An Invitation to Visual Computing

Yuhao Zhu

Department of Computer Science
Department of Brain and Cognitive Sciences
University of Rochester

yzhu@rochester.edu
https://yuhaozhu.com/
https://horizon-lab.org/

1 What is Visual Computing?

We can think of many things when it comes to visual computing. Cameras? Yes; they turn the world into visually pleasing images. Computer Graphics? Yes; they simulate how visually pleasing images are captured as if there was a camera placed in the scene. Computer vision? Yes; it interprets visual information (i.e., images) to infer semantic information of the world (e.g., object categories). Displays? Yes; they generate visual information (i.e., lights) to represent an intended scene. What about Augmented Reality (AR) and Virtual Reality (VR)? Of course; in fact, AR/VR requires all things above to work seamlessly together.

But what are the fundamental connections of the multitude of things that we can all loosely associate with visual computing? Figure 1 presents the key concepts that unify the different fields of visual computing: 1) representing the physical world in three fundamental information domains, i.e., the optical, electrical, and semantic domains, 2) processing signals within these domains, and 3) transforming signals across these domains.

We will use the Human Visual System (HVS) as an example to walk through some of the key concepts (Chapter 2). We will then expand to three more visual computing domains (computer imaging, computer graphics and rendering, and machine vision), comparing and contrasting how the signal representations, processing, and transformations are exercised in different systems (Chapter 3). We will introduce a power abstraction governing any visual computing systems. This abstraction allow us to reason about the limits of a system and design ways to improve a system (Chapter 4).

2 Human Visual System as a Visual Computing Platform

Imagine you are taking a walk in the woods and notice a butterfly. How does your visual system allow you to notice the butterfly and that it is flying? The input to an HVS are lights from the butterfly and the trees in the physical world; they are information represented in the optical domain. The output of the HVS is semantic information, e.g., the color and motion of the

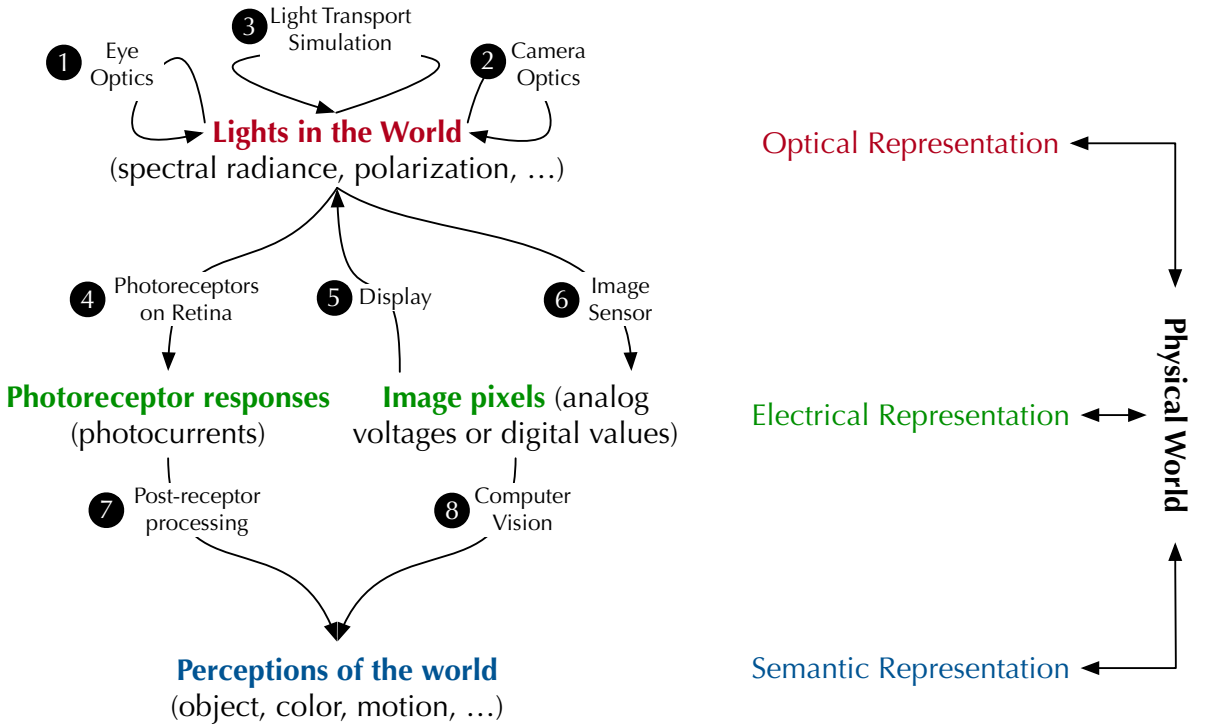

Figure 1: A framework unifying visual computing. The fundamental building blocks are the signals represented in three fundamental information domains: optical, electrical, and semantic. Visual computing systems transform signals across and process them within these domains.

butterfly. The HVS extracts semantics information from the optical signals through a sequence of signal transformations illustrated as ❶ → ❹ → ❼ in Figure 1.

2.1 Signal Representations, Processing, and Transformations in HVS

Optical Signal Processing

First of all, lights enter your eyes by traveling through the ocular media in your eyes, such as the cornea, pupil, and lenses, and eventually hit the retina. Just before the lights get processed by the retina, the optical signal is already being processed as lights propagate through the eye. This is illustrated by ❶ in Figure 1. For instance, the ocular media absorbs photons of certain wavelengths and transmit photons that are unabsorbed. The pupil controls how many photons are allowed in at any given time, and the lens bends and *focuses* lights on the retina — the chief goal of the eye.

The optical information after eye optics and right before being processed by the retina is usually called the *optical image*. An optical image is a lossy and aberrated version of the optical information in the scene — because the optical signal processing in the eye is lossy. For instance,

by focusing on the butterfly, which is at a particular depth, objects are other depths such as the trees in the background are blurred. The ocular media also absorb photons selectively across wavelengths, so the true light spectra in the scene are lost.

Optical to Electrical Signal Transduction

The optical image gets transformed to an electrical representation by the photoreceptors on the retina. This is step ❹ in Figure 1. Photoreceptors absorb incident photons; once a photon is absorbed, it could, through *phototransduction* cascade [Wald, 1968] (the discovery of which won George Wald his Nobel Prize), generate electrical responses in the form of photocurrents or, equivalently, photovoltages across the cell membrane of the photoreceptor. The responses of all the photoreceptors form the electrical representation of the optical image. The rest of the visual system is "just" a hugely complicated circuit that processes the electrical signals from the photoreceptors. In this sense, the optical to electrical transformation is the first step in seeing.

This optical to electrical signal transduction is once again lossy. Photoreceptors sample and integrate signals spatially, temporally, and spectrally. As a result, much of the optical information of the incident light, such as the incident angle of the rays, the wavelengths of the photons, and the polarization of the lights, is all lost. The main information that *is* retained, light intensity, is fundamentally limited by the sampling and integration, which establish the limits of vision.

Electrical to Semantic Signal Transduction

The electrical signals produced by the photoreceptors are first processed by the rest of the neurons on the retina, and then transmitted in the nervous system to the rest of the visual system, first to the Lateral Geniculate Nucleus (LGN) and then to the visual cortex, where the electrical signals undergo further processing and eventually the semantic meanings of the scene arise. You might now realize that the object is in fact a red lacewing butterfly (object recognition), the color of the butterfly is an astonishing bright red and pale brown interlaced by black and white (color perception), and the butterfly is flapping and flying (motion perception). We lump all the processing stages after the photoreceptor and call them "post-receptor" processing, which is denoted by ❼ in Figure 1.

The post-receptor processing progressively extracts richer and higher-level information as the signal progresses through the retina-LGN-cortex pathway. The retina encodes information such as the spatial/temporal frequency, contrast, and, to a large extend, color. This set information is generally regarded as "low level" information, which does not in anyway suggest that the information is somehow inferior; quite the contrary, this set information provides the basis for higher-order visual processing.

It is no small feat that our retina can extract such information: it must reliably do so across a very wide range of illumination conditions. For instance, the retina adapts to different illumination levels spanning several orders of magnitude. Perhaps somewhat surprisingly, much of the adaptation takes place within the photoreceptors, whose sensitivity changes based on the

incident light intensity. This suggests that photoreceptors are not merely signal transduction devices.

The LGN and early areas in the visual cortex extract information like edge and orientation, and other higher-order areas further refine the signals to extract information such as motion, depth, and object category. We hasten to add that information processing in the visual system is not purely feed-forward. There are many feedback paths between cortical areas and between the cortex and the LGN [Gilbert and Li, 2013].

2.2 The Transformations are Born of Necessity

This complex sequence of transformation that turns the physical realities to one's subjective percept is born of necessity. A comparison is another sequence of transformations that computer scientists are perhaps more very familiar with. To have a computer solve a problem for us, we first describe the algorithm in a program written in a high-level language and then transform the program to a low-level, machine understandable language (i.e., the Instruction Set Architecture), which is then executed on the microarchitecture implemented using circuits and, eventually, moving electrons. If we could directly talk to the electrons and instruct them to move to solve our problem, this sequence of transformations is not strictly necessary. Similarly, if we could crack open one's head and manipulate the nerve impulses at will, we could perhaps directly impose certain percepts on humans. But since we cannot (yet), the sequence of signal transduction is necessary.

3 Engineered Visual Computing Systems

While the example above is drawn from a biological system, engineered visual computing systems such as smartphones are fundamentally no different in that they all involve visual information represented in and transformed between different domains. We will consider three examples of engineered systems, and compare and contrast them with those in the human visual system.

3.1 Computer Imaging and Digital Photography

Imaging refers to the task of capturing images of the physical world. Photography is sometimes used interchangeably with imaging. Just to be pedantic, however, photography is a special case of imaging where the goal is to capture *visually pleasing* images for the human visual system. Scientific imaging is another branch of imaging, where the goal is to capture *physically accurate* information for scientific inquiry. Examples of scientific imaging include astrophotography, microscopy, and Computed Tomography (CT). We will focus on photography here. Conventional photography is purely analog; think of dark rooms and film development. Modern imaging is computer-assisted, hence the name "computer imaging", not to be confused with "computational imaging" or "computational photography", which we will see later.

An end-to-end photography system is a complicated sequence of signal transductions involving ❷ → ❻ → ❺ → ❶ → ❹ → ❼ in Figure 1 Lights enter the camera and are first processed by the optics in the camera with the main goal of focusing lights (❷), similar to eye optics. Camera

optics are designed completely by humans and we can, therefore, specifically engineer them to achieve a particular performance, whereas eye optics do not enjoy such flexibilities. An example is compound lenses, where a combination of lenses of different kinds are cascaded together to correct various aberrations that a single (spherical) lens introduces.

After the lenses, lights hit the image sensor, whose main job is to transform optical signals to electrical signals (❻). This is achieved by an array of light-sensitive photodiodes, or pixels, that convert photons to electric charges—using the *photoelectric effect* [Einstein, 1905a,b] (the discovery of which won Albert Einstein his Nobel Prize)—which are then converted to digital values, i.e., image pixels. From the signal transduction perspective, the pixels in an image sensor are "just" like photoreceptors on the retina. Vision scientists might take offense at this comparison, because the photoreceptors, as alluded to earlier, are much "smarter" and do a lot more than the pixels, e.g., visual adaptation. In fact, an active area of research is to design pixels so that they adapt like photoreceptors [Liao et al., 2022].

Eventually, an image needs to be displayed for the human visual system to see. The display performs an electrical to optical signal transduction, turning digital pixels to lights (❺). The photons from the display then enter human eyes, and what we have discussed before about the HVS apply.

3.2 Computer Graphics and Rendering

Computer graphics and rendering systems generate images, where photorealism is the main goal (although not the exclusive goal). What does it take to render photorealistic images? A rendered image is photorealistic if it almost looks like a photo taken by a camera, so to render something photorealistic, we want to simulate how a photo is taken! To that end, we must simulate two things: 1) how lights transport in space before entering the camera and 2) how lights gets turned into pixels by a camera, which follows the signal chain in an imaging system.

Comparatively speaking, the second simulation is easier; it amounts to simulating the image formation process in a camera (i.e., ❷ → ❻). Since cameras are built by humans, we know exactly how they work at least in principle. The first simulation is much harder, because it requires simulating the nature: modeling the complicated light-matter interactions (❺).

This is why most compelling rendering systems are physically-based. The kind of physical models used for rendering are *phenomenological* in nature; they describe the empirical rules governing the light-matter interactions but are not always derived from first principles. An example is that we model light reflecting at material surfaces using Bidirectional Reflectance Distribution Function (BRDF), which maps energy from incident rays to exiting rays while abstracting away the details of radiative/energy transfer resulted from photons interacting with particles in the material [Chandrasekhar, 1960].

Using phenomenological models is sometimes the only option when the actual underlying physics elude us. More importantly, however, simulating physics at the lowest level is simply unnecessary for rendering (which cares about photorealism rather than physical realism) and is computationally too costly for real-time rendering. A recent trend in graphics is neural rendering [Mildenhall et al., 2021], which parameterizes the phenomenological models using deep

neural networks and learns such models from actual images, which, by definition, are precisely simulated — by nature.

Similar to photography, the rendered images will also go through an electrical to optical signal transformation by the display, whose output is then consumed by the HVS.

3.3 Machine Vision Systems

For better or worse, machine vision systems are prevalent in modern society. Autonomous vehicles use machine vision to navigate the environment, drones are used in agriculture to monitor crop health, and facial recognition is increasingly used for security authentications. A machine vision system has two main components: 1) an imaging sub-system that, like discussed above, transforms the optical information in the scene to the electrical information encoded by the image pixels (❷ → ❻) and 2) a computing sub-system, which uses computer vision algorithms to interpret the images and extract meanings from the scene (❽).

At the risk of once again downplaying the capabilities and complexities of the HVS, one can argue that a machine vision system largely emulate the HVS — from a signal transduction perspective. Both aim to extract semantical information from the physical world, both do so by first turning the optical information in the world to its electrical representation, and one can even go as far as saying that today's dominant paradigm toward computer vision, i.e., deep learning, is heavily inspired by the HVS. The field of neuromorphic computing explicitly aims to mimic the structure and operation of the human brain.

A key difference between imaging in machine vision and imaging for photography is their respective consumer: the output of a photograph is meant to be consumed by a HVS, so visual quality is the main consideration, whereas images captured by, for instance, a robot are meant to be consumed by the downstream computer vision algorithms, which do not care about the visual appearance as long as the semantics information can be decoded from the images. This difference influences the design of the imaging system used in photography and for machine vision.

4 A Powerful Abstraction

A visual computing system enlists the work of multiple stages of signal transformations. At every stage in an application's pipeline, we have decisions to make. These decisions should not be made locally to optimize for a specific stage. A lot of the exciting research in visual computing is to jointly design and optimize all the stages in an end-to-end system. This section provides two concrete examples. But before we can entertain them, we want to first introduce a power abstraction that will allow us to reason about these research ideas.

4.1 The Encoding-Decoding Abstraction

We can take an information-theoretical perspective, and abstract virtually any end-to-end visual computing pipeline as an encoding-decoding process. Decoding is the ultimate goal, but encoding is necessary, because it transforms signals to a domain that can be processed by the decoder.

For instance in human vision and machine vision systems, while the ultimate goal is to generate percepts of the physical world, information in the world must be first encoded as electrical signals (through imaging), which are what the brain and computer vision algorithms can process. Imaging itself can also be regarded as an encoding-decoding pair, where the optical information of the scene is first encoded in the electrical domain and the computational algorithms, acting as a decoder, reconstruct an electrical representation that faithfully captures the information in the original scene.

A more complicated example is visual display devices such as a VR headset. When developing a VR application, we usually have a scene in mind, e.g., a red lacewing butterfly flying in the woods. We hope that users will perceive the object (butterfly), color (the astonishing bright red and pale brown interlaced by black and white), and motion (flapping and flying), but we cannot simply impose these percepts on humans. Instead, we generate visual stimuli on the display to encode the desired percepts. This encoding is done through a combination of rendering (generating electrical signals) and display (converting electrical signals to optical signals). The entire HVS then acts as the decoder, which ideally would provide the intended percepts to users.

Encoding Capabilities Set the Limits on Decoding

Once we take this encoding-decoding abstraction, we can start reasoning about limits of a visual computing system. The decoder consumes information generated by the encoder, so its utility is fundamentally limited by the encoding capabilities. Ideally, the encoder should faithfully capture all the information in the world. But in practice, encoding is almost always lossy — for a number of reasons.

First, the actual encoding device used in a system, be it biological or engineered, usually uses fundamentally lossy mechanisms such as sampling and low-pass filtering (e.g., integration). Take HVS as an example, where the optical information of the scene is encoded as photoreceptor responses. The photoreceptors sample the continuous optical image impinges on the retina. The sampling rate dictates, according to the Nyquist–Shannon sampling theorem [Shannon, 1949], how well the original optical image can be reconstructed, which in turns limits our ability to see fine details. Even before the photoreceptor sampling, the eye lens blurs signals in the scene not currently in focus and the pupil, when very small, further blurs even in-focus objects through diffraction, setting the first limit of vision. Blurring is a form of low-pass filtering and is one of the many optical aberrations introduced during the optical signal processing in the HVS.

Second, an encoding device might completely disregard certain information in the incident signal. For instance, the polarization information in the incident light is simply ignored by the photoreceptors, whose responses are, thus, invariant to the polarization states. As a result, humans cannot "see" polarization. Some animals, such as butterflies, have polarization-sensitive photoreceptors. So it is not surprising that monarch butterflies make use of the light polarization for navigation [Reppert et al., 2004].

Jointly Design Encoding and Decoding

The encoder-decoder abstraction also allows us to design strategies to enhance a visual system, both augmenting its capabilities and improving its execution efficiency. For instance, when certain information is not needed for an accurate decoding, it needs not be encoded in the first place and, of course, will not participate in decoding, reducing both encoding and decoding costs. Alternatively, if we know what information is crucial for decoding we can design the encoding system to specifically capture such information.

Ultimately, exploiting these ideas amounts of co-designing the encoder and decoder considering the end-to-end requirements of task performance, efficiency, and quality. We will discuss two classic examples.

4.2 Encoding-Decoding Co-Design: Two Examples

Computational Photography

The optical to electrical signal transduction in the image sensor are lossy due to various forms of signal sampling and integration. The signal transduction process itself is also not perfect due to fundamental physical limitations (e.g., quantal fluctuation in photon arrivals) and practical engineering considerations (e.g., sensor size). As a result, the sensor output, which is usually called *raw pixels*, is noisy (especially in low-light conditions) and does not accurately represent the luminance (especially under bright illuminations) and color information in the scene; certain information such as light-field and polarization is completely lost.

To overcome these limitations, modern smartphones and advanced imaging systems use computational algorithms to correct those imperfections, reconstruct the lost information, and sometimes can even add an artistic touch to the photo. Critically, such computational algorithms are usually jointly designed with the imaging system, i.e., the optics and image sensor. This is computational photography, co-designing camera optics, image sensor, and the computational algorithms to overcome fundamental limitations that conventional imaging systems face.

A classic example of computational photography has to do with a practical problem in photography. As a contemporary reader, you most likely have had the experience where you want to use your smartphone camera to capture a scene that has both a very bright region (e.g., the sunny sky) and a relatively dark region (e.g., a street corner). In technical terms, such a scene has a very high *dynamic range* (HDR), in that the ratio between the highest and lowest luminance in the scene is huge. The challenge is that image sensors on smartphones cannot capture a wide dynamic range: information at low-luminance region is noisy and high-luminance regions saturate pixels. So how do we capture the full luminance range in the scene? This is task of HDR imaging.

People over the years have come up with a variety of clever ideas for HDR imaging. The most well-known is perhaps exposure bracketing, where we take multiple captures of the scene, each with a different exposure time, and the computationally combine the captures to synthesize the full dynamic range in the scene. Another approach is Google's HDR+ algorithm [Hasinoff et al., 2016], which takes multiple exposures using the same (low) exposure time to ensure high luminance regions are accurately captured, which is then followed by denoising algorithms (e.g.,

frame averaging) to recover low luminance information. Yet another approach is the time-to-saturation (TTS) image sensor [Stoppa et al., 2002], where measures the time it takes for each pixel to saturate and uses the time to extrapolate the luminance information.

These HDR imaging techniques are all examples where the imaging system, i.e., the encoder, are intentionally designed to capture critical information (luminance) that is otherwise lost (either due to noise or due to saturation).

Gaze-Contingent Rendering

Certain information will necessarily be lost during decoding and, thus, need not be encoded in the first place. Consider an AR device, where the lights coming out of the display are processed by the HVS. The displayed stimuli must satisfy, but not exceed, the capabilities of the HVS. This presents opportunities to optimize encoding, i.e., everything before the photons enter the eye, including camera imaging, rendering algorithms, and displays.

Gaze-contingent rendering is a well-known technique in AR/VR that exploits this opportunity. Our peripheral visual acuity is extremely bad: we could not tell the details of an object in our peripheral vision. This is mainly a result of: 1) a higher degree of low-pass filtering due to neural convergence in the periphery and 2) a lower rate of sampling in the periphery due to drastically fewer photoreceptors.

When immersed in a virtual environment with a VR headset, the majority of the pixels rendered and displayed fall in the periphery of the retina. Therefore, one could improve the rendering speed by generating low-quality visual stimuli for the periphery with impunity [Patney et al., 2016, Guenter et al., 2012]. We could also alter pixel colors in the periphery to reduce display power without introducing artifacts [Duinkharjav et al., 2022].

References

Subrahmanyan Chandrasekhar. *Radiative transfer*. Courier Corporation, 1960.

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

Albert Einstein. Über einen die erzeugung und verwandlung des lichtes betreffenden heuristischen gesichtspunkt, 1905a.

Albert Einstein. On a heuristic point of view about the creation and conversion of light. *Annalen der Physik*, 17(6):132–148, 1905b.

Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature reviews neuroscience*, 14(5):350–363, 2013.

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

Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Fuyou Liao, Zheng Zhou, Beom Jin Kim, Jiewei Chen, Jingli Wang, Tianqing Wan, Yue Zhou, Anh Tuan Hoang, Cong Wang, Jinfeng Kang, et al. Bioinspired in-sensor visual adaptation for accurate perception. *Nature Electronics*, 5(2):84–91, 2022.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

Steven M Reppert, Haisun Zhu, and Richard H White. Polarized light helps monarch butterflies navigate. *Current Biology*, 14(2):155–158, 2004.

Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37 (1):10–21, 1949.

David Stoppa, Andrea Simoni, Lorenzo Gonzo, Massimo Gottardi, and G-F Dalla Betta. Novel cmos image sensor with a 132-db dynamic range. *IEEE Journal of Solid-State Circuits*, 37 (12):1846–1852, 2002.

George Wald. Molecular basis of visual excitation. *Science*, 162(3850):230–239, 1968.