewpoint (we do not consider eyes separately).
2. The viewpoint is coaxial to the image plane (that is, the closest point in the image plane to the viewpoint is at the centre of the image).
3. The image plane is perpendicular to the viewing direction.

We assume that the real distance to the image and the real width and height of the image are known.

In the following we determine the area of the pooling region required for a single pixel p. We define a coordinate system with the viewpoint at the origin, looking along the positive z-axis. We consider the 3 3D points in the image plane shown in figure 4: g, the gaze location, c, the image centre and p the current pixel being considered.

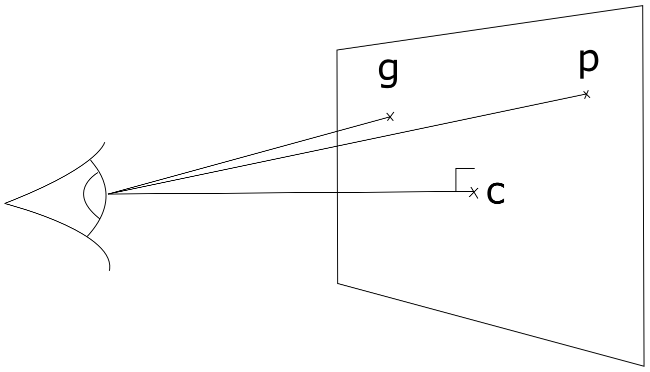

Figure 4: Gaze location g, image center c and current pixel p.

The eccentricity e of the pixel p is simply the angle between p and g, which may be computed using a dot product in the usual way:

$$e = \arccos\left(\frac{p \cdot g}{|p||g|}\right) \tag{5}$$

The desired pooling region angle θ is then computed based on e. Whilst in [1] the pooling size is set to be directly proportional to e, in this we set:

$$\theta := \alpha e^2 \tag{6}$$

The parameter α is an adjustable parameter controlling the aggressiveness of the foveation. We chose this quadratic relationship as it produced better results with the analysis method of [6].

As stated above, the pooling region in 3D may be visualised as a cone extending from the viewpoint. We are interested in the intersection of this cone with the image plane. This intersection is a conic section. In general conic sections can be ellipses, hyperbolae, parabolae or points, but in our case all the intersections will be ellipses.

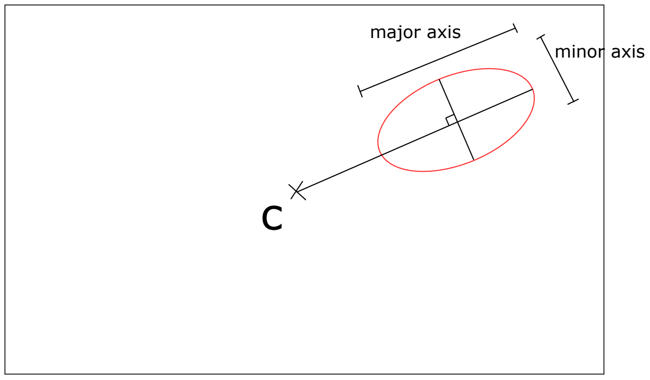

Figure 5: An elliptical pooling region (size exaggerated for readability).

In fact as shown in figure 5 they are each ellipses with major axis aligned radially, along the line connecting p to the centre of the image c, and minor axis perpendicular to this.

We now determine the lengths a and b of the major and minor axis of this ellipse.

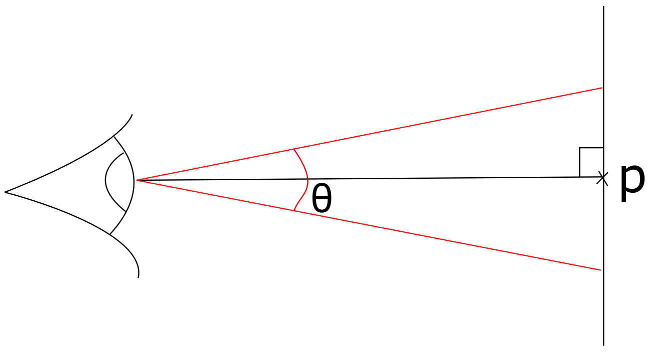

Figure 6: Determining the length of the minor axis of the ellipse.

To determine the length of the minor axis b we consider the triangle connecting the viewpoint with the two ends of the minor axis as shown in figure 6. From this it is clear that the length of the minor axis is as follows:

$$b = 2|p|\tan\left(\frac{\theta}{2}\right) \tag{7}$$

To determine the length of the major axis a we consider the triangle connecting the viewpoint, the centre of the image c and the pixel p shown in figure 7.

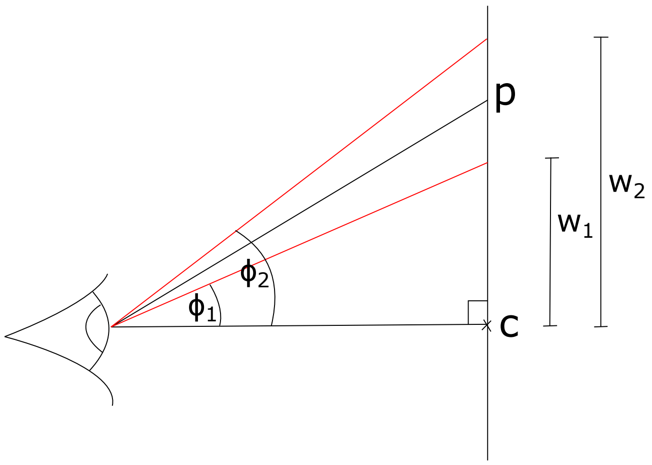

Figure 7: Determining the length of the major axis of the ellipse.

Note that the length of the major axis a is $w_2 - w_1$, so we first determine w_2 and w_1. First we find the angle ε between p and c using a dot product as before in equation 5. This allows us to find the angles $\phi_1 = \varepsilon - \frac{\theta}{2}$ and $\phi_2 = \varepsilon + \frac{\theta}{2}$. Now w_1 and w_2 can be determined:

$$w_1 = |c|\tan(\varepsilon - \frac{\theta}{2}), \quad w_2 = |c|\tan(\varepsilon + \frac{\theta}{2}) \tag{8}$$

Now the major and minor axis lengths a and b are known the area of the pooling region is simply $\frac{\pi ab}{4}$. However, in this work we accelerate pooling by trilinear sampling from mipmaps. As a result our effective pooling regions are square. To achieve the same area, our pooling region should thus have sides of length $s = \sqrt{\frac{\pi ab}{4}}$.

We convert this length into a value in pixels s_p by dividing by the real width of the image, and multiplying by its width in pixels. Finally, we determine the LoD level l of the mipmap we should sample from to approximate filtering with a box filter of size s_p:

$$l = \max(\log_2(s_p + \varepsilon), 0) \tag{9}$$

Here the small added ε avoids taking $\log_2(0)$ at the fovea. Pixels near the fovea may have pooling sizes smaller than a pixel, which would give negative output, so the max ensures no invalid LoD levels below zero are produced.

In practice a map of LoD levels can be precomputed for a gaze location and used for multiple applications of the loss function.

An advantage of computing the pooling region sizes in this way is that for a single value of alpha, the degree of foveation should appear the same for different viewing distances and image sizes, as long as these are supplied when finding the LoD levels.

3 Further Results

Figure 8 shows further simulated hologram examples using our method and MSE loss, both optimising for 200 iterations at a learning rate of 0.08.

Figure 9 shows some examples of holograms optimised using MSE loss and our method displayed on our protoype holographic display, optimised with the same learning rate and iteration count. We note that both examples suffer from typical holographic noise, but the metameric loss is able to achieve better sharpness in the fovea in many cases. These examples use the temporal averaging approach described in the paper. We found that without this approach, both holograms were severely affected by noise and the improvement was harder to see.

Acknowledgments

The authors thank Duygu Ceylan for the fruitful and inspiring discussions improving the outcome of this research, and Selim Ölçer for helping with the fibre alignment of laser light source in the proof-of-concept display prototype. This work was funded by the EPSRC/UKRI project EP/T01346X/1.

References

[1] J. Freeman and E. P. Simoncelli. Metamers of the ventral stream. *Nature neuroscience*, 14(9):1195–1201, 2011.

[2] J. C. Heurtley. Scalar rayleigh–sommerfeld and kirchhoff diffraction integrals: a comparison of exact evaluations for axial points. *JOSA*, 63(8):1003–1008, 1973.

[3] K. Matsushima and T. Shimobaba. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Optics express*, 17(22):19662–19673, 2009.

[4] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proceedings., International Conference on Image Processing*, vol. 3, pp. 444–447. IEEE, 1995.

[5] M. Sypek. Light propagation in the fresnel region. new numerical approach. *Optics communications*, 116(1-3):43–48, 1995.

[6] D. R. Walton, R. K. D. Anjos, S. Friston, D. Swapp, K. Akşit, A. Steed, and T. Ritschel. Beyond blur: Ventral metamers for foveated rendering. *ACM Trans. Graph. (Proc. SIGGRAPH 2021)*, 40(4), 2021.

[7] W. Zhang, H. Zhang, and G. Jin. Adaptive-sampling angular spectrum method with full utilization of space-bandwidth product. *Optics Letters*, 45(16):4416–4419, 2020.

[8] W. Zhang, H. Zhang, and G. Jin. Band-extended angular spectrum method for accurate diffraction calculation in a wide propagation range. *Optics letters*, 45(6):1543–1546, 2020.

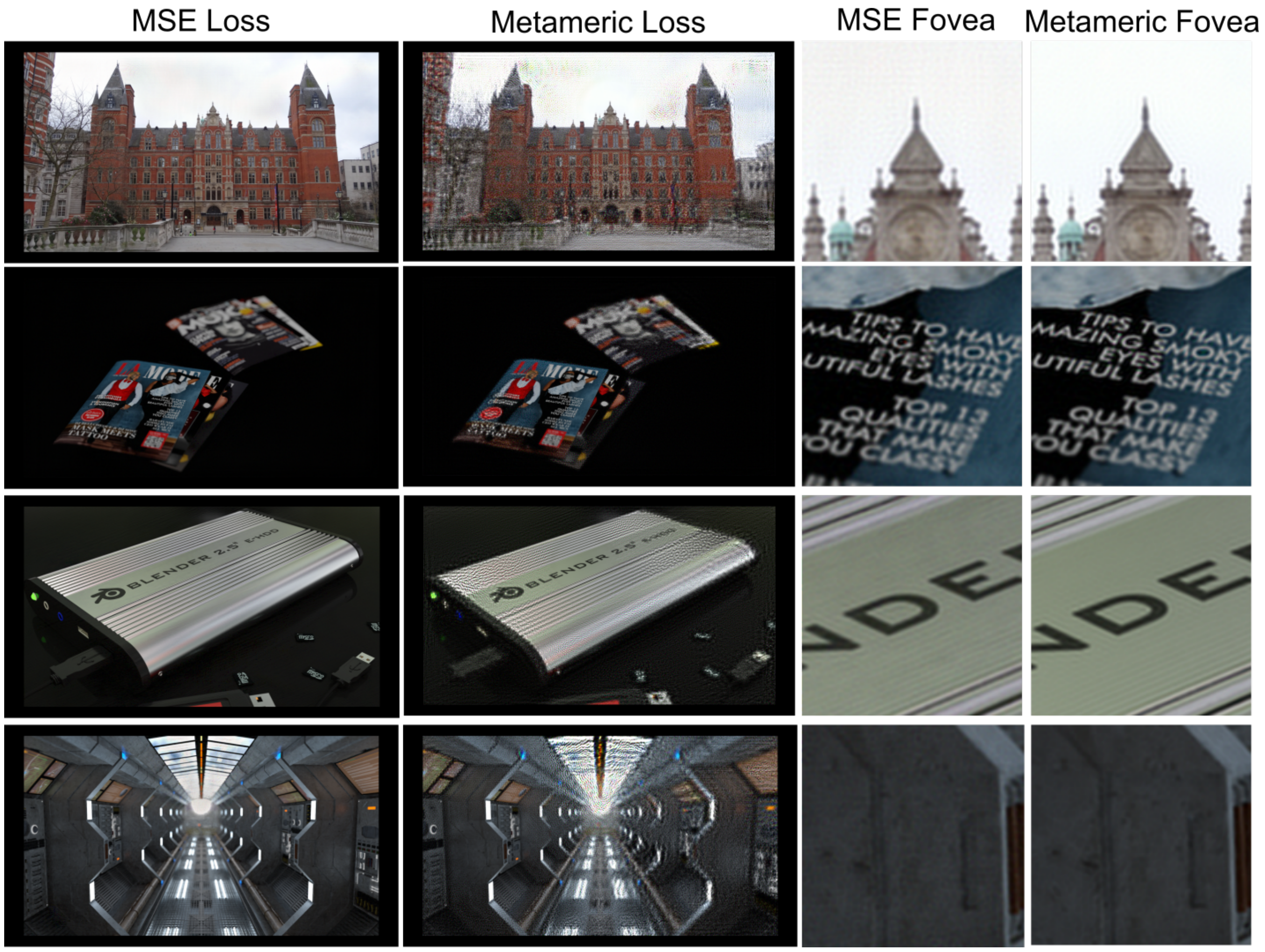

Figure 8: Additional simulated holographic results optimising using MSE loss and our method. Columns on the right show close-ups of the foveal region of the image for the intended gaze location.

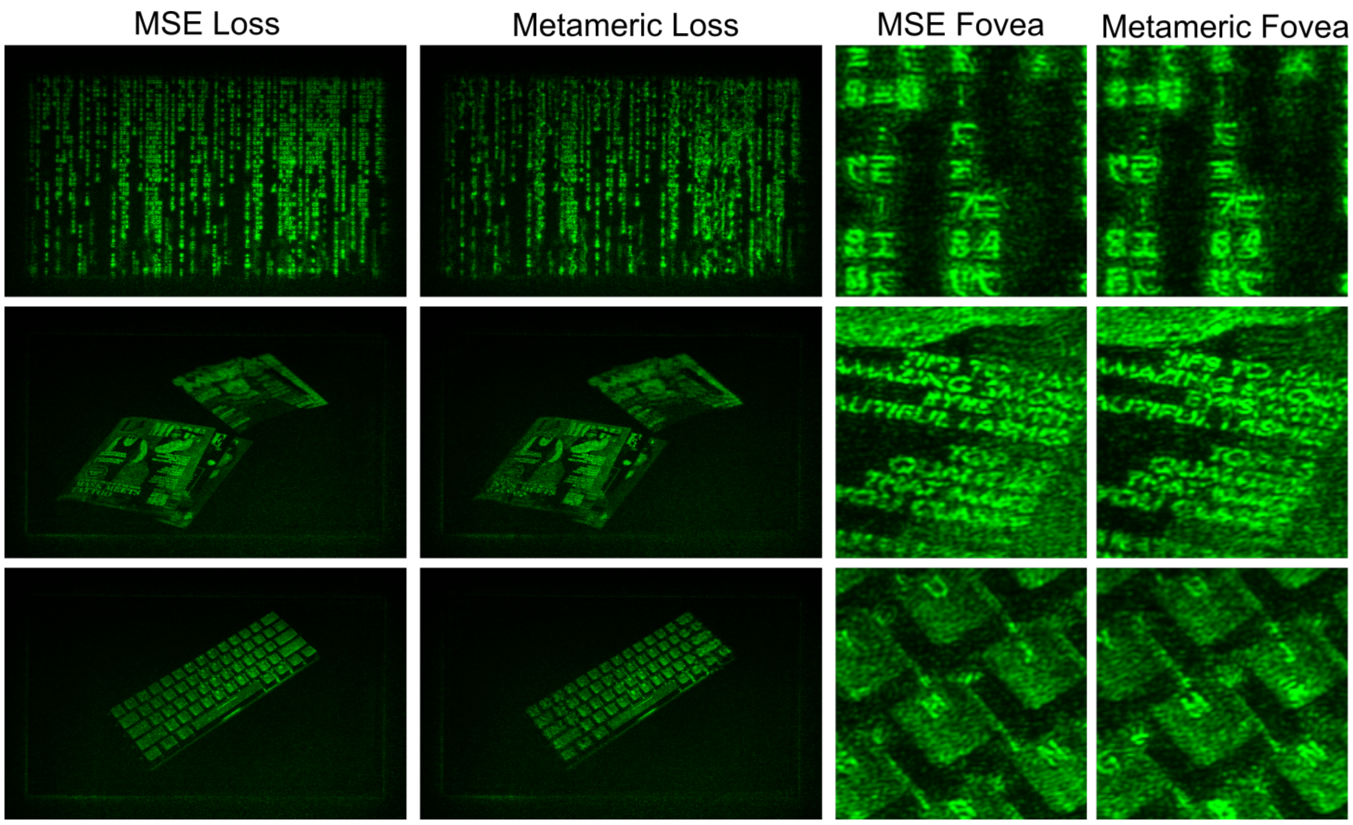

Figure 9: A comparison between holograms generated using MSE loss and our method, shown on our prototype holographic display using temporal averaging. Columns on the right show close-ups of the foveal region of the image for the intended gaze location.