清華大學
Tsinghua University

海洋楼空间管理平台搭建

OCEAN BUILDING SPACE MANAGEMENT PLATFORM

汇报人：刘龙祥、罗振华

指导老师：胡振中

时间：2023年6月2日

CONTENTS

清華大學
Tsinghua University

01

背景

Background

清华园区海洋大楼

存在问题

- **信息分散**：各楼层、房间、设备分布和使用信息分散存储，没有统一平台进行信息集成
- **安全隐患**：安全负责人无法得知房间的人员与设备信息，存在安全隐患

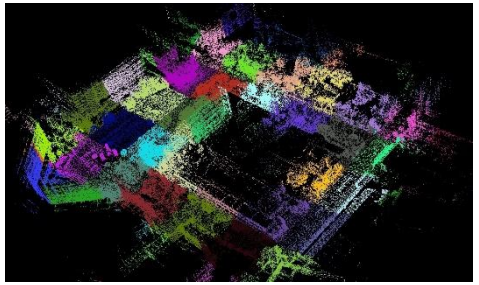
解决措施

- 借助**三维激光扫描技术**，对海洋楼进行整体建模，形成三维可视化立体全景模型 → 漫游展示系统
- 借助**网络与信息管理系统**，将三维可视化模型部署到云端，构建海洋楼空间管理平台 → 信息管理系统

三维激光扫描技术

- 利用高速激光扫描测量的方法，可大面积、高分辨率、快速地获取物体表面各个点的(x y z)坐标、反射率、(R.G.B)颜色等数据信息，为快速复建出1:1真彩色三维点云模型提供的一种全新技术手段。

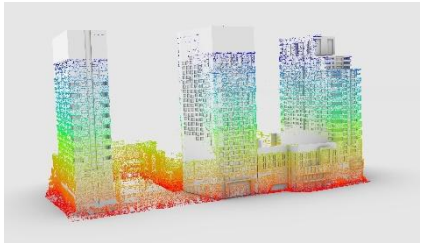
工作流程



点云图像

现场实测扫描 → 点云数据处理 → 依据点云建模 → 三维可视化立体全景模型

三维激光扫描仪



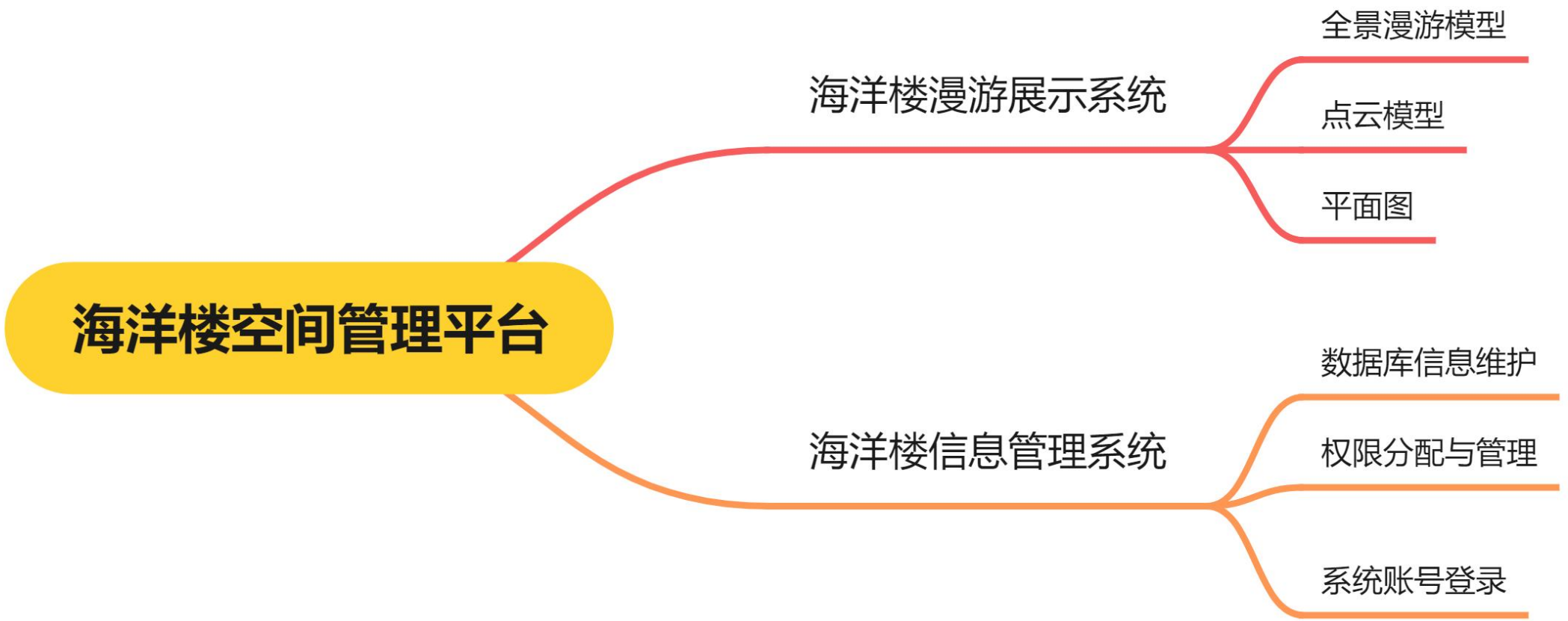
点云模型

02

平台概述

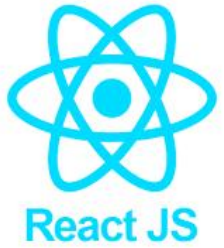
Platform Overview

平台结构示意图


海洋楼空间管理平台
海洋楼漫游展示系统
全景漫游模型
点云模型
平面图
海洋楼信息管理系统
数据库信息维护
权限分配与管理
系统账号登录

清華大學 Tsinghua University

平台框架



几何预处理

Javascript

CSS

Next.js

.NET API Gateway

API - 登入/注册 服务

API - 信息处理服务

后端服务

数据导入

存储数据

添加数据

信息导出

身份验证和授权
- JSON web Token

数据管理 - 网络协议
REST API 请求
- HTTP POST
- HTTP GET
- HTTP PUT
- HTTP UPDATE

数据 （类与属性映射）
- 标准化信息结构
- .NET Standard 类别库

.NET API Gateway

API - 登入/注册 服务

API - 信息处理服务

Microsoft SQL Server

API Controller

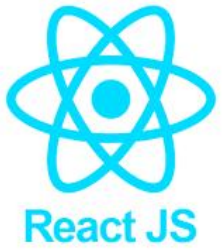
以控制器的方式实现后端 API 的功能，并能与相关的数据库做联动存储与编写数据

前端服务

数据可视化

三维模型渲染
- WebGL
- Three.js

网页交互
-React JS
-Next.js



使用者交互

信息与可视化交互
- javascript
- typescript

Next.js

一个构建于 Node.js 之上的开源 Web 开发框架，并支持基于 React 的 Web 应用程序功能，通常利用于服务器端处理和生成静态网站

几何预处理

信息模型
- rvt 格式

点云数据
- las 格式

全景照
- png 格式

数据清理

数据整合

几何预处理

信息轻量化

清華大學
Tsinghua University

03

功能介绍及具体实现

Function Introduction and Realization

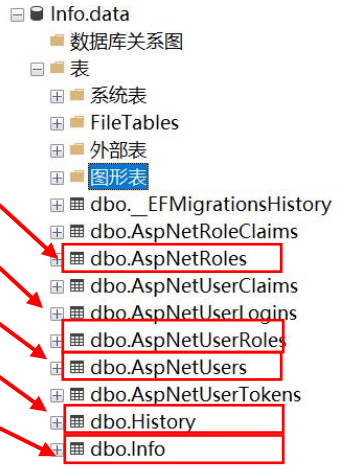
功能介绍及具体实现

海洋楼信息管理系统

- 数据库信息管理

Info数据库

Role表
User-Role表
User表
History表
Info表




```
public class Info
{
    [Key, Column(Order = 0)]
    3 个引用
    public string? RoomNumber { get; set; }
    0 个引用
    public int Floor { get; set; }
    0 个引用
    public float? Area { get; set; } = null;
    0 个引用
    public string? SpaceType { get; set; } = null;
    0 个引用
    public string? SpaceManager { get; set; } = null;
    0 个引用
    public string? SafetyManager { get; set; } = null;
    0 个引用
    public string? SafetyOfficer { get; set; } = null;
    0 个引用
    public string? SafetyAssistant { get; set; } = null;
```

Info类

楼层	房间标号	面积	空间类型	空间领用人
1	101		行政办公室	总务
	102		机房	总务
	103	25	行政办公室	总务
	安保中心	58		总务
	仓库	291	科研实验室	
2	2楼玻璃房	197	硕士共享学习空间	
	203	18.51	空置	黄翠

海洋楼部分房间信息

列名	数据类型	允许 Null 值
RoomNumber	varchar(50)	☐
Floor	int	☐
Area	real	☐
SpaceType	varchar(50)	☑
SpaceManager	varchar(50)	☑
SafetyManager	varchar(50)	☑
SafetyOfficer	varchar(50)	☑
SafetyAssistant	varchar(50)	☑
		☐

Info表

功能介绍及具体实现

海洋楼信息管理系统

- 系统注册与登录

Login

POST /Login

Register

POST /Register/CreateUserRole

POST /Register/SendResetPasswordToken

```
LoginRequest {
    userName        string
                    nullable: true
    password
}
```

```
RegisterRequest
    userName
    password
    email
    role
}
```

角色名称	访问范围
普通	公共空间、会议室、开放实验室
教师	公共空间、会议室、开放实验室、领用责任人/安全员为该教师的空间
管理员	所有空间

```
[HttpPost]
0 个引用
public async Task<IActionResult> Login(LoginRequest req,
            [FromServices] IOptions<JWTOptions> jwtOptions)
{
    string userName = req.UserName;
    string password = req.Password;
    var user = await userManager.FindByNameAsync(userName);

    不存在{userName}");

    er.CheckPasswordAsync(user, password);

    ");

    );
    es.NameIdentifier, user.Id.ToString()));
    es.Name, user.UserName));
    .GetRolesAsync(user);
    )

    mTypes.Role, role));

    claims, jwtOptions.Value);
```

web token

```
[HttpPost]
0 个引用
public async Task<ActionResult> CreateUserRole(RegisterRequest req)
{
    bool roleExists = await roleManager.RoleExistsAsync(req.Role);
    if (!roleExists)
    {
        Role role = new Role { Name = req.Role };
        var r = await roleManager.CreateAsync(role);
        if (!r.Succeeded)
        {
            return BadRequest(r.Errors);
        }
    }
    User user = await userManager.FindByNameAsync(req.UserName);
    if (user == null)
    {
        user = new User { UserName = req.UserName, Email = req.Email, EmailConfirmed = true };
        var r = await userManager.CreateAsync(user, req.Password);
        if (!r.Succeeded)
        {
            return BadRequest(r.Errors);
        }
        r = await userManager.AddToRoleAsync(user, req.Role);
        if (!r.Succeeded)
        {
            return BadRequest(r.Errors);
        }
    }
    return Ok();
}
```

Code Details

200

Response body

```
eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNobWFjLXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1laWRlbnRpZmllciI6IjIiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoibGx4IiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L2lkZW50aXR5L2NsYWltcy9yb2xlIjoidXNlciIsImV4cCI6
TvmgnNp8cPCMt6_m6GKxFqXA9AKD6moDiYsCS_1Q
```

Download

Response headers

```
content-type: text/plain; charset=utf-8
date: Wed,31 May 2023 12:00:35 GMT
server: Kestrel
transfer-encoding: chunked
```

Authorize

授权认证

功能介绍及具体实现

海洋楼信息管理系统

- 权限分配与管理

Swagger

ASP.NET

Authorize

授权认证

Info

GET	/Info/GetAllRoomInfo
GET	/Info/GetSingleRoomInfo_Admin/{RoomNumber}
GET	/Info/GetSingleRoomInfo_User/{RoomNumber}
GET	/Info/GetSingleRoomInfo_Teacher/{RoomNumber}
GET	/Info/GetSingleRoomHistoryInfo/{RoomNumber}
PUT	/Info/UpdateRoomInfo/{RoomNumber}
POST	/Info/AddRoomInfo
DELETE	/Info/DeleteRoomInfo/{RoomNumber}

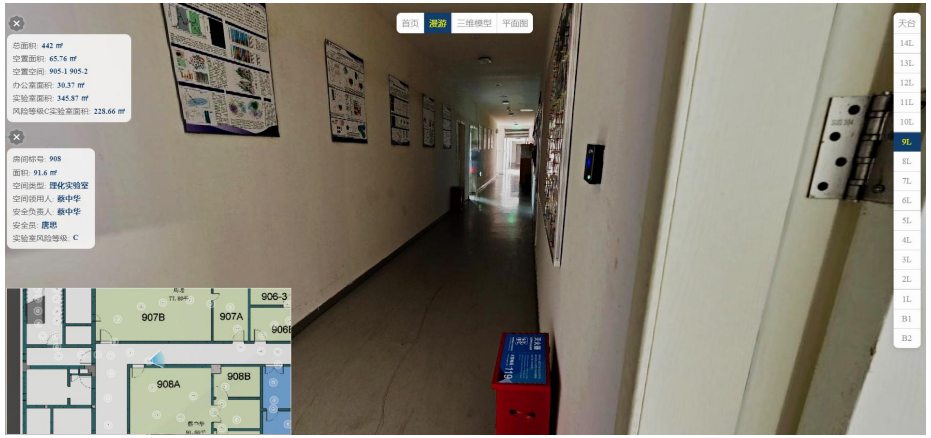
权限管理

权限管理

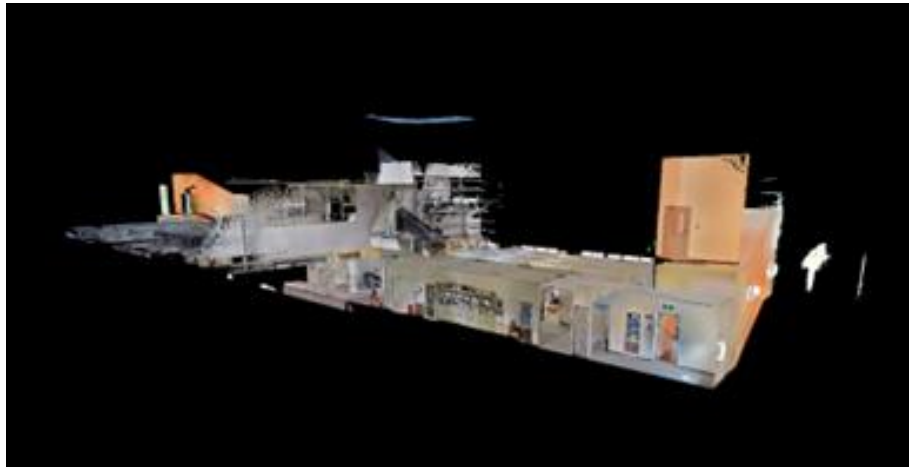
```
[HttpGet]
[Authorize(Roles = "Admin")]
0 个引用
public async Task<ActionResult<IEnumerable<Info>>> GetAllRoomInfo()
{
    if (_context.Info == null)
    {
        return NotFound();
    }
    return await _context.Info.ToListAsync();
}

// GET: InfoApi/OceanBuilding/RoomNumber
[HttpGet("{RoomNumber}")]
[Authorize(Roles = "Admin")]
1 个引用
public async Task<ActionResult<Info>> GetSingleRoomInfo_Admin(string RoomNumber)
{
    if (_context.Info == null)
    {
        return NotFound();
    }
    var RoomInfo = await _context.Info.FindAsync(RoomNumber);

    if (RoomInfo == null)
    {
        return NotFound();
    }

    return Ok(RoomInfo);
}
[HttpGet("{RoomNumber}")]
[Authorize(Roles = "user")]
0 个引用
public async Task<ActionResult<Info>> GetSingleRoomInfo_User(string RoomNumber)
```

功能介绍及具体实现

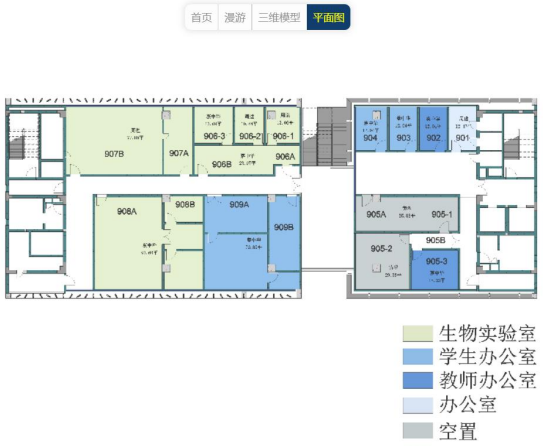
海洋楼展示系统



全景图漫游模式



点云浏览模式



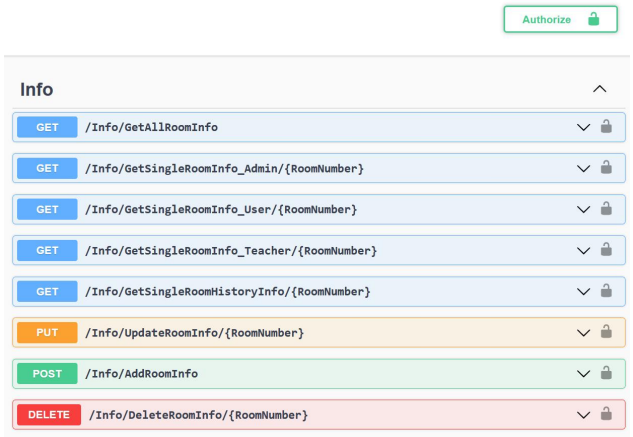
平面图显示模式

功能介绍及具体实现

□ 海洋楼展示系统

2023年空间管理信息

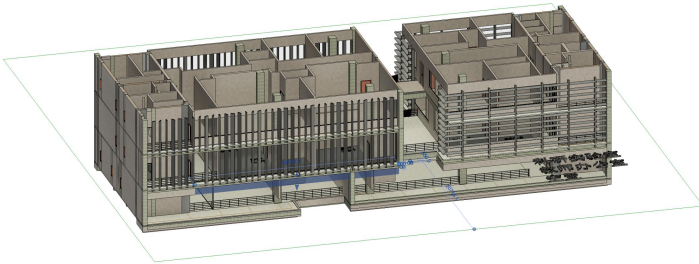
楼层	房间标号	面积	空间类型	空间领用人
1	101		行政办公室	总务
	102		机房	总务
	103	25	行政办公室	总务
	安保中心	58		总务
	仓库	291	科研实验室	
2	2楼玻璃房	197	硕士共享学习空间	
	203	18.51	空置	黄翠



海洋楼九层门号识走向图

平面图显示模式

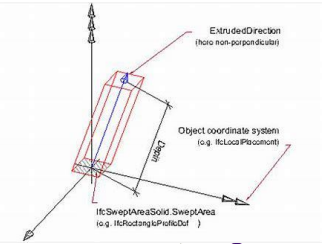
2014年海洋楼信息模型



```
IfcSpace
    IfcRectangleProfileDef
    IfcArbitraryProfileDefWithVoids
    IfcArbitraryClosedProfileDef
```

IfcOpenShell

```
point_array:[(-3475.2, -2627.86),
             (-3475.2, 6982.95), (
             3224.8, 6982.95), (3224.8,
             -2627.86)],
void:[],
depth:4200.0,
elevation:34750.0
```

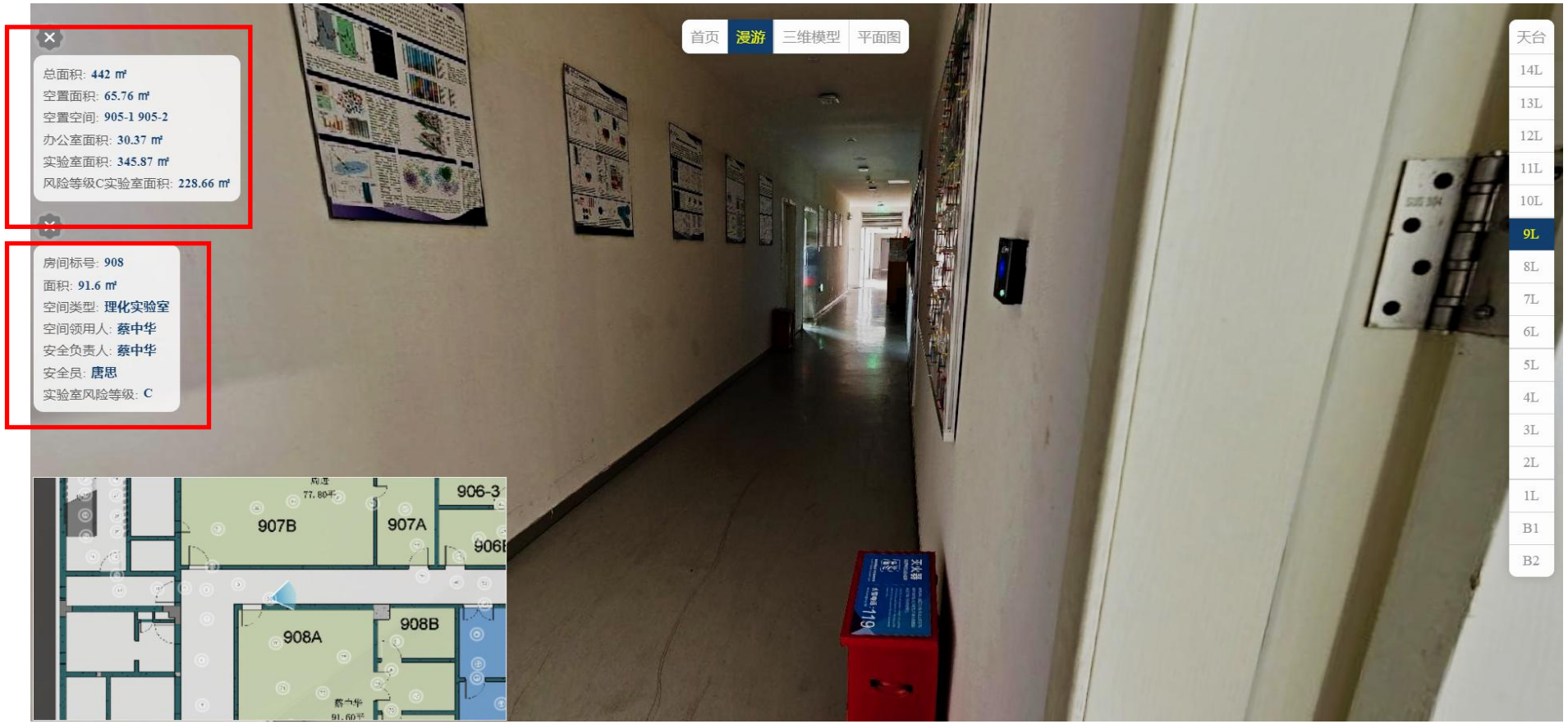


功能介绍及具体实现

海洋楼展示系统

空间信息

平面图与
扫描站位置



全景图漫游模式

功能介绍及具体实现

海洋楼展示系统

生成全景图

```
THREE.TextureLoader
```

three.js

功能介绍及具体实现

□ 海洋楼展示系统

las档案转换

- 颜色转换 - 16 bit 归一化转换 [0,1]
- las 转换并压缩成 pcd (binary)

✓ Level of Detail Down Sampling: OPEN3D

```python
import numpy as np
import open3d as o3d
import laspy
import math
from tqdm import tqdm
import os

def colorArray_16to8bit(matrix: np.array, name:str = "color"):
    modified = []
    for index in tqdm(range(len(matrix)),f"Calculating {name}"):
        color_8bit =  math.floor((matrix[index] +1)/(2**16)*256 -1)
        if color_8bit>0:
            modified.append(round(color_8bit/255,4))
        else:
            modified.append(0)

    return modified

f = laspy.read("./Source/laser.las")

point_cloud = o3d.geometry.PointCloud()

# points = np.asarray(np.vstack((f.x,f.y,f.z)).transpose())
# y_negative = -1 * np.asarray(f.y)
points = np.asarray(np.vstack((f.x,f.z,f.y)).transpose())

if not(os.path.exists("./color.npy")):
    red = colorArray_16to8bit(f.red, "red")
    green = colorArray_16to8bit(f.green,"green")
    blue = colorArray_16to8bit(f.blue,"blue")
    color = np.asarray(np.vstack((red,green,blue)).transpose())
    np.save("color",color)

colors = np.load("./color.npy")
point_cloud.points = o3d.utility.Vector3dVector(points)
point_cloud.colors = o3d.utility.Vector3dVector(colors)

pc_downSample = point_cloud.uniform_down_sample(10)


o3d.io.write_point_cloud(f"./testing_overall_100_10.pcd", pc_downSample, write_ascii=False, compressed=True,
```

不采用

分割点云档案： 100000 points/csv file

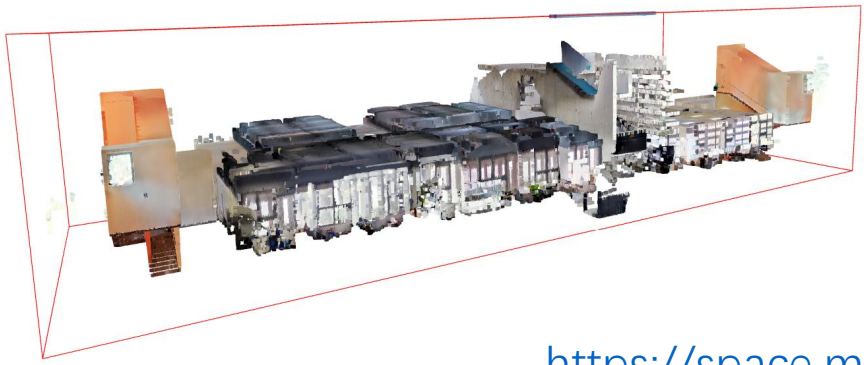
- 342个4500kb的档案 （约1.5GB）
- 缺点： 传输速率慢，
  传输流量占据大，
  计算机内存不足 (最多150个)

海洋楼展示系统

LAS (1.2GB) -> PCD (50MB)

PCDLoader

```
function PointCloudCanvas(props:{pointCloudSrc:string}) {
  const canvasRef = useRef(null!);
  const [loader_success, setLoader_Success] = useState<boolean>(false)
  const [progress, setProgress] = useState<number>(-1)
  useEffect(() => {
    if(!props.pointCloudSrc) return

    const loader = new PCDLoader();

    // load a resource
    loader.load(
      // resource URL
      // '/pointcloud/haiyanglou/9L/overall.pcd',
      props.pointCloudSrc,
      // called when the resource is loaded
      function ( points ) {
        setLoader_Success(true)
        const scene = new THREE.Scene();
        const camera = new THREE.PerspectiveCamera(
          70,
          window.innerWidth / window.innerHeight
          ,0.001,
          2000
        );
        const renderer = new THREE.WebGLRenderer({ canvas: canvasRef.current });
        renderer.setSize(window.innerWidth, window.innerHeight);
        scene.add( points );
        const controls = new OrbitControls(camera,renderer.domElement);

        camera.position.set(19.08,7.85,27.04 );
        // console.log(camera)

        camera.rotation.set(-0.2823,0.5955,0.1613)
        function animate() {
          requestAnimationFrame( animate );

          // required if controls.enableDamping or controls.autoRotate are set to true
          controls.update();

          renderer.render( scene, camera );
        }
        animate();
      },
      // called when loading is in progresses
      function ( xhr ) {

        // console.log( ( xhr.loaded / xhr.total * 100 ) + '% loaded' );
        setProgress(Math.ceil(xhr.loaded/ xhr.total * 100) )

      },
      // called when loading has errors
      function ( error ) {
          setProgress(-1)
        console.log( 'An error happened' );
        console.log(error);
        setLoader_Success(false)
      }
```



https://space.matrix4.xyz/

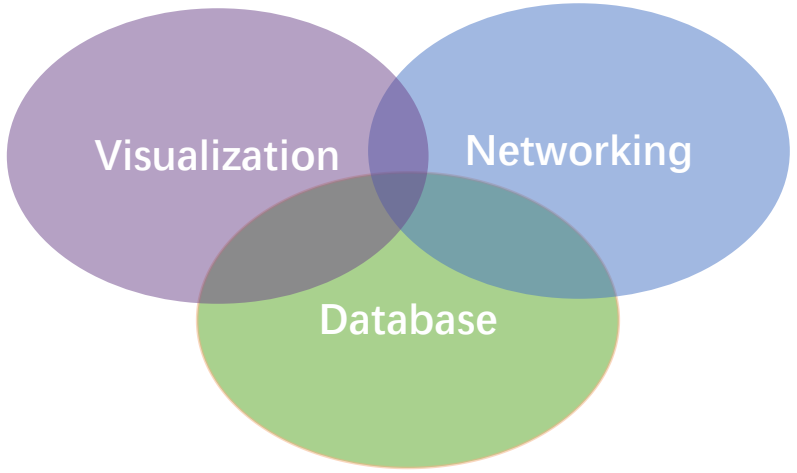
04

总结展望

Summary and Outlook

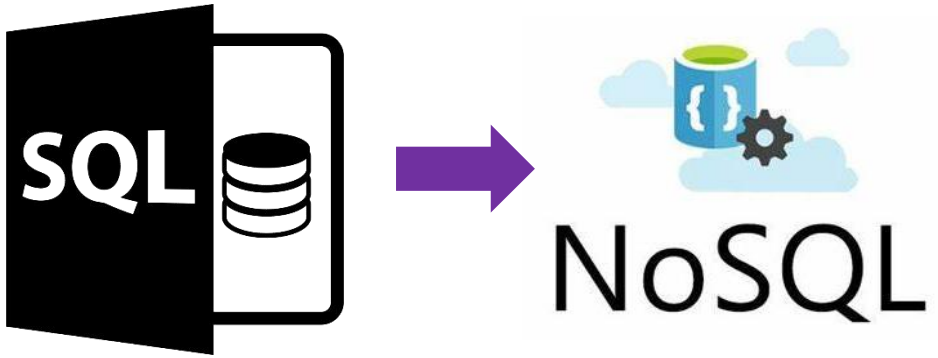
功能总结

- 实现房间信息、用户信息、角色信息以及历史记录的存储；
- 实现用户账号的注册与登录，分配不同的角色信息；
- 根据角色信息实现不同API功能接口的分配；
- 多方面利用并处理激光扫描器所收集与生成数据
- 将管理数据与可视化进一步结合，使得空间管理更便捷



下一步计划

- 考虑将关系型数据库转换成非关系型数据库结构，便于在云端存储
- 增加将点云数据与空间信息之间的互动，并有效同页面展示整栋楼的点云数据



清華大學
Tsinghua University

谢谢观看

Thank you for watching

汇报人：刘龙祥、罗振华

指导老师：胡振中

时间：2023年6月2日