ChatGPT

Spatial Menu Interfaces Anchored to Geometric Objects in 3D Environments

Introduction

User interfaces in virtual and augmented reality often break away from flat screens, instead leveraging **spatial, 3D menus** attached to objects or shapes in the environment. One intriguing design pattern is anchoring actions or menus to **geometric structures** – for example, placing tool options on the faces of a cube or at nodes of a 3D grid. These “geometric anchored” UIs exploit our spatial intuition by clustering commands on physical facets around us. This report explores such interfaces across VR, AR, and 3D applications, including product case studies, research comparisons of performance (speed, accuracy, learnability), and experimental designs using **platonic solids** (cube, dodecahedron, etc.) or **geometric grids** as interactive menu frameworks. A summary table is provided to overview key examples and findings.

Case Studies: Menus and Tools on Geometric Surfaces

Several VR applications have implemented menus that live on 3D shapes. Google’s **Tilt Brush** (a VR painting app) famously used a **menu cube** attached to the controller. The cube presented different toolsets on each face – brush types, colors, environment options, etc. – which the user could scroll or rotate through with their non-dominant hand [1]. Users likened it to holding a painter’s palette: “intuitive to scroll across the ‘cube’ menu on the left-hand controller, and easy to select options with the right… it feels like using a brush on a palette” [1]. After a brief learning period, rotating and tapping the cube became “second nature,” allowing fast access to core tools without any introductory tutorial [2]. **Figure 1** shows an example of such a cube menu interface.

Figure 1: VR painting application (Google Tilt Brush) using a cube-based menu. The user’s left-hand controller hosts a cube menu for tool selection, which can be rotated to access different facets (each facet holds a category like colors or brush types) [1]. Users found that after a short time, rotating and interacting with the cube “palette” felt natural and efficient [2].

Despite its innovation, the cube menu paradigm revealed some **usability challenges**. In user tests, Experience Dynamics found that the rotating Tilt Brush cube, while “a cool idea,” added **UI complexity that became a distraction** for many users [3]. The cognitive load of interpreting multiple faces and the physical effort of aligning the correct face could overwhelm new users in VR [3]. This suggests that while spatial facet menus can be engaging and immersive, they must be designed to minimize information overload. The Tilt Brush case illustrates both the promise (intuitive, hands-on feel) and the pitfall (potential complexity) of anchoring menus to a platonic solid.

Another product example comes from the **Neos VR/Resonite** platform, which introduced a modular UI system called *Facets*. Here, menu “panels” can be snapped onto a **grid layout on 3D surfaces** – such as the user’s hand-held dashboard or even the back of the hand and forearm [4] [5]. Users toggle into an *edit* *mode* and drag UI facet panels onto a grid overlay, where they lock into predefined slots on a controller or HUD element. This spatially organizes the interface into facets (literally like facets of a gem) that the user can glance at or activate quickly. The facets system essentially lets users **custom-arrange a personal 3D HUD**, grouping frequently used tools on fixed grid positions around them [4]. For example, one could attach a mini-map or avatar stats panel to their left wrist and a settings panel to a floating anchor in front of them. By snapping to a grid, the panels maintain alignment and consistent sizing, which aids in muscle memory for reaching toward a given tool. This case shows how **grid nodes on a 3D object or body** can serve as anchor points for menus, creating a customizable spatial dashboard. (Notably, Resonite allows facet anchors on the hands, forearms, or view, and turns panels on/off with a controller button press [5].) Such designs blur the line between interface and environment, as menus become **part of the virtual gear** the user "wears" rather than flat UI floating arbitrarily.

Spatial anchoring of UI is also used in VR games for diegetic effect. In *Fantastic Contraption VR*, instead of a conventional menu, a friendly **cat carries a tool-bag** of parts – the user physically pulls items from the cat's back rather than clicking buttons [6]. While not a platonic solid, this is a related pattern: the menu is attached to a **3D object/character**, making the interaction feel like grabbing real objects. The success of such approaches (also seen with "virtual tablet" menus or wearable backpack inventories in other games) underscores that 3D interfaces can leverage spatial memory and context. The key is that actions are **triggered by interacting with an object's facet or zone** (e.g. reaching into the cat's basket or touching a specific surface), rather than through abstract GUI widgets. This can improve immersion and discoverability of tools in a 3D space, as the context itself hints at the function (a bag holds items, a cube face labeled with a paint icon holds painting tools, etc.).

Geometric UI Paradigms in Research: Cube Menus and Beyond

In parallel to industry experiments, academic research has explored **"geometric menus"** for decades, often comparing them to traditional 2D menus. A landmark example is the **Command and Control Cube (C^3)**, proposed by Grosjean et al. (2002) as a 3D shortcut menu paradigm [7] [8]. The C^3 is essentially a six-faced cube where each face contains a 3×3 grid of menu items – giving 9 options per face, and multiple levels of hierarchy as needed by nesting cubes. Typically, the user would invoke the cube menu (often attached to their non-dominant hand or a fixed location) and then select a face and an item on that face. **Figure 2** illustrates the concept: a virtual cube with icons on its faces, which can pop up for system control commands. C^3 was **inspired by marking menus** (the idea of making a quick directional gesture to select a menu item) but extends it into 3D space [8]. Researchers envisioned that an expert user could eventually select menu items "blind" by gesturing toward a cube face (hence the C^3 had both a visual mode and a memory-based mode) [7].

Figure 2: The Command & Control Cube (C^3) – a 3D menu shaped as a cube with multiple options on each face. In this interface, users select commands by orienting to a face of the cube and choosing one of the items on that face (as illustrated by the user's hand reaching toward the cube). C^3 provides a rapid-access, immersive menu for virtual environments, inspired by 2D marking menus [7] [8].

The C^3 underwent user studies to evaluate its effectiveness. One study assessed variants of feedback: a fully visible cube menu, a "blind" mode (no visual, relying on muscle memory of where items lie), and enhanced feedback like audio or haptic cues. The **results indicated visual cues are crucial** – the fully visible cube yielded the best performance, while the blind mode was surprisingly not too far behind, but adding sound/touch feedback actually *worsened* performance (possibly due to sensory overload) [9] [10]. In other words, testers could learn to select items by remembering cube positions, but they performed fastest when they could see the cube and click the target. Extra feedback like a voice or vibration for each facet proved more distracting than helpful [9]. This finding echoes an important principle for geometric UIs: while spatial memory can be leveraged (expert users might eventually select by habit), for most users **clear visual anchors on the geometry are vital for accuracy and confidence**.

Researchers have also directly **compared geometric menus to other menu forms**. A 2016 ergonomic study by Jeong et al. systematically evaluated menu layouts for stereoscopic 3D UIs: they tested a **"cubic menu"** (3D cube with options on faces) against a **traditional 2D list** and a **circular/radial menu** in various conditions [11]. The outcome was that the **cubic menu was not as effective as the others** – it resulted in slower selection times and lower user preference overall [12]. In contrast, a simple planar "stack" menu (analogous to a pull-down hierarchy) combined with an easy selection technique performed best in their trials [11]. The authors note that the cube menu's spatial arrangement didn't pay off in practical efficiency, likely due to the time to rotate/aim and the mental effort to map items to faces [12]. Similarly, another experiment found a **cube menu was slightly slower to navigate** than a 2D floating window or a 3D ring menu, although the difference was not huge [13]. These studies suggest that while cube-based UIs are immersive, designers must be cautious: users might complete tasks faster with simpler layouts (especially for novices), unless the cube's extra dimensionality is truly adding value (e.g. more options visible at once or logical spatial grouping).

That said, geometric menus can shine when used in the **right context or in hybrid designs**. One variant is the **"spin cube" tangible menu**, which appeared in a research prototype by Lee and Woo (2010). This concept used a physical cube as an input device to cycle through menu items arranged in a ring [14]. By rotating the **tangible cube** in hand, users could scroll a virtual carousel of options (a 3D ring menu). The idea was to leverage kinesthetic sense – turning a cube in space – to select from a list without fiddling with buttons. While performance data from that poster is sparse, it exemplifies a creative **cross between geometry and interaction**: the cube here acted like a physical dial for menu navigation. Such tangible proxies may improve precision and enjoyment, though they require the user to hold a device.

Another research direction is exploring *larger platonic solids* for broader menus. In theory, a **dodecahedron (12 faces)** or **icosahedron (20 faces)** could host more categories or commands than a cube. In practice, however, these shapes become complex for users to orient; few documented interfaces use them explicitly. One could imagine a dodecahedral menu where each face is a mode (12 modes), but switching among 12 facets reliably would be challenging. We did not find prominent case studies of dodecahedron menus in real products, indicating that beyond cubes, higher-order polyhedra might be too cumbersome. (The *Command Cube* with 6 faces and ~54 total leaf items was already pushing limits of usability [7].) Some experimental art/visualization projects play with platonic solids as navigation structures, but these remain niche. For example, a developer's notes on a VR art tool mentioned supporting multiple shapes (tetrahedron, octahedron, cube, icosahedron, etc.) as **"attractors"** or indexing systems in their UI, potentially to explore different symmetry groupings [15] [16]. Yet, detailed evaluations of such complex solids as menus are scarce. It appears **cubes and spheres** have hit a sweet spot in balancing spatial richness with user comprehension, whereas 12- or 20-faced menus likely overwhelm all but expert users.

Spherical and Grid-Based Spatial Menus

While cubes offer discrete facets, **spherical arrangements** provide a continuous surface for placing menu items. Spheres can be thought of as an extreme case of a polyhedron with infinitely many tiny facets (or with items “floating” on the surface). Spherical menus often manifest as **radial menus extended into 3D**. For instance, research by Wentzel et al. (2020) examined a 3D **“marking menu” on a sphere** [17] [18]. In their study comparing VR menu archetypes, they found that a traditional 2D radial menu (marking menu) is very fast for small numbers of items, but doesn’t scale well to many items on one level. To accommodate 24 items in one menu, they “dispersed items over the surface of a sphere” – effectively creating a **spherical grid** of targets around the user [19]. Users would choose an item by moving a controller toward that item’s direction on the sphere. The results were illuminating: with fewer items, the radial (or spherical) menu was fastest; but with a large number of items, the performance slowed and users rated the spherical menu **less usable** than a simpler direct-selection menu [20]. In contrast, **direct interaction** (e.g. pointing directly at a nearby button or touching an item) remained consistently fast even as item count increased [21]. Meanwhile, standard **raycasting at world-space menus** (the common “laser pointer” approach) was *slowest* of all, though it was subjectively easy to use [22]. This aligns with intuition: having to precisely aim a laser at a distant UI (raycast) costs time, whereas grabbing a nearby object or making a quick gesture (direct or marking menu) can be more efficient if designed well. The *trade-off* highlighted by Wentzel et al. is between **speed and ease** – the spatial, gesture-based menus (like marking on a sphere) can be very fast for power users or shallow menus, but might intimidate novices or not scale elegantly; simpler point-and-click methods are more universally usable but not as quick.

Notably, the concept of the spherical menu in that study shows that **grid nodes on a notional sphere** can hold many options, but human motor control and perception impose practical limits. The team had to innovate a bit – for example, they allowed “marking menu” style gestures in 3D, and for larger sets turned it into a dispersed sphere of icons [18]. Even then, beyond a certain menu size, users preferred two levels of hierarchy to reduce clutter (which is telling; even in 3D, a huge flat menu of 24 items is hard to handle, so chunking into groups is beneficial).

In augmented reality and 3D visualization contexts, **attaching controls directly to objects via spherical or circular layouts** has proven effective. Englmeier et al. (2020) presented an AR manipulation tool using a **tangible sphere device** [23] [24]. When the user brought this physical sphere near a virtual object, a **ring of mode icons** would appear around that object (floating just outside it) [24] [25]. The menu consisted of three spherical icons placed equidistantly around the target – for example, “Move/Rotate,” “Scale,” and “Exit” modes for editing the object [26] [25]. To switch modes, the user simply moved their hand (or the sphere) toward one of those icons; the nearest icon would highlight, and pushing closer would select it [27]. This design anchored menu choices to **spatial positions on an invisible ring** encircling the object, essentially using the object’s local coordinate space as a menu framework. The study found this approach natural: the tangible sphere interface allowed for seamless mode switching and 7-DOF manipulation with minimal button presses [28] [25]. The spatial menu (three nodes around the object) meant the user didn’t have to look away or deal with a global menu – the controls were *right where the action was*. This illustrates a broader principle: **contextual menus on the object** (sometimes called toolglasses or in-place menus) can improve speed and usability in 3D tasks because they keep the user’s attention on the object itself. It’s a bit like having edit handles or widgets appear on the selected object – an approach also common in 3D modeling software (e.g., little cubes on the corners of an object to scale it, or rotation circles around it). Those are indeed geometric anchors (points, edges, faces of the object’s bounding volume) that trigger transformations [29] [30]. The AR sphere menu is a more explicit, mode-switching analog of that idea, proving effective in user testing.

AR Applications: Anchoring Information to Physical Object Grids

In augmented reality maintenance and data visualization, UIs often attach to real-world geometry. Instead of platonic solids created for UI's sake, AR interfaces tag **specific parts of an object with interactive icons or menus**. For example, an AR maintenance app might overlay numbered markers or buttons on a machine at various **grid-like nodes** – e.g. each bolt or each panel might get a virtual label that can be tapped for instructions. This essentially creates a UI grid on the 3D object. Technicians using AR glasses can then step through tasks by touching those spatially anchored prompts. Such a design was noted to greatly enhance context and reduce confusion: "with AR, rather than static manuals, technicians can access live, interactive documentation **linked to specific equipment components**" [31]. Each component becomes a **menu node** – select it to see its status, history, or maintenance options. This is analogous to having an invisible grid or coordinate system on the object where each important point triggers a mode or info panel. The interactive overlays are by nature aligned to the object's geometry, guiding the user's attention to the right spot in physical space [31].

While these AR examples might not use platonic shapes per se, they demonstrate the power of **3D spatial anchoring**. The user's performance (speed and accuracy) often improves when the UI element is directly where the action needs to happen. There is less cognitive shift – one doesn't have to find a menu in the periphery; the menu comes to the relevant location. This spatial mapping can also aid learnability: in essence, the physical layout of the machine becomes the menu structure (e.g. if step 1 is at the top-left of the machine, step 2 is below it, etc., the user naturally follows that sequence by moving in space). Academic reviews on AR in maintenance repeatedly emphasize that context-aware overlays (text, arrows, or buttons attached on equipment) reduce errors and speed up task completion, compared to referring back-and-forth to a separate UI or paper manual [31]. In sum, AR interfaces that treat the world as a canvas – placing clusters of info/actions on the world's geometry – exemplify the "grid node" anchoring pattern. Users are effectively interacting with a **3D graph of nodes** where each node is both a point in space and an actionable item.

Experimental and Conceptual Designs

Beyond the concrete cases above, designers have dreamed up many experimental UIs using geometric frameworks:

- **Hand and Body Anchored Menus:** Some VR interfaces map menu items to parts of the user's body (which can be seen as a "grid" of anchor points on a human figure). The TULIP menu by Bowman & Wingrave (2001) is a classic example: it placed first-level menu items at each fingertip of the left hand, and upon selection, showed second-level choices on the right hand [32]. This effectively turns your fingers into menu facets. Users wearing pinch-tracking gloves could select a command by touching a specific finger (left hand) with the right hand, then "pinch off" the desired submenu item from the right hand. The idea demonstrated moderate success – it kept the menu *within the silhouette of the user's hands*, making it always available and spatially consistent [32]. It's a very literal kind of geometric anchoring: each finger = one option. While TULIP required specialized hardware (pinch gloves) and might not scale beyond a handful of options (pun intended), it showed that **our body can serve as a platonic interface** (five points on a flat hand, etc.). Modern AR/VR toolkits echo this with **"hand menus"** – for instance, Microsoft HoloLens and MRTK allow popping up a slate or radial menu attached to the wrist or palm, which users can then poke with the other hand [33] [34]. These designs capitalize on the fact that we always know where our hands are; by anchoring UI to them, the interface becomes as "present" and personal as one's own body. The downside is limited real estate (only so many items fit on a palm or around an arm) and potential occlusion/awkward angles, but for quick mode switches or a small set of tools, body-anchored facets are highly discoverable.

- **World-in-Miniature and Polyhedral Worlds:** A conceptual UI pattern related to this topic is the *World-in-Miniature (WIM)*, where the user manipulates a small 3D model of the environment to navigate or issue commands. In some WIM implementations, the miniature world is bounded by a cube or sphere that the user holds [35] [36]. By touching parts of the mini-world (which correspond to big world locations), the user triggers teleportation or object interactions. While not exactly menus on the faces of a shape, it is a spatial interface where the faces/areas of a 3D model serve as input regions. One could imagine a WIM shaped like a dodecahedron that contains different scenes or data facets on each face – though this remains more hypothetical. The important notion is that **spatial clusters** (like miniature buildings in a model city) act as the "menu items" for navigation or selections. The WIM approach has been shown to improve **spatial understanding** and ease of navigation in VR [30], since it leverages an overview object to control the environment.

- **Diegetic and Immersive Controls:** Games and creative apps have tried diegetic geometric controls – for instance, a **virtual toolbelt** that is actually a ring (torus) around the player's waist with gadgets clipped to it. The player grabs items off their belt (physically reaching to their hip) instead of clicking a HUD. This can be seen as a **circular menu aligned to the player's body**. Such designs are highly immersive and can be faster for frequent tasks (you don't need to aim at a tiny icon – you just reach to roughly the right spot on your belt). However, they require good user calibration (the user needs to intuitively know where, say, the "back" of their belt is without seeing it).

In general, conceptual designs that use **regular geometric patterns** (grids, circles, solids) provide a structured way to organize 3D interfaces. A grid can impose order in an otherwise free-form 3D layout – Resonite's facet grid is one example, and others include VR home environments that let you pin multiple flat panels onto a curved **grid HUD** around you (some VR operating systems allow arranging app windows on a semi-spherical dome, essentially nodes on a geodesic grid in front of the user). These approaches have not all been formally studied, but they are appearing in various prototypes and products as VR/AR UX matures.

Performance and Learnability Considerations

Research and practice indicate several **trade-offs** when anchoring menus to geometry:

- **Speed vs. Accuracy:** Geometric menus like cubes and spheres can shorten selection time for experienced users by spatially grouping options (fewer hierarchical levels, and quick gestures to faces). Bowman's study found marking (radial) menus let expert users select very fast with a quick directional flick [20]. However, if too many options are presented at once (or if the geometry is complex), accuracy suffers – users may hit wrong facets or take longer to orient. Jeong's work showed a flat list could outperform a cube when measured by pure speed/accuracy metrics [12]. The optimal design often hybrids these: e.g. a cube menu might have a modest number of faces or items and rely on familiarity, or use highlighting/snapping assistance to improve accuracy (some implementations cause the cube to auto-rotate or "snap" to the face nearest the user's gaze to help target the correct side).

- **Learnability:** Anchoring to known shapes (like a cube or one's hand) can leverage prior knowledge – a user knows a cube has 6 sides, or that they have 5 fingers, etc., which gives a mental model for where things are. As noted in the Tilt Brush critique, after a short time users could rotate and tap the cube menu without hesitation [2]. This suggests decent learnability once the concept is grasped. However, initial discovery can be an issue: a new user might not realize a cube is rotatable or that more tools lie on other faces, especially if there are no prompts (Tilt Brush dropped users in with no tutorial, which worked for some but left others unaware of certain features hidden on back faces [37]). So, visual cues (like showing part of the back-face peeking out, or affordances indicating "turn me") are important to teach the geometric interaction. The **memory mode vs. visual mode** comparison in C^3 also touches on learnability – novices rely on seeing the menu, but over time they could memorize gestures to faces. In design, providing a **graduated approach** (visual aid at first, with the ability to go "eyes-free" later if desired) might yield both learnability and expert efficiency.

- **Ergonomics and Fatigue:** Interacting with 3D anchored menus can involve more **physical movement** than 2D menus. Reaching to a cube face, turning one's wrist to see a facet, or moving around an object to find the right node – all of these can introduce arm fatigue (the "gorilla arm" problem) if overused [38] [39]. Designers mitigate this by keeping frequently used options toward the front or in easy reach. For example, a cube menu might ensure the default face (front) holds the most common actions, so the user rarely needs to rotate it fully. Likewise, AR UIs keep markers within the technician's comfortable field of view to avoid excessive head/arm movement. Studies like Wentzel's also implicitly address this: direct input (like pressing a nearby virtual button) had an advantage in part because it was **less effortful** than waving a laser pointer or doing a wide arm swing for a marking gesture [21]. The goal is to design geometric menus such that they exploit natural body motion (turning wrist, tapping fingers) but **don't require sustained uncomfortable postures**.

- **Immersion and Context:** One clear benefit of spatial, geometric menus is improved **immersion** – they feel like part of the virtual world or tool. A floating flat panel can feel "HUD-like" and separate from the scene, whereas a contextual object menu (like grabbing a virtual knob or selecting a facet on a tool) is diegetic. This can strengthen presence and intuitiveness, as shown by examples like the Fantastic Contraption cat or AR component overlays. Users often prefer interfaces that **integrate with the content** (e.g., in VR painting, selecting a color off a 3D palette is conceptually satisfying). The trade-off is that these may not be as **flexible or discoverable** as traditional GUIs – a flat menu can list labels and have infinite pages, but a physical-inspired interface is constrained by the metaphor (a cube only has 6 faces – what if you need 7 categories? Do you add another cube, which complicates things?). Thus, some designs use **hybrid approaches**: for instance, a VR app might have both a radial menu on the controller for tool selection (spatial but structured) and a flat panel for less-used settings.

Summary of Examples and Findings

The table below summarizes key examples of spatial UI patterns using platonic solids or grid-based anchors, along with context and notable outcomes:

Interface / Example	**Platform / Context**	**Geometric Anchor**	**Notable Usage & Findings**
Google Tilt Brush (2016)	VR creative app (painting)	*Cube menu* (6 faces on controller)	Tool palette mapped to a cube on the off-hand controller. Provided an immersive "palette" metaphor – intuitive once learned, allowing quick scrolling through faces for colors/brushes [1] [2]. However, user tests noted the rotating cube UI could overwhelm some users, adding distraction and complexity if not carefully introduced [3].
Command & Control Cube (C^3)	Research prototype (VR, Workbench)	*Cube pop-up menu* (3×3 grid per face)	3D marking menu in cube form for system commands [7]. Enabled hierarchical menus (multiple levels of cubes). Evaluation showed **visual mode** (fully rendered cube) had best performance; a "blind" memorization mode was usable but slightly less effective, and adding audio/tactile cues hurt performance [9]. Demonstrated rapid selection potential but highlighted need for visual feedback on facets.
Tilt Brush vs 2D Menus (Jeong 2016)	User study (3D display)	*Cubic menu vs. list vs. radial*	Empirical comparison of menu layouts in 3D UIs. The **cubic menu was slower** and less preferred than a flat list or circular menu in selection tasks [12]. A simple stacked (pull-down) menu with raycast selection outperformed the cube. Indicates that without a strong use-case, a cube can be less efficient than familiar 2D-style menus.
Tangible "Spin Cube" (Lee & Woo 2010)	Experimental (VR/AR)	*Physical cube* (6 sides as input)	A tracked physical cube used to control a **3D ring menu** [14]. Rotating the cube cycled through menu items arranged in a virtual carousel. Aimed to leverage natural rotation for selection. (Demonstrated concept of using a platonic solid *device* to interact with menu – though performance data not published, concept influenced later tangible controller designs.)

Interface / Example	Platform / Context	Geometric Anchor	Notable Usage & Findings
Radial vs Spherical Menu (Wentzel 2020)	Research (VR menus archetypes)	*Spherical item layout* (for 8 vs 24 items)	Extended radial marking menus onto a **sphere's surface** to handle more items [19]. Found **direct touch selection** was fastest overall; marking (radial/spherical) was fastest for few items or in expert mode; raycast (point-and-click on world UI) was consistently slowest but very easy [22] [20]. Highlighted that **marking menus can use spherical layouts** for many items, but usability drops if too many targets – better to use hierarchy for large menus.
Object-Centered Spherical Menu (Englmeier 2020)	Research (AR object manipulation)	*3 nodes around object* (ring/ circle around target)	A **mode-selection ring** of icons appears on the object being edited [24]. User selects by moving a controller or a tracked sphere toward one of three spherical menu icons around the object [26]. Result: very fluid mode switching without looking away from the object. Demonstrated that anchoring tools *to the object itself* (at fixed spatial positions) yields efficient, intuitive interaction during 3D manipulation tasks.
Neos/Resonite Facets (2023)	VR platform (social/ creation)	*Grid-snapped panels* on user's UI anchors	Modular UI panels ("facets") that **snap to a grid** on the dash, hand, or head-up display [4] [5]. Users can customize arrangement (e.g. forming a cube, fan, or flat grid of panels) to their liking. Provides a structured yet flexible spatial HUD – facets stay put in user space, improving quick access. Emphasizes that even in open 3D UIs, grid layouts can impose order and predictability.
AR Maintenance Overlays (2018–2025)	Industry (Enterprise AR)	*Markers on equipment* (at key components)	AR UIs attach interactive symbols or info labels directly on physical machines at predefined points (like a virtual "grid" on the device). This links UI actions to the exact 3D location of the part. Studies report **2% error rates** with combined gaze-and-touch selection of grid menus in AR [40], and faster task completion when instructions are anchored to parts [31]. Shows effectiveness of contextually anchored menus for guiding real-world tasks.

Conclusion

Anchoring menus and actions to geometric structures in 3D interfaces offers an **immersive and intuitive paradigm**, but it must be employed with careful balance. Platonic solids like cubes can physically embody interface categories, turning abstract menus into tangible “objects” the user can manipulate. This can enhance learnability (by leveraging spatial memory and real-world metaphors) and delight users with more natural interactions – as seen with Tilt Brush’s palette cube and VR applications that let you “grab” menu items from world objects. Geometric anchoring also supports **contextual UIs**: placing controls right where they matter (on the object or body part in question) which streamlines interaction flow, especially in AR maintenance and VR creative workflows.

However, the research reminds us that **more 3D is not always better**. Cube or spherical menus can degrade performance if they introduce too much mental or motor load – e.g. hunting for the correct face or reaching around an object. Simpler layouts (2D panels or modest radial menus) sometimes outperform fancy 3D menus in speed and accuracy, especially for new or infrequent users [12]. The sweet spot tends to be hybrid approaches: for example, a spatial menu that uses a familiar shape (like a wristband or a half-sphere in front of the user) but limits the number of options visible, or provides guiding highlights to reduce search time [3].

User experience findings across case studies indicate a few best practices for geometric UIs: keep frequently used actions on easily accessible facets (avoid making users rotate a menu for every click), provide visual cues or previews of off-screen menu items (so users realize more options exist on other faces), and constrain spatial menus to ergonomic bounds (minimize the need for wide arm movements or contortions) [22] [3]. When done right, anchoring interface elements to platonic solids or grid points can improve **engagement** (the UI feels like part of the world), maintain **context** (tools appear where needed), and even boost **performance** for expert users (through spatial chunking and muscle memory). As VR and AR design continues to evolve, we can expect to see further innovations combining geometric frameworks with adaptive, user-friendly cues – perhaps dynamic menus that reconfigure their shape based on user proficiency, or AR UIs that project a “virtual toolcube” onto any object you focus on. The exploration of cubes, spheres, and beyond in interface design ultimately enriches the vocabulary of 3D interaction, aiming to make our experience as natural as handling objects in the real world.

---

[1] [2] [37] Critique Journal: Tilt Brush. Tilt Brush | by Andrew Buccellato | Medium
https://medium.com/@andrew_buccellato/critique-journal-tilt-brush-4818aae83b5

[3] "Less is More" is Key in AR/VR UX
https://frankspillers.com/less-is-more-is-key-in-ar-vr-ux/

[4] [5] Facets - Resonite Wiki
https://wiki.resonite.com/Facets

[6] [32] How to design Virtual Reality menus that do not suck | by Narendra Singh | UX Collective
https://uxdesign.cc/how-to-design-virtual-reality-menus-that-do-not-suck-9c06eb1df865?gi=fe937d0e78a2

[7] [8] [9] [10] (PDF) Evaluation of the Command and Control Cube
https://www.researchgate.net/publication/3998196_Evaluation_of_the_Command_and_Control_Cube

11 12 Youngjae Im's research works | Institute for Defense Analyses and other places
https://www.researchgate.net/scientific-contributions/Youngjae-Im-2202562902

13 Ergonomic evaluation of interaction techniques and 3D menus for ...
https://www.sciencedirect.com/science/article/abs/pii/S0169814116300014

14 A Design Space of Tangible Cubes for Visualizing 3D Spatio ... - arXiv
https://arxiv.org/html/2403.06891v1

15 16 devNotes 9-18-16 Platonic Solids Attractors – ArgosVu
https://argos.vu/devnotes-9-18-16-platonic-solids-attractors/

17 18 19 20 21 22 johannwentzel.ca
https://johannwentzel.ca/projects/vrmm/files/Wentzel_TVCG.pdf

23 24 25 26 27 28 A Tangible Spherical Proxy for Object Manipulation in Augmented Reality
https://sites.cs.ucsb.edu/~holl/pubs/Englmeier-2020-IEEEVR.pdf

29 30 35 36 The VR Book: Human-Centered Design for Virtual Reality :: UXmatters
https://www.uxmatters.com/mt/archives/2017/11/the-vr-book-human-centered-design-for-virtual-reality.php

31 Augmented Reality in Maintenance: Gauging the Future Impact
https://www.zapium.com/articles/role-of-augmented-reality-in-maintenance/

33 34 HandPoseMenu: Hand Posture-Based Virtual Menus for Changing Interaction Mode in 3D Space
https://3dvar.com/Park2019HandPoseMenu.pdf

38 39 VR Interface Design Manifesto [this insightful video deserves a lot more views] : r/oculus
https://www.reddit.com/r/oculus/comments/2tlqjb/vr_interface_design_manifesto_this_insightful/

40 Balancing Accuracy and Speed in Gaze-Touch Grid Menu Selection ...
https://pmc.ncbi.nlm.nih.gov/articles/PMC10708592/