Imaging, Computing, and Human Perception: Three Agents to Usher in the Autonomous Machine Computing Era

Yuhao Zhu
yzhu@rochester.edu
University of Rochester
Rochester, NY, USA

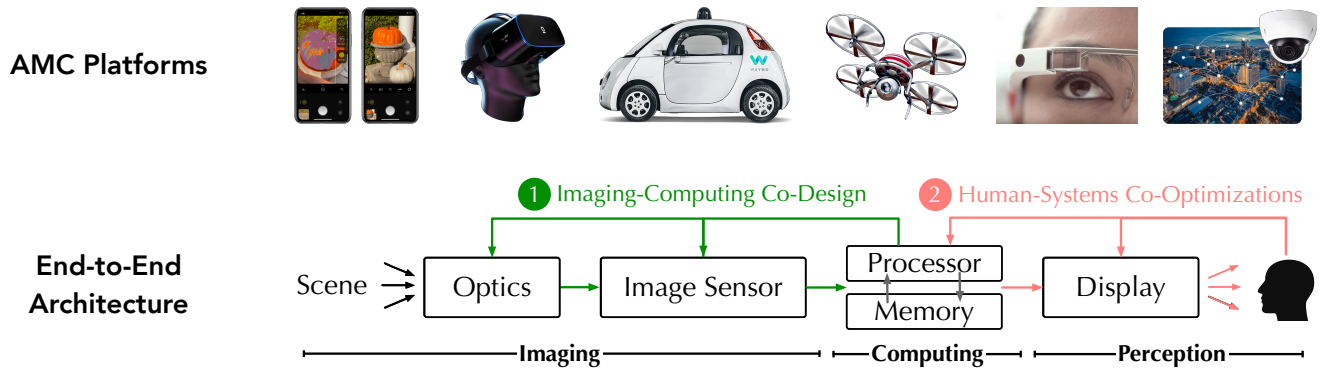

Figure 1: Imaging system, computer systems, and human visual systems are three necessary components in end-to-end AMC platforms. Jointly designing and optimizing them is crucial for maximizing overall efficiency and task performance.

Abstract

Autonomous machines such as drones, robots, and even Augmented and Virtual Reality headsets, while are of a computing nature, intimately interact with both the environment and humans. They must be built, from the ground up, with principled considerations of three main components: imaging system, computer system, and human perception and cognition. While our community has been, for very good reasons, focused almost exclusively on improving the computer systems, our position is that the continued progress in autonomous machine computing (AMC) must rely on co-designing and co-optimizing all three components. We call them, collectively, agents for progress in AMC.

This paper has three goals: discuss how the three agents play their (largely isolated) roles in today's AMC, describe a framework where the three agents are fundamentally connected, and finally, present exciting research opportunities that arise when jointly designing and optimizing across the three agents.

CCS Concepts

• **Computer systems organization** → **Heterogeneous (hybrid) systems**; **Optical computing**; **Special purpose systems**; • **Computing methodologies** → **Mixed / augmented reality**; **Perception**; **Virtual reality**.

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10
https://doi.org/10.1145/3676536.3697113

Keywords

Imaging, Graphics, Optics, Image Signal Processing, Human Visual System, Encoding, Decoding

ACM Reference Format:
Yuhao Zhu. 2024. Imaging, Computing, and Human Perception: Three Agents to Usher in the Autonomous Machine Computing Era. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3676536.3697113

1 Introduction

Autonomous Machine Computing necessarily encompasses three components: imaging systems, computer systems, and the human visual system. Imaging systems capture massive visual data, temporally and spatially. Computer systems interpret the visual data (e.g., performing machine vision tasks) and, in many cases, generate visual data for humans to consume. The human visual interpret both physical visual data from the scene and synthesized visual data generated by the computer systems. Figure 1 shows end-to-end pipeline architecture.

The goal of this opinion piece is to discuss opportunities that arise when we jointly design and optimize all three components. These opportunities come in two forms: co-designing imaging and computer systems (Section 2) and co-optimizing computer systems with human visual system (Section 3). We will conclude with an outlook for exciting research directions (Section 4).

2 Imaging-Computing Co-Design

Imaging and computing, which acquire and interpret visual data, respectively, are traditionally designed in isolation and simply

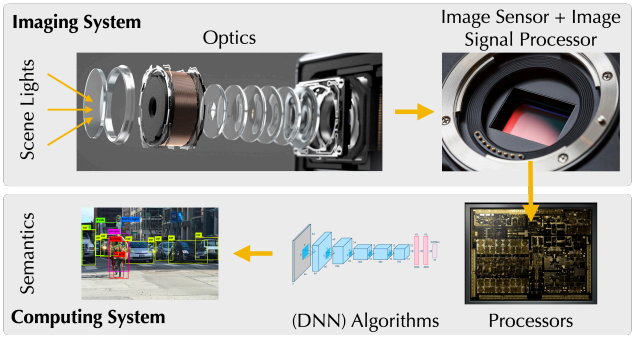

Fig. 2: A typical machine vision pipeline that encompasses both the imaging system and the computing system.

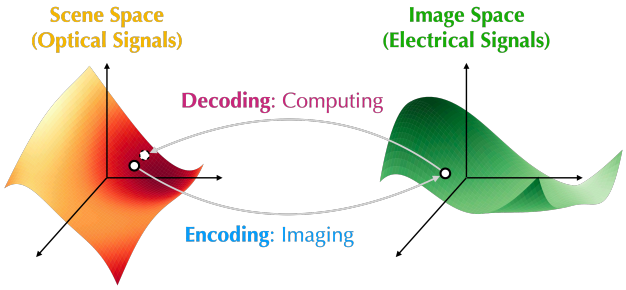

Fig. 3: The imaging system is an encoder that encodes the high-dimensional scene space into a low-dimensional image space; the computing system decodes the scene information back from the low-dimensional space.

stitched together in a system, resulting in a sub-optimal whole. Figure 2 shows how imaging and computing interacts in a typical machine vision pipeline. This section discusses some recent developments in co-designing imaging and computing systems to deliver orders of magnitude efficiency gains or task quality improvements.

Fundamentally, the imaging system is an encoder, and the computer system is a decoder. This is the idea illustrated in Figure 3. The physical world is a very high-dimensional space: there are many points in the scene; each point has many rays existing it, each of which carries spectral radiance, polarization, and phase information. In order to make sense of the scene, an imaging system encodes the high-dimensional information into a low-dimensional space, i.e., the image pixel space. The computer system, i.e., the machine vision algorithms executing on a piece of hardware substrate, operates on the low-dimensional information with the goal of reconstructing information (e.g., object detection) in the high-dimensional scene space. Once taking this information-theoretical perspective, it only makes sense to jointly design the encoder with the decoder.

The imaging system has three main components: optics, image sensor, and the Image Signal Processor (ISP). The optics, e.g., lenses, serve to focus lights on the sensor. The sensor turn optical signals, i.e., photons, to electrical signals, i.e., RAW pixels. The RAW image pixels are transferred to the host through the MIPI CSI-2 interface to be processed by an Image Signal Processor (ISP), which removes sensing artifacts and generates clean pixels either for machines to consume (i.e., computer vision) or for humans to consume (i.e., photography). We will discuss how each component can be co-designed with computer systems.

2.1 Deep Optics

Lenses focus lights onto the sensor, but introduce aberrations. Conventional lens design aims to maximize the imaging quality by minimizing aberration. Recent research focuses on co-designing optics with the downstream algorithms, both for better imaging quality or for improving the end-to-end system efficiency. Both are research directions share the same common strategy: parameterizing optical design using a differentiable model and co-train optics with downstream DNNs, hence the notion "deep optics".

2.1.1 Free-Space Optical Computing. While lenses are conventionally thought of as an imaging device, it can be used for computation. We can computationally model an optical system as a linear system that performs a convolution against the optical image incident on the sensor plane. An optical image $I(x, y)$ represents the irradiance at position $<x, y>$ on the sensor plane. The sensor then samples the convolved optical image at the pixel sites. The convolution kernel is called the Point Spread Function (PSF), which is uniquely dictated by the inherent optical parameters of the lenses and is dependent on both the depth of the scene points and the wavelength.

The fact that lenses perform convolution has inspired a whole host of work to offload convolution operations in a CNN to the optical domain, which, admittedly, is not a new idea [80]. For instance, PSF engineering, i.e., tuning the system PSF to achieve a particular convolution, has been a field in optics for decades [40]. The recent interests arise because, partially, we can model the design of the lenses in a differentiable way so that we can co-train the lenses with the downstream CNNs. For instance, Chang et al. [12] uses Diffractive Optical Elements (DOEs) as the optical device to implement convolution; Villegas Burgos et al. [10, 77] uses metasurface optics. In both cases, the optical parameters (height map in DOE and token orientation in metasurface) are co-trained with downstream CNNs.

While using lens for optical convolution is inefficient, it is generally challenging to implement non-linear operations in free-space optics. Therefore, one generally has to converts signals from the optical domain back to the electrical domain after each convolutional layer [47]. The per-layer signal transduction might hurt the overall system efficiency; as a result, using free-space optical computing for performance improvements might be applicable only to limited scenarios where the CNN is lean to begin with.

2.1.2 Domain-Specific Deep Optics. Orthogonal to improving performance is improving the quality of domain-specific tasks, many of which are critical to AMC such as depth estimation [13, 30, 33, 82], acquisition of high-dynamic-range images [54, 70], obtaining large depth-of-field [50], object detection [13], and extended depth of field [64]. The idea is to co-train the optical design with downstream computer vision algorithms to optimize for vision task loss. Such co-design usually applies additional manufacturing constraints to ensure that the optimized optical design is fabricable.

Fundamentally, these optical designs improve task quality because it learns what information in the incident light is essential to retain for a specific task. Imaging through lenses is necessarily a lossy encoding. For instance, the phase and polarization information in the scene lights is lost through this encoding. The diffraction limit and optical aberrations further degrade the signal. Reconstructing the original scene signal from the degraded (and sampled) signal is an ill-posed problem that does not have a general solution. Deep optics addresses this issue by learning from the large amount of training data, optimizing for a specific task domain.

The limitation of domain-specific optics is the lack of flexibility: the lenses, once fabricated, are fixed; they are optimized for a particular class of tasks and are usually sub-optimal for others.

2.2 Computational Image Sensors

While conventional CMOS image sensors are responsible for only "imaging", i.e., generating pixels from scene light, modern image sensors do much more: they run deep neural networks (DNNs) and buffer a large amount of data (at the order of Gb) — all in the same die! What's fueling the ever more capable image sensors is die stacking, a technology that is perhaps more commonly seen in processor design but is virtually everywhere in high-end image sensors today.

2.2.1 Stacked Image Sensors. Conceptually, an image sensor has two basic components: a light-sensitive pixel array that converts photons to electric charges and the read-out logic that converts charges to digital values (RAW pixels). Traditional CMOS image sensors lay the pixel array and the logic circuitry side by side. Virtually all image sensors today, however, stack the pixel array layer on top of the logic layer. Some of the early and classic examples are the SONY IMX240 sensor used in Galaxy S6 smartphones and the Samsung S5K2L2 sensor used in Galaxy S9 smartphones.

The usual advantages of die stacking, such as providing higher bandwidth and allowing for heterogeneous integration (i.e., the pixel layer and the logic layer can use their respective, optimal process node), still apply. For image sensors, however, perhaps the biggest benefit that stacking offers is the smaller form factor or, equivalently and more commonly, the ability to integrate more functionalities, mostly into the logic layer, given the same footprint.

Indeed, we are seeing a plethora of image sensors with ever more advanced processing capabilities. These pixels are generally classified into two categories, each with a complementary uses of the extra space available in the stacked design. The first class of sensors improve the fundamental imaging quality through advanced pixel circuitries [35, 48, 76]. One could argue that those circuitries are better to be placed inside a sensor anyways but had not been possible before due to the form factor limit. The more compact design in a stacked sensor just makes them possible.

The second, and arguably the more interesting, class of sensors integrate computations, such as image signal and DNN processing, that are traditionally carried out outside an image sensor [23, 32, 42]. The computations are usually carried out in a DNN accelerator/DSP stacked with the pixel array layer. For instance, the Sony IMX 400 sensor is a 3-layer design that integrates a pixel layer, a DRAM layer (1 Gbit), and a logic layer with an ISP. The DRAM layer buffers high-rate frames before steaming them out to the host. This enables super slow motion (960 FPS). Otherwise, the bandwidth of the MIPI CSI-2 interface limits the capturing rate of the sensor.

Moving computation into a sensor has clear advantages. Most importantly, by consuming data closer to where they are generated, we reduce the data communication energy, which, as is well recognized, dominates the overall energy consumption. Communication inside a sensor through a micro through-silicon-via (uTSV) consumes two orders of magnitude lower energy than that through the MIPI CSI-2 interface. Now imagine instead of transferring an HD image (6 MB) we transfer only an object label (a few bytes) by running an object detection DNN inside the sensor. The savings on data transmission are more significant if the data has to be transmitted to the cloud when, for instance, the sensor itself has little to none computation capability. The energy cost of wireless communication is five orders of magnitude higher than that of uTSV.

While computations in existing stacked sensors are customized for specific tasks and will likely continue to be so, it should not be surprising that some image sensors have started integrating some form of programmable processors with a relatively involved memory hierarchy to allow more flexible computation inside the sensor. For instance, SCAMP-5 integrates an ALU, a set of registers and SRAMs within each pixel [11].

2.2.2 Challenges. Moving computation into an image sensor, while appealing, is not without challenges.

Inefficient Computation. Most importantly, computation inside an image sensor is inherently inefficient compared to that in the main processing chip — for two main reasons, both of which are out of cost-driven, practical considerations. First, image sensors are smaller in area than the SoC and, therefore, offer lower peak performance. Second, the process node of image sensors usually lags at least one generation behind that of the main SoC. Today, many commercial processors are fabricated using a 7 nm process node or smaller, but even high-end image sensors still use a 14 nm or 28 nm process node.

Thermal-Induced Noise. Performing more computation inside a sensor naturally increases the temperature. Unfortunately, image sensors are susceptible to thermal-induced noise, such as read noise and dark current noise. Recent research has shown that computation inside a sensor noticeably degrades both the perceived imaging quality and the computer vision task accuracy [6, 41].

Interestingly, while thermal noise is a concern, thermal hotspots are unlikely an issue. Our recent work [51] shows that 3D stacking increases power density for compute-dominant applications. The *absolute* power density, however, is generally at the order of mW/mm^2, three to four orders of magnitude lower than the power density of typical CPUs (up to $1W/mm^2$ [19]) and GPUs (up to $0.3W/mm^2$[14]). Such a low power density will unlikely lead to thermal hotspots and create a cooling challenge [84].

2.2.3 Promising Solutions. Given the two limitations, the ideal workload for in-image sensor computing is one where a trivial amount of computation inside the image sensor can drastically reduce the data transmission volume. The trivial computation ensures that the thermal-induced noise is minimal and minimizes the inefficiencies of the actual computation inside the sensor. Unfortunately, many common visual computing algorithms do not fit this ideal model [25]. Therefore, we must purposefully design new

algorithms that are amenable for in-image sensor processing by construction.

One such approach is in-sensor sampling, where we predict and extract only the Region-of-Interests (ROI) that are relevant to the downstream tasks and then sample pixels in the ROI. This strategy has the advantage of reducing both the amount of work done inside the sensor (fewer pixels are captured and fewer ADC invocations) and the amount of work done outside the sensor (downstream algorithms operate on sparse pixels).

We demonstrate in-sensor sampling for gaze tracking [26], a crucial task in both AR/VR and driving assistant systems (e.g., tracking the attention of drivers). We use a light-weight DNN to predict the ROI in a near-eye image [24] and perform random sampling within the ROI. Critically, the downstream gaze detection CNN is co-trained with the randomly sampled pixels, ensuring high end-to-end accuracy in the presence of sparse pixels. We show that our eye tracking system reduces pixel volume by about 95%, leading to an 8.2× energy reduction and a 1.4× tracking latency reduction compared to existing eye tracking systems, all with little degradation on the tracking accuracy.

2.3 Image Signal Processing

The ISP plays an important role in turning "imperfect" RAW pixels in the Bayer domain to pixels in the RGB/YUV domain through a series of algorithms such as dead pixel correction, demosaicking, and white-balancing. In architecture terms, the ISP is a specialized IP block in a mobile SoC, organized as a pipeline of mostly stencil operations on a set of local SRAMs ("line-buffers") [34, 74].

While ISPs are usually fixed-function ASIC, they are increasingly programmable so as to accommodate different computations and algorithms in support of higher imaging quality, which has become a strong product differentiator for mobile devices. For instance, the ISP in Google's Pixel2 smartphone is a fully programmable processor with its own VLIW ISA [61]. ISPs are now capable of sophisticated computational photography algorithms that are traditionally performed as separate image enhancement tasks, possibly off-line, using CPUs or GPUs. Examples include HDR imaging [1–3] and (temporal) denoising [38, 49].

2.3.1 Machine Vision-Optimized ISP Algorithms. The majority of images captured by today's cameras are consumed by machine vision algorithms rather than humans. Traditional ISP design, however, optimizes imaging quality for humans rather than machine vision. A recent line of work has been to rethink the ISP design, both its algorithms and hardware architecture, for machine vision.

For instance, Dirty Pixels [21], Buckler et al. [9], and ISP4ML [31] all investigated the importance of different ISP stages to machine vision quality. Some ISP stages that are critical to imaging quality, such as demosaciking and white balancing, can be skipped altogether if the images are to be consumed by machine vision algorithms. This perhaps is not surprising, since CNNs can be trained to learn the functions of those stages. However, CNNs learned directly on RAW data are ineffectively. They either require a much larger model or do not generalize well to different datasets/scenes, as empirically studied by Hansen et al. [31].

There are also proposals that attempt to replace the entire ISP with a DNN [63, 73]. There are at least two benefits of replacing an ISP with a DNN. First, we can co-train the ISP with the downstream machine vision DNNs to optimize for end-to-end quality [73]. Second, it becomes relatively easier to update the ISP algorithm — by replacing the model/weights, assuming there is a general-purpose DNN accelerator supporting the neural ISP. However, replacing the traditional ISP pipeline with a DNN adds computational cost.

2.3.2 ISP-Assisted Machine Vision. Another line of work uses metadata generated by the ISP, especially the motion vector metadata, to simplify downstream computations. The idea is to transform continuous machine vision algorithms from a frame computation into an incremental computation.

Frames in a real-time video stream are not independent. Instead, pixel changes are correlated in time, due to visual object motion. Our algorithm leverages the pixel *motion* information to incrementally execute vision algorithms so as to greatly reduce the total compute requirement. This motion-based incremental computation is formulated as follows:

$$f(x_t) = f(x_{t-1}) \bigoplus \delta(x_t, x_{t-1})$$

where x_t is the current frame, x_{t-1} is the previous frame, δ denotes the operation that calculates the increments between two frames (i.e., motion), and $\bigoplus$ denotes the operation that produces the vision results for the current frame by combining the previous frame's result with the pixel motion data. If both the δ and $\bigoplus$ operations are computationally cheaper than the original vision algorithm $f(\cdot)$, the vision results for the current frame can be calculated in a much more efficient way.

While incremental computation in general is a technique used in program analysis and optimization [55, 60], Euphrates [86] applies it to the domain of continuous vision. We demonstrate one set of δ and $\bigoplus$ operations that perform well in practice. In particular, we propose to encode $\delta(x_t, x_{t-1})$ as motion vectors [37], and to realize the $\bigoplus$ operation through motion extrapolation, which is six orders of magnitude lighter than executing a full CNN inference (i.e., $f(\cdot)$).

Critically, the motion vectors are naturally generated by the ISP as a byproduct of temporal denoising. That is, the ISP exposes the motion vectors to the vision engine such that the δ operation is performed "for free." This strategy exploits the algorithmic synergies between the ISP and the vision engine to avoid redundant computations while simplifying SoC design. The additional hardware requirement is low: only a simple augmentation to the ISP that saves and transfers the motion vectors to the SoC.

There is a host of work that leverages motion vectors to simplify downstream vision tasks. For instance, EVA^2 [8] accelerates DNN inference in continuous vision by using the motion vectors to extrapolate intermediate feature maps; ASV [27] uses the motion vectors across two stereo images to accelerate depth estimation from stereo images. Neither explicitly reuses the motion vectors from the ISP, but conceptually can.

3 Human-Systems Co-Optimizations

AR/VR and, to some extent, robots and self-driving cars, are human-facing systems. An important goal of human-facing systems is to stimulate certain percepts from humans. To do so, computing systems as an encoder and the HVS acts as a decoder. This is the idea

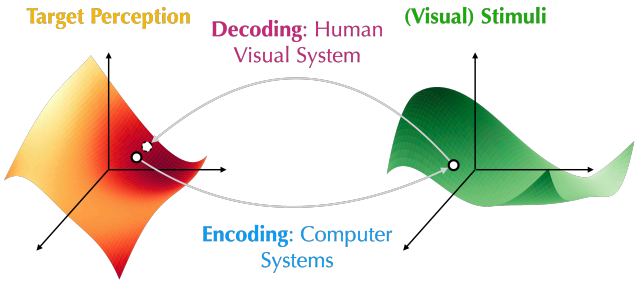

Fig. 4: The computer system encodes the percept intended to stimulate from humans (e.g., color, objects, depth, motion) as a set of visual stimuli (i.e., lights from the display pixels), which become the inputs to the HVS. The human visual system decodes the target percept from the incident lights.

illustrated in Figure 4. For instance, a VR system renders photorealistic images such that humans interpret the rendered objects as if they are from the real, physical world. In this sense, the computer system encodes the percept intended to evoke from humans (e.g., color, objects, depth, motion) as a set of visual stimuli (i.e., lights from the display pixels), which become the inputs to the HVS. The human visual system decodes the target percept from the incident lights. Again, this perspective naturally points to jointly optimizing the computer systems with the HVS.

3.1 Spatial Vision

Our peripheral visual acuity is extremely bad. If we fixate straight ahead, we will not be able to tell the details of an object in our peripheral vision. A great deal of work leverages the non-uniform spatial visual acuity to improve system performance.

Scientific Basis. The fundamental reason for low peripheral acuity is at least three-fold and is well-known.

(1) The receptive field (RF) sizes of Retinal Ganglion Cells (RGCs) increase with eccentricity, a result of larger dendritic fields [18, 62] and sparser RGC density in periphery [16]. A large RF means that a RGC integrates signals from a larger spatial area, i.e., more blurring in the (spatial) frequency domain.
(2) Cone cells (photoreceptors responsible for vision under normal daylight) become larger in size as eccentricity increases [17], also contributing blurring in spatial frequency.
(3) The distribution of cone cells on our retina is extremely non-uniform: over 95% of the cone cells are located in the central region of the retina (i.e., fovea) with an eccentricity of below 5 ° [17, 65]. The density of the cone cells decreases drastically in the visual periphery, which is, thus, significantly under-sampled spatially.

Improving Computer Systems. The low peripheral vision does not quite affect how computer systems are designed for PCs and smartphones, since the visual content coming from their displays will mostly fall in the fovea. When immersed in a virtual environment (e.g., when a user wears a VR headset), however, much of the visual stimuli generated from the computer systems will fall in the peripheral of the retina. This observation gave rise to the now well-established area of foveated rendering [29, 58, 66, 67], where one could improve the rendering speed by generating low-quality visual stimuli for the periphery with impunity. Our community has quickly picked up the idea and proposed hardware extensions to support foveated rendering in AR holograms [85] and cloud-assisted collaborative VR rendering [83].

While conceptually simply, we must answer a basic question: what exactly to render in the periphery without degrading perceptual quality? Perhaps unsurprisingly, today we simply blur or lower the resolution of the peripheral content. These empirical approaches, however, introduce suspicious artifacts [79], and it is not clear whether blurring content buys us any computation saving.

A scientifically sound answer requires understanding the complex processing that takes place in the entire human visual pathway, including processing on the retina, in the Lateral Geniculate Nucleus, and by the visual cortex of the brain. Assume we could model the human visual processing as a function f, and model the original input stimulus (without any degradation) as I. What we want to find is an alternative stimulus I' such that $f(I) = f(I')$, all the while minimizing the cost (of the underlying computing systems) to generate I'. This gives the following optimization problem:

$$\min_{I'} \; P(I') \quad s.t. \; f(I) = f(I') \tag{1}$$

As one might imagine, the central difficulty is how to model f, which we know is highly non-linear, dynamically self-adapting, feedback-driven, and most likely non-differentiable; one could even argue that the Holy Grail of neuroscience is to decipher f. Literature is rich with hypotheses and even models of bits and pieces in the entire system, but to date we simply do not know how to model f from first principles.

Since we cannot derive f from first principles (yet), we look for the next best thing: a computational model that fits experimental data. This is done through psychophysics, where we measure human behaviors under a set of controlled physical stimuli [28, 59].

Our recent work makes a good stride in this direction by computationally modeling one specific aspect of human vision: color perception. Through over 8,000 (IRB-approved) trials of psychophysical measurements on real participants, we build a computational model that predicts, for a given reference color at a given eccentricity, the set of colors that are perceptually no different from the reference color.

Leveraging the perception model, we design a VR rendering system that modulates pixel colors to minimize display power (dictated by colors) without affecting human color perception [15, 22]. Building on the principle of peripheral color confusion, we propose a compression algorithm that encodes perceptually similar colors together [75]. This algorithm, efficiently implemented in hardware, reduces the average memory traffic in a VR SoC by 67%.

Improving Imaging Systems. Complementary to improving the computer systems (and perhaps a bit out of place for this section) is to leverage spatial vision for improving imaging systems. Just like human vision, machine vision applications (e.g., drones and robots) do not require high-quality data at the camera periphery. Foveated image sensing is thus a natural idea: sense the center of camera field-of-view (FOV) with high quality at the expense of low-quality periphery sensing.

Foveated imaging can be realized in camera optics and/or sensor circuits. For instance, one might use a (3D printed!) micro-lens array to expose different pixels to different FOV sizes [72]. Alternatively, one might build the sensor circuit with non-uniform pixel shapes and sizes [57, 81].

A particularly interesting approach for foveated imaging is to integrate two imagers that share the same aperture in a camera [36]. he peripheral imager senses the entire FOV with low resolution, and the other foveated imager provides the "fine high-contrast details and color sensation of a narrow foveated region." Conceptually, this design is reminiscent of three-chip cameras for consumer photography [43] and multi-chip cameras for astrophysical imaging [5], both of which use multiple imagers to accurately capture color/spectrum information of the scene. The dual-sensor foveated imaging system has a similar idea, but applies it to foveated imaging.

3.2 Saliency

Complementary to peripheral vision, we can also exploit human visual saliency to reduce the rendering and streaming cost of VR (360 °) videos. Informally, saliency refers to stimuli in the scene that attract our attention. Our visual cortex builds a saliency map from the scene to guide our actions, e.g., gaze shifts. In practical terms, this means users will be more attracted to salient objects when watching a video.

Leveraging saliency, EVR is a one such cloud-client collaborative rendering system [46, 69]. The cloud service, deployed on Amazon EC2, extracts trajectories of salient objects in a video (i.e., stimuli that most likely attract user attention), pre-render them, and store them as much smaller "videolets". At rendering time given the real-time visual field of a user, only the best matching videolets are transmitted. EVR reduces the data transmission cost and avoids expensive on-device rendering, amounting to 58% overall energy reduction. One could also leverage saliency for compression: prioritize bits to perceptually more interesting visual areas. Vignette builds a DNN to predict saliency and presents a system for video compression and storage [53].

3.3 Temporal Vision

All the discussions so far are concerned with the spatial characteristics of human vision. The temporal dimension provides many interesting opportunities too. The most well-known aspect of temporal vision is saccades [4], where our eyes move rapidly when shifting visual attention between targets. Unless purposely trained, e.g., in the military, we simply cannot avoid saccades. On average, saccades occur 3–4 times per second (more frequent than heart beats) and last 20 ms – 200 ms each time, amounting to as many as 15 frames on a 90 FPS device.

Interestingly, human vision during saccades is momentarily blind, a phenomenon widely known as saccadic suppression [52, 71]. Application researchers use saccades to realize many interesting ideas, such as infinite walking in VR [68]. Saccades also temporally modulate the incident light signals, re-distributing power from 0 Hz temporal frequency to other temporal frequencies. This power re-distribution has been shown to have significant impact on visual sensitivity right after a saccade lands [7, 56]. Recent work exploits the post-saccade visual sensitivity change to improve the image resolution during VR rendering [44].

A related phenomenon is blink: our visual perception is also suppressed during eye blinks [78]. Vision research has shown that humans are functionally blind for about ten percent of the time due to blink-induced visual suppression, another opportunity to shave some computation cost. For instance, people have started using blinks for VR redirection/infinite walking [45].

Finally, keep in mind that the visual stimuli are generated from the displays, which have finite refresh rates. A low refresh rate reduces the computation and display power but introduces many artifacts such as flickering and blur (low refresh rate is the main reason moving objects look blurry to you on many TVs). A high refresh rate, in contrast, increases the computational load. This contention has led to recent work on variable refresh rate systems [20, 39].

4 Outlook

It is clear that continued progress in AMC, and visual computing in general, requires co-designing and co-optimizing imaging systems, computer systems, and human visual system. Many exciting opportunities lie ahead.

Rather than simply deciding which portion of an existing algorithm should be "offloaded" to the sensor, we must rethink algorithm design for in-image sensor processing. For instance, instead of mitigating or reducing the thermal-induced noise, can we embrace the noise and design downstream algorithms with the noise profile in mind? In the long run, holistically designing the sensing and computing architectures will provide a greater return of investment compared to designing each individually and simply stitching them together. This co-design hinges critically on the ability to model and explore the hybrid optical-electrical-mechanical design space across performance, power, and thermal measures.

As visual scientists keep questing for the fundamental operating principles of the HVS, systems researchers can help accelerate the rate of progress and increase the impact using our software/hardware optimization expertise. Many obvious questions follow: How to build lightweight computational models? How to coordinate the accelerators under the perceptual constraints? How to build a proactive, rather than reactive, system that re-configures itself by predicting human (visual) perception? Perhaps most importantly: what are the principled abstractions of the multifaceted HVS that we should expose the computer systems to? Without a doubt new phenomena in HVS beyond what are discussed here will be discovered and lend themselves to new systems opportunities. Having principled abstractions avoids the endless game of chasing after yet another new phenomenon.

Finally, while this opinion piece focuses on leveraging co-design for improving systems efficiency, a complementary and equally important direction is to enhance the capabilities both of humans and engineered systems.

References

[1] [n. d.]. ARM Mali Camera. https://www.arm.com/products/graphics-and-multimedia/mali-camera.

[2] [n. d.]. ASICFPGA Camera Image Signal Processing Core. http://asicfpga.com/site_upgrade/asicfpga/pds/isp_pds_files/ASICFPGA_ISP_Core_v4.0_simple.pdf.

[3] [n. d.]. DENALI-MC HDR ISP. http://pinnacleimagingsystems.com/embedded-products/.

[4] [n. d.]. Eye Wiki: Saccades. https://eyewiki.org/Saccade.

[5] [n. d.]. SDSS Camera. https://web.archive.org/web/20220201025926/https://www.sdss.org/instruments/camera/.

[6] Mohammad Faisal Amir, Jong Hwan Ko, Taesik Na, Duckhwan Kim, and Saibal Mukhopadhyay. 2018. 3-D stacked image sensor with deep neural network computation. *IEEE Sensors Journal* 18, 10 (2018), 4187–4199.

[7] Marco Boi, Martina Poletti, Jonathan D Victor, and Michele Rucci. 2017. Consequences of the oculomotor cycle for the dynamics of perception. *Current Biology* 27, 9 (2017), 1268–1277.

[8] Mark Buckler, Philip Bedoukian, Suren Jayasuriya, and Adrian Sampson. 2018. EVA^2: Exploiting Temporal Redundancy in Live Computer Vision. In *Proceedings of the 45th ACM/IEEE Annual International Symposium on Computer Architecture*.

[9] Mark Buckler, Suren Jayasuriya, and Adrian Sampson. 2017. Reconfiguring the imaging pipeline for computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*. 975–984.

[10] Carlos Mauricio Villegas Burgos, Tianqi Yang, Yuhao Zhu, and A Nickolas Vamivakas. 2021. Design framework for metasurface optics-based convolutional neural networks. *Applied Optics* 60, 15 (2021), 4356–4365.

[11] Stephen J Carey, Alexey Lopich, David RW Barr, Bin Wang, and Piotr Dudek. 2013. A 100,000 fps vision sensor with embedded 535GOPS/W 256× 256 SIMD processor array. In *2013 Symposium on VLSI Circuits*. IEEE, C182–C183.

[12] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. 2018. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific reports* 8, 1 (2018), 1–10.

[13] Julie Chang and Gordon Wetzstein. 2019. Deep optics for monocular depth estimation and 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10193–10202.

[14] John Y Chen. 2009. GPU technology trends and future requirements. In *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 1–6.

[15] Kenneth Chen, Budmonde Duinkharjav, Nisarg Ujjainkar, Ethan Shahan, Abhishek Tyagi, Jiayi He, Yuhao Zhu, and Qi Sun. 2023. Imperceptible Color Modulation for Power Saving in VR/AR. In *ACM SIGGRAPH 2023 Emerging Technologies*. 1–2.

[16] Christine A Curcio and Kimberly A Allen. 1990. Topography of ganglion cells in human retina. *Journal of comparative Neurology* 300, 1 (1990), 5–25.

[17] Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. 1990. Human photoreceptor topography. *Journal of comparative neurology* 292, 4 (1990), 497–523.

[18] Dennis M Dacey. 1993. The mosaic of midget ganglion cells in the human retina. *Journal of Neuroscience* 13, 12 (1993), 5334–5355.

[19] Andrew Danowitz, Kyle Kelley, James Mao, John P Stevenson, and Mark Horowitz. 2012. CPU DB: recording microprocessor history. *Commun. ACM* 55, 4 (2012), 55–63.

[20] Gyorgy Denes, Akshay Jindal, Aliaksei Mikhailiuk, and Rafał K Mantiuk. 2020. A perceptual model of motion quality for rendering with adaptive refresh-rate and resolution. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 133–1.

[21] Steven Diamond, Vincent Sitzmann, Frank Julca-Aguilar, Stephen Boyd, Gordon Wetzstein, and Felix Heide. 2021. Dirty pixels: Towards end-to-end image processing and perception. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–15.

[22] Budmonde Duinkharjav, Kenneth Chen, Abhishek Tyagi, Jiayi He, Yuhao Zhu, and Qi Sun. 2022. Color-perception-guided display power reduction for virtual reality. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.

[23] Ryoji Eki, Satoshi Yamada, Hiroyuki Ozawa, Hitoshi Kai, Kazuyuki Okuike, Hareesh Gowtham, Hidetomo Nakanishi, Edan Almog, Yoel Livne, Gadi Yuval, et al. 2021. 9.6 A 1/2.3 inch 12.3 Mpixel with on-chip 4.97 TOPS/W CNN processor back-illuminated stacked CMOS image sensor. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64. IEEE, 154–156.

[24] Yu Feng, Nathan Goulding-Hotta, Asif Khan, Hans Reyserhove, and Yuhao Zhu. 2022. Real-time gaze tracking with event-driven eye segmentation. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 399–408.

[25] Yu Feng, Tianrui Ma, Adith Boloor, Yuhao Zhu, and Xuan Zhang. 2023. Learned in-sensor visual computing: From compression to eventification. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 1–9.

[26] Yu Feng, Tianrui Ma, Yuhao Zhu, and Xuan Zhang. 2024. BlissCam: Boosting Eye Tracking Efficiency with Learned In-Sensor Sparse Sampling. In *Proceedings of the 51st International Symposium on Computer Architecture*. IEEE Computer Society, 1262–1277.

[27] Yu Feng, Paul Whatmough, and Yuhao Zhu. 2019. Asv: Accelerated stereo vision system. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 643–656.

[28] George A Gescheider. 2013. *Psychophysics: the fundamentals*. Psychology Press.

[29] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D graphics. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–10.

[30] Harel Haim, Shay Elmalem, Raja Giryes, Alex M Bronstein, and Emanuel Marom. 2018. Depth estimation from a single image using deep learned phase coded mask. *IEEE Transactions on Computational Imaging* 4, 3 (2018), 298–310.

[31] Patrick Hansen, Alexey Vilkin, Yury Krustalev, James Imber, Dumidu Talagala, David Hanwell, Matthew Mattina, and Paul N Whatmough. 2021. ISP4ML: The role of image signal processing in efficient deep learning vision systems. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2438–2445.

[32] Tsutomu Haruta, Tsutomu Nakajima, Jun Hashizume, Taku Umebayashi, Hiroshi Takahashi, Kazuo Taniguchi, Masami Kuroda, Hiroshi Sumihiro, Koji Enoki, Takatsugu Yamasaki, et al. 2017. 4.6 A 1/2.3 inch 20Mpixel 3-layer stacked CMOS Image Sensor with DRAM. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 76–77.

[33] Lei He, Guanghui Wang, and Zhanyi Hu. 2018. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing* 27, 9 (2018), 4676–4689.

[34] James Hegarty, John Brunhaver, Zachary DeVito, Jonathan Ragan-Kelley, Noy Cohen, Steven Bell, Artem Vasilyev, Mark Horowitz, and Pat Hanrahan. 2014. Darkroom: compiling high-level image processing code into hardware pipelines. *ACM Trans. Graph.* 33, 4 (2014), 144–1.

[35] Tomoki Hirata, Hironobu Murata, Hideaki Matsuda, Yojiro Tezuka, and Shiro Tsunai. 2021. 7.8 A 1-inch 17Mpixel 1000fps block-controlled coded-exposure back-illuminated stacked CMOS image sensor for computational imaging and adaptive dynamic range control. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64. IEEE, 120–122.

[36] Hong Hua and Sheng Liu. 2008. Dual-sensor foveated imaging system. *Applied optics* 47, 3 (2008), 317–327.

[37] M Jakubowski and G Pastuszak. 2013. Block-based Motion Estimation Algorithms–A Survey. *Opto-Electronics Review* (2013).

[38] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. 2010. Robust Video Denoising using Low Rank Matrix Completion. In *Proc. of CVPR*.

[39] Akshay Jindal, Krzysztof Wolski, Karol Myszkowski, and Rafał K Mantiuk. 2021. Perceptual model for adaptive local shading and refresh rate. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–18.

[40] Kedar Khare, Mansi Butola, and Sunaina Rajora. 2023. PSF engineering. In *Fourier Optics and Computational Imaging*. Springer, 249–260.

[41] Venkatesh Kodukula, Saad Katrawala, Britton Jones, Carole-Jean Wu, and Robert LiKamWa. 2021. Dynamic temperature management of near-sensor processing for energy-efficient high-fidelity imaging. *Sensors* 21, 3 (2021), 926.

[42] Oichi Kumagai, Atsumi Niwa, Katsuhiko Hanzawa, Hidetaka Kato, Shinichiro Futami, Toshio Ohyama, Tsutomu Imoto, Masahiko Nakamizo, Hirotaka Murakami, Tatsuki Nishino, et al. 2018. A 1/4-inch 3.9 Mpixel low-power event-driven back-illuminated stacked CMOS image sensor. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 86–88.

[43] Takao Kuroda. 2017. *Essential principles of image sensors*. CRC press.

[44] Yuna Kwak, Eric Penner, Xuan Wang, Mohammad R Saeedpour-Parizi, Olivier Mercier, Xiuyun Wu, Scott Murdison, and Phillip Guan. 2024. Saccade-Contingent Rendering. In *ACM SIGGRAPH 2024 Conference Papers*. 1–9.

[45] Eike Langbehn, Frank Steinicke, Markus Lappe, Gregory F Welch, and Gerd Bruder. 2018. In the blink of an eye: leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

[46] Yue Leng, Chi-Chun Chen, Qiuyue Sun, Jian Huang, and Yuhao Zhu. 2019. Energy-efficient video processing for virtual reality. In *Proceedings of the 46th International Symposium on Computer Architecture*. 91–103.

[47] Yingjie Li, Ruiyang Chen, Minhan Lou, Berardi Sensale-Rodriguez, Weilu Gao, and Cunxi Yu. 2023. LightRidge: An End-to-end Agile Design Framework for Diffractive Optical Neural Networks. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 4*. 202–218.

[48] Chiao Liu, Lyle Bainbridge, Andrew Berkovich, Song Chen, Wei Gao, Tsung-Hsun Tsai, Kazuya Mori, Rimon Ikeno, Masayuki Uno, Toshiyuki Isozaki, et al. 2020. A 4.6 μm, 512× 512, ultra-low power stacked digital pixel sensor with triple quantization and 127dB dynamic range. In *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 16–1.

[49] Ce Liu and William T Freeman. 2010. A High-Quality Video Denoising Algorithm based on Reliable Motion Estimation. In *Proc. of ECCV*.

[50] Yuankun Liu, Chongyang Zhang, Tingdong Kou, Yueyang Li, and Junfei Shen. 2021. End-to-end computational optics with a singlet lens for large depth-of-field imaging. *Optics express* 29, 18 (2021), 28530–28548.

[51] Tianrui Ma, Yu Feng, Xuan Zhang, and Yuhao Zhu. 2023. Camj: Enabling system-level energy modeling and architectural exploration for in-sensor visual computing. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–14.

[52] Ethel Matin. 1974. Saccadic suppression: a review and an analysis. *Psychological bulletin* 81, 12 (1974), 899.

[53] Amrita Mazumdar, Brandon Haynes, Magda Balazinska, Luis Ceze, Alvin Cheung, and Mark Oskin. 2019. Perceptual compression for video storage and processing systems. In *Proceedings of the ACM Symposium on Cloud Computing*. 179–192.

[54] Christopher A Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein. 2020. Deep optics for single-shot high-dynamic-range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1375–1385.

[55] Donald Michie. 1968. "Memo" functions and machine learning. *Nature* 218, 5136 (1968), 19.

[56] Naghmeh Mostofi, Zhetuo Zhao, Janis Intoy, Marco Boi, Jonathan D Victor, and Michele Rucci. 2020. Spatiotemporal content of saccade transients. *Current Biology* 30, 20 (2020), 3999–4008.

[57] Fernando Pardo, Bart Dierickx, and Danny Scheffer. 1997. CMOS foveated image sensor: Signal scaling and small geometry effects. *IEEE Transactions on Electron Devices* 44, 10 (1997), 1731–1737.

[58] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards Foveated Rendering for Gaze-Tracked Virtual Reality. *ACM Trans. Graph.* 35, 6, Article 179 (Nov. 2016), 12 pages. https://doi.org/10.1145/2980179.2980246

[59] Nicolaas Prins et al. 2016. *Psychophysics: a practical introduction*. Academic Press.

[60] William Pugh and Tim Teitelbaum. 1989. Incremental computation via function caching. In *Proc. of POPL*.

[61] Jason Redgrave, Albert Meixner, Nathan Goulding-Hotta, Artem Vasilyev, and Ofer Shacham. 2018. Pixel Visual Core: Google's Fully ProgrammableImage, Vision, and AI Processor For Mobile Devices. In *Proc. IEEE Hot Chips Symp.(HCS)*. 1–18.

[62] RW Rodieck, KF Binmoeller, and J Dineen. 1985. Parasol and midget ganglion cells of the human retina. *Journal of Comparative Neurology* 233, 1 (1985), 115–132.

[63] Eli Schwartz, Raja Giryes, and Alex M Bronstein. 2018. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing* 28, 2 (2018), 912–923.

[64] Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein. 2018. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

[65] Hongxin Song, Toco Yuen Ping Chui, Zhangyi Zhong, Ann E Elsner, and Stephen A Burns. 2011. Variation of cone photoreceptor packing density with retinal eccentricity and age. *Investigative ophthalmology & visual science* 52, 10 (2011), 7376–7384.

[66] Qi Sun, Fu-Chung Huang, Joohwan Kim, Li-Yi Wei, David Luebke, and Arie Kaufman. 2017. Perceptually-Guided Foveation for Light Field Displays. *ACM Trans. Graph.* 36, 6, Article 192 (Nov. 2017), 13 pages. https://doi.org/10.1145/3130800.3130807

[67] Qi Sun, Fu-Chung Huang, Li-Yi Wei, David Luebke, Arie Kaufman, and Joohwan Kim. 2020. Eccentricity effects on blur and depth perception. *Optics express* 28, 5 (2020), 6734–6739.

[68] Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie Kaufman. 2018. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

[69] Qiuyue Sun, Amir Taherin, Yawo Siatitse, and Yuhao Zhu. 2020. Energy-efficient 360-degree video rendering on fpga via algorithm-architecture co-design. In *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 97–103.

[70] Qilin Sun, Ethan Tseng, Qiang Fu, Wolfgang Heidrich, and Felix Heide. 2020. Learning rank-1 diffractive optics for single-shot high dynamic range imaging. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1386–1396.

[71] Alexander Thiele, Peter Henning, M Kubischik, and K-P Hoffmann. 2002. Neural mechanisms of saccadic suppression. *Science* 295, 5564 (2002), 2460–2462.

[72] Simon Thiele, Kathrin Arzenbacher, Timo Gissibl, Harald Giessen, and Alois M Herkommer. 2017. 3D-printed eagle eye: Compound microlens system for foveated imaging. *Science advances* 3, 2 (2017), e1602655.

[73] Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-Francois Lalonde, and Felix Heide. 2021. Differentiable compound optics and processing pipeline optimization for end-to-end camera design. *ACM Transactions on Graphics (TOG)* 40, 2 (2021), 1–19.

[74] Nisarg Ujjainkar, Jingwen Leng, and Yuhao Zhu. 2023. ImaGen: A general framework for generating memory-and power-efficient image processing accelerators. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–13.

[75] Nisarg Ujjainkar, Ethan Shahan, Kenneth Chen, Budmonde Duinkharjav, Qi Sun, and Yuhao Zhu. 2024. Exploiting Human Color Discrimination for Memory- and Energy-Efficient Image Encoding in Virtual Reality. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 166–180.

[76] Vincent C Venezia, Alan Chih-Wei Hsiung, Kelvin Ai, Xiang Zhao, Zhiqiang Lin, Duli Mao, Armin Yazdani, Eric AG Webster, and Lindsay A Grant. 2018. 1.5 μm Dual Conversion Gain, Backside Illuminated Image Sensor Using Stacked Pixel Level Connections with 13ke-Full-Well Capacitance and 0.8 e-Noise. In *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 10–1.

[77] Carlos Mauricio Villegas Burgos, Pei Xiong, Liangyu Qiu, Yuhao Zhu, and A Nickolas Vamivakas. 2023. Co-designed metaoptoelectronic deep learning. *Optics Express* 31, 4 (2023), 6453–6463.

[78] Frances C Volkmann, Lorrin A Riggs, and Robert K Moore. 1980. Eyeblinks and visual suppression. *Science* 207, 4433 (1980), 900–902.

[79] David R Walton, Rafael Kuffner Dos Anjos, Sebastian Friston, David Swapp, Kaan Akşit, Anthony Steed, and Tobias Ritschel. 2021. Beyond blur: Real-time ventral metamers for foveated rendering. *ACM Transactions on Graphics* 40, 4 (2021), 1–14.

[80] Gordon Wetzstein, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David AB Miller, and Demetri Psaltis. 2020. Inference in artificial intelligence with deep optics and photonics. *Nature* 588, 7836 (2020), 39–47.

[81] Robert Wodnicki, Gordon W Roberts, and Martin D Levine. 1995. A foveated image sensor in standard CMOS technology. In *Proceedings of the IEEE 1995 Custom Integrated Circuits Conference*. IEEE, 357–360.

[82] Yicheng Wu, Vivek Boominathan, Huaijin Chen, Aswin Sankaranarayanan, and Ashok Veeraraghavan. 2019. Phasecam3d—learning phase masks for passive single view depth estimation. In *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 1–12.

[83] Chenhao Xie, Xie Li, Yang Hu, Huwan Peng, Michael Taylor, and Shuaiwen Leon Song. 2021. Q-vr: system-level design for future mobile collaborative virtual reality. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 587–599.

[84] Ying-Ju Yu and Carole-Jean Wu. 2018. Designing a temperature model to understand the thermal challenges of portable computing platforms. In *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE, 992–999.

[85] Shulin Zhao, Haibo Zhang, Cyan Subhra Mishra, Sandeepa Bhuyan, Ziyu Ying, Mahmut Taylan Kandemir, Anand Sivasubramaniam, and Chita Das. 2021. Holoar: On-the-fly optimization of 3d holographic processing for augmented reality. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. 494–506.

[86] Yuhao Zhu, Anand Samajdar, Matthew Mattina, and Paul Whatmough. 2018. Euphrates: algorithm-SoC co-design for low-power mobile continuous vision. In *Proceedings of the 45th Annual International Symposium on Computer Architecture*. 547–560.