

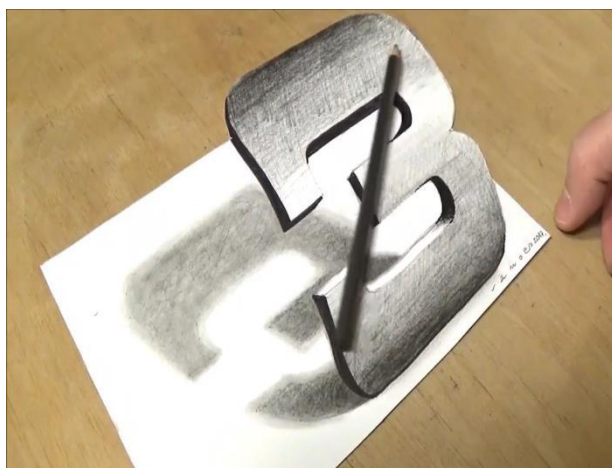
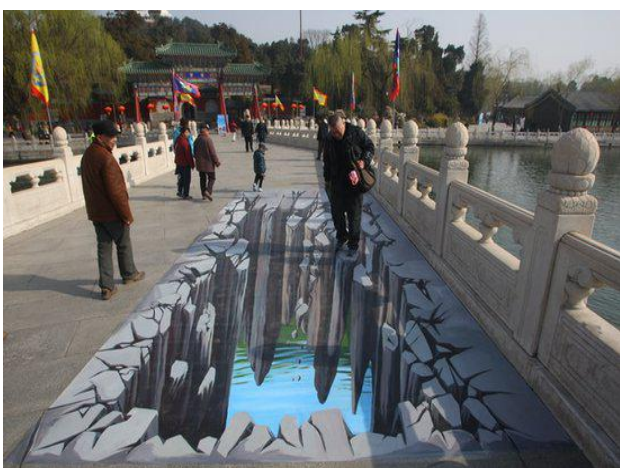
# Qwerty 3D

## 3D 立体显示解决方案

### 0 团队：Qwerty X

- 练孙鸿：华南理工大学 2015 级软件工程在读本科生，研究方向涉及各种计算机图形学(Computer Graphic)、计算机视觉(Computer Vision)的算法。有 3D 游戏、3D 扫描系统、3D 渲染引擎等项目的设计开发经验。希望能创造有趣的东西。github: <https://github.com/CHINA-JIGE>
- 常沛炜：华南理工大学 2015 级软件工程在读本科生，技术栈涉猎众多领域。
- 黄晓琳：华南理工大学 2015 级工业设计在读本科生，希望通过设计的手段表达对社会的思考，以自身绵薄的力量去改变世界。
- 陈泽琳：华南理工大学软件学院副教授，研究方向为计算机图形学与 3D 游戏引擎。

### 1 引入(Introduction)

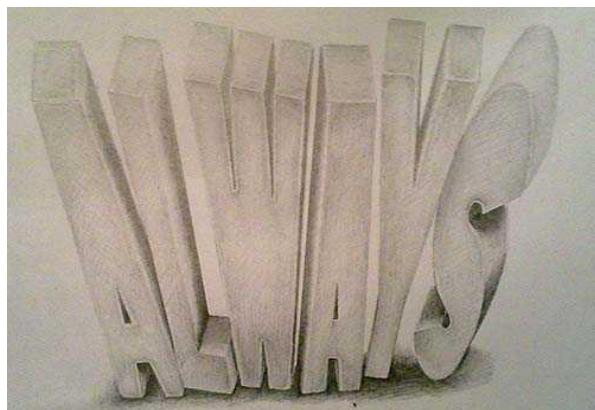


---

**3D 立体画**是一种考虑了观察者的透视投影成像之后，特殊设计出来的一种艺术作品。这种画在用一定的技巧精心设计并画出来后，从某一个角度拍摄/观察这幅画，会有一种非常立体、逼真的感觉。[1]中视频有上面立体的“3”的绘画过程。



图(a)



图(b)

上面两幅图里面，(a)是从某个特殊设计的拍摄角度上拍摄的 3d 立体素描画，“ALWAYS”几个字母像是立了起来。但如果从俯视的角度去拍摄同一张素描画，这个立体感就被打破了，或者说这个立体感表现的就不是原物体了。实际上只要从那个不是特殊设计好的角度拍的照片都无法呈现出图(a)所展现的立体感。

那么我们可以做如下的设想与推理：

既然 3d 立体画可以被画出来，那么它也必然可以被计算机渲染(Render)出来；如果可以用计算机图形学(Computer Graphic)的算法渲染出单帧的 3d 立体画，那么也必然可以进行持续地进行渲染，以使得立体感一直保持；因为 3D 画呈现的立体感与观察者位置密切相关，而为了使得立体感可以一直保持，我们就需要持续监测肉眼(观察者)的空间位置。

基于上面的考虑，我们团队决定为了实现这样效果，设计了一个特殊的、基于一种自研的透视校正渲染算法(注意与传统渲染管线的“透视校正插值”不一样)、基于即时定位与地图构建(SLAM, Simultaneous Localization and Mapping)、基于 Azure 云服务器的 3D 显示系统——Qwerty 3D。

## 2 项目技术概览(Project Technical Preliminary Summary)

### 2.1 系统的整体设计(General Design)

Qwerty 3D 是由团队独立设计的一种新型的 3D 立体显示解决方案。这可以说是一种全新的尝试。这套系统能让使用者有一种立体的、逼真的、独特的视觉体验。注意 Qwerty 3D 的显示设备与人的头部是分离的，且显示屏幕是在世界空间里是固定的（与此相对的就是 VR 头盔，显示屏幕在世界空间里是移动的）。整个系统包含多个主要模块：

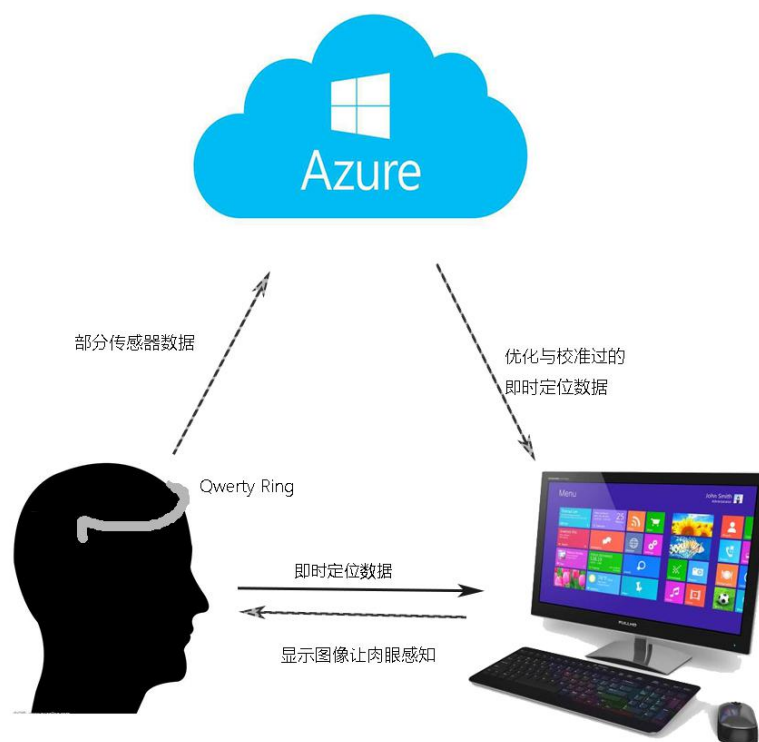
- 即时定位与地图构建(Simultaneous Localization And Mapping)数据采集装置:Qwerty Ring
  - Microsoft Azure 的服务器端
  - PC 端的实时渲染(Real-time Rendering)程序
-

---

## ■ \*交互与反馈设备（可选）

在现阶段的设计下，**产品最终形态**应是：用户需要佩戴一个经过设计的无镜片、安装少量传感器、**用于定位的轻便头戴式设备（类似于眼镜）**。此设备将采集所有需要用于**定位人眼相对于的屏幕位置**的所有数据后，把数据通过 Wifi 传输至 PC 端以及 Microsoft Azure 服务器。PC 端程序会实时接收传感器的数据并使用。而 Azure 服务器将接受更大吞吐量的数据，进行更复杂和大计算量的优化计算，并把优化和校准过的即时定位数据连同数据时间戳一起传回至 PC，以校正传感器数据噪音带来的系统状态漂移（积累误差）。PC 端上运行着一个实时渲染程序，其中的程序会使用一个由团队设计且尚未发表论文的渲染算法。渲染程序根据 Azure 服务器处理后并**实时传回**的参数**实时渲染 3D 虚拟场景/模型**。渲染出来的图像可以在任意一个固定的彩色屏幕上显示出来，并被用户肉眼看到。用户肉眼看到的屏幕上的图像应是有立体感的、逼真的。用户可以用特别设计的交互设备/反馈设备与计算机进行交互。

可以感受到这个系统的实现的工程难度比较高，因为涉及到计算机图形学(软件)、计算机视觉(软件)、计算机网络(软件)、嵌入式(硬件)、工业设计(设计)等领域。



图：Qwerty 3D 的产品形态与运作方式

## 2.2 即时定位(Simultaneous Localization)数据采集装置:Qwerty Ring

在引入里面提到，3d 立体画要有立体感就必须要从某个特殊角度去观察。所以为了可以正确、持续地在屏幕上渲染 3d 立体帧，我们必须持续地、实时地监测人眼与屏幕之间的相对位置、角度关系。而为了实现**绝对姿态、位置的检测功能**，我们为本系统设计一个轻便的头戴式设备：Qwerty Ring。此设备可以得到即时定位需要用到的传感器数据。在现阶段的设计下，此设备的传感器和模块有：**MCU (Micro Controller Unit, 微控制单元)**、**三轴陀螺仪**、**三轴地磁计**、**三轴加速度计**、**摄像头**、**网络传输模块**。其中摄像头是给基于计算机视觉(CV, Computer Vision)方法的即时定位模块采集数据的。而即时定位（可以看作是 SLAM 的简化版）也是需要结合九轴传感器的数据来实现姿态的融合估计的。所有的必要的传感器数据在硬件上

---



采集完毕以后，就会通过 WIFI 发送给 PC 端以及 Azure 服务器端进行下一步的处理（所以设备的附近需要有 WIFI 发射器，这一定程度上限制了使用范围）。Qwerty Ring 实时发送传感器数据给 PC 端，但是传感器数据可能会有积累误差。但是可以而 Azure 服务器端就可以有更多的时间和计算资源去做优化（为此，Azure 的优化信息不需要实时回传至 PC，只需要隔一定时间再回传优化后的数据给 PC 端）。

限于时间、成本、团队等因素影响，本阶段将用智能手机开发数据采集 App 的 Demo 来在一定程度上快速验证系统设计的正确性与效果，因为一般的手机集成了上述的所有传感器，并拥有网络传输功能。



图:Qwerty Ring 概念图

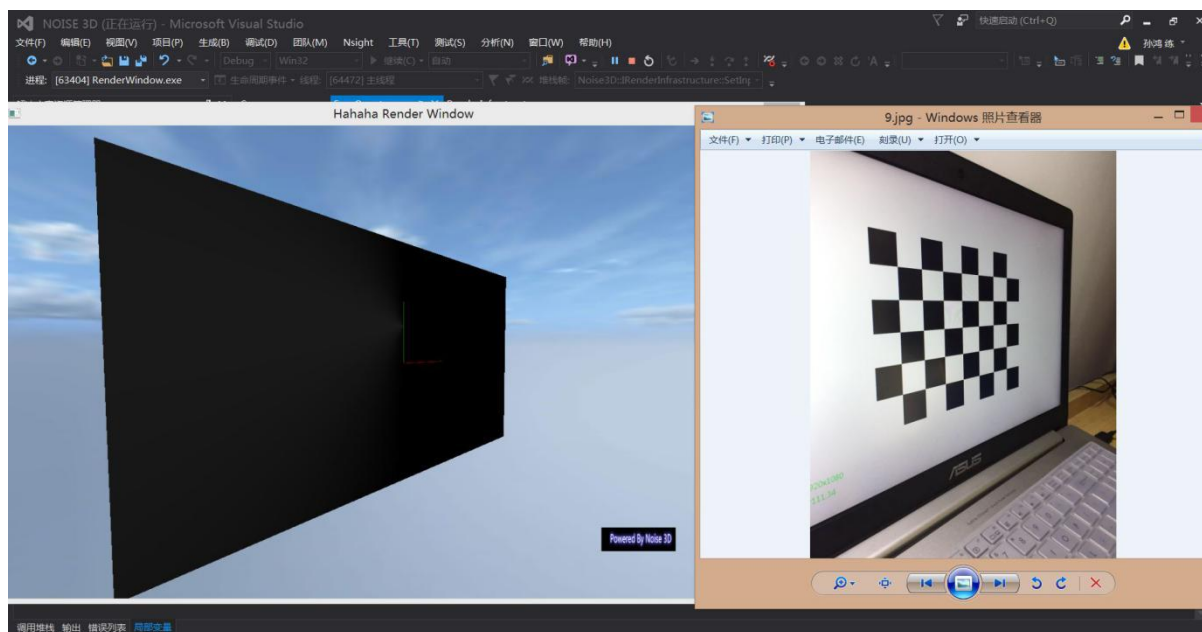


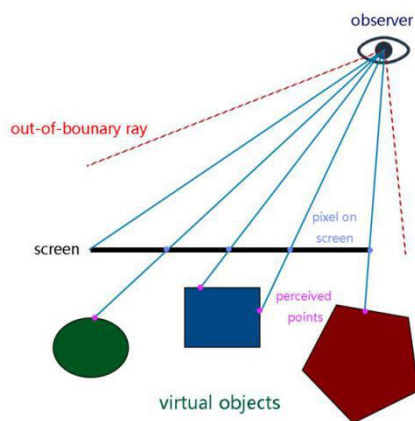
图:基于计算机视觉方法的摄像机初始姿态估计，Qwerty Ring 就是用来采集基于计算机视觉的定位算法所需数据的设备

## 2.3 Microsoft Azure 服务器端

本项目 Azure 的云服务不止一种用途（2.6 里将更详细地讲），但在实现核心功能的技术路线上，Azure

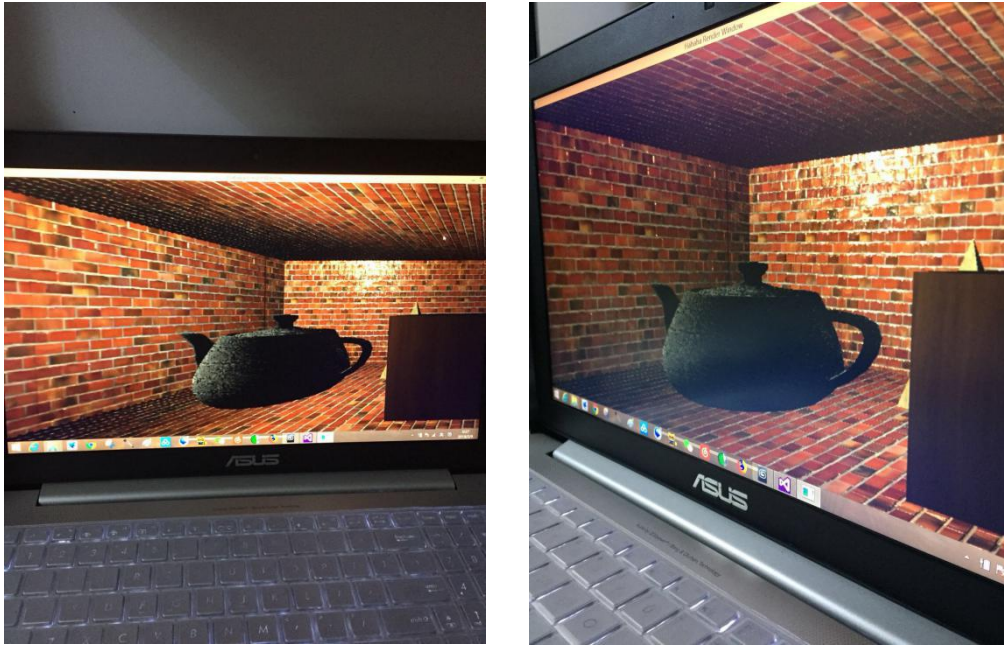
承担了基于传感器数据与摄像头图像数据的即时定位计算的优化(optimization)任务。拟采用解决方案是参考 PTAM / ORB-SLAM / Mono-SLAM/ VINS 等 SLAM 类的计算机视觉方法，经过精简以后做出适合本系统的修改和改进。在 Azure 上的服务器端会不断地接受传感器数据，并计算出肉眼相对于屏幕的相对位置与姿态。上面提到的 SLAM 系统的特征检测几乎都是用 SIFT、ORB 等人工特征，而在更后续的迭代开发中，团队会逐渐实现基于机器学习的计算机视觉算法（例如 CNN 系列的物体检测方法）。

## 2.4 PC 端的实时渲染(Real-time Rendering)程序

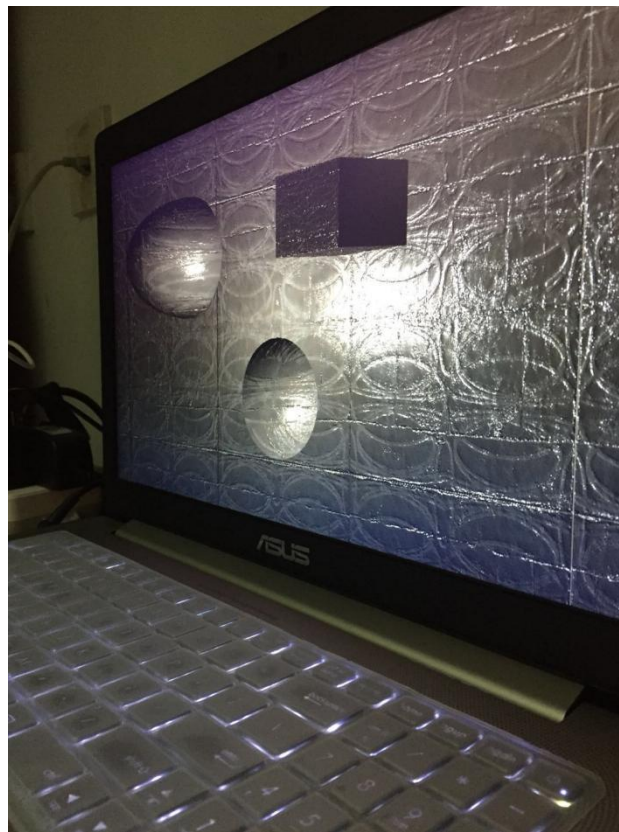


图：渲染算法的原理图

在 PC 端上运行着一个全屏的实时渲染程序，本项目的渲染底层实现基于 Direct3D 11 API。这个程序不断地接收从 Azure 服务器回传的已经过处理的人眼位置、姿态数据，并以这个数据为参数进行渲染。其中屏幕“后面”的“物体”是**计算机三维模型**，可以用专业的建模软件(如 3dsmax, maya)等制作。对于每一帧而言，渲染 3D 立体画不可以只对顶点施行普通的 World-View-Projection 变换并直接呈现在屏幕上。要渲染出 3D 立体画来欺骗人眼，则需要经过用于校正透视效应的图像重采样步骤，方可把最终图像呈现在显示屏上。这种方法比正常的实时渲染多一个 shader pass 来做后处理(post-processing)，但传入 GPU 的顶点数据与处理方式又跟普通的 post-processing 不一样。右图是这种校正透视效应的图像扭曲算法的示意图，用这个算法可以根据摄像机的位置与朝向、屏幕的几何描述来渲染单帧的 3D 立体画。此算法仍未发表论文。



图：左右为同一渲染图像在不同角度观察的效果。右图在预设好的位置上观察，就能观察到更合理的场景



图：半球看上去就像突出了屏幕一样，富有立体感（但由于没有添加更真实的光影效果，导致这张效果图的效果一般）。如果屏幕渲染的图像可以根据观察者位置不断更新，那么这种立体感就可以一直保持并欺骗人眼

## 2.5 \*交互与反馈设备

Qwerty 3D 主要是为了用户的视觉体验而服务的，但为了提升用户体验，系统可以配备人机交互设备。目前 (2018. 3) 尚未对这种人机交互设备进行设计，因为项目当前最重要的任务是解决用户视觉体验问题。

---

## 2.6 项目使用的 Microsoft 的技术

### 2.6.1 已使用的技术

Microsoft Visual Studio: c++程序的编写与调试。

Microsoft Azure: 服务器; 云计算。

Microsoft DirectX: 底层编程 API, 渲染算法的实现基于 D3D11.0 与 Shader Model 5.0(HLSL)。

### 2.6.2 Microsoft Azure 服务的使用

实时计算应用场景。Azure 在本系统里面,是处于数据采集端与图像输出端之间的重要位置。Qwerty Ring 采集好传感器数据以后把部分传感器数据通过 Wifi 传输到 Azure 服务器上(注意:考虑到网络带宽和传输速率的问题,这里不需要是实时传输)。紧接着, Azure 云服务器就可以对这些数据进行分析与计算,承担了基于传感器数据与摄像头图像数据的即时定位计算的优化任务。**Azure 可以分担即时定位(SLAM)的繁重的机器视觉计算负担,从而相对于 PC 端异步地进行全局优化**(例如可以周围环境的三维点云也构建出来,每次有新的点云生成就可以和历史数据进行对比,进行 Bundle Adjustment, 回环检测)。Azure 就能校正传感器的大部分积累误差,每隔一小段时间就可以把优化结果回传至 PC 端上。

往后的迭代周期可能会考虑在 Azure 上部署计算量更大的基于卷积神经网络(CNN)相关机器学习算法来实现即时定位计算,因为现在的 SLAM 系统都是检测 SIFT、ORB 等人工特征,而理论上基于机器学习的算法可能可以自动学习更好更准确的特征,从而使得定位更加精准。面因为 PC 端的渲染算法只需要估算的人眼的相对位置、姿态作为输入,所以数据从 Azure 传回 PC 端的网络时延会相对较小;但由于图片的传输时延会比较大,所以需要有相应的方案来弥补这个短板,例如 PC 端进行空间轨迹的插值与预测, Azure 端充分利用异步数据传输等待的时间进行计算,提高即时定位的精确度。

Azure 还有一些离线计算与储存的应用场景:

- 作为 3D 模型、场景数据的的储存服务器,并提供一个网站让用户挑选场景并下载。
- 作为场景离线渲染的云服务器(渲染农场),用光线跟踪渲染器烘焙贴图并增进静态场景的画质。

### 2.6.3 其他可能会使用的技术

**Hololens/Kinect** 本项目将不会直接使用 Hololens 或者 Kinect 的成品,但这两种产品背后的技术——基于深度摄像头与视觉的即时定位技术,却是非常的关键。微软的解决方案可能比我们成熟,但由于商业与知识产权的问题,本项目未能使用此项技术改进人眼即时定位与地图构建的效果。

## 3 项目描述

### 3.1 相关产品与技术

从特点或所用的部分技术上来看,我们团队决定将 Qwerty 3D 归类为 3D 显示技术,所以类似的产品和技术有:

**Microsoft Hololens(AR/MR)** Hololens 是一个头戴式设备,设备通过即时对周围的环境进行三维重建,并在设备上的半透明屏幕上渲染图像以叠加在真实图像上。但[2]网上资料显示, Hololens 的屏幕比较小,也意味着视场角小,用户可以看见的区域小。而且 Hololens 价格昂贵,距离平民化还有一定的距离。

---





图：Microsoft Hololens

**Magic Leap(光场显示技术)** Magic Leap 设计上也应该是一种头戴式设备，设备不使用普通意义上的屏幕，而是把四维“光场”(light field)投影到人的视网膜上，使得人眼甚至可以**自主对焦**——这是其他设备从原理上讲根本不可能实现的特性。但是，[4][5]表示 Magic Leap **开发进度远远落后与目标**，甚至工作室放出的令人惊叹的特效鲸鱼飞跃 demo 也是由动画特效工作室做的概念视频，并不是产品本身的效果。光场显示技术本身不是骗局，Magic Leap 要真正做出产品级的光场显示，就要把设备小型化，然而目前 Magic Leap 的设备仍非常巨大，显然短期内无法变成工业级产品。

**Oculus Rift(VR)** Oculus Rift 是一种虚拟现实设备[3][6]，此设备有一个主体头戴式显示器，分辨 640x800x2（双屏），有一条线需要连着 PC（这就限制了用户移动范围）。但是据各种用户体验评测[3]来说，尽管有一些诸如较低分辨率、微量头部跟踪时延的小缺点，很多 demo 都能给人一种眼前一亮、沉浸的感觉。Oculus Rift 在不计高性能电脑的售价为 300+美元。



图：Oculus Rift

**全息投影** [7][8]一般的全息投影的思路是通过光的干涉和衍射原理，来记录并再现物体的三维信息。亦即在拍摄过程中利用干涉原理记录物体光波信息，成像过程中利用衍射原理再现物体光波信息。但根据这样的原理，全息投影的成像阶段就需要用特殊的成像设备与技巧；而且需要一种特制的光栅全息显示屏才可以利用衍射来重现光波信息。而人类想象的全息投影最终形态——漂浮在空中的真正全息投影——但依旧因为成本高昂、像素少而停留在实验室内。全息显示技术(holographic display)已有了一定的应用，例如常见的透明四棱锥的全息显示设备。

---





图：一种常见的全息显示设备

**3D 电影** 这种技术已经是非常久远的了，应用广泛，现在大部分电影都是 3D 版的，用户需要佩戴一种没有任何电子元件的 3D 眼镜，利用双目视差来营造立体感。但实际效果比较一般，因为 3D 电影的立体效果还取决于拍摄的内容，如果电影只是把 3D 当作噱头而随便拍的话，用户在电影院里也感受不到多少立体感。

**裸眼 3D 显示屏** 这类 3D 显示技术一般不需要外部设备进行辅助即可营造出立体感，一般又称为“自由立体显示”[9]。根据百度百科[9]的说法，基于液晶屏的自由立体显示技术主要有如下几种：视差照明技术、视差屏障技术、微柱透镜投射技术、微数字镜面投射技术等。

更多 3D 显示的解决方案与思路可以参考[10]。

### 3.2 创新性与使用场景

Qwerty 3D 是一种低成本的 3D 立体显示解决方案。使用者需要有一个可以实时接收 Azure 数据并实时渲染的 PC 电脑，一个普通显示设备，Qwerty Ring。

首先从 Qwerty 3D 的产品形态是比较新颖的。相比于 3.1 中提到的各种解决方案，显示屏幕是与 Qwerty 3D 的头戴式设备 Qwerty Ring 分离的，所以头戴式设备将会非常轻便，制造成本低。而且屏幕只需要是具有实时显示功能的“平面”（电脑屏幕、投影屏幕等）即可，那么用户在显示器的选择上就有了很大自由度，且成本相对低廉。

Qwerty 3D 的用户体验也是比较新奇的。如果用一个电脑屏幕作为显示屏的话，用户可以把这个屏幕当作是一扇“玻璃窗”，在玻璃窗的这一面可以随意观察窗里面的虚拟世界。甚至可以在平面屏幕上模拟出“凸起”的感觉。

Qwerty 3D 的 PC 端渲染算法的创新。这个解决方案的最初灵感来自于 3D 立体画，于是队长练孙鸿自行研发一种简单又易用的 3D 立体画渲染算法，目前正在撰写论文。

Qwerty 3D 作为一种 3D 立体显示解决方案，使用场景包括但不限于：游戏等交互式应用；计算机模型的立体感视觉表现/艺术表现；计算机辅助设计（CAD）；

### 3.3 市场与商业模式

Qwerty 3D 是一套软硬件相互配合的系统/平台，则盈利点主要是在销售 Qwerty Ring 和就基于 Qwerty 3D 的内容（如游戏）。但是基于某个特殊平台的内容生产是与平台本身(Qwerty 3D)密切相关的，所以在产

---

---

品生态建设上做出很大的努力。例如 Sony 的 Play Station 4，这种游戏主机平台的盈利能力就跟硬件、软件、内容等整个生态有密切关系。

因为 PC 端渲染程序需要做出一点额外的修改，所以正常的交互程序并不能实现 Qwerty 3D 的设计要求。所以必须要有相应游戏引擎支持 Qwerty 3D，例如可以开发 Unity 的插件。此算法的 c++实现将会开源以提供更好的开发者生态。寻求与 steam 之类的游戏平台合作，构建类似“steam VR”之类的配套应用生态。而这个时候就需要招募更多的人员来开发基于 Qwerty 3D 的交互式应用 demo 来给广大开发者一点思路。但其实无论如何，最重要的还是要把成品级的 Qwerty Ring 做出来，因为软件部分开放一定的自由度给开发者自行开发（提供算法、开源库、SDK），而配套的硬件的制作难度就会高非常多，所以 Qwerty Ring 需要达到消费级的水平，然后进行零售。

综上，目标客户是各种对 3D 立体显示有需求的个人与组织，如游戏玩家、设计师等。

### 3.4 团队

练孙鸿(队长)：软件工程师，核心渲染算法设计、实现，系统架构设计、计算机视觉算法实现。

常沛炜(队员)：软件工程师，计算机视觉算法实现、Azure 云服务架构与部署

黄晓琳(队员)：工业设计师，Qwerty Ring 头戴式设备的外观设计。

陈泽琳(导师)：指导团队，与团队讨论并解决部分技术问题。

### 3.5 未来

**待解决的问题。**这里说的只说产品核心功能上需要解决的问题：双目视差效应的抵消；系统响应延迟；算法的移植与嵌入式硬件开发。

**团队的扩张。**以目前的团队构成来说，离真正做出 Qwerty 3D 的实物原型还有一定距离。首先从这个系统的设计上来说，其涉及的技术与工程的难点就比较多，需要的人才技能覆盖到计算机图形学、计算机视觉、网络开发、移动开发、工业设计、硬件设计与制造(电子电路)、嵌入式设计与开发等。若只考虑核心产品的研发及产品开发周期的话，可以得出一个明显结论：人员不足。必须扩张团队才能高效地做出产品原型并加以验证。参加这次比赛的三人小团队是一个技术团队，所以商业运营、管理团队需要在未来逐渐形成。

**系统原型的开发。**Qwerty 3D 从理论上可以让人眼从一个平面的屏幕上感受到立体感。但可以预料到的是，由于双目效应、人眼位置定位等可能存在的问题，Qwerty 3D 系统的用户体验不一定能做得令人满意。这需要通过开发出系统原型来验证效果，并根据存在的问题来设计解决方案。综上，现阶段(2018.3)最大的目标就是开发出能直接让真人体验的原型系统，并根据原型确定此系统的设计是否具有更大的潜力。

## 4 引用

[1]bilibili - <https://www.bilibili.com/video/av17092085/>

[2]知乎 - <https://www.zhihu.com/question/30011719>

[3]知乎 - <https://www.zhihu.com/question/21280028/answer/17747576>

[4]知乎 - <https://www.zhihu.com/question/53462751/answer/135141637>

[5] <http://tech.sina.com.cn/it/2016-12-09/doc-ifxyipt0672613.shtml>

[6] <https://baike.baidu.com/item/Oculus%20Rift/352914?fr=aladdin>

[7] <http://www.tmtpost.com/2878690.html>

[8] <https://baike.baidu.com/item/3d%E5%85%A8%E6%81%AF%E6%8A%95%E5%BD%B1/9548111>

---

---

[9]<https://baike.baidu.com/item/%E8%87%AA%E7%94%B1%E7%AB%8B%E4%BD%93%E6%98%BE%E7%A4%BA/1426305>

[10]<https://wenku.baidu.com/view/6cd0cab1dc88d0d233d4b14e852458fb770b3830.html>

---