

# 基于 WebGL 的全景技术智慧旅游系统<sup>①</sup>



罗雅丹, 罗琪斯

(广西壮族自治区地图院, 南宁 530023)

通讯作者: 罗雅丹, E-mail: lydhxx@qq.com

**摘要:** 针对目前国内的全景漫游技术兼容性和实现效率差等问题, 提出了一种基于 WebGL 的全景技术, 免除安装渲染插件的步骤, 系统更轻量, 计算机资源占用率低. 采取无人机航拍与地面全景云台架设相机拍摄相结合的方式收集数据; 按照尺度不变特征转换的方式完成图片的拼接; 结合 WebGL 技术和具体的 HTML5 框架添加场景和制作热点数据, 上传至服务器端完成发布. 该技术成功应用于桂平市智慧旅游建设项目当中, 实现了景区的全景虚拟漫游系统. 通过该系统, 用户可在线虚拟游览欣赏景区的景点风光, 体验全景漫游的沉浸感和逼真性.

**关键词:** 全景; WebGL; 智慧旅游; 虚拟漫游

引用格式: 罗雅丹, 罗琪斯. 基于 WebGL 的全景技术智慧旅游系统. 计算机系统应用, 2020, 29(1): 86-92. <http://www.c-s-a.org.cn/1003-3254/7179.html>

## Smart Tourism System of Panoramic Technology Based on WebGL

LUO Ya-Dan, LUO Qi-Si

(Guangxi Institute of Mapping, Nanning 530023, China)

**Abstract:** In view of the current domestic panoramic roaming technology compatibility and poor implementation efficiency, this study proposes a panoramic technology based on WebGL, which eliminates the installation of rendering plug-ins. The system is more lightweight and has lower computer resource usage. Firstly, data is collected by combining the aerial photography by unmanned aerial vehicle and ground point shooting. Then, image splicing process is performed with scale-invariant feature transform. Finally, WebGL technology is adopted to combine with specific HTML5 framework to add scenes and make hot data, which are uploaded to the server to complete the publication. It is successfully applying to Guiping smart tourism construction project. Through this system, users can enjoy the scenery of scenic spots on the online virtual tour, and experience the immersion of panoramic roaming.

**Key words:** panoramic; WebGL; smart tourism; virtual roaming

全景虚拟漫游技术是当前全球范围内流行的一种“基于图像”的虚拟现实体验视觉技术<sup>[1]</sup>. 360°全景图像是通过相机拍摄采集场景的图像数据, 后期经过图片拼接合成全景图, 在全景播放平台进行展示. 该技术的全景浏览提供了 360°环视视角, 可以控制内容的旋转方向或缩放其大小, 交互性好, 沉浸感较强. 智慧旅游

基于新一代信息与通信技术, 通过互联网为游客提供高品质、高满意度服务, 借助 PC 及平板、智能手机等移动终端, 使游客能够主动感知旅游资源、旅游经济、旅游活动等方面的信息<sup>[2]</sup>. 而全景漫游技术则为智慧旅游提供了重要的展示平台, 通过在便携式移动终端设备中提供 360°沉浸式全景虚拟漫游体验, 使游客

① 基金项目: 桂平市全域旅游智慧化建设项目 (GGZC2017-G3-9837-XPZB-177)

Foundation item: Smart Tourism Construction Project of Guiping City (GGZC2017-G3-9837-XPZB-177)

收稿时间: 2019-05-05; 修改时间: 2019-05-23; 采用时间: 2019-05-30; csa 在线出版时间: 2019-12-27

在旅游前、旅游中、旅游后能够主动感知景区风景,提升游客对景区风景的感知度,使得旅游产品的宣传变得更加真实,景区的知名度也能得到大大的提高,在旅游服务业中有着广泛的应用<sup>[3]</sup>.

传统的全景展示系统是通过对整个场景进行三维建模来表达,对硬件设备要求高,制作周期长,开发工作量较大<sup>[4]</sup>.Flash播放插件是全球电脑中安装数量最多的插件,因此用户无需再安装额外的插件便可播放,作为一个动画软件,Flash具有相当大的编辑潜力,可以为三维全景实现浏览查看功能.由Garden Gnome Software公司研发的Pano2VR,能够将全景图片转成QuickTime或者Flash格式,要求浏览者在电脑中安装QuickTime或者Flash播放插件.目前国内外主流全景虚拟漫游技术是基于WebGL和HTML进行开发的,有学者使用WebGL在3D画面中展示人体的运动<sup>[5]</sup>,也有的使用WebGL和HTML实现了基于网络的三维地形可视化<sup>[6]</sup>,还有学者研究基于WebGL技术构建虚拟校园的原型漫游系统<sup>[7]</sup>.这些方法是利用了Three.js的开源3D引擎,在浏览器上增加场景、物质、渲染器、灯光、模型等模型,由于目前的标准和方法尚未统一,存在诸如开放性、兼容性,显示效果以及效率不佳等问题<sup>[8]</sup>.本文提出了一种基于WebGL的轻便式全景虚拟漫游技术,通过HTML5和WebGL结合的方式,脱离中间插件的支持,适用于网络在线浏览,可以通过电脑或移动终端设备进行交互操作.该技术的数据量较小,对客户端硬件要求不高,功能丰富,系统的独立性更强,计算机资源占用率低.

## 1 智慧旅游系统架构

基于全景虚拟漫游的智慧旅游系统采用四层架构模式,包括支持层、数据层、服务层和应用层,系统架构如图1所示.支持层由云服务器、网络 and 数据库管理系统组成;数据层用于存储全景、旅游专题等原始数据;服务层包括全景、要素和网络分析等服务;应用层主要为移动终端和Web提供访问的接口,内容包括景区全景、语音解说、景区介绍和旅游资讯等;用户向应用层提出应用需求,应用层将用户数据访问数据提交到服务层,服务层则根据用户不同的需求向数据层请求不同的数据,并将结果返回应用层.

## 2 WebGL 技术原理

本文设计的系统旨在为游客和旅游景区提供一种

轻便式的全景虚拟漫游系统,前期的图像信息数据采集结合地面全景云台和无人机航拍两种方式,以全方位视角展现全景内容;全景拼接则是经过提取图像特征、特征点匹配和图像拼接完成;场景的生成和发布是系统的核心部分,建立在基本HTML结构之上,在HTML的主体中编写主程序<sup>[9]</sup>.Three.js作为原生Web3D引擎<sup>[10]</sup>,是当前应用最广泛的WebGL框架,本文选用Three.js来开发系统,实现了基于网络浏览器的无插件三维场景游览,网页包括支撑层、三维应用层和用户层.

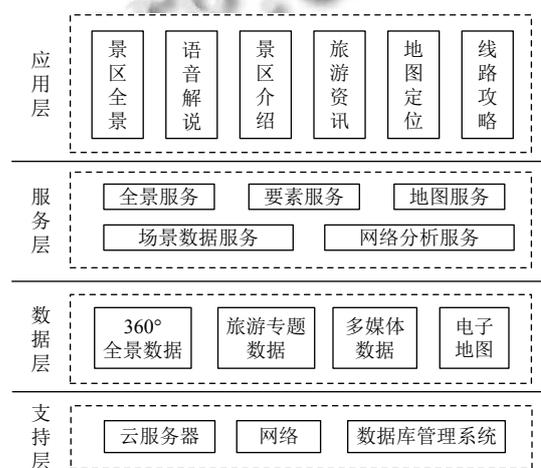


图1 智慧旅游系统架构

### 2.1 支撑层

支撑层包括了HTML网页基本框架和WebGL三维场景画布.

#### (1) HTML 网页基本框架.

网页文件的基本框架,涵盖了程序、链接、音乐和图片等元素.

#### (2) WebGL 三维场景画布.

采取canvas标签实现一个可以在HTML中画图的情景.

### 2.2 三维应用层

本层使用Three.js和JavaScript在支撑层完成三维场景的搭建、数据加载和循环渲染.

(1) 三维场景的搭建.具体的部件包含场景、相机以及渲染器.

① 场景:类似于一个装置物体的容器,将必须要表现的全部元素放到这个容器中.用THREE.Scene来构建一个新的场景,代码如下:

```
var scene = new THREE.Scene();
```

② 相机: 相机是模拟了人的眼睛, 根据所处的位置和朝向, 显示不同的物体. 在 WebGL 中, 提供了透视投影和正投影两种相机<sup>[11]</sup>. 采取透视相机 (THREE.PerspectiveCamera) 模拟一个相机, 代码如下:

```
var camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 1000);
```

③ 渲染器: 将空间三维物体转换到二维平面的软件被称为渲染器. 渲染工作采取 Three.js 库中成效好的 WebGLRenderer 完成. 代码如下:

```
var renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
```

(2) 数据加载. 为了保证网络传输及网页显示效率, 将全景数据转成 JSON 格式. 当用户所在的浏览器向服务器发送请求, 即可下载场景数据到浏览器, 本地解析场景模型数据, 将场景数据显示在浏览器页面上.

(3) 渲染循环. 用户在使用全景虚拟漫游系统切换视野时, 系统需要不断地调用渲染器对画面进行渲染, 从而可以在浏览器中显示出连贯的场景. 渲染循环代码如下:

```
function render() {
    cube.rotation.x += 0.1;
    cube.rotation.y += 0.1;
    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
```

### 2.3 用户层

本层重点完成全景虚拟漫游系统的用户交互. 采取 CSS 搭建整个全景网页的布局, 内容涵盖了网页的色彩、属性和元素等; 通过 JavaScript 语言实现鼠标和键盘的交互; 由 CSS 中的 2D/3D 转换属性实现页面内容的缩放和平移等操作.

## 3 基于 WebGL 的全景虚拟漫游技术

全景虚拟漫游技术的整体设计分为数据采集、图像拼接、场景生成、场景发布 4 个部分, 如图 2 所示: 确定旅游景区所需拍摄的景点和拍摄环境, 通过无人机航摄和地面点全景云台拍摄两种方式进行数据采集; 利用 PTGui 软件进行图片拼接和 Photoshop 完成图像美化修饰工作; 通过 WebGL 技术和 HTML5 技术的结合构建全景场景和数据发布, 实现全景漫游、自由浏览交互等多种形象生动的视觉体验.

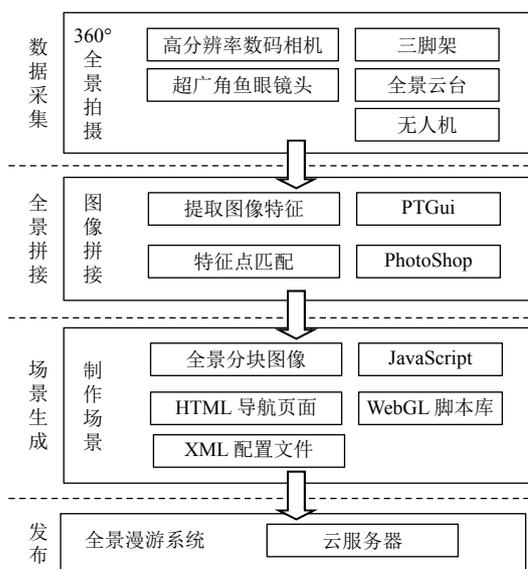


图 2 全景虚拟漫游技术整体设计

### 3.1 数据采集

考虑到应用场景的多样性, 数据采集采用无人机航拍与地面全景云台架设相机拍摄相结合的方式获得, 以便构建空中与地面两种全景浏览视角.

采集数据需要用到的设备: 高分辨率的数码相机、超广角的鱼镜头、三脚架、全景云台和无人机. 拍摄时相机应处于同一高度和水平面, 同时确保中心点在云台或无人机转动轴心的垂线上, 防止出现视差和偏斜的情况<sup>[12]</sup>. 为使后续的图像拼接能够顺利进行, 减少图像误差, 通常拍摄两个相邻图像须保持 30% 以上的重叠内容, 重叠度越高, 拼接出来的图像质量越好<sup>[13]</sup>; 而保持较高的重叠度, 则会增加初始数据的采集量. 为了解决该问题, 提高拍摄效率, 本文所有数据的采集均采用 180°超广角的鱼镜头, 在同一高度和水平面所需拍摄照片数量比普通的广角镜头少了 50% 以上. 在拍摄时手动设置镜头的曝光度和对焦, 保证拍摄的白平衡和相邻的照片之间的感光度, 进一步提高全景图像的质量.

全景数据采集流程如图 3 所示: 分别在水平方向、向下 45°方向、向上 45°方向进行该方向上每隔 90°拍摄一次的操作, 再在垂直方向的向上 90°和向下 90°分别拍摄一次, 通过以上采集可获取到一个场景的所有全景数据.

### 3.2 全景拼接

全景图的拼接, 是将前期数据采集的同一个场景的所有图片组经过拼接和处理合成一张球面全景图,

包括提取图像特征、特征点匹配和图像拼接 3 个部分.

(1) 提取图像特征: 采用尺度不变特征转换 (Scale-Invariant Feature Transform, SIFT) 的方法提取两张图片之间的特征<sup>[14]</sup>, 流程图如图 4 所示. 检测: 在高斯尺度空间中, 通过高斯差分函数找到两幅图像之间的相同的兴趣点; 过滤: 再对这些兴趣点进行拟合, 过滤掉不稳定或者低对比度的点, 找到关键点; 赋值: 根据关键点邻域像素的梯度值, 给每个点赋值特征方向; 描述: 根据关键点的方向信息, 生成特征向量.

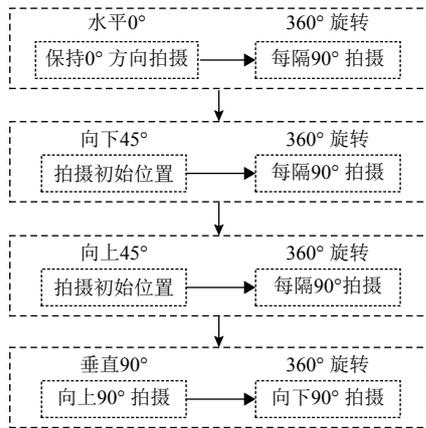


图 3 数据采集流程

(2) 特征点匹配: 得到特征点之后进行两幅图像之间的特征匹配, 会出现许多误差点. 因此, 需要采用随机抽样一致性 (RANDOM Sample Consensus, RANSAC) 算法进行最优化匹配<sup>[15]</sup>. 假设特征的分布呈直线方程分布, 随机抽取 2 个样本点, 对数据进行拟合; 假设一个容差范围, 找出拟合曲线容差范围内的点, 并统计点数; 重新随机选取 2 个点, 反复实验, 直到结束迭代; 迭

代完成后, 找出点数最多的情形, 即为最优匹配.

(3) 图像拼接: 目前常用的全景拼接软件有 PTGui、APG、Hugin 和漫游大师等软件<sup>[16]</sup>, 本文中采取的软件是 PTGui. 导入图片组, 根据图片的镜头参数, 识别重叠内容的像素特征, 自动生成控制点完成初步的图像缝合. PTGui 软件能够自动识别相邻图片中的图形转折折点、色差较大的细部节点等作为重叠部分, 并命名为①②③④⑤⑥等控制点. 通过设置与查找重叠区域控制点, 可以将相邻两张图片中的①②③④⑤⑥等控制点一一对应, 依次类推, 设置好相应控制点, 即可缝合度全景图. 每两张图连接部分不能少于 3 个点. 一般情况下 PTGui 软件自动建立的控制点就有很好的效果, 如果因某些原因软件不能自动找到控制点的话可以选择手工添加. 根据软件实际的拼接效果, 采用手动方式调整和添加控制点, 在相邻的图片之间尽可能多地添加控制点, 可以降低图片间的差异性, 提高图像匹配的精确度; 并通过优化器和曝光修正调整图像, 确认所有的拼接缝对齐后, 创建全景图像, 完成拼接工作.

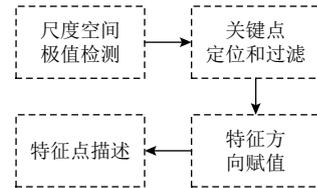


图 4 尺度不变特征转换流程

图像拼接处理后如图 5 所示. 对由于天气、拍摄环境等因素造成的相片质量不好的情况, 利用 PhotoShop 软件进行相片处理, 调节相片的色调、颜色、对比度等, 必要的时候利用素材库进行相片优化<sup>[17]</sup>.



图 5 拼接处理后的全景图像

### 3.3 构建全景场景及数据发布

单个场景的全景图像拼接处理完成之后, 需要将

多个场景的全景图关联起来, 通过场景链接和切换, 实现景区范围内的无缝全景虚拟漫游. 而常见的将漫游

系统呈现在互联网上的方式,需要通过展示图像的第三方插件进行渲染.传统的渲染插件有 VRML、X3D、Java、QuickTime、Flash 等<sup>[18]</sup>,在使用全景虚拟漫游系统之前必须在电脑或移动终端安装好这些浏览器插件,大大增加使用前的操作工序,无法做到简单轻便式的用户体验.

为了解决该问题,使用 JavaScript 语言,结合开源 WebGL 三维脚本库 Three.js<sup>[19]</sup>,在 HTML 网页基本框架中实现全景虚拟漫游系统.程序运行流程如图 6 所示,运用 Three.js 编程,初始化全景浏览需要的组件,首先在 HTML 中创建 DIV 标签,构造球体三维模型,设置球体半径为 360,水平方向和垂直方向上分段数分别为 80 和 60.获取 HTML 网页的全景展示的节点对象,并添加 WebGL 渲染模型,全景图片将作为球状模型纹理映射的材料.实例化照相机的参数,视景物垂直方向上的张角设置为 45°,容器 Canvas 的水平方向和垂直方向长度设为 1024 和 512,照相机到视景物最近、最远的距离为 1 和 1000.将渲染好的全景球体加载到场景中,即可实现全景的展示.

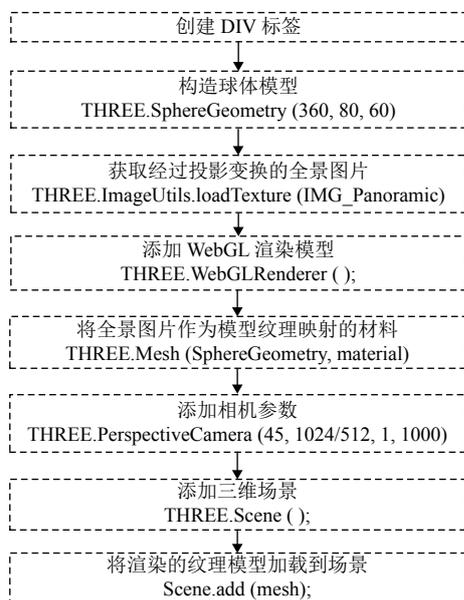


图 6 程序运行流程

本文的全景虚拟漫游系统如图 7 所示.全景漫游场景包含:链接主页、平移、缩放和语音讲解等.全景导航菜单包括:全屏、场景切换以及层叠样式表生成的场景照片墙等.全景数据文件包含全景分块图像、HTML 导航页面、XML 配置文件、界面皮肤文件等.

将打包好的全景数据文件发布到网络服务器上,即可通过浏览器访问的方式进行全景虚拟漫游体验.

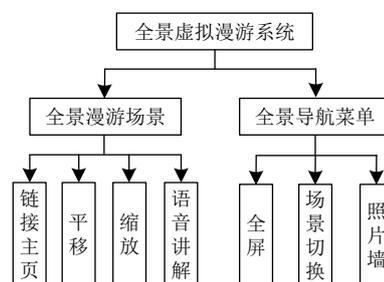


图 7 全景虚拟漫游系统

## 4 应用案例与效果分析

### 4.1 应用案例

以桂平市重要景区景点为研究对象,将基于 WebGL 的全景漫游技术应用到桂平市旅游智慧化建设项目当中,实现了桂平市景区的全景虚拟漫游系统.景区全景漫游系统截图如图 8 所示.

景区全景系统提供三大景区的全景虚拟漫游数据,包括桂平西山风景名胜区 29 个场景切换、大藤峡生态旅游区 9 个场景切换、桂平市太平天国金田起义 12 个场景切换.通过该系统,用户可以在浏览器中进行虚拟交互式全景体验,在线欣赏桂平市重要景区的景点风光,体验全景漫游的沉浸、逼真感.本文在虚拟漫游系统中加入了文字简介、语音解说、视频短片和背景音乐等内容,为用户提供更多样的视听体验.

### 4.2 效果分析

传统的全景漫游技术要求用户在体验全景系统之前在客户端安装 Flash 等第三方插件,插件通常有几十兆的大小,占用计算机的资源,增加了用户的操作工序.而目前常用的全景漫游技术软件多为商用收费软件,技术标准不统一,不支持开源开发.相比于传统的全景漫游技术和目前商用的全景漫游技术,本文利用 WebGL 技术能够直接对像素进行操作和编辑,改善像素级别的图像显示性能,具有较好的显示效果;省去了安装第三方插件的繁琐操作,对客户端硬件要求不高,系统独立性强;开放性方面,采用的 WebGL 第三方库和 HTML5 技术属于完全开放的,有利于系统后续的扩展;兼容性方面,系统各项功能模块在 IE11.0、Chrome、Firefox 等各类主流浏览器的不同版本中运行正常,具备较好的兼容性.



图8 景区全景虚拟漫游系统截图

常见的 Web3D 技术对比结果如表 1 所示. Cult3D 和 Viewpoint 必须要安装插件, 对客户端硬件要求高. Babylon.js 是一款 3D 游戏引擎, 侧重于游戏的开发应用; SceneJS 则尤其适用于需要高精度细节的模型需求; Cesium 属于地图引擎范畴, 常用于 3D、2D 和 2.5D 形式的地图展示; 本文采用基于 WebGL 的 Three.js, 与其他三维引擎相比, Three.js 文档齐全, 案例丰富, 简单直观, 将复杂的接口简单化, 扩展性很强.

表1 技术对比

技术名称	是否需要插件	特点
Cult3D	是	基于 Java, 文件量小, 图像质量好
Viewpoint	是	基于 XML, 主要运用于物品展示的产品宣传
Babylon.js	否	倾向于基于 Web 的游戏开发与碰撞检测和抗锯齿
SceneJS	否	适合工程学和医学上常用的高精度模型需求
Cesium	否	适合 3D 地球和 2D 地图的开发
Three.js	否	文档齐全, 方便拓展

在相同的硬件环境下 (处理器: Intel Core i7-6500U @ 2.50 GHz 双核, 内存容量: 16 GB), 将本文方法与以 Flash 为例的传统漫游技术和目前商用较为广泛的 Pano2VR 全景漫游技术进行对比, 计算机使用性能比较结果如表 2 所示, 3 种技术在主流浏览器的 CPU 使用率对比如图 9 所示. 由表 2 和图 9 可以看出, 本技术在网站平均加载速度、CPU 使用率和物理内存的使用情况 3 个方面都略胜一筹. 由此可见, 本文借助 Three.js 完成的全景漫游系统, 系统占用空间更少, 成本更低, 系统稳定性和加载效率更高.

表2 性能比较

方法	本文	Flash	Pano2VR
网站平均加载速度 (ms)	39	47	40
CPU 使用率 (%)	3	8	5
物理内存 (%)	47	52	49

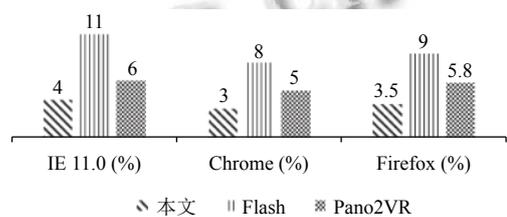


图9 3种技术在主流浏览器的CPU使用率对比

## 5 结语

本文以解决传统全景展示技术开发周期长、对硬件要求高、工序繁琐的问题作为出发点, 结合传统漫游技术和目前基于 HTML5 和 WebGL 的全景漫游技术, 在 HTML5 框架上设计开发了一套基于 WebGL 的全景虚拟漫游系统. 通过 HTML5 编辑 WebGL 的图像 API 进行图像呈现, 同时对 HTML5 的组件进行硬件加速渲染, 确保全景内容读取的流畅性和交互的灵活性, 实现在线快捷体验全景虚拟漫游的功能. 和传统全景展示技术相比, 免去安装 Flash 等渲染插件的繁琐工序, 系统要更加轻量级, 资源使用率更低, 开发更快捷. 本文将基于 WebGL 的全景技术应用到智慧旅游景区的虚拟体验当中, 采用高效的方式实现智慧旅游全景漫游系统的建立及其展示, 该系统可在门户网站、智能手机等终端设备上应用, 用户可以在浏览器中进行沉浸式的交互体验, 在线观赏智慧旅游景区的风光.

## 参考文献

- 董春侠, 司占军. 全景图技术在校园全景漫游中的应用研究. 电脑知识与技术, 2017, 13(4): 210-211, 220.
- 马磊. 基于智能解决方案的自助旅游系统. 计算机系统应用, 2017, 26(3): 57-62. [doi: 10.15888/j.cnki.csa.005636]
- 罗立宏, 陈志. Google Earth 在旅游景点展示中的应用. 计算机工程, 2009, 35(3): 277-278, 281.
- 吴勇, 罗腾元. 全景三维虚拟系统构建方法研究. 计算机工

- 程与设计, 2014, 35(5): 1858–1861.
- 5 Gray S. 3Dactyl: Using WebGL to represent human movement in 3D. City of Bath: University of Bath, 2012.
  - 6 谢磊. 基于 WebGL 的三维地形可视化研究. 科技视界, 2015, (13): 138, 141.
  - 7 赵孔阳, 朱军, 尹灵芝. 基于 WebGL 的虚拟场景网络漫游研究. 测绘与空间地理信息, 2015, 38(12): 40–43.
  - 8 黄沛然, 李莹利. 基于 HTML5 的全景图技术在社区规划中的应用. 电脑知识与技术, 2018, 14(30): 218–220.
  - 9 陶玲君. 基于 WebGL 的三维全景数字展览关键技术研究[硕士学位论文]. 杭州: 杭州电子科技大学, 2017.
  - 10 高齐琦, 江婷, 田世隆, 等. 基于 Three.js 的 3D 磁盘阵列设计. 计算机系统应用, 2018, 27(11): 241–246.
  - 11 Dirksen J. Learning Three.js: The JavaScript 3D library for WebGL. Birmingham: Packet Publishing Ltd. 2013.
  - 12 乔国臻, 张茹涵, 赵国梁. 基于 HTML5 的虚拟校园三维漫游系统研究. 测绘技术装备, 2018, 20(1): 33–36.
  - 13 Lowe DG. Object recognition from local scale-invariant features. Proceedings of the 7th IEEE International Conference on Computer Vision. Kerkyra, Greece. 1999. 1150–1157.
  - 14 文伟东, 张明. 基于 SIFT 算法的全景图像拼接技术研究. 计算机系统应用, 2017, 26(7): 227–231. [doi: [10.15888/j.cnki.csa.005899](https://doi.org/10.15888/j.cnki.csa.005899)]
  - 15 兰红, 洪玉欢, 高晓林. SIFT 优化算法及其在全景拼接图像配准中的应用. 小型微型计算机系统, 2016, 37(5): 1052–1056.
  - 16 王晨. 基于全景技术的校园三维实景展示应用研究[硕士学位论文]. 天津: 天津科技大学, 2016.
  - 17 张巧云, 张文君. 基于无人机摄影的 720°全景图像制作技术. 现代计算机, 2018, (31): 64–67. [doi: [10.3969/j.issn.1007-1423.2018.31.016](https://doi.org/10.3969/j.issn.1007-1423.2018.31.016)]
  - 18 魏云申. 基于 WebGL 的全景 3D 漫游系统的设计与实现[硕士学位论文]. 南京: 南京大学, 2016.
  - 19 陈永峰. 车载连续全景漫游系统设计与实现[硕士学位论文]. 北京: 北京建筑大学, 2017.