

2012 年英特尔杯大学生电子设计竞赛嵌入式系统专题邀请赛

2012 Intel Cup Undergraduate Electronic Design Contest

- Embedded System Design Invitational Contest

作品设计报告

Final Report



指尖上的设计

-- 基于 Intel® Atom™ 平台的室内交互设计游览系统

学生姓名: 姜逸坤 谢君朋 王雨舟

指导教师: 杨 刚

参赛学校: 西安电子科技大学

2012 年英特尔杯大学生电子设计竞赛嵌入式系统专题邀请赛

参赛作品原创性声明

本人郑重声明：所呈交的参赛作品报告，是本人和队友独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果，不侵犯任何第三方的知识产权或其他权利。本人完全意识到本声明的法律结果由本人承担。

参赛队员签名：

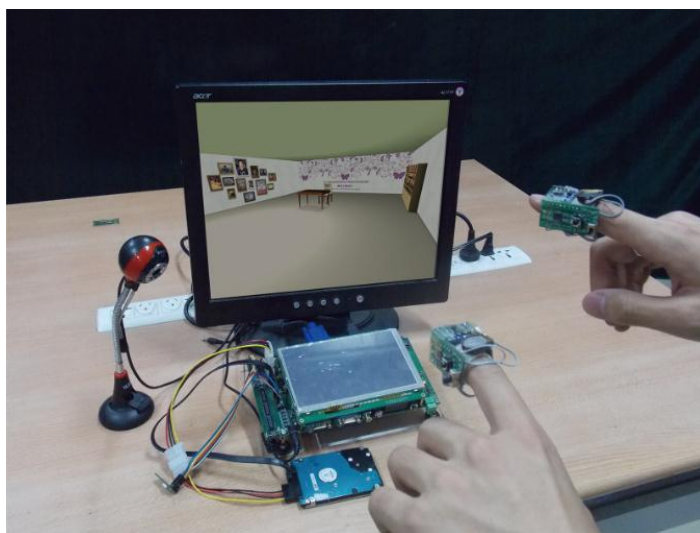
日期： 年 月 日

指尖上的设计

—— 基于 Intel® Atom™ 平台的室内交互设计游览系统

THE DESIGN ON THE FINGERTIPS

— THE INTEIOR DESIGN TOUR SYSTEM BASED ON Intel® Atom™ Platform



摘要

针对室内装修设计这一民众关注的问题，以构建一个直观、立体、便捷的室内交互设计系统为目的，设计了这一新概念的室内交互设计游览系统。系统将增强现实技术、现代计算机技术与室内设计相结合，以搭载着 Yocto 操作系统的高性能、可配置的 Intel Atom 嵌入式系统为平台，结合精心设计的手势交互操控指环，实现了新概念的室内装修的交互设计游览系统，并在其中使用 3D 引擎，增加了碰撞检测、重力场以及逼真的图像渲染，使得系统场景更接近于现实。特意设计的模型与场景管理功能，也使得系统的可扩展性、易定制性大大提高。

本系统旨在提供一个立体的室内交互设计系统，实现了一个全新的人机交互方式，在售楼宣传、模型展示、室内设计、设计教育等诸多领域有良好的应用前景。

关键词：室内设计，人机交互，3D 引擎，增强现实

THE INTERIOR DESIGN TOUR SYSTEM

ABSTRACT

Faced with the problem of indoor design which has caught much attention from the public, and in order to build an intuitive, three-dimensional, and convenient indoor interactive design system, we designed the new indoor interactive design tour system. This system combined virtual-reality technology, modern computer technology and indoor design together, and is based on the high-performance, configurable Intel Atom embedded system which carries the Yocto operating system. Together with the delicately designed gesture interactive manipulation ring, we carried out this system. In addition, we used the 3D engine, introduced the collision detection, gravity field and realistic image rendering, making the system scenario closer to reality. In particular, the management features of the model and scene also made the system more scalable, easier to customize. This system is designed to provide a three-dimensional indoor interactive design system, realize a new mode of interpersonal interaction. This system is promising in promotional sales, model exhibits, interior design, designing education and other fields.

Keyword: Indoor Design, Human-computer Interaction, 3D Engine, Augmented Reality

目录

一 系统概述	5
1.1 背景与意义.....	5
1.2 设计目标与预期效果.....	5
二 系统设计方案.....	7
2.1 系统总体设计方案.....	7
2.2 系统硬件设计.....	8
2.2.1 硬件结构概述.....	8
2.2.2 Intel® Atom™ 平台	9
2.2.3 手势交互操控指环.....	9
2.2.4 视频采集设备.....	10
2.3 系统软件设计.....	10
2.3.1 软件结构设计概述.....	10
2.3.2 系统与软件整体信息.....	11
2.3.3 室内设计中心(主界面)模块	12
2.3.4 外景模型展示功能.....	12
2.3.5 室内设计游览功能.....	13
2.3.6 场景管理功能.....	13
2.3.7 模型管理功能.....	13
三 系统实现原理.....	14
3.1 手势交互操控指环的设计.....	14
3.1.1 手势识别部分.....	14
3.1.2 手势映射驱动部分.....	14
3.2 外景模型展示的设计.....	15
3.2.1 标记识别.....	15
3.2.2 坐标系建立与转换.....	15
3.2.3 模型实时渲染.....	17
3.3 室内设计游览的设计.....	18
3.3.1 场景与模型的设计.....	18
3.3.2 基于图像引擎的场景渲染.....	18
3.3.3 世界与相对坐标的转换.....	20
3.3.4 物体之间的碰撞检测.....	20
3.3.5 物体的拣取与响应.....	21
四 系统测试与结果分析.....	22
4.1 外景模型展示的标志物识别率.....	22
4.2 室内交互设计场景的图形性能.....	22
4.3 手势交互操控指环的识别率.....	22
五 系统特色与创新.....	23
参考文献	24
附录	24

一 系统概述

1.1 背景与意义

室内设计是一门以“创造满足人们物质和精神生活需要的室内环境”为目的的艺术。通过对家具、灯光等室内装饰的设计布置，来设计一个有舒适体验的室内家居环境。在人的一生之中，很大一部分时间是在自己的家中度过的，家里的室内环境更是会影响人们每天的情绪与心情。因此，房屋的室内装修设计，几乎已经成为每个人需要考虑的问题。

然而，由于室内设计的专业性很强，枯燥繁琐的图纸让人们望而却步；室内设计的专业人员设计出来的效果图，不能完全满足用户的主观需要；进行室内设计的人，不能真切地感受到室内设计完成后的立体化效果。

这样导致人们浪费了宝贵的时间和钱财，却得不到一个自己想要的“家”。

1.2 设计目标与预期效果

本作品依托 Intel Atom 平台，将增强现实、现代计算机视觉、现代电子科学等技术紧密结合，实现了一个具有丰富交互体验的室内设计游览系统。人们可以在立体的室内场景中穿梭游览，同时，也可以便捷地用直观的手势去设计布置室内环境。

为了能够实现室内交互设计与室内装修效果立体游览效果，系统设定了如下设计目标与预期效果：

(1) 手势交互操控指环

提供了一个轻便、精巧的操控设备，可以直观地操控布置家具、自由的穿梭。该设备为一对精巧的指环，用它可以完全取代鼠标，键盘等传统的输入设备。只需要带上指环，舞动自己的双手，便可以直观、高效地完成立体的室内设计。同时，利用指环可以自由穿梭在室内设计的场景之中，达到一种身临其境的感觉。

如图 1.1，是预期的效果图展示



图 1.1 手势交互操控预期效果图

(2) 外景模型展示功能

提供了一个基于增强现实的外景模型展示，使用者可以直观、立体地浏览房屋模型。利用外景模型展示功能，用户可以 360° 立体的观看房屋的外景模型，只需拿着精美制作的带有识别标记的房屋设计展示图。在摄像头前，便可以实时的展示一个精美的房屋外景模型，这一功能极大地丰富了模型展示的效果。预期的效果图如图 1.2 所示。

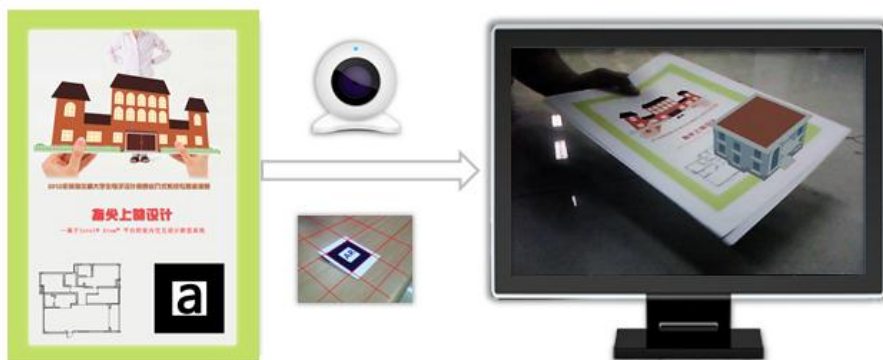


图 1.2 外景模型展示预期效果图

(3) 室内交互设计功能

提供了一个立体的设计场景，使用者可以轻松地对家居用品进行切换、调整、摆放。利用室内交互设计功能，配合手势交互指环，可以完成家居环境的自定义布置。另外，配合上场景精心制作的音乐、音效，使得用户的体验大大提高。

(4) 场景立体游览功能

提供了一个逼真的游览场景，使用者可以穿梭在设计完成的室内场景中全方位的体验。利用这个功能，室内设计等宣传公司，可以将事先设计好的室内布局展示给用户，配合上背景发声解释，能够较为立体的展示家具场景。

(5) 室内场景管理器

提供了一个方便快捷的室内场景管理器，可以轻松地完成室内场景的管理。

(6) 家具模型管理器

提供了一个高效便捷的家具模型管理器，可以方便地完成家具模型的管理。



图 1.3 室内设计软件的主界面预期效果图

二 系统设计方案

2.1 系统总体设计方案

为了能够实现室内设计的外景模型展示、交互设计、立体游览、场景管理、模型管理等功能，系统使用具有高效性能与优秀兼容性的 Intel Atom 平台作为控制与处理的核心，利用 E6x5C 芯片的可配置这一灵活特性，在其封装的现场可编程门阵列(FPGA)中实现了手势识别部分。

系统充分发挥了 Intel Atom E6x5c 系列芯片的灵活可配置性、低功耗性，将整个系统可扩展化、低功耗化。包括手势操控指环在内的各个部分都可以很方便的携带和移动，可以很便捷的完成系统，可以应用在售楼公司、装修公司、室内设计公司等场合。Intel Atom 平台的低功耗能够保证系统在这些场合内，长时间地运行，只产生非常低的功耗与发热量，保证了系统的稳定性。系统的总体设计方案结构图如图 2.1 所示。

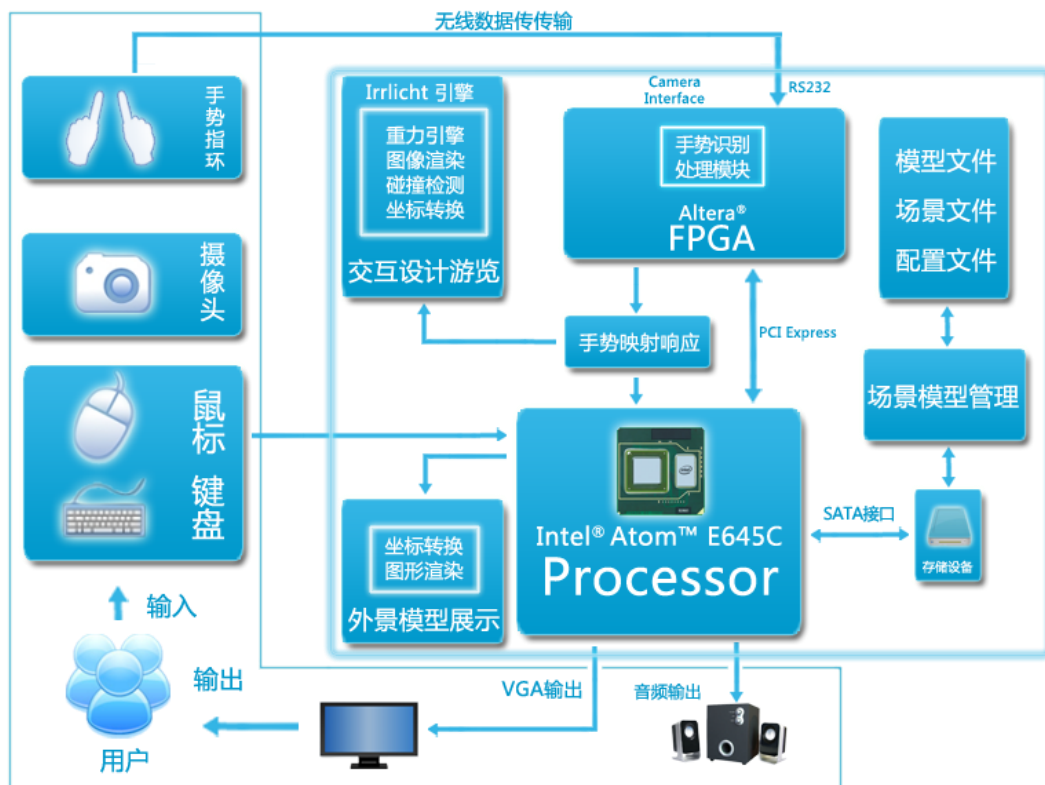


图 2.1 系统整体结构图

根据室内设计的相关特性，系统特别精心设计了一对手势交互操控指环，以保证使用者在进行室内设计、室内游览时的灵活性与直观性。系统开发了 Intel Atom 平台中的方便扩展的 FPGA，将 FPGA 作为数据通信的接口，并在其中进行数据处理、动作识别等实时性要求较高的工作。系统使用 MSP430 超低功耗单片机作为手势交互操控指环的核心，对加速度传感器的数据进行采集后，使用无线数据传输模块发送给系统控制核心。

在外景模型展示的过程中，需要增强现实的全方位立体展示，我们特意使用了 ARToolKit 与 Irrlicht 物理引擎相结合的解决方案，利用 ARToolKit 进行标记的识别与坐标转换，结合图像引擎，将精心设计的房屋外景的模型展示在实时的视频中。

由于室内交互设计、游览时，需要直观、美观、可操作强的视觉展示，系统使用可定制的嵌入式 Linux 系统——Yocto 系统，结合 Irrlicht 物理引擎出色的实时渲染特性，将一个 3D 立体的场景，展示在使用者眼前，使得整个系统的展示极具视觉性。其中，Intel Atom 平台对 OpenGL 提供了完美的支持，其高性能集成的显卡对 Irrlicht 封装的 OpenGL 渲染进行硬件加速，提高了系统性能。

在场景与模型管理中，也正是 Yocto 的灵活可定制性，完美支持了 Qt 这一跨平台的 C++ 图形界面应用程序开发框架，利用 Qt 的可视化图形的界面，将模型的预览图与信息直观地展示给使用者。

整个系统在结构上分为硬件和软件两大组成模块。

硬件主要有 Intel Atom 平台、手势交互操控指环、摄像头、显示设备、音频设备组成。

软件部分主要由室内设计中心（主界面）、外景模型展示界面、交互设计游览界面、模型管理界面、手势分析识别模块以及保证各个功能模块达到运行效果的算法组成。

2.2 系统硬件设计

2.2.1 硬件结构概述

系统硬件结构图如图 2.2 所示

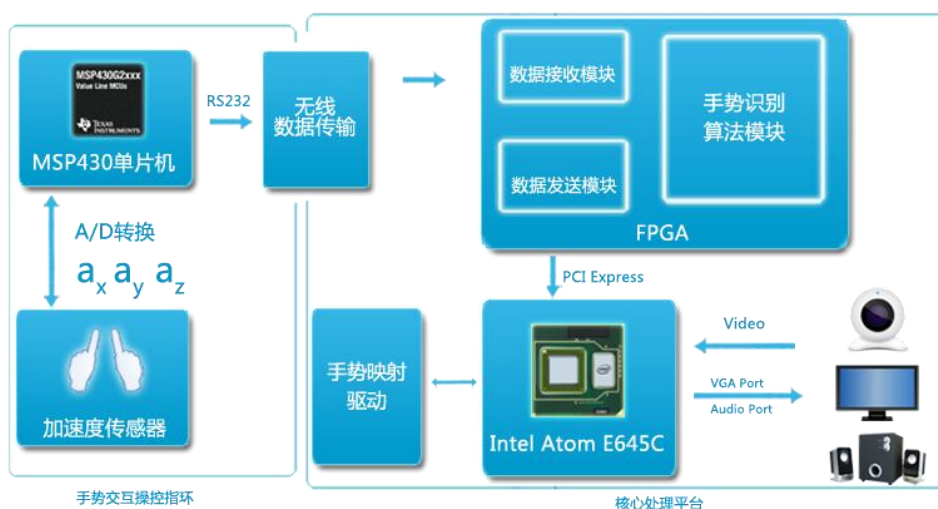


图 2.2 系统硬件结构图

系统的硬件结构主要分为两部分。

第一部分是为我们系统精心设计的的手势交互操控指环，该指环由超低功耗单片机、纽扣电池、无线数据传输模块、加速度传感器等元件组成，主要用于新型的立体操控交互，使用者只需将指环轻轻地套在手指上，便可以完成操控。

第二部分是核心处理平台部分，平台核心处理芯片为 Intel® Atom™ E645C 处理器，具有 1.0Ghz 主频，并支持 Intel® 超线程处理技术，另外，该芯片集成了 Altera FPGA，使得系统的灵活性大大提高，系统也将手势识别的算法集成在 FPGA 的模块之中，由于 FPGA 高性能的并行处理能力，使得识别速度大幅度提高，完全满足系统需求。

2.2.2 Intel® Atom™ 平台

系统使用英特尔凌动 EPC-M-505C 平台作为控制核心并与用户直接交互。系统利用了该平台的丰富资源，将凌动平台卓越的性能发挥的淋漓尽致。值得一提的是，系统发挥了凌动平台的可配置性这一新特性，在可编程逻辑器件内实现了自定义手势识别模块。

(1) 平台的可编程逻辑器件的灵活运用

Atom 平台中集成的 Altera 现场可编程门阵列，提供了足够的可编程空间。依靠这一可定制特性，系统在 FPGA 中，通过控制 SPI 接口对无线数据进行接收，并实现了手势识别算法。FPGA 的强劲的并行处理能力，大大地提高了手势识别算法的速率。

(2) 平台的多媒体性能运用

系统使用基于 OpenGL 的 irrlicht 物理引擎技术实现系统软件的室内设计与游览功能，Atom 平台的 E645C 自带的集成显卡支持 OpenGL 渲染加速。这一良好的支持特性，使得图像处理的能力大大的提高，足以满足系统实时渲染 3D 场景的需求。此外，支持 VGA 输出的 Atom 平台为系统提供了高清信号输出，这使得室内设计时的场景显示更为细腻柔和。

(3) 平台的高速处理能力的运用

Atom 平台使用的 E645C 处理芯片主频为 1.0Ghz，并且拥有 512KB 的二级缓存，支持英特尔超线程技术(Intel® Hyper-Threading Technology),这对于系统在使用八叉树算法加载场景、实时刷新立体场景时，能够更好地利用 CPU 的多线程处理能力，加快并行计算。

(4) 平台的低功耗特性的运用

Atom 平台支持节电技术 (Enhanced Intel SpeedStep Technology)，能够根据不同的系统工作量自动调节电压和频率，以减少耗电量和发热量。这对于长时间运行系统时，系统保持低功耗、稳定等特性有了强力的保障。

(5) 平台丰富的接口资源的利用

系统利用集成的 FPGA，自定制了 RS232 接口，实现了手势交互操控指环与主机的数据通信；系统利用平台提供的 FPGA 上的视频输入端口，实现了摄像头数据的采集；此外，系统使用平台的 SATA 接口拓展了大容量的硬盘，大容量的硬盘为模型、场景等数据库的存储提供了充裕的控件。

2.2.3 手势交互操控指环

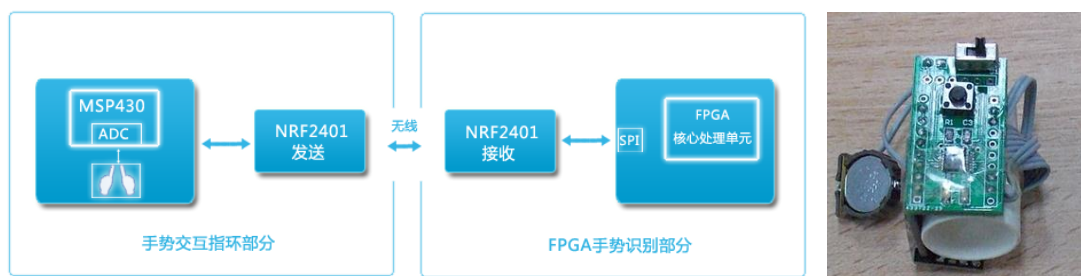


图 2.3 手势交互操控指环结构与实物图

如图 2.3 所示，手势交互操控指环由超低功耗单片机 MSP430G2231 小系统模块、低功耗无线收发模块 NRF2401 和模拟加速度传感器模块三个部分组成。指环将人手部动作加速度变化数据通过 NRF2401 发送到接收端 NRF2401。FPGA 接收模块利用 SPI 接口控制 NRF2401，以达到接收数据的目的，之后再对此数据进一步处理，识别出人的手部动作。

2.2.4 视频采集设备

视频采集设备用来采集现实场景，直接连接到 FPGA 上的视频输入端口之上，之后 FPGA 通过 PCI/E 接口与 CPU 进行数据传输，平台中的 FPGA 高性能的并行处理能力足以满足视频采集的需求，达到了实时的图像输出的目的。

2.3 系统软件设计

2.3.1 软件结构设计概述

系统软件由室内设计中心主界面、外景展示模块、室内设计游览模块、模型管理模块、场景管理模块等模块组成，其中包括了手势识别算法、手势映射驱动、坐标转换算法、碰撞检测算法等。整个软件的设计图如下所示。

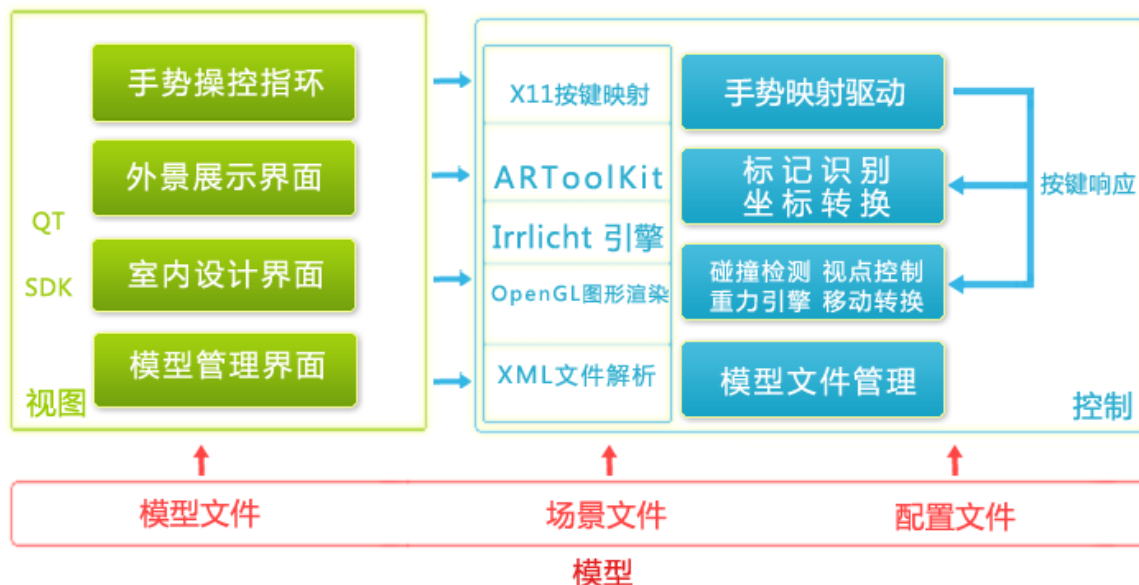


图 2.4 系统软件设计结构图

借鉴 Model View Controller（MVC）这一经典的软件设计模型，完成整个软件的架构。模型(Model)、视图(View)、控制(Controller)各部分紧密结合，M 指的是模型、场景的数据文件、配置文件，V 指的是整个软件的用户界面与交互接口，C 则是用于程序的控制。

2.3.2 系统与软件整体信息

室内交互设计游览系统采用了 Intel 开发的基于 Linux 的 Yocto 系统。经过内核定制与指令集的优化，使得 Yocto 更能发挥 Intel Atom 处理器平台的高性能的优势。

系统使用的操作系统的 Linux 内核版本号为 3.0.18。整个系统使用 Qt 4.7.4 相关 API 开发，软件运行环境为 Qt 4.7.4 Framework。整个系统均使用 C++ 语言编写，具有良好的跨平台性。

系统软件的整体结构图如图 2.5 所示

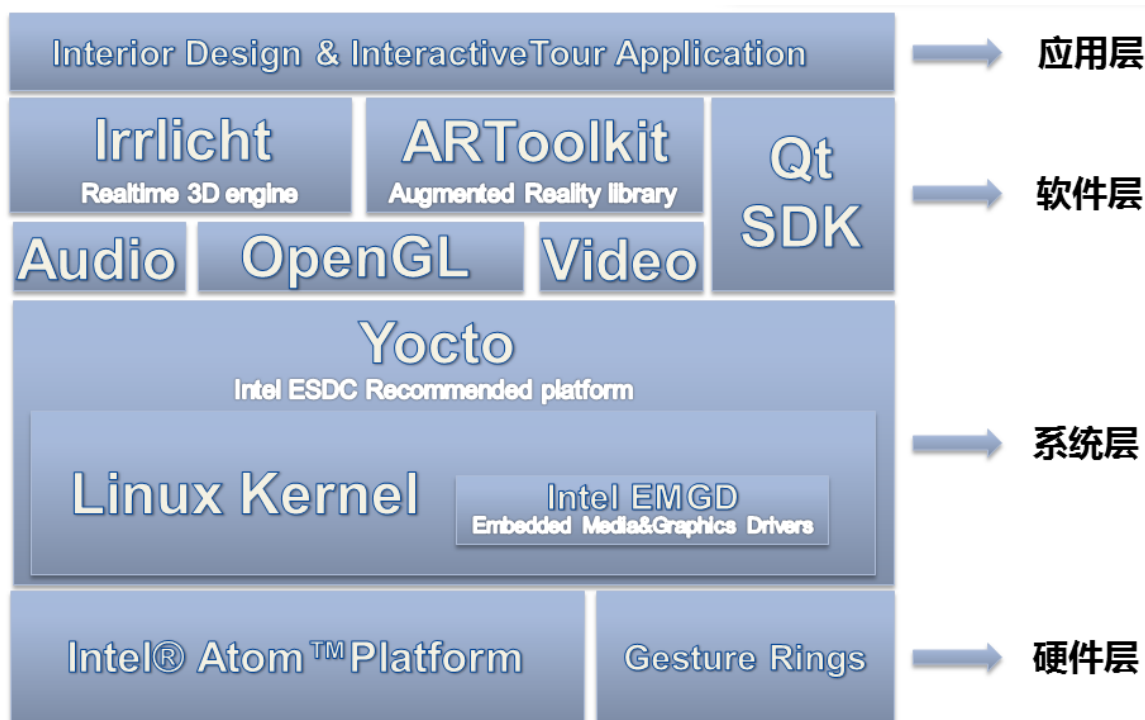


图 2.5 系统软件架构图

Irrlicht 是一个用 C++ 书写的高性能实时 3D 引擎，是一个跨平台的引擎，使用 OpenGL 渲染程序，支持碰撞检测、重力场，这些特性使得系统能够快速地进行场景立体渲染，并保持一定的帧速。

ARToolKit 它是一个 C/C++ 语言编写的库，通过它可以让我们很容易的编写增强现实应用程序，即把电脑虚拟的图像覆盖到真实世界画面中。这一特性为系统的房屋外景展示提供了强力支持。

Yocto 是一个开源社区，它通过提供模版、工具和方法帮助开发者创建基于 linux 内核的定制系统。利用 XLib 结合系统完成手势映射驱动，利用其中的 Xlib API 模拟键盘按键响应。利用手势直接完成系统的控制，使得系统的交互体验性也大大提高。

Intel® EMGD 全称英特尔® 嵌入式媒体和显卡驱动程序，专门满足使用英特尔® 凌动™ 处理器 E6xx 系列的嵌入式平台的需求，正是因为 Intel EMGD 驱动对 OpenGL 的加速支持，使得图形渲染性能大幅度提高；

2.3.3 室内设计中心(主界面)模块

软件主界面使用 Qt 编写，主要是通过 Qt SDK 的相关 API、控件实现软件的界面。利用 Qt 强大的图形表达能力、丰富的 UI 设计效果、以及自定义的动画效果，完成了主界面的框架。同时，我们也精心设计了软件的用户交互界面，增强了用户的使用体验。

室内设计中心（主界面）主要分为四个 Tab。四个 Tab 从左至右依次分别是：

·模型管理模块

包含了场景管理、模型管理功能。可以在这里进行场景、模型的查看、修改、增减。

·设计与展示模块

包含了外景模型展示、室内设计与游览功能。可以在这里进行基于增强现实的外景模型展示，以及进入室内的立体设计与交互游览。

·系统设置模块

包含了串口相关设置、手势数据信息查看等功能。开发者在此可以进行相关调试工作

·系统介绍模块

包含了对系统相关功能的简单介绍。



图 2.6 软件主界面截图

2.3.4 外景模型展示功能

外景模型展示功能利用增强现实技术，借助 ARToolKit 这一开发包，结合 Irrlicht 图像渲染引擎。首先系统对待识别纸张进行识别，识别后进行现实坐标与模型坐标的转化，建立纸张所在的坐标系，将所识别的标记所对应的模型渲染于屏幕中的坐标系之中。效果如图。



图 2.7 外景模型展示程序截图

2.3.5 室内设计游览功能

室内设计游览功能提供了一个室内设计交互设计的 3D 立体界面，利用物理引擎，加入了碰撞检测、重力场，使得整个虚拟的环境与现实环境相差无几，因此，具有良好的用户体验。用户可以通过手势直接操控三维场景中的家具，来布置自己心仪的家居环境。

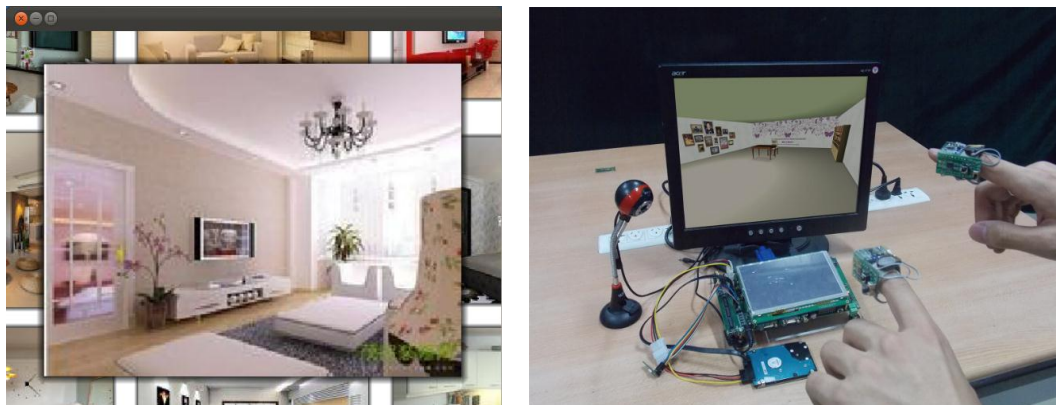


图 2.8 室内选择与设计界面截图

如图所示，用户首先选择需要进入的场景，系统专门设计的 3D 选择界面，增强了用户体验，进入场景之后，用户可以以第一人称视觉在场景里任意穿梭。

同时，可以完成对选中模型的切换修改、旋转方向、改变大小、立体移动等功能。灵活的自定义功能，使得用户可以便捷高效地完成家居环境的布置。

2.3.6 场景管理功能

场景管理功能提供了一个便捷的管理界面，使用户可以轻松修改、场景的模型、贴图。修改之后，将保存为系统的配置文件，供场景读入时使用。

2.3.7 模型管理功能

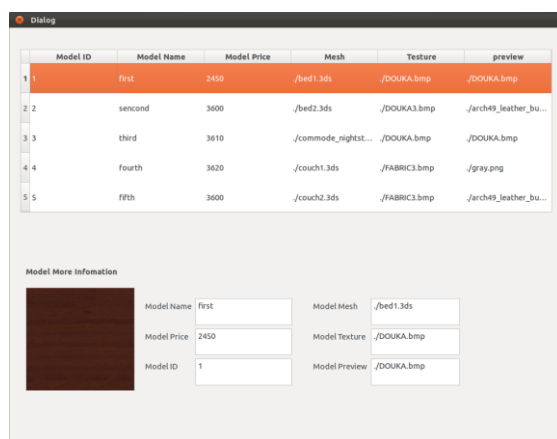


图 2.9 模型管理界面截图

模型管理功能提供了一个高效的管理界面，用户可以轻松地进行模型的管理，方便地修改模型结构、模型贴图、模型名称等相关内容。

三 系统实现原理

3.1 手势交互操控指环的设计

3.1.1 手势识别部分

MCU 模块通过 MSP430G2231 内部 10 位 ADC 对三路模拟加速度电压值采样，得到 X、Y、Z 三路的加速度信息，并通过 SPI 接口控制无线收发模块 NRF2401 将三路电压值发送给接收端 FPGA 控制的 NRF2401。

指环在 X, Y, Z 方向的位移可以由三个方向的加速度进行二次积分求得

$$S = v_0 t + \int_0^t \int_0^t a dt \quad (3-1-1)$$

经过多次测量，求得指环在 X, Y, Z 平面运动位移绝对值的均值为 $\overline{S_x}$ 、 $\overline{S_y}$ 和 $\overline{S_z}$ ，为了三个平面内运动幅度可以对变量进行归一化处理，根据公式

$$\begin{aligned} S'_x &= \overline{S_x} \\ S'_y &= n_y \overline{S_y} \\ S'_z &= n_z \overline{S_z} \end{aligned} \quad (3-1-2)$$

使得 $S'_x = S'_y = S'_z$ 在 FPGA 收到指环三个方向加速度后，通过分析比较 S'_x 、 S'_y 、 S'_z 便可得知指环的运动在哪个坐标轴，进而通过位移的正负判定运动方向。

3.1.2 手势映射驱动部分

在 FPGA 中，通过手势识别算法对手势交互操控指环的加速度进行分析识别后，将识别出的用户双手动作信息传送至平台的处理核心中。系统设计了手势映射驱动，以保证手势对系统进行操控，在映射部分，主要是利用了 XTest (the X test extension) 这个 Xlib 的扩展库，用它可以实现键盘按键的模拟。

具体映射表如下所示

表 3.1 手势按键映射表

手势识别结果	按键映射与 X11 编码		手势识别结果	按键映射与 X11 编码	
q(左手 Z 轴正方向)	XK_q	0x071	a(左手 X 负方向)	XK_a	0x061
z(左手 Z 轴负方向)	XK_z	0x07a	d(左手 X 正方向)	XK_d	0x064
w(左手 Y 轴正方向)	XK_w	0x077	x(左右逆时针旋转)	XK_x	0x078
s(左手 Y 轴正方向)	XK_s	0x073	c(左右顺时针旋转)	XK_c	0x043
+(左右放大动作)	XK_greater	0x03e	-(左右缩小动作)	XK_less	0x03c

当识别到手势动作触发时，做出按键响应，例如，当做出向上的动作时，FPGA 检测出 Z 轴的正向运动，则通过键盘映射产生 q 键的按键响应，通过调用 Xlib 的 API 模拟一次按键按下与释放的过程。

3.2 外景模型展示的设计

3.2.1 标记识别

ARToolKit 通过使用图形标记来确定世界坐标系，标记是由一个正方形的黑色边框和内部的模板图像构成。如图所示：

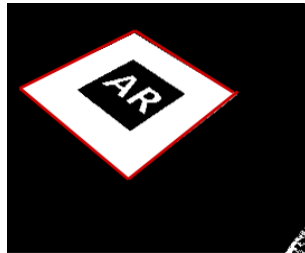


图 3.1 标记模板截图

ARToolKit 识别图形标记的具体流程如下

- 1) 图像处理：根据用户设定的阈值将采集到的一帧彩色图像转换成一幅二值(黑白)图像。
- 2) 标记检测：对该二值图像进行连通域分析，找出其中所有的四边形区域作为候选匹配区域，确定候选区域的四条边（四个角点）的位置。
- 3) 标记校正：将候选区域经过旋转、缩放等步骤校正为标准的方形，从而恢复原来的图形标记的各种几何特征，
- 4) 标记匹配：将每一个校正后的候选区域与模板库中的模板进行匹配。匹配接近的程度可借助向量的范数来描述，如欧几里得距离。设候选区域为合法标记的置信度为 cf ($0 < cf < 1$)，当接近程度大于 cf 时，匹配成功，否则匹配失败。图 3.1 表示了图像标记被识别出来。

3.2.2 坐标系建立与转换

由于需要给用户一个虚拟与现实相结合模型展示，当标记被识别后，应当建立一个坐标系来显示模型，并将实时采集的图像与立体模型叠加起来。这其中利用到了坐标转换的算法，主要步骤如下所示：

- (1) 视频获取初始化以及读入图形标记模板和摄像机参数。
- (2) 摄像机捕获真实世界的视频，并将它传送给计算机。
- (3) 监控视频流中的每一帧图像，并在其中搜索是否有匹配的图形标记。
- (4) 如果找到了，将通过计算机视觉技术计算出图形标记和摄像头的相对位置。
- (5) 得到摄像头的位置之后，结合 Irrlicht 引擎，将模型渲染到视频帧中图形标记所在位置。
- (6) 真实的场景与虚拟的立体模型叠加后的视频流通过与 VGA 相连的显示设备显示出来，使得用户感觉一个真正的房屋模型跃然纸上，实现了增强现实的效果。

在图形标记被识别之后，系统中所使用的各个坐标系之间的关系就确定了。图 3.2 表示了世界坐标系 (X_m, Y_m, Z_m) (即图形标记坐标系) 和摄像机坐标系 (X_c, Y_c, Z_c) 与摄像机屏幕坐标系 (X_i, Y_i) 之间的关系。

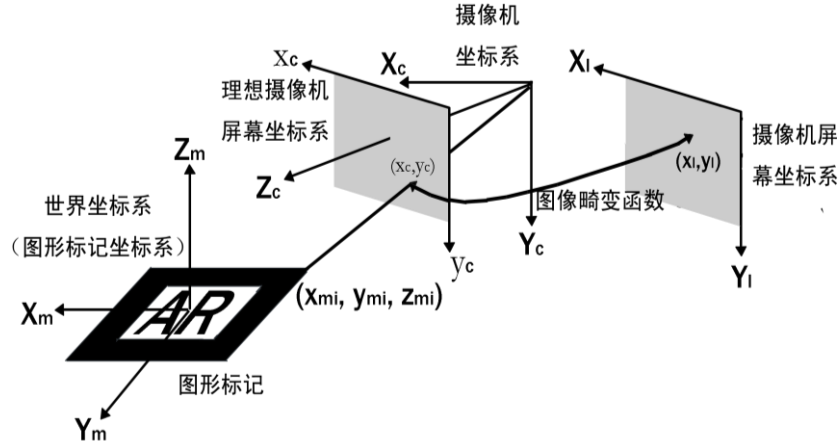


图 3.2 世界坐标、屏幕坐标、摄像机坐标转换

两者之间的关系可以表述为

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T_{CM} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (3-2-1)$$

其中 T_{cm} 是坐标系转化矩阵。 $R_{3 \times 3}$ 是旋转矩阵， $T_{3 \times 1}$ 是平移矩阵

摄像机屏幕坐标系 (X_i, Y_i) 和摄像机坐标系 (X_c, Y_c, Z_c) 之间的关系为

$$\begin{bmatrix} hX_i \\ hY_i \\ h \end{bmatrix} = \begin{bmatrix} sf_x & 0 & x_c & 0 \\ 0 & sf_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3-2-2)$$

式中 C 矩阵为摄像机的参数矩阵。

从而得到摄像机屏幕坐标系 (X_i, Y_i) 图形标记的坐标系 (X_m, Y_m, Z_m) 间的转换关系为

$$\begin{bmatrix} hX_i \\ hY_i \\ h \end{bmatrix} = C \cdot T_{CM} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (3-2-3)$$

由于图形标记的四个角点的坐标均已知，可记为 (X_{mi}, Y_{mi}, Z_{mi}) ， $i=1,2,3,4$ 。设四个角点在摄像机屏幕上的坐标为 (x_i, y_i) ， $i=1,2,3,4$ ，则有

$$\begin{bmatrix} \hat{h} x_i \\ \hat{h} y_i \\ \hat{h} \end{bmatrix} = C \cdot T_{CM} \begin{bmatrix} X_{mi} \\ Y_{mi} \\ Z_{mi} \\ 1 \end{bmatrix} \quad (3-2-4)$$

摄像机参数矩阵 C 可事先求得，通过求解 (\hat{x}_i, \hat{y}_i) 和实际测得的 (x_i, y_i) 之差的目
标函数的最小值，可以最终确定各变量的值，即

$$MINdif = \frac{1}{4} \sum_{i=1,2,3,4} \left\{ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right\} \quad (3-2-5)$$

这样就得到了各坐标系间的转换关系，系统之后便可以据此在各个坐标系之间方便地进行
转换。

3.2.3 模型实时渲染

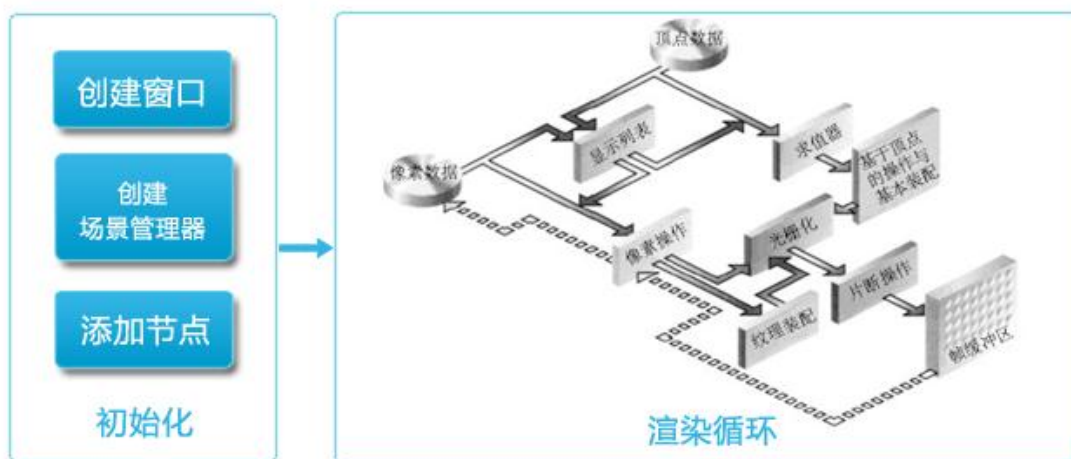


图 3.3 模型实时渲染工作流程图

如图 3.3，为 Irrlicht 引擎利用 OpenGL 对模型进行渲染的操作顺序。可以看到，模型经过
顶点变换、光栅化、纹理装配、片段操作等过程完成模型的实时渲染。而图像引擎封装了
OpenGL，使得开发者不用关心底层的操作，并且提供了方便调用的 API。

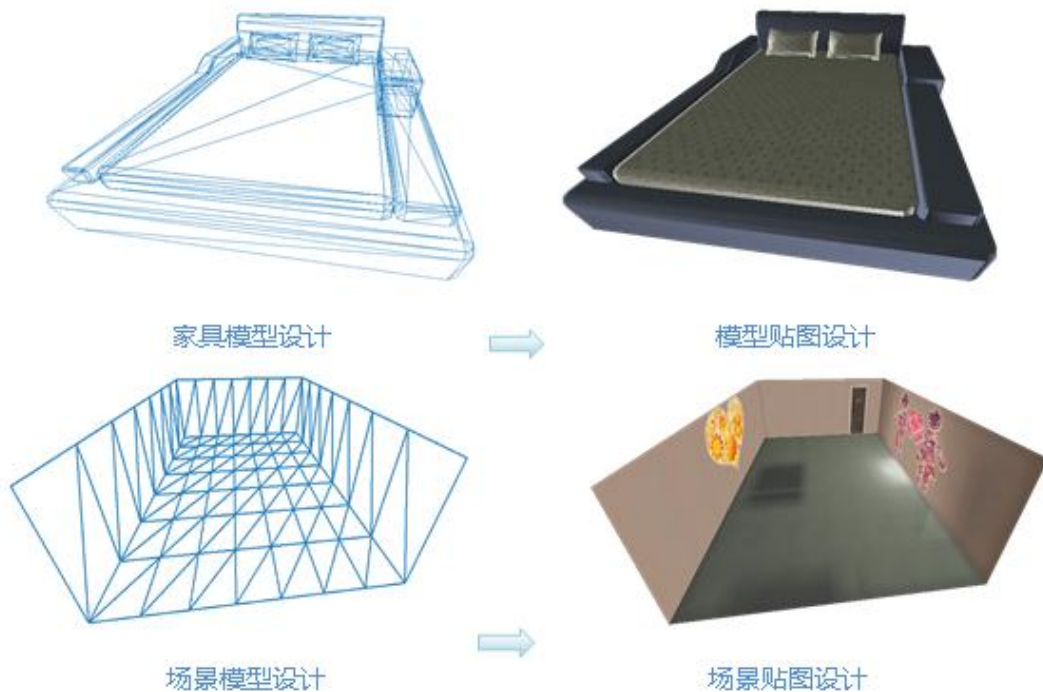
利用图形引擎完成模型的实时渲染的主要步骤为：

- (1) 创建窗口，指定窗口宽、高、是否全屏、色彩位深等相关参数；
- (2) 创建场景管理器，用于管理用户界面、节点、场景、碰撞检测；
- (3) 添加节点、灯光、音乐、用户界面、按键响应到创景管理器中；
- (4) 如果窗口为激活态，进入场景帧刷新循环，直到窗口退出为止；

通过以上步骤，即可完成对模型的实时刷新，进入到最后的循环中，每执行一次循环相
当于系统完成了一帧的刷新，通过计算每秒刷新的帧数便可以得到系统当前的 FPS。

3.3 室内设计游览的设计

3.3.1 场景与模型的设计



场景制作时使用引擎提供的 CopperCube 进行制作。首先设计出场景的网格结构，如图所示；网格结构设计完成之后，为场景添加贴图，使得房屋场景更加真实生动。

模型制作时使用专业的 3D 模型制作软件 3DSMAX 进行设计。与场景制作步骤类似，先设计出模型的网格结构，之后加上模型贴图。**图 3.4 模型与场景模型设计原理**

图像引擎本身 3D 模型所支持文件类型十分丰富，因此，可以选择利用专业的模型设计软件进行设计，这样大大增添了系统的可扩展性。

3.3.2 基于图像引擎的场景渲染

系统利用物理引擎对场景进行管理，主要渲染流程如图 3.5 所示。

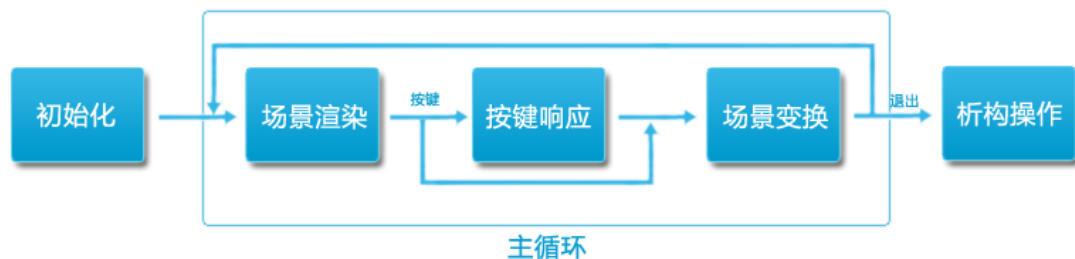


图 3.5 场景渲染流程图

利用 Irrlicht 这一实时渲染引擎，使得场景的管理变得方便，场景中包括 GUI、节点、灯光和按键响应。首先系统执行初始化工作，主要是对 GUI、节点等相关内容进行初始化。之

后，进入渲染的主循环中，如果接到按键消息，则执行相关响应，否则继续刷新场景界面。每执行一次循环便完成一帧的渲染。当接收到关闭响应时，退出循环，执行相关变量的析构。

值得一提的是，系统使用图像引擎中常用的八叉树算法对场景进行加载，只对可见部分进行渲染，这样在降低了系统功耗与渲染时间的同时，使得整体的渲染效率大大提高。

八叉树是一种用于描述三维空间的树状数据结构。八叉树的每个结点表示一个正方体的体积元素，每个结点有八个子结点，将八个子结点所表示的体积元素加在一起就等于父结点的体积。八叉树的定义是：若不为空树的话，树中任一结点的子结点恰好只会有八个，或零个，也就是子结点不会有 0 与 8 以外的数目。

假设要表示的形体 V 可以放在一个充分大的正方体 C 内， C 的边长为 $2n$ ，它的八叉树可以用以下的递归方法来定义：

八叉树的每个结点与 C 的一个子立方体对应，树根与 C 本身相对应，如果 $V=C$ ，那么 V 的八叉树仅有树根，如果 $V \neq C$ ，则将 C 等分为八个子立方体，每个子立方体与树根的一个子结点相对应。只要某个子立方体不是完全空白或完全为 V 所占据，就要被八等分如下图所示，从而对应的结点也就有了八个子结点。这样的递归判断、分割一直要进行到结点所对应的立方体或是完全空白，或是完全为 V 占据，或是其大小已是预先定义的体素大小，并且对它与 V 的交集作一定的“舍入”，使体素或认为是空白的或认为是 V 占据的。

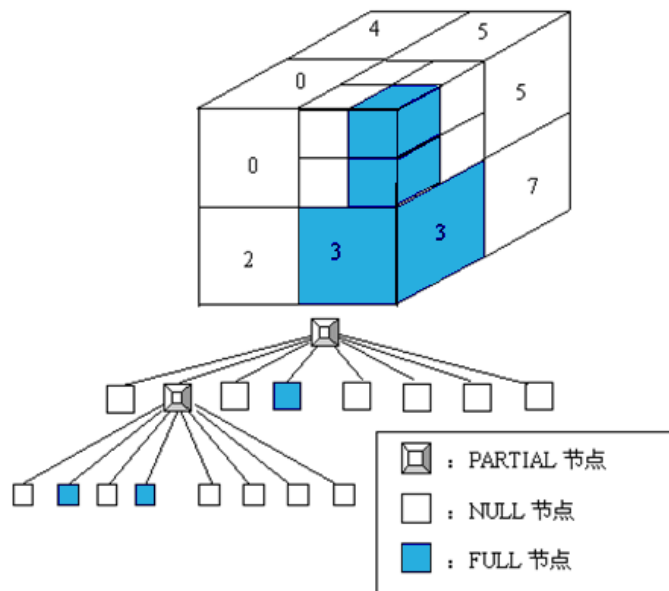


图 3.6 八叉树原理图

如此所生成的八叉树上的结点可分为三类：

第一类是灰结点，它对应的立方体部分地为 V 所占据；

第二类是白结点，它所对应的立方体中无 V 的内容；

第三类是黑结点，它所对应的立方体全为 V 所占据。

后两类又称为叶结点。

形体 V 关于 C 的八叉树的逻辑结构是这样的：它是一颗树，其上的结点要么是叶结点，要么就是有八个子结点的灰结点。根节点与 C 相对应，其它结点与 C 的某个子立方体相对应。八叉树的最大深度表示空间分割所达到的层次，称为空间分辨率，而八叉树的终结结点对应分割后的空间网格单元。由八叉树结点的编码很容易找到空间任一点所在的空间网格单元。

3.3.3 世界与相对坐标的转换

在移动家具节点时，视点的方向具有不确定性，若要产生真实移动的感觉，使得节点的移动不能单纯地减世界坐标，需要对家具节点的坐标进行坐标转换。

用户所习惯的坐标是相对于视点的坐标，需要转换为系统所需要的世界坐标，保证视点的方向无论朝向何方，当节点被移动时，都可以以人的相对坐标移动。这里主要是应用了矩阵的基本变换中的旋转变换，其基本原理是：

将平面上任意一点绕原点旋转 θ 角，一般规定逆时针方向为正，顺时针方向为负。

$$\begin{aligned}x_1 &= r \cdot \cos(\theta + \varphi) \\&= r \cdot (\cos \theta \cdot \cos \varphi - \sin \theta \cdot \sin \varphi) \\&= x \cdot \cos \theta - y \cdot \sin \theta \\y_1 &= r \cdot \sin(\theta + \varphi) \\&= r \cdot (\sin \theta \cdot \cos \varphi + \cos \theta \cdot \sin \varphi) \\&= x \cdot \sin \theta + y \cdot \cos \theta\end{aligned}$$

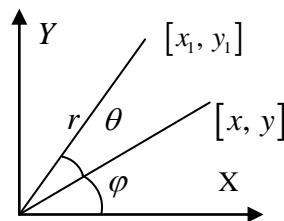


图 3.7 坐标转换原理图

用矩阵运算表示为：

$$P_1 = P \cdot R = \begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

其中 R 称为旋转变换矩阵。

3.3.4 物体之间的碰撞检测

在进行室内设计的家具摆放与移动的过程中，系统特意增加了碰撞检测，使得场景更加接近于现实世界，Irrlicht 物理引擎本身对碰撞检测的支持，使得碰撞检测的实现更加容易。基本步骤为：添加一个节点，为这个节点添加三角选择器，三角选择器描述了节点的立体轮廓，之后遍历除了这个节点之外的所有节点，并将其他节点的碰撞轮廓添加至此节点。这样变完成了碰撞检测。

由于场景地图比较大，需要进行碰撞检测的区域也比较多，故系统使用八叉树选择器 (OctTreeTriangleSelector) 为场景创建选择器，八叉树三角选择器使得三角形数目的输出优化，减少了 3D 场景的平均三角面数，这样做对于大型的地图是很有用的。之后只需为节点创建碰撞检测的响应，物理引擎会自动完成碰撞检测。

如图 3.8，可以看到这些节点碰撞检测的轮廓用白色的线条先标记出来。



图 3.8 碰撞检测效果图

3.3.5 物体的拣取与响应

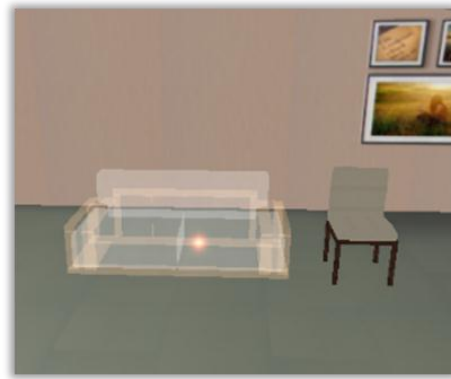
在室内设计模式中，需要对模型进行拣取，并对已选定模型做出诸如大小、位置、方向的调整，系统设计了物体拣取的相关算法。

基本原理是：以 Camera(视点)的中心作为射线的起点，以 Camera 的方向作为射线的方向，做一条射线，该条射线与场景中的节点进行碰撞检测，场景中立体的节点，都由许多小三角组合而成，当射线与节点碰撞检测成功，且节点可被修改时，则该节点被拣取，并将模型设定为选中状态。

如图 3.8(a)所示，为利用射线进行拣取的效果图，我们可以看见当射线与可被修改的节点碰撞成功时，节点被选中的三角形区域被标记为特殊的颜色。



(a) 物体拣取效果图



(b) 物体拣取响应效果图

图 3.8 物体拣取与响应效果图

当模型被选中时，系统利用贴图的变换作出相应的响应，达到增强用户交互体验的目的，实际效果如 3.8(b)所示。即对选定模型的贴图进行混色处理，混色就是将某个像素的颜色和已绘制在屏幕上与其对应的像素颜色相互混合，这借助于颜色的 Alpha 通道的分量值。

基本原理是：图像的 RGB 颜色分量*alpha 值 + 背景的 RGB 颜色分量* (1-alpha)，公式如下：

$$RGBA = (Rs \times Sr + Rd \times Dr, Gs \times Sg + Gd \times Dg, Bs \times Sb + Bd \times Db, As \times Sa + Ad \times Da)$$

式中 s 和 d 分别代表源像素和目标像素，S 和 D 代表相应的混色因子。

四 系统测试与结果分析

4.1 外景模型展示的标志物识别率

为了计算标记探测的准确度，采用内边框长 80mm 的矩形标记，通过改变摄像头和标记的距离和倾角，测试对标记的识别率。

表 4.1 外景模型展示的标志物识别率

距离 角度	15	30	45	60	75	90
10 cm	89%	97%	97%	96%	94%	92%
40 cm	82%	92%	93%	93%	94%	92%
60 cm	80%	89%	90%	91%	91%	91%

分析数据表明，摄像头与标记保持 50cm 以内距离和 30° 到 60° 倾角时识别较为稳定，其中 45° 的倾角识别率最高。

4.2 室内交互设计场景的图形性能

系统在 Intel EPCM-505C 平台上对室内设计场景的图形渲染进行了测试，测试结果如下。

表 4.2 室内交互设计场景的图形性能

三角面数 条件	2750	4926	7102
开启 OpenGL 加速	67 FPS	32 FPS	19 FPS
关闭 OpenGL 加速	19 FPS	13 FPS	8 FPS

根据测试结果可得到，在开启对 OpenGL 的渲染加速后，系统的图像渲染性能大幅度提高，完全满足系统对图形渲染的要求。

4.3 手势交互操控指环的识别率

系统对手势交互操控指环的识别率进行了测试，测试结果如下

表 4.3 手势交互操控指环的识别率

单个动作种类	手势识别率	复合动作种类	识别率
X 正向手势	96%	放大手势	91%
X 负向手势	95%	缩小手势	89%
Y 正向手势	97%	顺时针手势	88%
Y 负向手势	96%	逆时针手势	90%
Z 正向手势	94%	确定手势	92%
Z 负向手势	95%		

根据测试结果可知，动作识别率满足交互操作的需求。

五 系统特色与创新

本系统依托搭载着可深度定制化的 Yocto 系统的 Intel Atom 平台，提出了一种全新的、真实的、自然的人机交互方式，实现了室内的交互设计、游览功能，同时也提供了便捷的模型、场景管理模块。充分利用了系统所提供的丰富接口，将系统的性能发挥的淋漓尽致。

室内交互设计系统具有以下特色与创新点：

- (1) **精巧的手势交互操控指环。**提出了一种全新的人机交互方式，系统精心设计的指环，在外型上与一般平面的结构不同，而是打破常规，采用一种立体的电路结构，减小了物理结构的大小，使得空间利用率大大提高。
- (2) **增强现实的模型展示特性。**实现了一种极具展示效果的模型展示，结合了传统纸质宣传材料与现代的计算机视觉技术，利用纸质的材料作为一把钥匙，开启丰富的多媒体资源，使得使用者有全新的视觉体验
- (3) **高度互动的设计游览体验。**实现了一种高度沉浸式的设计体验系统，使用户直观的设计自己的家居环境，自由的穿梭在自己设计的场景中，即使没有设计基础的用户也可以马上学会立体化的室内设计。
- (4) **低功耗下优良的图形性能。**系统使用平台 Atom 处理器采用低功耗设计，使得系统的功耗很低。该系统充分利用了 Atom 芯片中的集成显卡，其支持 OpenGL 优化的特性也将系统的图形性能发挥的淋漓尽致。
- (5) **方便快捷的模型场景管理。**系统为管理者设计的模型与场景管理功能，管理者可以很方便的的对系统进行修改、定制。也可以根据用户需要添加更多丰富的模型，同时，管理者可以高效可视化的对模型进行管理。
- (6) **良好的可移植性和扩展性。**整个系统使用 C++编写，物理引擎与界面的跨平台性使得系统的可移植性很强。同时系统支持常用 3D 模型格式，可方便地进行自定义定制与导入。

整个系统将现代视觉技术、增强现实技术、现代电子技术结合起来，实现了一个室内交互设计的系统，同时，达到了预期的设计要求，完成了所有的设计目标。



图 5.1 实际效果展示图

参考文献

- [1] 李军,徐波,等. OpenGL 编程指南[M]. 北京:机械工业出版社,2010:01.
- [2] (加)Jasmin Blanchette (英)Mark Summerfield 等.C++ GUI Programming with Qt 4 (2nd Edition) [M]. 电子工业出版社,2008:8-29.
- [3] Aung Sithu Kyaw, Johannes Stein.Irrlicht. 1.7 Realtime 3D Engine Beginner's Guide [R]. 2011:08.
- [4] 谢楷,赵建. MSP430 系列单片机系统工程设计与实践[M]. 北京:机械工业出版社,2009:07.
- [5] 王金明. 数字系统设计与 Verilog HDL (第三版) [M]. 北京:机械工业出版社,2009:01.

附录

- 附录一 手势交互指环电路原理图
- 附录二 手势操控交互指环 PCB 图
- 附录三 FPGA 手势识别部分 Quartus 工程顶层 bdf 文件截图
- 附录四 室内设计中心软件核心代码
- 附录五 手势操控交互指环核心代码
- 附录六 FPGA 手势识别部分代码