ChatGPT

Toward the Modular Integration of In-Air Holographic Display Systems: A Technical Framework for Multi-Modal, Neural-Responsive Toroidal Visualizations

Abstract

In this paper, we survey and connect emerging technologies in **holographic display, volumetric visualization, and bio-responsive interfaces** toward a modular system design. Our goal is to outline a flexible framework for producing **in-air "holographic donut" (toroidal) imagery** that can respond to multi-modal inputs such as EEG brainwaves. We evaluate the feasibility of projecting a **toroidal light-field in mid-air**, mapping known research and commercial tools to each component of the system. The result is a proposed roadmap for constructing a **neural-responsive holographic torus**: a dynamic light structure that can serve artistic, meditative, and interactive purposes. Feasibility, technical challenges, and integration strategies are discussed with references to state-of-the-art volumetric display methods, real-time holography engines, sensor interfaces, and control middleware. This framework is intended to guide future prototyping and inspire cross-disciplinary innovation in **3D display and biofeedback applications**.

Introduction

The pursuit of realistic **holographic displays** – visuals that appear to float freely in space – has accelerated in recent years. From the **laser-plasma 3D images** demonstrated in the mid-2000s [1] to modern **light field panels** and **volumetric projectors**, the evolution of holographic display technology is closing the gap between science fiction and reality [2] [3]. A parallel trend has been the exploration of **biosignals (e.g. EEG brainwaves)** as input for interactive art and visualization, allowing a viewer's mental or emotional state to influence the visuals they see [4] [5]. Merging these domains, we envision a **mid-air volumetric display** in the form of a **toroid (doughnut shape)** that can respond to neural and other inputs in real time.

The **toroidal form** carries rich significance in both science and culture. Geometrically, a torus is a ring-shaped surface generated by rotating a circle about an axis – a shape exhibiting continuous curvature and symmetry. Symbolically, the torus appears in **sacred geometry as a representation of universal energy flow** and of the "human toroidal field," suggesting harmony between inner and outer worlds [6]. Its geometry embodies a **perpetual, self-referential cycle** – energy expanding outward from the center and returning through the core, forming a dynamic loop [7]. Such attributes have made the torus a metaphor for concepts of unity, infinity, and self-sustaining systems. Recent artistic works have even employed holographic torus shapes to evoke these ideas: for example, artist Deana Lawson's 2021 exhibit at the Guggenheim featured a **holographic torus sculpture** as a "force" symbolizing the power and potential of her subjects [8].

A holographic torus on display as the centerpiece of an art installation. From one angle it resembles an ordinary ring ("a bagel"), but walking around it reveals an otherworldly portal effect – the torus appears to rotate and fold into itself in mid-air [9]. Such a toroidal light sculpture illustrates the cultural fascination with this geometry and demonstrates mid-air display technology applied in an artistic context.

Motivated by these converging threads – technological advances in **free-space displays** and the symbolic allure of the **torus** – we propose a modular system concept nicknamed the "holographic vortex." It aims to create a **suspended torus of light** that can be dynamically modulated. The system would be flexible and hardware-agnostic, integrating multiple display techniques and input modalities. In practical terms, this could enable scenarios such as: a **public installation** where passersby see a luminous donut of light responding to their gestures; a **wearable halo or pendant** that projects a tiny volumetric torus above the device; a **meditative biofeedback display** where one's calm brainwaves steady the rotation of a light vortex; or an **art-tech performance** where music, motion, and EEG all drive the behavior of a floating torus visualization.

In the following sections, we first describe the envisioned system and usage modes. We then survey the **technology layers** needed – from the display hardware core to rendering techniques, sensor interfaces, and control software – citing current research, commercial products, and open tools relevant to each. Next, we illustrate **integration scenarios** linking inputs to the torus output. We discuss **technical challenges** (e.g. brightness, alignment, safety) and potential workarounds. Finally, a **roadmap to prototyping** is outlined, from low-fidelity AR experiments to large-scale installations and future wearable designs, before concluding with reflections on feasibility and interdisciplinary opportunities.

System Concept and Use-Case

Envisioned System: The proposed system is essentially a **modular holographic display platform** shaped into a torus (doughnut). The "torus" could be rendered via various means (as discussed in Section 4.1) – for example, as a 3D volume of light points in mid-air, or as a persistence-of-vision projection that creates the *illusion* of a ring. The torus might be **suspended in space** (e.g. atop a pedestal or within a chamber) or even attached to a person (for wearable use-cases). Crucially, the design is **hardware-agnostic**: the system architecture would allow swapping in different display modules (laser plasma, mist projector, etc.) as long as they can produce the desired toroidal imagery. Likewise, the input sources and interaction modalities are modular.

Modular Design Goals: The system is conceived as a set of layers or modules with defined interfaces. For instance, the **display module** takes abstract shape data (e.g. coordinates or a 3D model of a torus) and produces the visual output. Upstream, a **rendering engine** generates that shape data, possibly applying colors, motion, or patterns. The **input module** feeds in sensor data (brain signals, gestures, audio, etc.), and a **control module** translates those inputs into real-time visual modulations. By isolating these layers, one could mix-and-match components – for example, using a high-end light field panel in a lab setup, but a portable LED-based projector for a wearable scenario – without redesigning the whole system. Scalability is also a goal: the framework should handle a small 5 cm torus as well as a large 5 m installation by adjusting parameters and possibly using arrays of synchronized display units.

Interaction Model: A core feature is **responsiveness** – the torus visualization should respond to user input or environmental data in meaningful ways. We envision multiple modes of operation:

- **EEG-reactive Mode:** The system connects to an **EEG headset** worn by a user. Brainwave patterns (e.g. alpha relaxation waves vs. beta focus waves) influence the torus's appearance – for example, changing its color or the speed at which it spins. A calm meditative state might produce a slow-rotating, coherent torus with cool colors, whereas high excitement could make it pulsate or flicker with intense hues. This creates a biofeedback loop, where seeing the torus calm down encourages the user to relax, and vice versa [4].

- **Touch/Gesture-Responsive Mode:** Using **motion sensors or hand-tracking** (like a Leap Motion controller), the user can manipulate the floating donut by hand movements. For instance, reaching out and "grabbing" the torus could tilt or move it, or making a pinch gesture could squeeze the ring, altering its thickness. Recent research has even enabled *direct touching* of certain volumetric displays – e.g. the FlexiVol elastic screen system lets users safely poke and drag floating holograms [10] [11] – suggesting an interactive future where the torus could be physically "grabbed" in mid-air without harm.

- **Audio-Modulated Mode:** Here the input is sound or music. The torus can act like a three-dimensional **visualizer**, with patterns on its surface reacting to audio frequency and amplitude. A deep bass beat might send a ripple around the ring, or a calm ambient track might make the torus emit gentle radial waves. Using microphones or direct audio feed, the system performs real-time audio analysis (e.g. FFT to detect frequency bands) and links those features to visual effects – analogous to how traditional VJ software creates graphics driven by music, but now in volumetric form.

- **Autonomous "Generative" Mode:** In the absence of external inputs, the torus display can run on an **algorithmic or AI-generated script**. It might cycle through a series of preset patterns (akin to a screensaver), or use generative art algorithms to continuously evolve the colors and textures on the torus. This mode is useful for ambient or decorative usage – e.g. a toroidal sculpture that continually changes like a digital lava lamp. It could also integrate simulated sensor inputs (for example, procedurally varying the display as if it had a heartbeat or breath, to give it a *living* quality even when no user is actively controlling it).

Use-Case Scenarios: To ground these ideas, consider a few example scenarios:

1. *Public Installation:* A large (1–2 meter diameter) holographic torus hovers in a public plaza at night. Spectators walk around and inside the torus's loop. The system runs in **audio-reactive mode**, visualizing a soundscape or music track. As people approach, **motion sensors** trigger the torus to gently orient toward them or change color, inviting interaction. The effect is a **social, collective experience**, much like gathering around a digital campfire – recalling Voxon's vision of volumetric displays as a "digital campfire" for shared viewing [12].

2. *Meditation and Neurofeedback:* A small toroidal display sits on a desk or altar in a dimly lit meditation room. The user wears a **Muse headband (EEG)**. In **EEG-reactive mode**, the torus serves as a real-time mirror of the user's mind – perhaps its glow turns from red to blue as the user relaxes, and its rotation slows to a gentle swirl when deep alpha waves are detected. This could leverage known

neurofeedback techniques used in arts and wellness [13]. For example, an installation called *Mutual Wave Machine* synchronized brain data of participants to reflect shared meditative states [14]; similarly, our torus could pulse in unison with group brainwave coherence during a guided meditation session.

3. *Art-Tech Performance:* An experimental musician outfits themselves with a **wearable holographic pendant** – a mini projector that produces a 10 cm wide torus floating just above their outstretched hand. They also use a **gesture controller** on the other hand. During a live performance, they switch the system between modes: at times the torus visuals modulate by the **music** (audio mode), then they use hand gestures to twist its shape (gesture mode), and finally when they don an EEG cap for one piece, the torus responds to their emotional state (EEG mode). The audience sees a **multi-modal visual instrument**, adding a compelling layer to the performance art.

4. *Educational/Scientific Demo:* In a science museum exhibit about electromagnetism, a holographic torus represents a **magnetic field donut** (like a tokamak plasma or Earth's magnetosphere). Visitors can interact by moving a magnet or coil (sensed via hardware inputs), and the torus will deform or realign to show the field lines. This is more a simulation use-case, but it shows the potential for **interactive volumetric education**. Multiple data sources (magnetic sensor, perhaps an EEG to signify human influence, etc.) could feed into the display (multi-input integration).

These scenarios illustrate the range of applications, from contemplative personal use to collective interactive art. Underlying all is the same technical framework. In the next section, we delve into that framework in detail, breaking it down into four layers: the display hardware, the rendering geometry, the input interfaces, and the control logic. For each, we survey the state of the art and identify candidate technologies for building our holographic torus system.

Technology Layers and Components

To construct a multi-modal, neural-responsive holographic system, we can decompose it into **four key layers**: (1) the **Display Core**, which produces visible light in mid-air (or at least creates the 3D image); (2) the **Geometry & Rendering Layer**, which generates the torus shape and handles the computational aspects of holography or graphics; (3) the **Interaction & Input Layer**, which captures signals from EEG, audio, motion, etc.; and (4) the **Control Layer**, which maps inputs to rendering parameters and orchestrates the real-time behavior. Below, we examine each layer, citing current research prototypes, commercial products, and open technologies relevant to implementing that layer. We also note integration considerations and examples of usage.

Display Core – In-Air Projection Methods

At the heart of the system is the **display mechanism** that makes light appear in mid-air in a toroidal form. Several approaches exist for free-space or volumetric displays, each with pros and cons:

- **Laser Plasma Voxels:** One of the most futuristic methods uses **focused laser pulses to excite points in the air into plasma**, creating tiny "sparks" of light (voxels) floating at desired 3D coordinates. Pioneering work by Japanese researchers (e.g. Burton Inc. and university teams) showed that **femtosecond lasers can generate an arbitrary 3D image by scanning the focal point in space**, essentially 3D printing with points of light [1]. The advantage is true volumetric output visible from any angle without glasses. For example, a laser-plasma system can draw a small spiral or SOS letters in mid-air as a collection of glowing dots [15]. Newer developments addressed safety and interactivity: early versions had nanosecond lasers that *would burn skin* if touched, whereas newer femtosecond laser systems fire ultra-short bursts so that the plasma voxels **don't linger enough to cause burns** and are *safe to touch*, even imparting a slight tactile impulse (the plasma's shockwave) to the finger [16] [17]. These "Fairy Lights" demonstrations even worked under normal indoor lighting, displaying visible floating images that you can poke [18]. For a torus display, a laser-plasma setup could directly draw a glowing ring in the air, point by point. However, current limitations include: small image size (often a few millimeters to centimeters) and high power requirements to generate plasma. It's also inherently monochromatic (typically bluish-white) unless combining multiple lasers with different wavelengths. Nonetheless, ongoing research is advancing color volumetric voxels using multi-photon techniques [19] [20] and ways to enlarge the drawing volume (e.g. using moving lenses or special gases). Safety-wise, **viewers must not look into the laser beam** and often must wear eye protection; the content should be observed only as scattered light [21]. Integrating this into a product means strict safety interlocks.

A demonstration of laser-plasma volumetric display: a tiny fairy-shaped hologram produced by femtosecond laser pulses, floating in mid-air. The plasma voxels are safe to touch – when the user's finger contacts a voxel, they feel a slight impulse due to plasma shockwaves [17]. This shows the potential for interactive, tangible volumetric imagery in free space.

- **Mist or Fog Projection (Mid-Air Screen):** Another class of displays creates an **airborne projection screen** using microscopic particles. The **Heliodisplay** by IO2 Technologies is a classic example: it produces a **thin mist of condensed water vapor** as a projection surface and then uses a standard projector to cast images onto that mist [22] [23]. The result looks like a picture floating in air, visible from front and back (if double-sided projection is used). Importantly, Heliodisplay is *not a true 3D volumetric display* but a 2D free-space screen – essentially a clever way to create an image plane in mid-air. Dark backgrounds appear transparent as the mist only scatters light where there's image content [24], giving a more realistic "hologram" feel than a physical screen. Interactive versions allow mid-air finger tracking on the image [25]. To display a torus with such a system, one could either project a torus shape (appearing as a flat ring) or use multiple screens. However, depth and solidity are limited; you can't walk around to see different sides of the torus – it's like a hovering 2D donut unless combined stereoscopically. Another limitation is **ambient light** – mist screens work best in dim environments since the projected image can be washed out by bright light. They also require refilling water and have some drift in the fog flow. Newer fog display prototypes (and even **water vapor curtains** in theme parks) continue to explore this space. For our modular design, a fog screen could be a quick way to get an atmospheric, semi-3D torus visualization (e.g. projecting a torus that appears to hover, albeit as a planar illusion). Integration concerns: controlling airflow, and possibly providing a dark backdrop or enclosure for contrast.

- **Volumetric LED "Rotor" Displays:** A more mechanical approach to volumetric imagery uses **spinning emissive elements** to fill a volume with light. One commercial product is the **Voxon Photonics VX1**, a box that projects 3D visuals by sweeping a screen surface through space rapidly. Specifically, the VX1 has a **high-speed digital projector** and a **flat screen panel oscillating up and down 15 times per second**. By projecting 2D slices of a 3D model in synchronization with the moving screen, it reconstructs a volume of up to ~18×18×8 cm, viewable from all sides [26] [27]. The stacked slices refresh fast enough (at 30 volumetric frames per second) to appear as a solid 3D object to the human eye [28]. Essentially, it's *like a 3D printer running in reverse at video rates* – the object is “printed” and erased many times a second by the projector’s light instead of plastic [27]. Such volumetric displays truly allow multiple viewers to walk around and see the appropriate side of the object without glasses. For a torus, the Voxon could easily render a rotating wireframe or solid donut shape floating in its volume. The resolution is a function of how many slices and the projector resolution; Voxon’s system creates ~500 million voxels per second via 192 horizontal slices and a 4,000 FPS projection engine [29] [30]. A challenge is scaling up – making the volume much larger is difficult because moving a bigger screen at high speed is mechanically and energetically hard [31]. Voxon hints at solutions for larger volumes, perhaps using multiple projectors or different geometries [31]. Another example of this approach are the inexpensive “hologram fan” displays seen in malls, which spin LED strips to create POV (persistence of vision) images; those are usually planar 2D illusions, but multi-plane versions exist. Integration: devices like Voxon VX1 come with **SDKs and Unity plugins** [32], which would ease connecting our control software to display content. The **trade-offs** for rotor displays are the need for moving parts (wear and maintenance, plus safety – one must not insert hand into the box with a fast oscillating screen, unless innovations like FlexiVol’s elastic surface are used [10]) and typically limited volume size and resolution. Still, for a tabletop torus demo, something like Voxon is one of the most straightforward ready-made solutions.

- **Light Field and Holographic Panels:** Finally, there are **next-gen flat panel displays** that produce a glasses-free 3D image via light field projection. These aren’t volumetric in the sense of having points in space, but they can project a 3D visual within a certain view cone. Examples include the **Looking Glass Factory** displays and the experimental **Light Field Lab’s SolidLight** panels. Looking Glass’s new 27-inch display, for instance, emits **45 to 100 discrete view angles simultaneously** so that multiple people can see 3D content with parallax within a 53° viewing cone [33]. It’s essentially a dense arrangement of projectors or lenticular lenses that present slightly different images to different viewing positions, mimicking a hologram. The result is a “floating” image with up to 16 inches of depth that appears behind and in front of the screen plane [33]. For a torus, one could render it such that it appears to protrude out of the display, giving a convincing half-donut that one can move around to some degree. Light Field Lab’s approach goes for true holographic effect: their **SolidLight panels** are modular 28” tiles each generating **2.5 billion “pixels” that form a holographic volume** when tiled, scaling to wall-sized displays with hundreds of billions of pixels [34]. These panels create *converging wavefronts* that form visible objects in front of the screen with high resolution [35] [34] – purportedly objects can appear *indistinguishable from real* to the naked eye in terms of solidity. For instance, SolidLight could project a torus floating several feet in front of a video wall, viewable by a crowd without wearables. The downside: this technology is cutting-edge and expensive, currently aimed at large installations and not widely available; also the projection distance (how far the object can “escape” the screen) may be on the order of a few feet. Integration is complex, likely requiring custom content pipelines. But conceptually, if cost were no issue, a SolidLight wall could be an amazing way to display a human-sized torus in mid-air as if by magic.

In summary, the Display Core options range from **fully volumetric but small-scale (laser plasma, bubble displays, spinning volumes)** to **large-area but view-cone-limited (light field panels)**. A **modular system** might even combine two: e.g., use a fog screen for a backdrop and laser sparkles for highlights. For our design, we remain flexible – the torus visualization can adapt to the capabilities of the display. Table 1 (Appendix) lists some example display technologies, vendors, and specs. The integration concerns mostly involve ensuring the content generation (Section 4.2) matches the display’s input requirements (e.g. computing interference patterns for a holographic SLM vs. providing 3D mesh frames for Voxon vs. rendering multiview images for Looking Glass). We will address those next.

Geometry & Rendering Layer

Once the hardware is chosen to actually show a mid-air image, we need to generate the *right signals or images* to feed that hardware. This is the role of the **Geometry & Rendering layer**. It encompasses everything from the mathematical description of the torus shape, to the algorithms that convert a 3D model into the appropriate driving signals (voxels, hologram, etc.), to the content engine that can animate and color the shape in real time.

Key components and technologies in this layer include:

- **Digital Holography & SLMs:** If using a true holographic approach (as opposed to direct view or volumetric), a **Spatial Light Modulator (SLM)** can display a computed interference pattern (hologram) that, when illuminated by a laser, reconstructs a 3D image via diffraction. This is how *computer-generated holography (CGH)* works. Software like **VividQ** (a UK startup) and academic frameworks like **OpenHolo** are at the forefront of making real-time CGH feasible. OpenHolo is an open-source library covering hologram generation, reconstruction, and signal processing, aiming to reduce the development effort for holographic devices [36]. Essentially, one provides a 3D model or desired light field, and the library computes the phase pattern to load onto a phase SLM. Historically, CGH computation was extremely intensive – generating one hologram could take hours or days. But recent advances use GPU algorithms (Fast Fourier Transforms, layer-based methods, and even machine learning) to speed this up drastically [37] [38]. In fact, in 2020 an Arm/VividQ collaboration demonstrated **real-time holographic video**: an animated Unity 3D scene converted to hologram on-the-fly, displayed to the researcher's eyes as a genuine 3D image [39]. This was described as *"mind blowing"* to see a Unity scene appearing in mid-air via an SLM and laser [39]. What this means for us is that it's increasingly practical to have a dynamic hologram of a torus – we could render a 3D torus model (which Unity or other engines can easily do) and let a CGH engine produce the phase patterns for an SLM-based holographic display. For example, one could use a reflective phase SLM (like those from Holoeye or Hamamatsu) in a 4f optical setup to project a real 3D torus image floating at some distance. The Geometry layer would incorporate the driver to compute the hologram every frame, potentially utilizing OpenHolo for algorithms (like Fresnel diffraction calculations, point cloud hologram methods, etc.). Challenges include managing the **speckle and diffraction orders** (like the unwanted zero-order light from SLM, which techniques exist to minimize [40]) and the limited viewing angle of holograms. Also, SLMs have finite resolution, so high-definition holograms might only show a portion of the torus at a time or require eye-tracking hologram adjustment in advanced setups [41]. Nonetheless, for at least small-scale prototypes, digital holography is viable and gives the benefit of *true depth cues* (the image actually emits wavefronts from the apparent 3D location, so focus cues are correct – no vergence-accommodation conflict as with 2D images).

- **Real-Time Graphics Engines (Game Engines):** Whether or not we are doing full CGH, we likely will use a standard 3D graphics pipeline (like Unity or Unreal Engine or OpenGL) to generate and manipulate the torus geometry in real time. The torus is a well-defined shape – mathematically, the set of points $(R + r\cos\phi,\; r\sin\phi)$ rotated around an axis, where R is the major radius and r the tube radius. We can easily create a torus mesh in code or load a model. The engine can apply transformations (rotation, scaling "breathing", twisting) and animations (shaders that ripple color, etc.). If our display is something like Voxon or Looking Glass, we might even use their **Unity plugins** to just treat the torus as an object in a 3D scene that their SDK handles for display [32]. If using a custom display, the engine would hand over either a set of slice images (for volumetric) or a point cloud or a hologram computation pipeline. In Unity, for example, we could write a custom renderer that, each frame, takes the torus mesh vertices and computes where to plot voxels or how to modulate an SLM. **Shader programming** can also be used creatively: one could write a fragment shader that encodes a hologram pattern of a torus, or a compute shader that updates the positions of points representing the torus surface. There are even specialized **holographic shader libraries** and research that integrate with Unity's rendering to output phase patterns directly [42]. The benefit of leveraging game engines is rapid development of interactive visuals, integration of effects (lighting, textures on the torus), and easier multi-platform control. We could simulate the torus in VR or AR as a prototyping step (see Roadmap section).

- **Generative Torus Algorithms:** Beyond static geometry, we might want algorithms that generate *variations of a torus*. For example, a **procedural donut** that changes its cross-section or splits into a double torus under certain inputs. Since the torus is a simple parametric shape, one can generate its points via equations: $(x,y,z) = ((R + r\cos\phi)\cos\theta,\; (R + r\cos\phi)\sin\theta,\; r\sin\phi)$ for $0\le\phi,\theta<2\pi$. By modifying these parameters or using trigonometric noise, we can get wavy or evolving toroids. There are also known **sacred geometry patterns** related to the torus (like the "flower of life" mapped on a torus) which one might encode algorithmically. If this project is for an artistic/spiritual bent, one could incorporate those generative designs (for instance, using Lissajous loops on the torus surface). Software such as **TouchDesigner** or shader platforms like **Shadertoy** have examples of rendering torus shapes with swirling textures; those could be ported into our engine. Furthermore, an AI generative model could be trained or used to create torus patterns – though that is exploratory, e.g., using a neural net to generate evolving volumetric light patterns (not unlike how some art installations use GANs to generate abstract visuals).

In implementation terms, the Geometry layer might consist of a **module that holds the current 3D model of the torus** (with any visual attributes like color gradient), and an **update function** that applies input-driven changes (from the Control layer, see Section 4.4) such as "EEG calmness = blue color, rotate speed = f(brain focus)". Then it passes this to a **renderer**: if it's a direct volumetric device, the renderer might produce a set of voxel coordinates/intensities. If it's a holographic device, the renderer computes the hologram (possibly using OpenHolo or VividQ's API if available). If it's a light field panel, the renderer computes the multiview images (the Looking Glass, for example, comes with an SDK where you essentially render your scene from ~45 cameras at once and feed those images). Thus, this layer is where much of the complexity resides, bridging art and science: it ensures the **toroid geometry is faithfully produced in a form that the display hardware can show**.

To give a concrete example: suppose we use a **Voxon volumetric display** for a prototype. The Geometry layer would use Voxon's Unity plugin – we create a Unity scene with a torus mesh and perhaps particle effects swirling around it. The plugin intercepts the 3D scene each frame and drives the projector/layered display accordingly (192 z-slices, etc.) [27]. Now, if we later switch to a **Laser plasma setup**, our geometry layer might instead output a set of point coordinates along the torus and send those to a galvo scanner controlling the laser. The upstream logic (how the torus responds to inputs) remains the same; only the final rendering routine differs. This underscores the value of modular design.

Interaction & Input Layer

This layer encompasses all the **sensors and input devices** through which users (or the environment) can affect the holographic torus. Our system is envisioned as multi-modal, meaning it can take input from brainwave EEG, audio, motion, possibly touch or other biosignals. We review the main categories of inputs and representative technologies for each:

- **EEG Brain-Computer Interfaces:** Reading brain activity in real-time is possible with consumer-grade EEG headsets. Notable examples include **Muse** (InteraXon's headband, 4 channels, popular for meditation apps), **Emotiv** headsets (like the Epoc and Insight, with 14+ channels and software for detecting certain cognitive states), and **OpenBCI** (open-source, DIY BCI kits with 8-16 channel boards and even EEG headcaps like Ultracortex). These devices measure brainwaves (typically the EEG spectrum from delta up to gamma frequencies). Integration-wise, many have SDKs or at least output methods to get the data into a PC. For instance, Emotiv provides a **Cortex API** that streams raw EEG and processed metrics (like "Engagement" or band power) which can be piped into applications [43]. In fact, Emotiv even published a guide on connecting their EEG to **TouchDesigner via OSC** or their API, allowing developers to drive visuals with brain data [44] [43]. OpenBCI similarly has an open data stream (and a Unity plugin). For our purposes, the **EEG interface module** would capture metrics like relaxation level, attention focus, or specific mental commands if using a system that supports that (Emotiv's higher-end API can detect trained "mental commands" like push, pull [45]). Simpler integration might just use the power in certain frequency bands (alpha, beta, etc.) or an EEG-derived meditation score. These values become input parameters to our visualization – e.g., map alpha power to torus color hue. There have been numerous art projects using EEG to influence visuals: for example, the *NeuroCosm* installation translated brainwaves into generative art visuals for a meditative experience [4]. Also, artist **Lia Chavez** built installations where audience EEG directly altered colors and light in the art [46]. Such precedents guide how we might map EEG to visuals (often using smoothing and thresholding to make changes gentle and not erratic). We will need to filter noise, account for signal quality issues (e.g., if a headset loses contact). But overall, hooking up EEG is quite feasible with today's tech; OpenBCI and Muse even have community examples of Unity and TouchDesigner visualizations [47] [48]. Our system might provide a calibration or training phase – for instance, have the user relax and tense up to see the torus respond, thereby learning the mapping.

- **Motion and Gesture Sensing:** Interacting with a hologram naturally begs for **hand gestures or body movement** input. The simplest form is a standard **camera-based motion tracking** – for example, using an RGB-D camera like Microsoft Kinect or an **AI-based pose tracker** from a webcam. However, for fine hand interaction, the **Leap Motion Controller (now Ultraleap)** is a well-established option. It uses stereo infrared cameras to track hand and finger positions with low latency and high precision. A Leap Motion mounted facing the interaction volume could let a user "grab" the torus or pass their hand through it. Indeed, a research prototype called **Pseudo-Hologram with Aerohaptic feedback** used Leap Motion to track hand inside a pyramid display and provide ultrasound haptics [49]. Ultraleap also offers **mid-air haptics** using focused ultrasound to create a tactile sensation on the hand [50] – potentially, that could be added to give feedback when someone touches our torus (though it requires an array of ultrasonic transducers and is a complex addition). For our integration, the Leap Motion has a well-documented SDK (for Unity, C++, etc.), giving the 3D coordinates of each finger and gestures recognized. We can define interactions like: if the user's hand is detected *inside the torus center* and moves, we translate that to moving the torus (like a hoop moving with your hand). A pinch gesture (index and thumb together) could be used as a "selector" to tweak settings (maybe pinching and twisting like a dial to change brightness). The challenge is ensuring the display and sensor align – e.g., the Leap's coordinate system must match the hologram's coordinate space so that when your hand appears to touch the hologram, the Leap also sees your hand at that position. Calibration might be needed. Other gesture options include **glove-based sensors** or wearable rings, but those are less natural. If doing a large public installation, one might use **LiDAR sensors** or floor pressure mats to detect people around the piece (for simpler triggers like change color when people approach). In summary, motion/gesture input is a rich area: we likely incorporate a combination (Leap for precise hand intractability, and maybe a camera for full-body movement if needed). Our input layer should be able to handle multiple streams – e.g., both EEG and Leap Motion at once, which then are fused or used in different modes.

- **Audio and Voice Input:** Sound provides a non-invasive, readily available input channel. We can equip the system with a microphone or line-in from an audio source. **Audio analysis** can range from basic volume detection to full spectral analysis. A common approach is computing an FFT of the audio to get frequency bands (bass, mid, treble amplitudes), or detecting beats and tempo. Many tools exist for this (e.g. the **Max/MSP** or **Pure Data** environments excel at feeding audio to visuals; game engines also have audio analysis capabilities). The idea is to make the torus "dance" with music: for instance, *rotate faster or expand in radius with the beat*, *map low frequencies to an inner color glow and high frequencies to outer ring sparkles*, etc. If the system has a microphone, it could also respond to **claps or voice commands** (with some speech recognition if needed). A voice command could switch modes ("calm mode on" to use EEG input, for example), although integrating full voice recognition might be beyond scope unless using existing APIs. In general, audio reactivity is well-trodden: many visual installations and LED sculptures sync to music. We simply need to incorporate an audio analysis library. If using something like TouchDesigner or even Ableton Live, there are ways to route audio into our visualization via MIDI or OSC. For instance, a musician could send a MIDI CC that the torus interprets as "pulse now". Our system's control layer will mediate this (Section 4.4). The good news is this is a relatively straightforward layer – the main considerations are filtering noise (if using a mic in a noisy area) and latency (ensuring the visuals are tightly in sync with audio beats).

- **Other Biosignals and Inputs:** Beyond EEG, one could include **heart rate (ECG or PPG)**, **galvanic skin response (GSR)**, or breathing rate (via a chest strap) to make the torus respond to the user's physiological state. For example, a slow pulsing torus could be tied to the user's heartbeat, creating a biofeedback loop where seeing it might further calm or engage them. Devices like smartwatches or Polar heart rate straps can output data; or a simple finger pulse sensor could be connected. GSR could correlate with stress, and one might visualize that as a "spikiness" of the torus surface. There are art installations using these signals; for instance, some biofeedback art projects use **multi-sensor fusion** to drive visuals (EEG + heart rate for emotional state). If pursuing a wellness angle, this could be compelling. Our modular design could allow plugging in such sensors through a standard interface (possibly via an Arduino or BLE device streaming the values).

Additionally, **touch input** could be considered if the display surface is tangible (for example, the **FlexiVol system** with elastic diffuser can detect where the user touches the screen because it deforms [11]). But in mid-air, touch input per se is just the gesture sensing as above, unless using something like Ultraleap's ultrasonic haptics to also *output* touch feeling.

Mobile App / Dashboard: While not a sensor, a **smartphone or PC dashboard** can be part of the input layer to give users manual control and configuration. We could have a companion app that sliders for parameters (color, mode switching) or to select presets. This is practical for debugging and also for performance scenarios where an operator might want to override or schedule changes. Communication could be via Wi-Fi (app sends OSC or HTTP to the system) or a simple UI on the PC running the show. The app could also relay data from online sources (imagine the torus reacting to real-time stock market data or collective social media sentiment in some conceptual art usage – showing the flexibility of input).

In summary, the Interaction/Input Layer is the **eyes, ears, and skin of the system** – it senses the user and world. For integration, we plan to utilize existing APIs: e.g., *Muse Direct* or *BrainFlow* for EEG, *Ultraleap SDK* for hand tracking, *PortAudio* or an engine's audio input for sound, and possibly *OSC/MIDI* for any external controllers. Each input will be standardized into a set of **control variables** (like a dictionary: `{"EEG_alpha": 0.6, "EEG_beta": 0.2, "hand_position": (x,y,z), "music_bass_level": 0.8, ...}` updated in real time). These then feed the next layer.

Control Layer

Sitting between the inputs and the rendering is the **Control Layer**, essentially the software "brain" or middleware that **maps signals to visuals**. This layer handles logic, data processing, and synchronization to achieve a cohesive real-time experience. Key aspects include:

- **Signal Processing and Mapping:** Raw inputs often need to be processed (e.g., filtering EEG noise, smoothing motion jitters, normalizing audio levels) and then mapped to visual parameters. The control layer will likely maintain a set of **mappings or rules**. For instance:
- Map *EEG meditation level* (0 to 1 value) to torus color along a blue-red gradient.
- Map *EEG focus level* to rotation speed of the torus (higher focus = faster spin).
- Map *heart rate* to a subtle pulsation (scaling the torus radius in and out periodically at that BPM).
- Map *audio bass* to torus minor radius modulation (the torus "thickness" bounces with the beat) and *audio treble* to a flicker or particle effect on the torus surface.
- Map *hand distance* to overall torus size or position (bringing hand closer could shrink the torus as if you're compressing it).

These mappings can be linear or non-linear; we might use ease-in curves so that small changes don't jitter the output. Also, there could be modes where different mappings apply (like a preset for meditation vs. a preset for performance). The design could include an easily configurable mapping (maybe a JSON or a visual node-based logic if using something like Max or TouchDesigner in tandem).

- **Middleware/Frameworks:** Instead of coding all from scratch, we might leverage existing **visual programming environments** known for interactive installations. **TouchDesigner** is a popular node-based system that can integrate inputs (including EEG via OSC as Emotiv's guide suggests [51]) and generate graphics or control Unity via OSC/MIDI. It's possible one could build the whole control logic in TouchDesigner, which then sends commands to Unity or directly to hardware. **Max/MSP or Pure Data** similarly could serve as the logic engine, especially for audio reactive aspects (Max has objects for FFT, filters, etc.). However, using too many tools might complicate the pipeline; a simpler approach is to implement control logic in the main application (Unity C# scripts or Python scripts if using an environment). There are also specialized libraries like **Processing or OpenFrameworks** that many artists use for EEG and interactive art – these could be alternatives if not using Unity.

Another angle is using **game engine scripting**: e.g., in Unity, one can write a script that reads variables and updates the material properties of the torus object accordingly each frame. Unity also supports custom timelines or animations that could be triggered by inputs. So one could create, say, an animation that makes the torus explode into particles, and trigger that when a certain EEG threshold is crossed (like the user gets too excited – a playful feedback).

- **AI/ML Integration:** We might incorporate machine learning to interpret inputs or even to generate visuals. For example, a **classification model** could take EEG + other biometrics and classify the user's mental/emotional state (like "stressed", "calm", "focused"). This classification could then switch the torus into different predefined modes or colors. Such classification could be done with a simple rule-based approach or training an ML model on data. Emotiv's API internally classifies "Performance Metrics" like engagement, stress, etc., which we might use directly [43]. If more custom, we could use a lightweight neural net or even a heuristic (e.g., high beta and GSR = stress).

On the generative side, one could imagine using an **AI generative model** to decide the visual pattern. For instance, one project mentioned in the Neuroelectrics blog used EEG-derived emotion (valence and arousal) and then **leveraged ChatGPT to produce images and haikus from that** [52] – a creative multi-step pipeline. In our case, perhaps an ML model could generate a unique torus texture or animation style based on live input, though that's ambitious in real-time. More feasible is using AI for **predictive smoothing** (predict the next likely user state to pre-emptively adjust visuals) or for **high-level control** (a "mood" classifier selects one of several artistic shaders for the torus).

- **State Management and Mode Switching:** The control layer should manage different operational modes (as described earlier). It decides when to be in EEG mode vs. audio mode if multiple are available. This could be manually toggled by the user via an app or automatically based on context (e.g., if an EEG device is connected and user is wearing it, prioritize EEG input). It also could blend modes – for example, if both EEG and music are active, perhaps the torus primarily follows EEG but adds music-driven pulses as a secondary effect. Designing how these inputs layer or override each other is a control logic challenge. We can incorporate a **priority system** or weight blending system for inputs. This might be exposed in a UI for an operator to tweak.

- **Timing and Synchronization:** With multiple inputs and outputs, keeping everything in sync is crucial. The control layer likely runs on a loop (possibly the frame update loop of the visualization at e.g. 60 FPS). It needs to gather the latest data from all inputs (which might come at different rates – EEG could be e.g. 256 Hz sampling but we might reduce to a rate of, say, 10 updates per second after smoothing; audio might be continuous but we get beat events, etc.). It then applies logic and updates the visuals. Because displays like Voxon or SLMs also have their own refresh considerations, we might need to lock to a certain framerate or use time-stamps. For instance, if the volumetric display runs at 30 fps, and our input loop is 120 fps, we should throttle visual updates to not exceed what can be displayed. Similarly, **latency** should be minimized for responsive feel (we don't want a noticeable lag between raising one's hand and the torus reacting). Using event-driven updates (like trigger on beat detect) helps immediate response.

In integrating everything, frameworks like Unity help because they are essentially game loops that handle timing. Alternatively, a custom Python app with threads for each sensor might be used, but thread-safety and sync become complicated. Most likely, a single application (Unity or TouchDesigner or custom C++ app using something like OpenFrameworks) will ingest all inputs and produce output.

To illustrate, imagine the control flow in a meditation session: The EEG input comes in, we smooth it over a 5-second window to get a stable "relaxation score". The control logic maps that to color, slowly transitioning the torus color. At the same time, ambient music is playing; a beat triggers a slight pulse in brightness. The user reaches out to touch the torus; the Leap Motion detects intrusion and the control logic momentarily increases the torus's transparency or deforms it as if disturbed, then returns it to normal. All these concurrent influences are summed up in the control layer, which ensures a harmonious output rather than chaotic. The **design of these interactions** would be refined through iterative testing – e.g., making sure EEG changes aren't so fast as to be distracting, or that one input modality doesn't overwhelm the others unexpectedly.

In summary, the Control Layer is the **integration hub**. It requires a balance of engineering (signal processing, possibly multithreading) and artistic design (what mappings feel intuitive or meaningful). By leveraging existing tools (Unity's update loop, TouchDesigner's nodal logic, APIs from device vendors), we can implement this layer in a robust way. The end result is a sort of **central nervous system** for the holographic torus installation, taking raw stimuli and generating a coordinated visual response.

With the layers defined, we can now consider some concrete integration case studies, showing how all these pieces come together in practice.

Integration Scenarios

To demonstrate how the modular components work in concert, we detail a few **example integration scenarios**. These scenarios will tie together specific input modalities with the torus display output, highlighting the flow from sensor to hologram. Each example assumes a certain hardware setup just for illustration.

Example 1: EEG → Color Modulation on a Volumetric Donut.
Setup: A user wears an **Emotiv Insight EEG headband** feeding data to a PC. The display is a **Voxon volumetric unit** showing a 3D torus hovering above its surface.
Integration: The EEG interface module reads the user's brainwaves and computes two metrics: "relaxation" (based on alpha/theta power) and "arousal" (based on beta/gamma). The Control layer linearly maps relaxation to a color gradient from cool blue (when relaxed) to fiery orange (when tense). Arousal is mapped to the torus's rotation speed (calmer brain = slower rotation). The user sits and practices breathing to calm themselves. As they do, the system detects rising alpha waves; the control logic gradually shifts the torus from orange to a soothing blue hue, and its spinning slows to a gentle turn. The volumetric display updates at 30 fps, showing the color changes smoothly. Observers see a clear biofeedback loop: when the user opens their eyes or thinks hard (arousal spike), the torus immediately brightens and spins up. If the user returns to a meditative focus, the torus dims and returns to blue, almost like a live mood ring. This scenario demonstrates a **one-to-one mapping** of EEG to a visual parameter, with volumetric output. It builds on prior art where EEG data drives art lighting – here it's in free space. A connected source example is the *Mutual Waves Machine*, which used collective EEG to drive a shared visual; in our case it's single-user but similarly reflects neural state in an immersive way [14].

Example 2: Audio Reactivity in a Mist-Based Projection Vortex.
Setup: In a small venue, an **ultrasonic mist projector** (similar to Heliodisplay) creates a thin curtain of fog mid-air. A standard projector beams on it. We display a graphical torus (rendered in 2D but giving an illusion of 3D on the mist). The system has a **microphone** or line-in from a DJ's audio mixer.
Integration: The audio input module analyzes the music in real time. The Control layer uses a beat detection algorithm to find the BPM and beat triggers. It also splits the audio spectrum: bass vs mid vs treble levels.

These control different aspects of the torus graphic: - Bass drum hits cause the torus to **pulse in size** (scaling up 5% briefly). - The dominant melody (mid-range) controls the torus's **color** along a preset palette (e.g., the harmony intensity shifts it between green and purple). - High-frequency sounds (hi-hats, etc.) spawn little particle sparks that orbit the torus. The projector displays this in the fog: to viewers, it looks like a **glowing donut-shaped equalizer** that throbs with the music. Since the Heliodisplay isn't volumetric, it's more a planar visual, but appearing in mid-air gives it a captivating presence. This can be enhanced by using two projectors on either side to give a slight stereoscopic depth or simply to allow viewing from both front/back. The synchronization is critical – the Control layer ensures each beat from the music aligns with the visual pulse (latency tuned perhaps via calibration). This demonstrates a classic **audio-visualizer scenario**, now embodied in a floating torus. Many VJ software outputs to 2D screens; here we adapt it to a mid-air screen. The mapping can be adjusted on the fly by the DJ or system: e.g., during a calm song, the control logic might switch to a slower, ambient visualization (maybe the torus slowly undulates to the waveform shape). This example could also incorporate **voice**: if a narrator speaks, the system might switch to voice-reactive mode (perhaps the torus rings ripple as if water when words are spoken, with amplitude driving ripple intensity). The versatility of the Control layer allows different audio profiles for different content.

Example 3: Group Meditation – Averaged Brain State → Torus Pulse.
Setup: A **circle of users** (say 5 people) sit around a **large holographic torus installation** in a dim room. Each wears a simple EEG headband (Muse 2) transmitting data wirelessly. The display could be a **Looking Glass 3D panel** or a small **holographic projector** producing an image of a torus in the center that all can see.
Integration: The system collects EEG data from all participants. The Interaction layer either runs multiple EEG threads or uses a cloud service if available (for Muse, an aggregator via OSC could be used). The Control layer computes a combined metric – for example, the **synchronization or average of their alpha rhythms**. Research has shown that when people meditate together, certain brainwave patterns can become synchronized; one art project, *Mutual Waves Machine*, visualized the level of brainwave synchrony between two people [14]. Inspired by that, our system calculates a "group calmness" value (high if most members are calm). This value drives the **torus's pulsation and brightness**. At first, the group might be distracted – their brainwaves are varied, and the metric is low, so the torus might flicker or show a fragmented form. As they settle into meditation and perhaps their breathing and brainwaves align, the metric rises; the torus correspondingly becomes more coherent – maybe it **glows steadily and expands/contacts in a slow rhythmic pulse** resembling a breathing pattern. The torus effectively acts as a **real-time collective biofeedback** object. If one person's mind wanders (their EEG spikes beta), the system could dim slightly or wobble, indicating the disturbance, which might subconsciously encourage that person to refocus for the sake of the group. Technologically, averaging signals is straightforward; one just needs to ensure all devices' data are time-aligned (perhaps using timestamps or by starting simultaneously). The display being a Looking Glass panel allows multiple viewers around it to all see the 3D torus (within its 50° view cone) without glasses [33]. Alternatively, a true 360° volumetric display (like a big Perspex spinning volume) could let everyone sit around and see it from their side. The group scenario emphasizes the **social and even spiritual dimension** of the torus system – it becomes a focal point for group coherence. It merges technology with a kind of ritual experience, which is in line with the concept of torus as a "harmonizing" geometry. (Appendix lists some open-source libraries that could help with multi-EEG handling and coherence calculation.)

Example 4: Wearable Mini-Donut (Holographic Pendant Display).
Setup: A prototype **wearable device** about the size of a chunky necklace pendant. It contains a small

battery, a micro projector or LED array, and possibly a half-mirror to create a floating image just above the pendant. (For instance, one could use a **Pepper's Ghost illusion** or a micro LED fan display in a clear dome.) The user also has a smartphone linked to the pendant via Bluetooth for control and input.
Integration: The wearable's display is limited (perhaps it can show a 3 cm torus that appears above the pendant). The smartphone acts as the input hub: it might use the phone's **motion sensors** (so when the user moves or dances, the torus reacts), and the phone's **microphone** (to react to ambient sound or the user's voice). The phone could also pick up the user's heart rate via a smartwatch or its own camera (PPG). The Control app on the phone processes these – say the user's steps (detected by accelerometer) cause the torus to bounce, and if the user speaks, the torus oscillates like a waveform. The phone then sends commands to the pendant (which might just display pre-defined frames or patterns based on those commands, to keep the pendant electronics simple). This scenario is like a **personal holographic jewelry piece**. It's speculative, but rooted in trends: we have LED necklaces and small POV displays commercially; a next step is making them true 3D-looking. The torus shape is ideal for a pendant (it looks like a halo or mystical amulet when floating above one's chest). The main challenge is miniaturization and brightness in varied lighting. The integration shows the modularity: because we separated layers, we can offload heavy computation to the phone (Geometry & Control layers on phone, Display is the pendant). If the user switches modes via the phone app (taps "music reactive" vs "biofeedback"), the system adjusts accordingly. For instance, in a club the user might enable music mode so their wearable torus pulses to the beat (cool fashion statement), whereas on a walk they might enable heart mode to have it gently pulsing to their heartbeat (a personal mindfulness aid). This wearable case might be a future roadmap item, but it illustrates that the **same architecture scales down** to portable form, using different hardware optimized for power and size. It also underscores the need for efficient power management (battery for projector, etc.) and possibly **Edge AI on phone** to do any needed classification (taking advantage of phone's CPU/GPU).

These examples cover individual, group, fixed installation, and wearable uses – demonstrating the flexibility of the system. Each required selecting appropriate tech for display and input, but the core ideas carried through. They also reveal various **technical challenges**, which we will discuss next along with potential solutions or workarounds encountered in these scenarios.

Technical Challenges & Workarounds

Building a modular holographic torus system is an ambitious undertaking, and several technical hurdles must be addressed. We outline the major challenges anticipated, along with possible workarounds or mitigations identified from research and engineering practice:

- **Power and Portability:** High-end holographic displays (laser systems, volumetric projectors) often require significant power and are not easily portable. For instance, the laser-plasma setup needs a high-power laser and focusing system, usually benchtop and power-hungry. **Wearable or mobile implementations** are therefore challenging. A Voxon VX1 is a small table device but still requires mains power and is ~$10,000 [53] [26]. Workaround: start with **desktop and installation scenarios** where power is available. If portability is needed (Example 4's pendant), use low-power displays like microLEDs or leverage a smartphone's processing (as we did) to avoid heavy onboard computing. Battery technology improvements could allow short-term mobile use (e.g., a backpack to carry batteries for an outdoor installation). Also, consider **modular power design** – the system could run on AC when available, but have a battery buffer for brief untethered operation. Optimizing software to reduce unnecessary computations (especially for battery-bound devices) will extend life. As tech progresses, things like **LED-based holographic glasses** or solid-state AR (like Microsoft HoloLens) could be incorporated to simulate the effect with less power.

- **Ambient Light and Contrast:** Many display methods struggle in bright ambient conditions. A mist screen in daylight is almost invisible; laser-plasma dots are visible outdoors only if very intense or if using special gases [54]. The IBTimes example showed that even with femtosecond lasers, broad daylight viewing was achieved for small "fairy" voxels [18], but that's a controlled demo. For a general installation, **lighting control** is important. Workaround: If possible, present the holographic torus in a **dim or enclosed space** to improve contrast (the Problem Solutions blog notes darkened environment is often needed for best image [55]). For outdoor or bright settings, one could boost brightness: e.g., **Xenon gas** was used in experiments to make plasma voxels brighter under ambient light [54]. That suggests filling the interaction volume with a medium that enhances visibility (like mist, or even nanoparticles in air for scattering). Another approach is to use **LED volumetric displays** which can be extremely bright (some LED fans are visible in daylight). However, with brightness comes power and safety issues (like glaring lights). Possibly a combination of **physical shading** (an art structure or tent around the installation) plus high brightness is the pragmatic solution. For AR-based prototypes, the issue is similar – AR glasses often wash out in sun, so tinted visors or higher luminance projectors are needed.

- **Image Stability and Alignment:** In volumetric displays with moving parts (oscillating screens, spinning LEDs), **mechanical stability** is vital. Vibrations or misalignment can cause the image to blur or appear doubled. If, for example, the Voxon's vibrating screen is disturbed by someone bumping the device, the reconstruction might break until it re-syncs. Similarly, any multi-projector setup (like Light Field Lab's tiled panels) must keep panels precisely aligned. Workaround: include **feedback systems or calibration routines**. The Voxon case likely uses sensors to track the moving screen's position each moment and sync the projection [56] – such closed-loop control ensures stability. Our system should calibrate upon start: e.g., if using an SLM hologram, align the optical path so the torus appears at the intended spot (perhaps using a reference marker that the system adjusts to). For mechanical volumetric units, mounting them on a steady base and possibly using damping to reduce external vibrations will help. In software, we can implement an **auto-calibration step** where the system displays a test pattern and uses a camera or sensor to verify it looks correct (this is more applicable if we integrate a camera for feedback).

- **Safety (Laser & General):** Safety cannot be overstated, especially for laser-based displays. Direct laser exposure to eyes can be hazardous; even looking at bright volumetric voxels too closely might pose a risk (though in plasma, the emitted light is like tiny fireworks – not coherent, but still bright). Also, if using rotating parts, there's a physical injury risk (the **"lose a finger"** issue with old volumetric displays was literally because a fast spinning panel could cut you [10]). Solutions:

- For lasers: Use **femtosecond lasers with fast scanning** as demonstrated, so the voxels are eye-safe to touch [57]. But still enforce that no one looks into the beam source. Ideally enclose the beam path and only allow the floating image to be accessible. Provide **laser safety goggles** if appropriate (though that defeats easy viewing, so better design the system so they're not needed – e.g., IR lasers with frequency-doubled visible output might require blocking invisible IR, etc.). The IBTimes piece recommends viewers wear IR glasses during development [21] – so during testing that's a must.
- For moving parts: If using an oscillating screen or fan, put it in a **case or dome** so people cannot accidentally stick their hand into it. The FlexiVol project solved the interaction issue by making the

surface **elastic** so it won't injure and doesn't break if touched [58] [11]. We can adopt similar ideas: e.g., a flexible membrane as a projection surface that won't harm fingers. Or use **ultrasound haptics** to simulate touch without actual contact surfaces (Ultraleap's approach projects tactile feeling in air [59], meaning one could feel the torus without needing to physically touch hardware).
- Electrical safety: Ensure all gear is properly insulated. Wearables should not pose any shock or heat risk to the user – using low voltage DC and proper thermal management is necessary (LED and laser components can get hot).

- **Standards Compliance:** Follow laser safety standards (like IEC 60825) for classification of the device (aim for Class 1 or 2 laser product if possible). For any installation, have emergency cutoff switches, and perhaps sensors to detect if someone approaches dangerously close to a high-power projector (and then dim or shutter it).

- **Multi-Input Synchronization:** Juggling data from EEG, audio, motion simultaneously can lead to **timing conflicts or jitter**. For example, if EEG updates at 10 Hz and our display is 60 Hz, naive implementation might cause a stuttering in the parameter updates. Workaround: use interpolation and buffering. The control layer can interpolate slow signals (like EEG) across frames for smooth changes. Also employing a **common clock or timestamp** for sensor data helps in fusing signals meaningfully (for instance, tag EEG samples with time, and if we also have an event like a button press, we can correlate them if needed). In software, adopting a framework that supports **real-time scheduling** or high-frequency loop (like game engines do) can reduce variability. One might also designate one input as master timing (e.g., audio beat events could drive a keyframe moment where visuals align).

Another aspect is ensuring that if multiple people use it sequentially or concurrently, the system properly resets states. If the torus was calibrated to one person's EEG baseline, when another wears the headset, we need to reset baseline. These operational details mean the control software should have modes for calibration, possibly automatically detect when devices change, etc.

- **Eye Strain and Visual Coherence:** While volumetric displays avoid the vergence-accommodation mismatch of 2D 3D displays, they introduce other issues: limited resolution can cause eye strain as the eye tries to focus on a voxel that's actually a small light point. Also, if the image has flicker (low frame rate or unsynchronized update), that can be fatiguing or cause motion sickness for some. Ensuring the **visual output is high quality and comfortable** is a challenge. Solutions include:
- Keep the **refresh rate high** (aim for 30+ fps volumetric, and avoid flicker in brightness modulation). The Voxon's 30 fps is just adequate; some people might notice flicker at peripheral vision. Higher-end should be 60 fps if possible. Some light field displays run at 60 fps for each perspective set [60].
- **Anti-aliasing and smoothing:** For instance, a low-res volume might show a jagged torus. Using scattering or slightly defocusing can blur it enough to look smoother (trading sharpness for coherence). Some research has looked at *occlusion* in volumetric displays – because one drawback is that all voxels are glowing points, you can always see "through" an object, which is unnatural and can confuse depth perception. New techniques like multi-color excitation or temporal multiplexing can simulate occlusion to some extent [61] [62]. In our case, the torus shape being relatively thin might not suffer heavy occlusion issues, but if it were a solid object, the back side would normally be hidden. Without occlusion, the brain may perceive it oddly. We mitigate by either not showing fully opaque solids or by using brightness to attenuate rear parts (for example, programmatically dim voxels that represent the far side of the torus as it would be in shadow).

- **User adaptation:** As with any novel display, some user orientation is helpful. We might start the experience gently, giving eyes time to adjust to the type of imagery. If using AR goggles in prototypes, limit session lengths to avoid eye fatigue.

- **Software Complexity and Integration Issues:** With many components (EEG software, graphics engine, custom hardware APIs), there's a risk of integration bugs and system crashes. We need the whole system to run reliably especially in public demos. This is more a project management challenge: heavy testing, using robust libraries (OpenBCI's drivers, Emotiv's SDK, etc., which are fairly well-tested), and having **fallback modes**. For instance, if EEG data fails mid-demo, the system could automatically switch to audio-reactive mode or a pre-set animation rather than freezing. Logging and diagnostics are important too, to troubleshoot sensor connections.

- **Cost and Accessibility:** High-end holographic tech is expensive and specialized. If our goal is to allow artists or researchers to replicate or build upon this, we should consider **low-cost alternatives** in the design. For example, as a fallback, an AR headset showing a torus is far cheaper than a SolidLight wall. Open-source software like OpenHolo helps avoid pricey software licenses. We might include multiple options in documentation (use a $300 Looking Glass Portrait for a small demo, or scale up to a $Xk Voxon for full volumetric effect). By designing modularly, one can start with what they can afford and upgrade pieces later. This democratizes experimentation with the system concept.

Each of these challenges – from practical (power, cost) to perceptual (eye strain) – has at least a partial workaround. Importantly, ongoing research continuously improves these aspects (for instance, every year brings better AR displays, more efficient algorithms, etc.). By acknowledging the constraints and incorporating mitigations into our design (like safety interlocks, calibration routines, and flexible mapping logic), we can increase the robustness and user-friendliness of the holographic torus system.

Next, we will outline a possible roadmap for prototyping this system, starting with simple setups and gradually increasing complexity, which inherently addresses challenges step-by-step (learning from the small scale to inform the large scale).

Roadmap to Prototyping

Developing the full envisioned system is a complex project. It's wise to proceed in **iterative stages**, building and testing each aspect in increasing scale and integration. Here we propose a roadmap of prototyping steps, each focusing on certain subsystems:

Stage 1: Low-Fidelity Prototype (AR + EEG Mockup).
Goal: Validate the concept of EEG-driven torus visualization using readily available tech, without needing any actual holographic display hardware.
Setup: Use an **Augmented Reality (AR) headset or AR smartphone app** to render a virtual torus in the user's environment. For example, use the Microsoft HoloLens or Magic Leap to place a floating torus model in front of the user. Alternatively, use AR on a phone (ARKit/ARCore) to overlay a torus in the camera view. Connect a **consumer EEG device** (like Muse 2) to a laptop.
Implementation: The AR device displays a 3D torus with some default animation. The EEG is read on the laptop (perhaps using Muse's OSC stream or the BrainFlow library) and a simple processing script extracts one metric (like alpha power). This metric is sent to the AR app (maybe via network or if using HoloLens, an

app on it reads directly from a shared source). The AR torus then changes color or spin based on the EEG. This stage tests end-to-end: user thinks -> data -> visual change, but in a simulated hologram. AR is great because it inherently solves 3D viewing for one person – the HoloLens hologram of a torus will appear spatially in front of them with correct perspective. Of course, it's only visible to the wearer. But as a test, it avoids needing specialized displays. We also test **basic control mapping** and user experience: Does the color feedback indeed correlate with relaxation? Is it noticeable and smooth? We might find, for instance, that the mapping needs a time lag to not be jumpy. Or we discover the user finds it intuitive (or not). Adjust accordingly.

Outcome: A working AR demo of a neural-responsive torus. This confirms the viability of the software pipeline and gives an early feel for the concept's impact. If AR glasses users report it is engaging (like seeing a halo that calms when they do), that's a green light. We also gain insight into calibration (perhaps each user needed a baseline recording for EEG), which we can incorporate into UI.

Stage 2: Medium-Scale Prototype (Desktop Torus Projection with SLM or Light Field Display).

Goal: Implement a real mid-air or 3D display of a torus on a small scale, integrating one real input modality (if not already done).

Setup A: If available, use a **desktop holographic display**. This could be a Looking Glass Portrait (a small 7.9″ light field display) or a similar device. Or, if we have access to a lab, use a **phase SLM with a laser** to project a simple hologram of a torus floating above an optical bench.

Setup B: Alternatively, use a **Voxon Photonics VX1** or DIY a simple volumetric display (e.g., a rotating mirror with projection – there are hobby projects that spin a small mirror on a motor and use a pico projector to create a 3D image).

For input, we can incorporate **one primary input** besides any demo triggers. For instance, use **Leap Motion for gesture control** in this stage, since we did EEG in Stage 1.

Implementation: In Setup A (Looking Glass): Develop a Unity application that renders the torus and outputs to the Looking Glass using their SDK. We program some interaction: using keyboard or an attached sensor to change the torus. For input, say we use Leap Motion – integrate its Unity SDK so hand movements can tilt or poke the torus. We might also play with an **audio-reactive shader** on the torus for variety (e.g., feed microphone input to an animation). In Setup B (SLM): Use OpenHolo or a known CGH algorithm to generate a hologram of a 3D ring. We might start with static (just verify the SLM can make a torus hologram appear). Then connect a simple control loop to maybe rotate it (by updating the CGH) or change size. If we incorporate an input, perhaps link keyboard keys or a MIDI knob to refocusing the hologram (just as proof of concept control).

Outcome: We now have an actual **holographic/volumetric torus visible in real life** at small scale. This stage will teach us technical integration of display hardware: e.g., calibrating an SLM for best image, or learning the quirks of the Looking Glass (like needing to maintain 60 FPS and how depth works). If using Voxon, we'll gain experience with their Unity plugin and limitations (like resolution 100×100×100 voxels, etc.). We should also evaluate the **visual quality**: Is the torus recognizable and pleasant to view? Do we need to refine its rendering (maybe add more vertices for smoothness, or adjust brightness distribution)? Any alignment issues? Solve those here. We also test another input mode – the Leap – see how intuitive it is to move a hologram with your hand. Perhaps we find a slight lag with Leap data; we could tune that. This stage ensures that by now we have *both* a working input pipeline and a working output pipeline, just not all modalities at once.

Stage 3: Large-Scale Installation Prototype (Multi-Modal Integration).

Goal: Build a full-scale version suitable for a **small public demo**, integrating multiple input modes and outputting to a larger display system.

Setup: Choose the primary display method for the installation. If budget allows, perhaps a **1m^3 volumetric display** (though none commercial at that size yet – one could attempt an array of smaller ones or a projection-based volume). More realistically, maybe use a **rear-projection fog screen or scrim** to create a hologram-like effect about 1m in size. Or use multiple **high-lumen projectors and a transparent 3D printed torus sculpture** to create a hologram illusion (like projecting onto a spinning glass structure – an art hack). Another possibility: coordinate a set of **drones with LEDs** forming a floating torus (there are drone light shows that form 3D shapes, but that's quite complex – we mention but likely not our route!).
Anyway, for integration, plan to incorporate at least **two sensor modalities simultaneously** – e.g., EEG + audio, or gesture + audio, etc., to demonstrate multi-modal. Possibly have two modes the audience can try: one person puts on EEG and "mind-controls" the torus, then switch to a mode where someone else waves hands to sculpt it in the air.
Implementation: This will involve robust software architecture: possibly a combination of Unity for graphics + TouchDesigner or Max for sensor fusion, depending on what we found easier. We ensure the control layer can handle toggling modes and blending inputs. We add a simple **UI or control panel** to start/stop sessions, calibrate EEG, switch presets. For output, we focus on making it **visible and cool**: perhaps employing **lighting and sound in the room** that complement the hologram (e.g., when the torus pulses, surround sound or a subwoofer emphasizes it – immersive effect). We also set up safety barriers if needed (like if using lasers or a physical rotor).
Once built, invite a test audience (colleagues or small public) to experience it. Gather feedback – do they understand what to do? Are the responses of the torus clear and engaging? This stage might reveal if the concept truly resonates: e.g., people might say the torus reacting to their calm made them feel more aware of their state (success!), or maybe they found it confusing that multiple people's inputs were mixed. That feedback is gold for refining interaction design.
Outcome: A functioning **multi-modal holographic torus installation**, albeit maybe not the ultimate hardware envisioned, but something that can be physically demonstrated. This serves as an MVP (minimum viable product) of the concept. From here, one could either present it as an artwork/tech demo or use it to attract support for further development.

Stage 4: Toward Wearables and Distributed Systems.
Goal: Explore miniaturization (wearable) and multi-node setups (networked displays).
After a large install, the next steps could diverge: - *Wearable Prototype:* Try making a **portable torus device**. Perhaps use a small round LED fan display (those cheap "hologram fan" things) and see if it can display a torus convincingly. Combine it with a smartphone EEG (like the Muse 2 which is portable) to replicate the personal neurofeedback use-case. This may involve writing a phone app that both reads EEG and controls the fan display (some of those fans have an SDK or accept video input). This will be very experimental, but it might result in the first "holographic necklace" as discussed. Test it in different environments (indoors, outdoors at night) to gauge viability. - *Networked Multi-User:* Another angle is to have **multiple torus displays in different locations** sharing data. For example, two installations in two cities where each shows the combined inputs of both sets of participants – a kind of telepathic link art piece. That requires robust networking and time-sync of data. We could simulate this by having two computers in lab representing two sites, each with an AR or small volumetric unit, and stream data between. The purpose is to see how scaling to multiple units might work and any latency issues.

Outcome: These are stretch goals. The wearable test might be rudimentary (maybe it doesn't look great, but one learns what needs improvement, like resolution or brightness on small form factor). The multi-node test could show how to architect a cloud or peer-to-peer system for the torus signals (maybe using MQTT or similar IoT protocol for sending biofeedback data).

Stage 5: User Studies and Iteration:
If the aim is for meditative or healing use, conducting a **formal user study** would be valuable. For instance, have a group use the EEG-torus system daily for a week and record their subjective relaxation or any biometrics (heart rate changes, etc.). Or compare group meditation with vs. without the torus biofeedback to see if it enhances synchronization. Their feedback can guide tweaks: maybe the torus needed a calmer color palette, or the feedback loop was too direct and caused anxiety when they lost focus (in which case we might slow the response so it's gentler). If it's more for art/exhibition, gather audience feedback on what they found meaningful or not. Iterate the design – could be in software (change mappings), hardware (maybe choose a different color laser for visibility, etc.), or presentation (the narrative or instructions given to users).

Through these stages, each component (display tech, EEG integration, gesture control, audio reactivity) is progressively validated and refined. Risks are mitigated early (e.g., if EEG proves too noisy in Stage 1, we know to improve filtering or electrode contact before trying to use it in a big show). By Stage 3, we'd have a decent chance of a polished system.

The roadmap also implicitly builds a **knowledge base and toolkit**: by the end, we'll have libraries for handling inputs, code for rendering torus on various devices, and an understanding of user interaction patterns. All of this can be documented and shared, enabling others to recreate or build upon it (perhaps open-sourcing the non-proprietary parts like the Unity scripts or TouchDesigner networks, and listing recommended hardware).

Thus, prototyping in steps ensures the project is feasible and grounded at each milestone, steadily moving toward the vision of a fully integrated, multi-sensory holographic torus platform.

Conclusion

This paper has explored the technological and conceptual framework for a **modular in-air holographic display system**, with a focus on realizing a **toroidal (donut-shaped) visualization responsive to multi-modal inputs**. We began by examining the cultural and symbolic significance of the torus – a shape emblematic of cyclical flow and unity – and proposed it as an ideal centerpiece for an interactive holographic experience. The envisioned system connects advances across several domains: **volumetric and holographic displays, real-time 3D rendering, brain-computer interfaces, gesture and audio interactivity, and control software** to weave these together.

Our survey of the technology layers showed that as of 2025, many building blocks are already in place or emerging: laser plasma voxels that can sparkle mid-air [1] [16], mist screens that create floating images [22], volumetric LED tables for 360° 3D views [27], and light field panels delivering group-viewable holograms [33]. Meanwhile, accessible EEG headsets and motion sensors provide streams of data that artists and developers are channeling into interactive works [46] [43]. By designing a system modularly, we can harness these developments: substituting different display "cores" or input devices as needed, without changing the overall system architecture.

We presented **integration scenarios** – from an EEG-relaxation torus for meditation to an audio-beat torus for concerts – illustrating the versatility of the concept. These examples, grounded in existing research and projects, suggest that a holographic torus could serve not just as a novelty, but as a meaningful medium for feedback and expression. For instance, the group brainwave scenario built on the idea of mutual brain-state art [14] hints that such a device might facilitate a new form of social connection or collective mindfulness. The wearable torus scenario extrapolates into the future of personal AR/holographic devices, where one's emotional state could be externally manifested as ambient visuals – a kind of **"mood ring 2.0"** in volumetric light.

In discussing **challenges**, we acknowledged that realizing this vision faces technical hurdles: display brightness and safety, stability of complex hardware, multi-sensor coordination, and user comfort. However, current research is actively tackling these issues. For example, the safety of touching mid-air plasma has been demonstrated by using ultra-fast lasers [57], and new materials and optical tricks are bringing full-color volumetric pixels to life [19] [62]. We highlighted sensible precautions and design adjustments (from enclosing moving parts to calibrating and smoothing sensor inputs) that can mitigate many problems. The path to a polished product is certainly non-trivial, but the trajectory of innovation in AR/VR and human-computer interaction suggests that **cross-disciplinary collaboration** – among optical engineers, software developers, UX designers, neuroscientists, artists – will progressively lower these barriers.

The **roadmap** we outlined provides a structured approach to proceed from today's capabilities to the ultimate goal. Starting with simpler AR prototypes ensures the ideas work in principle before investing in costly hardware. Intermediate builds will allow incremental learning and improvement. This stepwise strategy mirrors how many complex tech-art installations evolve: through iterative prototyping and user testing. It also allows for incorporating new breakthroughs along the way (for instance, if a new type of hologram projector comes out in 2026, we could integrate it at stage 3 or 4).

In conclusion, the feasibility of a **multi-modal neural-responsive torus display** is supported by the convergence of technologies we have surveyed. It is an ambitious convergence, but one that aligns with the current direction of immersive media – moving beyond flat screens into the spatial and experiential realm, and making interfaces more **intimate and intuitive** by tapping into human biosignals. The torus, as both a technical shape (closed loop geometry) and a rich metaphor (energy circle), offers a compelling canvas for these efforts. A successful implementation would be more than the sum of its parts: it could be experienced as a living light sculpture, one that *breathes* with a user's mind, reacts to their presence and sound, and perhaps even connects multiple people's consciousness in a shared visual symbol.

The implications span **art, wellness, education, and entertainment**. In art, it provides a new medium for exploration of themes like connectivity, aura, and the intersection of digital and spiritual (imagine exhibitions where viewers influence ethereal sculptures by thought alone, merging observer and artwork). In wellness, neurofeedback displays like this could facilitate meditation or therapy by making the invisible states of mind visible and playful. In education, scientific concepts like magnetic fields or toroidal vortices (e.g., in fluid dynamics) could be demonstrated as interactive holograms one can touch and hear. In entertainment, we get closer to the sci-fi trope of the holographic projector that responds to gestures and voice – a natural interface for gaming or performance.

Realizing the full system will require overcoming technical gaps and careful human-centered design, but as we have shown, the foundation is being laid in labs and maker spaces around the world. By mapping out existing knowledge and tools (see Appendix for specific technologies and resources), we hope this paper serves as a guidepost for anyone aiming to build or study such a system. It invites collaboration: perhaps the light-field experts team up with BCI developers and interactive artists to push this forward.

Ultimately, the project exemplifies the exciting frontier where **optical engineering meets interactive computing and neuroscience**. The **“holographic vortex”** can be seen as a platform for exploring that frontier – an embodiment of how technology can create new forms of expression and connection. As we progress toward prototypes and eventual deployment, the learnings will not only solve technical challenges but also refine our understanding of how humans perceive and engage with volumetric, responsive media. This knowledge will feed into broader advances in AR, VR, and human-computer symbiosis.

In a broader vision, one can imagine a future where such holographic systems are commonplace – perhaps the architecture of a room includes free-space displays that respond to residents’ moods, or public parks have interactive holographic sculptures that react to the collective energy of visitors. The torus, symbol of wholeness, might become a recurring sight in these environments, a reminder of the continuous loop between human and environment, input and output, self and other. By striving toward the modular integration of the technologies discussed, we take a step in that direction, turning a timeless geometric form into a **21st-century canvas of light, interactivity, and human expression**.

Appendix

Table 1: Sample of Commercial and Research Technologies for Holographic/Volumetric Displays (not exhaustive):

Name/Model	Developer/ Source	Type of Display	Key Specs (Year)	URL/Ref
Aerial Burton Laser Display	Aerial Burton (JP)	Laser-induced plasma voxels in air	~100 voxels, 50k pulses/s (2006)	[16] [18] (Ochiai et al.)
Fairy Lights (Femto Plasma)	Utsunomiya Univ. (JP)	Femtosecond laser volumetric	Safe touch, visible in daylight (2015)	[57] [17] (Open-access paper)
Heliodisplay M3	IO2 Technology (US)	Mist/Fog Screen (2D free-space)	22–42” image, interactive (2007)	[22] [63] (Wikipedia)
Voxon VX1	Voxon Photonics (AU)	Swept-surface Volumetric Display	18×18×8 cm volume, 30 fps (2018)	[26] [27] (New Atlas)
Looking Glass 27”	Looking Glass Factory	Light Field Panel Display	5K, 45–100 views, 16” depth (2025)	[33] (PetaPixel news)
SolidLight Surface	Light Field Lab (US)	Holographic Wall Panel (modular)	28”, 2.5B px/ panel, tiled (2022)	[34] (Press release / PVC article)
OpenHolo Library	KETI (KR) – Open Source	Software (CGH generation)	C++ API for hologram calc (2020)	[36] (Optica abstract)

Name/Model	Developer/ Source	Type of Display	Key Specs (Year)	URL/Ref
Emotiv EEG (EPOC/Insight)	Emotiv (US)	EEG Headsets (14-channel, 5-channel)	Wireless EEG, API/OSC (2010s)	[43] [51] (Emotiv docs)
Muse 2 Headband	InteraXon (CA)	EEG Headband (4-channel + PPG)	BLE, meditation app integration (2018)	*(Manufacturer site)*
OpenBCI Ganglion/Cyton	OpenBCI Community (US)	EEG Boards (4-8-16 channel)	Open-source, Unity/PD support	*(OpenBCI docs)*
Leap Motion Controller 2	Ultraleap (UK)	Hand Tracking (Stereo IR)	~180° FoV, ~0.01 m precision (2022)	*(Ultraleap site, v2 specs)*
Ultraleap STRATOS	Ultraleap (UK)	Mid-air Ultrasonic Haptics	256 emitters, ~8 kHz focal points	*(Ultraleap dev kit info)*
TouchDesigner + EEG Plugins	Derivative & Community	Software (visual programming)	OSC, Python support for EEG (2020s)	[64] [65] (Forum, examples)
Unity 3D + VividQ SDK	Unity Technologies + VividQ	Engine + Hologram Generation	Real-time CGH from Unity (2020)	[39] (Arm Community blog)
Max/MSP (EEG objects)	Cycling '74	Software (visual audio/graphics)	EEG integration via EEGLab, etc.	*(Cycling74 forum)*

(Note: URLs in the table are references in this document, not direct links, per formatting.)

Open-Source Libraries & Tools for EEG-Visual Integration:

- *BrainFlow (2021)* – Uniform API in Python/C++ for various EEG devices (Muse, OpenBCI, etc.), making it easier to stream brain data into apps. Can be used with Processing or Unity (with a C# wrapper).
- *Processing EEG Visualization sketches* – The Processing community has examples of real-time brainwave art; these can be adapted to torus graphics. E.g., Greg Gage's "Meditation Graphic" sketch.
- *OSC (Open Sound Control)* – Many EEG apps (like Muse Monitor, Emotiv's BCI-OSC) output data as OSC messages [51] . Libraries like oscP5 (Processing) or python-osc can listen and feed into the visual program.
- *Neurofeedback frameworks:* `OpenNFT` (Open Neurofeedback Training) – not directly for visuals, but could provide algorithms for interpreting EEG (like detecting when user is in state X).
- *Three.js or p5.js 3D* – If someone wanted a web-based torus visualizer, these JavaScript libraries can render 3D in browser; combined with websockets for EEG input, one could create a simple online demo.

Torus Generation & Shader Resources:

- Mathematical formula references for torus mesh generation (any 3D math textbook or Wolfram Mathworld entry on “Torus” for parametric equations).
- *Shadertoy.com* has several user-contributed shaders rendering tori with funky patterns – useful for inspiration on visual effects to apply on the torus surface (like animated textures, swirling colors that could link to input frequencies).
- `Unity Asset Store` might have pre-made torus models and materials (e.g., a plasma donut material). While simple to make ourselves, these could save time.
- For advanced generative patterns: consider L-Systems or noise functions on the torus surface. Libraries like `FastNoise` can generate 3D noise which one can map onto surface for procedural texturing.

Standards and Best Practices:

- *Laser Safety:* Refer to “IEC 60825-1:2014 – Safety of Laser Products” for classification guidelines. Ensure any laser used falls in safe limits for intended use (or proper protective measures are taken).
- *Display Calibration:* For AR and VR displays, standards like “FIDELITY (2019) – best practices for AR calibration” could be useful to align virtual content to real world which is analogous to aligning components in our system.
- *BCI Ethics:* If using brain data, consider privacy and consent. While our use is benign (just visualizing in real-time), one should still inform users what data is being used and not store sensitive data unnecessarily. Neurodata should be treated akin to medical data in terms of confidentiality.

Lastly, it’s worth noting that many aspects of this project live at the intersection of disparate fields. Continued **cross-disciplinary dialogue** will be key. An engineer might build the apparatus, but an artist will shape the experience, and a neuroscientist could guide the interpretation of EEG patterns, while a UX designer ensures it’s approachable for users. We encourage readers from all backgrounds – whether you tinker with LEDs, code shaders, record brainwaves, or choreograph performances – to contribute to this vision. The modular approach means there are entry points for different expertise (maybe you improve the display fidelity, or you create a new input mapping for emotional expression).

The holographic torus is ultimately a **platform**. Today it might visualize brain states; tomorrow someone could use it to represent Internet data flow or collective social media sentiment as a throbbing donut of light. By building it in a flexible way, we open it to myriad applications. Our work here provides a foundation – a technical and conceptual framework – upon which these future innovations can be built. We look forward to seeing this “holographic donut” concept evolve, hoping that it not only showcases technological prowess but also inspires moments of wonder, reflection, and connection in those who experience it.

---

1 15 (PDF) Laser-plasma scanning 3D display for putting digital contents in free space
https://www.researchgate.net/publication/253684853_Laser-plasma_scanning_3D_display_for_putting_digital_contents_in_free_space

2 Japanese scientists show off their latest 3D volumetric display – 311 Institute
https://www.311institute.com/japanese-scientists-show-off-their-latest-3d-volumetric-display/

3 12 26 27 28 29 30 31 32 53 56 Voxon's US$10,000 hologram table – no glasses required
https://newatlas.com/vr/voxon-photonics-3d-hologram-volumetric-displays/

4 Neurcosom: Bio-feedback Interactive Art Installation by Alma Digital Studio. :: Neuroarts Resource Center
https://www.neuroartsresourcecenter.com/post/neurcosom-bio-feedback-interactive-art-installati

5 13 14 46 52 EEG Art Exploration: Beyond The Canvas
https://www.neuroelectrics.com/blog/eeg-art-exploration-beyond-the-canvas

6 7 The Torus shape: convergence of science and spirituality
https://mandalashop-online.com/en/blog/the-torus-symbol-of-vital-energy-and-universal-connection-n136?srsltid=AfmBOoq4zftwvGpZ86bJGFL5WRi0mTyAD9QIITvmDcsmHAYHN3aIi0li

8 9 Celebrated Photographer Deana Lawson Takes an Unexpected Turn in Her New Guggenheim Show—and It Involves Holograms
https://news.artnet.com/art-world/front-deana-lawsons-lens-domestic-communes-divine-1970873

10 11 58 59 The world's first interactive 3d hologram, FlexiVol, shows promise
https://newatlas.com/technology/flexivol-interactive-3d-volumetric-display/

16 17 18 21 57 Touchable 3D holograms in daylight now possible using superfast femtosecond lasers | IBTimes UK
https://www.ibtimes.co.uk/touchable-3d-holograms-daylight-now-possible-using-superfast-femtosecond-lasers-1508599

19 20 54 61 62 Colour volumetric display based on holographic-laser-excited graphics using drawing space separation | Scientific Reports
https://www.nature.com/articles/s41598-021-02107-3?error=cookies_not_supported&code=ebcab27b-22b3-46ec-abd6-70c1d4d91692

22 23 24 25 63 Heliodisplay - Wikipedia
https://en.wikipedia.org/wiki/Heliodisplay

33 60 27-inch Looking Glass Light Field 3D Display Is Its Most Powerful Yet | PetaPixel
https://petapixel.com/2025/04/17/new-27-inch-looking-glass-light-field-3d-display-is-its-most-powerful-yet/

34 35 SolidLight, a scalable holographic display for huge video walls by Jose Antunes - ProVideo Coalition
https://www.provideocoalition.com/solidlight-a-scalable-holographic-display-for-huge-video-walls/

36 OpenHolo: Open source library for hologram generation, reconstruction and signal processing
https://opg.optica.org/abstract.cfm?uri=dh-2020-HF3G.1

37 38 39 41 42 The Magic and Science Behind Mobile Holographic Display - Mobile, Graphics, and Gaming blog - Arm Community blogs - Arm Community
https://community.arm.com/arm-community-blogs/b/mobile-graphics-and-gaming-blog/posts/magic-behind-holographic-display

40 Holographic Near-Eye Displays Based on Overlap-Add Stereograms
https://www.computationalimaging.org/wp-content/uploads/2019/08/olas2019.pdf

43 44 45 51 64 How to stream EEG data from EMOTIV to TouchDesigner
https://www.emotiv.com/tools/knowledge-base/emotivbci-technical-information/how-to-stream-eeg-data-from-emotiv-to-touchdesigner?srsltid=AfmBOopRsgwW0c8jL1T6qGXuwbhZhdjlDkzdu_i3hWNDE8RAybgrL5WN

47 48 EEG | Derivative - TouchDesigner
https://derivative.ca/tags/eeg

49 Mid-Air Haptics - Worth the Try? : r/embedded - Reddit
https://www.reddit.com/r/embedded/comments/1ltee79/midair_haptics_worth_the_try/

50 Ultraleap mid-air haptics
https://docs.ultraleap.com/haptics/

55 Immersive Technology Showdown: Volumetric Displays - Problem Solutions
https://problemsolutions.net/2023/04/19/immersive-technology-showdown-volumetric-displays/

65 We keep expanding our TouchDesigner EEG patch! : r/virtualreality
https://www.reddit.com/r/virtualreality/comments/1ksa2qn/we_keep_expanding_our_touchdesigner_eeg_patch/