

基于 VR 简笔画的模型检索
中期答辩

罗宇辰 516030910101
陈志扬 516030910347
陈 诺 516030910199

May 6, 2019

Contents

| | | |
|-------|----------------------|----|
| 1 | 课程设计题目 | 2 |
| 2 | 项目目标与创新简述 | 2 |
| 2.1 | 项目目标 | 2 |
| 2.2 | 创新简述 | 2 |
| 2.2.1 | 以 VR 信息为输入的基于内容的模型检索 | 2 |
| 2.2.2 | VR 下的交互与特征提取 | 2 |
| 3 | HCI 思想 | 3 |
| 4 | 设计思路与技术 | 4 |
| 4.1 | 设计思路 | 4 |
| 4.1.1 | VR 模块 | 4 |
| 4.1.2 | 模型检索模块 | 5 |
| 4.2 | 主要技术 | 6 |
| 4.2.1 | VR 模块 | 6 |
| 4.2.2 | 模型检索模块 | 9 |
| 5 | 工作内容 | 15 |
| 5.1 | 已完成工作内容 | 15 |
| 5.2 | 分工完成情况 | 15 |
| 5.3 | 后续工作内容 | 15 |

1 课程设计题目

本小组的设计题目是 基于 VR 简笔画的模型检索。

我们希望能够研发一个通过 VR 简笔画输入来进行模型检索来获得模型的系统，并且将这个系统作为一个子模块加入中学生 VR 实验系统中，为中学生提供一个更便利的获得模型的交互方式。

2 项目目标与创新简述

2.1 项目目标

我们希望能够研发一个 VR 交互友好、模型检索高效准确的系统。

通过这个系统，我们希望能够改善学生在 VR 中做实验的交互方式，并且降低实验设计者的工作量。

2.2 创新简述

创新点主要有两部分：

1. 将 VR 信息作为输入的基于内容的模型检索
2. VR 下的交互与特征提取

2.2.1 以 VR 信息为输入的基于内容的模型检索

常见的模型检索方法分为基于语义的检索和基于内容的检索。

基于语义的检索显然不适合 VR 下的交互，因为 VR 中大多数操作都以 Touch, Grab 来完成，所以在 VR 环境下输入文字是一件很费时费力的事情。

而基于内容的交互中比较传统的方法是基于图片的搜索。例如百度就推出了百度搜图的功能。在 VR 环境下，得到承载检索信息载体的内容确实一件不太容易的事情——让用户在 VR 中挑图片未免太不方便。因此我们想直接使用 VR 中用户能很方便使用 Touch, Grab 等操作的特点，直接让用户在 VR 环境下使用较少的步骤画出简笔画作为检索需要的信息。

2.2.2 VR 下的交互与特征提取

由于交互方式为 VR，因此我们需要实现 VR 下的交互与特征提取。

处于直接在三维中绘图会有出界、不同视图会变得奇怪的问题，我们选择在 VR 中为用户提供一块黑板来作为交互方式。用户通过 Touch 来在 VR 环境下绘制二维图像。VR 端再将包含有用户输入信息的图像发送给模型检索端。

模型检索端获得用户输入后，需要先预处理用户的输入，再将处理过的图片传递给特征提取模块。

3 HCI 思想

1. 贴近用户习惯

从用户的角度出发，考虑方便性：开题时考虑到在三维空间中画三维图更方便，于是设定了画三维图像 → 特征提取 → 转化为二维图像 → 匹配识别的流程。但是在实际的实践中，我们发现了一些问题。

对于普通的中学生和老师，在未经训练的情况下，更习惯于在二维平面作画，也就是所谓的纸笔作画。三维空间中的凭空绘制虽然也并不难以学习，但是其稳定程度、作画的舒适感相对来说有些不符合人体工程学。

我们发现黑板这样一个在三维空间中提供作画平台的工具在中学教室里非常常见，几乎每个人都有机会接触，也有过接触经验。在课程中我们讲到过人的记忆仓库会对一些长期存在的记忆有偏好，一些长期记忆会渐渐规则化、潜意识化。我们人机界面的设计原则中很重要的一点就是要减少学习适应的信息总量，使用用户更为熟悉的固有记忆。我们发现黑板作画放在我们的 VR 环境中会非常容易接受。

2. 按键位置

设计在黑板右侧，符合大多数人的习惯。

3. 图标

图标图案选取是大家比较熟悉且易懂的标识符

4 设计思路与技术

系统整体架构如图一所示，主要分为两个模块：VR 模块和模型检索模块。

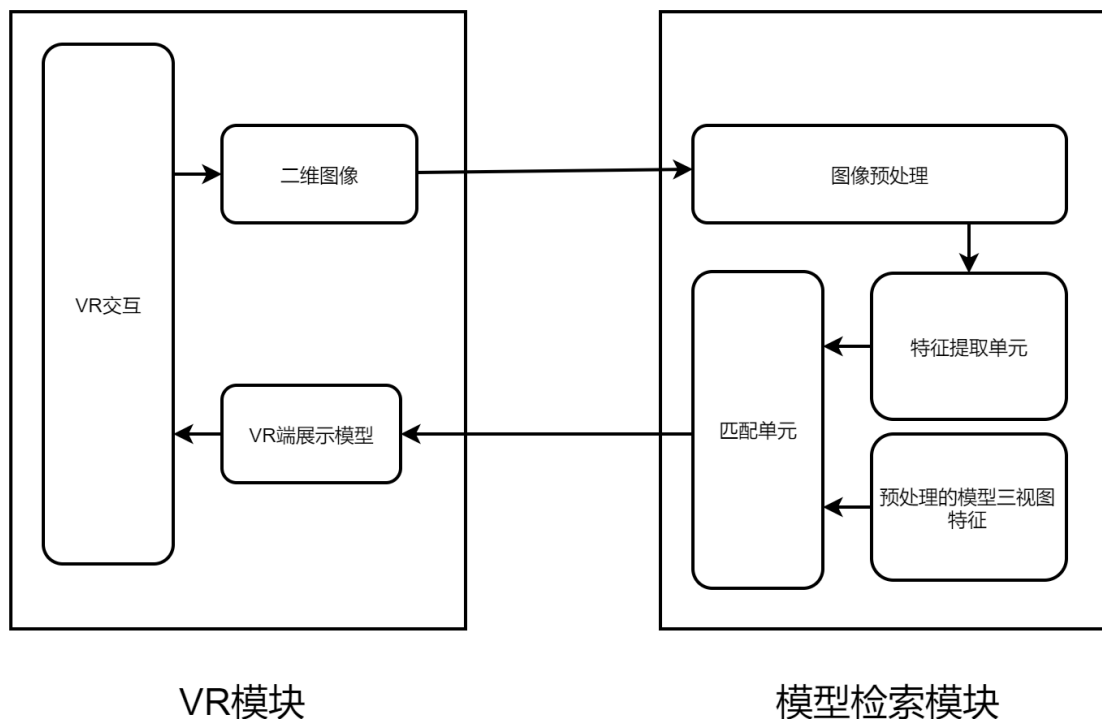


Figure 1: 系统架构

4.1 设计思路

4.1.1 VR 模块

设计考虑与调整：

在开题报告中我们考虑到，用户在三维空间中更方便进行三维的绘图，于是决定使用 tilt brush 在 unity 中进行开发。但在实践阶段遇到了两个问题：

1. tilt brush 整合

google tilt brush 的相关教程较少。google tilt brush 更多是一个开发好了的绘图工具，与 unity 整合在一起获取画出的模型资源有难度。

2. 三维绘图的误差

我们在开题阶段就注意到，三维模型的绘制会出现从不同视角看过去差别较大的情况。考虑到大部分未经过训练的用户更习惯于普通的在画纸上进行二维创作的习惯，在实践中也会习

惯在三维空间里画二维图像（也就是物体的主视图）。而此时二维图像的线段容易不在一个平面内，考虑过的多视图加权计算思路操作起来并不容易。

综合以上考虑，为了在中期能够收获一个实验结果，在 VR 端的设计上做出了改动：在空间中固定一个画板，为用户提供一个可参考的二维平面，然后提供画笔让用户进行绘画。

4.1.2 模型检索模块

模型检索模块分为如下几个部分：

- 图像预处理模块
- 特征提取模块
- 预处理的模型三视图特征模块
- 匹配单元模块

1. 图像预处理模块

图像预处理主要是将 VR 端输入的信息进行二次处理。

用户在进行操作时很可能会将多余的信息引入图片中，例如不小心点上的点、断断续续的线、歪歪扭扭的直线等。为了降低此类因素给特征提取单元带来的误差，所以我们在 VR 输入和特征提取单元之间加入了图片预处理模块。

2. 特征提取模块

特征提取模块主要目的是提取出 VR 输入信息中的特征。

此模块可以复用从模型三视图中提取特征模块的工程代码。主要目的是找出 VR 端输入信息中的特征，并以此为基础来匹配出和输入特征重合度较高的模型。

3. 预处理的模型三视图特征模块

预处理模型三视图部分存储了已有模型三视图对应的特征值。

通过直接使用预处理的模型三视图特征，能够大大降低模型检索耗费的时间。

4. 匹配模块

匹配单元主要通过 VR 信息提取出的特征值与预处理的模型三视图的特征值来匹配出合适的模型，并返回给 VR 端。

4.2 主要技术

4.2.1 VR 模块

1. 场景设计

首先定义了一个黑板（在场景中建一个 Quad）和许多画笔。

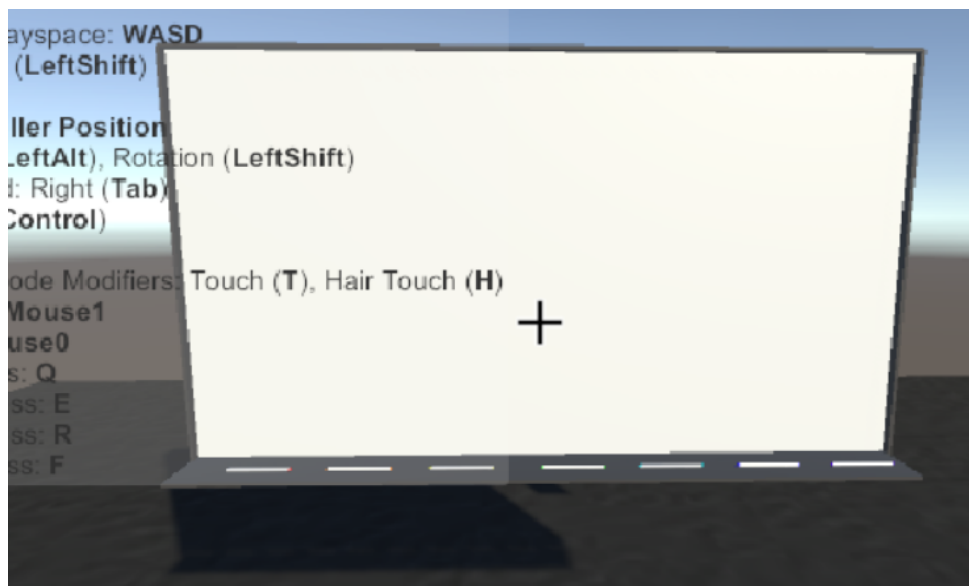


Figure 2: 画板

画笔的设计上有些复杂，分为笔的头、尾（控制握笔的方向）还有笔尖（用来绘画）。在画笔触碰到黑板时触发函数记录画笔此刻的位置并将此处的像素点设置颜色，然后实时更新黑板面的 texture 来显示绘画图案。

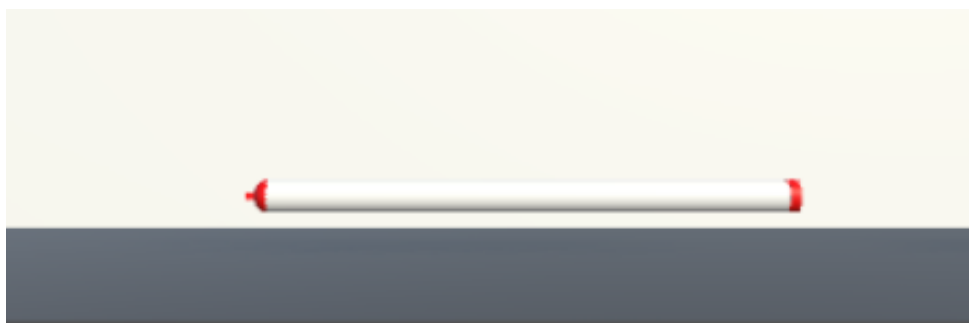


Figure 3: 画笔

在黑板旁边设置两个新的碰撞体，一个用来触发提交功能，将此刻的 texture 转换成 PNG 形式保存提交。另一个触发 load 模型的功能，将检索系统返回的模型信息从模型库中找到并显示在画面中。

2. 功能实现



Figure 4: 两个处理按钮

1. 关于像素点绘制:

运用 Unity Texture2D 类中的两个函数:

```
1 public void SetPixels32(int x, int y, int blockWidth, int blockHeight, Color32[] colors, int  
miplevel = 0);
```

SetPixel

前面 4 个参数相当于一个矩形, x 和 y 就是矩形的左下角的那个点, $blockWidth$ 和 $blockHeight$ 分别是矩形的宽和高, 这个矩形所代表的范围就是 $blockWidth * blockHeight$ 个像素所在的位置, 不妨称这个矩形范围为一个色块;

$colors$ 这个参数的大小必须等于 $blockWidth * blockHeight$, 因为这个方法就是给坐标 (x, y) 开始, 从左到右, 从下到上, 一行一行的对矩形范围内的每个像素赋值; 也就是把 $colors[0]$ $colors[blockWidth - 1]$ 分别赋值到坐标为 (x, y) $(x + blockWidth, y)$ 的像素, 以此类推;

```
1 public void Apply(bool updateMipmaps = true, bool makeNoLongerReadable = false);
```

Apply

当对图片改动完成以后, 需要调用这个方法, 才能让改动真正的应用在图片上;

2. 关于绘画信息传输:

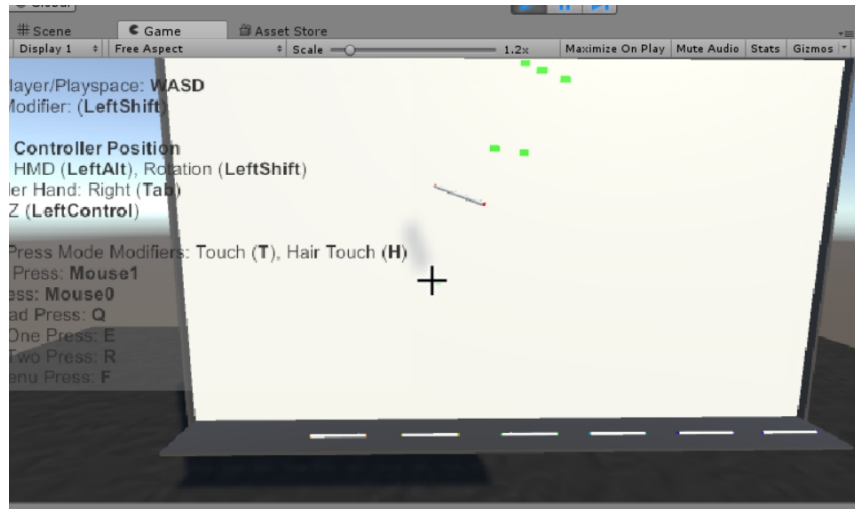


Figure 5: 执笔绘画场景

当触碰到信息传输的碰撞体时，会将此刻的黑板画面由 Texture2D 格式转成 PNG 格式保存下来，用以比对检测。

3. 关于模型的展示:

当触碰到展示模型的碰撞体时，会在场景中显示 prefab 库中对应名称（名称信息由检索系统返回）的模型，展示在场景中。

4.2.2 模型检索模块

图片预处理模块的架构如图所示

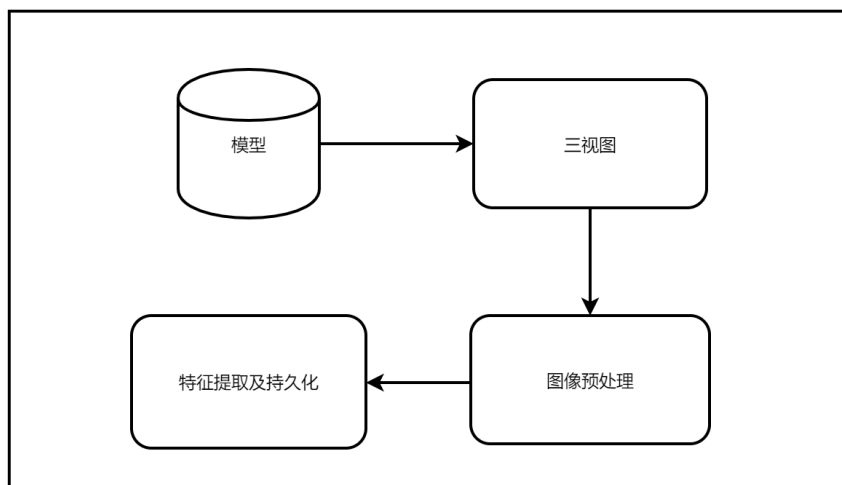


Figure 6: 图片预处理模块

1. 模型三视图渲染及预处理

我们使用了 Blender 场景和 Python 脚本、MATLAB 脚本来得到三视图。

(a) Blender 场景

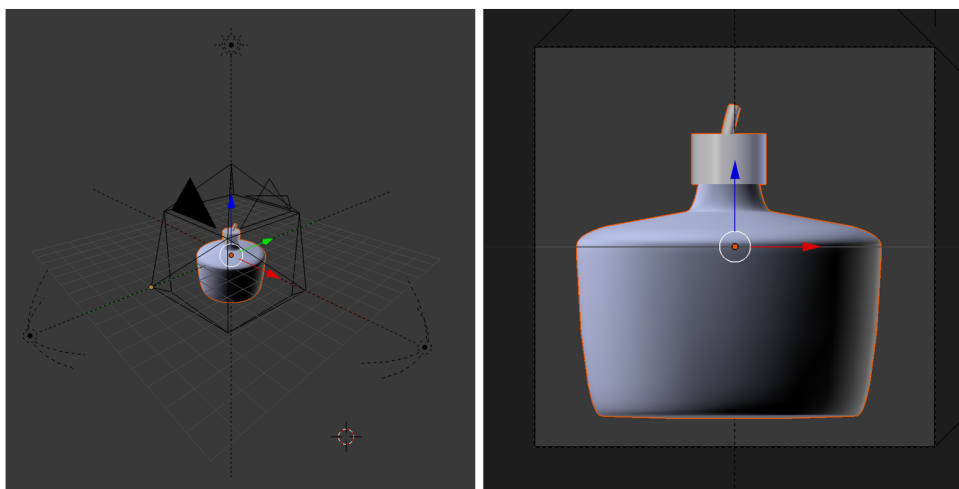


Figure 7: Blender 场景

在场景中设置了 3 个 camera，分别拍摄模型的正视图、侧视图和俯视图。
由于模型的部分材质是玻璃或者水这样的透明物资，所以我们对模型进行了一些调整，

改变了部分 Material 的高光、透明度、颜色等参数。同时为了渲染出位置、大小合适的图像，对模型的重心和尺寸也进行了调整。

(b) Python 脚本

使用 Python 脚本来打开 Blender 场景文件，load 模型，设置渲染并且最终输出三视图图片。

(c) MATLAB 脚本

调用 Python 脚本对模型进行成批渲染。

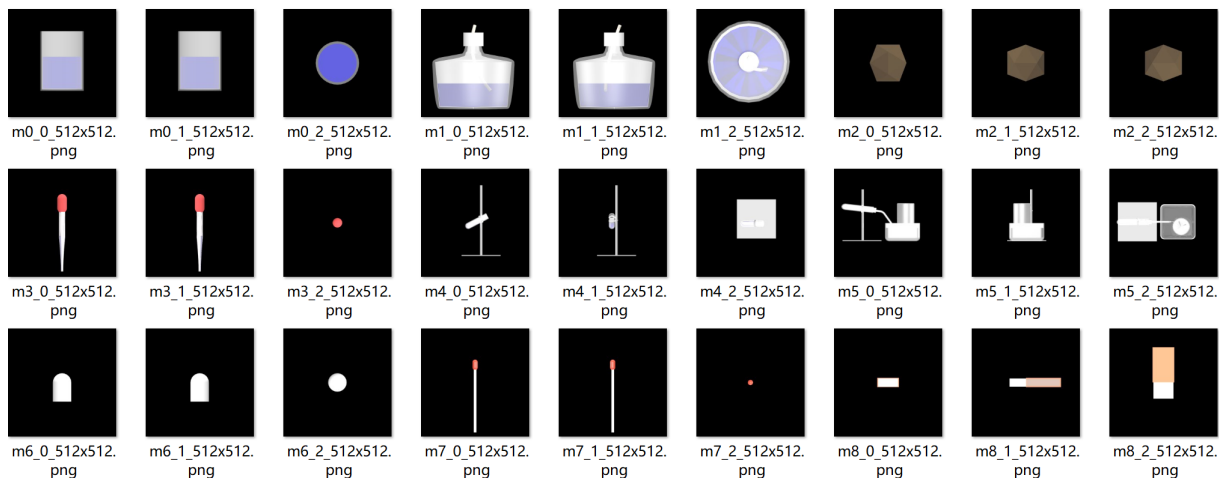


Figure 8: 渲染结果

2. 视图特征提取

得到三视图之后，我们期望能够从三视图中提取出模型的特征。

(a) 提取轮廓特征

为了提取出视图特征，首先需要提取出视图中的轮廓图。

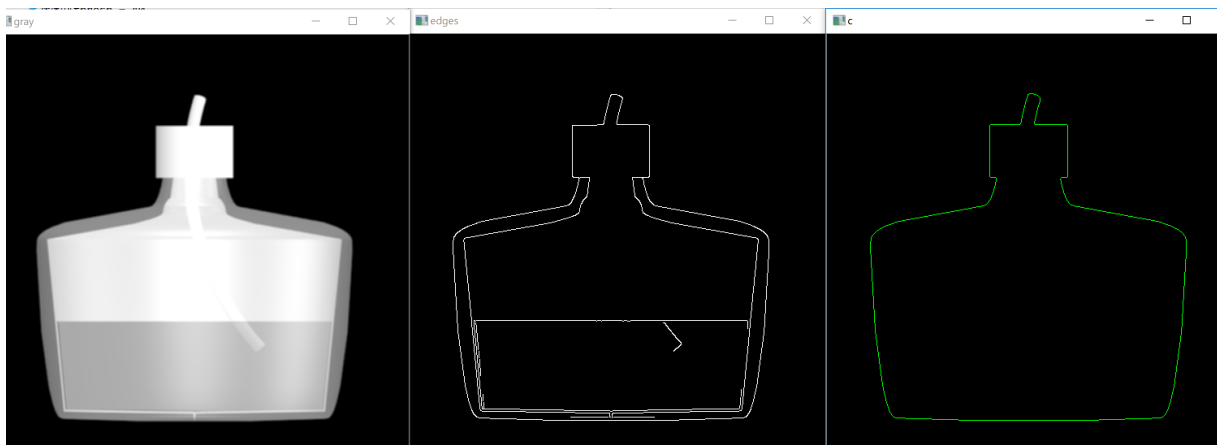


Figure 9: 均值滤波后的灰度图像/Canny 边缘检测/外轮廓

提取轮廓前首先对图片进行了均值滤波处理。

Canny 能够清晰的识别出图片的内外轮廓。但是使用 Canny 会有一些局限性：输出结果是二值图像，大小也和原始图像相关。所以使用 Canny 的话需要对输出结果进行二次处理。

最后我们放弃了 Canny 来提取轮廓，选择了直接使用 opencv 提供的 API: `findContours()` 来完成外轮廓的提取（图 9 最右图），获得了较好的效果。

(b) 特征提取

图片特征提取中的 SIFT (Scale-invariant feature transform) 算子，由于具有尺度不变性、旋转不变性，并且检测效果受图像亮度、拍摄视角影响较小，所以我们首先尝试了 SIFT 算子提取特征。

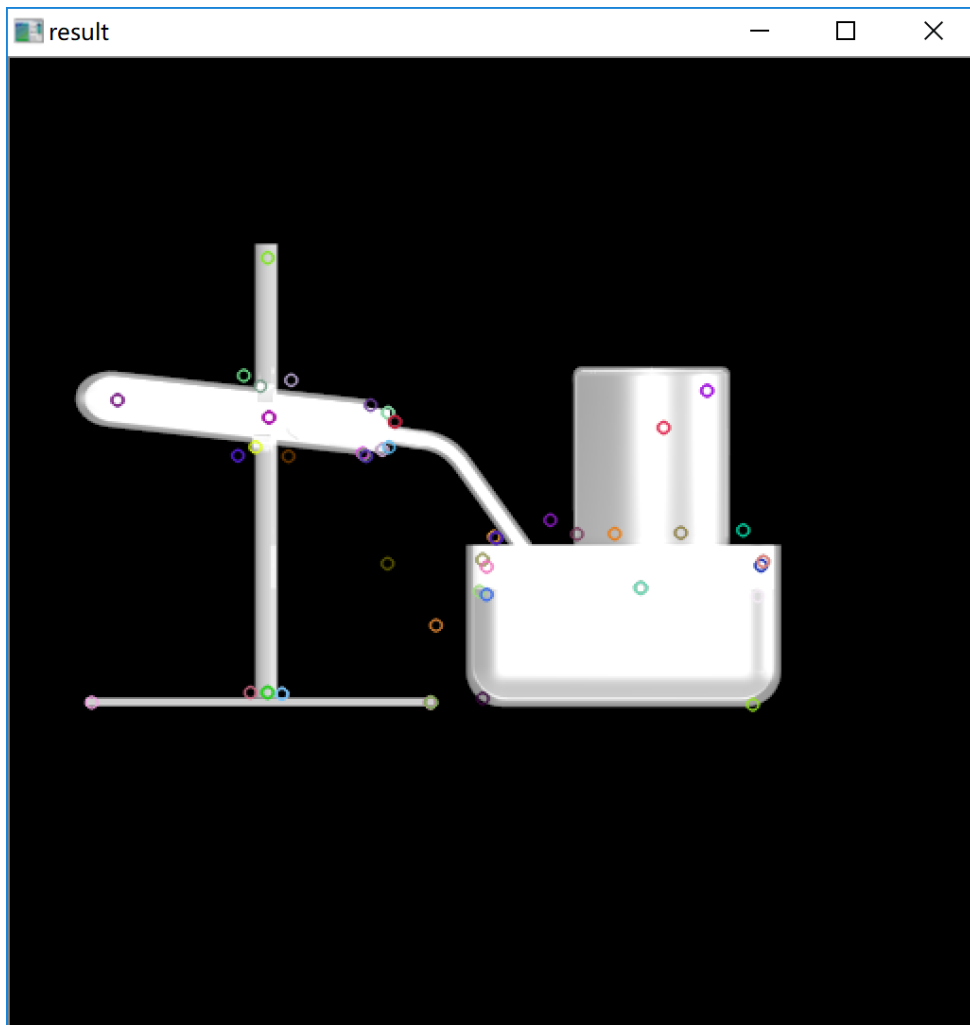


Figure 10: 使用 SIFT 检测原图特征

检测效果受到了模型的高光影响，甚至在背景中也出现了 key point。为了去除这些干扰信息，我们又使用 SIFT 对轮廓图进行了特征提取。

然而，使用轮廓图作为检测的原图又会导致 key point 减少的问题。在示例轮廓图中，

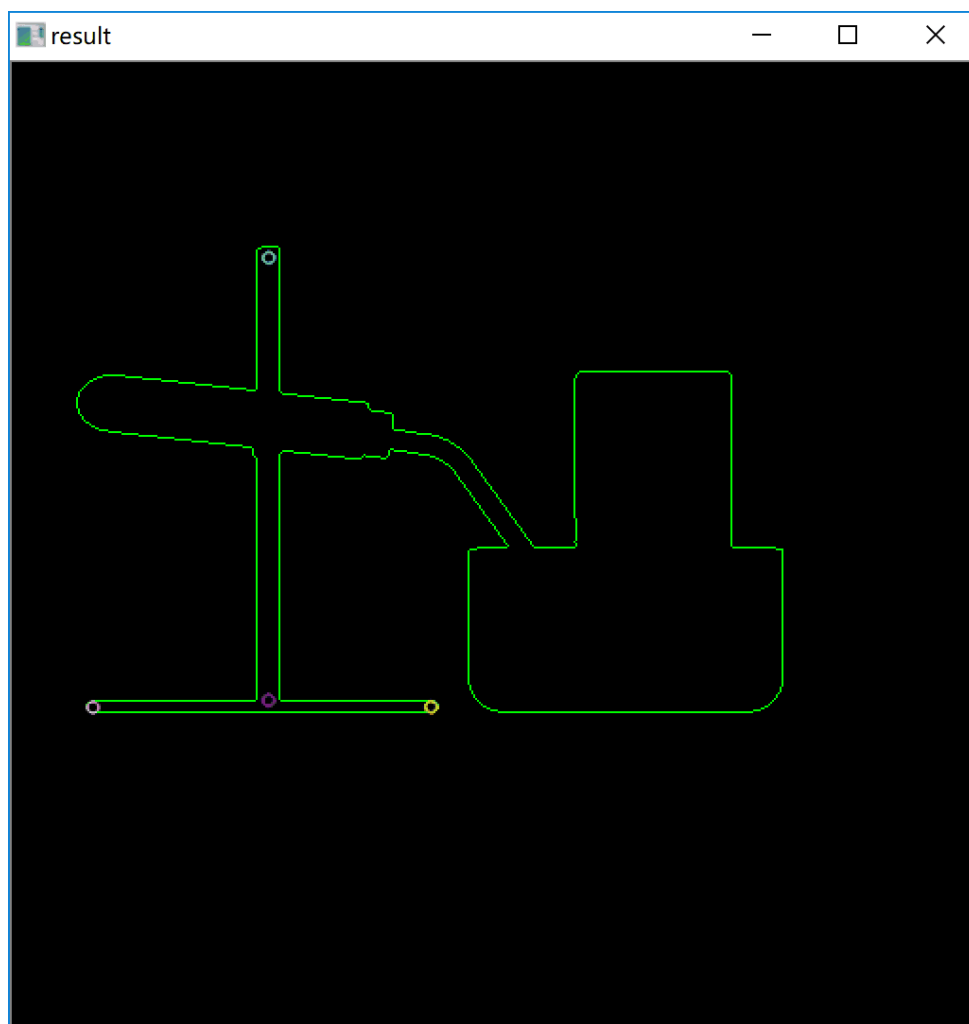


Figure 11: 使用 SIFT 检测轮廓图特征

整个右部都没有特征点。考虑到 SIFT 算子相对来说实时性还不够高、特征点较少、对边缘光滑的目标无法准确提取特征点等缺点，我们在这两次实验后放弃了 SIFT 算子提取轮廓的方法。

这之后我们又尝试了傅里叶描述符 (Fourier Descriptor)。

傅里叶描述符 (Fourier Descriptor) 将物体的形状看做是一条封闭的曲线，称为边界曲线。把曲线上的点 (x,y) 表示为复数形式 $(x+yi)$ ，就可以把边界曲线看作描述点变化的周期函数，这个函数用傅里叶级数展开，将得到一系列复数形式的系数，它们共同描述了边界的形状。

使用傅里叶描述符来提取特征的过程：

- i. 得到原始傅里叶描述符

首先提取图片的轮廓曲线，将曲线中的点坐标表示为

$$x + yi$$

的形式，对复数点集进行傅里叶展开，这样就得到了原始的傅里叶描述符

ii. 降低描述符 degree

由于轮廓曲线中的点比较密集，所以得到的描述符的 degree 也比较高，所以可以对描述符进行重建，降低 degree，方便减少存储的空间，也利于特征匹配。

iii. 对提取的描述符进行逆变换

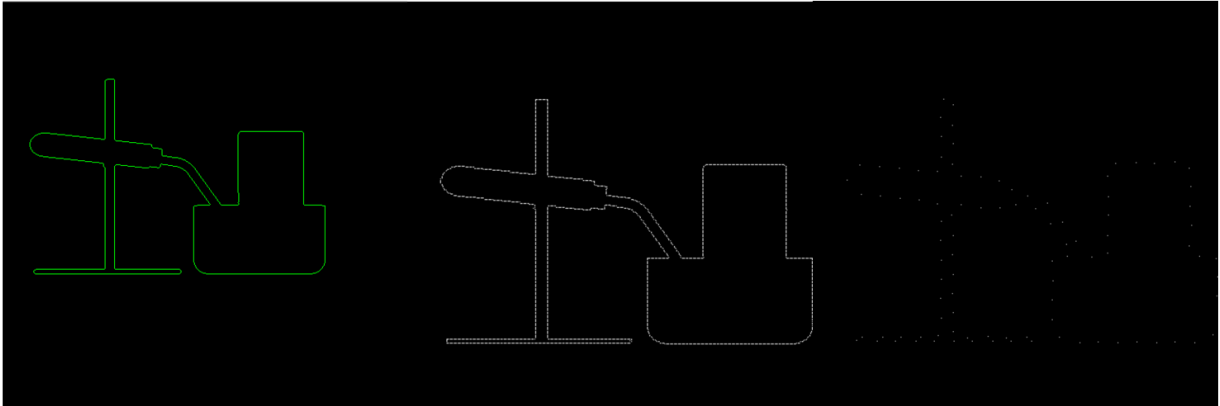


Figure 12: 原始轮廓 / 高度数描述子逆变换轮廓 / 低度数描述子逆变换轮廓

使用傅里叶描述符提取效果良好，描述子也可以很方便的存储为 `numpy` 数组文件格式，因此我们决定使用傅里叶轮廓描述符进行特征提取。

3. 视图特征匹配

我们尝试了两种特征匹配方法：

(a) 向量机

把匹配问题看作一个分类问题，使用支持向量机 (Support Vector Machine, SVM) 对输入的数据分类到不同的模型。

(b) K-最近邻

使用 K 最近邻 (kNN, k-NearestNeighbor) 分类算法，使用最近的 K 个邻居的类别作为输入的类别。

由于目前没有足够的训练数据，我们对 20 个模型的正视图提取了描述符，然后对描述符增加高斯噪声来生成训练和测试数据。

设置训练集和测试集的大小为 2000 (每个模型有 100 个样本)，在不同描述符 degree 下分类的粗略结果如图表所示：

对此测试结果，我们分析如下：

- degree 对分类准确度有较大影响，后续需要调整合适的训练参数
- 目前的准确率都比较低，但是考虑到随机样本的不真实性，我们打算后续使用手绘简笔画进行进一步的测试和调整

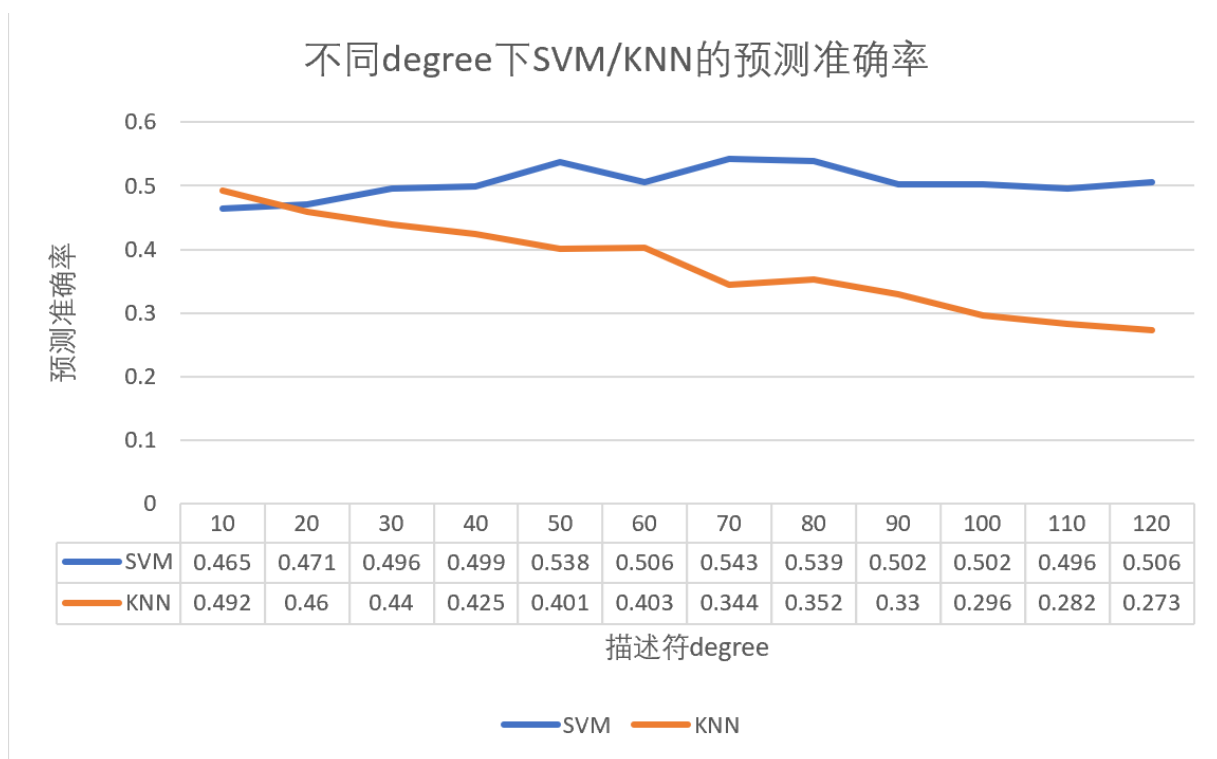


Figure 13: 不同 Degree 下 SVM/KNN 的预测准确率

5 工作内容

5.1 已完成工作内容

1. VR 模块

- (a) 绘画模块
- (b) 导出绘画信息
- (c) 导入模型

2. 模型检索模块

- (a) 模型三视图渲染
- (b) 三视图特征提取
- (c) 特征匹配

5.2 分工完成情况

| 姓名 | 学号 | 分工 | 完成情况 |
|-----|--------------|--------|------------------------|
| 罗宇辰 | 516030910101 | 模型检索模块 | 模型三视图渲染, 三视图特征提取, 特征匹配 |
| 陈志扬 | 516030910347 | VR 模块 | VR 绘图, 导出模型 |
| 陈 诺 | 516030910199 | 模型检索模块 | 资料查找, 三视图特征提取, 文档及 PPT |

5.3 后续工作内容

1. VR 模块

- (a) 提高画笔在黑板上绘画的真实感
- (b) 提供板擦
- (c) 提供多种颜色和粗细的画笔

2. 模型检索模块

- (a) 提高检索准确率
- (b) 针对简笔画输入优化特征提取模块
- (c) 提供快速扩充模型库的方法