

数图大作业 3 报告

班级：_____ 自 93

学号：_____ 2019010850

姓名：_____ 王逸钦

完成日期：_____ 2022/1/16

注：油画效果代码在 oilpainting.m，对内存容量要求较高；其余代码均位于 proj3.m

一 模板与整体思路

选择三个模板如下，分别为 1.有遮挡的情形；2.有射影变换的情形；3.有弹性变换的情形。以“Jerry”图片为例，对模板进行展示：



原图



转换目标



卡车模板



画廊模板



食谱模板

我们的目标是将三个模板上抱奶酪朝左的 Jerry 替换为转换目标，也即张开双手朝右的 Jerry。为此，我们可以对任务进行拆分：

- 1) 标定出朝左 Jerry 所处的矩形在模板中的具体位置，得到掩膜 Mask；
- 2) 设计空间变换算法，将朝右 Jerry 映射到模板的目标位置；
- 3) 对朝右 Jerry 进行风格变换，让其风格朝左 Jerry 尽量统一或实现独特风格；

二 算法与实现

1) 目标区域掩膜的生成

采用颜色分割的策略，将纯色图片导入 PhotoFunia，提取输出图片中像素 RGB 处于某一范围的像素点，即可得到一个 Mask。



卡车(红)

```
Truck_blue<100 & ...
Truck_green<100 & ...
Truck_red>150;
```

卡车 提取目标区域



画廊(绿)

```
Gallery_blue<100 & ...
Gallery_green>200 & ...
Gallery_red<100;
```

画廊 提取目标区域

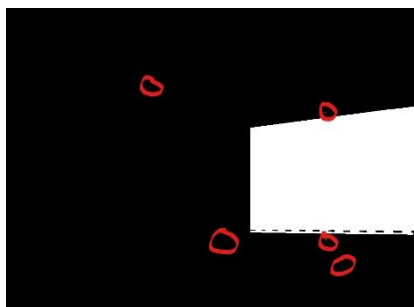


食谱(蓝)


```
Cookbook_blue>200 & ...
Cookbook_green<120 & ...
Cookbook_red<120;
```

食谱 提取目标区域


三个模板分别采用了红、绿、蓝色，尽量选取背景中相应原色较少的原色。分别对每种原色设置阈值，即可提取出标记有目标区域的二值掩膜，如下所示：



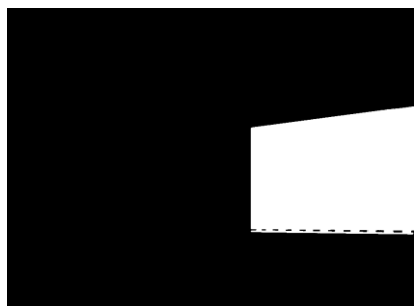
卡车 Mask




画廊 Mask



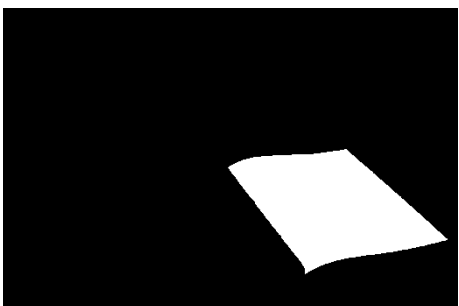
食谱 Mask



卡车 Mask



画廊 Mask



食谱 Mask

通过上述方式得到的掩膜存在一定的缺陷：卡车 Mask 在目标区域外部有白色亮斑，因为外部部分像素也符合了提取条件；画廊 Mask 白色区域宽度不够，叠加到目标 Jerry 之后将导致 Jerry 图片左右两侧被“削去”；食谱 Mask 靠近书脊和书外沿位置的白色中夹杂着黑色斑点。为此，我采用简单的形态学开、闭运算和赋值方法对上述 Mask 改进，改进后 Mask 在上方第二行。

2) 目标区域标定与射影空间变换

$$\bar{x}' = \bar{H}\bar{x} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

射影变换将正方形投射为梯形，其中齐次射影变换阵 \bar{H} 具有 8 个自由度。在本次作业中，卡车、画廊两例的空间变换算法都可以用射影空间变换实现。在 MATLAB 实现中一般用 4 个顶点的位置来标定变换前后的位置，然后调用 `fitgeotrans` 函数，将参数设为 'projective' 即可按投影方式生成变换法则。我对课堂展示代码 `registration_control.m` 进行修改，利用交互式标定 `ginput` 确定顶点位置之后就将顶点位置写死在代码中，根据效果稍做微调，实际运行时不再需要重复标注。

注意：只有在掩膜为真的位置，原始图像才可能被目标图像(朝右 Jerry)所替换，掩膜为假的位置一律保持原图状态。



卡车 射影变换效果



画廊 射影变换效果

可以看出射影空间变换很好地解决了卡车、画廊两模板的图像变换需求。但卡车广告“暗沉”的光线效果并未体现，Jerry 的背景仍为 $RGB=(255,255,255)$ 的纯白色，这一点将在【4）光线风格变换】中得到解决。

3) 目标区域标定与弹性空间变换

书页各边并非直线，涉及弹性空间变换。弹性空间变换的方式多种多样，包括 B-样条变换，TPS(薄板样条)变换等。

本例中我采用了原理更为直观的多项式拟合变换法，按 Matlab 文档进行参数设置，只需将 `fitgeotrans` 函数参数设为 'polynomial' 即可按多项式方式生成变换法则。为保证精准性，经过反复实验我最终决定使用 3 阶多项式拟合+24 个标定点，其中每条边分配 6 个标定点。标定点的确定仍采用先 `ginput`，后写死的方式。

与前述类似，仅掩膜为真区域可被替换；变换结果也存在光线不真实、页面过白的现象，将在【4）光线风格变换】中得到解决。Jerry 图片边缘留白多，为更好表现边缘切分情况，补充展示一张我的自拍图片：



食谱 弹性变换效果

4) 光线风格变换

卡车、食谱两模板涉及明显的光线和色温变化，卡车图案整体颜色更加暗沉，食谱颜色整体偏暖且页面左上方有反光；画廊模板无明显的光线风格变化。为模拟上述效果，我首先将纯白图片输入 Photofunia，将处理后的图片叠加在空间变换后的图片上，即可将 Photofunia 原始模板的光线变化转移至我的图片上。



卡车(白)



食谱(白)



卡车 光线变换效果



食谱 光线变换效果

可见经过光线变换，风格已经与原图非常接近，很好地模拟了原图效果。

5) 纹理风格变换（油画效果）

虽然画廊模板并未进行任何明显的风格变换，但作业文档中展示了加入预设纹理风格变换的案例，因此我决定在画廊模板中实现相对难度较高的油画效果，作为

纹理变换的典型代表，同时也和画廊主题相匹配。

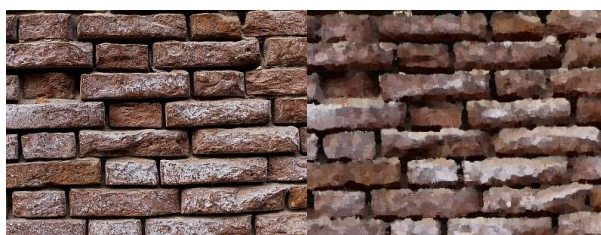
油画效果的实现思路主要是：将像素点按亮度等分为 L 个区间，落在同一区间内视为亮度相同。选取一个半径为 r 的正方形核，让核扫过全图，用整个核中出现频率最高的亮度所对应的像素(主导像素)替换核中心的像素。经过这样的一轮扫描，图片的细节就会在很大程度上被抹去，取而代之的是一个一个“色块”，从而模拟了油画颜料。改变 L 和 r 的取值，就会改变油画的“细腻”/“粗糙”程度。

Jerry 图片等卡通图片不能很好地展示油画效果，因为卡通图片本身细节较少，为此我又选用了一些其他图片，一并展示如下：



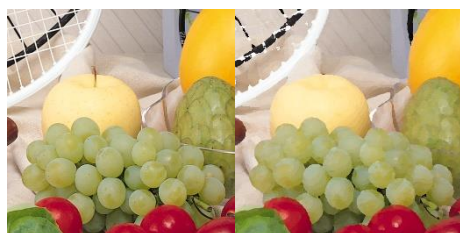
原图

油画



原图

油画



原图

油画



原图

油画

将油画风格应用到画廊模板：



三 难点与亮点

1) “油画效果”算法及其并行化实现

油画效果的实现方式是我本次作业中的最大亮点。上述油画算法的实现方式，最直观的无疑是使用双重 for 循环，让半径为 r 的正方形核扫遍全图，对每个核单独进行运算和操作。这种方法在空间上具有优势，内存中只需要保存循环中某一次运算产生的变量，但在时间上效率低下，特别是对于 1920×1080 规模或更大的图像，像素多达 200 万，串行运算十分缓慢。

为此，我花费半天多的时间终于将串行化的“油画效果”算法成功并行化。其主要思路如下：

1. 首先将原图的灰度图降采样后，向各个方向平移(空白处补零)，获得 $(2r+1)^2$ 张图片，排列在宽、高之后的第 3 维。此时，核的移动就转化为了图片的移动，只要确定了宽、高、就能找到对应的 $(2r+1)^2$ 个像素，这些像素恰好对应着以该宽、高为核中心时核所囊括的 $(2r+1)^2$ 个像素。对彩色原图也采用上述移动方法，将 $(2r+1)^2$ 排列在第 4 维。

2. 对于每个位置，查看降采样灰度图中 $(2r+1)^2$ 个像素里亮度的众数(零除外)，该像素即为主导像素，记录下它在 $(2r+1)^2$ 个像素中的序号 Index。依据此 Index 即可在彩色图像中定位到该 Index 对应的 RGB 三通道值，用其替换当前位置 RGB。

在第二步中，为避免使用 for 循环，我将高维数组使用 num2cell 函数，在需要压缩的维度压缩为元胞数组，从而可以使用 cellfun 对每个元组同时进行操作，实现并行化，大大提高了运算效率。此外，我还解决了无法取到众数导致数组为空，进而使索引为 0，众数直接取到 0 等等 corner case，也被我一一排除。

最终算法可以在 30 秒实现 90 万像素图片(上述自拍图片)的油画效果(radius=4, intensity_level=20, cpu=AMD R7-5800H)

2) 使用开运算提升分割效果

相关算法已在【1) 目标区域掩膜的生成】中简述，卡车模板使用开运算消除白色噪声，而食谱模板使用闭运算消除黑色噪声。这是对前几门课程所学形态学处理方法的灵活运用。

3) 多项式弹性空间变换

采用直观，简单易行的多项式拟合方法对弹性空间进行拟合，本作业中我求取二元 3 次多项式函数与 24 个采样点的最小平方偏差，即可得到多项式。相关算法在[1] Goshtasby, Ardeshtir, "Image registration by local approximation methods," Image and Vision Computing, Vol. 6, 1988, pp. 255-261. 中描述，文档附于作业文件夹根目录中，不再赘述算法。

4) 利用纯色图片

巧妙使用 Photofunia 对纯原色图片的输出结果，帮助分割目标区域前景和背景；巧妙使用 Photofunia 对纯白色图片的输出结果，帮助提取光线变化。

四 困难、解决、收获

本次作业最关键的是在代码编写前理清思路，将任务进行切分，并确定每个任务需要用到的核心算法，确定最关键 matlab 函数。而后就是常规的编写和 debug 过程了。由于 fitgeotrans 较为易用，ginput 也让标记工作并不困难，因此空间变换这一最为关键的算法实现起来较为顺利。

真正让我卡住很久的是油画效果的并行化实现，想将原本用 for 循环很容易实现的思路转变为矩阵式运算，很多细节都要推翻重来，重新建构，这一部分让我充分掌握了 matlab 代码并行化的关键函数：cellfun、arrayfun、structfun，它们都可以对内部元素同时应用函数，达到并行作用。

本次作业综合运用掩膜、空间变换、形态学运算等前述课堂所学知识，由于我的代码结构清晰直观，只需要在 proj3.m 最上方改变模板号和目标图片路径，更改最下方图片保存路径，然后运行即可得到结果，因此没有单独编写图形界面的必要性。

数图课任务量很大，但助教设计的历次作业都很自洽，难度定位适中，可以充分考察课堂所学，同时也能充分调动我们的创新积极性。感谢你们本学期的付出！

附：模板链接

<https://photofunia.com/effects/at-the-gallery>

<https://photofunia.com/categories/misc/truck-advert>

<https://photofunia.com/categories/books/cookbook>