摄像头扫描房型成像方案设计

# 需求

使用IOS提供的AR扫描平面的功能，实现用户使用移动设备的摄像头，在房间内扫描墙体平面，然后根据平面的位置和尺寸，绘制户型图。（详细的需求可以看需求分析文档）

# 方案设计

## 2.1墙体数据的收集与生成方案

### 2.1.1 实现原理

每一个平面被扫描出来了之后并不是一个规则的矩形，而是一个不规则的多边形，多边形的顶点数据是一组二维的数据，现在使用的坐标轴的Y轴负方向是重力方向，所以这些顶点数据的Y轴都是0。通过旋转、平移之后可以计算出平面顶点数据在世界坐标中的位置。通过这些顶点数据，我们可以计算出墙的位置、宽和高。

### 扫描墙体

扫描墙体主要是收集墙体的宽度、高度和位置。收集墙体的宽度只需要一个表示墙体宽度的平面。收集墙体高度只需要一个能表示墙体高度的平面。墙体的位置可以使用宽度平面计算出来。

**（1）扫描墙体**

因为在一个房型中，每面墙的高度都是一致的，为了提高扫描房型的效率，只需要扫描一面墙的高度，最后使用最高的一面墙作为所有墙的高度。

****

第一步：扫描墙体高度

找一面墙扫描高度方向的平面

第二步：扫描墙体的宽度和位置

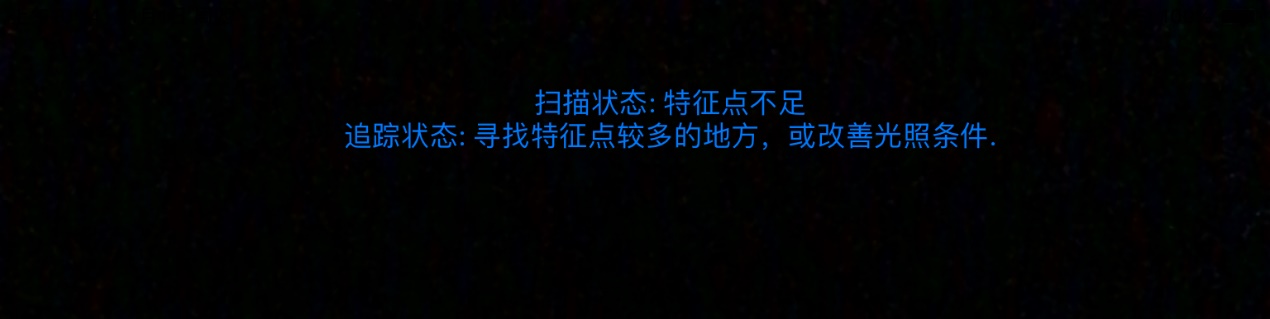
扫描墙宽度方向的平面

第三步：删除多余的平面

在房间可能会存在不是墙体的垂直面，需要删除掉。ios给出了平面的类型，但是研究之后发现目前还不能使用。所以还是要手动删除。

**（2）扫描的过程中给用户提示当前扫描的状态**

提示用户如何操作能提高扫描的效果



### 收集数据

平面的边缘几何数据是一份二维的数据，是以x轴和z轴构成的一份俯视图数据，并且坐标原点在平面的中心点。这份数据是不能直接使用的，使用之前必须要做旋转和平移。旋转矩阵可以使用锚点在世界坐标中的变换矩阵就可以了，平移的位置可以使用节点在世界坐标中的位置。

1. **计算平面上的顶点的在世界位置**

第一步：获取平面的边缘几何数据

可以使用Anchor.geometry的属性去获取

第二步：把二维的几何数据转换成三维的数据

每个锚点都有一个齐次矩阵，这个矩阵我们只用到3\*3的部分，就可以把二维的数据转换成3维的。

第三步：把几何数据中的所有的点都平移到平面上

平面上的父节点是扫描之后提供的，父节点可以提供平面中心点在世界坐标中的位置，可以使用这个位置把所有的点平移到平面上

**（2）收集高度数据**

方法一

• 因为y轴负的方向与重力的方向是一致的，可以对顶点数据做一次按照y轴排序，使用最高点减去最低点，就可以计算出墙的高度

方法二

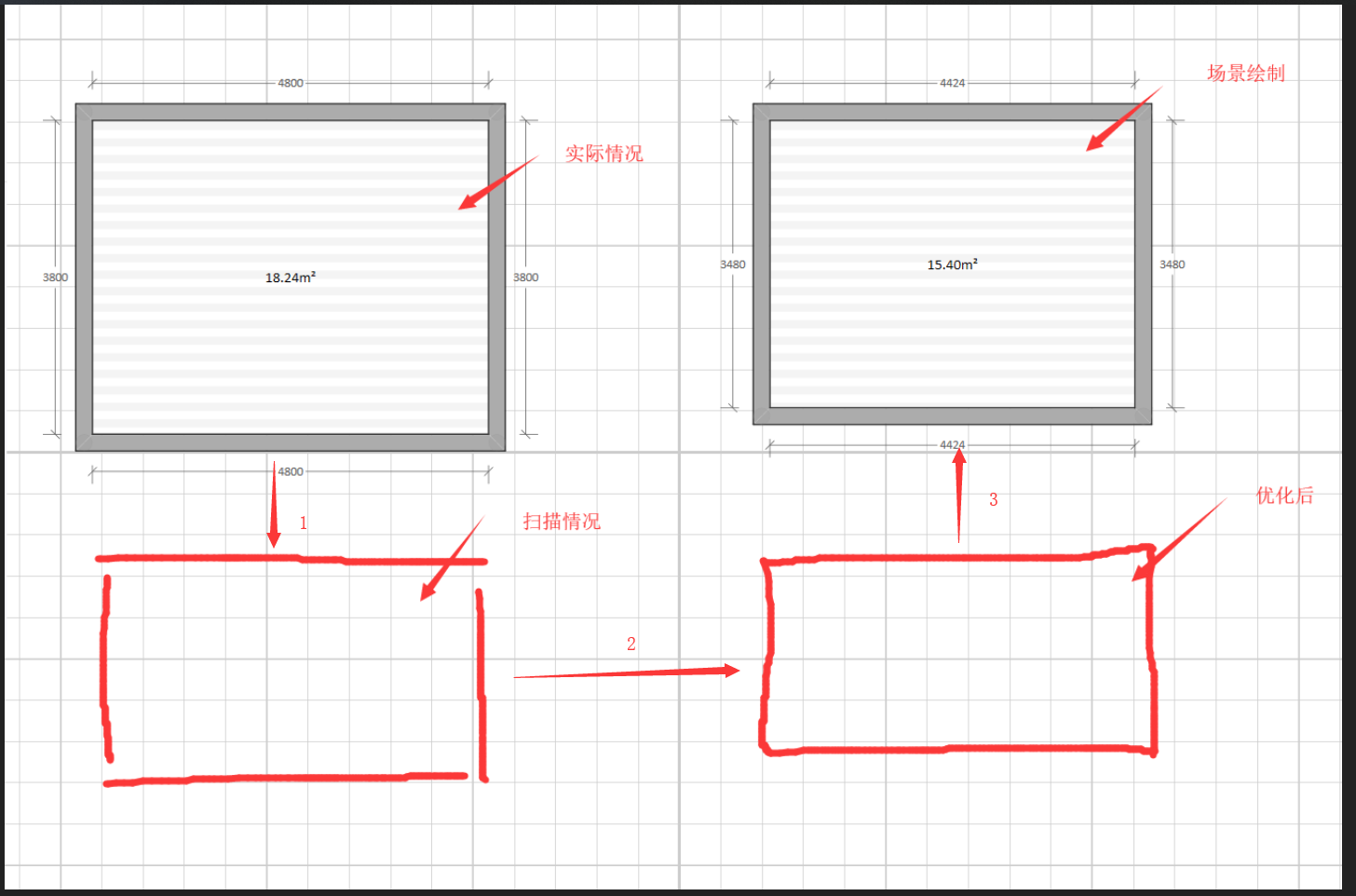
• 使用ios提供的高度预估Anchor.heigth

**（3）收集宽度和位置的数据**

把平面的顶点数据延重力方向做一次投影，然后计算出距离最远的两个点，用这两个点最为起始点和终点，两点之间的距离就是宽度。

### 2.1.4 生成对接数据

因为线段可能是不连续的，需要一个算法把这些线段关联起来。而且一堵墙可能会被扫面多个面的情况，所以需要对线段的信息做一次优化。**优化的算法不计划在移动设备上实现，因为以后可能多种移动设备都会使用这套算法与场景对接。所以建议在场景或者服务器上实现这个算法。**



1. **优化数据**

**情况一：两条线段是首尾相连的**

首先计算两条线段首尾之间的距离，距离小于0.3m的认为是有关联的，将有关联的两端，连接起来。

第一步：使用一元一次方程计算出两条线段的交点

第二步：使用线段的两个点计算出一元一次方程的斜率和常量值

第三步：联立方程组求出条线段的交点

第四步：把两条线段接近的两端坐标改为交点的坐标

**情况二：两条线段是中间相连的**

第一步：首先判断两条线段是否相交，并且不是首尾相交

第二步：判断两条线段是否垂直，不垂直则认为两条线段没有关系（定义为垂直的角度范围是88-92，因为扫描的结果可能有误差）

第三步：求两条线段的一元一次方程

第四步：计算两条线段的交点

第五步：两条线段分别计算，到交点的距离，距离超过0.3m，则认为两条线段没有关联

第六步：两条线段分别计算出与交点最近的顶点，最后顶点位置改成交点的位置

**情况三：线段重叠**

线段相交，并且平行的则认为线段重叠。

**情况四：扫描的过程中扫描了一堵墙的两个面或多个面的情况**

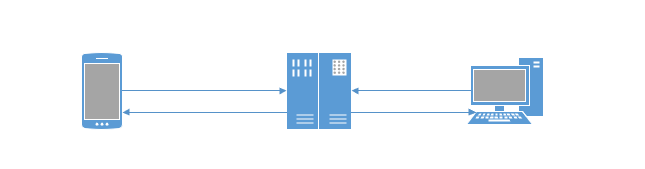
还没有研究过方案（或者让用户手动删除）

1. **生成与场景对接的数据**

对接场景使用的是线段，把数据都转换成线段的列表

## 2.2通信方案

使用web server作为服务器，移动设备通过服务器发送信息给场景设计器。



### 2.2.1移动设备的身份识别

使用衣柜设计器web版的账号作为为身份识别的账号。移动设备通过，移动设备上的登陆界面输入账号密码登陆。场景通过场景的登陆界面登陆。

注册是在衣柜设计器web版的的注册界面注册。

### 2.2.2移动设备与远程服务器

信息交互使用的是http协议，提交方式都是使用post方法，目前不考虑信息加密。提交的文件格式是json文件格式。

**（1）登陆的信息交互**

客户端提交账号和密码的信息，然后服务器验证。登录成功之后，服务器返回一个用户的GUID，后面的通信都用这个GUID去识别用户的身份。因为现在服务器是搭在web衣柜设计器上，所以尽量不使用web上的身份识别方式，这样方便以后分离。

**（2）提交墙的数据**

客户端提供墙体的线段数据，同时提供用户的GUID。服务器通过GUID判断用户的身份，最后把墙体数据发送给对应的场景设计器。

### 2.2.3远程服务器与场景

因为用户在移动设备登陆后要交换墙体数据，所以用http协议不太方便。所以使用socket。目前场景与渲染服务器已经有一套比较成熟的通信模块，所以与场景的通信可以使用渲染组封装好的服务器模块。

**（1）登陆的信息交互**

客户端提交账号和密码的信息，然后服务器验证。登录成功之后，服务器返回一个用户的GUID，后面的通信都用这个GUID去识别用户的身份。所以场景需要记住这个GUID。

**（2）接收墙的数据**

移动设备提交墙体数据之后，服务器会向场景发送一份墙体数据，场景只需要接收就可以了。

## 2.3与场景对接方案

因为在场景绘制房型时，只需要绘制墙体，地板和天花板是自动生成的，所以与场景对接时，只需要传递墙的信息就可以了。对接使用的是中间的数据，场景使用中间数据，自行转换成场景需要的数据。由于坐标系是不固定的，场景绘制的时候应该使用相对位置去画墙。

**（1）数据结构**

墙体数据是线段的列表，线段的数据，包括起始点、终点和高度。坐标的单位是毫米

## 2.4界面

先做一个简单的界面，设计完成了之后，再让美工帮设计新的界面样式。

### 2.4.1场景

场景只需要做一个登录的界面就行了，让用户输入账号、密码。跳转到注册的网页。

### 2.4.2移动设备

**（1）登陆的页面**

界面提供账号、密码的输入，还有注册的网页的跳转。

**（2）****提交到场景的按钮**

用于把墙体数据提交到场景，给场景绘制墙体