

基于 matlab 的图片漫画风格化程序

富诗涵 北京邮电大学 2019213493

目录

1. 研究背景与动机	2
2. 研究问题和挑战	2
3. 国内外相关工作	3
4. 技术方案概要设计和详细设计	3
4.1 弱化细节	4
4.2 边缘线条检测	4
4.3 增强图像色彩	5
5. 实验方法	5
5.1 算法与代码	5
5.1.1 使用 Sobel 算子进行边缘线条检测	5
5.1.2 使用双边滤波器弱化细节	7
5.1.3 使用图像插值法增强图像色彩	8
5.2 实验环境及功能代码	9
6. 实验结论分析	10
7. 总结	13
8. 参考文献	13

1. 研究背景与动机

人类的文明发展离不开文化建设，高屋建瓴的文化建设才能造就强盛的名族和国家。随着文化建设的发展，人们开始更加注重自己的艺术感与审美能力。艺术对一个人的熏陶是潜移默化的，一个学习艺术的人经过长期的艺术熏陶，在谈举止，穿着打扮方面都会有不同的气质，并且他们会更加懂得生活，对生活有自己的理解和更高的追求。因此，越来越多的人愿意培养自己的艺术情操以及绘画能力，以生动多彩地再现现实生活、表达自己对于审美主体地内心反应、提升自己的艺术感。但由于种种原因，很多热爱美术地人并不能系统并长久地打下坚实的绘画基础，因此在实际绘画中会遇到很多问题，如对于结构的把握不到位、无法调配出最佳合适的色彩。

为了解决基础薄弱的绘画着临摹难、构思难的问题，我们提出将图片漫画化的程序，通过将图片解构为线条与色彩，帮助使用者更好的观察作品。本程序旨在帮助使用者将图片结构并提取其中的具有鲜明对比度的色彩以供绘画者作为绘画时的参考，辅助漫画和动画制作者进行快捷的作品创作，也可以帮助绘画新手进行绘画学习。

2. 研究问题和挑战

现实中的摄影作品的细节较多，边缘轮廓并不清晰，且照片的颜色对比度相对较低。然而动漫图片不同，动漫图画有以下三个主要特点，动漫中的细节相对少，动漫中的边缘轮廓更突出以及动漫的色彩更鲜艳^[1]。因此将图片转化成漫画的形式，需要主要考虑两个过程，需要从线条及色彩两方面重点考虑，即从中提取出精炼的线条，从中捕获并强化图片的色彩。

在图像领域，即图片边缘化及图像色彩增强。

3. 国内外相关工作

在线条生成方面，现有的自动生成线条的方法可以分为数据驱动和非数据驱动两类^[2]。

非数据驱动的方法大多基于滤波。Gooch^[3]提出了基于 DOG 生成卡通风格线条画的方法。但 DOG 滤波容易产生较多噪声，不能识别关键线条。罗光蕊等^[4]提出了一种基于小波多分辨率的线条画生成方法，但其仅在风景图片上进行了实验测试，且未能解决如何良好地反映画面中光线的入射角度的问题。该方法对物体阴影处理得不够理想，有的线条可能会出现部分断线。王少荣等^[5]提出了一种自适应的线条画绘制方法。通过将场景图像分割成若干个区域并分别计算每个区域亮度的方差以及每个像素到边界的最小距离，将每个区域的方差和面积的比值作为该区域的复杂度。再利用计算出地复杂度反映边缘切线，使用基于流的各向异性高斯差分滤波生成线条画。该算法的缺点时，其提取的线条是由一系列边缘点组合而成，其生成的线条在表现艺术性和抽象性方面仍不及基于笔刷绘制的方法。

数据驱动方法一般先对转化信息进行定位，再根据该定位生成系统中存储的漫画化实体^[6]。陈洪等^[7]提出一个基于样本学习的人脸线条画生成系统，使用非参数化采样方法和灵活的线条画模板。对于给定图像上的任意像素点及其邻域，通过在样本空间搜索并匹配所有的相似邻域，计算该像素点在相应的线条画上出现的条件概率。其系统的特点是，还引入了艺术家的风格，系统将根据其与条件概率共同绘制期望的线条画，然后使用模板匹配得到最终输出的线条画。Zhang 等^[8]提出了一种数据驱动的方法来从彩色照片生成风格化的肖像画，得到的结果自然真实，具有一定的吸引力。

在颜色处理方面，Land^[9]在 1965 年提出 Retinex 理论，并且成功应用到图像的动态范围压缩和图像色彩保真^[10]。但其在处理夜间彩色图像时容易出现光晕、颜色失真、细节丢失与噪声干扰等问题。赵宏宇^[11]等提出基于马尔科夫随机场(MRF)的针对单幅图像的 Retinex 图像增强算法。该算法在 HSV 颜色空间下采用线性引导滤波估计图像照度分量。在 MRF 模型下求解仅包含物体本身特性的反射分量，并通过颜色恢复函数与增益补偿方法进行颜色恢复与校正，最终实现了夜间彩色图像的增强。虽然算法效果较好，但是算法构建的边的数目非常多，运算速度较慢。邵帅等^[12]提出了利用改进的多尺度 Retinex 算法与局部对比度自适应调整相结合的方法来改善图像质量，并利用 Sigmoid 函数替换对数函数对 MSR 算法进行改进，避免图像数据的损失；利用局部对比度自适应调整方法提高图像的局部对比度，使得图像的边缘等细节信息有效的保留。F. Marques 等^[13]发明了根据区域参数化、区域插值、区域排序和分区创建进行图像插值进而处理图片颜色的方法。该技术已在大量序列上进行了测试使用不同的分割技术。在所有情况下，结果的质量水平均较高。

4. 技术方案概要设计和详细设计

简单来说，图片漫画风格化程序主要需要图片边缘化，并适当弱化细节，并增强图片色彩对比度，使图片颜色更加鲜艳。本程序的流程图如图 1 所示。

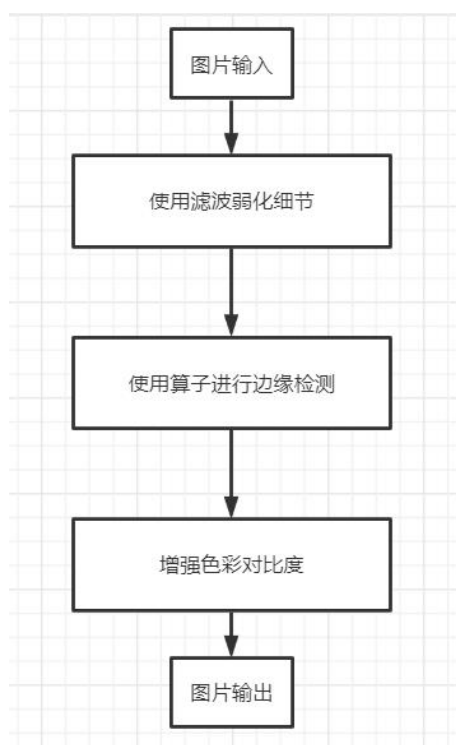


图 1 设计流程

4.1 弱化细节

细节就是图像中的高频成分。

通常使用滤波进行图片平滑处理以消除过多图片中的细节。其原理为：将掩膜内所有像素点值进行排序，然后用中位数（中值）来作为当前像素点的值。常用的滤波器有均值滤波器、高斯滤波器、中值滤波器等。但是这些滤波器都会将所有高频信息进行弱化，其中也同样包括图像的边缘。

为了避免边缘被模糊，我们使用双边滤波器。这种滤波器的特点是可以“保边滤波”（区域平滑），它可以在保留清晰的边缘的前提下只模糊区域内部。其大致原理是在高斯滤波器基础上加上了相似度权重，在高斯滤波模板的每个点上再乘以一个与中心点的相似度系数，从而将边缘与内部区分处理^[15]。

其具体实现将在第 5 节详细阐述。

4.2 边缘线条检测

边缘检测的目的是检测识别出图像中亮度变化剧烈的像素点构成的集合。边缘检测可以减少源图像的数据量，剔除与目标不相干的信息，保留图像重要的结构属性。在图片漫画风格化的使用中，即为给图片打出轮廓草稿，方便使用者进行绘画参考。

边缘检测分为以下三个类型^[14]：

1. 一阶微分为基础的边缘检测，通过计算图像的梯度值来检测图像的边缘，如 Sobel 算子、Prewitt 算子、Roberts 算子及差分边缘检测。
2. 二阶微分为基础的边缘检测，通过寻求二阶导数中的过零点来检测边缘，如拉普拉斯算子、高普拉普拉斯算子、Canny 算子边缘检测。
3. 混合一阶与二阶微分为基础的边缘检测，综合利用一阶微分与二阶微分特征，如

Marr-Hildreth 边缘检测算子。
本程序中选择一阶微分为基础的边缘检测。

表 1 不同算子比较^[14]

roberts 算子	sobel 算子	prewitt 算子																																												
<table><tr><td>-1</td><td>0</td><td>0</td><td>-1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr></table> <p>Roberts</p>	-1	0	0	-1	0	1	1	0	<table><tr><td>-1</td><td>-2</td><td>-1</td><td>-1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>-2</td><td>0</td><td>2</td></tr><tr><td>1</td><td>2</td><td>1</td><td>-1</td><td>0</td><td>1</td></tr></table> <p>Sobel</p>	-1	-2	-1	-1	0	1	0	0	0	-2	0	2	1	2	1	-1	0	1	<table><tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>-1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>-1</td><td>0</td><td>1</td></tr></table> <p>Prewitt</p>	-1	-1	-1	-1	0	1	0	0	0	-1	0	1	1	1	1	-1	0	1
-1	0	0	-1																																											
0	1	1	0																																											
-1	-2	-1	-1	0	1																																									
0	0	0	-2	0	2																																									
1	2	1	-1	0	1																																									
-1	-1	-1	-1	0	1																																									
0	0	0	-1	0	1																																									
1	1	1	-1	0	1																																									

利用这些算子进行边缘检测的本质就是将算子与图像进行空间二维卷积。具体的实现将在第 5 节详细阐述。

4.3 增强图像色彩

增强图像色彩的本质就是提高其颜色饱和度。颜色的饱和度高，就显得鲜艳，反之，饱和度低，就显得灰暗。

现有的提升饱和度有图像插值法，或者转换到 HSI 色彩空间然后增强饱和度分量等等^[15]。
本程序选择图像插值法，其具体实现将在第 5 节详细阐述。

5. 实验方法

5.1 算法与代码

5.1.1 使用 Sobel 算子进行边缘线条检测

索伯算子（sobel operator）常用于边缘检测，在粗精度下，是最常用的边缘检测算子，以广泛应用几十年。sobel 算子由两个 3X3 的卷积核构成，分别用于计算中心像素邻域的灰度加权差。分为垂直方向和水平方向的索伯滤波器 Gx and Gy。

边缘检测时：Gx 用于检测纵向边缘，Gy 用于检测横向边缘。

计算法线时：Gx 用于计算法线的横向偏移，Gy 用于计算法线的纵向偏移

sobel 算子应用时进行给定图像的卷积操作，卷积为计算图像大矩阵周围像素和滤波器矩阵对应位置元素的乘积，然后把结果相加到一起，最终得到的值就作为该像素的新值，。这就是完成一次卷积的计算过程。随后，卷积核将继续移动与大矩阵每个位置进行卷积运算，直到大矩阵中的每个位置都已经进行过卷积计算。因为相邻像素卷积结果一般具有相似输出，

导致大量冗余信息产生，因此为了减小输出值，进行池化操作，通过求取最大小值或者平均值进行池化以减小输出值^[16]。

假设一副图像的 3×3 区域如图 2(a) 所示，图 2(b)、图 2(c) 即分别为上述 G_x 与 G_y。

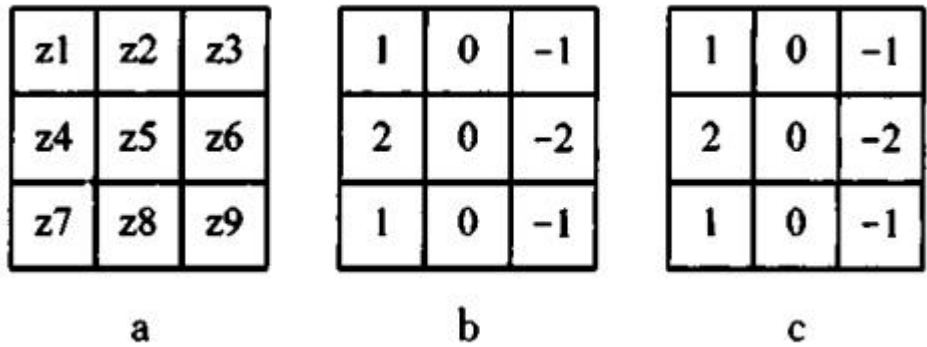


图 2 图像 3×3 区域和 sobel 算子^[16]

将图像区域分别与两个模板中的对应权值相乘，并将所有乘积相加，得到 3×3 模板垂直和水平两个方向的近似偏导，如式(1)、式(2)。

$$g_x = (z_1 + 2z_4 + z_7) - (z_3 + 2z_6 + z_9) \quad (1)$$

$$g_y = (z_1 + 2z_2 + z_3) - (z_7 + 2z_8 + z_9) \quad (2)$$

在计算位置像素梯度值时，一般选择式(2)并对梯度图像进行阈值的二值化处理，进而能除掉部分伪边缘，得到清晰二值化边缘图像。我们会根据不同环境的要求设置合适的阈值，当梯度值大于阈值，则认为该像素点为图像的边缘点，当梯度值小于阈值，则认为该像素点不是图像的边缘点。

由于 Sobel 算子具备一阶算子算法简单，处理速度快的优点，同时兼具使用加权平均算法，对图像中的一些噪声具有一定的抑制能力，因此本程序在线条化方面使用了自定义 sobel 边缘检测函数 my_edge。下图为具体实现代码展示。

```

function output = my_edge(input_img)
if size(input_img,3)==3
    input_img=rgb2gray(input_img);
end
input_img=im2double(input_img);
sobel_x=[-1, -2, -1;0, 0, 0;1, 2, 1];
sobel_y=[-1, 0, 1;-2, 0, 2;-1, 0, 1];
psf=fspecial('gaussian',[5,5],1);
input_img=imfilter(input_img,psf);%高斯低通滤波,平滑图像,但可能会使图像丢失细节
input_img=medfilt2(input_img);%中值滤波消除孤立点
[m,n]=size(input_img);
output=zeros(m,n);
for i=2:m-1
    for j=2:n-1
        local_img=input_img(i-1:i+1, j-1:j+1);
        output(i,j)=sqrt(sum(sum(sobel_x.*local_img))^2+sum(sum(sobel_y.*local_img))^2);
    end
end
output=imadjust(output);%使边缘图像更明显
thresh=graythresh(output);%确定二值化阈值
output=bwmorph(im2bw(output,thresh),'thin',inf);%强化细节

```

图 3 自定义 sobel 边缘检测函数

在动漫化过程中,为了节省计算资源,直接调用了 matlab 库函数中的 sobel 边缘检测函数。

```

edge_thresh = 0.02;
edge_operator = 'sobel';

edge_mask = uint8(edge(img_gray, edge_operator, edge_thresh));

```

图 4 调用了库函数 edge 进行 sobel 边缘检测

5.1.2 使用双边滤波器弱化细节

双边滤波器是 1998 年由 Tomasi 等^[17]基于高斯滤波器提出的一种改进算法^[18],它可以被看成是一种加权的非线性高斯滤波。双边滤波器有两个核函数,同时关注了像素在空间和幅度两个域上的相似性,具有平滑保边的优点。

图像滤波可用式(3)表示:

$$I(x, y) = \frac{1}{w_p} \sum_{i,j \in \Omega} w(i, j) * n(i, j) \quad (3)$$

其中, $I(x, y)$ 是滤除噪声后的清晰图像, $n(i, j)$ 是需要滤波处理的含噪声图像, Ω 是像素的邻域, $w(i, j)$ 是滤波器在点 (i, j) 处的权。对于高斯滤波来说,权值 $w(i, j)$ 和像素的空间距离线性相关,距离越近相关性越大,权值也越大,其滤波核函数可以定义如下^[19]:

$$w_k(i, j) = \exp\left(\frac{-(i-x)^2 + (j-y)^2}{2\sigma_k^2}\right) \quad (4)$$

σ_k 是高斯函数的标准差。

高斯滤波只关注像素的空间距离而忽略了像素值的变化，也就是图像灰度的改变，因此它在滤除噪声的同时，边缘也同时被平滑了。而双边滤波在上面权值的基础上增加了一个衡量像素值变化的权，计算公式如式(5)：

$$w_h(i, j) = \exp\left(-\frac{(I(i, j) - I(x, y))^2}{2\sigma_h^2}\right) \quad (5)$$

σ_h 是高斯函数标准差。

双边滤波的权值 w 是 $w_k(i, j)$ 和 $w_h(i, j)$ 的乘积，图像边缘处像素值变化大， $w_h(i, j)$ 值较小，从而使得 w 也变小，因此滤波器在边缘处的滤波作用降低，从而在滤波的同时保持了边缘。下图为具体实现代码。

```
for i = 1:dim(1)
    for j = 1:dim(2)
        % Extract local region.
        iMin = max(i-w, 1);
        iMax = min(i+w, dim(1));
        jMin = max(j-w, 1);
        jMax = min(j+w, dim(2));
        I = A(iMin:iMax, jMin:jMax);

        % Compute Gaussian intensity weights.
        H = exp(-(I-A(i, j)).^2/(2*sigma_r^2));

        % Calculate bilateral filter response.
        F = H.*G((iMin:iMax)-i+w+1, (jMin:jMax)-j+w+1);
        B(i, j) = sum(F(:).*I(:))/sum(F(:));

    end
    waitbar(i/dim(1));
end
```

图 5 双边滤波器实现代码

5.1.3 使用图像插值法增强图像色彩

两个图像之间的插值和外推为许多公共点和面积图像处理操作提供了一种通用的统一方法。亮度、对比度、饱和度、色调和锐度都可以通过一个公式单独或同时进行控制。在某些情况下，还有性能优势^[20]。

线性插值通常用于混合两个图像。混合分数 (α) 和 ($1 - \alpha$) 用于每个像

素的每个分量的加权平均值:

$$\text{out} = (1 - \alpha) * \text{in0} + \alpha * \text{in1} \quad (6)$$

通常, α 是 0.0 到 1.0 范围内的数字。这通常用于线性插值两个图像。较少被考虑的是, α 的范围可能超过 0.0 到 1.0 的区间。大于 1 的值在缩放 in1 时减去 in0 的一部分。低于 0.0 的值具有相反的效果。

如果将图像的退化版本用作“远离”的图像, 则外推特别有用。从黑白图像外推会增加饱和度。从模糊的图像中推断出可提高清晰度。插值/外推公式提供单参数控制, 使一系列图像的显示, 每个图像的亮度, 对比度, 清晰度, 颜色或饱和度都不同, 特别易于计算, 并且具有诱人的硬件加速。

要改变饱和度, 像素分量必须朝向或远离像素的亮度值。通过使用黑白图像作为退化版本, 可以使用插值降低饱和度, 并使用外推来增加饱和度。这避免了与 HSV 空间之间的计算成本更高的转换。交互式应用程序中的重复更新特别快, 因为不需要重新计算每个像素的亮度^[20]。



图 6 α 值对于色调的影响

```
% Implement interpolation and extrapolation.  
img_res = (1-S_scalar).*S_template + S_scalar.*img_double;  
img_res = uint8(img_res);
```

图 7 实现图像插值法增强图像色彩

5.2 实验环境及功能代码

实验环境: matlab r2019a

调用开源代码^[21]: "bfilter2.m", "bfltColor.m", "bfltGray.m", "saturation_adjust.m"

测试方法: 使用 matlab 打开 untitled666.m, 可以进入程序。

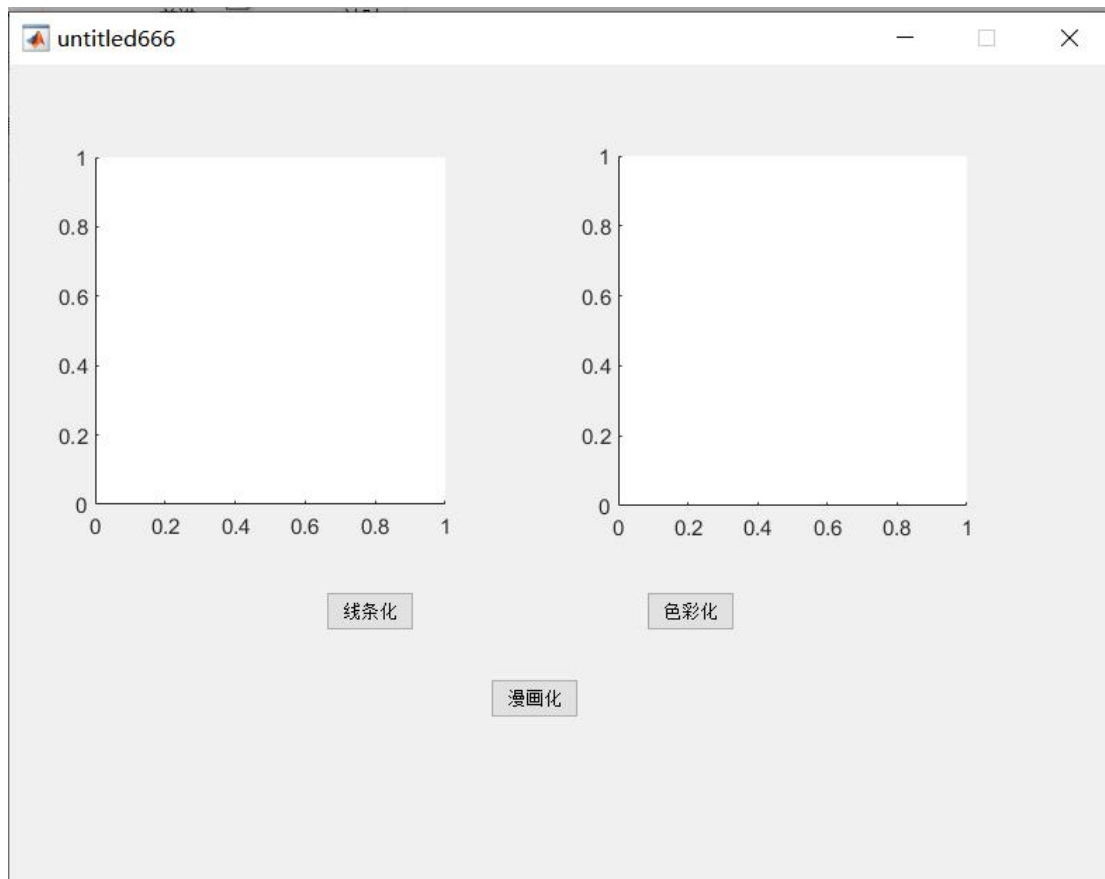


图 8 主程序使用

可以通过点击不同按钮，得到不同分析后的图片。

6. 实验结论分析

使用 matlab 构建 GUI，通过点击不同按钮，可以任意选择电脑中的任何一张图片并对其进行图片处理。有三种方法可供选择——图片线条化，图片色彩化以及图片漫画化。

图片线条化指，主要使用滤波及图片边缘处理，将其转化为黑白的、突出边缘的图片以供使用者更直观的看到图片的结构。以 lena 为例，其线条化后结果如图 9 所示。

图片色彩化是指，使用图像插值法增强图像色彩，使其色彩对比度提高，饱和度提高，更加偏向漫画而不是现实的摄影作品。以 lena 为例，其色彩化后结果如图 10 所示。

图片漫画化则将前两点结合起来，既将其边缘以线条形式画出，也将其色彩对比度提高，使得其更像一副美术作品。以 lena 为例，其漫画化后结果如图 11 所示。

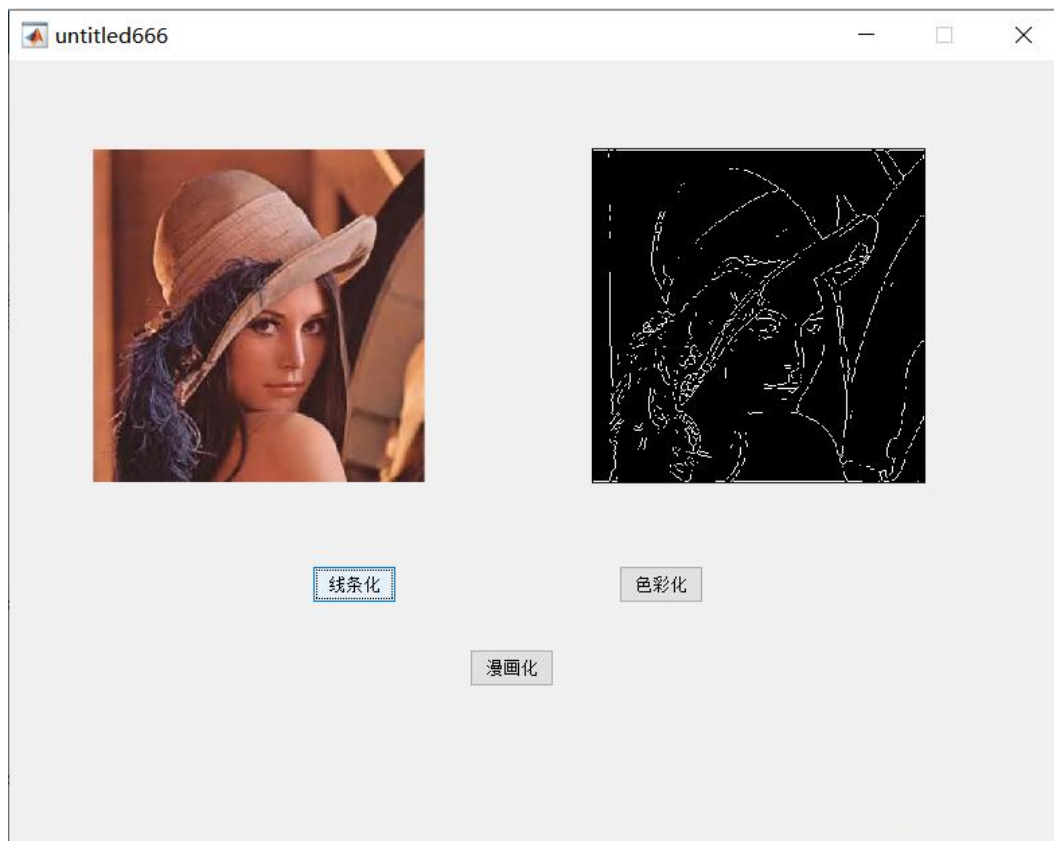


图 9 图片线条化示例

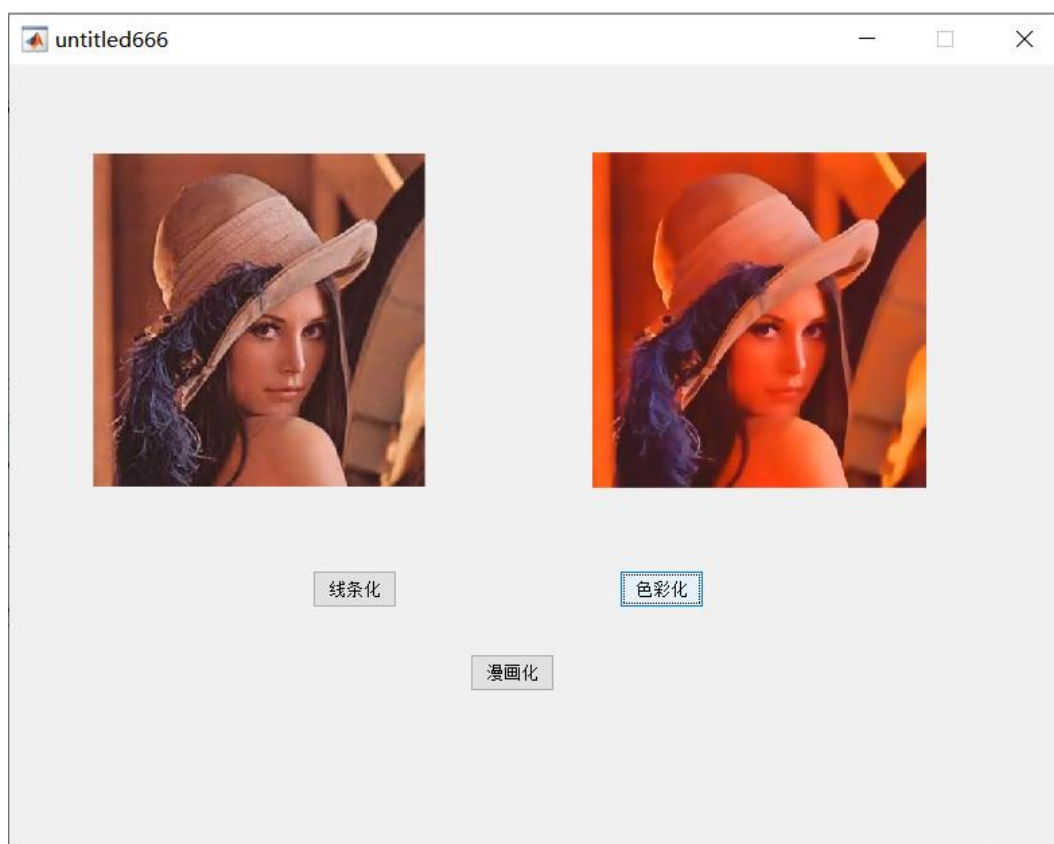


图 10 图片色彩化示例

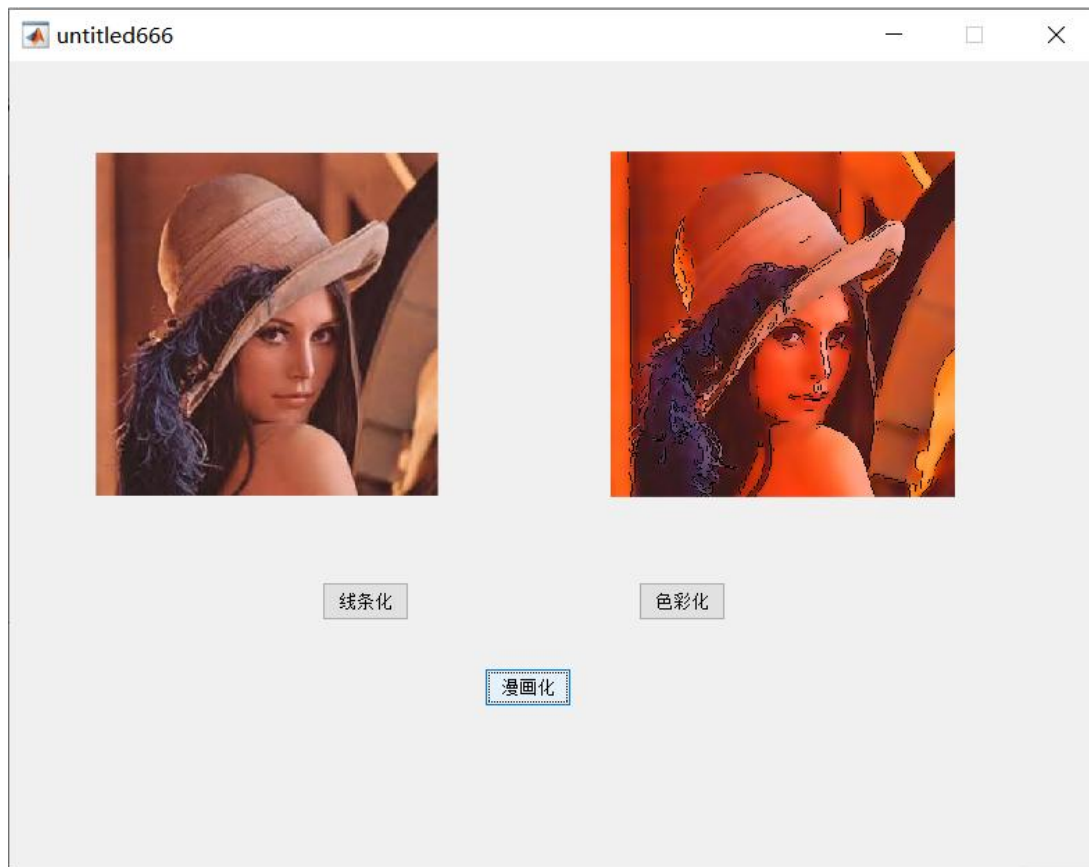


图 11 图片漫画化示例

在实际操作过程中，我们发现以先滤波后边缘检测的结果好于先边缘检测后滤波。因为如果先进行边缘检测的话，会将区域内部的某些明暗光影等检测出来，而这些是我们不希望过分突出的部分，导致最终生成的图像会掺杂过多非重点要素。而先进行保边滤波可以将区域内部的细节弱化乃至去除，在此基础上再进行边缘检测，效果会更好。

此外，我们发现在输入图片已经是漫画图时，再将其进行漫画化并不能得到很好的结果，只能将其主要颜色对比度进行提升。如图 12 所示。未来我们将对其进行改进。

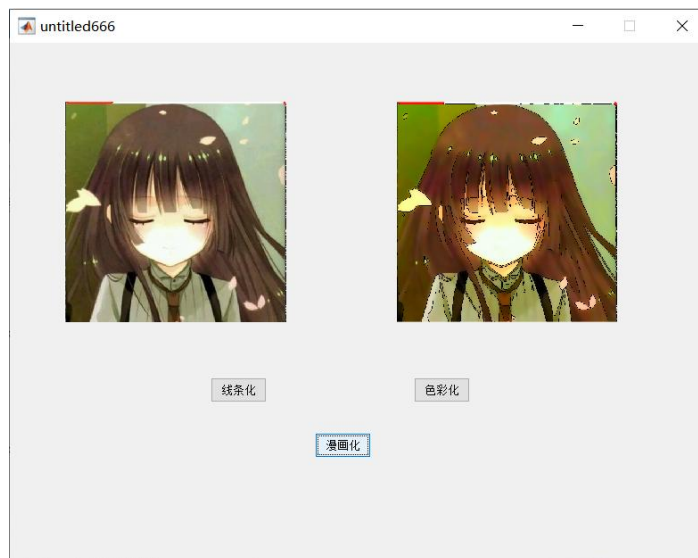


图 12 将漫画图进行漫画化示例

7. 总结

总体上,我们的程序通过以下三步把照片动漫化。首先,使用保边滤波进行区域平滑化,然后进行边缘检测,从原图中减去边缘像素,以突出显示边缘线条,最后进行色彩饱和度提升。我们使用了算法简单,处理速度快、同时兼具使用加权平均算法抑制对图像中的一些噪声功能的 Sobel 算子,缩短计算时间同时使边缘结果图更见简单。通过使用双边滤波器而非单纯的高斯滤波,缩小了滤波对边缘本身的影响,即只弱化边缘内部的高频信息而不会过多影响边缘。通过使用图像插值和外推增强了色彩对比度,使结果更加偏向漫画作品而非实际摄影作品。

本程序在输入图片本身是动漫图片时,其结果较差,因为其输入图片本身边缘的简洁及平滑程度已经满足漫画需求,对其进行更多操作反而会影响整体图片美观性。未来我们预期在判断输入图片风格上进行限制,若输入已经为漫画图片,则不会对其进行边缘线条检测。

8. 参考文献

- [1]. https://blog.csdn.net/qq_42333641/article/details/90199125
- [2]. 刘子奇, 刘世光. 风格线条画生成技术综述[J]. 计算机科学, 2019, 46(7): 13-21
- [3]. Gooch B, Reinhard E, Gooch A. Human facial illustrations: creation and psychophysical evaluation[J]. ACM Transactions on Graphics, 2004, 23(1): 27-44
- [4]. 罗光蕊, 苏鸿根. 基于小波多分辨率分析的线条画生成方法[J]. 计算机工程与设计, 2010, 31(17): 3848-3851
- [5]. 王少荣, 敖知琪, 要曙丽, 等. 自适应的线条画绘制[J]. 中国图象图形学报, 2018, 23(5): 730-739
- [6]. 范可悦, 刘世光. 肖像照片漫画风格化方法[J]. 计算机辅助设计与图形学学报, 2022, 34(1): 1-8. DOI:10.3724/SP.J.1089.2022.18819.
- [7]. 陈洪, 郑南宁, 徐迎庆, 等. 基于样本学习的人像线条画生成系统[J]. 软件学报, 2003, 14(2): 202-208.
- [8]. ZHANG, YONG, DONG, WEIMING, MA, CHONGYANG, et al. Data-Driven Synthesis of Cartoon Faces Using Different Styles[J]. IEEE Transactions on Image Processing, 2017, 26(1): 464-478. DOI:10.1109/TIP.2016.2628581.
- [9]. I AND E H, MCCANN J. Lightness and Retinex theory[J]. Journal of Optical Society of America, 1971, 61(1): 1-11.
- [10]. CHANG J L, IRENE CH, YI ZH, et al. Enhancement of low visibility aerial images using histogram truncation and an explicit Retinex representation for balancing contrast and color consistency[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 128: 16-26.
- [11]. 赵宏宇, 肖创柏, 禹晶, 等. 马尔科夫随机场模型下的 Retinex 夜间彩色图像增强[J]. 光学精密工程, 2014, 22(4): 1048-1055. DOI:10.3788/OPE.20142204.1048.
- [12]. 邵帅, 郭永飞, 刘辉, 等. HSI 色彩空间下的低照度遥感图像增强[J]. 光学精密工程, 2018, 26(8): 2092-2099. DOI:10.3788/OPE.20182608.2092.
- [13]. F. Marques, B. Llorens and A. Gasull, "Interpolation and extrapolation of image partitions using Fourier descriptors: application to segmentation-based coding schemes," Proceedings,

International Conference on Image Processing, 1995, pp. 584-587 vol.3, doi: 10.1109/ICIP.1995.537702.

- [14].https://blog.csdn.net/wenhao_ir/article/details/51743382
- [15].MATLAB 图像处理：把照片变成动漫风格- zhihu.com
- [16].郑欢欢,白鱼秀,张雅琼. 一种基于 Sobel 算子的边缘检测算法[J]. 微型电脑应用,2020,36(10):4-6. DOI:10.3969/j.issn.1007-757X.2020.10.003.
- [17].KUANG Z, CHEN Z. An effective multi—objective optimization spectrum allocation algorithm in cognitive wireless mesh networks [J]. Journal of Central South University: Science and Technology, 2013, 44(6): 2346—2353.
- [18].GONG M, CHEN X, MA L, et al. Identification of multi—resolution network structures with multi—objective immune algorithm[J]. Applied Soft Computing, 2013, 13(4): 1705—1717.
- [19].李知菲,陈源. 基于联合双边滤波器的 Kinect 深度图像滤波算法[J]. 计算机应用,2014,34(8):2231-2234,2242. DOI:10.11772/j.issn.1001-9081.2014.08.2231.
- [20].Image Processing By Interp and Extrapolation (sourceforge.net)
- [21].wyfunique/Image_cartoonlization (github.com)