The Constructs of Augmented Reality

A Developer's Guide to AR—Foundational Constructs, Skills, and Tools Required to Make AR a Reality

Qualcomm
developer network

Table of Contents

Building a New Reality

Augmented reality, better known as AR, requires an understanding of spatial architecture, features, and best practices. As a developer, learning how AR is constructed will help you build simpler mobile device experiences utilizing headworn units your users will love.

AR comes down to the three layers shown on the right, made possible by several constructs. Traditional computing is transformed into AR technology through systems and processes that correlate digital data with real-world locations in a 3D environment. Real-time video capture, rendering, sensors, inputs, and code are the essential components to tie it together. With this, developers can build the necessary layers to augment reality.

AR blends these 3 layers:

#3 Experience Layer—the presentation or application layer with which the users interact.

#2 Data Layer—information layer where an understanding of the physical world is developed, maintained, and stored.

#1 Physical Layer—corresponds to the real, physical world we experience through our 5 senses.

Experiencing AR

Users interact in the **Experience Layer** through a smartphone screen, an ergonomic interactive headworn unit, or a combination of both. AR allows the physical 3D space to be a canvas for more immersive experiences, new ways of interacting, and establishing more natural user connections. Based on the idea that the brain works best in 3D environments, AR is well-aligned with our human cognitive abilities and poised to revolutionize users' fascination with the digital world.

Early in its evolution, AR developers added digital content like overlays and static objects using systems that could remember their positions and orientations, while respecting users' viewports (typically smartphones—see Figure 1). This created the illusion of digital objects that appeared to live in the real world, but had no geometric or spatial understanding of the real-world environment.

Today's latest AR advances facilitate the illusion of realistic interactions between complex digital content and the real world. An AR system associates that digital content with real-world locations and geometry, letting users interact from virtually anywhere while adhering to real-world boundaries and limits.

For example, AR technology allows digital objects like a character to be placed and anchored relative to a physical object, like a couch in a room (see Figure 2). Just like in the real world, if the couch moves, the digital object anchored to it moves accordingly. However, the digital object cannot fall through the physical object or be pushed through barriers, like the couch's pillows.

Figure 1: Early AR with simple overlays on a smartphone.

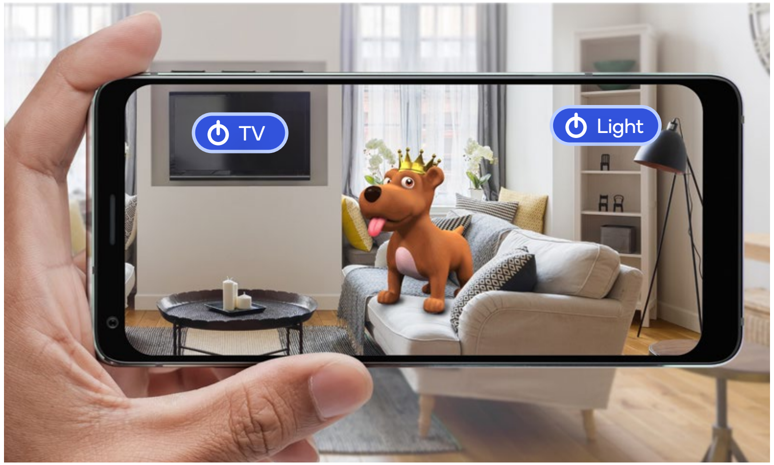


Figure 2: Advanced AR showing digital content (dog character) anchored to a physical object (couch).

Defining Attributes of AR

Several attributes define AR experiences:

Awareness: A semantic understanding or awareness of the environment. For example, computer vision can identify features that make up a wall, floor, parts of specific objects, etc. This data can be used for various uses, such as highlighting parts of a room and even replacing them with digital content (e.g., showing the walls painted in a different color).

Interactivity: As discussed above, digital content appears to live within the real world. A digital twin (a 3D mesh representation of a physical environment or object) helps an AR system ensure the content adheres to the real-world environment's boundaries, changes, and physics.

Persistence: Digital content can maintain state and evolve when users aren't present, thus ***persisting*** across time and adding to the illusion of a parallel, digital world. For example, digital objects can be ***anchored*** to physical locations and then viewed from different locations by different users and across sessions. Similarly, an object—such as a digital pet—can appear to age and evolve over time as the user revisits its location. Such persistence works closely with scene understanding (*see "Foundational Constructs—Scene Understanding" on page 10*) to create the illusion of digital content adhering to the shape and rules of the surrounding environment.

Presence: Being immersed in digital content blended with physical content creates an inherent sense of physicality and taps into our neurologic sense of ***presence***. For example, users anywhere in the world can send their presence digitally to the same physical location via realistic avatars, which reflect their movements and emotions in real-time. As in the physical world, there can be conversations and situations in the same environment, further adding to the realism and immersion while creating the illusion of an ever-evolving, parallel digital world.

Scale: Digital objects can be rendered at real-world scale. For example, a digital twin of a building to be constructed can be rendered at scale to fit onto its real-world property. The building's footprint and height respect real-world scale so that it appears in the exact size and position as its forthcoming real-world counterpart (see *"Foundational Constructs—Rendering" on page 11*).

Sentience: AR is a form of ***sentient computing***, where sensors and cameras provide the ***digital perception*** of the surrounding environment and facilitate reactions accordingly. This is essential for location-aware and context-aware functionality, such as placement of digital objects at GPS locations, digital objects that self-navigate, and more.

A Convergence of Technologies

Mobile devices are capable of delivering premium AR experiences. This is due to the evolution and convergence of several technologies:

3D Spatial Audio: Sounds in the real world come from different sources and directions. AR experiences are enhanced by ***binaural*** audio technology that reproduces 3D sound experiences in the digital world.

Camera Technology: The perception of augmented reality depends on a high level of detail. For example, when building precise digital twins of real-world environments and objects, mobile device cameras must capture video at high framerates in resolutions upwards of 8K. Multiple cameras capture different aspects of the incoming feeds and their results are ***fused*** to generate ultra-high levels of detail. This may be further enhanced with advanced ***computational photography*** methods.

Cloud Connectivity: High bandwidth, low-latency connectivity from 5G mmWave and Wi-Fi 6E and Wi-Fi 7 can allow for round-trip data transfers to/from the cloud for low motion-to-photon latency. Interactions can be captured and sent to the cloud for processing, and the results are rendered on the device in near real-time. Developers gain flexibility over processing done at the ***device edge*** versus the cloud.

Machine Learning (ML): ML algorithms, particularly for ***computer vision***, are key for detecting and tracking objects and features in scenes and mapping the real-world environment.

Processor Advances: Advances in mobile processors deliver powerful compute with power efficiency. For example, the heterogeneous design of Snapdragon® technology and our Qualcomm® Kryo™ CPU with powerful and energy-efficient cores, Qualcomm® Adreno™ GPU for PC-quality rendering, and Qualcomm® Hexagon™ DSP for signal processing and heavy vector processing (e.g., for ML).

Snapdragon, Qualcomm Kryo, Qualcomm Adreno, and Qualcomm Hexagon are products of Qualcomm Technologies, Inc. and/or its subsidiaries.

Split Processing

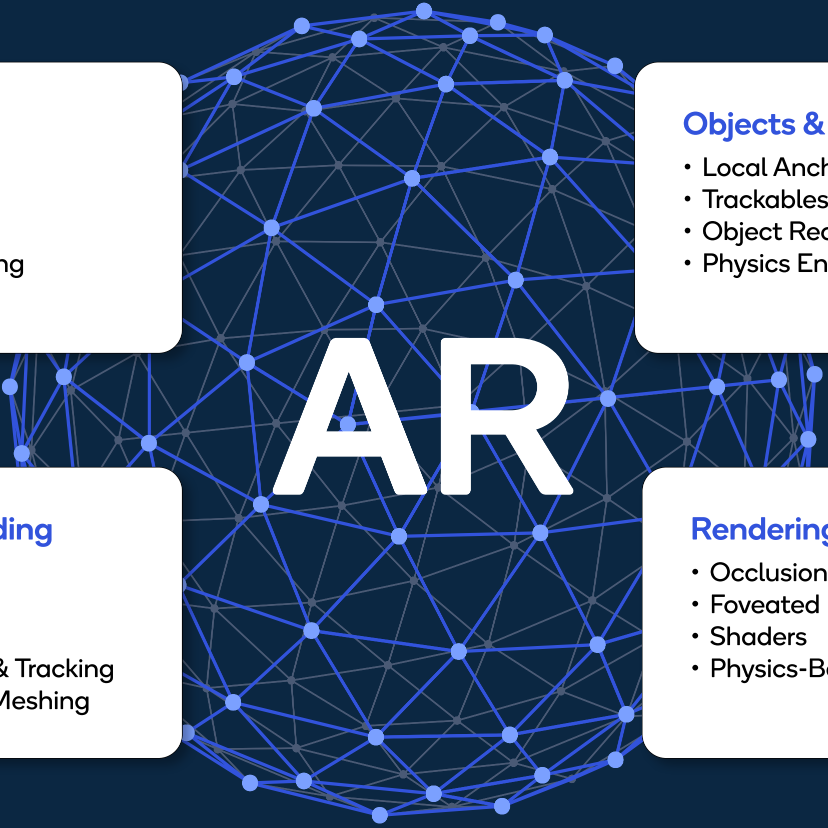
New XR devices, like lightweight headworn AR glasses, are poised to become the next evolution of the smartphone. They can offer more immersive experiences than 2D screens and will transition users from looking down at their phones, to looking around at their surroundings, while retaining that digital view.

Split processing powers this experience. Here, the glasses gather and send data to the user's smartphone for processing and then render an AR view based on data returned from the phone. By engaging the smartphone to perform this heavy lifting, headsets remain light and power efficient, ideally consuming less than one watt.

This processing is expected to be shared by a constellation of other devices surrounding us, including our PCs, connected cars, home routers, etc.

Foundational Constructs

Several foundational constructs allow developers to augment reality. These include hardware to bridge the physical and digital worlds, logic to work with objects and give them behaviors, scenes based on real-world surroundings, and efficient rendering to present the experience layer.



Hardware

- Viewport
- Sensors
- Positional Tracking
- Input & Hand Tracking
- Face & Eye Tracking

Objects & Behaviors

- Local Anchors & Persistence
- Trackables
- Object Recognition & Tracking
- Physics Engine

Scene Understanding

- Scenes
- Digital Twins
- Plane Detection
- Image Recognition & Tracking
- Spatial Mapping & Meshing

Rendering

- Occlusion
- Foveated Rendering
- Shaders
- Physics-Based Rendering

Let’s review the highlighted foundational constructs in more detail.

Foundational Constructs—Hardware

Hardware provides the viewport into AR.

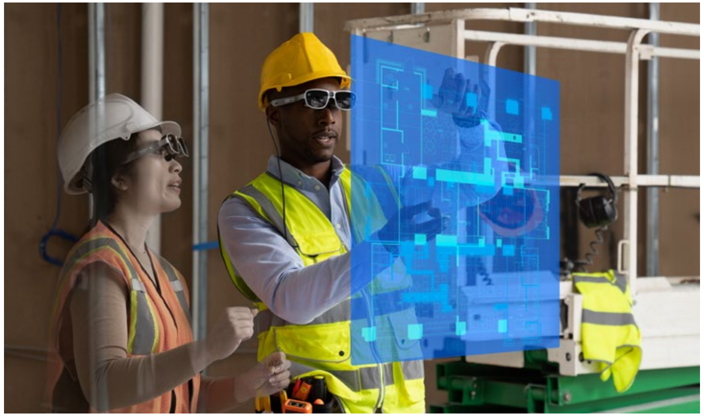
A device's **viewport** consists of a display (e.g., a touch screen or a headworn device) with a camera view to capture a real-time video stream and render digital content. The viewport acts as the ***canvas*** where the real-world blends with the digital world.

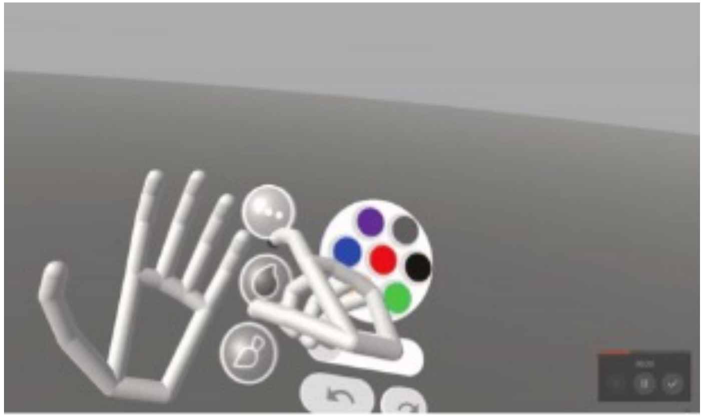
Many **sensors** are required to digitize data gathered from the real world. Inertial measurement units (IMUs) like the compass, accelerometer, gyroscope, and GPS can track movements, associate virtual locations with real-world locations, and track device orientation. ***Sensor Fusion*** combines data from multiple sensors to derive more complex information (e.g., to estimate a position when GPS line-of-sight is obscured).

Input allows users to interact with digital objects. Input can come from movements detected by IMUs, gesture recognition on touch screens, or handheld controllers tracked in 3D space.

Sophisticated ***hand tracking*** methods can involve sensors or computer vision to track the position and orientation of arms, hands, and even fingers in 3D space. Input data can be used to manipulate digital objects, interact with 3D GUIs, or animate digital representations of the user (e.g., realistic avatars, on-screen hands, etc.).

Developers may also incorporate face and eye tracking. ***Face tracking*** tracks facial features and movements (e.g., to convey emotions on virtual avatars). ***Eye tracking*** follows where a user is looking and can reproduce eye movements on virtual avatars or highlight certain objects in the user's view based on where they're looking.





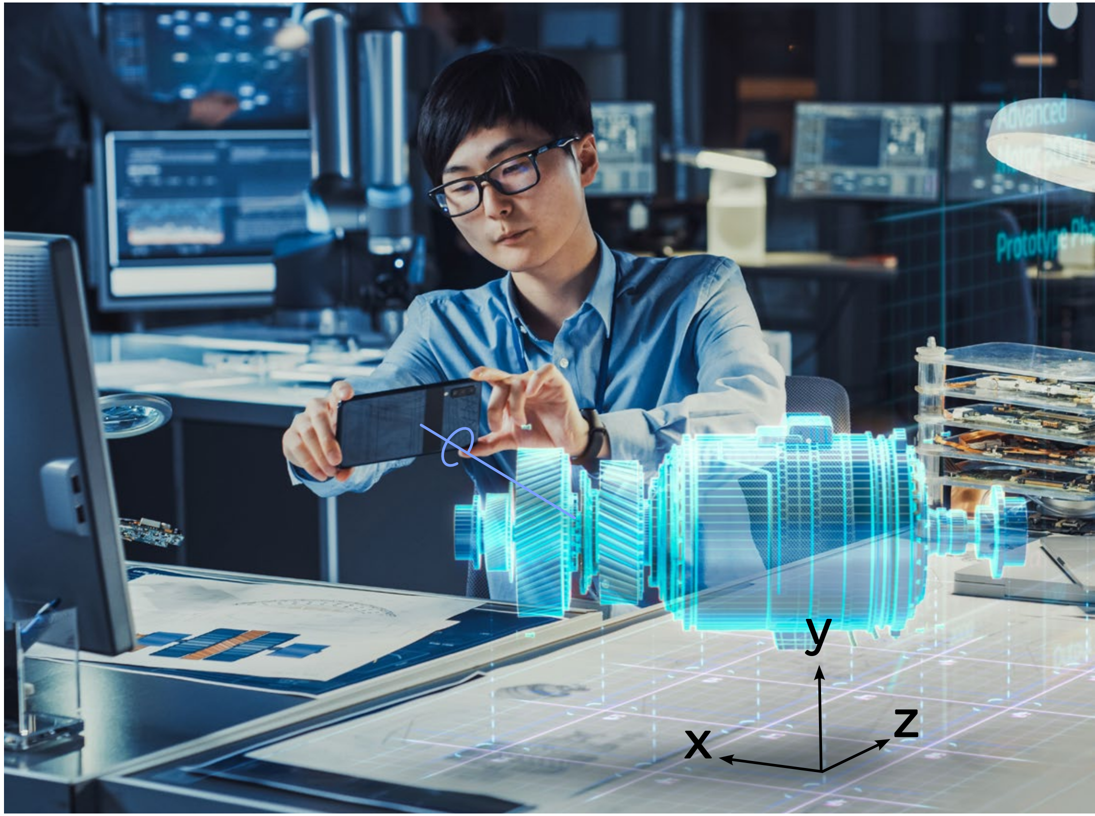
Foundational Constructs—Objects & Behaviors

Metadata is required to position, track, and persist digital content.

Anchors offer the ability to anchor (i.e., lock or pin) digital assets in space, associating them with real-world GPS locations or objects. ***Local anchors*** are those created and used by a single user. ***Cloud anchors*** are shared by multiple users, usually in relation to a coordinated space managed by a cloud server.

Trackables are virtual points or planes to which anchors may be attached. For example, a trackable associated with a moving surface in a dynamic environment causes all anchored objects to reposition/reorient accordingly. This information is often persisted across sessions. For example, a virtual object placed at a location in one session should be visible in a subsequent session, even if viewed from a different location or orientation.

A **physics engine** can simulate real-world physics, allowing for realistic behaviors and reactions of virtual objects (e.g., to simulate how objects fall given gravity, aerodynamics, etc.). A physics engine defines the rules of what's possible and can even be made to bend the rules for different types of experiences.



Foundational Constructs—Scene Understanding

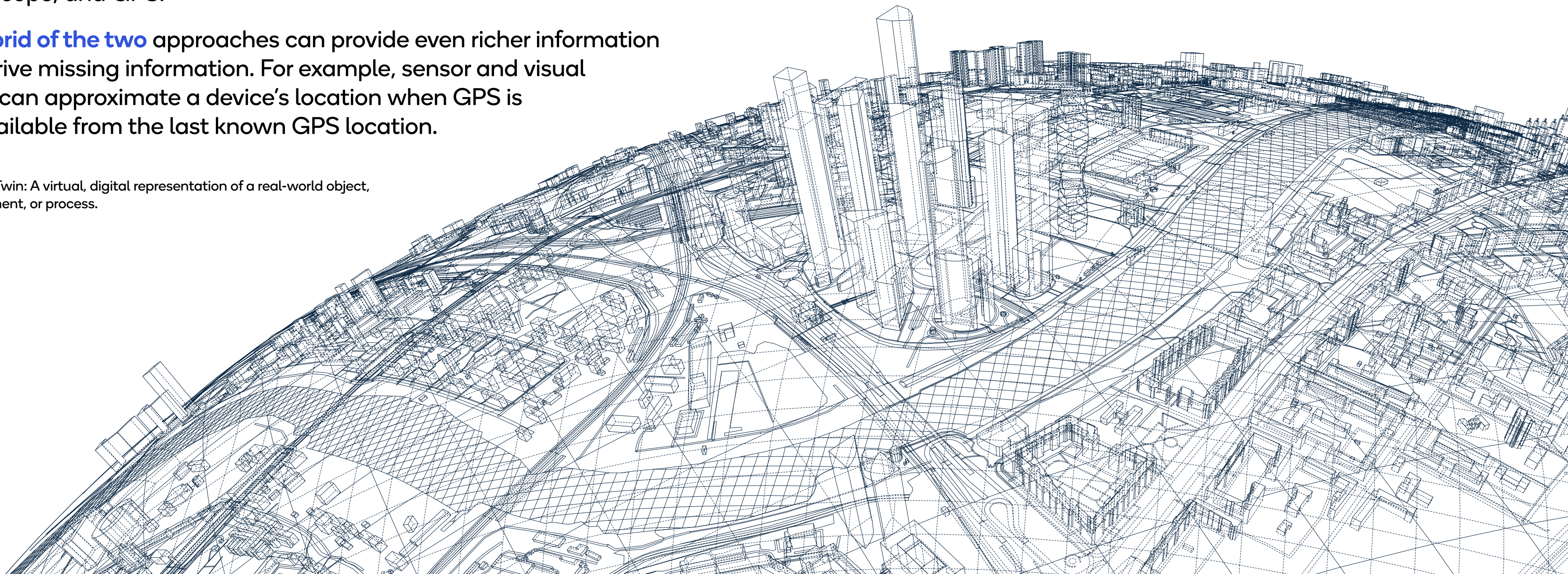
A key construct of AR is a digital twin,[1] a 3D mesh representation of a physical environment or object. Digital twins may be built and used for ***spatial mapping***, to map and store ***world data*** for ***scene understanding***, and to compute the interactions with digital content.

Methods for spatial mapping include:

Marker-based approaches identify visual features or markers captured by a camera. Image processing algorithms and computer vision techniques, like image recognition, are often employed to detect features like corners, edges of objects, etc. This can include plane detection that detects surface planes to define boundaries (e.g., walls, tabletops, etc.). Virtual planes can even be generated to create virtual surfaces to render details (e.g., infographics, signage, etc.).

Markerless-based approaches use data from IMUs, like the compass, accelerometer, gyroscope, and GPS.

A hybrid of the two approaches can provide even richer information or derive missing information. For example, sensor and visual data can approximate a device's location when GPS is unavailable from the last known GPS location.



---

[1] Digital Twin: A virtual, digital representation of a real-world object, environment, or process.

Foundational Constructs—Rendering

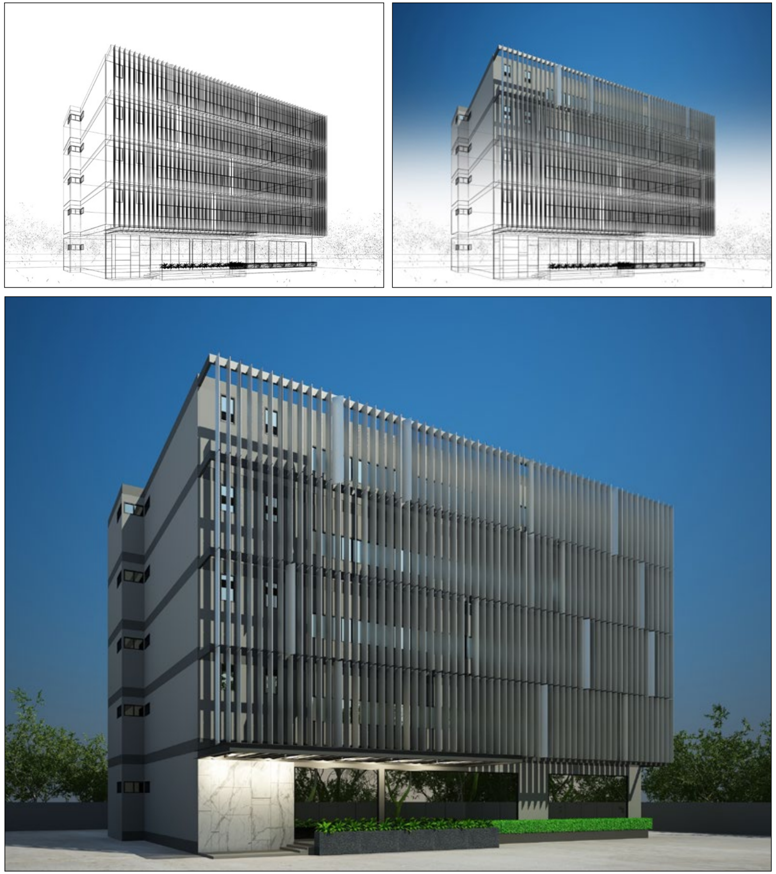
Rendering techniques play a key role in blending digital content with the physical world in a realistic and believable manner, and may rely heavily on scene understanding.

Depth understanding (aka, depth estimation) is an important calculation and plays a key role in deriving distances to features, objects, and parts of scenes. It can facilitate interactions (e.g., prevent the user from pushing a virtual object through a physical wall) and ***occlusion***, where physical objects can cover virtual objects. Depending on the user's viewpoint, a digital object can be partially or even fully covered by a physical object, so it needs to be shown or hidden accordingly, just like a real occluded object.

Foveated Rendering *(applicable only to headworn units)* can optimize performance by reducing the details and resolution to be rendered in peripheral areas. In some units, dynamic foveated rendering based on eye tracking can update the foveated areas based on where the user's eyes are looking.

Shaders can add realism by implementing effects, such as lighting and shadows, particle systems and water effects, motion blur, bump mapping, and distortion. Shaders operate at many levels of granularity, from geometry right down to the pixel level.

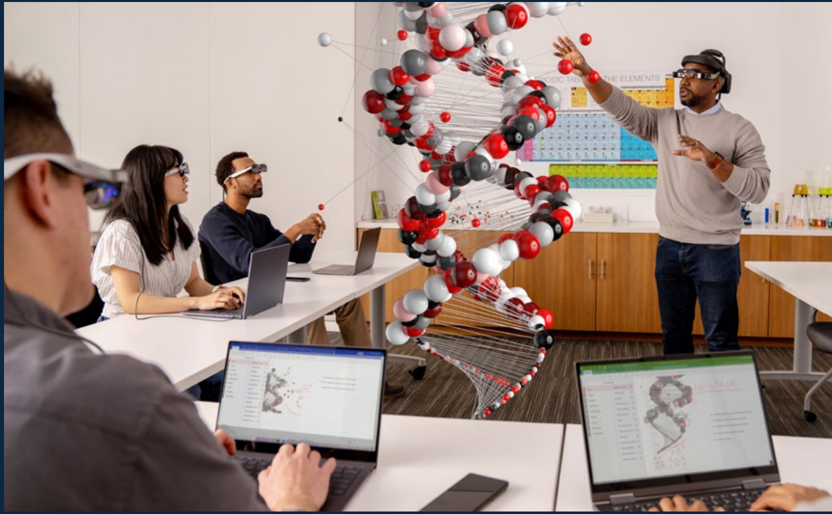
Physical-based Rendering (PBR), often performed using shaders, simulates how light reflects off different materials like in the real world. This improves realism and helps virtual objects blend in with the physical world. For example, shadows and light sources from digital objects and particle systems can be rendered on the environment.



Current Use Cases & Opportunities

Below are just a few use cases possible today involving interactive AR that exhibits some of the defining attributes mentioned earlier.

Enterprise Examples



Virtual Gatherings (enterprise and consumer): Realistic avatars that capture their respective real-world participants' facial expressions, movements, and other aspects. Users can send their presence to the same virtual location and see the same digital scene environment, regardless of physical location and time zone. The types of venues possible are virtually endless, ranging from meetings and concerts to gaming environments. Some virtual venues can incorporate digital currencies to buy and sell digital assets, like objects and real estate.

Education and Training: Training can be enhanced by immersing trainees in the digital twin of the workplace environment. Employees can get a feel for the surroundings, understand how objects behave, and build memories of where things are located. This can be enhanced with knowledge from expert workers (e.g., how to handle certain items) shared through information popups, narrated audio, or even animated digital objects and avatars. Persistence can be used to save the trainee's progress across follow-along training sessions or to share objects while learning team processes.

Industrial Design and Manufacturing: Virtual objects can be designed at a physical location (e.g., in a design studio) that adheres to real-world scale, physics, and local lighting conditions. This can help designers visualize their products in the real world and even identify design issues before production begins.

Medical/Healthcare: AR technology can provide doctors with enhanced precision for robot-assisted surgeries or help to train medical staff on digital twins that behave like real-world entities.

Remote (aka, what-you-see) Assist: A technician repairing an object through an AR viewport can share their view with a remote support worker. Interactive digital twins of components can be rendered so both users can manipulate them, share information, and collaborate on a resolution.

Remote Maintenance: Similarly, remote monitoring and machinery maintenance can be conducted using digital twins that capture the state of their real-world counterparts.

Digital Storefronts: Developers can augment physical brick-and-mortar locations with a digital scene in which users can interact with digital objects and make purchases. Items for sale can range from digital assets purchased with digital currency to digital twins placed in a cart, representing the physical objects the store will ship to the customer.

Consumer Examples

Games: Gaming environments based around real-world locations with realistic avatars representing players worldwide offer huge potential to create new types of games and social interactions.

Consumer Life Enhancing Experiences: Simple, yet effective life-enhancing experiences are poised to drive AR in the near term. For example, a recipe app could allow users to pin a virtual baking recipe in physical space and interact with it via hand tracking (e.g., scroll, check off steps, etc.) using even the dirtiest of fingers. In the classroom, AR can help bring subject matter—such as historical figures—to life with room-scale, interactive models. In museums, AR can display extra information for displays, such as how certain works of art were constructed. And for retail, AR can allow customers to visualize how products like furniture will look when placed in their room.

The Metaverse

AR is poised to drive the ***spatial web***, an evolution of the Internet that goes beyond linking websites together to also link people, spaces, and assets. Perhaps the ultimate use cases for AR will be found in metaverses—environments where users can immerse themselves more deeply than ever before.

These immersive environments can exist parallel to the real world, allowing users to interact, explore, and even exchange digital assets. They'll take many of the aforementioned use cases to the next level by elevating the feeling of human presence, social interactions, and realism of digital content behaviors. For example, realistic avatars are poised to replace today's 2D profile pictures, and one day, users may have multiple avatars for different uses (e.g., work versus personal).

Development & Content Creation Skills

To start developing in AR, you should be comfortable with the following:

- **Working in 3D space**, including 3D math like **vectors** and **matrices**.
- **Graphics pipelines** to convert assets from art software packages to a format optimized for a given platform.
- **Shaders** to implement special effects.
- **Scene management** to load/render only what's required for the current viewport (i.e., the field of view provided by the user's 2D screen or immersive headworn device).
- **Real-time, frame-based software architectures**. For example, a typical game loop acquires user input, updates game logic based on that input, and then renders it accordingly. An AR loop adds sensor input collection and considers the physical world in the update and rendering phases.
- **Real-time debugging techniques**, including remote debugging of embedded devices.
- **Digital content creation**, including character modelers and animators, object modelers, UI designers, and texture artists. 2D art assets can include imagery and textures for signage, information boards, virtual UIs, as well as headworn units that remain fixed on screen. Textures are also used for effects like particle systems (e.g., water effects, smoke, etc.). 3D art assets include objects, characters, and environmental models that augment the surroundings. ***Model rigs*** can be created to procedurally animate or use streams of animation data.

UI and UX

Quality AR interactions should have quality user interfaces and quality user experience design.

Many of the gestures used in today's 2D mobile apps (e.g., taps to select objects, swipes to move objects around, and pinches to resize objects), generally translate well for AR interactions so that most mobile app developers can apply these in AR development.

This article from Google provides a good overview of gestures in AR. In addition to implementing these gestures on touchscreens, developers may also implement them via hand tracking.

And always remember, safety first! AR experiences are more immersive than standard applications, so users can lose track of their surroundings or even experience cybersickness. To help prevent this, remind users to be aware of their surroundings and avoid having them walk backward. Also, limit AR session time so users can re-ground themselves in reality, but make it easy to resume their session in the same state.

See our eBook on Cybersickness for more tips.

Options and Paths to AR

Developers have several options for AR development, ranging from porting existing experiences to writing apps from scratch. Choose the one that works best for your project and your skill level.

Add to Mobile/Web AR

Developers can add AR as a feature by gradually migrating parts of their 2D content into AR to enhance existing experiences. This can reduce the barrier of entry into AR by adding an AR layer that works together with existing 2D content. For example, a racing game on a smartphone can be viewed through an AR headset that renders a map during gameplay.

Migrate from VR

VR developers can port their existing VR experiences and content to AR. In the simplest case, start by removing VR backgrounds from existing games or apps, so objects exist in AR space.

Port from Other Platforms

Thanks to tools like our Snapdragon Spaces™ XR Developer Platform (discussed below) and standards like OpenXR, it's easier than ever to port AR apps to mobile devices, including smartphones and headworn displays powered by our Snapdragon SoCs.

Build from Scratch

Starting from scratch? Build immersive experiences for headworn AR from the ground up in Unity or Unreal with our Snapdragon Spaces XR Developer Platform.

Qualcomm Technologies, Inc. Tools & Resources for AR

AR solutions include development tools and SDKs, SoCs, and reference designs.

Snapdragon Spaces XR Developer Platform: XR platform for building immersive AR experiences. It comprises an HDK based around the Motorola edge+ smartphone and Lenovo ThinkReality A3 smart glasses, rich SDKs for Unity and Unreal, and an OpenXR compliant runtime.

Snapdragon XR2 5G Platform: SoC that brings the best of our mobile compute innovations to XR.

Snapdragon XR2 Reference Design: Reference design for building immersive 5G AR headsets with eye tracking, powered by the Snapdragon XR2 5G Platform.

Snapdragon Profiler: Analyze the processing load, memory, power, thermal characteristics, network data, and more in real-time.

Qualcomm® Computer Vision SDK: Rich SDK packed with mobile-optimized computer vision processing functions. Use the SDK to accelerate feature detection, motion detection, and object tracking, etc., for AR applications.

Qualcomm® 3D Audio Tools: Tools to record, edit, and produce spatial audio that enhances the immersive experience.

Qualcomm® 3D Audio Plugin for Unity: Binaural spatial audio plugin for Unity. Visually place, author, and tune sound objects and fields in 3D.

Innovate Together

Qualcomm Developer Network is a collection of software and hardware tools, inspiring our community of developers to push the boundaries of mobile. We're continuously creating some of the most innovative, powerful, and disruptive technologies in the world, and Qualcomm Developer Network is the gateway through which you can discover the tools you need, whether you're building high-performance apps, smart Internet of Things (IoT) devices, immersive virtual reality experiences, or for other emerging technologies.

developer.qualcomm.com

Qualcomm
developer network