

天津大学

本科生毕业论文



学 院____软件学院____

专 业____软件工程____

年 级____2014 级____

姓 名____韦又少____

指导教师____翁仲铭____

2018 年 5 月 28 日

天津大学

本科生毕业论文任务书

题目：WebVR 让 VR 无限传播的技术与方法

学生姓名_____韦又少_____

学院名称_____软件学院_____

专 业_____软件工程_____

学 号_____3014218102_____

指导教师_____翁仲铭_____

职 称_____副教授_____

一、原始依据

由于 Web 不需要下载程序的简便性使得更多的人倾向于在 Web 页面开发 VR/AR 应用（快速捕捉客户）。目前谷歌 Chrome 浏览器已经支持查看 WebVR 内容，在今天的 I/O 开发者大会上，谷歌正式提出 WebAR 概念，谷歌 Chrome 将会推出 WebAR 功能。

所谓 WebVR，就是 VR on Web。它把虚拟现实这项酷炫的技术带到 Web 领域来，用 JavaScript 来写虚拟现实相关的应用，从而在浏览器或者 Web runtime 上跨平台运行。而 WebAR 将会是一个为我们提供与网页内容进行实时交互的平台。从 AR 设备到 WebAR 技术的背后，是一个市场化普及的过程，它的适应性更强，开放程度更高，更符合当前面向更多的消费者这样的体验习惯。既然是 VR on Web，首先需要在浏览器里支持。所以先看看浏览器的支持情况：

Firefox and Chromium: 可在 Windows 上支持 HTC Vive 和 Oculus

Samsung Internet on Android / Oculus Carmel: 可在 Android 上支持 Samsung GearVR

Chrome for Android: 可在 Android 上支持 Google Daydream and Google Cardboard

Edge: 可在 Windows 上支持 Microsoft HoloLens

开发 WebVR 程序不是一件简单的事情，所以好的 framework 或者 tool 有事半功倍的效果，无论是做个简单的 demo 还是真正的产品。支持 WebVR 的框架和工具有：

aframe, three.js, React VR, Vizor, PlayCanvas for VR, WebVR emulator, 目前本课题比较常用 aframe 和 three.js。

目前和 WebVR 相关的 specification 有：

WebVR in W3C, 最基本的。

WebGL in Khronos, 因为 WebVR 是基于 WebGL 来绘制 3D 场景，所以对 WebGL 的了解也很重要。

Holographics, 微软看起来并没有积极参与 W3C 和 Khronos 社区，有自己的 VR 标准。

和标准相关的其它工作，则是致力于上文提到的 framework/tool，甚至是开发新的好用的 framework/tool，来推动 WebVR 的发展。另外也很重要，在 Unity, Unreal, Cocos 等游戏引擎或集成开发环境里添加最新的 WebVR/VR 的支持，从而推动 WebVR/VR 的应用和发展。

所以本课题主要是以 VR 的技术为实现的主体，主要研究与探讨 WebVR 技术，并从 WebVR 进一步研究 WebAR 技术和设计技巧，并且建立一个 WebAR 实用的说明范例。本课题的目的就是对 VR 和 AR 技术在 Web 上传播的进一步的探讨学习，跟

上时代的发展。

二、参考文献

- [1] Zhang M, Lv Z, Zhang X, et al. Research and Application of the 3D Virtual Community Based on WEBVR and RIA[J]. 2009, 2(1).
- [2] Barsoum E, Kuester F. WebVR: an interactive web browser for virtual environments[C]// Electronic Imaging. International Society for Optics and Photonics, 2005:540-547.
- [3] Mendigochea P. WebAR: creating augmented reality experiences on smart glasses and mobile device browsers[C]// ACM SIGGRAPH. ACM, 2017:10.
- [4] Danchilla B. Three.js Framework[M]// Beginning WebGL for HTML5. Apress, 2012.
- [5] T. Luhmann, S. Robson, S. Kyle, J. Boehm, Close-range Photogrammetry and 3D imaging, DE Gruyter, 2013.
- [6] Rodrigues A B, Dias D R C, Martins V F, et al. WebAR: A Web-Augmented Reality-Based Authoring Tool with Experience API Support for Educational Applications[J]. 2017:118-128.
- [7] L. Freina, M. Ott, "A Literature Review on Immersive Virtual Reality in Education: State of the Art and Perspectives", e-learning and Software for Education Bucharest, pp. 131-141, 2015.
- [8] V. Vukicevic, B. Jones, K. Gilbert, C. Van Wiemeersch, (n.d.). W3C Github. Retrieved January 04 2017 from WebVR Editor's Draft, December 2016, [online] Available: <https://w3c.github.io/webvr/>.
- [9] D. Gonzalez-Zuniga, A. Chistyakov, J. Carrabina, "Redefining a Pattern: Stereoscopic Web Perception", IEEE Latin America Transactions, pp. 514-519, 2014.
- [10] G. Saucedo-Tejada, S. Mendoza, D. Decouchant, "F2FMI: A toolkit for facilitating face-to-face mobile interaction" in Expert Systems with Applications, Elsevier, vol. 40, no. 15, pp. 6173-6184, 2013.

三、设计（研究）内容和要求

本课题是一个在 Web 环境下对 webVR/AR 技术的开发与研究。主要设计内容是利用 aframe 框架和 ar.js 库在 webstrom 中进行 Web 的设计和 3D 模型的建立，并延续到 AR 中，设计一个实用实例。

学生的要求是在现有的技术基础上，进一步的学习 three.js 以及 AR 相关知识，要具有 Web 开发的基本编程基础，怎样能更好的呈现 VR/AR 效果，沉浸感更好等。

在此项毕业设计中，能够提高学生的理论知识学习能力，充分锻炼和培养学生的创造想象力，能够有助于了解目前最新的 WebVR/AR 技术，使我们的生活更加的智能化。

指导教师（签字）

年 月 日

审题小组组长（签字）

年 月 日

天津大学本科生毕业论文开题报告

课题名称	WebVR 让 VR 无限传播的技术与方法		
学院名称	软件学院	专业名称	软件工程
学生姓名	韦又少	指导教师	翁仲铭

一、课题的来源及意义

随着 VR/AR 的发展，越来越多的人接触到虚拟现实技术并应用到各个领域，因此我们也在思考将 VR、AR 技术应用到 Web 上。由于 Web 不需要下载程序的简便性使得更多的人倾向于在 Web 页面开发 VR/AR 应用（快速捕捉客户）。

所谓 WebVR，就是 VR on Web。它把虚拟现实这项酷炫的技术带到 Web 领域来，用 JavaScript 来写虚拟现实相关的应用，从而在浏览器或者 Web runtime 上跨平台运行。而 WebAR 将会是一个为我们提供与网页内容进行实时交互的平台。

目前谷歌 Chrome 浏览器已经支持查看 WebVR 内容，在今年的 I/O 开发者大会上，谷歌正式提出 WebAR 概念，谷歌 Chrome 将会推出 WebAR 功能。今年 6 月 1 日，Mozilla 官方博客发布预告，宣布自今年 8 月 8 日开始，Windows 平台的 Firefox 55 版本将默认启用 WebVR 接口，这也就意味着开发者能够使用 HTC Vive、Oculus Rift 等设备对 webVR/AR 进行开发。

Mozilla 表示：“WebVR 将虚拟现实技术整合到上网体验中，在开放和交互的网络平台上为用户创造无限的可能。” Mozilla 表示：“该技术配合 WebGL 来渲染 3D 图形，API 能够将浏览器打造成为允许 VR 内容在网络上传播的大平台，而且消费者可以通过 VR 设备进行观看体验。”

二、国内外发展状况

尽管成百上千万新用户会在 2017 年试用和购买虚拟现实体验，但是，仍有大量的用户不使用这些设备。虚拟现实技术之所以能够有如此大的吸引力，主要就是因为它能够让用户体验到 3D 立体效果。随着虚拟现实技术的发展，开始涌现更多的 3D 立体效果，同时让开发者能够制作和研究开发，因此用户当然也想和周围的人一起体验 3D 立体效果和沉浸感。因此，Mozilla 的 A-Frame 框架将成为一个很有用的基础架构，从而让任何开发者都能够直接通过网页浏览器提供 3D 图像来进行 webVR/AR 的开发。Web 技术不仅使创作 VR 的成本更加低廉，而且大大降低技术门槛。众所周知，虽然许多 VR/AR 技术可以应用到多种平台，但是比如 VR 游戏，VR 玩家和非 VR 玩家是基本不能实现共享游戏的，这就是 VR 玩家稀少比较局限的主要原因，所以 webVR/AR 的一个很关键的优势就是，具有能够跨越移动端、网站和 VR/AR 平台来达到众多用户体验的能力，这就使得 webVR/AR 可以让 VR/AR 中的用户与非 VR/AR 用户进行连接，使 webVR/AR 应用能够大范围的普及，进而影响更多的用户，得到更好的用户体验。同时，webVR/AR 技术可以让内容不再受到平台、硬件的约束，再加上现阶段很多的浏览器，例如谷歌 chrome、火狐 firefox 等这样的浏览器已经积累的大量的潜在用户，而且这也是

为 webVR/AR 浏览器市场奠定了坚实的用户基础。此外 webVR/AR 技术大大降低了开发者进入 VR/AR 领域的门槛，促使大量内容的涌进。

Google 通过发布 ARCore 和 ARKit 的实验性 WebAR 浏览器，向开放网页展示了他们的奉献精神，以便网络开发人员可以开发交叉兼容的 AR 应用程序。谷歌预见了一个未来发展到超越现有的独立应用程序模式，但这需要一些技术创新，包括由 AI 所提供的环境感知环境计算以及 Google I / O 宣布的虚拟定位系统。还有一些其他的生产力应用程序，谷歌正在继续尝试，但屏幕分辨率仍然需要改善从视力测量 20/100 到更接近 20/40 的视力。

从国外来看，实际上，苹果 safari 是目前唯一一个还不能支持 webVR/AR 的浏览器，这意味着为了能够让 webVR/AR 的内容可以在 safari 上快速反应，开发者必须要自己添加能够与 webVR/AR 类似的接口来提供信息，因此就有了 polyfill 规范，虽然 polyfill 可能仍然不能适用于一些浏览器（例如安卓版的 Edge 等），但是大多数情况下还是能给出用户不错的体验。当 webVR 技术在浏览器中呈现的时候，就可以展示 3D 立体内容，同时手机端就能接受到信息并且做出反应。

从国内来看，很多的信息技术公司都在开始研究 webVR/AR 技术，并把这两个技术不断的应用于不同的社交 app 中，比如 wechat、新浪微博等社交应用，同时也会在 webVR/AR 中陆续加入更好的识别等功能。而我们也在致力于向微信公众号中实现 webVR/AR，目前由于微信浏览器还不能支持 webVR/AR，因此只能是点击网址跳转至特定的浏览器，而该技术由于微信公众号的接口权限问题目前还无法做到，相信不久的将来，webVR/AR 一定会在微信公众号以及国内市场展现它的魅力。

三、研究目标

讨论 WebVR 和 WebAR 的设计，并且做一个 WebAR 的实验范例

四、研究内容

本课题是一个在 Web 环境下开发 VR/AR 的研究。主要设计内容是利用 aframe 框架和 ar.js 库在 webstrom 中进行 Web 的设计和 3D 模型的建立，并延续到 AR 中，设计一个实用实例。

五、研究方法

根据现有的技术基础，使用 WebGL 与 WebVR API 结合，在常规 Web 端三维应用的基础上通过 API 与 VR 设备进行交互，进而得到对应的 VR 实现。A-Frame 是一个通过 HTML 创建 VR 体验的开源 WebVR 框架。aframe 框架和 three.js 库在 webstrom 中进行 Web 的设计和 3D 模型的建立，并延续到 AR 中。

六、研究手段

1. 查阅相关资料和研究成果，了解目前简单的 WebVR 和 WebAR 并进行学习。
2. 研究 WebVR 和 WebAR 实现的原理，并准备相关器材，搭建环境。
3. 寻找最好的浏览器能完美的呈现 3D 效果。

4. 学习 A-Frame 框架和 ar.js, 说明如何在这两个库上制作 WebVR 内容。
5. 渲染中主要渲染程序的三打要素: 建立好 scene、renderer、camera。
6. 通过 WebVR 技术延伸到 WebAR 技术, 制作一个简单的实例。

七、进度安排

2017 年 12 月 11 日—2017 年 12 月 24 日 查阅资料完成任务书, 完成开题报告。

2017 年 12 月 25 日—2018 年 3 月 4 日研究 WebVR 和 WebAR 实现的原理, 并准备相关器材, 搭建环境。

2018 年 3 月 5 日—2018 年 3 月 11 日 寻找最好的浏览器能完美的呈现 3D 效果。

2018 年 3 月 12 日—2018 年 4 月 23 日 学习 A-Frame 框架和 ar.js, 说明如何在这两个库上制作 WebVR 内容。

2018 年 4 月 24 日—2018 年 5 月 6 日 毕业设计中期检查。

2018 年 5 月 7 日—2018 年 5 月 27 日 通过 WebVR 技术延伸到 WebAR 技术, 制作一个简单的实例。

2018 年 5 月 28 日—2018 年 6 月 10 日 检查核对论文。

2018 年 6 月 11 日—2018 年 6 月 19 日 毕业设计答辩。

八、可行性分析

1. 技术可行性

本课题基于 WebVR 技术, 并且已有国内外成功的案例作为参考。通过该技术可以实习 WebVR 和 WebAR 的传播。

2. 经济可行性

由于本项目无需投入额外的设备购买及人员培训费用, 主要是技术的设计技巧, 所以在经济上本课题的开发是可行的。使用个人电脑就可制作出实例, 而相关的眼镜头盔则又实验室提供, 可行性更高。

3. 操作可行性

开发人员已经具备基本的 Web 知识。另外, 对该项目有充裕的时间保证, 具备了操作可行性。

九、已具备的实验条件

硬件条件: 个人电脑, 三星的 gear 眼镜, 手机摄像头, HoloLens

软件条件: webstrom, 3D 扫描仪等

十、参考文献

- [1] [1] Zhang M, Lv Z, Zhang X, et al. Research and Application of the 3D Virtual Community Based on WEBVR and RIA[J]. 2009, 2(1).

- [2] Barsoum E, Kuester F. WebVR: an interactive web browser for virtual environments[C]// Electronic Imaging. International Society for Optics and Photonics, 2005:540-547.
- [3] Mendigochea P. WebAR: creating augmented reality experiences on smart glasses and mobile device browsers[C]// ACM SIGGRAPH. ACM, 2017:10.
- [4] Danchilla B. Three.js Framework[M]// Beginning WebGL for HTML5. Apress, 2012.
- [5] Rodrigues A B, Dias D R C, Martins V F, et al. WebAR: A Web-Augmented Reality-Based Authoring Tool with Experience API Support for Educational Applications[J]. 2017:118-128.
- [6] Rodrigues A B, Dias D R C, Martins V F, et al. WebAR: A Web-Augmented Reality-Based Authoring Tool with Experience API Support for Educational Applications[M]// Universal Access in Human - Computer Interaction. Designing Novel Interactions. 2017.
- [7] 田彦, 张猛. 基于 WEBVR 和 RIA 技术的三维虚拟社区研究与应用[J]. 电脑知识与技术, 2011, 07(7X):4988-4990.
- [8] 俞璋凌. 基于 web&VR 技术的产品展示方法研究[D]. 合肥工业大学, 2009.
- [9] 鲁徐, 朱齐勇, 薛飞. WebVR 在网络教学中的应用[J]. 中国商界, 2010(11X):394-394.
- [10] 谭云兰, 贾金原, 康永平, 等. 基于 WebVR 的井冈山虚拟旅游系统架构设计[J]. 井冈山大学学报(自然科学版), 2012(6):46-50.

选题是否合适: 是 ☐ 否 ☐

课题能否实现: 能 ☐ 不能 ☐

指导教师 (签字)

年 月 日

选题是否合适: 是 ☐ 否 ☐

课题能否实现: 能 ☐ 不能 ☐

审题小组组长 (签字)

年 月 日

摘 要

webVR/AR 是可以应用在广泛领域里的一项重要技术，本课题主要探讨的是 webAR 技术，通过 A-Frame 框架、ar.js、artoolkit 等技术构建一个 webAR 的实例。文章将介绍制作技术的基本原理，建模过程，需要用到的设备以及软件，详细设计，以及案例应用的领域，最后结合实际案例说明优点和局限性，并对 webVR/AR 研究进行了展望。

关键词：webAR；建模；A-Frame 框架；ar.js

ABSTRACT

webVR/AR is an important technology that can be applied in a wide range of fields. This topic focuses on the webAR technology and builds an example of a webAR through technologies such as the A-Frame framework, ar.js, and artoolkit. The article will introduce the basic principles of production technology, the modeling process, the equipment and software that need to be used, the detailed design, and the field of case application. Finally, the advantages and limitations of the actual case will be described, and the future of webVR/AR research will be presented.

Key words: webAR; Modeling; A-Frame frame; ar.js

目 录

第一章	绪论.....	1
1.1	课题背景与意义	1
1.2	国内外研究现状	1
1.3	研究目标与工作内容.....	2
1.4	本章小结	2
第二章	技术支持原理.....	3
2.1	A-Frame 框架.....	3
2.2	ar.js	4
2.3	three.js-ARToolKit.....	4
2.4	webGL	4
2.5	WebRTC	5
2.6	three.js	5
第三章	模型建立.....	7
3.1	建模工具	7
3.2	建模过程	7
3.3	Artec 扫描建模注意事项	12

第四章	详细设计.....	14
4.1	设计工具	14
4.2	设计思路及原理	15
4.3	设计过程	17
4.4	案例展示	21
第五章	案例应用.....	26
5.1	5G 时代与移动端 WebAR 应用	26
5.2	webAR 在 H5 新媒体的应用	26
5.3	文化遗产保护应用.....	26
5.4	汽车领域中的应用.....	27
5.5	医学领域应用	27
第六章	案例优势与局限性.....	28
6.1	优势	28
6.2	局限性	28
第七章	总结与展望.....	30
7.1	总结	30
7.2	展望	30

参考文献.....	319
-----------	-----

外文资料

中文译文

致谢

第一章 绪论

1.1 课题背景与意义

随着 VR/AR 的发展,越来越多的人接触到虚拟现实技术和增强现实技术并且应用到各个领域,因此,我们思考将 VR/AR 技术应用到 web 平台上,现如今,webVR/webAR 逐渐进入广大用户的视野,由于 webVR/AR 无需下载应用就能体验,能够快速的捕捉客户。众所周知,虽然许多 VR/AR 技术可以应用到多种平台,但是比如 VR 游戏,VR 玩家和非 VR 玩家是基本不能实现共享游戏的,这就是 VR 玩家稀少比较局限的主要原因,所以 webVR/AR 的一个很关键的优势就是,具有能够跨越移动端、网站和 VR/AR 平台来达到众多用户体验的能力,这就使得 webVR/AR 可以让 VR/AR 中的用户与非 VR/AR 用户进行连接,使 webVR/AR 应用能够大范围的普及,进而影响更多的用户,得到更好的用户体验。同时,webVR/AR 技术可以让内容不再受到平台、硬件的约束,再加上现阶段很多的浏览器,例如谷歌 chrome、火狐 firefox 等这样的浏览器已经积累的大量的潜在用户,这也为 webVR/AR 浏览器市场奠定了坚实的用户基础。此外 webVR/AR 技术大大降低了开发者进入 VR/AR 领域的门槛,促使大量内容的涌进。

目前相关部门已经对外宣布将 webVR/AR 的 API 更改为 webXR,但是正在不断的完善,现在官方把它叫做“webXR Device API”。其实 webVR 和 webAR 是两个不一样的技术,不过这两个技术也有很多一样的地方,比如他们都需要头显设备,通过头显设备可以感知到用户的方向、位置以及感知。就像电影《头号玩家》一样,男主角穿上感应装备以后能感受到触觉、听觉、视觉以及痛觉,这样就能真实的感受到虚拟现实世界里所呈现的立体世界。然而这两者最明显的不同就是 AR 是将模拟层叠加在现有的世界之上,而 VR 是创造全新的模拟现实,这样二者使用的技术重叠、开发者处理的例子就会碰到相同的挑战,于是 W3C VR 团队决定开发一个 API 能够同时处理这两个技术,它就是 webXR。2017 年 google 对外宣布消息称,从 2018 年一季度开始,webXR 这个标准就会开始大范围使用,这样开发者可以在稳定版的谷歌浏览器上使用实验框架,但是由于 webXR 还在发展初步阶段,目前 webXR 还不太稳定,虽然可以进行简单的开发,但是还是会有很多的局限性,因此仍然需要不断的改进和优化,webXR 的出现,让开发者能更加容易轻松的进行开发,让开发者从迷茫到清晰的认识,而且 webXR 还支持真正的本地 AR 渲染技术。技术的发展是为了让开发者能够更加轻松更加便捷的开发,如果是以前,如果想学习 XR,就必须首先理解 3D 模型,然后再研究游戏引擎(比如 unity3D)还有编程技术,但是 webXR 如果能流行起来,很多问题都能得到缓解,这样开发者可以用现有的框架开发,不需要从头开始。这也是让 XR 用户和非 XR 用户都能够利用 webXR 技术来进行开发,这就会让开发会更加的简单和方便。

1.2 国内外研究现状

WebVR 将提供给用户一个 3D 世界的通道。虚拟现实技术(VR)之所以能够有这么大的吸引力,首要还是因为其能够给用户带来不一样的体验,这使得虚拟现实开始涌现更多的“创造性”体验。

实际上,苹果 safari 是目前唯一一个还不能支持 webVR/AR 的浏览器,这意味着为了能够让 webVR/AR 的内容可以在 safari 上快速反应,开发者必须要自己

添加能够与 webVR/AR 类似的接口来提供信息，因此就有了 polyfill 规范，虽然 polyfill 可能仍然不能适用于一些浏览器（例如安卓版的 Edge 等），但是大多数情况下还是能给到用户不错的体验。当 webVR 技术在浏览器中呈现的时候，就可以展示 3D 立体内容，同时手机端就能接受到信息并且做出反应。通过观察可以发现在浏览器的右下角有个“enter VR”的按钮，用户可以通过点击这个按钮进入双目沉浸模式，当用户将手机与头显设备连接的时候，就可以实现网页的全面沉浸，给用户带来了更加完美的沉浸感，这就是大家都认为它是 VR 行业最强大分支之一的原因。而对于 webAR 来说，相信大家都有看过《奇异博士》中的任意门吧，现在使用 webAR 技术就能实现在现实中。随着 iPhone X 的发售出行，以及 iPhone 和安卓手机对 webAR 技术的支持，webAR 技术正逐步完善、改进并慢慢的融入我们的生活中。

在国内市场中，很多的信息技术公司都在开始研究 webVR/AR 技术，并把这两个技术不断的应用于不同的社交 app 中，比如 wechat、新浪微博等社交应用，同时也会在 webVR/AR 中陆续加入更好的识别等功能。而我们也在致力于向微信公众号中实现 webVR/AR，目前由于微信浏览器还不能支持 webVR/AR，因此只能是点击网址跳转至特定的浏览器，而该技术由于微信公众号的接口权限问题目前还无法做到，相信不久的将来，webVR/AR 一定会在微信公众号以及国内市场展现它的魅力。

1.3 研究目标与工作内容

目标：本课题主要讨论 webVR/AR 设计，并且做一个 webAR 的实验范例

内容：本课题是在 web 环境下开发 VR/AR，制作一个玛尼石博物馆的展示，能够全方位的观察玛尼石，技术方面主要是利用 aframe 框架、ar.js、artoolkit 在 webstorm 中进行 webVR/AR 的开发，导入 3D 模型，调取摄像头，识别 marker，并实现虚拟场景和现实场景的切换，最后实现交互。

1.4 本章小结

本章主要是对 webVR/AR 技术进行一个简单的介绍，包括课题研究的背景和意义、国内外研究现状、研究目标以及工作内容，对 webVR/AR 有个初步的认识，在此基础上，对 webAR 进行深入的学习和研究。

第二章 技术支持原理

2.1 A-Frame 框架

A-Frame 是一个只要用一些 web 标签和组件就能构建 webVR 和 webAR 的并且不需要下载任何插件的框架。使用 HTML 标签和实体组件来制作 WebVR/AR，适用于 HTC Vive, Oculus Rift, Daydream, GearVR, 桌面等，A-Frame 是一个构建虚拟现实（VR）3D 效果的 Web 框架。A-Frame 最初来自 Mozilla，被开发成为开发 VR 内容的简单而强大的方式。作为一个独立的开源项目，A-Frame 已经发展成为最大和最受欢迎的 VR 社区之一，更有甚者，A-Frame 已经能够支持简单的 webAR。

A-Frame 基于 HTML 的最上方，让 webVR/AR 开始变得简单。但是 A-Frame 不仅仅能够实现 3D 立体渲染，同时它的核心是一个强大的实体组件框架，为 three.js 提供了一个声明式的、可扩展的和可组合的布局。

A-Frame 支持大多数 VR 耳机，如 Vive, Rift, Windows 混合现实, Daydream, GearVR, Cardboard，甚至可以用于增强现实。更有甚者，A-Frame 实现了使用户能够完全沉浸其中的交互方式，只用使用一行 HTML，包括：canvas、场景、渲染器、渲染循环、摄像机以及 raycaster。然后，开发者可以为场景添加对象，不需要重新构建。

2.1.1 特性

（1）虚拟现实变得简单

只需要放入<script>标签和<a-scene>标签便可。A-Frame 能处理 3D 样板、VR 设置和默认控件。不需要开发者安装，并且没有构建步骤。

（2）声明性 HTML

HTML 标签能够比较容易让开发者阅读，并且可以复制粘贴和理解。基于 HTML 之上，A-Frame 适用于所有人：网页开发者，VR 爱好者，艺术家，设计师，教育者，制作者，孩子们。这使得 VR/AR 不再局限于开发者。

（3）跨平台 VR

为 HTV Vive, Oculus Rift, Windows Mixed Reality, Daydream, GearVR 和 Cardboard 构建 VR 应用程序，并支持所有相应的控制器。即使没有耳机或控制器，A-Frame 仍可在标准台式机和智能手机上使用。

（4）实体 - 组件体系结构

A-Frame 是一个功能强大的 three.js 框架，提供了一个声明式的、可组合的和可重用的实体组件结构。HTML 只是其中的一部分，开发人员可以无限制地访问 JavaScript, DOM API, three.js, WebVR 和 WebGL。

（5）性能

A-Frame 在 WebVR 的基础上进行了优化。当 A-Frame 使用 DOM 时，其元素不会触及浏览器布局引擎。而在一次 requestAnimationFrame 调用下，3D 对象会更新全部在内存中完成，开销很小。

（6）工具

由于 Web 是基于 HTML 的概念构建的，因此 A-Frame 与大多数库、框架和工具（包括 React, Preact, Vue.js, d3.js, Ember.js, jQuery 等）兼容。

（7）Visual Inspector

A-Frame 提供了一个方便的内置可视 3D 检查器。打开任何 A-Frame 场景，按下<ctrl> + <alt> + i，然后四处飞行以在引擎盖下可以查看。

(8) 组件

使用 A-Frame 的核心组件(如几何图形,材质,灯光,动画,模型,raycasters,阴影,位置音频,文本以及 Vive / Touch / Windows Motion / Daydream / GearVR / Cardboard 控件)运行。甚至获得更多组件比如粒子系统,物理,多用户,海洋,山脉,语音识别,动作捕捉,远距传物,超级手和增强现实等。

2.2 ar.js

AR.js 是一个可以在 github 上高效率地完成网络上的增强现实 (AR) 的一个库。它有几个特点第一是非常快,即使在手机上也能有效运行,即使是几年前的手机上也可达到 60 fps,第二是基于网络,这是一个纯粹的网络解决方案,因此不需要安装。并且是一个基于 three.js + jsartoolkit5 的完整 JavaScript,第三是开源,它是完全开源的,免费的,第四是它适用于任何带有 webgl 和 webrtc 的手机,目的是让人们容易做增强现实 (AR),可以在当今手机上轻松体验,并且可以使用 Web 技术轻松设计,AR.js 项目旨在帮助这些开发者或用户。所以现在,任何拥有现代手机的人都可以免费享受开放源代码的 AR,并且无需安装。由此可见,不必等待每个人都购买支持 AR 的设备(例如 Tango, Moverio 等)用自己口袋里的手机就可以实现。AR.js 基于标准,可在任何使用 WebGL 和 WebRTC 的手机上工作。当然目前还只能在安卓手机上使用。然而不幸的是,AR.js 目前不适用于 iOS。例如 Safari 尚不支持 WebRTC,但 Apple 正在努力开发它。

2.3 threex-artoolkit

ARToolkit, 是一个开源的适用于 AR(增强现实)的 SDK。threex.artoolkit 是可轻松处理 artoolkit 的 three.js 扩展。threex.artoolkit 由以下 3 个类组成:

THREEx.ArToolkitSource: 这是要进行位置跟踪分析的图像,它可以是网络摄像机,视频甚至图像;

THREEx.ArToolkitContext: 这是主引擎,它实际上会找到标记位置在图像源中;

THREEx.ArMarkerControls: 这是控制标记的位置,它使用经典的 three.js 控件 API,会确保将开发者的内容放置在标记上方。

2.4 WebGL

WebGL (全称 Web Graphics Library) 是一种 3D 绘图协议,这种 3D 绘图技术标准允许把 JavaScript 和 OpenGL ES 2.0 组合在一起,通过增加对 OpenGL ES 2.0 的 JavaScript 连接,WebGL 可以为 HTML5 的场景画布提供硬件 3D 立体效果,实现加快渲染,这样开发者就能够在浏览器留直观的清晰的展示 3D 场景和模型。WebGL 是基于 OpenGL ES 的低级 3D 图形 API 的跨平台,免版税的 Web 标准,通过 HTML5 Canvas 元素暴露给 ECMAScript。了解 OpenGL ES 2.0 的开发者可以通过使用 GLSL 将 WebGL 识别为在 Shader 基础上的 API,WebGL 1.0 公开了 OpenGL ES 2.0 功能集; WebGL 2.0 公开了 OpenGL ES 3.0 API。

WebGL 将 3D 带入网络,直接在浏览器中展现。主要浏览器厂商 Apple (Safari), Google (Chrome), Microsoft (Edge) 和 Mozilla (Firefox), 这些都是 WebGL 工作组的成员。兼容性如图所示:

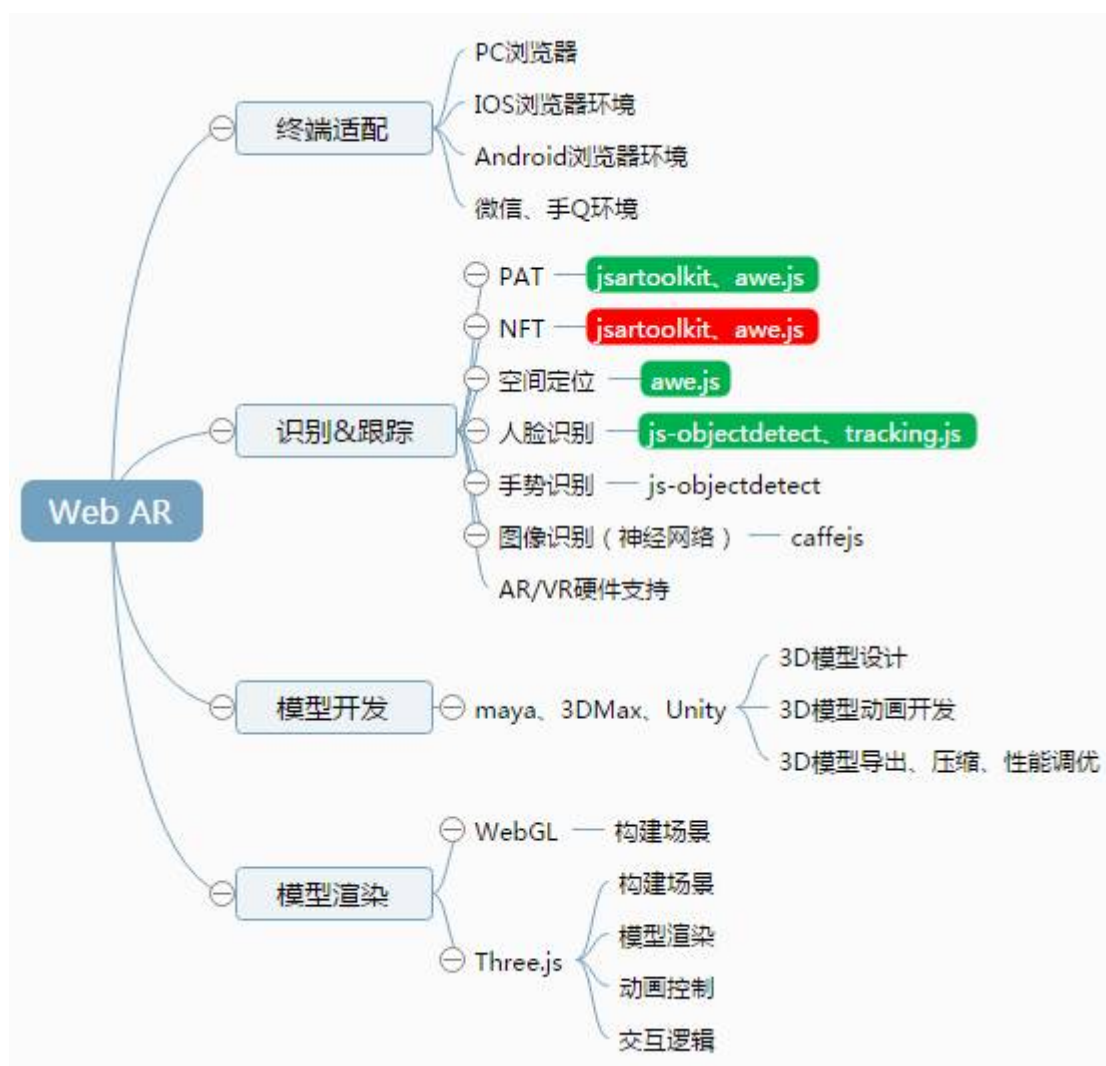


图 2-6-1 three.js 结构图

第三章 模型建立

3.1 建模工具

由于本课题建模所需要的格式是 obj、mtl 以及 mtl 所能摄取到的纹理材质格式 jpg、jpeg、png，OBJ 文件是 Wavefront 公司为它的一套基于工作站的 3D 建模和动画软件“Advanced Visualizer”开发的一种文件格式，因此对模型的要求很高，从中选出了两个适合的工具进行对比。

3.1.1 3dsmax

3D Studio Max（通常简称为 3d Max 或 3dsMAX），是一个应用于 window 电脑系统的可以进行三维动画渲染和制作的软件。现在被广泛应用于医学、动画、游戏、汽车行业等领域。具有性价比高，使用者多，便于交流，上手轻易的优势。3dsmax 导出 obj 格式文件的方法：依次点击文件—导出—导出外部文件格式—保存类型为 obj 格式。

优点：性价比高；使用者多，便于交流；上手容易。

缺点：导出的模型材质，纹理等贴图容易丢失，同时会产生很多附加文件，不利于模型的载入。

3.1.2 artec 扫描仪

Artec Space Spider 是一款面向 CAD 用户和工程师的新型增强型精密仪器，是一款基于蓝光技术的高分辨率 3D 扫描仪。这种新一代计量装置配备了高精度扫描自动温度稳定功能。非常适合以高分辨率捕捉小物体或大型工业物体的复杂细节，具有稳定的精确度和鲜艳的色彩。

扫描仪能够呈现复杂的几何形状，锋利的边缘和细小的肋骨，使我们的技术与与众不同。它是一款理想的工业三维扫描仪，可用于高分辨率捕捉诸如注塑零件，印刷电路板，钥匙，硬币甚至人耳等物体，然后将最终 3D 模型导出到 CAD 软件中。

Space Spider 在汽车零部件设计，控制质量，产品设计和制造等领域提供了很多的可能性。

这款新一代工业三维扫描仪最初专为国际空间站的规格而开发，具有强大的温度稳定性和高等级的电子设备，不仅可以使扫描仪在三分钟内达到最高精度，而且可确保数据采集的长期可重复性，设备的准确率不受环境条件波动的影响。在这款功能强大的三维扫描仪中，该技术经过精心研制，确保扫描质量达到最佳状态，并确保设备的稳定性。

优点：

- (1) 达到精度要求，携带方便，为解决中小型物体提供了良好的方案。
- (2) 分辨率高。
- (3) 扫描速度、质量、便携式令人震撼。
- (4) 使用简单。
- (5) 用时短。
- (6) 高分辨率、高精度，高测量工具。

缺点：价格昂贵。

3.2 建模过程

(1)在电脑上安装 Artec Studio,并将 Artec 扫描仪与电脑连接(通过 USB),等待驱动安装完毕,在计算机上运行安装好的 Artec Studio,新建项目,并且点击左侧菜单栏的“扫描”进入扫描界面,根据玛尼石的形态复杂特征,选择“几何 + 纹理”选项。其他选项如果没有特殊要求为默认设置。如图所示:



图 3-2-1 Artec Space Spider

(2)将被扫描的玛尼石正面朝上放置在准备好的旋转台,调整位置,将 Artec 扫描仪对准玛尼石与旋转台形成 45° 夹角。启动开始按钮,同时启动旋转台,观察扫描成像区,可以看到扫描的部分,以及左边的距离-信号图表,控制距离在绿色区间,扫描一周即可,扫描完毕后,关闭按钮,暂停扫描。

(3)将玛尼石背面朝上放置在旋转台上,重复(2)的操作,扫描完毕。

(4)将取得的两组扫描信息进行后期处理。首先是对齐,选中右边的扫描数据,接着点击“对齐”,然后选中扫描数据,点击“自动对齐”。如果“自动对齐”不能完全将不同角度的扫描信息对齐,则可以点击其下面的“对齐”按钮,手动微调至对齐即可。如图所示:

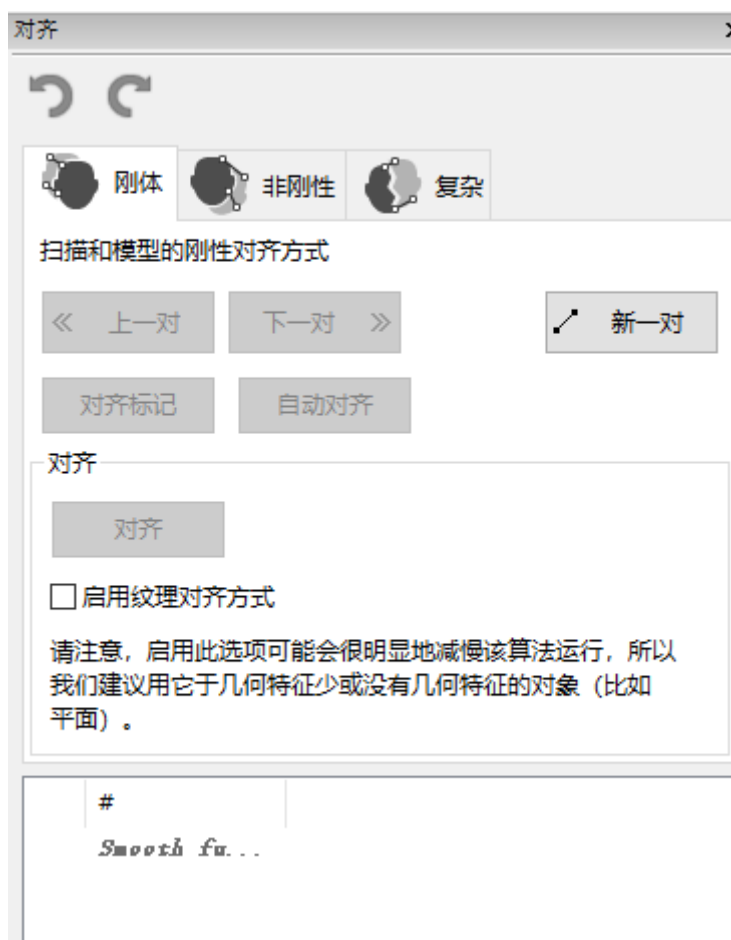


图 3-2-2 Artec Studio 对齐界面

(5) 接着对模型进行修整。点击“工具”，进入工具模块，依次选择“整体配准”、“删除离群噪点”、“光顺融合”选项。如图所示：



图 3-2-3 Artec Studio 工具界面 1



图 3-2-4 Artec Studio 工具界面 2

(6) 导出纹理和材质。选择贴图“导出”选项，下面则勾选“启用纹理正常化”和“修复缺少的纹理”，根据具体实际情况来选择分辨率，这里选择的是4096*4096。如图所示：

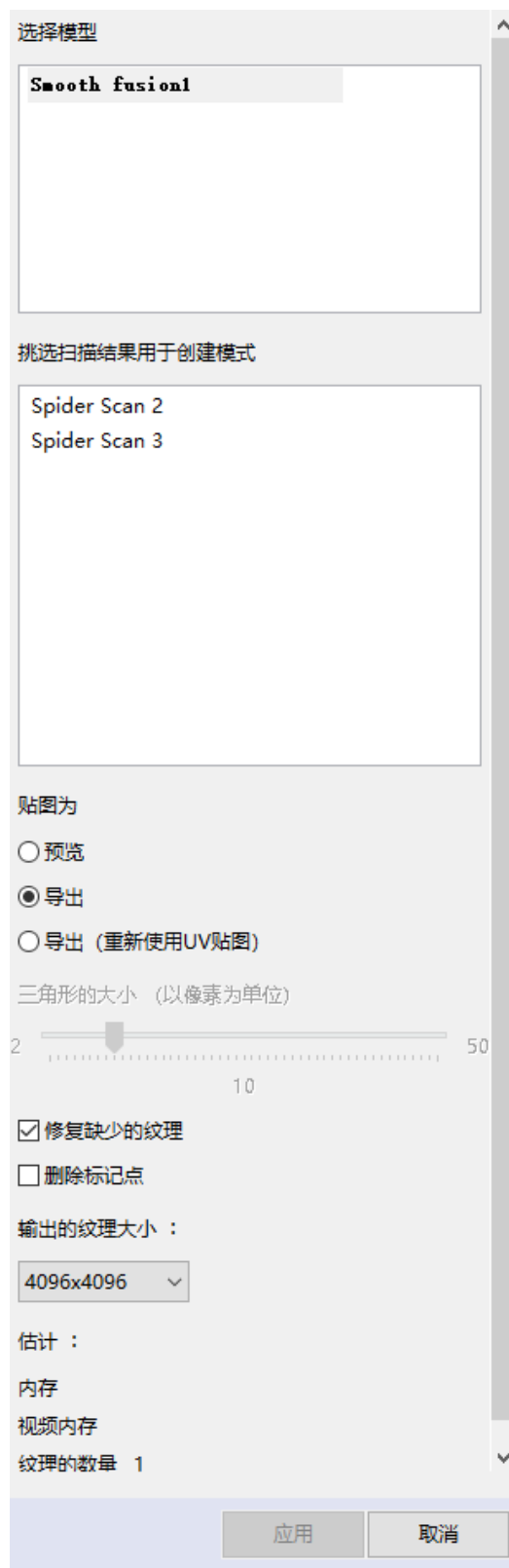


图 3-2-5 Artec Studio 纹理导出界面

(7) 导出模型。点击菜单栏的“文件”，然后再点击“导出”，保存为所需要的 obj 格式，保存后查看导出的模型有三个文件：“obj”，“mtl”以及“png”格式，这样就成功了。如图所示：

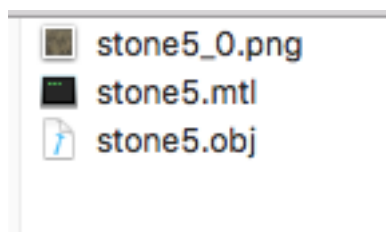


图 3-2-6 Artec Studio 模型导出文件



图 3-2-7 模型渲染效果

3.3 Artec 扫描建模注意事项

(1) 对于复杂的玛尼石模型扫描，当旋转台旋转的时候，需要及时的对扫描的位置和角度进行调整，尽量保持 artec 扫描仪与玛尼石的角度和距离保持一致，因此在扫描的时候要根据玛尼石的旋转来调整位置，同时还要保持 artec 扫描仪的平稳性，尽量不要抖动，如果 artec 扫描仪对物体的距离和角度发生了改变，就有可能无法扫描，并且在 Artec Studio 中会发生错位的现象，构建的模型就不准确，由此反复的话会耗费时间。

(2) 扫描时，尽量保持在旋转台旋转一到两周之内完成，因为 artec 扫描是会不断的叠加的，扫描的时间越长，那么就会越占内存空间，而占的内存空间越大，设备的流畅性就会受到影响，会发生卡顿的可能，因此为了保持设备的流畅性，同时也缩短扫描所消耗的时间，尽量在旋转台旋转到一到两周内完成，避免重复扫描。

(3) 在融合选项中，提供了三种不同的融合方式：快速融合、尖锐融合和光顺融合。快速融合就是能够加快模型的加载效率，但是融合的效果相对较低，存储空间占据最小。尖锐融合是模型融合效果最好，但是模型的加载效率比较低，存储空间占据最大。光顺融合则是一种比较平衡的方式，模型融合效果和模型加载效率都相对适中，则本课题选择模型融合效果和模型加载效率都适中的光顺融合。

(4) 在导出的模型格式中首先必须有个 obj 格式，然后是管理材质纹理贴图的 mt1 格式，这个格式可以查找 jpg、jpeg、png 等图片格式，将这几个格式文件放在一个文件夹中，并且不能参杂其他多余的格式的贴图，否则在载入模型的时候就不能识别并且呈现。

第四章 详细设计

4.1 设计工具

4.1.1 WebStorm

WebStorm 是一个适用于 JavaScript 的开发工具。WebStorm 为开发者提供适用于现代 JavaScript 的全部有效功能，并且能够使用智能代码，即时 bug 检测，庞大的 JavaScript 导航系统和重新构造的能力，同时拥有 TypeScript 功能，样式表语言和目前最流行的框架。在 IDE 中轻松调试客户端和 Node.js 应用程序，充分利用 linters，构建工具，测试运行器，REST 客户端以及更多工具，这些工具都与 IDE 深度集成。但是任何时候开发者需要终端，它也可以作为一个 IDE 工具窗口。立即在编辑器中查看测试状态，或在方便的树状视图中快速跳转到测试。使用简单的统一用户界面来处理 Git, GitHub, Mercurial 和其他 VCS。与此同时，开发者可以在 IDE 编辑器中上传文件，然后通过查看更改，来处理各种矛盾。WebStorm 平台提供了有效的工具，可适用于 debug，测试，跟踪客户端和 Node.js 应用程序，WebStorm 可以快速并且有效率的完成以上目标。WebStorm 与用于 Web 开发的常用命令行工具结合，为开发者提供高效，简单的开发体验，并且能够实时的保存，避免电脑突然的关机丢失文件。

特性：

(1) 智能编码协助

WebStorm 为开发者带来了基于 JavaScript、Node.js、HTML 和 CSS 的智能编码辅助功能。实现代码完成，强大的导航功效，即时 bug 检测以及所有这些语言的重组。

(2) 现代框架

WebStorm 为 Angular, React, Vue.js 和 Meteor 提供高级编码帮助。能够享受 React Native, PhoneGap, Cordova 和 Ionic 的移动开发支持，并使用 Node.js 为服务器端开发。所有在一个 IDE。

(3) 智能编辑器

IDE 会分析开发者的项目，为所有支持的语言提供最佳的代码完成结果。数百个内置检测报告会在开发者键入内容并提供快速修复选项时报告任何可能的问题。

(4) 导航和搜索

WebStorm 可帮助开发者更有效地解决代码，并节省处理大型项目时的时间。只需点击一下即可跳转到方法，函数或变量定义，或搜索用法。

(5) 调试

WebStorm 为开发者的客户端代码（适用于 Chrome）和 Node.js 应用程序供应了一个内置的调试器。放置断点，浏览代码并评估表达式，所有这些都不会离开 IDE。

(6) 测试

WebStorm 与 Karma 测试运行器，Mocha, Jest 和 Protractor 集成，轻松进行测试。在 IDE 内部实现运行和 debug 测试，以良好的可视化形式检测结果，并且实时导航到测试代码。

(7) 追踪和分析

WebStorm 具有 spy-js 功能，这是一款内置工具，可帮助开发者追踪 JavaScript 代码。探索文件如何与函数调用进行连接，并有效识别任何可能的瓶颈。

(8) 代码质量工具

除了数百个 WebStorm 自己的检测外，它还可以针对开发者的代码运行 ESLint, JSCS, TSLint, Stylelint, JSHint 或 JSLint，并在编辑器中即时突出显示任何问题。

(9) 项目模板

使用受欢迎的项目模板（如 Express 或 Web 入门工具包）从欢迎屏幕启动新项目，并通过与 Yeoman 集成来访问更多的项目生成器。

4.1.2 安卓手机

AR 技术的目标是在电脑屏幕或者手机屏幕上讲虚拟模型、场景镶嵌在现实场景中并且实现交互。所有安卓手机都能体验到 WebVR/AR 平台，由于 iPhone 并没有对 webAR 开放权限，无法获得本地摄像头，iPhone 手机若是要实现 AR 功能，需要通过摄像头或者创建新的传感器，然后定位开发者要拍摄的物体，并通过 Slam 算法在手机上实现出来，这需要等待好几年甚至更久。

4.1.3 firefox

firefox 是一个相对其他浏览器来说比较简洁的浏览器。不会包含大多数用户所不需要的功能和插件，因此，firefox 不太容易崩溃，同时 firefox 的安全性也是本课题选择它的原因，迄今为止，firefox 还没有成为一些黑客们的重要攻击对象。同时 Firefox 有一个优势，用户可以从一个浏览器窗口的单个标签页面中打开各个网站，这样就避免了在浏览器窗口之间的频繁切换，只需要点击工具栏下显示的标签，就是可以在两个或者多个不同的网页之间切换。同时 firefox 的用户量也相对较多，这也为 webVR/AR 的发展奠定了一定的用户基础。去年浏览器出现了个漏洞，这个漏洞存在于 Microsoft Edge 浏览器、老版本的 Google Chrome 浏览器（CVE-2017-5033）以及基于 Webkit 的浏览器（如 Apple Safari, CVE-2017-2419）中。为了安全策略，不得不修复这个漏洞，只能访问 https 协议的本地网页，由于本课题要调取本地摄像头，所用的是 http 协议，但是无法在这些浏览器中打开，而 firefox 是唯一能够访问的浏览器，因此本课题决定用 firefox 进行开发。

4.2 设计思路及原理

4.2.1 设计思路

本课题主要设计一个博物馆玛尼石展示的一个 webAR 案例，由于传统的博物馆展示馆中的藏品比较昂贵，有时候距离较远，不能近距离的观察，在这样的基础上加上了 webAR 技术，可以通过扫描藏品旁边的标记 marker 图像，在手机上立刻呈现一个 3D 立体效果，并且附上文字介绍，可以让参观者更加直观更有沉浸感的了解和感受玛尼石藏品，同时也增加了趣味性和参观者的参与性，如果想要更仔细的观察，也可以将藏品放大，摆脱了真实藏品的麻烦性，增加了观察者的参与感，让观察者更加能够参与其中。

4.2.2 设计原理

AR 的优势在于把目之所及的现实场景变成了背景，并将现实世界和数字世界无缝连接，而 webAR 虽然是在开发初期，但是非常的有前景，AR 的实现主要有两种方式：光学透视式 (Optical see-through) 和视频透视式 (Video see-through)。目前，市面上的头戴式设备通常采取 1 种或 2 种都采取，而本课题所需要的手持设备（手机、平板等）通常采用视频透视式。而目前，web 端想要实现 AR 技术，主要还是靠视频透视式技术。要想实现 webAR 技术这里需要三步：设备识别、追踪感应器和 3D 渲染，最重要的步骤就是设备识别和追踪感应器。

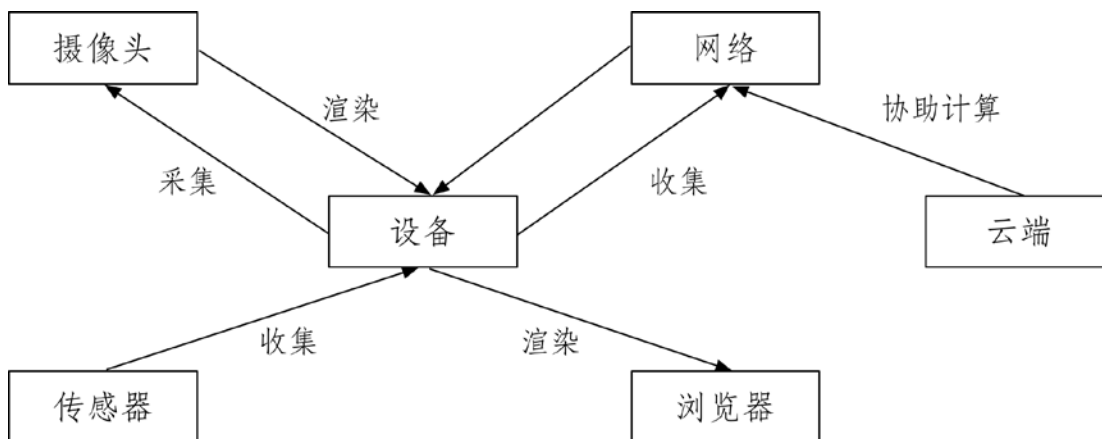


图 4-2-1 webAR 流程图

当然目前还没有涉及到云端。webVR 规范中的部分 API 对 webAR 也同样适用，本课题将用 aframe 框架进行开发，这是一个封装了一个 three.js 的库。

4.2.3 webRTC 获取视频流

首先 AR 要识别，那就会用到 WebRTC 技术，WebRTC（全称 Web Real-Time Communication），就是一个支持 web 浏览器进行实时语音或视频对话的技术。它其中有个很重要的接口函数 `getUserMedia()`，可以实时获取所添加的摄像头的视频流，这样就可以调取摄像头，完成了 webAR 中的重要一步。

4.2.4 aframe 渲染与交互

A-Frame 是 Mozilla 团队在早年开发的一个做 WebVR 的框架，但日前 A-Frame 团队发布的 aframe-xr 其中包括了一些 Web AR 组件。在之前的章节中也介绍了 webVR/AR 技术是有一些内容是相似的，因此利用 A-Frame 框架中的标签可以让开发者用相对较少的代码来制作 webAR。aframe 框架对 three.js 进行了封装和优化，因此只需要用一些标签和组件就能完成 webAR 的设计。Three.js 直接提供了一个 Raycaster 类，能够实现 ray casting 算法。这个算法的实现理论很简单，就是把设置的摄像头（这里的摄像头不是指手机的摄像头，而是开发者在渲染时的 Camera）看作视点，与开发者在屏幕上触碰的点坐标连起来构成一条射线，观察这条射线和开发者视图中的哪些物体有相交。事实上在实现 AR 技术的时候，设备识别、感应器追踪和 3D 渲染交互是同时进行的。

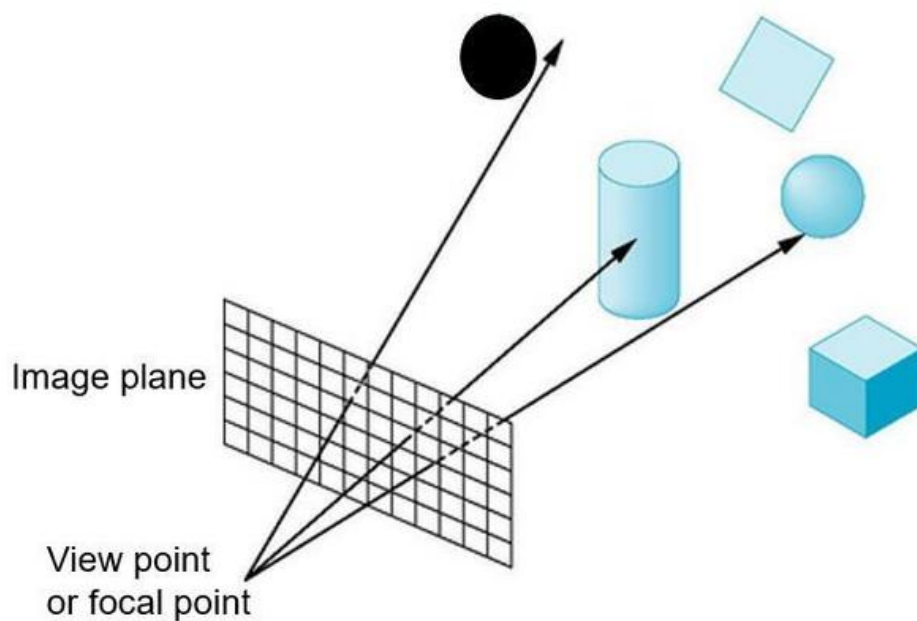


图 4-2-2 Ray casting 算法

4.3 设计过程

4.3.1 下载 js 库

A-Frame 非常易于使用，为了确保 AR.js 能够很好地运行。因此，结合 A-Frame 和 AR.js，每个开发者都能很容易在网络上创建 AR 内容。在 github 上有很多开源的 js 库，下载 aframe.js 和 ar.js，本课题下载的是 aframe.min.js 和 aframe-ar.js，如图所示：

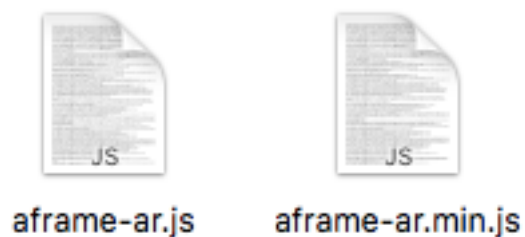


图 4-3-1 下载的 js 库

```
<!-- include a-frame -->
<script src="vendor/aframe/build/aframe.min.js"></script>

<!-- include ar.js for aframe -->
<script src='../build/aframe-ar.js'></script>
<script>ARjs.Context.baseURL = '../..three.js/'</script>
```

图 4-3-2 加载 js 库

4.3.2 创建场景

添加场景是由<a-scene>元素来表示。场景是全局根对象，所有的实体组件都包含在场景中。场景从 Entity 类继承，因此它继承了它的所有属性，方法，附加组件的能力以及等待其所有子节点（例如<a-assets>和<a-entity>）的行为。在开始渲染之前加载。<a-scene>标签为我们处理了所有的 three.js 和 WebVR 样板文件：

设置画布，渲染器，渲染循环；

默认相机和灯；

设置 webvr-polyfill, VREffect；

添加 UI 以输入调用 WebVR API 的 VR；

而这些同样适用于 webAR，如果要包含 AR.js，就需要包含 aframe-ar.js。然后在<a-scene>中初始化 ar.js。如图所示：

```
<!-- Define your 3d scene and enabled ar.js -->
<a-scene embedded arjs='trackingMethod: best;'>
```

图 4-3-3 添加场景

Embedded 是嵌入的意思，从画布中移除全屏样式。而这里是嵌入 ar.js，只有在场景中嵌入 ar.js，才能实现 AR 的效果，然后设置 trackingMethod（跟踪方法）为最好的。

4.3.3 导入模型

这里需要导入 obj model，obj-model 组件使用 Wavefront 的.OBJ 文件和.MTL 文件来加载 3D 模型和它的材质纹理贴图。我们可以通过指向指定.OBJ 格式和.MTL 格式文件路径的资源来加载.OBJ 模型。

Obj:选择器指向.OBJ 文件的<a-asset-item>资源库或者指向.OBJ 文件的内联路径。

Mtl:选择器指向一个.MTL 文件或者一个.MTL 文件的内联路径的<a-asset-item>资源库。如图所示：

```
<a-assets>
  <a-asset-item id="tree-obj" src="stone5/stone5.obj"></a-asset-item>
  <a-asset-item id="tree-mtl" src="stone5/stone5.mtl"></a-asset-item>
</a-assets>
```

图 4-3-4 将 obj 和 mtl 文件放入资源

```
<a-entity obj-model="obj: #tree-obj; mtl: #tree-mtl" position="-0.5 0 0" scale="0.004 0.004 0.004" rotation="0 45 0" color="#4CC3D9">
```

图 4-3-5 加载模型

A-Frame 框架通过<a-entity>标签元素来表示一个实体。可以在实体组件中

加入我们想要添加的模型，并且设置位置、方向、大小等属性。在 A-Frame 框架中，实体本身与它的位置，旋转和缩放等组件是息息相关的。

4.3.4 添加锚点

创建一个锚点来附加你的增强现实 (AR)，<a-anchor>锚可用于创建指向另一个文档的链接（通过 href 属性），或者创建文档内的书签，hit-testing-enabled='true' 是启用命中测试为“真”。如图所示：

```
<!-- Create a anchor to attach your augmented reality -->  
<a-anchor hit-testing-enabled='true'>
```

图 4-3-6 添加锚点

4.3.4 获取摄像头

相机组件定义用户从哪个角度观看场景。照相机通常与控制组件配对，以允许输入设备移动和旋转照相机。我们可以通过修改相机实体的位置和旋转来更改视口。当退出 VR 时，相机会在进入 VR 之前将其旋转恢复到旋转状态。这是当我们退出 VR 时，相机的旋转恢复到正常的桌面屏幕，同理 AR 也是如此。开发者告诉 A-Frame 你需要 ar.js 来控制摄像机。为此，只需添加一个<a-camera>，如图所示：

```
<!-- Define a static camera -->  
<a-camera-static/>
```

图 4-3-7 添加相机

这里添加的是静态相机。

4.3.5 marker 图像标记

首先，上传自己的图像至标记生成器并生成一个模式文件 pattern-marker.patt。其次，下载出标记。第三，打印标记。这里不建议使用电子设备显示标记，电子设备设计频闪问题会影响标记的识别。本课题选用默认的标记“hiro” marker 标记图像。如图所示：



图 4-3-8 标记图像

4.3.6 虚拟场景切换

这里添加一个天空盒<a-sky>，天空盒原始图像为场景添加背景色或 360° 图像。天空盒是一个大的球体，使其添加的颜色或纹理映射到内部。为了让虚拟场景呈现的效果是无缝的、连接的，我们选择的图片应该是等距的。如图所示：

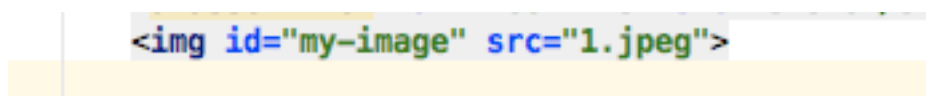


图 4-3-9 将图片放入资源库

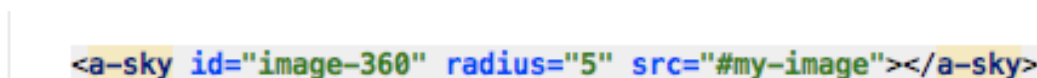


图 4-3-10 添加天空盒

将图片放入天空盒后通过 radius 属性调整半径。

4.3.7 交互

首先是添加文字，mixin 提供了一种组合和重用常用组件属性集的方法。使用<a-mixin>元素进行定义，并将它放置在<a-assets>资源库中。<a-mixin>标签应该设置一个 id 或者 class，当一个实体将该 id 设置为它的 mixin 属性时，那么这个实体就会查看并且吸纳所有的 mixin 属性。除此之外，还可以创建多个 mixin 属性复合使用，与使用<a-entity>标签来为场景添加一个对象不同，我们使用<a-mixin>来创建可多次使用的元素，使用它们就像使用一般的实体一样。为了让多个属性的组件能够整合在一起，需要用<a-mixin>标签或者多个实体去 define。在这里用来添加文字到资源库，设置文字居中。如图所示：

```
<a-mixin id="text"
  text="align: center; width: 2;
  value: There are many moments in the Tibetan language: weng,ni,mi,mie,hong (transliteration in T
></a-mixin>
```

图 4-3-11 文字内容添加到资源库

这里用<a-text>标签，这是包装文本的一个组件。可以用这个组件来添加文字，调整文字的位置、方向、字宽以及每行的字数。如图所示：

```
<a-text mixin="text" position="0 1 -1" wrap-count="70" rotation="-45 0 0"></a-text>
```

图 4-3-12 添加文字

其次是本课题要展示的是玛尼石，希望能够全方位的观察玛尼石，因此要设置旋转。通过在 A-Frame 框架中添加动画效果，并将<a-animation>元素作为实体的子元素进行动画处理，这样处理方便并且有效。

为了定义一个围绕 z 轴的实体需要 20 秒，我们可以抵消它的位置并为父实体的旋转制作动画。该动画从围绕 0 度的 z 轴的初始旋转开始，并绕 360 度旋转。它定义为 20000 毫秒的持续时间，在动画的每个循环中保持最终值，并无限循环。如图所示：

```
<a-entity obj-model="obj: #tree-obj; mtl: #tree-mtl" position="-0.5 0 0" scale="0.004 0.004 0.004" rotation="0 45 0" color="#4CC3D9">
  <a-animation
    attribute="rotation"
    from="0 45 0"
    to="0 45 360"
    direction="normal"
    dur="20000"
    repeat="indefinite"
    easing="ease">
  </a-animation>
```

图-4-3-13 添加动画

attribute: 这个属性是设置动画效果的选项。为了实现动画效果，就要指定组件的属性，这里设置为旋转。

from: 开始值。

to: 结束值，必须指定。

direction: 动画的方向（from 和 to）之间。可选值有 alternate, alternateReverse, normal, reverse。

dur: 动画的持续时间（毫秒）。

repeat: 重复计数或不确定。

easing: 动画缓和功能，有很多可供选择。

4.4 案例展示

(1) 代码展示

在<script>中加载 aframe.min.js 和 aframe-ar.js 这两个库，并插入 artoolkit 插件，首先在<body>中添加 3D 场景<a-scene embedded arjs='trackingMethod: best;'>，在场景中嵌入 ar.js，接着在资源库<a-assets>中加载想要展示模型、文字和虚拟场景的背景图片。这里放入了 4 号、5 号和 6 号玛尼石。接下来添加一个锚点 anchor，加强增强现实（AR），然

后就是添加模型，在属性中调整模型的位置，方向以及大小，这里是将模型固定在 marker 标记图像中，然后添加模型的动画效果，这里添加的是旋转，能够 360 度的无限旋转，转速为 20000 毫秒，接下来是将文字介绍放置在模型之上，设置位置、方向以及每行的字数。设置好实体组件了以后就是添加一个天空盒 <a-sky>，由于此次课题需要做一个博物馆的展示，因此添加了一个博物馆的背景图。以上这几个组件都是写在锚点中，为了能够加强增强现实，最后是添加照相机，这里添加了一个静态照相机。到这里一个 webAR 案例就完成了。代码如下：

```
<script src="vendor/aframe/build/aframe.min.js"></script>

<!-- include ar.js for aframe -->
<script src='../build/aframe-ar.js'></script>
<script>AR.js.Context.baseUrl = '../three.js/'</script>

<!-- start the body of your page -->
<body style='margin: 0px; overflow: hidden;'>
  <!-- add some info at the top of the page -->
  <div style='position: fixed; top: 10px; width: 100%; text-align: center; z-index: 1;'>
    <a href="https://github.com/jeromeetienne/AR.js/" target="_blank">AR.js</a> - tango example for a-frame by <a href="https://twitter.com/jeron">
  </div>

  <!-- Define your 3d scene and enabled ar.js -->
  <a-scene embedded arjs='trackingMethod: best;'>
    <a-assets>
      <a-asset-item id="tree-obj" src="stone5/stone5.obj"></a-asset-item>
      <a-asset-item id="tree-mtl" src="stone5/stone5.mtl"></a-asset-item>
      <a-asset-item id="tree-obj2" src="stone6/stone6.obj"></a-asset-item>
      <a-asset-item id="tree-mtl2" src="stone6/stone6.mtl"></a-asset-item>
      <a-asset-item id="tree-obj3" src="stone4/stone4.obj"></a-asset-item>
      <a-asset-item id="tree-mtl3" src="stone4/stone4.mtl"></a-asset-item>
      

      <a-mixin id="text"
        text="align: center; width: 2;
        value: There are many moments in the Tibetan language: weng,ni,mi,mie,hong (transliteration in Tibetan), various Buddhist classics and
      ></a-mixin>
    </a-assets>
    <!-- Create a anchor to attach your augmented reality -->
    <a-anchor hit-testing-enabled='true'>

      <!-- Add your augmented reality here -->
      <a-text mixin="text" position="0 1 -1" wrap-count="70" rotation="-45 0 0"></a-text>
      <a-entity obj-model="obj: #tree-obj; mtl: #tree-mtl" position="-0.5 0 0" scale="0.004 0.004 0.004" rotation="0 45 0" color="#4CC3D9">
        <a-animation
          attribute="rotation"
          from="0 45 0"
          to="0 45 360"
          direction="normal"
          dur="20000"
          repeat="indefinite"
          easing="ease">
        </a-animation>
      </a-entity>
    </a-anchor>

    <a-sky id="image-360" radius="5" src="#my-image"></a-sky>
  </a-scene>
</body>
```

图 4-4-1 代码展示 1

```
<a-sky id="image-360" radius="5" src="#my-image"></a-sky>

</a-anchor>

<!-- Define a static camera -->
<a-camera-static/>

</a-scene>
</body>
```

图 4-4-2 代码展示 2

(2) 案例展示 (PC 端)

本课题将相关代码 push 到了自己的服务器，由于兼容性的问题，选择在 firefox 中打开网页，将该课题的文件上传到服务器后，可以在网址栏输入网址 <http://www.tjuwebvr.com/AR.js-master/aframe/examples/basic.html>，首先会弹出一个是否允许打开摄像头，点击“同意”之后，则出现了一个摄像头打开的界面，之后将识别标记 marker 图像对准摄像头，则出现了以下界面，如图所示：

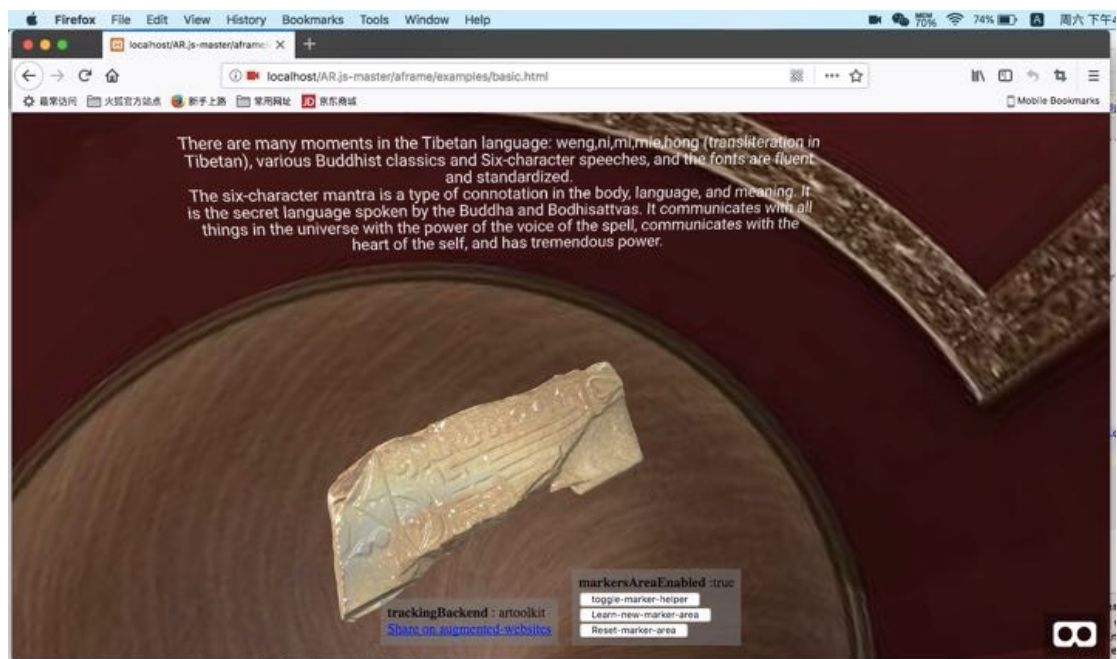


图 4-4-3 4 号石头 PC 端 1

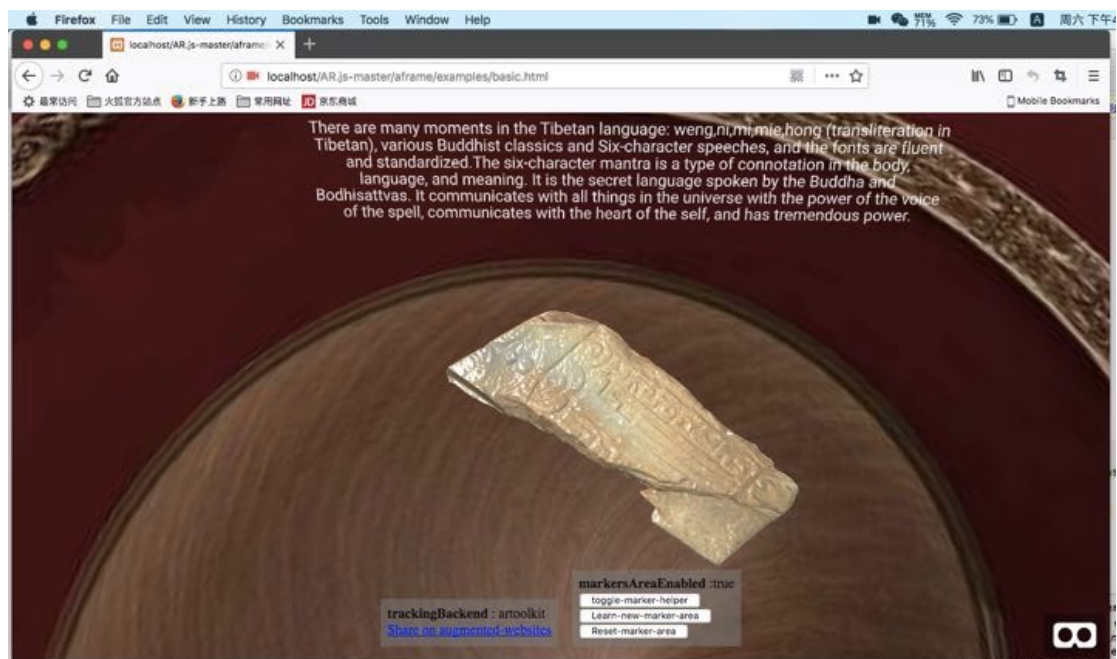


图 4-4-4 4 号石头 PC 端 2

(3) 案例展示（手机端）

在手机端的 firefox 中打开已经上传到服务器的网址，并在网址栏输入网址 (<http://www.tjuwebvr.com/AR.js-master/aframe/examples/basic.html>)，这里需要等待一些时间，因为模型的精度比较高，需要加载的时间，手机端也会弹出摄像头允许通知，点击“同意”，则打开了摄像头界面，这里打开的是手机后置摄像头，此时将手机摄像头对准打印好的标记 marker 图像，便能呈现如下场

景，如图所示：



图 4-4-5 手机端摄像头打开界面

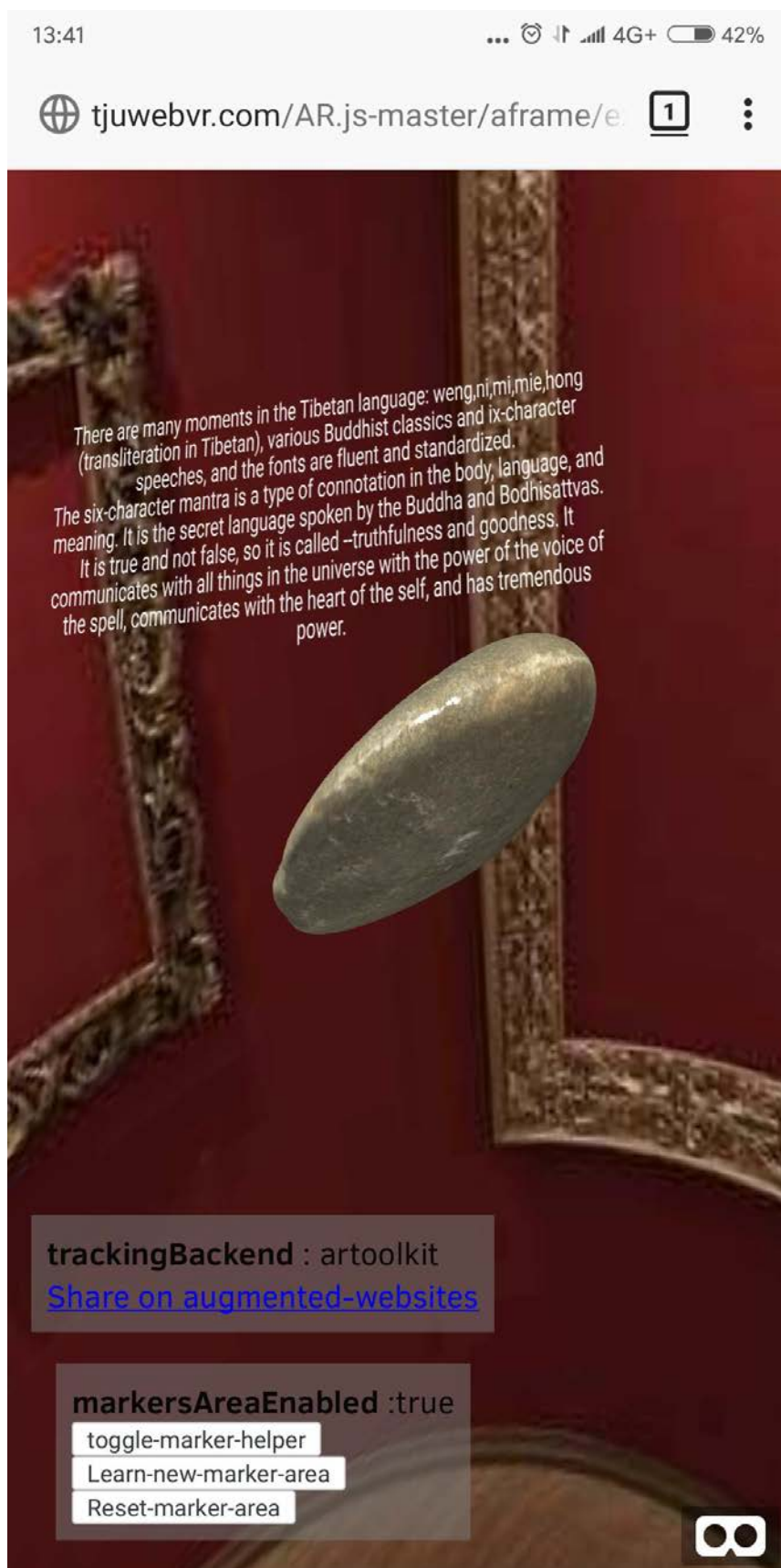


图 4-4-6 5 号石头手机端

第五章 案例应用

5.1 5G 时代与移动端 WebAR 应用

在我国信息技术的大力推广下,我们即将迎来了 5G 时代在国内的商业试点,况且它的技术正在不断的成熟化。不同于 4G 时期,此次将会融合了大数据、移动互联网、人工智能、VR、AR 等新兴技术,由此可见 5G 时代不再只是局促的通信范畴规模,更是经过与大数据、移动互联网、人工智能、VR、AR 等新兴技术的结合,在各个企业、民用领域能够深入的融合,这也预示着我们将迎来新一波的信息技术时代,是对 VR、AR 的良好接纳。

VR 技术是 3D 模型的一个交互环境,而 AR 技术是在 VR 技术的基础之上,VR 技术的到来使得用户再也不户再也不只是通感观过视觉的感观效果所感知,它是通过全方位的视觉、听觉、触觉等感知,实现了人与人、人与自然、人与物之间的交互,这使得用户接收的信息量更大了,更真实了,这都是符合人类的应用习惯的。而物联网技术使得人们生活的信息来源更加朴实朴实、真实和有效率。这使得我们的生活正逐步向着智能化和信息化发展,这也是我们所追求的,随着时代的发展,这两个技术能够实现结合,也为这种新的实现交互的方式奠定了一定的技术基础。技术的普及应用,也将会改变人们目前固前固对于外部信息的接收方式以及互习惯,通过这两个技术的结合,也使得 AR 技术更加的有实用性,而不是纯粹的好玩有趣,而是将这些运用到我们的实际生活中。

而对于 webAR 技术,特别是对于智能移动终端的 webAR 技术,也在慢慢的得到普及与接纳,这种不需要下载任何 app 的方式,只需要通过网页浏览器就能打开 AR 应用,给广大用户提供了便利,况且 webAR 还可以应用在我们熟悉的社交应用上,比如微信、微博、今日头条等,实现了快速的传播,同时也为实现 webAR 技术各个平台的实现奠定了坚实基础,同时也提供了新的研究方向。

手机端的 webAR 无疑是 webAR 技术的重中之重,在手机端,开发者只需要在社交应用中用超链接的形式通过微信朋友圈、新浪微博、微信公众号等方式进行传播,就能加快 webAR 技术对用户的影响力。用户只需要在浏览器中点开网址,并将摄像头对准标记 marker 图像,就可以看到 3D 效果,给人视觉的享受。

5.2 webAR 在 H5 新媒体的应用

尽人皆知,传统的 H5 都是只能局限在屏幕上对它进行点击操作,而当 webAR 加入到了 H5 中,就可以把用户带到了真实世界里,在真实世界里操控这些本来只能在屏幕上操控的功能,并且每个打开 H5 的用户都能够获得自己独特的用户体验,使得用户更加的有沉浸感和新鲜感。能够让用户参与到其中,提升了用户的参与感。由此可见,webAR 的出现,能够更好的满足了新时代用户彰显独特个性的需求,无疑是为传统的 H5 营销注入了新的能量。

不同于传统 H5 的普通点击和滑动等操作,webAR 可以实现触觉上的感受,给用户带来真实感和沉浸感,让 H5 具有更强的互动性和可玩性。同时本案例中将虚拟场景与真实场景的融合,给用户不一样的体验。

从大趋势上来看,webAR 会给 H5 带来全新的互动展示形式,摆脱传统的形式,带领 H5 走上全新的时代。

5.3 文化遗产保护应用

在文化遗产中引用 webAR 技术还在初步阶段，很多博物馆以及历史文化遗产的保护也都开始涉足于 webAR 技术，可以通过遗产展示、文字介绍等一系列手段，给文化遗产的信息化保护增加了很多的可能性，并且让体验用户有很强烈的参与感，让文化遗产焕发新的生机。目前有很多的博物馆都开展了 AR 的展示，比如旧金山博物馆兵马俑展、敦煌莫高窟 AR 展览、百度 AR 博物馆、纽约 MOMA 博物馆等等。

本案例呈现的也是一个博物馆的展示，将制作好的玛尼石模型加载到 webAR 中，通过手机摄像头对标记 marker 图像的识别扫描，在网页中呈现玛尼石，并且有相关石头的简介。这位文化遗产的保护以及数字化处理都开辟了新的途径，让体验者对文化遗产充满了兴趣，也能体验到文化遗产背后的历史情境。

5.4 汽车领域中的应用

随着 VR/AR 技术的流行，汽车行业领域也越来越多的使用 VR/AR 技术。首先是汽修教育，通过一个 webVR 系统，不再只是枯燥的课本或者 ppt 的学习，可以更直观的通过 3D 立体效果来对一辆汽车进行维修教程，这样的方式更加的直观，学员只需点开一个网页，就可以进入 VR 视野进行学习，随时随地的学习，学员能够体验 3D 视野的汽车维修，方便而且高效。其次是汽车评审，目前的一下企业评审人员一般都是异地的，而普通的汽车评审需要经过汽车进行油泥模型喷漆等等工序，耗时大概几个月，而如果加入了 webVR/AR 技术，则可以建立汽车模型导入，对汽车的零部件以及外观进行评审，既缩短了评审时间，又能高效率的完成，也能促进汽车行业的发展。

5.5 医学领域应用

随着医学技术不断发展，医学与信息科技正在不断的融合。可以将 webVR/AR 技术与中医结合，建立一套标准的中医养生推播商业模式，可以让医学院的团队可以简易、方便的方式来设计想要推播的中医养生信息。摆脱了传统的文字图片的枯燥方式，而是用一种 3D 立体的效果，让其患者可以很直观接受相关中医知识，可以让患者达到沉浸式学习与体验。

第六章 案例优势与局限性

6.1 优势

由于去年 Pokémon Go 的出现,掀起了 AR 的浪潮,摆脱了昂贵的头显硬件,使用手机便能体验 AR 技术。从 AR 设备、本地 app 再到现在的 webAR 技术,这是一个市场化普及的过程。不同 AR 类型的技术特性决定了对软硬件的需求。这也就凸显了 webAR 技术的优势。它的适应性更强,开放程度更高,更符合当前面向多种用户体验习惯。

(1) 多用户性

与以往的 AR 不一样,以往的 AR 只能是 AR 玩家才能体验,而非 AR 玩家就不能参与其中,AR 玩家和非 AR 玩家之间就缺少了沟通,从而他们之间就会产生一定的隔阂,不利于 AR 技术的发展和应用,而且用户体验比较少,但是 webAR 技术的出现与传播,是能够连接 AR 玩家和非 AR 玩家之间的重要桥梁,能够让更多的用户轻松、便捷的体验到 AR 技术,使得 AR 技术能够得到广泛的传播和更好的普及,同时也为 webAR 技术的到来和发展奠定了一定的用户基础。

(2) 成本低

webAR 技术不需要头显设备,比如,HTC vive、Oculus Rift、hololens、AR 镜片等头显设备,这些设备的价格都相对高,例如 HTC vive 价格为 5488 元,Oculus Rift 价格为 4578 元,hololens 价格为 23488 元,对于普通的学生或者一般的开发者这样的价格太贵,开发的成本太高了。同时 webAR 技术不需要下载专门的硬件,不需要下载特定的 app,只需要用一个简单的 aframe 框架以及 ar.js 库,便可以进行 webAR 的开发,对于任何的开发者甚至小白都是个不错的选择。

(3) 降低门槛

webVR 与 webAR 是两种不同的技术,不过二者也有许多重叠相似的地方:它们都与穿戴在脸上的设备有关,都安装许多传感器,可以追踪动作、位置、方向。二者最明显的不同就是 webVR 创造全新的模拟现实,而 AR 将模拟层叠加在现有世界之上。尽管如此,二者使用的重叠技术、开发者处理的范例经常会碰到相同的挑战,于是 W3C VR 团队决定开发一个 API,同时处理两种技术,它就是 WebXR。webXR 将 webVR 与 webAR 都包装在一起,只需要一个库,便可以开发,即使你对代码一无所知,但是也能轻松的上手。

6.2 局限性

因为体验设备的硬件需求太低,因此 webAR 对于其他的 AR 技术,仍然存在着许多的问题,这也是开发者可以继续研究探讨的一部分,有缺陷才能更加的完善也会更加促进 webAR 技术的优化。

(1) 本地应用的兼容性不够高

目前由于苹果手机的权限问题,webAR 还不能在苹果手机上使用,因此只能在安卓手机上使用,并且目前只能在 firefox 浏览器打开,对用户的使用受到一定的局限性,不能满足广大用户的要求,一定程度上减少了用户体验,不能大范围的传播和推广。

(2) 识别局限

目前我们只能做到通过标记 marker 图像来识别,在一定程度上来说是很局限也很不方便的。当摄像头离开标记 marker 图像或者有点偏移的时候,就没办法

法识别，用户体验有点差。

（3）模型要求高

只有特定格式的模型才能加载到 webAR 中，目前还并不能加载别的格式的模型。因此对模型的制作要求就会非常的高，尤其是模型的纹理贴图材质，如果模型没有按照指定要求的来制作，就无法将模型加载进 webAR 中。在一定程度上模型的加载也有一定的局限性。

（4）交互少

与 webVR 的交互不同，webAR 可开发的交互目前还太少，只能是实现一些动画效果，旋转，移动等等，能让用户的体验感比较少。

（5）加载慢

本课题所需要的玛尼石模型因为纹理材质贴图的精度比较高，在当初制作模型的时候，融合方式选择了光顺融合，光顺融合相对于快速融合的加载效率没有那么高，因此，在网页每一次刷新的时候都会耗费一定的时间，并且模型的加载花费的时间也很大，有时候天空盒加载出来了，但是模型还未加载，就会影响用户的体验效果。

第七章 总结与展望

7.1 总结

本课题实现了西藏玛尼石的展示的 webAR 案例, 可以让用户 360 度的观察玛尼石, 拜托了传统的观赏方式, 提供了一些新的思路。通过本次课题的研究, 我基本掌握了 webVR/AR 技术的开发方法, 对于 webVR/AR 的技术背景、设计原理、操作方法以及最后的呈现方式和交互都有了一定的了解, 最后呈现效果达到了最初的设计目的。总的来说 webVR/AR 技术相对于其他方式呈现 VR/AR 具有很大的前瞻性和便捷性, 不仅是对前端技术有了很深的了解, 同时也对 VR/AR 技术有了不一样的体会和感受, webVR/AR 技术相对于其他呈现 VR/AR 的方式还是有一定的不同的, 相对于别的方式, webVR/AR 适合各个开发者, 上手比较简单, 能够很快的适应和掌握, 加上 aframe 框架的简化, 只需要一些标签组件便可以制作一个简单的 webVR/AR, 随着信息时代的发展, 技术不是越来越复杂, 而是将复杂的技术用简单的方式让更多的人能够了解, 体验和使用, webVR/AR 就是这么一个简单的方式, 连接着 VR/AR 开发者和非 VR/AR 开发者, 从而让更多的人能够参与其中, 这也是本课题研究的意义。

7.2 展望

webVR/AR 技术的出现不仅是一个时代的开始, 更是预示着越来越多的用户能够参与其中, 而近日, 亚马逊的一个适用于 webVR/AR 技术的平台 Sumerian 出现在大众的视野中, 这个平台的出现标志着开发者不再是用复杂的代码才能开发 webVR/AR, 只需要用很少的代码就可以实现 webVR/AR 的开发。Sumerian 是一个能够在 web 端实现 webAR 的新型平台, 降低了对开发者的要求, 开发者不需要具备很专业的编程能力, 只需要将需要的 3D 模型拖放到构建好的场景中, 或者导入来自第三方资源库的 3D 模型, Sumerian 这个编辑器可以在任意一个 web 浏览器中运行, 能够支持 ARkit 和 ARCore, 兼容性特别好。如果开发者具有一定的 coding 能力, 那么 Sumerian 还会提供一个能够写代码的接口和 Sumerian 命令行进行更加深入的制作场景, 并且编写复杂的应用程序。这个平台的出现对于 webVR/AR 内容有着比较大的意义。我们相信随着 webAR 技术的深入研究, 能够探索到更多实现交互的可能。不久的将来, webVR/AR 技术会应用到更多的领域, 还会有更多的交互, 这都是有可能的, 未来, webVR/AR 将会是大势所趋, 成为主流。

参考文献

- [1] Marx R, Vanhove S, Vanmontfort W, *et al.* DOM2AFRAME: Putting the web back in WebVR[C]//International Conference on 3d Immersion. 2017:1-8.
- [2] Barsoum E, Kuester F. WebVR: an interactive web browser for virtual environments[C]//Electronic Imaging. International Society for Optics and Photonics, 2005:540-547.
- [3] Mendigochea P. WebAR: creating augmented reality experiences on smart glasses and mobile device browsers[C]// ACM SIGGRAPH. ACM, 2017:10.
- [4] Danchilla B. Three.js Framework[M]// Beginning WebGL for HTML5. Apress, 2012.
- [5] Rodrigues A B, Dias D R C, Martins V F, *et al.* WebAR: A Web-Augmented Reality-Based Authoring Tool with Experience API Support for Educational Applications[J]. 2017:118-128.
- [6] Zhang M, Lv Z, Zhang X, *et al.* Research and Application of the 3D Virtual Community Based on WEBVR and RIA[J]. 2009, 2(1).
- [7] L. Freina, M. Ott, "A Literature Review on Immersive Virtual Reality in Education: State of the Art and Perspectives", e-learning and Software for Education Bucharest, pp. 131-141, 2015.
- [8] V. Vukicevic, B. Jones, K. Gilbert, *et al.* W3C Github. Retrieved January 04 2017 from WebVR Editor's Draft, December 2016, [online] Available: <https://w3c.github.io/webvr/>.
- [9] D. Gonzalez-Zuniga, A. Chistyakov, J. Carrabina, "Redefining a Pattern: Stereoscopic Web Perception", IEEE Latin America Transactions, pp. 514-519, 2014.
- [10] G. Saucedo-Tejada, S. Mendoza, D. Decouchant, "F2FMI: A toolkit for facilitating face-to-face mobile interaction" in Expert Systems with Applications, Elsevier, vol. 40, no. 15, pp. 6173-6184, 2013.
- [11] Mendigochea P. WebAR: creating augmented reality experiences on smart glasses and mobile device browsers[C]// ACM SIGGRAPH. ACM, 2017:10.
- [12] 鲁徐, 朱齐勇, 薛飞. WebVR 在网络教学中的应用[J]. 中国商界, 2010(11X):394-394.
- [13] 田彦, 张猛. 基于 WEBVR 和 RIA 技术的三维虚拟社区研究与应用[J]. 电脑知识与技术, 2011, 07(7X):4988-4990.
- [14] 俞璋凌. 基于 web&VR 技术的产品展示方法研究[D]. 合肥工业大学, 2009.
- [15] 谭云兰, 贾金原, 康永平,等. 基于 WebVR 的井冈山虚拟旅游系统架构设计[J]. 井冈山大学学报(自然科学版), 2012(6):46-50.

外文资料

Marx R, Vanhove S, Vanmontfort W, et al. DOM2AFRAME: Putting the web back in WebVR[C]// International Conference on 3d Immersion. 2017:1-8.

中文译文

DOM2AFrame: 将 Web 放回到 WebVR 中

1. 引言

虚拟现实 (VR) 正在上升[11]! 随着具有 VR 功能的硬件 (例如, 耳机, 控制器和个人计算机) 的普通大众的可用性的增加, 对优质 VR 内容的需求也随之爆发。尽管这些内容可以通过立体视频和互动游戏相关体验的形式得到充分的体现, 但基于网络的 (2D) 内容丰富的生态系统尚未在 VR 媒体中得到充分探索。这很遗憾, 因为在 VR 中使用标准的 2D 基于 Web 的技术 (例如 HTML, CSS 和 JavaScript (JS)) 有各种有趣的用例 (UCs)。例如:

UC0: 按原样呈现整个网页并进行互动

UC1: 在 3D 体验中呈现 2D 用户界面 (UI) 或其他 2D 元素 (例如, 游戏中的复杂菜单, 文本阅读器)

UC2: 在 VR 中观看时以不同的方式呈现 (部分) 网页 (例如, 将 3D 对象集成为电子商店的一部分, 自动选择立体视频, 水平而不是垂直放置)

最近的一些发展尝试在专用 VR 浏览器 (例如, Janus VR 浏览器嵌入网页作为更大的 3D 世界的一部分) 中解开大量现有 Web 内容, 或者作为单独的 VR 渲染引擎或视图存在于现有的网页浏览器 (例如谷歌浏览器和三星互联网)。令人遗憾的是, 这些浏览器通常只允许使用单一用例 UC0, 并且不会直接控制 Web 开发人员如何在 VR 中呈现其内容, 也不会向用户提供与其交互的方式。这是有问题的, 因为浏览器的典型方法 (简单地在平面 3D 表面 (例如, 平面, 柱面) 上显示网站) 不可能产生令人满意的用户体验[16]。

为了支持 UC1 和 UC2, 我们可能会考虑采用新的 WebVR 标准[24]。WebVR 在各种 VR 硬件设置之上提供抽象, 并使开发人员能够使用 HTML5 <canvas>元素 [21]的现有 WebGL [10]渲染功能轻松创建立体 3D 体验。通过基于 JS 的 Gamepad API [24]提供了各种 VR 输入方法。Google Chrome 和 Mozilla Firefox 目前都支持 WebVR, 而 Apple 的 Safari 最近加入了 WebVR 标准化工作[5]。

理论上, Web 开发人员可以使用 WebVR 在 VR 中呈现 2D 网页内容, 并支持所有三种用例。但实际上, 现代浏览器不允许开发人员直接访问其渲染管道, 也不提供 API 来将典型 Web 内容渲染/栅格化到<canvas>元素 1 上。不提供这种耦合或 API 的主要观点始终与安全 and 隐私相关, 因为它可能导致滥用者获取并保存敏感用户数据的“屏幕截图”(例如, 攻击者可能会在<iframe 中加载在线银行页面>, 将其渲染为纹理并将其发送到服务器;虽然这当然是一个可解决的问题, 但可能的解决方案在渲染管线中并不容易实现[20])。

无论好坏, 想要创建 VR 内容的网络开发人员目前仅限于由<canvas>和 WebGL 的 2D 和 3D 渲染上下文提供的功能, 这意味着大多数功能更强大且已建立的 Web

技术,例如 HTML, CSS 和 JS 的大部分(例如,输入事件监听器)不能直接在 WebVR 中使用。这意味着网络开发人员不能使用他们已有的技能组合开始创建 VR 内容,而是要求他们培养通常教给游戏开发人员的新技能(例如,3D 基元知识,着色器)。

在这项工作中,我们首先看看现有的选项来绕过这些限制,以及如何将全功能的 2D 网页内容呈现为交互式 3D 和 VR 体验的一部分(第 2 节)。我们发现开发人员一直在使用变通方法将网页内容呈现为 2D 纹理,然后将它们映射到 3D 原始图像以供在 VR 中使用。我们发现这些方法太慢而无法支持交互式 3D WebVR 体验,并且在处理用户时不灵活输入。我们贡献了我们自己的 DOM2AFrame 框架(第 3 节),它不是先渲染 2D 纹理,而是直接将每个 2D Web 元素耦合到等效的 3D 表示。我们的方法提供了完全交互式的帧速率和更灵活的输入处理(支持 HTML 表单),代价是渲染结果不是像素完美的浏览器引擎结果副本。我们在第四部分验证了我们的方法,其中两个案例研究显示了对 UC1 和 UC2 以及部分 UC0 的支持。请注意,我们主要关注渲染和输入捕捉方面,并考虑与(Web)VR 内容进行交互的技术超出了本作品的范围。

2. 相关工作

2.1 调整原生渲染管道

渲染一个网页是一个迭代过程(见图 1a)。浏览器引擎从原始 HTML / CSS / JS 字符串开始,并构建一个内部表示:文档对象模型(DOM2)。这个内部表示然后在多个步骤中使用,主要是 Layouts 和 Styling。布局处理元素的位置和大小(例如,使用 CSS 属性,例如宽度,高度,边距),而样式更改元素的外观(例如边框,颜色,字体重量)。计算出布局和样式后,浏览器可以使用此信息通过 GPU 绘制网页的栅格化表示。当网页上的内容发生变化时,浏览器仅更新其内部表示的相关元素,以防止为每次更新重新绘制整页。因此,浏览器在这种内部表示的基础上提供一个 VR 特定的渲染视口是很简单的,因为它主要需要改变内容被分发到 GPU 的方式,并且可以重用大部分现有的实现。难怪之前的大部分工作都集中在扩展现有的浏览器实现以实现立体渲染和/或为 Web 开发人员提供访问 VR 渲染设置的方法。在他们的开创性论文[6]中,Barsoum 和 Kuester 使用外部 VR 渲染引擎并在后台运行 Internet Explorer Web 浏览器进程。ActiveX 插件捕获网页的内容,并将生成的位图流传送到 VR 引擎。他们的结果表明刷新率仅为 4-7 FPS。Jankowski 和 Decker [16]使用自定义网页浏览器来允许用户在“普通”2D HTML 视图和 3D 视图之间进行切换,其中 HTML 内容覆盖在平面“窗口”之上。最近,王等人。[26]扩展了 Webkit,以便轻松实现 HTML 内容的立体渲染,引入了多种 API,如 HTML-S3D, CSS-S3D 和 JSS3D。同样,Lim 等人。[17]提出了几个 CSS 扩展以允许立体渲染和 3D 操作,它们使用仿真渲染引擎 3 进行

验证。然而，最终，所有这些方法都需要定制浏览器实现/插件和/或对现有 Web 标准提出扩展，使得它们难以在实践中部署，并且与现代 API（例如 WebVR / WebGL）结合使用。此外，大多数讨论的方法主要集中在支持 UC0 和有时 UC2 上（突出显示无需内置浏览器支持，传统上很难实现 UC2），但很少使用 UC1。

2.2 渲染 2D 纹理

正如前面部分所讨论的那样，在不改变浏览器源代码的情况下渲染网页内容是很困难的，因为开发人员不能通过 JS 或任何其他方式直接访问浏览器的内部渲染管道⁴。但是，有些解决方法通过使用<canvas>元素的可用 2D / 3D 渲染上下文来获得非常相似的结果。这些方法首先通过经由 SVG 图像（2.2.1 节）以迂回方式使用内部管线或者通过重新实现浏览器的样式逻辑（第 2.2 节），将（大）DOM 元素组渲染成 2D 图像/纹理。然后将此纹理映射到 VR 中使用的 WebGL 3D（四元组）基元。

2.2.1 rasterizeHTML 和 html2ree

使用基于 XML 的 Scalable Vector Graphics (SVG) 图像标准的特殊<foreignObject>标记，HTML 和 CSS 内容可以嵌入到 SVG 图像元素中。然后，<canvas> 2D 绘图 API 允许绘制/栅格化 SVG 图像，从而使浏览器在内部呈现 SVG 的方式与使用默认浏览器渲染管道的普通 HTML 和 CSS 内容大致相同。但是，再一次出于安全和隐私的考虑，各种网络功能并不直接支持。例如，在 SVG 内禁用 JS 代码执行/事件，并且外部 JS 无法访问生成的（沙盒）SVG DOM 副本。这意味着不支持与表单元素（例如<input>，<textarea>）的交互。此外，大多数外部资源（例如图像，CSS 样式表，字体）不能直接包含在内，必须首先在 SVG 之外下载，然后作为 base64 编码的字符串进行内联。在这个过程中涉及许多边界情况，浏览器不支持所有可能的 HTML 和 CSS 功能。这种方法的各种开源实现已经出现，例如 rasterizeHTML [9]。实验 html2three 库 [18] 在内部使用 rasterizeHTML 来渲染 WebVR 上下文中的网页。有限数量的顶级<div>容器呈现为单独的四边形（每个容器一个纹理/四边形）。每个容器元素都被视为一个“子页面”，可以单独重新渲染。通过捕获 UV 纹理坐标空间中的指针事件坐标，然后通过标准 document.elementFromPoint API 查找相应的 DOM 元素来提供交互，该 API 可以通过文档空间坐标查找 DOM 元素。然后通过 JS Mutation 事件系统捕获对原始内容所做的任何结果更改（请参阅第 3 节），并可通过创建新的<foreignObject> SVG 来完全重新呈现触发内容的父级<div>容器。

2.3 讨论

现有的学术著作提出了在 VR 中渲染 DOM 内容的广泛而优化的方法，但通常需要更改 Web 浏览器的源代码，使其难以在实践中使用。相比之下，渲染到 2D 纹理是一种直接适用的解决方法，可以清楚地导致浏览器行为的整体可视副本，

但它也有严重的缺点。首先，该方法在输入捕捉方面不灵活（例如，输入不再直接接收在单个 2D 元素上，表单元素支持受到限制）。其次，当 DOM 有很多变化时，重新渲染较大的 2D 纹理往往不足以提供交互式帧速率。简单的优化是简单地每个单独的 DOM 元素渲染单独的 2D 纹理，以提高更新性能。然而在实践中，用于创建和管理这些纹理的 JS API 开销很快累积起来，这种“优化”在以任何方式应用时都会损害性能，但却是一种非常粗糙的方式（即，超过 5 到 10 个单独纹理我们的测试）。这就是为什么所讨论的实现选择不同的中间道路的原因：html2three 的方法（b）适用于几个基本上独立的容器元素的情况，而 HTML GL（c）将在多个移动部件分布这一页。这也可能是两个 3D 库都不支持<video>标签的原因之一，尽管这可以通过纹理映射内部视频帧缓冲区来完成。

3. DOM2AFrame

我们的贡献 DOM2AFrame 旨在为 3D / VR 渲染 2D DOM 提供网络标准兼容解决方案，具有高更新性能和灵活的输入支持。虽然我们的方法在概念上与 html2canvas 类似，但它采用了自定义样式和渲染逻辑，但它不同之处在于，不使用 2D <canvas> API，我们使用 3D WebGL 功能去除渲染 2D 纹理的中间步骤。我们在 DOM 元素与其 3D 对象之间创建并维护一个直接映射，从而使修改后的 DOM 元素的 3D 表示更容易。因此，我们通过以下方式实现了双重性能提升：a）消除 2D 纹理开销，b）仅更新所需内容。我们直接在正确的 3D 元素上接收（基于指针的）用户输入（通过 raycasting / “picking”），并可以通过我们维护的映射将其传递给原始 DOM 元素。我们方法的主要缺点是复杂样式设置的浏览器基本事实相比，很难实现高度的视觉保真度（参见第 3.3 节）。DOM2AFrame 的概念验证实现，以及所有讨论的测试页面，案例研究和一些电影可在 <https://webvr.edm.uhasselt.be>

3.1 DOM 到 A 帧映射

实际上，大多数 DOM2AFrame 的功能并不直接使用低级 WebGL 功能，而是依赖于 Mozilla 的高级 A-Frame 库[2, 23]，后者又使用较低级别的 Three.js 库。A-Frame 允许用户使用类似 HTML 的标记来构建 3D VR 场景，例如参见清单 1 的底部。这个标记在运行时由 JS 解析并用于创建 3D 元素，为 Web 开发人员提供了一种熟悉的以声明方式组成 VR 场景[8]。请注意，本文讨论的大部分技术挑战都是我们整体方法所固有的，而不是由于选择 A 框架作为基础框架。的确，类似的结果可以通过使用例如 Three.js, ReactVR 框架[4]或 X3DOM 方法[7]上的变体来实现。

清单 1: DOM2AFrame 示例（不包括定位/大小）

```


<p style="background-color: red; color: blue;">
  ↳ Lorem ipsum dolor</p>

Is transformed by DOM2AFrame into:

<a-asset id="abc123" type="img" src="logo.png" />
<a-image src="#abc123" />
<a-entity>
  <a-plane color="#FF0000" />
  <a-text color="#0000FF" value="Lorem ipsum
  ↳ dolor" />

```

但是，尽管 A-Frame 的 HTML 类似标记，它仅支持通常具有直接 3D 等价物的自定义元素（例如<a-box>，<a-plane>），而不支持标准 HTML 元素（例如<div>，<p>），也不支持这些自定义元素的直接 CSS 操作。所以，虽然 A-Frame 的方法看起来和基于 HTML 的开发感觉一样，但它仍然不适用于缺乏以前 3D 体验的开发人员。为了解决这个问题，DOM2AFrame 从各种 HTML / CSS 元素创建自动化的自定义映射到等效的 A-Frame 对象和 WebGL 着色器。它循环遍历选定根元素的子元素，并为每个元素创建相应的 A-Frame 元素。例如，清单 1 显示了如何将<p>和元素转换为用于呈现的等效 A-Frame 表示。对于主要充当其他内容（例如，<div>，<table>）的容器的许多 HTML 元素，简单的<a-plane>是足够的模拟。显示文本的元素由<a-plane>和<a-text>元素组成，因为后者不直接支持背景颜色或边框。其他更专业的 HTML 元素（例如<select>）可能需要更复杂的设置，但通常情况下，我们的案例研究（参见第 4 节）所需的所有 HTML 元素可能由简单的 A-Frame 组件组成。

一旦建立了 DOM 和 3D 元素之间的映射关系，DOM2AFrame 就会使用浏览器的本地 Layouts 计算来定位和调整 3D 中的元素的大小，由 get-BoundingBoxClientRect 方法（所有 DOM 元素都可用）提供。此方法在浏览器根据 CSS 属性定位所有元素之后，返回 DOM 元素边界框相对于文档左上角的坐标和大小。这种方法意味着 DOM2AFrame 支持任何类型的开箱即用的复杂 CSS（正如第 2 节中讨论的使用相同 API 的其他方法一样），而且原始 2D 页面及其 DOM 必须保持活动状态（尽管它可以被隐藏）。

然而，与 Layouts 步骤相比，浏览器不能轻松访问其原生的 Styling / Drawing 引擎，我们需要自行重新实现大部分 Styling 逻辑。浏览器的内部 CSS 表示（CSSOM）可以通过 window.getComputedStyle 接口获取，该接口返回所有 CSS 属性的解释值。幸运的是，对于这些属性中的许多属性，可以使用直接的 A-Frame 类似物（例如，文本和背景颜色，背景图像，文本对齐），并且通常直接在片段着色器中实现。其他属性不太直截了当。例如，A-Frame 元素默认没有设置边框的选项，而 CSS 在这方面提供了许多灵活的可能性。因此，DOM2AFrame 通过在<a-plane>的边上创建新的细三角形条/线来支持矩形元素的边界。另一个

困难的 CSS 方面的例子是溢出（即，隐藏内容大于其父边界矩形的地方）。html2canvas 通过使用<canvas>的 2D 剪辑方法来实现这一点，该方法在光栅化过程中丢弃剪辑区域外的像素。然而，这个专长在 3D 中很难完成。我们当前的解决方案是使用四个单独的正交裁剪平面（矩形元素的每个边），并在元素的片段着色器中丢弃这些平面外的像素。此外，DOM2AFrame 还包含对各种非标准 HTML 属性/元素和 CSS 属性的支持。这些允许开发人员在 3D / VR 中更轻松地操作 HTML 内容的外观，或者创建一个外观和行为不同的网页，具体取决于它是否在 VR 中查看（以最大程度地重用源代码）。

最后，处理基于指针的输入事件（例如鼠标，VR 控制器）相对容易。A-Frame 为大量的基于光标的交互概念提供了条款，包括“凝视点击”，并使用光线投射找到正确的元素。DOM2AFrame 然后拦截这些 A 帧事件并将它们转换为等效的 DOM 事件（例如，点击，悬停，鼠标移出）。然后，它可以将 DOM 事件分派到正确的 2D 元素，然后触发已注册的现有 JS 事件处理程序并保持对事件冒泡的完整支持。

3.2 更新和动画

给定 DOM 的 A 帧表示以及相应 2D 和 3D 元素之间的映射，我们可以通过更新正确的 A 帧属性或创建/删除 A 帧元素来有效地反映任何 DOM 更改。这里面临的挑战在于如何通过 JavaScript 实际检测这些 DOM 更改。

浏览器提供了一个标准的 MutationObserver API [3]，它可以用来观察一个 DOM 元素（及其子元素），并在发生某些变化时生成 Mutation 事件（例如，更改 CSS 属性，分配新的 CSS 类名，一个子元素被创建）。虽然这种设置捕获了大多数突变，但有一些重要的边缘病例。例如，CSS 动画和转换不会触发 MutationObserver，因此需要单独的事件侦听器（例如，animationstart 和 animationend）和更新循环逻辑（通过 requestAnimationFrame）。此外，如果元素由于 CSS 属性更改而更改大小/位置，则会专门为该元素生成突变事件；其他可能因此更改而重新定位/调整大小的元素需要单独检查。最后，如果 CSS 属性不是直接设置在对象上，而是使用常规 CSS 选择器（例如，* {color: #FFF;}），则这也不会触发突变事件。目前，我们知道无法以各种可能的方式观察这种特定情况，而是要求开发人员手动调用所有受影响元素的更新方法。

3.3 限制

我们方法的主要挑战在于令人满意的性能和达到与浏览器的 2D 渲染相匹配的视觉逼真度之间的平衡。首先，大多数核心问题源于几个 CSS 样式属性在 WebGL 及其着色语言中没有直接或简单的对应物。例如，我们需要实现/生成复杂的自定义着色器/边框网格以支持圆角。此外，在片段着色器中支持通过剪切平面隐藏的内容溢出效果很好，但不能很好地扩展，因为父元素及其所有子元素都需要单独在其着色器中执行剪裁。有趣的优化可能是在创建时更改元素的基本网格定

义以匹配剪切设置。

其次，二维和三维的字体渲染接近不同。没有直接访问浏览器的字体渲染引擎（也就是 .ttf, .woff 和其他网页标准字体格式）



图 1：全息图案例研究

A-Frame 支持基于位图的字体，但也支持更先进的（多通道）带符号距离场（MSDF）字体渲染方法[14]。在实践中，A-Frame MSDF 实现和 CSS 字体约定之间的差异非常大，很难在两者之间找到一致/可靠的映射。我们的实施在某些情况下运行良好，但在其他情况下表现出很大的差异。这需要根据每种字体/每种情况进行微调，以便在 A 帧中获得浏览器文本呈现行为的像素完美副本。

最后，直接使用 WebGL 后端意味着 DOM2AFrame 的性能更可能是 GPU 绑定的（与 CPU 绑定相反）。特别是在低端/移动硬件上，这可能意味着我们的方法比 CPU 更敏感的方法变慢。此外，使用复杂的片段着色器意味着厌倦了填充率限制（即每帧可以绘制多少个像素）。例如，如果元素 1 在元素 2 和元素 3 之前完全绘制并着色，则在计算最终不可见的像素时浪费了时间。尽管大多数 GPU 和渲染逻辑已经考虑到了这一点（例如，首先通过深度排序元素），但是在我们的测试中，不渲染不可见的 DOM 元素对性能有相当大的帮助（例如，第 4 号元素背景颜色/边界，并且仅用作其他内容元素的定位/尺寸容器）。

4. 案例研究

为了验证 DOM2AFrame，我们实施了两个定制案例研究。我们注意集成相关工作（例如，表单输入，视频，动画）所支持的 HTML / CSS 功能，以强调我们方法的优势。

4.1 全息图

第一个案例研究主要目的在于说明 DOM2AFrame 可用于实现用例 1（UC1）：在 3D 体验中呈现 2D 用户界面（UI）或其他 2D 元素。演示模拟一个未来派设置中的大型机用户控制台，

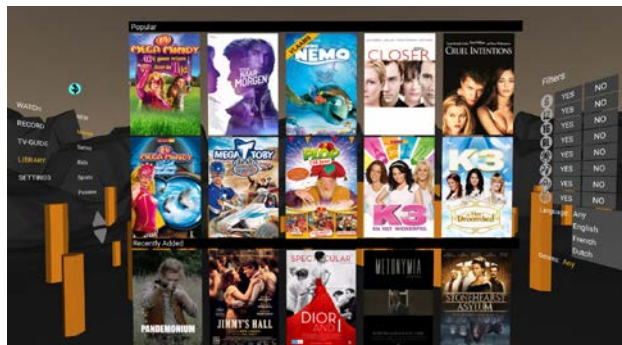


图 2: VodLib 案例研究 (所有电影图像版权属于其各自所有者)。

它包括一个右手边的“手风琴”菜单（非活动菜单项随着 CSS 过渡滑出，因为活动项占用所有可用空间，使用 CSS 溢出实现），视频显示可以在平面视频和叠加在中间的立体视频视频，以及可用于在 3D “全息图”（顶部）上选择动画的形式。控制台命令可以在底部执行，导致它们被添加到右侧的第三个手风琴窗格中。该演示在非 VR 视图也具有完整功能：3D 全息图模型呈现为平面<canvas>，仅播放非立体平面视频。这样，它也是 UC2 的验证。

4.2 视频点播库

我们的第二个案例研究旨在表明 DOM2AFrame 可用于实现用例 2 (UC2): 在 VR 中查看时, 以不同的方式呈现网页的部分内容。该演示显示了一个“视频点播库”(VodLib), 类似于 Netflix 这样的网站, 用户可以浏览视频内容并浏览并选择合适的(立体/ 360°)电影观看。右侧的过滤器选项(通常是普通网站上的复选框)会自动显示为 VR 中的按钮, 以方便用户交互。类似地, 通常水平的下拉菜单被替换为两个相邻的垂直列表(左)。两个侧面菜单也围绕垂直轴旋转, 以获得更好的 3D 世界存在感。请注意, 此演示与 Hugo Hedelin [15] 的补充相同, 该演示还实现了基于 VR 的 VodLib, 但着重于用户交互方面并直接在 A-Frame 中实现 2D UI。

通过第三个演示，我们尝试使用 DOM2AFrame 作为用例 0 (UC0)：按原样呈现完整的网页。我们的实现能够正确渲染复杂的现有 VodLib 网页的大部分，但无法处理一些复杂的 CSS 概念，如“伪元素/类”（例如：hover，::before）和复杂背景图像设置。目前，在 JS 内部操纵或解释这些 CSS 规则通常是困难的或不可能的。我们预计这种情况将来会随着更强大的 CSS API 而发生变化[25]。

致 谢

首先感谢我的指导老师翁仲铭老师，自从我加入实验室以来，翁老师给了我不少的指导和帮助，带领我去实训，并且定期讨论我的论文进度，在我遇到问题的时候总是耐心的为我答疑解惑，帮助我搭建实验环境和修改完善论文。在他的引导和帮助下，我顺利的完成了本次毕业设计。翁老师对于科研的关注与投入也激励了我在以后的工作和人生道路中要时刻的关注时代的潮流，对新兴的前沿科技保持一定的敏感度和热情，并且不断的追寻。

感想天津大学软件学院的各位老师，在 4 年的本科学习生活中，给了我很多的帮助和指导，给我的专业知识打下了扎实的功力，让我在软件工程领域里掌握了一些知识。感想我的每位同学，在一起的 4 年里将会是我人生中最难忘的四年，也许，毕业之后有些人很难再见面，但是我们之间建立的友情是时时刻刻都存在的。

感想天津大学，给我提供了一个很好的学习和科研环境，在这里上的每一堂课，做的每一个项目，都使我受益匪浅。

最后，我要感想我的家人，是他们一直无私的奉献，一直鼓励我，帮助我，也是有你们的陪伴，我才能长大成为一个有能力有作为的人，谨以此文献给所有关心我和爱我的人们，谢谢大家！