

Supplemental Material for Holospeed: High-Speed Holographic Displays for Dynamic Content



References to the main paper are denoted in **red**. In Section 1, we briefly provide details on our visualization website. Section 2 summarizes the additional results shown in this document. In Section 3, we add more discussion of our stochastic and kernel approaches for motion-aware optimization. In Section 4, we add more detail on our sequential approach for on-the-fly motion-aware high-speed displays. In Section 5, we describe a preblurring approach for mitigating stroboscopic artifacts. In Section 6, we describe our holographic system model, and how we calibrate it. In Section 7, we discuss how we use optical flow calculations to build a distribution of eye motion. Finally, in Section 8 we provide details on the 3D models used in this work.

1 VISUALIZATION WEBSITE

In `site_vis`, we share a local website for visualizing slow motion videos of perceived content from our frameworks. Please open `site_vis/index.html` in your browser with Javascript enabled.

2 MORE COMPARISONS

In Figure 1, we show a comparison of the sequential approach described in Section 4.2 with a direct optimization of Equation 13. In our results, performance is similar between them, justifying our use of a sequential approach. We compare them further in Section 4.

In Figure 4, we evaluate our approach on a scene with more complex motion trajectories, where a leaping tiger and a landing bird provide more complicated object dynamics. Our proposed motion-aware modeling and high-speed regularization together provide the best results again.

In Figure 5, we demonstrate our framework applied to a binary SLM with the same resolution and framerate as our original SLM, along with a single sideband mask. Using high-speed regularization outperforms not using high-speed regularization by a larger factor than on a higher bit-depth SLM — we attribute this to challenges in convergence with the highly-quantized binary SLM.

In Figure 6, we evaluate our proposed methodologies over a spectrum of different SLM framerates. Motion artifacts continue to appear at lower framerates, which are mitigated by our proposed motion-aware approaches with high-speed regularization.

In Figures 8 and 9, we show full versions of Figures 6 and 7 from the main paper, respectively.

3 MORE DISCUSSION ON STOCHASTIC VS KERNEL METHODS

Aside the improvement under low-compute settings (Figure 9), there are certain computational arguments for the kernel-based approach. For one, the stochastic approach requires recomputing the perceived target for every iteration, as the selected motion is different per iteration. In contrast, the “perceived target” under the kernel-based approach can be precomputed a single time and used for the entire optimization. A very similar notion also yields benefits in the sequential setting. Under the stochastic approach, the perceptual contribution from previous SLM patterns must be recomputed for every potential motion and iteration. Conversely, under the kernel approach, the previous perceptual contribution can be simply computed once and reused.

On the other hand, the kernel-based approach requires convolution operations, adding overhead over the simple shift-and-add required by the stochastic approach. One potential speed-up is to approximate the kernel with a separable version, at the cost of some visual quality — instead of performing a full 3D convolution, perform the 2D spatial convolutions as separate 1D convolutions in x and y , and then aggregate in time. If scene motion is mostly along a single axis such as a cervical rotation of the head in AR/VR, such a simplification may be more exact — see Figure 2.

4 MORE DISCUSSION FOR ON-THE-FLY MOTION-AWARE HIGH-SPEED DISPLAYS

In Algorithm 1, we provide pseudocode for the sequential algorithm for stochastic high-speed displays. If eye tracking information is available, a history of previous eye motions would also need to be kept. Note that only information from the last duration of persistence-of-vision $N_{\text{POV}}\tau_{\text{SLM}}$ needs to be maintained, imposing an upper bound on potential additional memory usage.

We also note that our sequential approach theoretically results in less overall computational expense than a full solve of Equation 13. To put it briefly, our method only considers



Fig. 1. **Comparing sequential solve with full solution (simulation).** Performance is for the most part similar between our sequential methodology given in Section 4.2 and a full joint optimization over all to-be-displayed frames.

one perceived image per SLM pattern like a traditional time-multiplexed display, while a full solve would consider all perceived images that depend on that pattern. Given that performance between them is similar as shown in Figure 1, our sequential algorithm may be a reasonable choice even outside the on-the-fly context.

Algorithm 1 Sequential high-speed display

Require: $N_{\text{SLM}} > 0 \rightarrow$ Maximum number of frames

Require: $\text{getFrame}[i] \rightarrow$ Returns scene at time $i\tau_{\text{SLM}}$

Require: $\text{Display}(\phi[i]) \rightarrow$ Displays SLM frame $\phi[i]$

$i \leftarrow 0$

$V_{\text{HS}} \leftarrow []$

$\phi_{\text{hist.}} \leftarrow []$

while $i\tau_{\text{SLM}} < \tau_{\text{duration}}$ **do**

$V_{\text{HS}}.\text{append}(\text{getFrame}[i])$

$\phi[i] \leftarrow \text{argmin}_{\phi[i]} \mathcal{L}_{\text{seq.}}(\phi[i], \phi_{\text{hist.}}, V_{\text{HS}})$

$\text{Display}(\phi[i])$

$\phi_{\text{hist.}}.\text{append}(\phi[i])$

$i \leftarrow i + 1$

end while

5 PREBLURRING CONTENT

Instead of using high-speed regularization, an alternative solution for remedying stroboscopic effects is preblurring target content according to expected object and eye motion. Mathematically, such an approach can be seen as replacing I_{target} in Equation 16 with $\mathcal{P}\{V_{\text{HS}}, \mathbf{m}\}(\tau)$. We visualize

results of such an approach for our stochastic approach in Figure 7. Using this technique produces a slight increase in contrast relative to high-speed regularization, and most stroboscopic effects are avoided. However, thanks to the underlying low rate of regularization, various visual inaccuracies can still be produced. For one, at timestamps between regularized ones, more motion blur than necessary will be produced, as part of the frames will replicate the contribution of future content due to Equation 10. Furthermore, since the output perceived content is unregularized at these timestamps, we found that unwanted behavior can manifest, like the spurious high frequencies in the motion-blurred plane windows in Figure 7(b). Finally, since motion-blur is in essence artificially produced, this approach can also be more sensitive to error or latency in eye motion estimation, relative to high-speed regularization [1].

6 HOLOGRAPHIC SYSTEM MODEL

In Figure 3, we show a system diagram for our prototype holographic display. As described in the main paper, we use a 4F system with a mask at the Fourier plane to block the DC component of the laser, and a higher-order filter to block any diffraction modes.

To model this system for our framework, we start by using a multi-layer perceptron network $\mathcal{N}(\phi)$ to model the SLM's phase curve, taking in an input phase value and outputting a corrected one. Next, we model any non-idealities in the incident laser illumination as well as any SLM non-uniformities with a complex modulation $\mathcal{M}_{\text{source}}$.

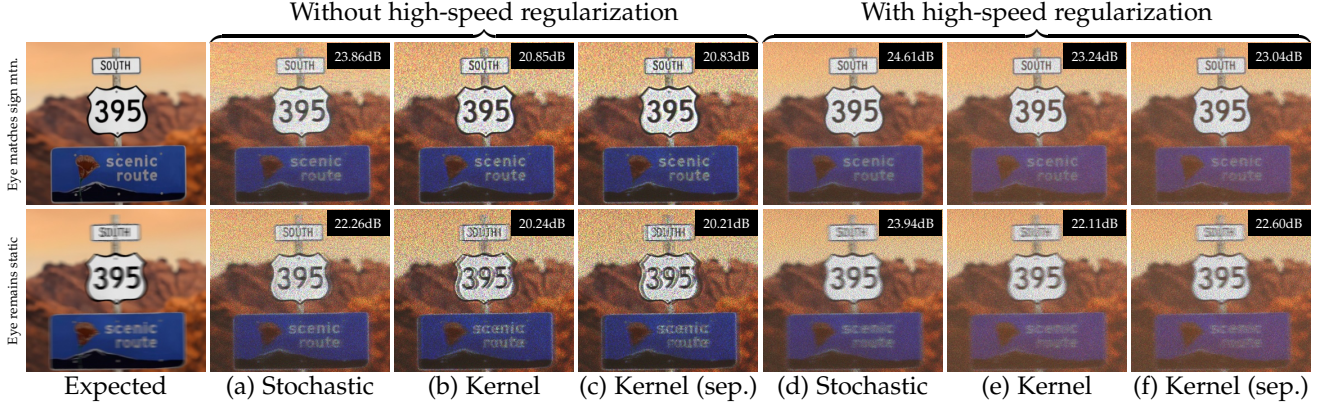


Fig. 2. **A separable kernel (simulation).** For certain scenes where motion happens primarily along a single axis, a separable approximation may be sufficient for our kernel approach. Here, we show an example with world-locked content, where objects predominantly move in the same direction as the user moves. The separable version (c), (f) produces very similar solutions to the original (b), (e). Here, motion occurs predominantly along the x-axis, so we compute our 1D filters by summing the estimated flow distribution along x and y . Other axes could be handled by rotating the images, at some compute cost.

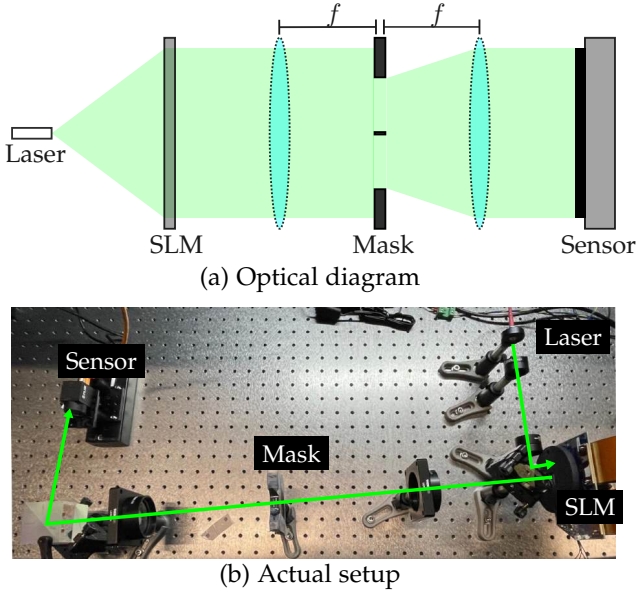


Fig. 3. **System diagram for our holographic display.** Beyond a baseline holographic display architecture, we include an additional 4F system with a DC block and higher-order filter, with appropriate beamsplitters to help route light.

After, we apply a spatially varying convolution $\mathcal{C}(\cdot)$ to handle any generic optical non-idealities, such as an imperfect 4F system. We then use an angular-spectrum (ASM) propagation [2] to propagate the wavefront from the SLM to the sensor plane $\mathcal{P}_{\text{ASM}}(\cdot)$. We also add a learnable convolution $\mathcal{M}_{\text{aper}}$ to represent the higher-order filter and DC block. To model any undiffracted laser light, we add an additional complex term $\mathcal{A}_{\text{undiff}}$, as well as $\mathcal{I}_{\text{ambient}}$ for any ambient incoherent light. We add an additional non-negative real mask $\mathcal{M}_{\text{sensor}}$ to model any vignetting on the camera. In whole, our model can be expressed mathematically as follows:

$$\mathcal{I}_{\text{model}} = \mathcal{M}_{\text{sensor}} \cdot (|\mathcal{M}_{\text{aper}} * \mathcal{P}_{\text{ASM}}(\mathcal{C}(\mathcal{M}_{\text{source}} \cdot \mathcal{N}(\phi))) + \mathcal{A}_{\text{undiff}}|^2 + \mathcal{I}_{\text{ambient}}) \quad (1)$$

In practice, $\mathcal{C}(\cdot)$ is implemented using regions that overlap by 20%, and small 5×5 pixel kernels. To avoid local minima, $\mathcal{M}_{\text{source}}$, $\mathcal{M}_{\text{aper}}$, $\mathcal{A}_{\text{undiff}}$ and $\mathcal{I}_{\text{ambient}}$ are all represented with

lower resolution maps that have 0.25 times the resolution of the SLM, and are upsampled using bilinear interpolation as needed.

To calibrate these parameters, we calculated phase patterns that produce images from a natural image dataset using a baseline version of the model with just ASM propagation. We then displayed these patterns to capture pairs of phase patterns and output images. We then ran Adam [3] over this dataset to optimize all of the above parameters. To properly map captured images to simulated outputs, we also capture sequences of Gray codes to estimate an initial homography, on top of which we apply differentiable thin-plate splines [4], [5] for more accurate warping.

In practice, fast SLMs have reduced bit depth, which can potentially cause reduced quality if not properly accounted for during phase retrieval [6], [7]. To tackle this, we adopt an approach where the forward pass is computed with quantization while backwards gradients are computed assuming a differentiable version — we use the soft rounding function from Agustsson *et al.* [8], which uses a hyperparameter α to control the strength of the rounding. We gradually increase α over the phase retrieval process.

7 ESTIMATING AN EYE MOTION DISTRIBUTION USING OPTICAL FLOW

To avoid eye tracking as described in Section 4.1, optical flow can be utilized to estimate a distribution of eye motion, assuming that the eye will be tracking a scene object. In our work, for simplicity we use Farneback optical flow to estimate scene motion [9], but any modern flow estimation algorithm, *e.g.* RAFT [10] could be used.

Without high-speed regularization, we compute optical flow between our current and previous low-speed target. Then, we have a flow vector for every pixel in the image — we build a 2D histogram of these flow vectors. We then use this histogram for the probability distribution described in Section 4.1. For simplicity, we discretize our estimated flow to pixel units.

With high-speed regularization, we look at the previous N_{POV} target frames, and estimate optical flow between every sequential frame. We build a 2D histogram per pair,

which we then average to estimate an overall eye motion distribution. We then proceed as in the previous case.

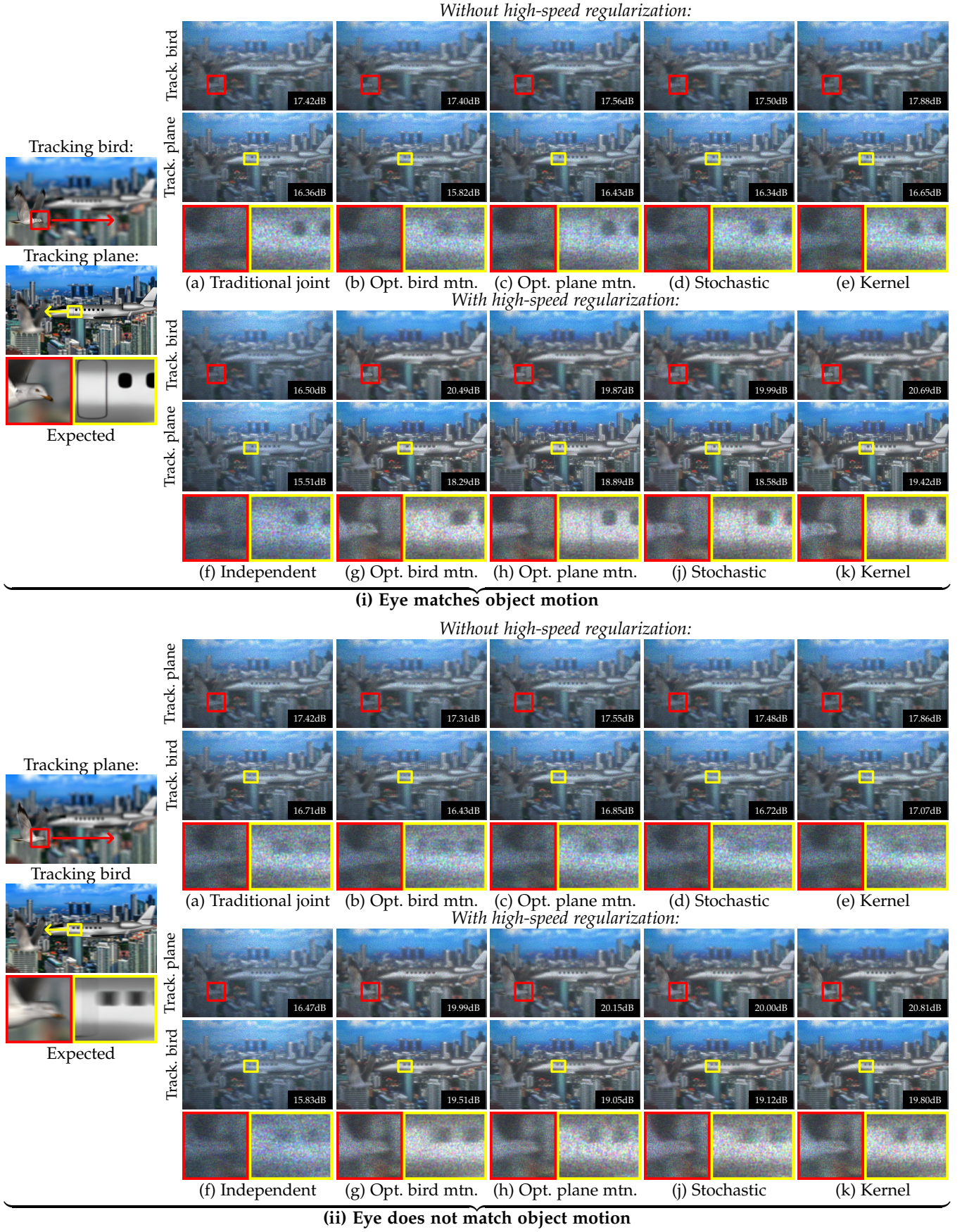
8 SCENE MODEL SOURCES

- Highway scene:
 - Road sign courtesy of KaramellGlass on SketchFab. Licensed under CC BY 4.0.
- Basketball scene:
 - Basketball courtesy of afurokn on SketchFab. Licensed under CC BY 4.0.
 - Basketball hoop courtesy of paethon on SketchFab. Licensed under CC BY 4.0.
 - Arm courtesy of Just8 on SketchFab. Licensed under CC BY 4.0.
- Skyline scene:
 - Bird courtesy of Oregon State University on SketchFab. Licensed under CC BY 4.0.
 - Plane courtesy of nikdoo on cgtrader. Licensed under cgtrader’s Royalty Free No AI License.
- Jungle scene:
 - Tiger courtesy of Amil (francescolima74) on SketchFab. Licensed under CC BY 4.0.
 - Bird courtesy of mr.film on SketchFab. Licensed under CC BY 4.0.

REFERENCES

- [1] M. Rejhon, “The stroboscopic effect of finite frame rate displays,” <https://blurbusters.com/the-stroboscopic-effect-of-finite-framerate-displays/>, 2019.
- [2] J. W. Goodman, *Introduction to Fourier Optics*, ser. McGraw-Hill physical and quantum electronics series. W. H. Freeman, 2005.
- [3] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference for Learning Representations*, 2015.
- [4] J. Duchon, “Splines minimizing rotation-invariant semi-norms in sobolev spaces,” in *Constructive Theory of Functions of Several Variables: Proceedings of a Conference Held at Oberwolfach April 25–May 1, 1976*. Springer, 1977, pp. 85–100.
- [5] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, “Kornia: an open source differentiable computer vision library for pytorch,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3674–3683.
- [6] S. Choi, M. Gopakumar, Y. Peng, J. Kim, M. O’Toole, and G. Wetzstein, “Time-multiplexed neural holography: a flexible framework for holographic near-eye displays with fast heavily-quantized spatial light modulators,” in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9.
- [7] B. Lee, D. Kim, S. Lee, C. Chen, and B. Lee, “High-contrast, speckle-free, true 3d holography via binary cgh optimization,” *Scientific Reports*, vol. 12, no. 1, p. 2811, 2022.
- [8] E. Agustsson and L. Theis, “Universally quantized neural compression,” *Advances in neural information processing systems*, vol. 33, pp. 12 367–12 376, 2020.
- [9] G. Farneback, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis: 13th Scandinavian Conference*. Springer, 2003, pp. 363–370.
- [10] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.





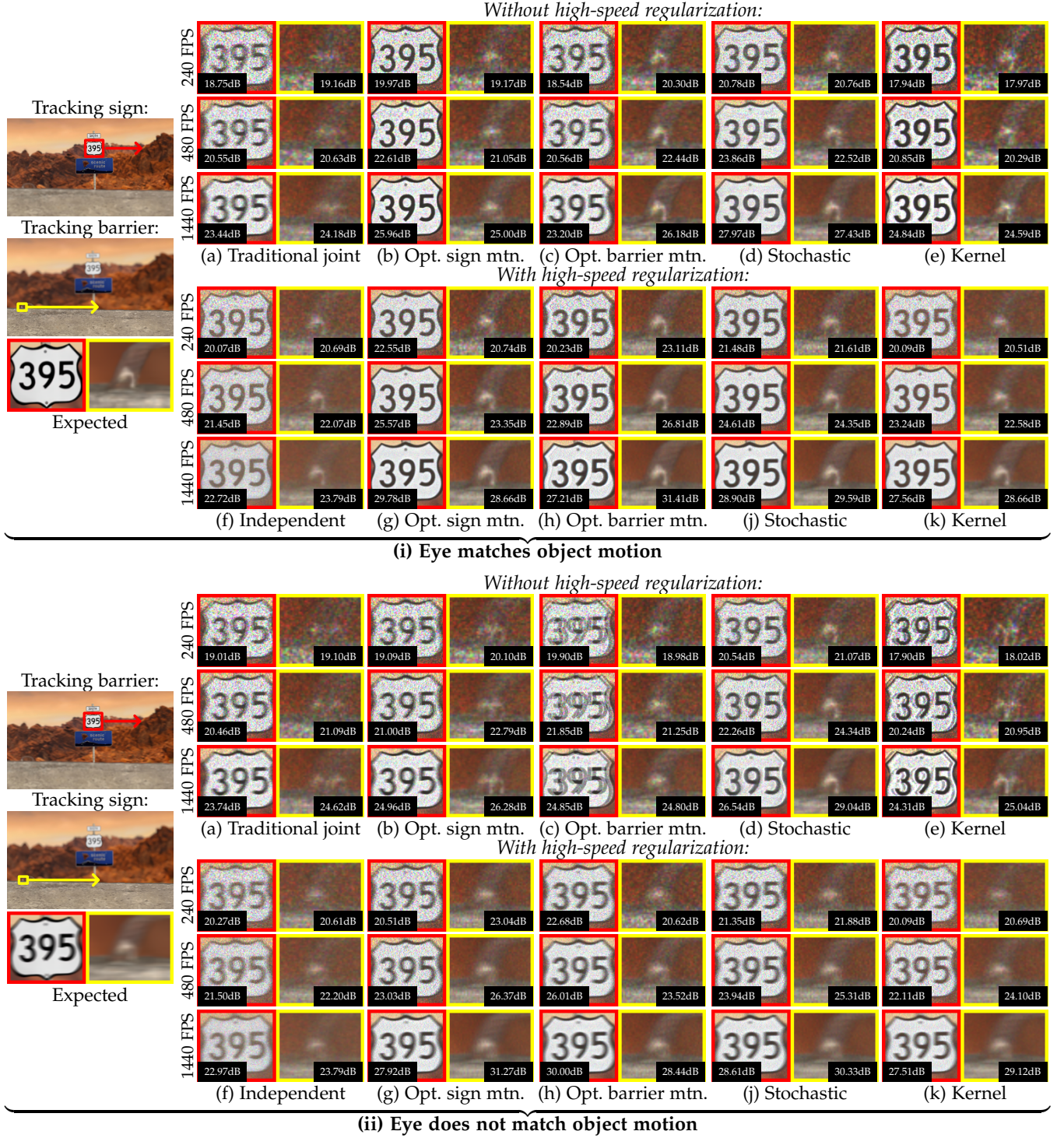


Fig. 6. **Varying framerate of the SLM (simulation).** We validate the effects of eye motion and dynamic motion for different SLM framerates. Using a lower framerate results in less time multiplexing and more visible speckle, but still suffers from the same motion artifacts as higher framerates unless a motion-aware approach is applied with high-speed regularization.

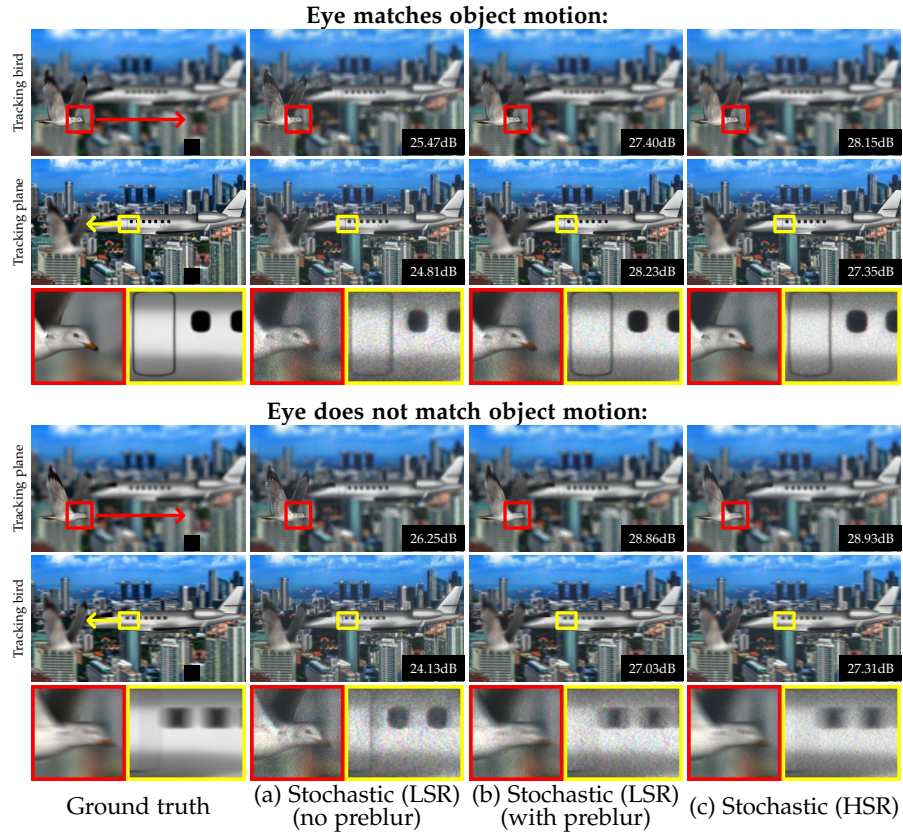


Fig. 7. **Preblurring (simulation).** We test out a preblurring approach for stroboscopic minimization with our stochastic motion-aware approach. While stroboscopic artifacts are minimized, visual inaccuracies can still be produced as shown by the high frequencies in the plane windows in **(b)**, thanks to the underlying low rate of regularization.

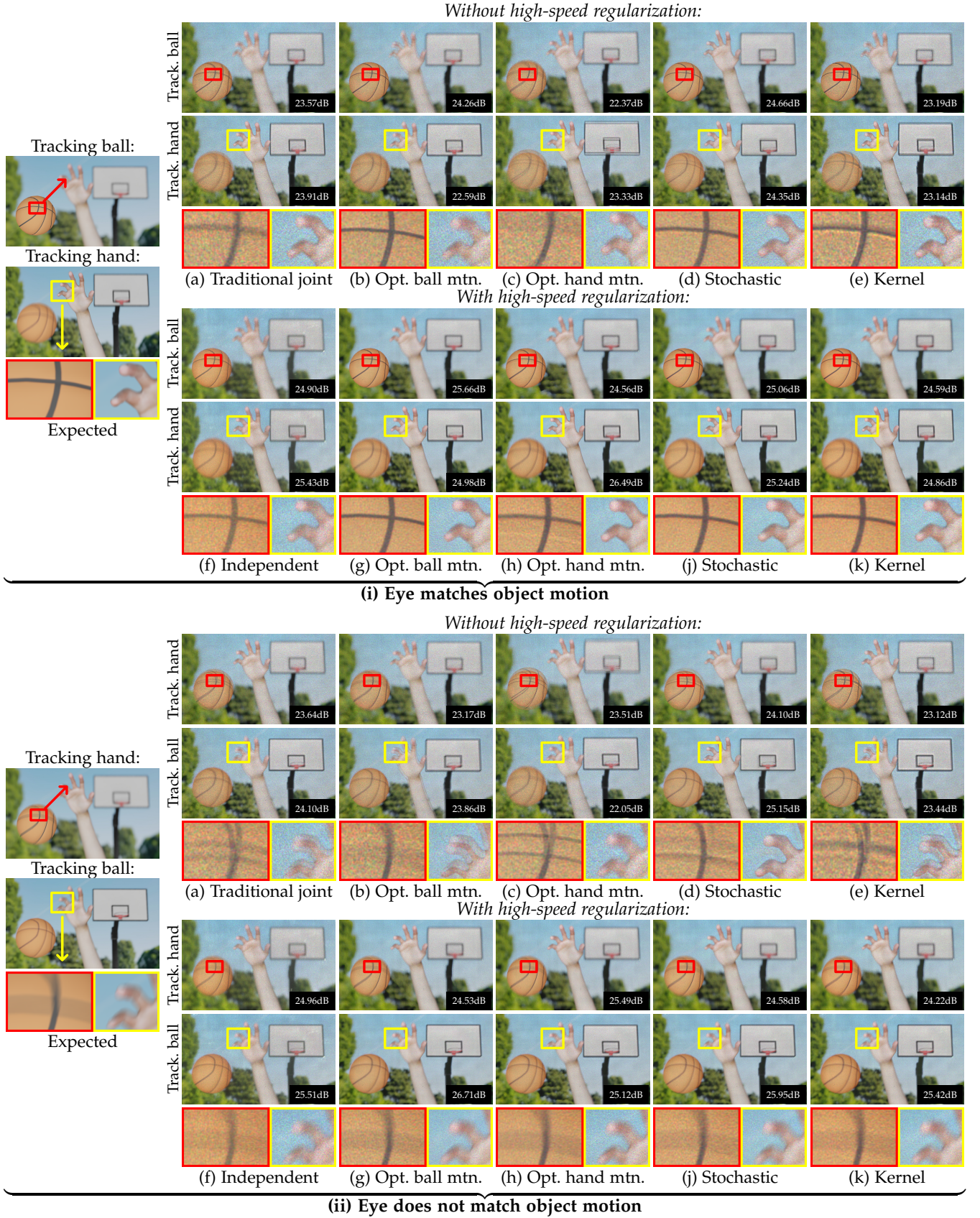


Fig. 8. **Results in a real system (full)**. Traditional time-multiplexing (a) fails to reproduce sharp images when the eye tracks a moving object, or realistic blur when the eye doesn't. Independent high-speed display (f) lacks these artifacts, but suffers from decreased contrast. Our motion-aware optimization can remedy these effects, but without high-speed regularization results in (ii) strobing artifacts (b), (c), (d), (e). High-speed regularization and our motion-aware optimization combined produces the best results (g), (h), (i), (k). Our stochastic and kernel approaches mitigate uncertainty in potential eye motion (j), (k).



Fig. 9. **Low compute visualization (simulation, full).** We test our approach with just 4 iterations of gradient descent instead of 500. Our proposed framework is again able to mitigate sample-and-hold blur as well as strobing effects. Unlike higher-compute settings, the kernel-based approach slightly outperforms the stochastic approach, as insufficient iterations are performed to cover the entire space of possible motions under the stochastic methodology.