

# 空间计算的全息组件

耿怀宁

[jarvis.reality@gmail.com](mailto:jarvis.reality@gmail.com)

[www.holocom.org](http://www.holocom.org)

# 摘要

XR 的硬件已经发展到我们可以佩戴它们舒适地享受游戏体验的程度，但摆在我们面前的瓶颈是体验的种类相对于台式电脑和智能手机来说太少了。为了让新设备更好地服务人类，我们必须发明合适的软件架构范式，发现更具吸引力的场景，以扩展人们可以探索的应用领域。通过在空间计算中应用全息思维，我们构思并提出了几个软件组件来解决这个问题。使用这些组件开发的 XR 软件将涵盖广泛的空间应用，包括通信、轻娱乐、生产应用和网络应用。

在当前涉及虚拟化身的空间计算环境中，聊天应用和游戏占据了主要的应用场景。它们利用同步技术来捕获和传输数据以实现工作机制。像《Horizon Worlds》这样的应用程序就是一个完美的例子，它同步捕获玩家的动作和声音并将它们发送到远程玩家。虽然很多用户希望有类似的同步体验，但是它们远远不能满足人们对于空间计算的各种异步需求。为了显著扩展空间计算的场景，一种由化身模型、动作和声音组成的全息数据模型被提出以构建新的异步应用程序。

空间计算中最突出的视频格式是 VR180 和 360 度球形视频。一般 VR180 视频是用立体摄像头拍摄的，每只眼睛使用不同的图像来呈现立体视图，而 360 度视频通常是单眼视图。虽然两者都带来了传统 2D 视频所没有的沉浸感，但它们都只有 3 个自由度，这意味着观看者无法向任何方向移动以更深入地了解所观看的主题或其周围环境。为了提高空间沉浸感，我们提出了一种称为空间图的新视频格式，以实现 6 自由度观看体验。

当前的空间计算用户接口方案已经采用了大空间，其中通常至少 3 个二维面板排列在以观看者为中心的圆上。每个面板就像一个二维屏幕，通常会比实际情况放大以改善观看体验，有时会缩小以方便使用。在苹果 visionOS 中，包含三维对象的容积应用程序可以与二维应用程序一起放置在共享空间中，而游戏等更身临其境的体验可以占据整个空间。鉴于这些设计存在一些影响 XR 设备体验的缺点，为了充分利用空间计算，我们提出了一种称为 SUI 的用户接口方案来开发三维空间应用。

在当前的元宇宙迭代版本中，虽然一些 app（例如 Rec Room）已经在应用程序内部实现了元宇宙的部分功能，但当将所有 app 放在一起时，它们中的每一个都是一个信息孤岛。我们提出了一种基于 web 的解决方案，称为空间网，它连接每个信息空间以形成一个开放的元宇宙网络。基于 web 的元宇宙比基于 app 的元宇宙有更多优点。

# 目录

<b>1. 天图——用于空间计算的全息化身数据模型</b> .....	<b>7</b>
1.1 介绍 .....	8
1.1.1 同步现实和异步现实 .....	8
1.1.2 天图 - 全息化身 .....	8
1.1.3 天图即输入 .....	8
1.2 特征 .....	9
1.2.1 异步性 .....	9
1.2.2 全息性 .....	9
1.2.3 扩展性 .....	9
1.3 数据结构 .....	9
1.3.1 模型 .....	9
1.3.2 动作 .....	10
1.3.3 声音 .....	10
1.4 与相关格式的比较 .....	10
1.4.1 gITF .....	10
1.4.2 FBX .....	10
1.4.3 MP3 .....	10
1.5 应用场景 .....	10
1.5.1 即时通讯 .....	10
1.5.2 社交网络 .....	10
1.5.3 空间社交 .....	11
1.5.4 电子邮件 .....	11
1.5.5 留言系统 .....	11
1.5.6 游戏和聊天程序 .....	11
1.5.7 空间电商 .....	11
1.5.8 空间图 - 全息场景 .....	11
1.5.9 空间用户接口 .....	11
1.5.10 空间网 .....	12
1.5.11 机器人技术 .....	12
1.5.12 光学全息 .....	12
<b>2. 空间图——用于空间计算的全息场景数据模型</b> .....	<b>13</b>
2.1 介绍 .....	14
2.1.1 2D 视频 .....	14
2.1.2 3D 视频 .....	14
2.1.3 广角 .....	14
2.1.4 空间图 .....	14
2.2 特征 .....	15
2.2.1 异步性 .....	15
2.2.2 全息性 .....	15
2.2.3 扩展性 .....	15
2.3 数据结构 .....	15
2.3.1 世界 .....	15

2.3.2	天图	15
2.3.3	声音	15
2.4	与相关格式的比较	16
2.4.1	gITF	16
2.4.2	OpenUSD	16
2.5	模型数据采集	16
2.5.1	摄影测量	16
2.5.2	生成式人工智能	16
2.5.3	深度传感器	16
2.5.4	建模软件	16
2.6	应用场景	17
2.6.1	6 自由度空间捕捉	17
2.6.2	6 自由度虚拟偶像	17
2.6.3	6 自由度轻娱乐	17
2.6.4	空间应用程序	17
<b>3.</b>	<b>SUI——空间计算的用户接口</b>	<b>18</b>
3.1	介绍	19
3.1.1	Meta Quest 通用菜单	19
3.1.2	苹果 visionOS	19
3.1.3	SUI	19
3.2	特征	20
3.2.1	沉浸感	20
3.2.2	生产力	20
3.2.3	3 自由度	20
3.3	架构	20
3.3.1	球坐标系	21
3.3.2	多球体	22
3.3.3	面板形状	23
3.3.4	应用布局	23
3.3.5	控件布局	25
3.3.6	三维控件	25
3.3.7	控件标准化	25
3.3.8	立体视觉	26
3.3.9	一般模型	26
3.4	应用场景	26
3.4.1	三维控件库	26
3.4.2	SUI 编辑器	26
3.4.3	SUI 桌面集成开发环境	26
3.4.4	SUI 空间集成开发环境	27
3.4.5	空间应用	27
3.4.6	空间操作系统 (SOS)	27
<b>4.</b>	<b>空间网——元宇宙的开放网络</b>	<b>28</b>
4.1	介绍	29
4.1.1	基于 app 的元宇宙问题	29

4.1.2 空间网 - 基于 web 的元宇宙 .....	29
4.2 特征 .....	29
4.2.1 原子性 .....	29
4.2.2 瞬时性 .....	30
4.2.3 互联性 .....	30
4.2.4 开放性 .....	30
4.3 组件 .....	31
4.3.1 开放化身 .....	31
4.3.2 web 世界 .....	31
4.3.3 web 空间 .....	31
4.3.4 web 传送门 .....	32
4.3.5 扩展元素 .....	32
4.3.5.1 天图 .....	32
4.3.5.2 空间图 .....	32
4.3.5.3 VR180 和 360 度视频 .....	32
4.3.5.4 立体缩略图 .....	33
4.3.5.5 空间 UI 应用程序 .....	33
4.3.5.8 传统二维元素 .....	34
4.3.5.9 一般三维元素 .....	34
4.4 相关框架 .....	35
4.4.1 WebXR .....	35
4.4.2 WebGL .....	35
4.4.3 Three.js, Babylon.js, PlayCanvas, Unity .....	35
4.4.4 A-Frame .....	35
4.5 应用场景 .....	36
4.5.1 空间网浏览器 .....	36
4.5.2 空间网搜索引擎 .....	36
4.5.3 Web service .....	36
4.5.4 开放化身提供商 .....	36
<b>5. 结论 .....</b>	<b>37</b>
<b>参考文献 .....</b>	<b>39</b>

# 图列表

图 1 天图架构 .....	8
图 2 天图数据结构 .....	9
图 3 空间图数据结构 .....	15
图 4 球坐标系 .....	21
图 5 多球体 .....	22
图 6 在球体之间移动面板 .....	22
图 7 面板形状 .....	23
图 8 圆柱面曲率与圆柱半径 $r$ 的关系 .....	23
图 9 应用布局 .....	23
图 10 在单个球体内移动面板 .....	24
图 11 调整面板 .....	24
图 12 应用程序的容积状态 .....	24
图 13 应用程序的生命周期 .....	25
图 14 控件布局 .....	25
图 15 立体图像的常见格式 .....	26
图 16 空间网架构 .....	29
图 17 空间网组件 .....	31
图 18 空间网与相关框架的关系 .....	35

# 第一章

天图——用于空间计算的全息化身数据模型

## 1.1 介绍

### 1.1.1 同步现实和异步现实

万尼瓦尔·布什创造了记忆扩展<sup>[1]</sup>，以概念化特殊设备中的信息存储并补充人类记忆，这些记忆通过计算机在命令行和图形系统中实现。正如之前的计算机科学家为从键盘输入的文本创建了 ASCII 和 Unicode，为从相机拍摄的图像创建了 JPEG，为从麦克风录制的声音创建了 MP3，为从摄影机拍摄的视频创建了 MP4 一样，我们应该为空间计算<sup>[2]</sup>发明一种新的数据格式来保存从 XR<sup>[3]</sup>输入传感器捕获的数据。

与二维计算一样，空间计算可以支持同步和异步场景。在空间计算的 XR 领域，聊天、游戏等同步应用已经探索到了相当的水平，克服了空间障碍，将远程用户聚集在一起。然而，为了更好地使人们能够相互交流，数据需要被保存和读取，这样才不会受到时间的阻碍。安德烈亚斯·雷内·芬德等人为此提出了异步现实<sup>[4]</sup>的概念，但并没有给出详细的软件解决方案。为了克服时间约束，我们提出了一种称为天图的新模型作为所需数据的载体。

### 1.1.2 天图 - 全息化身

虚拟化身作为 XR 中最突出的数据模型，是角色数据的重要组成部分。与二维计算中的数据格式不同，角色数据有潜力成为 XR 中的基本信息单元。角色的模型、动作和声音可以融合成一种叫做天图的全息数据结构，用于保存、读取、编辑、传输和显示全息化身。

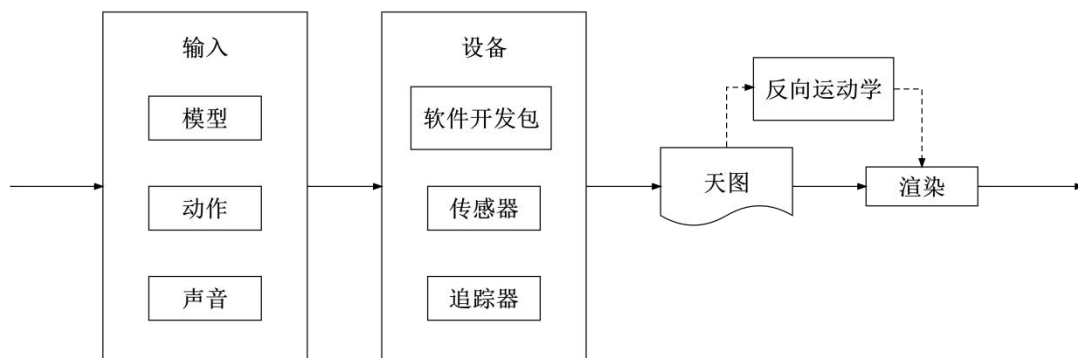


图 1 天图架构

虽然物理世界中的全息技术使用干涉和衍射原理来保存和读取从物理对象表面反射的光的波前，但在 XR 中可以在天图上利用骨骼绑定、蒙皮和反向运动学来实现类似的全息效果。

### 1.1.3 天图即输入

XR 的基本输入设备是跟踪器和传感器，包括头戴和控制器手柄上的传感器。由于动作和声音是 XR 最自然的输入方式，因此我们应该从这些特定的输入设备捕获数据，将这些数据保存在叫做天图的特殊类型的数据结构中，并将其作为基本信息单元读出并显示。



## 1.2 特征

### 1.2.1 异步性

天图中的所有数据都是异步持久化的，仅在传输或显示数据中包含的信息时才进行检索。

### 1.2.2 全息性

从天图检索到的数据必须以双眼接收到的最终输出呈现 6 自由度观看体验的方式进行处理。

### 1.2.3 扩展性

天图不仅可以满足新兴的 XR 需求，而且一旦满足异步性和全息性，它还可以为移动计算等二维平台增加新的应用类型。从某种意义上说，具有这些属性的所有信息都可以纳入空间计算的范围。

## 1.3 数据结构

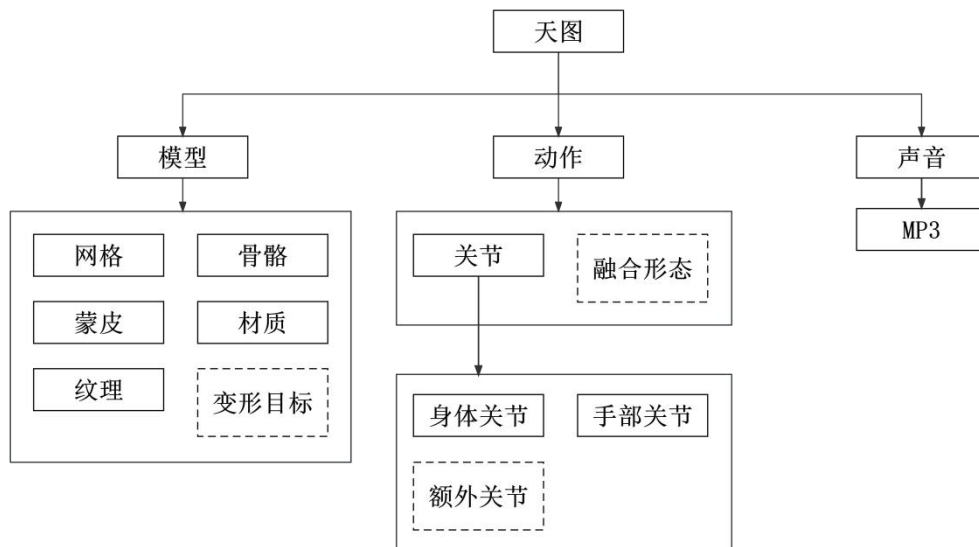


图 2 天图数据结构

### 1.3.1 模型

理论上天图可以兼容所有模型格式，例如 glTF<sup>[5]</sup>、VRM<sup>[6]</sup>和其他 3D 模型数据格式。只需要由实现方来分析并使用具体的数据格式进行渲染和呈现。模型数据将放入天图文件中或通过 URI（统一资源标识符）引用外部文件。

## 1.3.2 动作

动作数据由面部和身体部位组成。面部动作包含角色面部表情的混合形状。身体动作包括身体和手的骨骼关节的位置和方向。

## 1.3.3 声音

它可以是任何音乐、配乐或角色的声音。

## 1.4 与相关格式的比较

天图的核心原则是它应该兼容尽可能多的模型格式。

### 1.4.1 glTF

虽然 glTF 是引擎和应用程序使用的常见 3D 模型格式，但还有许多其他 3D 模型格式，例如 OBJ 和 glTF 派生的 VRM 等。动作和声音可以作为 glTF 的扩展添加，但 glTF 不太可能与其他模型格式兼容。最好使用 glTF 作为通用模型格式而不是全息数据格式。

### 1.4.2 FBX

FBX 通常包含模型和骨骼动画数据。尽管它包含除声音之外的模型和动作，但它是一种专有文件格式，也不太可能与其他模型格式兼容。

### 1.4.3 MP3

MP3 是一种数字音频编码格式，因此它不能替代天图。

## 1.5 应用场景

### 1.5.1 即时通讯

天图可以替代即时通讯应用中的文本。天图作为全息通信的基本信息单元，能够带来最自然的输入输出体验。

### 1.5.2 社交网络

用户可以选择一个模型，并将他们的动作和声音记录在天图文件中，以便与其他用户共享。

他们还可以手动制作复杂的动画，并将其与配乐混合到可共享的天图文件中。

### 1.5.3 空间社交

天图可以根据小空间（例如房间）或大空间（例如世界地图）中的位置进行共享。可以建立一个空间社交应用，人们可以通过混合或增强现实以天图的形式在现实世界的地标位置发布和查看评论。

### 1.5.4 电子邮件

将天图与 SMTP（简单邮件传输协议）等电子邮件协议相结合，可以制作全息邮件应用程序以将消息显示为全息图。

### 1.5.5 留言系统

留言板可以通过记录和共享天图来实现。可以开发更复杂的交互式讨论系统并由许多应用程序普遍使用。

### 1.5.6 游戏和聊天程序

如果可以合理地将全息化身放置在任何游戏或聊天应用程序中，天图将是其中的理想游戏对象。例如，天图可用于向玩家介绍游戏机制，以替代文本和图像向导。

### 1.5.7 空间电商

天图可以成为虚拟导购，在实体或虚拟商店中为您提供帮助。顾客可以使用天图在商店中提出问题或留下评论。

### 1.5.8 空间图 - 全息场景

空间图是一种全息场景的数据结构，包含天图和构成整个场景的其他元素。它可以用来创建 6 自由度的轻交互场景，包括漫画、动画、电影和视频等。

### 1.5.9 空间用户接口

空间用户接口是一个基于球坐标系构建的用于空间计算的用户界面，其中沉浸式立体效果可用于实现适合坐着或站立姿势使用的应用程序，例如新闻、办公和视频分享软件等。

## 1.5.10 空间网

空间网是一个基于万维网的概念。它使用 URI（统一资源标识符）和 HTTP（超文本传输协议）进行定位和通信，同时扩展了 HTML（超文本标记语言）以创造沉浸式体验。天图是该扩展的一个组成部分，可以通过特殊标签来使用它。

## 1.5.11 机器人技术

机器人技术就像天图的物理镜子一样使用人形形状、运动学和声音，因此天图支撑机器人技术是合理且合乎逻辑的。

## 1.5.12 光学全息

光学全息的输入和输出均采用光学原理，如果能将光学全息存储介质的波前转换为天图，则可以更方便地利用捕获的信息，因为它是一种数字存储格式。

# 第二章

空间图——用于空间计算的全息场景数据模型

## 2.1 介绍

### 2.1.1 2D 视频

此类视频是 2D 相机捕获的像素阵列，并且在数据文件中找不到维度信息。这种类型的视频格式用于许多流行的计算机程序，例如 Youtube 和 Tiktok。

### 2.1.2 3D 视频

这种类型的视频也是像素阵列，但不是由 2D 相机捕获，而是由专门设计的双镜头相机（例如佳能的双鱼镜头相机系列）捕获。与 2D 视频不同，3D 视频为每只眼睛使用单独的像素阵列来实现透视效果。《阿凡达》等 3D 电影和许多 VR180<sup>[7]</sup> 视频大量使用这种视频格式，为观众提供更加身临其境的内容。

### 2.1.3 广角

除了改善维度信息之外，我们还可以通过广角视图来提升沉浸感。例如，全景视频、VR180、360 度球形视频<sup>[8]</sup> 可以让您看到观看者周围的环境，提供更身临其境的感觉。

### 2.1.4 空间图

利用 3D 视频和广角视图，我们可以在视频上实现相对沉浸的效果，但观看者只能旋转头部，无法在位置维度上移动。这感觉非常不自然或者不真实，因为在现实世界中你可以在旋转头部的同时向任何方向自由移动。

就像 3D 视频的立体效果是通过为每只眼睛保存单独的像素阵列来实现的一样，我们需要更多的维度信息才能获得更真实的感觉。

传统的 3D 场景格式通常被 3D 建模软件用作中间数据文件，以生成最终在动画、电影、电视节目和社交媒体视频中具有经济价值的像素阵列，或者被游戏引擎用来实现高度交互的体验。

在空间计算中，我们应该使用与它们类似的新格式作为最终的输出格式，因为它可以帮助我们利用 6 自由度<sup>[9]</sup> 的运动来实现最自然的沉浸感，因此它有潜力成为一种具有经济价值的空间视频格式。

我们提出了一种名为空间图的新数据格式，它本质上是包含世界信息和天图的用于空间计算的全息场景，以实现 6 自由度沉浸式视频效果。

## 2.2 特征

### 2.2.1 异步性

空间图中的所有数据都是异步持久化的，仅在传输或显示数据中包含的信息时才进行检索。

### 2.2.2 全息性

从空间图检索到的数据必须以双眼接收到的最终输出呈现 6 自由度观看体验的方式进行处理。

### 2.2.3 扩展性

空间图不仅可以满足新兴的 XR 需求，而且一旦满足异步性和全息性，它还可以为移动计算等二维平台增加新的应用类型。从某种意义上说，具有这些属性的所有信息都可以纳入空间计算的范围。

## 2.3 数据结构

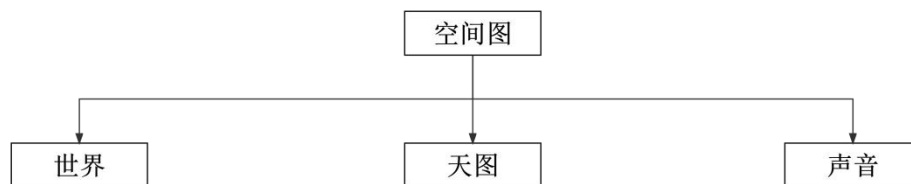


图 3 空间图数据结构

### 2.3.1 世界

世界是空间图中的背景。它可以是一个天空盒或更复杂的包括各种道具的环境。

### 2.3.2 天图

天图是空间图中最重要的属性。空间图通常有许多天图位于其中的许多不同位置。

### 2.3.3 声音

空间图的空间声音分布在其中的不同位置。

## 2.4 与相关格式的比较

### 2.4.1 glTF

可以将天空盒或声音等 3D 资源放入 glTF，但它不适合用作 6 自由度视频格式，并且缺少天图。

### 2.4.2 OpenUSD

OpenUSD<sup>[10]</sup>虽然是一种动画场景格式，但它是一种中间格式，主要用于皮克斯的 3D 动画制作软件中，用于生成 3D 动画的像素阵列。它还在 Reality Composer Pro 中用作前端应用程序格式，而英伟达将其用作许多 3D 工业应用程序的交换格式。尽管如此，它仍然缺乏天图的概念，也不是专门为 6 自由度视频格式（例如空间图）设计的。

## 2.5 模型数据采集

### 2.5.1 摄影测量

该信息可以基于使用摄影测量<sup>[11]</sup>技术捕获的图像进行建模，然后用于空间图。

### 2.5.2 生成式人工智能

生成式人工智能<sup>[12]</sup>是一种机器学习算法，它可以从生成的模型所代表的对象的多个图像生成模型，这些模型将在空间图中使用。

### 2.5.3 深度传感器

使用飞行时间<sup>[13]</sup>，深度传感器可以捕获深度图像或点云，这些图像或点云可以通过表面重建转换为基于网格的模型。这些转换后的模型可以在空间图中使用。

### 2.5.4 建模软件

创作者和设计师可以使用任何建模软件来创建任何世界、道具和化身概念的模型，以在空间图中使用。



## 2.6 应用场景

### 2.6.16 自由度空间捕捉

在互动游戏或 XR 化身聊天应用中，我们可以记录过程并捕捉 6 自由度空间体验。

### 2.6.26 自由度虚拟偶像

具有 6 自由度沉浸式体验的新应用程序可以通过单独或同时记录玩家的天图并将它们合并到单个空间图中来开发。

### 2.6.36 自由度轻娱乐

从电影到电视剧的任何轻娱乐内容都可以在空间图中创建，以提供 6 自由度沉浸式体验。例如，电影《阿凡达》可以重新制作，为观众提供 6 自由度版本。

### 2.6.4 空间应用程序

在空间用户接口中，应用程序可以为空间图呈现立体视图，在用户触发播放按钮后可以播放 6 自由度沉浸式体验。

# 第三章

## SUI——空间计算的用户接口

## 3.1 介绍

在 SUI 的概念中，我们关注的是 XR 空间中用户接口元素的排列、布局 and 形式。尽管我们更喜欢最自然、最直接的输入方式，例如苹果 visionOS<sup>[14]</sup>所提供的手势与眼动协调的输入方式，但我们在本文中不讨论任何输入概念。

### 3.1.1 Meta Quest 通用菜单

Meta Quest 通用菜单<sup>[15]</sup>是 Meta Quest 系列的用户接口。它可以在空间中显示 3 个以上的面板。面板的内容通常是常规的安卓应用程序或内部构建的二维应用程序。它没有针对三维应用程序的多任务处理，并且二维应用程序中没有立体效果。由于它没有建立完整的空间用户接口系统，我们无法使用标准的三维用户接口控件构建空间应用程序。

### 3.1.2 苹果 visionOS

苹果 visionOS 是苹果 Vision Pro 的操作系统。它可以显示 3 个或更多平面窗口，状态栏和菜单栏位于侧面。它通过在一个空间中运行多个应用程序来支持三维应用程序的多任务处理，在该空间中您可以运行至少一个三维容积应用程序。面板中的二维应用程序通常是 visionOS 重新构建的 iOS 应用程序。它在对三维应用程序多任务处理的支持方面比 Meta Quest 通用菜单更好。尽管如此，对于非容积和非完全沉浸式应用程序来说，它还没有立体效果，也没有适用于所有应用程序的标准三维用户接口控件。

虽然 Reality Composer Pro 可以帮助构建三维容积和完全沉浸式应用程序，但它没有建立三维应用程序的用户接口系统，也没有用于构建三维应用程序的标准化三维控件。Reality Composer Pro 更适合构建娱乐体验而不是严肃的应用程序。

### 3.1.3 SUI

尽管当前的用户接口系统通过使用多个面板围绕用户来利用 XR 空间，但体验更像是通过单个 XR 设备的帮助来使用更多计算机显示器。如果打算以这种方式使用，那么使用真实的显示器总是更好，因为物理单元提供了最真实的体验。您永远无法通过模拟来超越设备的原始体验。

面板中的应用程序本质上是构建在二维用户接口系统上的 iOS 或安卓应用程序，因此它们无法仅通过以空间方式布局的面板提供立体和沉浸式体验。

人们在佩戴 XR 设备时期望获得身临其境的体验。VR 游戏已经证明了这一点，它呈现出最身临其境的体验，并赢得了很大的游戏机市场份额。对于应用来说，我们应该仍然保留我们在游戏中所关注的东西，那就是为用户提供身临其境的体验。

我们提出一种用于空间计算的用户接口方案，称为 SUI 或空间用户接口。在 SUI 的系统中，

我们强调立体视觉的重要性、三维用户接口控件的使用和标准化以及三维应用程序多任务处理。SUI 注重并表达三维立体、沉浸式的体验。我们打算构建一个沉浸式用户接口系统，促进 XR 在应用领域的采用，并将 XR 的使用扩展到游戏和聊天（归属于一种电话会议）应用之外。

## 3.2 特征

### 3.2.1 沉浸感

SUI 基于球坐标系<sup>[16]</sup>和多个球体的系统布局从 3 个维度充分利用了 XR 空间。作为三维空间应用程序的边界框，面板是球体上的切平面，可以是圆柱形或矩形表面。我们不是直接在面板中嵌入 iOS 或安卓应用程序，而是在每个面板中使用三维布局和三维控件来构建三维空间应用程序。由于三维布局比二维多了一个维度，并且三维控件本质上是立体的，因此将获得比二维应用程序更强的沉浸感。

### 3.2.2 生产力

通过使用多个球体，活动应用程序所在的球体将被带到前面，并在其中的应用程序不再活动时被推回。在球体中，多个三维应用程序排列在球体表面的面板中，用户可以通过拖放特定面板来自由移动它们。由于 SUI 可以容纳更多的应用程序，并且立体风格更加直观，因此用户在使用 SUI 时会感觉更加强大和高效。

### 3.2.3 自由度

当人们使用三维空间应用程序时，通常他们是坐着或站着，而无需在六自由度中移动。球面坐标系始终以用户为中心，类似于观看三自由度 VR180 或 360 度视频的体验。SUI 是建立在 3 自由度思想之上的，所以我们选择球坐标系而不是笛卡尔坐标系<sup>[17]</sup>。

## 3.3 架构

当我们使用命令行接口（CLI）时，我们在一维文本行中操作信息。当我们使用图形用户接口（GUI）时，我们操作二维位图上的信息。当我们使用空间用户接口（SUI）时，我们应该操作三维网格（Mesh）上的信息。

与游戏不同，应用程序需要在整个系统中保持一致的风格和外观，以提高开发和使用效率。我们需要一个特定的基础设施来表达三维网格的含义，并为每个特殊网格提供特定的功能。因此，我们为 SUI 设计了一个基于球坐标系的架构。

### 3.3.1 球坐标系

GUI 基于笛卡尔坐标系，而 SUI 基于球坐标系。在球坐标系在 SUI 的应用中，我们将信息组织在面板中，每个面板都是以用户为原点的球体上的切平面。面板的位置由三元组（ $\rho$ 、 $\theta$  和  $\phi$ ）确定，该三元组给出了面板在球体上的接触点的径向距离、极角和方位角。

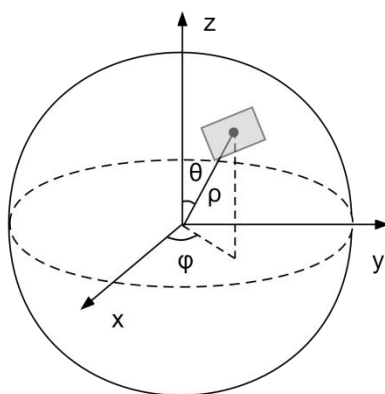


图 4 球坐标系

球坐标系有助于理解应用的空间位置，但使用笛卡尔坐标系进行计算更有用。笛卡尔坐标  $(x,y,z)$  和旋转四元数  $(q)$  可以由球坐标（径向距离  $\rho$ 、极角  $\theta$ 、方位角  $\phi$ ）得到，其中  $\rho \in [0, \infty)$ ， $\theta \in [0, 180]$ ， $\phi \in [0, 360)$ 。

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

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

$$z = \rho * \cos(\theta * 2\pi / 360)$$

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

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

$$st = \sin(\theta * 2\pi / 360 / 2)$$

$$sp = \sin(\phi * 2\pi / 360 / 2)$$

$$q = (cp * ct, -sp * st, cp * st, sp * ct)$$

### 3.3.2 多球体

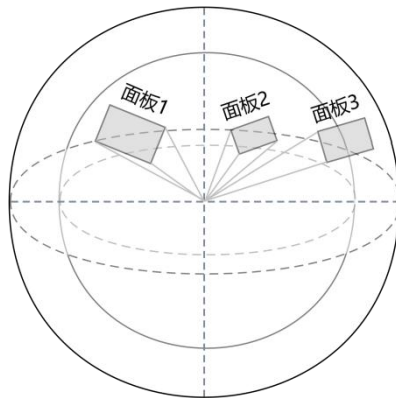


图 5 多球体

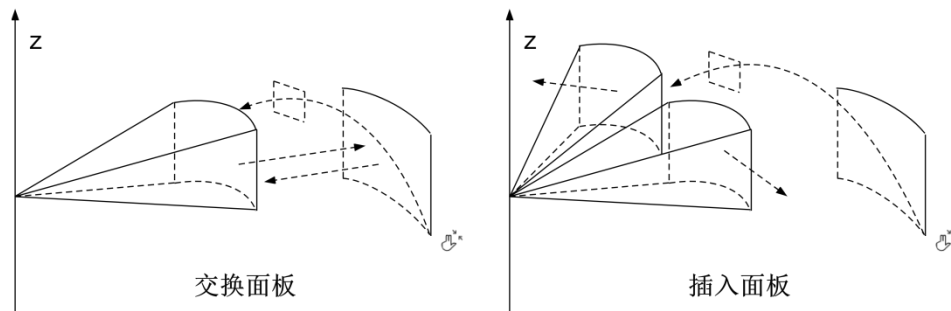


图 6 在球体之间移动面板

通过在球坐标系中为每个球体指定不同的径向半径长度，我们可以拥有多个球体。拥有多个球体的好处是我们可以充分利用空间，使空间计算比二维计算更加高效。应用程序可以放置在不同的球体上，并且可以将球体置于前面以增强观看体验。每个球体的应用程序都可以主动运行或休眠以降低功耗。理论上可以使用无数个球体，但实际上 3 个应用球体加上 1 个休眠球体和 1 个退出球体就足够了。我们可以通过同时捏住一只手的拇指、食指和中指，拖动应用程序并将其放到另一个球体中来更改应用程序所在的球体。当应用程序被放到空白区域时，它将占据该区域。当它被放到另一个应用程序上时，即使它们来自不同的球体，它们也会相互交换位置。

### 3.3.3 面板形状

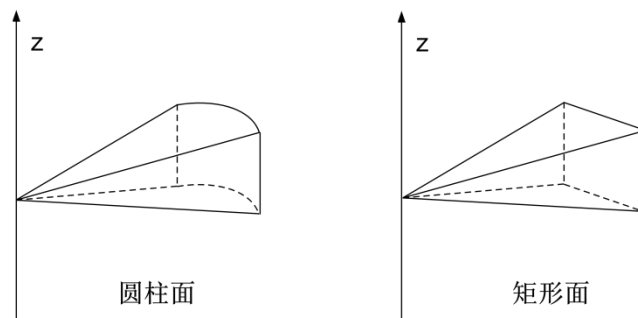


图 7 面板形状

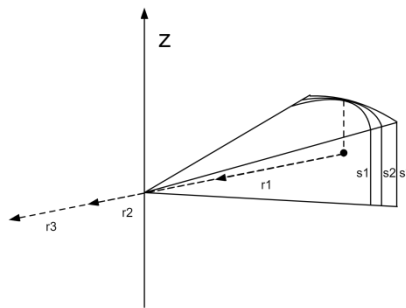


图 8 圆柱面曲率与圆柱半径  $r$  的关系

作为球体上的切面，面板可以是圆柱面或矩形面，并且两者可以互换使用。圆柱表面的曲率取决于穿过表面在球体上的接触点和球体原点的圆柱半径 ( $r$ ) 的长度。半径越长，表面越平坦。建议当面板靠近用户时逐渐变平和缩小，反之亦然。

### 3.3.4 应用布局

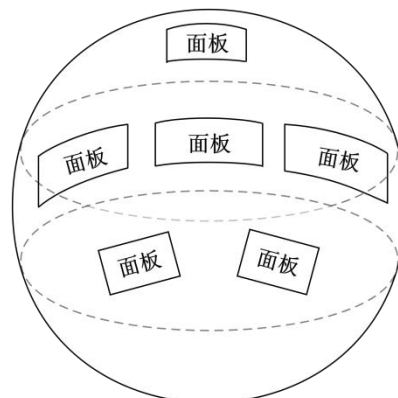


图 9 应用布局

球体的表面分为上、中、下 3 个区域。在每个区域中，并排排列着一组应用程序。最重要的应用位于中间区域以供正常使用，移至下部区域以缓解颈部疲劳，或放置在上部区域以节省更多工作空间。

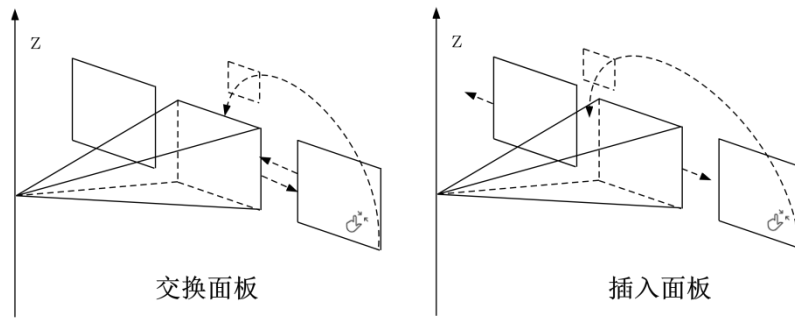


图 10 在单个球体内移动面板

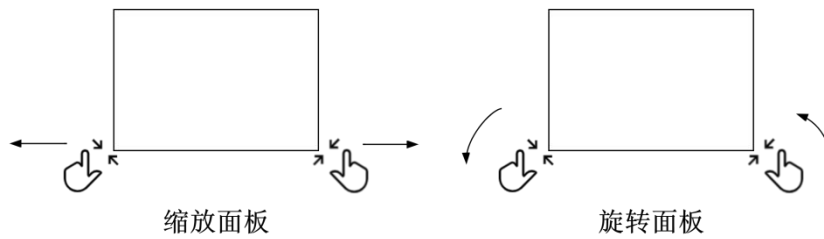


图 11 调整面板

通过向左或向右拖动该区域，每个区域的应用程序可以沿左或右方向流动。在单个球体中，可以通过捏住一只手的拇指和食指拖动应用程序并将其放下来，将应用程序放置在表面上的空白区域或者与另一个应用程序交换位置。通过捏住双手的拇指和食指并向外或向内拖动可以放大或缩小应用程序，或者通过捏住双手的拇指和食指并顺时针或逆时针旋转双手来旋转应用程序。

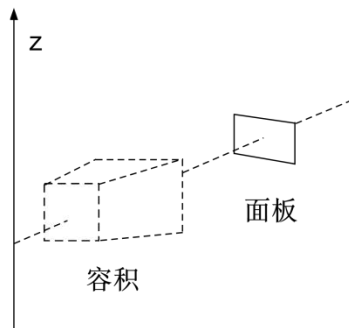


图 12 应用程序的容积状态

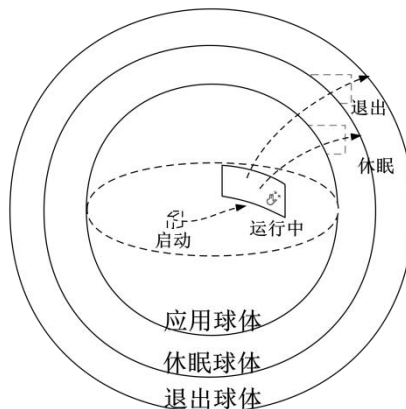




图 13 应用程序的生命周期

活动应用程序可以作为容积应用程序带到用户面前，或者完全占据整个空间以提供最身临其境的体验。应用程序可以启动、运行、休眠或退出。我们可以通过将应用程序拖放到特殊的球体上来更改应用程序的状态。当应用程序的图标被拖放到任意应用球体上时，该应用程序就会进入运行状态。当应用程序被拖放到休眠球体上时，它将进入休眠状态。当它被拖放到退出球体上时，它将从内存退出。

### 3.3.5 控件布局

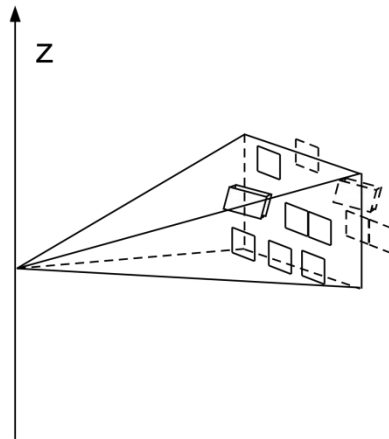


图 14 控件布局

在应用程序中，我们可以让控件像我们通常在二维布局中所做的那样流动，但还需要做一件事。由于我们比二维多了一维，因此控件可以有一个 z 坐标来定义控件和面板之间的相对距离。具有不同 z 坐标的控件将使应用程序在不同的区域具有不同的深度。如果在应用程序中控件的 z 坐标上定义动画，它会比二维应用程序中的内容看起来更立体、更生动。

### 3.3.6 三维控件

二维用户接口中的常见控件元素可以在 SUI 中具有等效的三维对应元素。控件将具有一定的厚度以呈现立体效果，例如，三维文本将看起来更加立体和清晰。此外，我们可以为其他用例继承二维元素。除了常见的控件之外，我们还可以使用模拟物理的控件来增强现实感。天图和空间图也可以用作三维控件。

### 3.3.7 控件标准化

控件的外观和功能应该得到明确的定义，这样人们在同一系统甚至异构系统中使用相同的控件时会感到熟悉、舒适和高效。

### 3.3.8 立体视觉

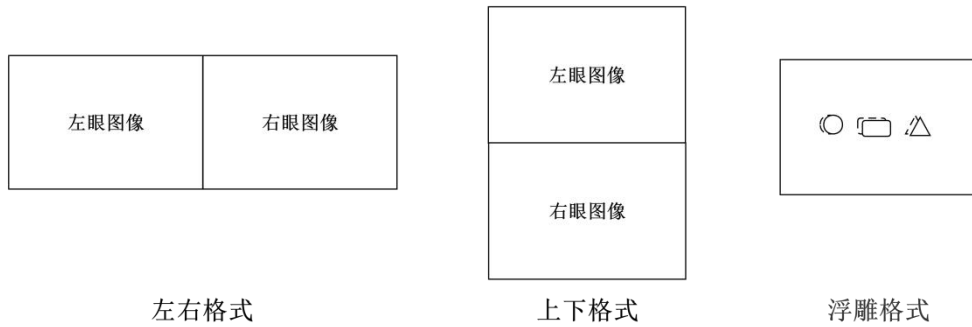


图 15 立体图像的常见格式

我们应该尽力保持控件的沉浸感。对于具有复杂数据格式的文件，其内容不应在没有显式命令的情况下直接呈现，我们可以使用立体图像作为占位符，例如，我们可以从 360 度视频中提取帧的像素并将其在应用程序中呈现为立体图像，以便用户可以查看视频文件的立体预览。

### 3.3.9 一般模型

为了扩展功能，通常在游戏和动画等交互式体验中使用的一般模型可以在空间应用程序中使用。例如，在空间电商应用上展示商品时，模型比图片更具表现力。一般模型在操作时没有明确定义的行为，而 SUI 控件具有用户预期的标准交互行为，无论它是基于三维模型的控件还是任何其他控件。

## 3.4 应用场景

### 3.4.1 三维控件库

要构建一个 SUI 系统，我们首先应该构建一个三维控件库作为工具包，然后再构建交互系统。

### 3.4.2 SUI 编辑器

SUI 编辑器可以加速空间 UI 制作和编辑。内置的三维 UI 控件可以直接被拖放到应用程序布局上。

### 3.4.3 SUI 桌面集成开发环境

SUI 桌面集成开发环境可以帮助人们使用台式计算机开发空间应用程序。

### 3.4.4 SUI 空间集成开发环境

SUI 空间集成开发环境作为原生开发工具，为空间应用开发提供了所见即所得的开发体验。

### 3.4.5 空间应用

空间应用可以应用于多种场景，提高 XR 中的工作效率。

### 3.4.6 空间操作系统（SOS）

SUI 是空间操作系统的重要组成部分，空间操作系统不仅包含 UI 系统，还包含输入和交互系统。

# 第四章

空间网——元宇宙的开放网络

## 4.1 介绍

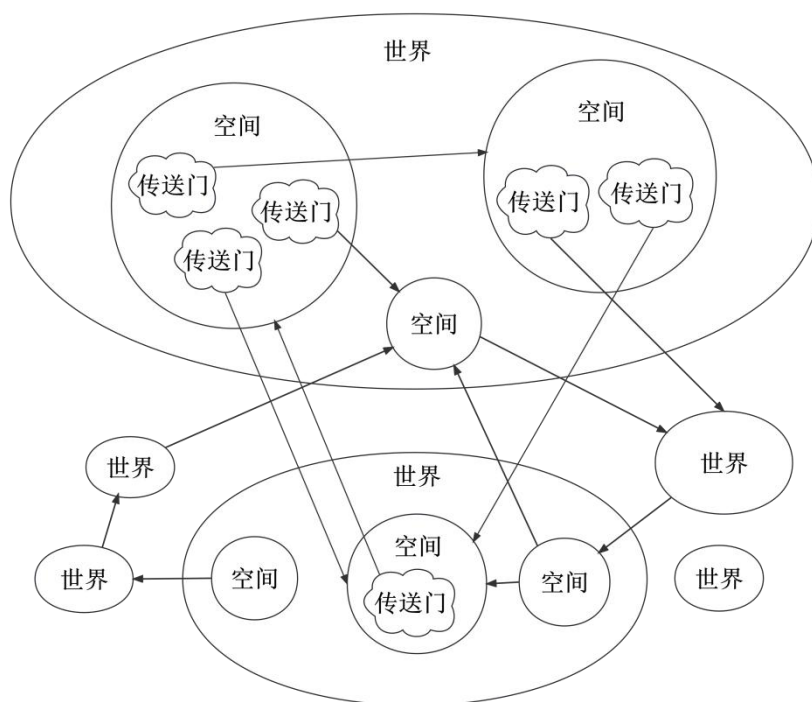


图 16 空间网架构

### 4.1.1 基于 app 的元宇宙问题

当前大多数 XR app 都实现了自己的元宇宙<sup>[18]</sup>。用户可以使用他们的化身来探索应用程序中的世界。一个臭名昭著的问题是，如果一组用户想要同时从当前的 app 转移到另一个 app 的世界进行娱乐，他们很难做到这一点。假设他们中至少有一人不拥有或尚未安装指定的 app，他们就无法完成愿望。

### 4.1.2 空间网 - 基于 web 的元宇宙

为了找到上述问题的解决方案，我们提出了一种基于 web<sup>[19]</sup>的元宇宙，称为空间网。我们使用超文本<sup>[20]</sup>衍生概念（称为 web 传送门）来表示元宇宙中两个不同世界之间的连接，因此任何一组用户都可以同时传输到任何人创建的另一个世界。

## 4.2 特征

### 4.2.1 原子性

空间网中的世界元素可以通过统一资源标识符<sup>[21]</sup>（URI）进行访问。世界中的元素松散耦合，

非常方便随意创建或修改空间布局。在一个世界中使用的模型也可以通过使用其 URI 地址在另一个世界中使用。开发人员可以仅使用其他世界的模型快速构建一个新世界。

## 4.2.2 瞬时性

当访问空间网中的世界时，访问的是 web 空间。由于访问时只是将 web 空间的内容下载到本地计算机，因此下载的资源大小比整个世界要小得多。因此，进入世界的速度要快得多。相反，当您暂时只需要访问 app 的一部分时，却必须下载整个 app 程序包。

## 4.2.3 互联性

凭借 web 传送门的概念（基本上是另一种超文本），所有的世界都通过传送门连接在一起，创建一个超大型的元宇宙。用户可以从任何一个 web 空间开始，通过直接进入传送门传送到任何世界的另一个 web 空间。

## 4.2.4 开放性

由于空间网基于万维网<sup>[22]</sup>，因此可以通过任何 web 空间的 URI 地址访问它。我们不需要 app 商店等应用中心来查找特定的 web 空间。此外，我们不必关心我们在哪个平台上，因为我们可以通过浏览器从任何地方访问世界。

## 4.3 组件

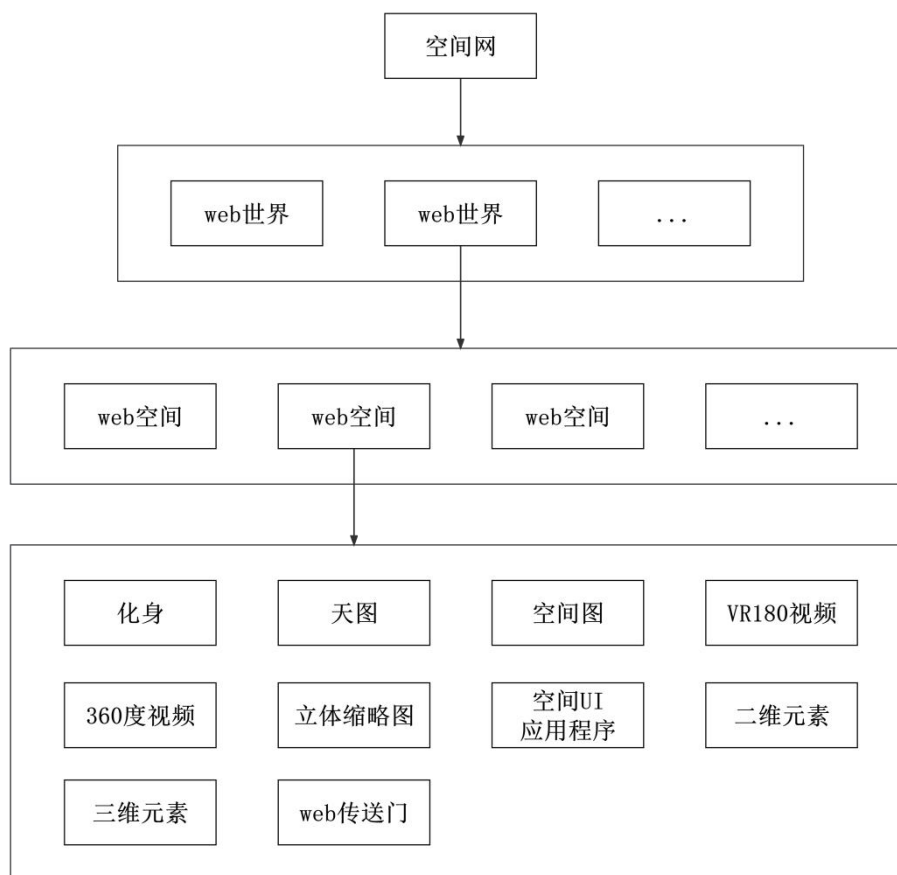


图 17 空间网组件

### 4.3.1 开放化身

当用户进入元宇宙时，它可以渲染他们的化身<sup>[23]</sup>，或者仅渲染控制器或虚拟手形，具体取决于场景。当处于一个确实需要展示化身的场景时，我们可以使用开放化身的方式从第三方提供商处获取化身。开放化身的原理类似于 Oauth<sup>[24]</sup>，它对用户进行身份验证并为发出请求的世界提供化身。因此，我们可以使用相同的化身从一个世界到达另一个世界。

### 4.3.2 web 世界

web 世界是由几个 web 空间构成的沉浸式世界。它是元宇宙中的一个空间网站。

### 4.3.3 web 空间

web 空间是空间网的基本单位，其对于空间网的作用与普通网页对于万维网的作用相同。一个世界由空间网中的多个 web 空间组成，因此当您访问基于 web 的元宇宙中的世界时，您

实际上正在访问 web 空间。

### 4.3.4 web 传送门

web 传送门实际上是空间网中的超链接<sup>[25]</sup>，但它呈现为三维的门或传送门，而不仅仅是万维网中高亮显示的文本字符串。空间网中的任何对象都可以成为 web 传送门，因此不仅可以通过常规 web 传送门，还可以通过任何传送门化和高亮显示的对象传输到另一个 web 空间。

### 4.3.5 扩展元素

通过用更多的元素扩展 web，我们可以在元宇宙中完成除了游戏和聊天之外的更多种类的工作。

#### 4.3.5.1 天图

作为人类，我们创造了字符和文字，因为我们无法保存自己的全息信息。因此，我们需要一种方法来高效、紧凑地将我们所说和所做的内容保存在周围任何可雕刻表面（包括石头、纸张和二维电子屏幕）中。

也许是因为我们在几千年的发展中已经习惯了这种交流方式，所以我们在二维屏幕上阅读作为最基本的信息单位的文字感觉很自然，但在元宇宙中的二维面板上阅读它们却感觉有点违和。由于元宇宙比二维屏幕更加沉浸，因此我们在元宇宙中查看全息信息比阅读文本感觉更舒服。就像文本一样，天图可以用来保存和异步传输人类所说和所做的事情，但以更全息的方式。所以我们大胆提出使用天图代替文本作为元宇宙中的基本交流单位。

我们应该用天图来填充元宇宙中最自然、最全息的内容。没有它，我们只能找到几个空白而荒凉的世界，漫无目的地徘徊。这也许就是为什么大多数早期经历过元宇宙的人不理解它，甚至不再在那里停留的原因。有了它，我们将拥有像二维屏幕上一样的异步现实，我们可以绕过时间障碍相互交流，并感觉被更多的内容包围。

#### 4.3.5.2 空间图

就像天图可以帮助解决元宇宙中全息信息呈现和交流的最基本问题一样，空间图可以让人们在以全息风格和外观体验电影、电视剧、动画等轻娱乐时感到更加舒适。

#### 4.3.5.3 VR180 和 360 度视频

VR180 和 360 度视频虽然只能为视频提供 3 自由度体验，但在元宇宙中它们比二维视频更具沉浸感。



#### 4.3.5.4 立体缩略图

为了在元宇宙中以全息方式识别和组织元素，我们应该使用立体缩略图而不是二维图像来表示元素。

#### 4.3.5.5 空间 UI 应用程序

正如我们上面提到的，我们应该能够在元宇宙中完成更多的任务。除了信息交流、互动游戏、轻娱乐之外，我们还可以完成办公等工作。还有许多其他类型的应用程序有可能帮助我们更多地利用元宇宙，例如全息新闻或 6 自由度视频共享应用程序。

#### 4.3.5.6 同步现实的问题

在当前的同步现实中，开发人员追求应用程序中对物理世界的 1:1 模拟。相反，在传统的二维应用中，开发人员更倾向于利用计算机的力量来解决人们在现实世界中无法轻松解决的问题。空间计算相对于 PC 和移动计算的优点在于能够呈现全息信息并使人类沉浸其中，而不是更多的计算能力。我们应该继承我们通过 PC 和移动计算的计算能力所做的事情，并将其与渲染全息信息的能力结合起来。如果我们只是对物理世界进行 100% 的模拟，那么我们使用计算机就没有任何意义，因为我们已经拥有了一个完美的现实世界。

聊天作为当前元宇宙中最流行的同步体验类型，有几个突出的问题。人们对它的某些方面存在误解。当然，它在信息展示上与传统的聊天应用程序不一样，因为它可以显示沉浸式环境和化身，但实际上在工作流程上也不尽相同。因为它力求完美模拟物理世界，所以它更像是电话会议，而不是我们在二维聊天应用程序中实现的样子。在很多方面，它甚至与二维聊天应用程序相比都没有相对竞争力。从某种意义上说，二维聊天根本不是同步的，因此它在各个方面都很好地利用了计算机，并且变得更加有用，尽管它通常看起来是同步的。

目前的聊天应用程序通常会显示同一房间中所有参与者的化身，人们在群组或一对一的情况下面对面交谈，这更像是将许多私人聊天放在一个房间中，而不是公共聊天。在公共聊天中，人们期望收到房间中每个人的消息，而不打算实时了解其他人正在做什么。

一个问题是，在传统的二维聊天室中，来自房间中的一个人的消息将广播给同一房间中的每个人，这是非常高效和有用的，因为每个加入房间的人都希望从来自同一个房间的其他人那里收到更多信息。这就是聊天室的目的。但在当前的虚拟世界中，一个人所说或所做的事情只能传播到说话者周围适当范围内的少数人，这比二维聊天室效率更低。

另一个问题是房间里的一个人是否真的更喜欢看到同一个房间里所有其他人的化身，甚至是那些相隔几十米的人的化身，以及是否真的有必要渲染甚至被房间墙壁遮挡的化身。在二维聊天室中，用户可以看到房间里的人数，从房间的用户列表中找到另一个特定的用户。由于用户不必实时查看所有其他用户，因此不需要一次渲染所有个人资料头像。该方法如果用在元宇宙中，将会显著降低功耗。

只渲染当前用户特定范围内的其他用户的化身，并将说话人的化身远程传送给房间中的所有参与者怎么样？用户可以选择加入聊天，就像他现在在聊天应用程序中所做的那样，或者向聊天室中的所有用户广播，甚至录制天图并将其发送给所有人。这样做的好处是，天图消息可以保留在聊天记录中，方便以后查找它们以进行更合理的对话。

有人可能会说，我们无法通过这种方式看到房间里到底有谁。答案是我们可以用任何更有效的方式来表示它们，例如，使用上面有名字的球体、用户个人资料头像的立体缩略图，甚至是低多边形版本的化身来表示任何超出了特定范围但还不算太远的用户。我们不必为距离很远的用户呈现代表性元素，而我们可以以三维样式显示所有用户的列表，就像我们在二维聊天室中所做的那样，以帮助查找和定位用户。当当前用户逐渐靠近另一用户，且两者之间的距离小于特定长度时，该用户的化身会逐渐出现，而以当前用户为中心的有效范围之外的用户化身会逐渐消失。

这样我们就可以显著减少不必要的计算资源使用，降低功耗，同时也可以获得与传统二维聊天室一样的性能。

#### 4.3.5.7 同步现实还是异步现实？

目前，大多数元宇宙应用程序默认使用同步现实，即显示世界中所有人的化身，甚至不考虑是否适合他们。这是一种常见的情况，而且非常尴尬，你进入元宇宙应用程序除了找人聊天之外无所事事。虽然我们可以渲染所有人在元宇宙中的化身，但是我们有必要这样做吗？用户进入一个世界时是否总是希望看到其他人？也许是，但也许不是。当我们打算与他人聊天时，我们会进入专门为此目的而构建的聊天室，但当我们只想消费某些内容或异步交流时，我们并不期望实时看到其他人的在线化身。我们最好好好思考一下我们在元宇宙中的世界的目的，并仔细考虑是否同步显示所有化身。有时候不展示别人的化身反而会带来更多的益处。

#### 4.3.5.8 传统二维元素

空间网是建立在空间计算基础上的 web，因此它应该能够包含和渲染二维元素，例如文本、音频、图像、视频，甚至二维 html。

#### 4.3.5.9 一般三维元素

三维元素包括原始形状和三维格式的文件。原始形状列表包括球体、长方体、圆柱体和其他基本形状。三维格式列表包括 glTF、VRM、USD、FBX 和 OBJ 等。

## 4.4 相关框架

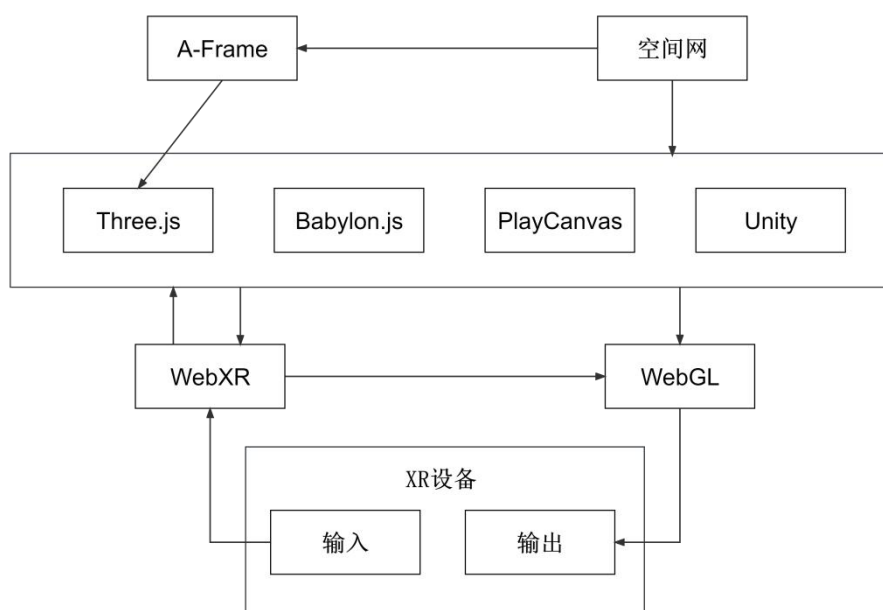


图 18 空间网与相关框架的关系

### 4.4.1 WebXR

WebXR 是一个 web API，它接收来自 XR 的输入并通过 WebGL 的帮助将输出发送到 XR，而空间网是使用万维网的基本概念在空间计算中实现通用 web 的一种理念和方法。WebXR 是空间网获取输入和发送输出的必要基础。

### 4.4.2 WebGL

WebGL 是用于渲染 3D 图形的 JavaScript API。WebXR 使用它在 XR 显示器上呈现屏幕图像。一些网络图形库使用它来渲染场景和模型。这是空间网所必需的。

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

这些库或工具是高级图形 API 框架或游戏引擎，可以帮助您更轻松地开发基于 WebGL 的应用程序或游戏。它们主要用于制作游戏等互动体验。它们可以采用空间网的概念来扩展它们的应用场景。

### 4.4.4 A-Frame

A-Frame 是一个基 Three.js 和 WebXR 的实体组件库。开发者可以用它来构建游戏等 XR 体验。它可以采用空间网的概念来扩展其应用场景。

## 4.5 应用场景

### 4.5.1 空间网浏览器

空间网浏览器默认呈现网络空间的沉浸式内容。它可以显示二维 `html` 的内容，但只能在空间位置中显示它们。二维 `html` 可以是 `web` 空间的一部分，甚至可以转换为三维版本，以将其显示为类似于 `SUI` 应用程序的内容。

### 4.5.2 空间网搜索引擎

典型的二维网络搜索引擎主要以二维格式搜索和呈现信息，而空间网搜索引擎则搜索沉浸式内容（例如天图）并以沉浸式形式输出。

### 4.5.3 Web service

在基于 `app` 的元宇宙中，大多数资产都捆绑在构建的程序包中。为了提高效率并减小应用程序的大小，可以在 `web service` 中提供资源，`app` 可以根据需要获取和呈现它们。

### 4.5.4 开放化身提供商

`Ready Player Me` 等第三方化身提供商是一个选择。

# 第五章

结论

计算机提供同步现实（远程聚会）和异步现实（记忆扩展）。在命令行和图形用户界面上，人们创建了许多同步和异步程序来方便日常生活并使其更加方便。随着空间计算的日益突出，我们已经擅长推出同步体验，但迫切需要异步体验。作为空间计算异步问题的一种可能解决方案，天图很可能为其开辟新的路径。

人们在计算机上使用的媒体已经经历了多次演变，从文本、音频、图像到视频，并且出现了许多使用这些格式作为内容载体的应用程序。和以前一样，像空间图这样的 6 自由度视频格式将继续这种范式，并有望成为计算机上新型应用程序的下一个媒介。

SUI 是一种全新的用户接口系统设计方式，旨在最大限度地利用 XR 的大空间，高效地操作空间中的用户接口组件。我们可以同时获得沉浸感和生产力。

当前的元宇宙是一堆彼此之间没有联系的应用程序和游戏。我们将元宇宙视为一个由松散耦合的信息空间组成的沉浸式世界，这些信息空间通过人们可以随意通过的传送门连接起来。我们借鉴万维网并构想了一个基于 web 的解决方案，称为空间网，用于构建开放且互联的元宇宙。此外，我们还深入思考了当前元宇宙中的同步现实等问题，并提出了专门的解决方案。

## 参考文献

- [1]. 万尼瓦尔·布什, 《As We May Think》, 1945 年。
- [2]. 西蒙·格林沃尔德, “空间计算”, 麻省理工学院研究生论文, 2003 年 6 月, <https://acg.media.mit.edu/people/simong/thesis/SpatialComputing.pdf>
- [3]. 米卡·布隆伯格, “Epic Games 的尼克·怀廷解释了 XR 的起源(XR =虚拟现实+增强现实)”, <https://www.svgn.io/p/how-the-term-xr-was-coined-its-so-funny-hint-it-does-not-mean-extended-reality-4e7fff539676>
- [4]. 安德烈亚斯·雷内·芬德和克里斯蒂安·霍尔兹, “保留因果关系的异步现实”, 载于 2022 年美国计算机协会计算系统人机交互会议论文集, 2022 年 4 月, 第 634 号文章, 第 1-15 页, <https://dl.acm.org/doi/pdf/10.1145/3491102.3501836>
- [5]. Khronos® 3D 格式工作组, “glTF™ 2.0 规范”, <https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.pdf>
- [6]. VRM 联盟, “VRM - VR 3D 化身文件格式”, <https://vrm.dev/en/>
- [7]. 谷歌开源, “VR180 视频格式”, 2018 年 4 月 10 日, <https://github.com/google/spatial-media/blob/master/docs/vr180.md>
- [8]. 弗兰克·尼尔森, “环绕视频: 多头摄像头方法”, 《视觉计算机》, 第 21 卷, 第 92–103 页, 2005 年 2 月 1 日。
- [9]. 谷歌开发者, “自由度”, 2018 年 9 月 21 日, <https://developers.google.com/vr/discover/degrees-of-freedom>
- [10]. OpenUSD 联盟, “OpenUSD(通用场景描述)”, <https://www.openusd.org/release/index.html>
- [11]. 阿尔布雷希特·梅登鲍尔, 《摄影测量》, 1867 年, <https://opus4.kobv.de/opus4-btu/files/749/db186714.pdf>
- [12]. 加文·克里斯托弗·纽瑟姆和雪莉·纳什·韦伯, “行政命令 N-12-23”, 加利福尼亚州行政部门, 2023 年 9 月 6 日, [https://www.gov.ca.gov/wp-content/uploads/2023/09/AI-EO-No.12-\\_GGN-Signed.pdf](https://www.gov.ca.gov/wp-content/uploads/2023/09/AI-EO-No.12-_GGN-Signed.pdf)
- [13]. 模拟器件公司, “飞行时间 什么是飞行时间?”, <https://www.analog.com/en/design-center/glossary/time-of-flight.html>
- [14]. 苹果公司, “探索 VisionOS”, 2023 年全球开发者大会, 2023 年 6 月, <https://developer.apple.com/visionos/>
- [15]. Meta 公司, “在 Meta Quest 上使用通用菜单”, 2023 年 10 月, <https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-2/universal-menu-for-quest-2-and-quest/>
- [16]. 国际标准化组织, “ISO 80000-2:2019 Quantities and units – Part 2: Mathematics”, 第 20–21 页, 项目编号 2-17.3。
- [17]. Nykamp DQ, “笛卡尔坐标”, 数学洞察, [http://mathinsight.org/cartesian\\_coordinates](http://mathinsight.org/cartesian_coordinates)
- [18]. 尼尔·汤·史蒂芬森, “雪崩”, 矮脚鸡出版社, 1992 年 6 月。
- [19]. 蒂莫西·约翰·伯纳斯·李爵士, “信息管理: 一项建议”, 1989 年 3 月。 <https://cds.cern.ch/record/369245/files/dd-89-001.pdf>
- [20]. 西奥多·霍尔姆·纳尔逊, “复杂信息处理: 复杂、变化和不确定性的文件结构”, ACM '65: 1965 年第 20 届全国会议记录, 1965 年 8 月。 <https://dl.acm.org/doi/pdf/10.1145/800197.806036>

- [21]. 互联网工程任务组, “统一资源标识符 (URI): 通用语法”, 征求意见稿文件 2396, 1998 年 8 月。 <https://datatracker.ietf.org/doc/html/rfc2396>
- [22]. 蒂姆·伯纳斯·李和罗伯特·卡尤, “万维网: 超文本项目提案”, 1990 年 11 月 12 日。  
[http://cds.cern.ch/record/2639699/files/Proposal\\_Nov-1990.pdf](http://cds.cern.ch/record/2639699/files/Proposal_Nov-1990.pdf)
- [23]. 尼尔·汤·史蒂芬森, “雪崩”, 矮脚鸡出版社, 1992 年 6 月。
- [24]. IETF, “OAuth 1.0 协议”, 征求意见稿文件 5849, 2010 年 4 月。  
<https://datatracker.ietf.org/doc/html/rfc5849>
- [25]. 西奥多·霍尔姆·尼尔森, 《Project Xanadu》项目, 1965 年。