The Holographic Components of Spatial Computing

Huaining Geng

jarvis.reality@gmail.com

www.holocom.org

Abstract

The hardware of XR has evolved to the extent that we can wear the headsets to comfortably enjoy the game experience, but the bottleneck in front of us is that the types of experiences are too few compared to what we can get from desktop computers and smart phones. To better serve humanity with the new devices, we have to invent appropriate software architecture paradigms and discover more engaging scenarios to expand the application domains where people can explore. By applying the holographic thinking in spatial computing, we conceived and proposed several software components to solve this problem. XR software developed using these components will cover an extensive spectrum of spatial applications, including communications, light entertainment, production applications, and networking.

In the current spatial computing environment involving avatars, chat applications and games occupy the main application scenarios. They utilize synchronization technology to capture and transmit data to implement working mechanisms. An application like Horizon Worlds is a perfect example, synchronously capturing actions and sounds of a player and sending them to a remote player. Although many users hope to have a similar synchronous experience, it is far from enough to meet people's various asynchronous needs in spatial computing. In order to significantly expand spatial computing scenarios, a holographic data model consisting of avatar models, motions and sounds is proposed to build new asynchronous applications.

The most prominent video formats in spatial computing are VR180 and 360-degree spherical video. Generally a VR180 video is shot with a stereo camera, using a distinct image for each eye to present a stereoscopic view, while a 360-degree video usually has a monocular view. While bringing in a level of immersion not found in traditional 2D videos, they both only have 3 degrees of freedom, meaning the viewer can't move in any direction to gain a deeper understanding of the subject being viewed or its surroundings. To improve the spatial immersion, a new video format called spagraph is proposed to enable a 6DOF viewing experience.

Current spatial computing UI schemes already employ large spaces, typically at least three 2D panels arranged in a circle centered on the viewer. Each panel is like a 2D screen, usually scaled up compared to the real world to improve the viewing experience, and sometimes scaled down for ease of use. In Apple visionOS, volume applications that contain 3D objects can be positioned with 2D applications in a shared space, whereas more immersive experiences such as games can take up the whole space. Given that there are some flaws in these designs that affect the experience on XR devices, in order to make full use of spatial computing, a UI scheme called SUI is proposed to develop 3D spatial applications.

In the current iteration of the metaverse, while some applications such as Rec Room have implemented some features of the metaverse in the app per se, each one of them is an information silo when putting all the apps together. We proposed a web-based solution called spatial web that connects each silo to form an open web for metaverse. The web-based metaverse has more merits than the app-based one.

Contents

List of Figures

Chapter 1

Avagraph - A Data Model of Holographic Avatars for Spatial Computing

1.1 Introduction

1.1.1 Synchronous and asynchronous reality

Vannevar Bush created memex[1] to conceptualize the information storage in a special device and supplement the human memory, which is materialized by computers in both command line and graphic systems. Just as early computer scientists created ASCII and Unicode for text typed on a keyboard, JPEG for images captured by cameras, MP3 for sounds captured by microphones, and MP4 for videos recorded by cameras, we should invent a new kind of data format for spatial computing[2] to save the data captured from XR[3] input sensors.

Like 2D computing, spatial computing can support both synchronous and asynchronous scenarios. In the XR field of spatial computing, synchronous applications such as chats and games have been explored to a considerable level, overcoming spatial barriers and bringing remote users together. However, in order to better enable people to communicate with each other, data needs to be saved and read so that it is not hindered by time. Andreas Rene Fender et al. proposed the concept of asynchronous reality[4] for this, but didn't come up with a detailed software solution for it. To conquer the temporal constraint, we proposed a new model called avagraph as the carrier of the required data.

1.1.2 Avagraph - holographic avatar

As the most prominent data model in XR, avatar is an important part of the character data. Different from the data formats in the 2D computing, the character data has a potential to be the basic information unit in XR. The model, motion, and sound of the character can be fused into a type of holographic data structure called avagraph for saving, reading, editing, transmitting and displaying the holographic avatar.

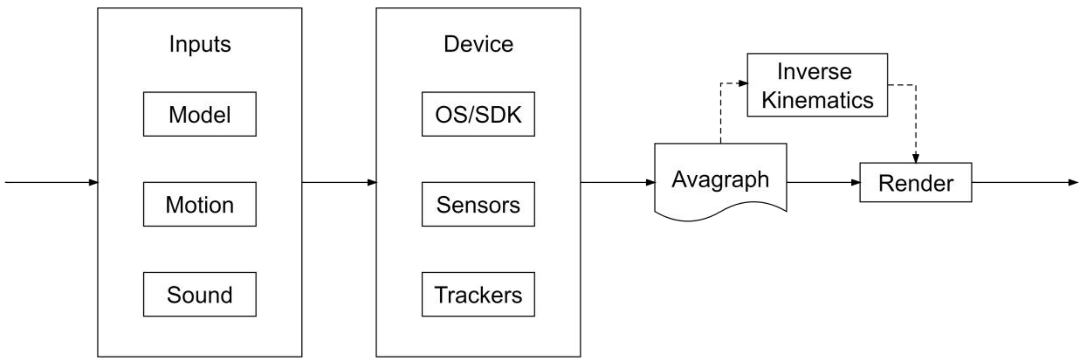

Figure 1The architecture for avagraph

While holography in the optical world uses interference and diffraction principals to save and read the wavefront of light reflected from the surface of objects, it's possible in XR to makes use of the skeleton rigging, skinning, and inverse kinematics on avagraph to implement a similar holographic effect.

1.1.3 Avagraph as an input

The basic input devices of XR are the trackers and sensors, including those on the headsets and controllers. Since motion and sound are the most natural input methods of XR, we should capture the data from the specific input devices, keep those data in a special type of data structure called avagraph, and read it out and display it as a basic information unit.

1.2 Characteristics

1.2.1 Asynchronous

All the data in avagraph is asynchronously persisted, retrieved only when it’s time to transfer or display the information contained in the data.

1.2.2 Holographic

The retrieved data from avagraph must be processed in a way that the final output received by eyes will present a 6DOF viewing experience.

1.2.3 Extensible

Not only can avagraph cater for the emerging XR, but also will add new application genres for 2D platforms like mobile computing, as soon as it meets the asynchronous and holographic properties. In a sense, all information having those properties can be scoped to spatial computing.

1.3 Data structure

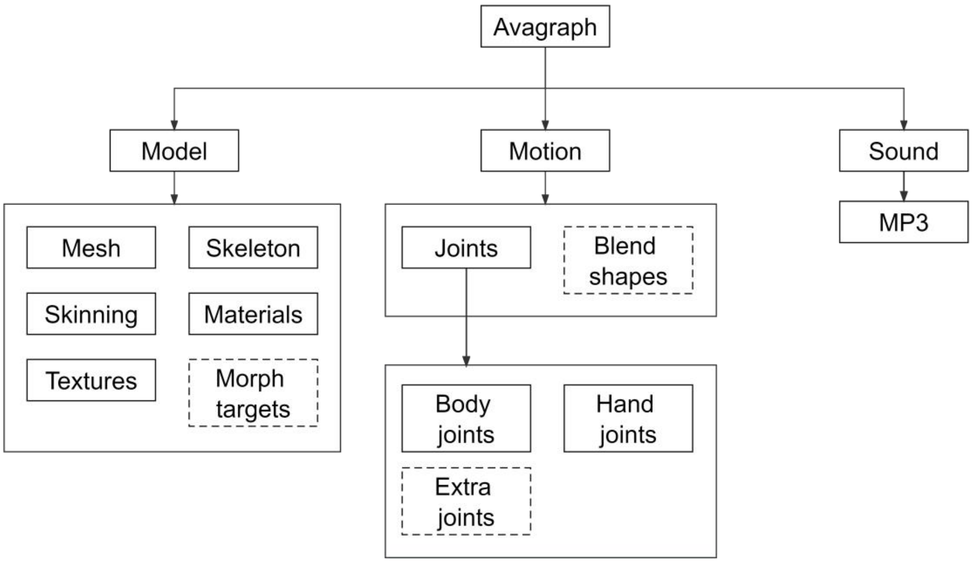

Figure 2The data structure for avagraph

1.3.1 Model

Theoretically avagraph can be compatible with all model formats such as glTF[5], VRM[6], and other 3D model data formats. It only depends on the implementation side to analyze and use the specific data format for rendering and presentation. The model data will be put in the avagraph file or referenced by a uri to the outside file.

1.3.2 Motion

The motion data is comprised of face and body parts. The face motion contains the blend shapes of the character's face expression. The body motion includes the positions and orientations of the skeleton joints of body and hands.

1.3.3 Sound

It can be any music, soundtrack or voice of characters.

1.4 Comparison to relevant formats

The core principle of avagraph is to be compatible with as many model formats as possible.

1.4.1 glTF

While glTF is a common 3D model format used by engines and applications, there are many other 3D model formats such as OBJ and glTF-derived VRM. Motion and sound can be added as extensions to glTF, but glTF is not likely to be compatible with other model formats. It is better to use glTF as a general model format rather than a holographic data format.

1.4.2 FBX

FBX typically contains model and skeletal animation data. Although it contains model and motion except sound, it's a proprietary file format and also unlikely to be compatible with other model formats.

1.4.3 MP3

MP3 is a coding format for digital audio, so it can't be an alternative of avagraph.

1.5 Application scenarios

1.5.1 Instant messaging

Avagraph can be an alternative of text in an instant messaging app. Avagraph as the basic information unit of the holographic communication can bring the most natural input and output experience.

1.5.2 Social network

A user can select a model and record motion and voice in an avagraph file to be shared with other users, and also manually make a complex animation and mix it with a soundtrack into a shareable avagraph file.

1.5.3 Spatial SNS

Avagraph can be shared based on the location in a small space, such as a room, or in a large space, such as a world map. A spatial SNS can be built where people can post and view comments via mixed or augmented reality in the form of avagraphs at landmark locations in the real world.

1.5.4 Email

Combining avagraph with an email protocol like SMTP, a holographic mail app can be crafted to show messages as holograms.

1.5.5 Message system

A message board can be implemented by recording and sharing avagraphs. A more complex interactive discussion system can be developed and commonly used by many applications.

1.5.6 Games and chat apps

Where it's reasonable to place a holographic avatar in a game or chat app, avagraph will be an ideal game object there, e.g., avagraph can be used to introduce game mechanics to players, replacing wizards that use text and images.

1.5.7 Spatial e-commerce

Avagraph can serve as a virtual shopping guide to help you shop in physical or virtual stores. Customers can use avagraphs to ask questions or leave reviews in the store.

1.5.8 Spagraph - holographic scene

Spagraph is a data structure of a holographic scene which contains avagraph and other elements that make up the whole scene. It can be used to create 6DOF lightly interactive scenes such as comics, animations, movies, and videos.

1.5.9 Spatial web

Spatial web is a concept based on World Wide Web. It uses URI and HTTP for location and communication, while HTML is extended to create the immersive experience. Avagraph is a component of the extension where it can be used via a special tag.

1.5.10 Spatial UI

Spatial UI is a user interface built on the spherical coordinate system for spatial computing where an immersive stereoscopic effect can be used to implement applications suitable for sitting or standing modes such as news, office, and video sharing software.

1.5.11 **Robotics**

Robotics uses humanoid shapes, kinematics and voices in a way like a physical mirror of avagraph, so it’s reasonable and logical for avagraph to underpin robotics.

1.5.12 **Optical holography**

The input and output of optical holography adopt optical principles. It’d be more convenient to utilize the captured information if the wavefront of the storage medium for optical holography can be converted to avagraph, because it’s a digital storage format.

Chapter 2

Spagraph - A Data Model of Holographic Scenes for Spatial Computing

2.1 Introduction

2.1.1 2D video

Such videos are pixel arrays captured by a 2D camera, and no dimensional information is found in the data file. This type of video format is used in many popular computer programs such as Youtube and Tiktok.

2.1.2 3D video

This type of video is also a collection of pixels, but instead of being captured by a 2D camera, it's captured by a specially designed dual-lens camera (such as Canon's dual fish eye lens camera series). Unlike a 2D video, a 3D video uses a collection of pixels for each eye to achieve the perspective effect. 3D movies like Avatar and many VR180[7] videos make heavy use of this video format to provide viewers with more immersive content.

2.1.3 Wide angle

In addition to improving dimensional information, we can also increase immersion by increasing the view angle. For example, panoramic, VR180, and 360-degree spherical videos[8] allow you to see the environment around the viewer, providing a more immersive feeling.

2.1.4 Spagraph

Using a 3D video format and the increased view angle, we can achieve a relatively immersive effect on the video, but the viewer can only rotate his head and cannot move in the positional dimension. It feels very unnatural or unreal, as you can move freely in any direction while rotating your head in the real world in real life.

Just like the perspective effect of 3D video is achieved by saving a collection of pixels for each eye, we need more dimensional information to get a more realistic feeling.

Traditional 3D scene formats are often used by 3D modeling software as intermediate data files to generate pixel arrays that are ultimately economically valuable in animations, movies, TV shows, and social media videos, or by game engines to enable highly interactive experiences.

In spatial computing, we should use a new format similar to them as the final output format, because it can help us utilize 6DOF[9] movements to achieve the most natural immersion, so it has the potential to become an economically valuable spatial video format.

We propose a new data format called spagraph, which is essentially a holographic scene for spatial computing containing world information and avagraphs to achieve 6DOF immersive video effects.

2.2 Characteristics

2.2.1 Asynchronous

All the data in spagraph is asynchronously persisted, retrieved only when it's time to transfer or display the information contained in the data.

2.2.2 Holographic

The retrieved data from spagraph must be processed in a way that the final output received by both eyes would present a 6DOF viewing experience.

2.2.3 Extensible

Not only can spagraphs cater for the emerging XR, but also will increment new application genres for 2D platforms like mobile computing, as soon as it meets the asynchronous and holographic properties. In a sense, all information having those properties can be scoped to spatial computing.

2.3 Data structure

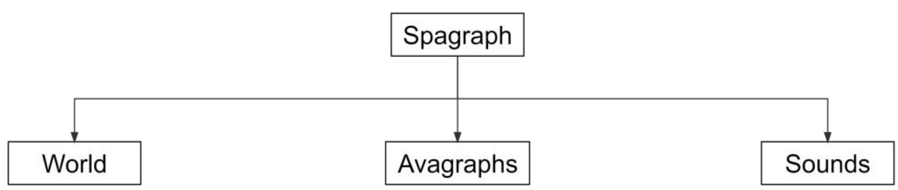

Figure 3The data structure for spagraph

2.3.1 World

The world is the background in the spagraph. It can be a skybox or more complex environment including various properties.

2.3.2 Avagraphs

Avagraph is the most important property in spagraph. Spagrahs usually have many avagraphs located at many different locations within them.

2.3.3 Sounds

Spagraph features spatial sounds distributed at different locations within it.

2.4 Comparison to relevant formats

2.4.1 glTF

It is possible to put 3D assets such as skyboxes or sounds into glTF, but it is not meant for a 6DOF video format and lacks avagraph.

2.4.2 OpenUSD

Although OpenUSD[10] is an animation scene format, it is an intermediate format mainly used in Pixar's 3D animation production software for generating pixel arrays of 3D animation. It is also used as a front-end application format in Reality Composer Pro, and Nvidia uses it as an interchange format for many 3D industrial applications. Still, it lacks the concept for avagraph and is not specifically designed for 6 DOF video formats such as spagrah.

2.5 Model Data collection

2.5.1 Photogrammetry

This information can be modeled based on images captured using photogrammetry[11] techniques and then used in spagraphs.

2.5.2 Generative Artificial Intelligence

Generative Artificial Intelligence[12] is a kind of machine learning algorithm that can generate the models from several images of the objects represented by the generated models which would be used in spagraphs.

2.5.3 Depth sensors

Using time-of-flight[13], depth sensors can capture depth images or point clouds that can be converted into mesh-based models through surface reconstructions. These converted models can be used in spagraphs.

2.5.4 Modeling software

Creators and designers can use any modeling software to create models of any world, property, and avatar concept for use in spagraphs.

2.6 Application scenarios

2.6.1 6DOF spatial capture

In interactive games or XR avatar chat applications, we can record the process and capture the 6DOF spatial experience.

2.6.2 6DOF Vtuber

New applications with 6DOF immersive experiences can be developed by recording the avagraphs of players individually or simultaneously and merging them into a single spagraph.

2.6.3 6DOF light entertainment

Any light entertainment content from movies to TV series can be created in spagrahs to provide a 6DOF immersive experience. For example, the movie Avatar can be remade to provide a 6DOF version to audiences.

2.6.4 Spatial applications

In Spatial UI, applications can present a stereoscopic view for any spagragh, which can play a 6DOF immersive experience after the user triggers the play button.

Chapter 3

SUI - A User Interface for Spatial Computing

3.1 Introduction

In the concept of SUI, we focus on the arrangement, layout, and form of UI elements in the XR space. Although we favor the most natural and direct input methods such as the coordination between hand gesture and eye motion featured by Apple visionOS[14], we don't discuss any input concept in this essay.

3.1.1 Meta Quest universal menu

Meta Quest universal menu[15] is the user interface of Meta Quest series. It can show more than three panels in the space. The content of the panel is generally a regular Android or built-in 2D app. It has no multitasking for 3D applications and stereoscopic effect for 2D apps. Since it doesn't establish a full spatial UI system, we can't build a spatial application using the standard 3D UI controls.

3.1.2 Apple visionOS

Apple visionOS is the operating system of Apple Vision Pro. It can display 3 or more flat windows with status bar and menu bar on the side. It supports multitasking for 3D applications by running multiple applications in a single space where you can run at least one 3D volume application. The 2D applications in the panel are generally iOS apps rebuilt for visionOS. It's better than Meta Quest universal menu in that it supports multitasking for 3D applications. Still, it has no stereoscopic effect yet for non volume and non fully immersive applications, and has no standard 3D UI controls for all apps.

While Reality Composer Pro can help build 3D volume and fully immersive applications, it doesn't establish a UI system of 3D applications and has no standardized 3D controls for building a 3D application. Reality Composer Pro is more suitable for building an entertainment experience instead of a serious application.

3.1.3 SUI

Although the current UI systems have utilized the XR space by surrounding the user with multiple panels, the experience ends up more like working with more computer monitors through the help of a single XR device. If it's intended to be used in such a way, it will always be better to use real monitors because of the most realistic experience provided by physical units. You never beat the original experience of a device by simulation.

The application in a panel is essentially an iOS or Android app built on a 2D UI system, so it can't offer a stereoscopic and immersive experience simply from the panel laid out in a spatial manner.

People expect an immersive experience when wearing XR devices. VR games have proved it by presenting the most immersive experience and winning a large share of gaming consoles. For applications, we should also preserve what we’ve focused on in games, which is to offer an immersive experience to users.

We propose a UI scheme for spatial computing called SUI or Spatial User Interface. In the system of SUI, we emphasize the importance of stereoscopic vision, the use and standardization of 3D UI controls, and 3D application multitasking. SUI focuses on and expresses a 3D stereoscopic and immersive experience. We intent to build an immersive UI system that facilitates the adoption of XR in the application domain and extends the utilization of XR outside of games and chat (attributing to a conference call) applications.

3.2 Characteristics

3.2.1 Immersive

The system layout of SUI based on the spherical coordinate system[16] and multiple spheres take full advantage of the XR space from three dimensions. Serving as the bounding frame of a 3D spatial application, a panel is tangent plane on a sphere and can be a cylindrical or rectangular surface. Instead of directly embedding iOS or Android apps in panels, we use 3D layout and controls in each panel for building 3D spatial applications. As the 3D layout has one more dimension than 2D and the 3D controls are inherently stereoscopic, an enhanced immersion over the 2D apps will be obtained.

3.2.2 Productive

Through use of multiple spheres, the sphere where active applications are located will be brought to the front and pushed back when the applications are no longer active within. In a sphere, multiple 3D applications are arranged in panels on the surface of the sphere and moved freely by the user through dragging and dropping the specific panel. Since SUI can accommodate more apps and the stereoscopic style is more intuitive, people will feel more powerful and productive when using SUI.

3.2.3 3DOF

When people are using a 3D spatial application, usually they are sitting or standing without having to move in 6DOF. The spherical coordinate system is always user-centered, which is similar to the experience of viewing 3DOF VR180 or 360 videos. SUI is built on the idea of 3DOF so we choose the spherical coordinate system instead of the Cartesian coordinate system[17].

3.3 Architecture

When we are using a command-line interface (CLI), we operate information in one dimensional text lines. When we are using a graphical user interface (GUI), we operate information on two dimensional bitmaps. When we are using a spatial user interface (SUI), we should operate information on three dimensional meshes.

Unlike games, applications require a consistent style and appearance across the system to increase efficiency in development and use of them. We need a specific infrastructure to express the meaning of 3D meshes and provide specific functionality for each special mesh. Thus, we designed an architecture based on the spherical coordinate system for SUI.

3.3.1 Spherical coordinate system

GUI is based on the Cartesian coordinate system, whereas SUI is based on the spherical coordinate system. In the application of the spherical coordinate system on SUI, we organize information in panels each of which is a tangent plane on the sphere with the user as the origin. The position of a panel is determined by the 3-tuple (ρ, θ, and ϕ) which gives the radius (or radial distance), inclination (or polar angle), and azimuth (or azimuthal angle) of the contact point of the panel on the sphere.

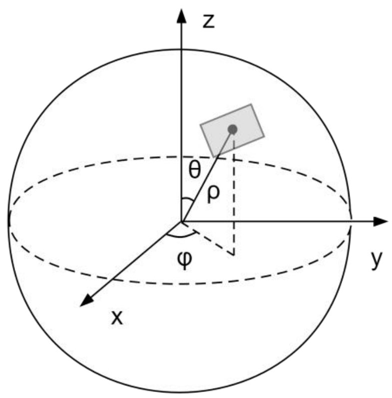

Figure 4Spherical coordinate system

The spherical coordinate system is helpful for understanding the spatial position of an application, but it's more useful to use the Cartesian coordinate system for computing. Cartesian coordinates (x, y, z) and the rotation quaternion (q) can be obtained from the spherical coordinates (radius ρ, inclination θ, azimuth ϕ), where $\rho \in [0, \infty)$, $\theta \in [0, 180]$, $\phi \in [0, 360)$.

$$x = \rho * \sin(\theta * 2\pi / 360) * \cos(\phi * 2\pi / 360)$$
$$y = \rho * \sin(\theta * 2\pi / 360) * \sin(\phi * 2\pi / 360)$$
$$z = \rho * \cos(\theta * 2\pi / 360)$$

$$ct = \cos(\theta * 2\pi / 360 / 2)$$
$$cp = \cos(\phi * 2\pi / 360 / 2)$$

st = sin(θ * 2π / 360 / 2)
sp = sin(ϕ * 2π / 360 / 2)
q = (cp * ct, -sp * st, cp * st, sp * ct)

3.3.2 Multi-sphere

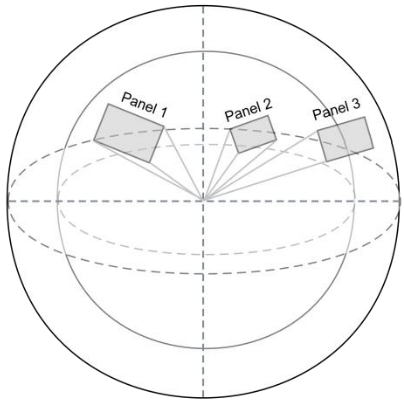

Figure 5Multiple spheres

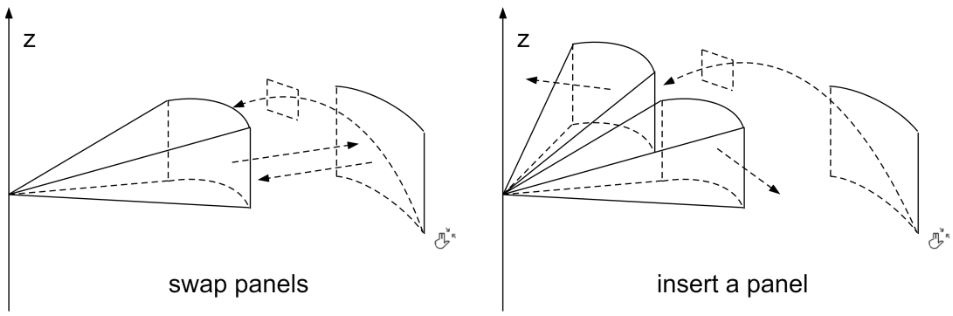

Figure 6Panel movements between spheres

We can have multiple spheres by specifying a distinct length for the radial radius for each sphere in the spherical coordinate system. The boon of having multiple spheres is that we can utilize the space to the extent that we will be more productive in spatial computing than we are in the 2D computing. Applications might be placed on different spheres, and a sphere can be brought to the front to enhance the viewing experience. Applications in each sphere can actively run or hibernate to reduce the power consumption. Theoretically infinite spheres can be used, but in practice, three application spheres plus a hibernation sphere and an exit sphere should be enough. We can change the sphere where an application is located by simultaneously pinching the thumb, index, and middle finger on one hand, dragging the application, and dropping it to another sphere. When an application is dropped on an empty area, it will take up the area. When it's dropped on another application, they will be swapped with each other even though they are from different spheres.

3.3.3 Panel shapes

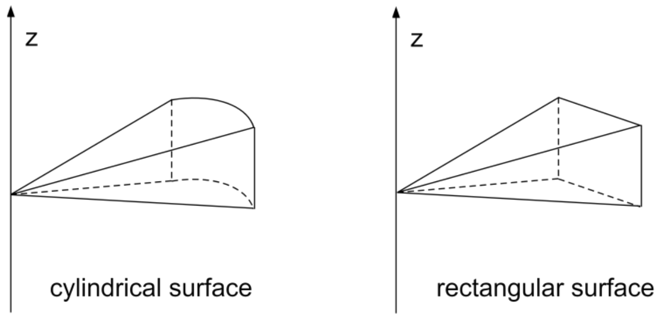

Figure 7Panel shapes

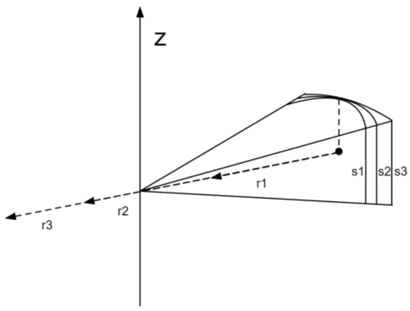

Figure 8The relationship between the curvature and radius of a cylindrical surface

As a tangent plane on the sphere, a panel can be a cylindrical or rectangular surface, and the two can be used interchangeably. The curvature of a cylindrical surface depends on the length of the cylinder radius (r) that passes through both the contact point of the surface on the sphere and the origin of sphere. The longer the radius, the flatter the surface. It's suggested that panels gradually flatten and shrink when getting closer to the user and vice versa.

3.3.4 Application layout

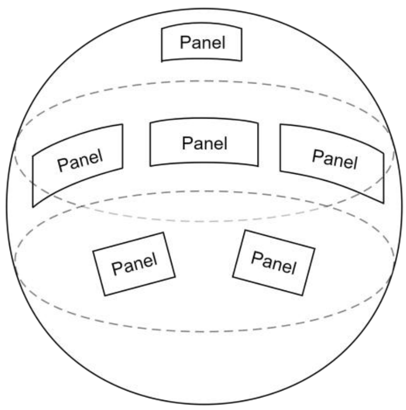

Figure 9Application layout

The surface of a sphere is divided into three parts, which are the upper section, the middle section, and the lower section. In each section, a collection of applications are arranged side by side. The most important applications are located in the middle section for normal use, moved into the lower section to relieve the neck fatigue, or placed in the upper section to save more work space.

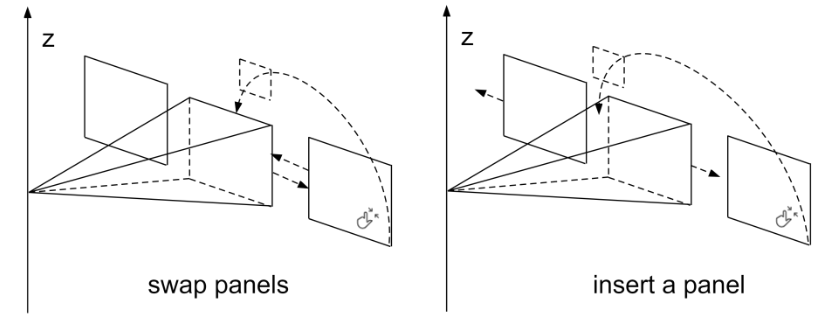

Figure 10Panel movements inside a single sphere

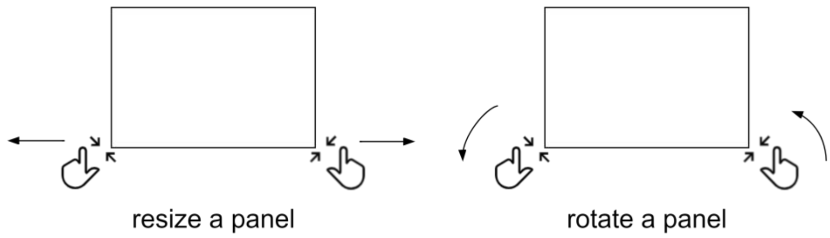

Figure 11Panel adjustments

The applications from each section can flow along the left or right direction by dragging the section to the left or right. In a single sphere, an application can be placed at an empty area on the surface or swap position with another one by pinching the thumb and index finger of one hand, dragging the application, and dropping it. An application can be resized by pinching the thumb and index finger of both hands and dragging towards outside or inside, or rotated by pinching the thumb and index finger of both hands and rotating both hands clockwise or counterclockwise.

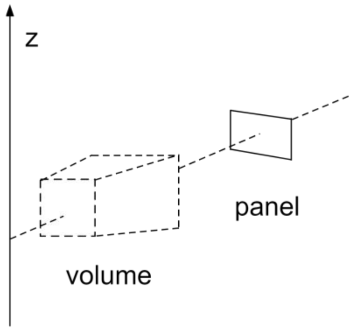

Figure 12The volumetric state of an application

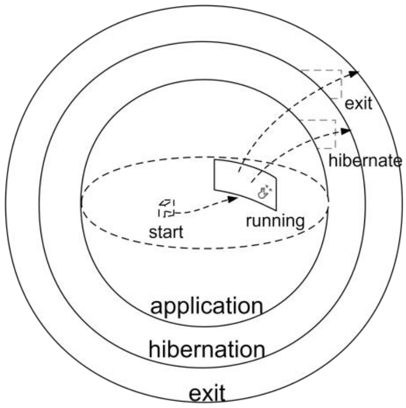

Figure 13Application life cycle

The active application can be brought to the front of a user as a volumetric application or take up the whole space to offer the most immersive experience. An application can start, run, hibernate, or exit. We can change the status of an application by dragging it and dropping it onto a special sphere. When the icon of an application is dragged and dropped onto any application sphere, it will enter the running state. When an application is dragged and dropped onto the hibernation sphere, it will enter the hibernation state. When it's dragged and dropped onto the exit sphere, it will exit from RAM.

3.3.5 Control layout

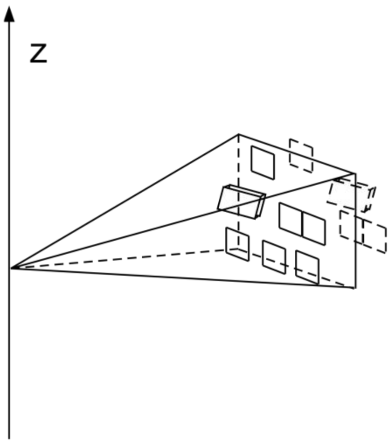

Figure 14Control layout

In an application, we can let the controls flow like what we generally do in a 2D layout with one more thing. Since SUI has one more dimension than a 2D UI, a control can have a z coordinate that defines the relative distance between the control and the panel it belongs to. Having a few controls with different z coordinates in an application will provide the application a distinct depth in each zone. If an animation is defined on the z coordinate of a control in an application, it will look more stereoscopic and vivid than the content in 2D applications.

3.3.6 3D controls

Common control elements in a 2D UI will have equivalent 3D counterparts in SUI. Controls will have a thickness to present a stereoscopic effect, e.g., 3D texts will look more solid and legible. Furthermore, We can inherit 2D elements for other use cases. As well as for the common controls, we can use controls that simulates physics to enhance the feeling of reality. Avagrah and spagraph can be used as 3D controls too.

3.3.7 Control standardization

The appearance and functionality of controls should be well defined, so people will feel familiar, comfortable, and efficient when using identical controls in the same system or even across multiple heterogeneous systems.

3.3.8 Stereoscopy

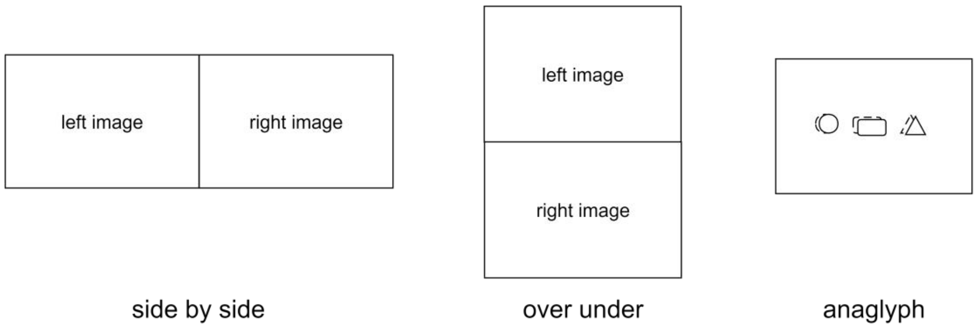

Figure 15Stereoscopic image

We should try to keep the controls immersive. For files with a complex data format whose content shouldn't be directly rendered without an explicit command, we can use a stereoscopic image as the placeholder, e.g., we can extract the pixels of a frame from a 360-degree video and present it as a stereoscopic image in the application so that people can see the stereoscopic preview of the video file.

3.3.9 General models

In order to extend the functionality, general models typically used in interactive experiences such as games and animations can be used in spatial applications. For example, when displaying goods on a spatial e-commerce application, general models are more expressive than pictures or videos. A general model doesn't have the well defined standard behavior when it's operated on, while a SUI control has the standard interaction behavior expected by the user whether it's a 3D model based control or any other control.

3.4 Application scenarios

3.4.1 3D controls libraries

To build a SUI system, we should at first build a 3D controls library as the toolkit before building the interaction system.

3.4.2 Spatial UI editors

A spatial UI editor can accelerate the UI composition and edition. The built-in 3D UI controls can be directly dragged and dropped onto the application layout.

3.4.3 Desktop IDE

A desktop IDE (or integrated development environment) can help people develop spatial applications on their desktop computers.

3.4.4 Spatial IDE

As a native development tool, the spatial IDE provides a WYSIWYG developing experience for spatial applications.

3.4.5 Spatial applications

Spatial applications can be applied in a variety of scenes and improve work efficiency in XR.

3.4.6 Spatial operating system (SOS)

SUI is a significant part of a spatial operating system which contains not only the UI system but also the input and interaction system.

Chapter 4

Spatial Web - An Open Web for Metaverse

4.1 Introduction

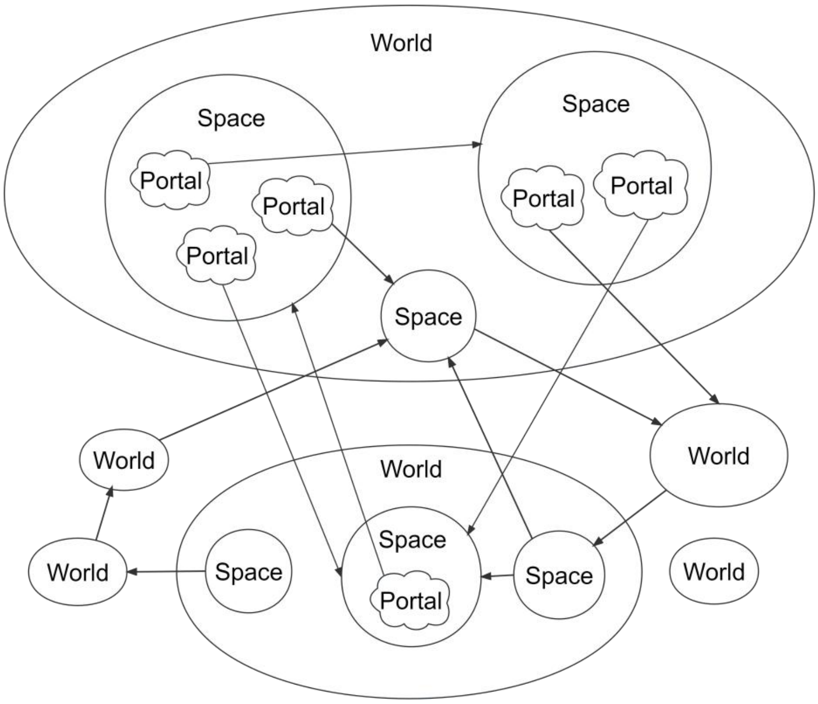

Figure 16The architecture for spatial web

4.1.1 The problem of an app-based metaverse

Most current XR applications implement their own metaverse[18]. People can use their avatars to explore the world within the app. One notorious problem is that if a group of users want to simultaneously transport from their current app to another app's world for entertainment, they can't easily do so. Assuming that at least one of them does not own or has not yet installed the specified application, they can’t fulfill their wishes.

4.1.2 Spatial Web - a web-based metaverse

To find a solution to the above issue, we proposed a web[19] based metaverse called spatial web. We use a hypertext[20] derived concept called web portal to represent the connection between two different worlds in the metaverse, so any group of users can simultaneously transport to another world created by anybody.

4.2 Characteristics

4.2.1 Atomic

The elements of a world in the spatial web can be accessed with a URI[21]. The elements are loosely coupled so it's very convenient to create or modify the space layout at will. One model used in one world can also be used on another world by using its URI address. A developer can quickly compose a new world using only models from other worlds.

4.2.2 Transient

When a world in the spatial web is accessed, it's the web space that is accessed. Since only the content of the very web space is downloaded to the local computer when it's accessed, the size of downloaded resource is much smaller than that of the whole world. Thus it's much faster to access a world. On the contrary, it's necessary to download the whole bundle of an app when you need to access only part of it at a moment.

4.2.3 Connected

By virtue of the concept of the web portal that is basically another hypertext, all of the worlds are connected together by portals to create a super large metaverse. A user can start from any one web space and transport to another web space of any world by directly walking into the web portal.

4.2.4 Open

Since spatial web is based on W3[22], it's accessible by the URI of any web space. We don't require a main hub such as an app store to find a specific web space. Also we don't have to care about which platform we are on, because we can access a world from anywhere using a browser.

4. 3 Components

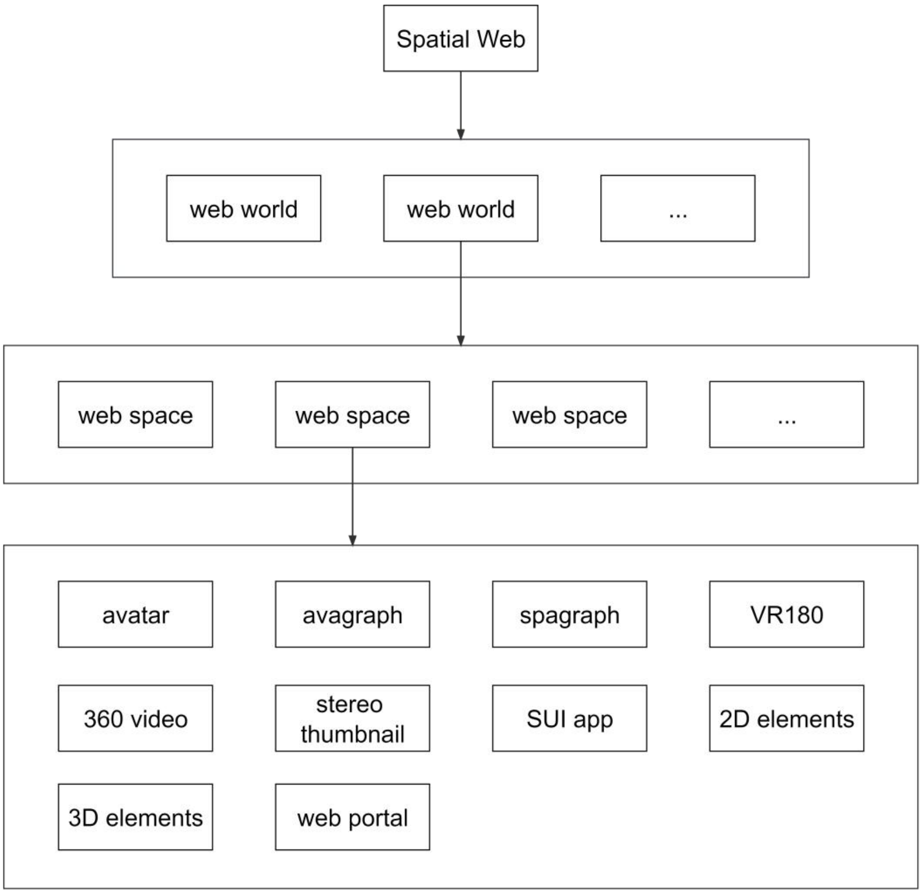

Figure 17The components for spatial web

4. 3. 1 Open avatar

When people enter the metaverse, it can render their avatars[23] or just only the controllers or virtual hands, depending on the scenes. When it's a scene that indeed requires showing avatars, we can use open avatar to retrieve the avatars from a third party provider. The principle of open avatar is similar to Oauth[24] which authenticates the user and offers the avatar for the requesting world. Thus we can transport from one world to the other with the same avatar.

4. 3. 2 Web world

A web world is an immersive world constituted by multiple web spaces. It's a spatial web site in the metaverse.

4.3.3 Web space

Web spaces are the basic units of the spatial web, serving the same role for spatial web as a normal web page does for W3. A world consists of several web spaces in the spatial web, so you're actually accessing the web space when you visit a world in the web-based metaverse.

4.3.4 Web portal

A web portal is actually the hyperlink[25] in the spatial web, but it's presented like a gate or portal in 3D, not just a highlighted string of text in W3. Any object in spatial web can become a web portal too, so it's plausible to transport to another web space by not only a regular web portal but also any portalized and highlighted object.

4.3.5 Extended elements

With extending the web with more elements, we can do more kinds of work in the metaverse except games and chat.

4.3.5.1 Avagraph

We as humans create characters and words, because we can't preserve our own holographic information. So we need methods to efficiently and compactly save what we say and do in some engravable surfaces, including stones, papers, and 2D electronic screens, of the surrounding.

Maybe because we are used to this method of communication in the development of thousands of years, it feels very natural for us to read text as the most basic information unit on 2D screens. However, it feels a bit inconsistent or disharmonious to read text on 2D panels in the metaverse. Since the metaverse is more holographic than 2D screens, we feel more comfortable to view holographic information other than to read text in the metaverse. Just like texts, avagraphes can be used to save and asynchronously transmit what humans say and do, but in a more holographic style. So we boldly propose to use avagraphs instead of texts as the basic communication unit in the metaverse.

We should use the avagraph to fill the most natural and holographic content in the metaverse. Without it, we can only find a few blank and lonely worlds, and wander aimlessly. That's maybe why most of people who've experienced the metaverse earlier don't understand it and even don't stay there any more. With it, we will have an asynchronous reality as what we have in the 2D screens, we can bypass the time barrier to communicate with each other and feel surrounded by more content.

4. 3. 5. 2 **Spagraph**

Just like avagraph can help solve the most basic problem of holographic information presentation and communication in the metaverse, spagrach can make people feel more comfortable when they are experiencing light entertainment including movie, TV series, and animations in a holographic style and appearance.

4. 3. 5. 3 **VR180 and 360-degree videos**

Although VR180 and 360-degree videos can offer only 3DOF experience for videos, they are more immersive than the 2D videos in the metaverse.

4. 3. 5. 4 **Stereoscopic thumbnails**

In order to recognize and organize elements in a holographic way in the metaverse, we should represent elements using stereoscopic thumbnails rather than 2D images.

4. 3. 5. 5 **Spatial UI applications**

As we mentioned above, we should be able to perform many more tasks in the metaverse. Except information communication, interactive games, and light entertainment, we can also do jobs like office work. There are also many other types of applications that have a potential to help us take more advantage of the metaverse, e.g., a holographic news or 6DOF video sharing applications.

4. 3. 5. 6 **The problem of synchronous reality**

In the current synchronous reality, developers pursue a 1:1 simulation of physical world in the application. Instead, in traditional 2D applications, developers prefer to use the power of computers to solve problems that people can’t easily solve in the real world. Advantages of spatial computing over PC and mobile computing is the ability to present holographic information and immerse humans in there, not more computing power. We should inherit what we do with the computing power of PCs and mobile computing, and combine it with the ability to render holographic information. If we only achieved 100% simulation of the physical world, there is no point in us using computers because we already have a perfect real world.

Chat, as the current most popular type of synchronous experience in the metaverse, has several prominent issues. There are misunderstandings about some aspects of it. Of course, it is different from traditional chat applications in terms of information presentation because it can display immersive environments and avatars, but it is actually also different in terms of workflow. Because it strives to perfectly simulate the physical world, it looks more like a conference call than what we achieve in 2D chat apps. In many ways, it can't even compete with 2D chat apps. In a sense, 2D chat is not synchronous at all, so it makes good use of the computer in every way and becomes more useful, even though it often appears to be synchronous.

Current chat apps typically display avatars of all participants in the same room, with people talking face-to-face in groups or one-on-one, making it look more like many private chats taking place in one room rather than a public chat where people expect to hear from everyone in the room but have no intention of knowing what others are doing in real time.

One problem is, when in the traditional 2D chat rooms, a message from a person in the room will be broadcast to everybody in the same room, which is very efficient and useful since each person who joins the room expects to receive more information from everyone else of the same room. That’s the purpose of a chat room. But in the current virtual metaverse, what a person says or does can only spread to a few people within an appropriate range around the speaker, which is less efficient than a 2D chat room.

The other problem is whether one person in the room really prefers to see the avatars of all other people in the same room, even those who are dozens of meters away in the world, and whether it’s actually necessary to render avatars that are even blocked off by room walls. In a 2D chat room, people can see the number of users in the room and find another specific user from the user list of the room. Since people don’t have to see all the other users in real time, there's no need to render all the profile images at once. This method, if used in the metaverse, will significantly reduce power consumption.

How about only rendering the avatars of other users within a specific range of the current user and transporting the speaking avatar to all participants in the room? People can choose to join a chat just like they do now in chat apps, broadcast to all users in the chat room, or even record an avagraph and send it to everyone. The advantage of this is that the avagraph message can be saved in the chat history, making it easier to find it later and have a more reasonable conversation.

One may argue that we can’t see who are actually in the room this way. The answer is that we can represent them in any more efficient way, e.g., using a sphere with a name on it, a stereoscopic thumbnail of the user’s profile image, or even a low poly version of an avatar to represent any user who is outside of a specific scope, but not too far yet. Instead of having to present representative elements for users who are far away, we can display a list of all users in a 3D style, like we do in a 2D chat room, to help find and locate them. When the current user gradually approaches another user and the distance between the two is less than a specific length, the user's avatar will gradually appear, while the user's avatar outside the effective range centered on the current user will gradually disappear.

This can significantly reduce unnecessary computing resource usage and power consumption, while also achieving the performance of traditional 2D chat rooms.

4.3.5.7 Synchronous reality or asynchronous reality?

Currently, most metaverse apps default to synchronous reality, which displays avatars of everyone in the world without even considering whether it's appropriate for them. It's a common, and very embarrassing, situation where you enter a metaverse app with nothing to do but find someone to chat with. While we could render everyone's avatars in the metaverse, do we need to? Do people always expect to see other users when they enter a world? Maybe, but maybe not. When we intend to chat with others, we enter chat rooms specifically built for that purpose, but when we just want to consume some content or communicate asynchronously, we don't expect to see other users' online avatars in real time. We'd better think hard about the purpose of a metaverse and carefully consider whether to display all avatars synchronously. Sometimes not showing other people's avatars can bring more benefits.

4.3.5.8 Traditional 2D elements

The spatial web is a web built on spatial computing, so it should be able to contain and render 2D elements such as texts, audios, images, videos, and even 2D htmls.

4.3.5.9 General 3D elements

3D Elements include primitive shapes and files in 3D format. The list of primitive shapes includes spheres, boxes, cylinders, and other basic shapes. The list of 3D formats includes glTF, VRM, USD, FBX, and OBJ, among others.

4.4 Relevant frameworks

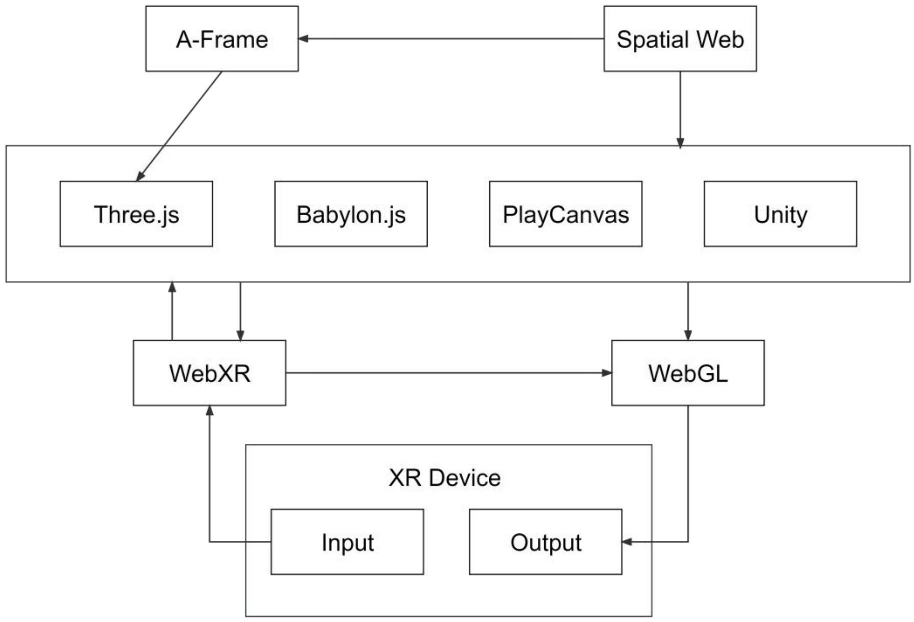

Figure 18The relationship between the spatial web and relevant frameworks

4.4.1 WebXR

WebXR is a web API that receives input from XR and sends output to XR with the help of WebGL, while the spatial web is an ideal way to build the universal web in spatial computing using the essential concept of W3. WebXR is the essential foundation for spatial web to take input and send output.

4.4.2 WebGL

WebGL is a JavaScript API for rendering 3D graphics. WebXR uses this to render screen images on XR displays. Several web graphic libraries use it to render scenes and models. It's required for spatial web.

4.4.3 Three.js, Babylon.js, PlayCanvas, Unity

These libraries or tools are high-level graphics API frameworks or game engines that can help you develop WebGL based applications or games more easily. They are mainly used to create interactive experiences such as games. They can adopt the concept of the spatial web to expand their application scenarios.

4.4.4 A-Frame

A-Frame is an entity component library based on Three.js and WebXR. Developers can use it to built XR experiences such as games. It can adopt the concept of the spatial web to expand their application scenarios.

4.5 Application scenarios

4.5.1 Spatial web browser

A spatial web browser presents immersive content of a web space by default. It can display the content of 2D html, but only in the position of the space. The 2D html can be part of a web space and can even be converted to a 3D version to display it similar to a spatial UI application.

4.5.2 Spatial web search engine

A typical 2D web search engine primarily searches and presents information in 2D formats, while a spatial web search engine searches for immersive content (such as avagraphes) and output it in an immersive form.

4.5.3 Web service

In the app-based metaverse, most assets are bundled in built packages. To increase efficiency and reduce application size, resources can be provided in web services, and applications can obtain and render them as needed.

4.5.4 Open avatar provider

Third-party avatar providers like Ready Player Me are one option.

Chapter 5

Conclusion

The computer provides synchronous reality (remote gathering) and asynchronous reality (memory expansion). On the command line and graphical user interface, people have created many synchronous and asynchronous programs to facilitate daily life and make it more convenient. As the spatial computing becomes more prominent, we've become adept at rolling out synchronous experiences, but there's an urgent need for asynchronous experiences. As a possible solution to the asynchronous problem of the spatial computing, avagraph is likely to open up new paths for it.

The media people use on computers has evolved many times, from text, audio, images to video, and many applications have emerged that use these formats as content carriers. As before, 6DOF video formats like Spagrah will continue this paradigm and hopefully become the next medium for new types of applications on computers.

SUI is a brand-new UI system design method that aims to maximize the use of the large space of XR and efficiently operate UI components in the space. We can get immersion and production at the same time.

The current metaverse is a bunch of apps and games that have no connection to each other. We think of the metaverse as an immersive world made up of loosely coupled information spaces connected by portals that people can move through at will. We learned from W3 and conceived a web-based solution called spatial web for building an open and connected metaverse. In addition, we have deeply thought about issues such as synchronous reality in the current metaverse and proposed specialized solutions.

References

[1]. Vannevar Bush, "As We May Think", 1945.

[2]. Greenwold, Simon, "Spatial Computing", MIT Graduate Thesis, June 2003. https://acg.media.mit.edu/people/simong/thesis/SpatialComputing.pdf

[3]. MICAH BLUMBERG, “Nick Whiting from Epic Games explains the origin of XR (XR = Virtual Reality + Augmented Reality)”, https://www.svgn.io/p/how-the-term-xr-was-coined-its-so-funny-hint-it-does-not-mean-extended-reality-4e7fff539676

[4]. Andreas Rene Fender and Christian Holz, “Causality-preserving Asynchronous Reality”, in Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, April 2022, Article No. 634, Pages 1–15. https://dl.acm.org/doi/pdf/10.1145/3491102.3501836

[5]. The Khronos® 3D Formats Working Group, “glTF™ 2.0 Specification”, https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.pdf

[6]. VRM Consortium, “VRM - 3D Avatar File Format for VR”, https://vrm.dev/en/

[7]. Google Open Source, "VR180 Video Format", Apr 10, 2018. https://github.com/google/spatial-media/blob/master/docs/vr180.md

[8]. Nielsen Frank, "Surround video: a multihead camera approach", in The Visual Computer, volume 21, pages 92–103, 01 February 2005.

[9]. Google for Developers, “Degrees of freedom”, September 21, 2018. https://developers.google.com/vr/discover/degrees-of-freedom

[10]. The Alliance for OpenUSD, “OpenUSD (Universal Scene Description)”. https://www.openusd.org/release/index.html

[11]. Albrecht Meydenbauer, "Die Photometrographie.", 1867. https://opus4.kobv.de/opus4-btu/files/749/db186714.pdf

[12]. Gavin Christopher Newsom and Shirley Nash Weber, "Executive Order N-12-23", Executive Department, State of California, September 6, 2023. https://www.gov.ca.gov/wp-content/uploads/2023/09/AI-EO-No.12-_-GGN-Signed.pdf

[13]. Analog Devices, Inc., “Time of Flight What is Time of Flight?”. https://www.analog.com/en/design-center/glossary/time-of-flight.html

[14]. Apple Inc., “Discover visionOS”, 2023 Worldwide Developers Conference, June 2023. https://developer.apple.com/visionos/

[15]. Meta Platforms, Inc., “Use the universal menu on Meta Quest”, October 2023. https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-2/universal-menu-for-quest-2-and-quest/

[16]. ISO, "ISO 80000-2:2019 Quantities and units – Part 2: Mathematics", pp. 20–21, Item no. 2-17.3.

[17]. Nykamp DQ, “Cartesian coordinates”, Math Insight. http://mathinsight.org/cartesian_coordinates

[18]. Neal Town Stephenson, "Snow Crash", Bantam Books, June 1992.

[19]. Sir Timothy John Berners-Lee, "Information Management: A Proposal", March 1989. https://cds.cern.ch/record/369245/files/dd-89-001.pdf

[20]. Theodor Holm Nelson, “Complex Information Processing: A File Structure for the Complex, the Changing, and the Indeterminate”, ACM '65: Proceedings of the 1965 20th national conference, August 1965. https://dl.acm.org/doi/pdf/10.1145/800197.806036
[21]. IETF, “Uniform Resource Identifiers (URI): Generic Syntax”, in RFC 2396, August 1998. https://datatracker.ietf.org/doc/html/rfc2396
[22]. Tim Berners-Lee and Robert Cailliau, “WorldWideWeb: Proposal for a HyperText Project”, 12 November 1990. http://cds.cern.ch/record/2639699/files/Proposal_Nov-1990.pdf
[23]. Neal Town Stephenson, "Snow Crash", Bantam Books, June 1992.
[24]. IETF, “The OAuth 1.0 Protocol”, in RFC 5849, April 2010. https://datatracker.ietf.org/doc/html/rfc5849
[25]. Theodor Holm Nelson, Project Xanadu, 1965