

天津大学

软件工程综合实践结课报告

基于红外热像传感器的WEB开发



学 院	智能与计算学部
专 业	软件工程
年 级	2019级
姓 名	刘格伦 郑睿恺 周子涵 樊振宇 李世军
学 号	3019244337 3019244342 3019244335 3019244338 3019244341

2021 年 9 月 9 日

目 录

第一章	引言	1
1.1	背景	1
1.2	开发目的	1
第二章	项目需求	2
2.1	硬件需求	2
2.2	服务器端需求	2
2.3	前端需求	2
2.4	系统运行环境	2
第三章	项目设计与实现	3
3.1	总体设计	3
3.2	硬件系统设计	3
3.3	后端服务器设计	4
3.4	前端页面设计	5
3.5	数据库设计	5
第四章	测试与开发过程	6
4.1	传播延时方面	6
4.2	WiFi传输方案的选择	6
4.3	插值算法的选择	6
4.4	温度区间色彩	6
4.5	图片的产生和输出	7
4.6	服务器的部署	7

第五章	团队成员分工描述	8
第六章	异构计算的uid及实验截图	9

第一章 引言

1.1 背景

2019年12月以来，湖北省武汉市部分医院陆续发现了多例有华南海鲜市场暴露史的不明原因肺炎病例，现已证实为2019新型冠状病毒感染引起的急性呼吸道传染病。新型冠状病毒感染，初期症状可能和感冒差不多，可能有咳嗽，咳痰，发热，胸闷呼吸道症状。可能有腹痛，腹泻消化道症状。可能有乏力，四肢酸痛全身症状。可能有眼红，流泪眼部症状。因此，为了更好的防控防治，无论是个人及家庭，或是社区、车站、医院等地方都急需测温来筛查发热病人，因此测温仪需求量大增。

红外测温仪由光学系统、光电探测器、信号放大器及信号处理、显示输出等部分组成。光学系统汇聚其视场内的目标红外辐射能量，视场的大小由测温仪的光学零件及其位置确定。红外能量聚焦在光电探测器上并转变为相应的电信号。该信号经过放大器和信号处理电路，并按照仪器内存的算法和目标发射率校正后转变为被测目标的温度值。与传统接触式温度计相比，红外线测温仪有着响应时间快、非接触、准确测量、使用安全及使用寿命长等优点，结合成熟的软件技术，更能使其发挥传统测温仪难以实现的优越性能。

目前应用较为广泛的主要是单点式的非接触红外测温系统，虽然能满足一般需求，但由于没有一个对人体温度的全面感知，存在漏检问题。红外热像仪是红外测温仪的升级版本，它可以将人体表面的温度发布用彩色图像的形式输出到显示器或屏幕上，让我们可以直接“看见”温度，不同的色彩代表着不同的温度，温度高低一目了然。红外成像仪适用于人流量很大的公共场合的人群排查，在保证更高精度的同时，也提高了测温的效率。本课题提出使用MLX90640红外热像传感器的热像体温测量方案。

1.2 开发目的

本项目主要目标为：以MLX90640传感器和ESP32完成红外数据的采集，在Web端实时显示热成像与关键数据，同时可以对出入人员的信息进行登记与查询。适合放置于商场入口、公司大门等人流量多，对测温需求大的地点。

第二章 项目需求

2.1 硬件需求

- 实现ESP32和红外传感器的数据采集方案
- 实现利用ESP32的WiFi模块将采集得到的温度数据发送到服务器

2.2 服务器端需求

- 通过socket接收ESP32传输的数据
- 使用插值算法将32*24的图片变成512*384像素
- 实现红外温度到可视颜色的伪彩色编码转换
- 绘制热成像图片，提取关键信息
- 对数据库进行操作

2.3 前端需求

- 实时显示生成的热成像图片
- 实现登录、信息登记、信息查询等数据库操作
- 美观的界面

2.4 系统运行环境

- 开发工具: IntelliJ IDEA , Arduino
- 前端: Html + JSP + CSS
- 后端: Java + Servlet
- 服务器: 阿里云 + Tomcat

第三章 项目设计与实现

3.1 总体设计

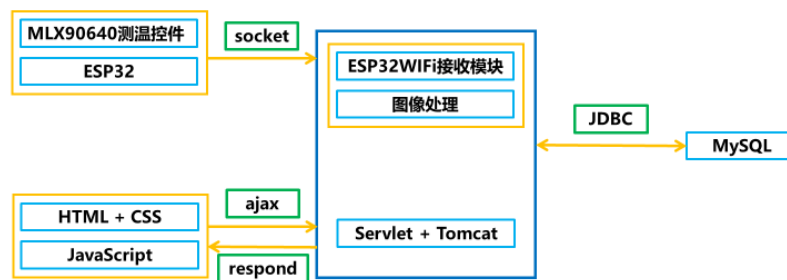


图 3-1 总体设计

3.2 硬件系统设计

硬件系统由MLX90640传感器和ESP32组成，开发环境为arduino。

主要分为两个部分来实现，首先是对传感器的数据进行采集。这一部分通过在网上收集资料，按照说明引入`MLX90640_API.cpp`，`MLX90640_I2C_driver.cpp`中的函数来实现。

```
1 // 读取全部校正参数
2
3 int MLX90640_DumpEE( uint8_t slaveAddr , uint16_t *eeData )
4
5 // 解析校正参数为计算参数
6
7 int MLX90640_ExtractParameters( uint16_t * eeData ,
8     paramsMLX90640 *mlx90640 )
9
10 // 读取一帧计算用实时数据
11
12 int MLX90640_GetFrameData( uint8_t slaveAddr , uint16_t *
```

```

    frameData)
12
13 // 得到外壳温度
14
15 float MLX90640_GetTa(uint16_t *frameData, const
    paramsMLX90640 *params)
16
17 // 返回电压值VDD
18
19 float MLX90640_GetVdd(uint16_t *frameData, const
    paramsMLX90640 *params)
20
21 // 返回绝对温度
22
23 void MLX90640_CalculateTo(uint16_t *frameData, const
    paramsMLX90640 *params, float emissivity, float tr,
    float *result)

```

另一部分的内容就是使用WiFi将获取到的数据发送出去。这里用到了ESP32的WiFi库，使用WiFi.begin()方法，传入WiFi名称和密码就可以连接。

对于需要更换网络环境这一问题，改进了WiFi的连接。启动时首先尝试连接上一次连接的WiFi，在一定时间内如果没有连接成功，则转为热点模式，此时可以通过手机连接热点，访问ESP32开设的小服务器，输入WiFi名称和密码，使其能够连接上新的WiFi。

发送数据用到的是WiFiClient，将得到的浮点数数组*100转为整形，拼接为字符串后后调用WiFiClient.print()方法即可将其发送出去。

3.3 后端服务器设计

后端服务器主要由两部分构成。第一部分是利用Java的socket通信接收ESP32通过WiFi发送的数据，将其读入float数组以待后续使用。Insertdata类实现了二元和三元插值算法，将32*24像素的图像扩大至512*384像素并且满足图像及其轮廓符合人眼的生理观感。PixelConversion类实现了温度和可视颜色的伪彩编码转换，其中我们调整了温度在色条上的分布，使得其更加满足人体测温的需要，并且使得图像更加美观Draw类实现了图像的绘制。将图像转换为Base64编码，使其能够在Web上显示。

另一部分利用了Servlet来实现，Servlet可以实现处理前端发送的http请求，生成动态的Web页面，从而能够将生成的图片的Base64编码持续发送给前端接

收，实现热成像图片的实时显示。

3.4 前端页面设计

3.5 数据库设计

表 3-1 admin表

变量	<i>id</i>	<i>name</i>	<i>account</i>	<i>password</i>	<i>tel</i>	<i>email</i>
类型	int	varchar	varchar	varchar	varchar	varchar

表 3-2 peoplelist表

变量	<i>id</i>	<i>address</i>	<i>name</i>	<i>tel</i>	<i>id_num</i>	<i>tem</i>	<i>date</i>
类型	int	varchar	varchar	varchar	varchar	varchar	date

第四章 测试与开发过程

4.1 传播延时方面

- 最初方案：对float数组逐个传输，传完一次数据约耗时2s
- 改进方案：将float数组拼接为字符串，传完一次数据约耗时0.6s
- 尝试：使用udp进行传输，受包的大小限制，且不可靠传输导致顺序的错乱，放弃
- 最终方案：在ESP32上对float做乘100后类型转换为int，而后拼为字符串进行传输，降低了字符串的长度，传完一组数据约耗时0.3s

4.2 WiFi传输方案的选择

- 最初方案：使用ESP32开设热点，电脑连接进行数据发送。缺点：电脑不能上网
- 改进方案：使ESP32和电脑连接同一个局域网。缺点：每次更换网络环境，需要重新烧写。
- 第二次改进：先让ESP32开设热点，作为服务器提供一个html网页来输入网络名称和密码。缺点：每次启动都需要输入一次。
- 最终方案：保存上次连接的WiFi，启动后先尝试连接上次连接的WiFi，不成功再开启热点连接新的网络。

4.3 插值算法的选择

- 最初方案：简单的取平均值算法，精度不高，生成的图像比较模糊，细节缺失较多。
- 最终方案：实现了二元和三元插值算法，由于两种算法得到的图像差距较小，且二次插值算法在时间复杂度上有较大的优势，因此选择了二次插值算法。

4.4 温度区间色彩

- 最初方案：简单的让温度在色彩条带上均匀分布，由于温度跨度较大，而实际测温时的温度范围比较小，使得人和环境难以区分。
- 尝试：将温度低于一个值的颜色置为黑/白色，细节缺失较多，观感也不佳。
- 最终方案：根据实际情况，将人体体温附近的温度在色彩区间上占比增加，调整了其他温度区间的色彩区间占比，显著提升了成像的质量。

4.5 图片的产生和输出

- 最初方案：先在本地生成图片，再对本地生成的图片转为Base64编码。由于刷新速率快，每次运行产生大量图片。
- 最终方案：将生成的图片赋给一个image对象，直接对其进行Base64编码的转换。

4.6 服务器的部署

- 最初方案：服务器部署于本机。缺点：必须通过局域网才能访问，硬件部分每次重新启动服务器也需要重新启动。
- 改进方案：将服务器部署在阿里云上，使得可以通过公网ip访问。
- 最终方案：设立两个服务器轮流运行，每次重启时连接另一个服务器并关闭前一个服务器进程，使得服务器可以一直运行，随用随连。

4.7 前端页面的改进

- 使得Web适配了电脑、手机、平板，真正实现了多端访问的需求。

第五章 团队成员分工描述

刘格伦

- 实现红外温度到可视颜色的伪彩色编码转换
- 建立服务器，对收到的数据进行处理
- 实现热成像图的绘制
- 实现前后端的通信
- 前端页面的美化

郑睿恺

- 实现ESP32和红外传感器的数据采集方案
- 实现和改进ESP32的WiFi通信
- 实现二元和三元插值算法
- LaTeX文档的编写

樊振宇

- 数据库的建立，sql语句的编写
- 协助完成LaTeX文档的编写

周子涵

- 协助完成前端页面
- 协助完成LaTeX文档的编写

李世军

- 英文文献的翻译
- 协助完成LaTeX文档的编写

第六章 异构计算的uid及实验截图

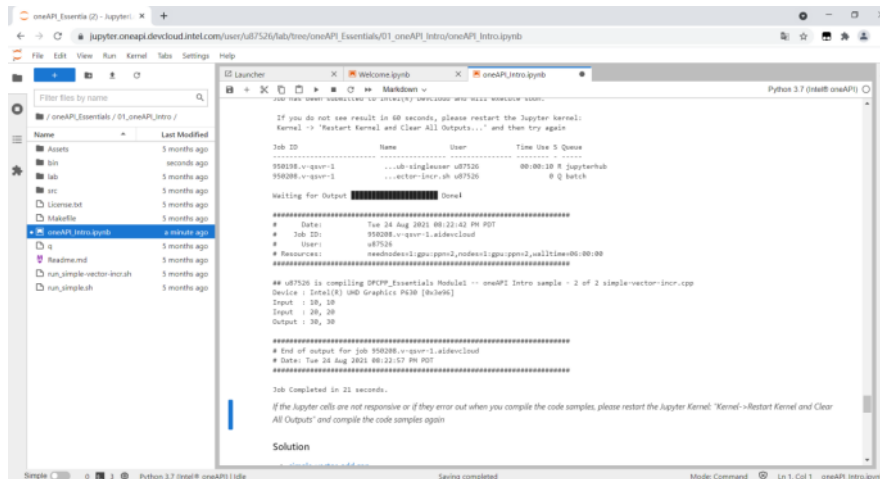


图 6-1 郑睿恺截图uid=u87526

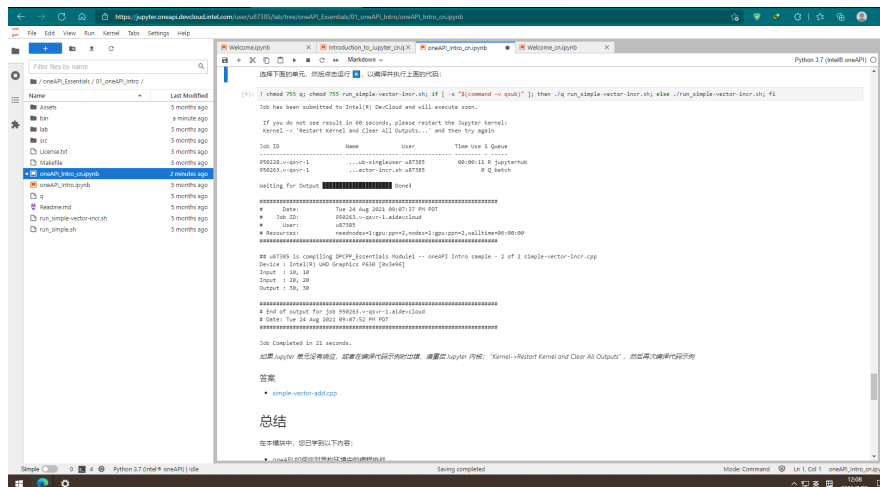


图 6-2 刘格伦截图uid=u87385

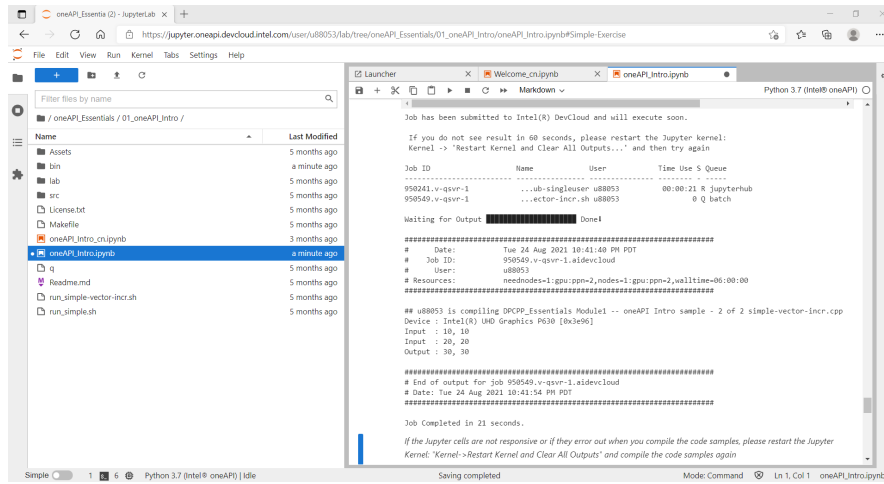


图 6-3 樊振宇截图uid=u88053

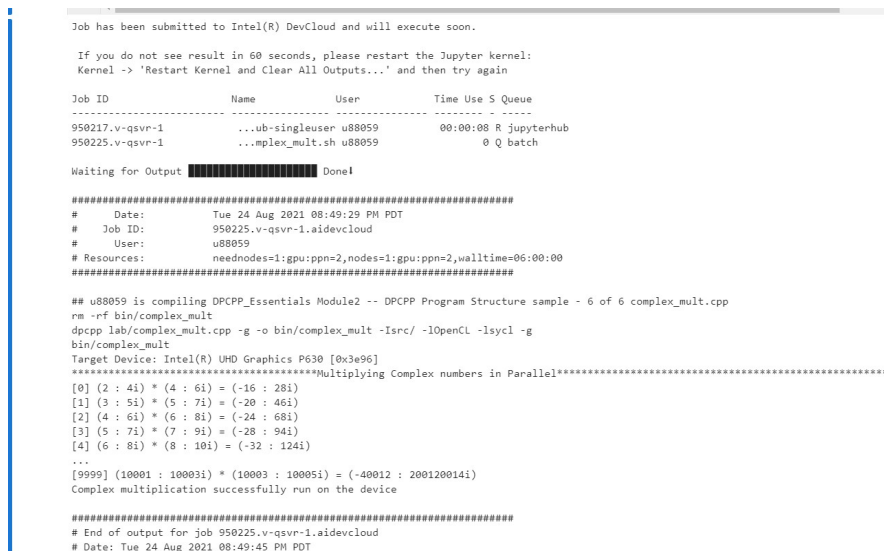


图 6-4 李世军截图uid=u88059

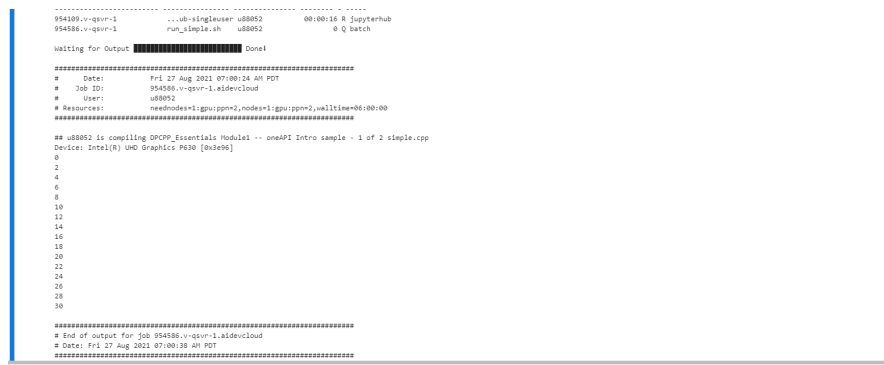


图 6-5 周子涵截图uid=88052