

天津大学

2020 年信号与系统大作业



学院名称 智能与计算学部
专业 软件工程
姓名 陈姝宇
学号 3017218119
年级 2017 级
班级 3 班
同组人员 刘兴宇、乔雨菲

目 录

| | |
|-------------------------------|-----------|
| 1. 需求分析..... | 4 |
| 1.1 作业 1 系统需求..... | 4 |
| 1.2 作业 1 功能需求..... | 4 |
| 1.3 作业 2 系统需求..... | 4 |
| 1.4 作业 2 功能需求..... | 5 |
| 2. 作业 1——声音分离系统设计..... | 6 |
| 2.1 过程设计..... | 6 |
| 2.2 层次结构设计..... | 7 |
| 2.3 系统功能模块划分说明..... | 7 |
| 1) 读入声音信号模块..... | 8 |
| 2) 信号过滤分离模块..... | 8 |
| 3) 信号写入文件模块..... | 9 |
| 4) 播放声音文件模块..... | 9 |
| 5) 时域信号转频域模块..... | 10 |
| 2.4 系统开发..... | 11 |
| 3. 作业 2——图片转换系统设计..... | 17 |
| 3.1 过程设计..... | 17 |
| 3.2 逻辑层架构设计..... | 18 |

| | |
|----------------------------|-----------|
| 3.3 模块设计与实现..... | 18 |
| 4. 系统使用手册..... | 25 |
| 4.1 作业1——声音信号分离系统..... | 25 |
| 4.2 作业2——图片转换系统..... | 27 |
| 5. 系统测试..... | 29 |
| 5.1 测试目的..... | 29 |
| 5.2 测试环境..... | 29 |
| 5.3 作业1——声音信号分离系统功能测试..... | 30 |
| 5.4 作业2——图像转换系统测试..... | 33 |
| 5.4.3 测试结果..... | 34 |
| 附录..... | 38 |
| 附录1 人员分工..... | 38 |

1. 需求分析

1.1 作业 1 系统需求

设计一套声音分离系统，系统输入是一个两个不同频率声音混合的模拟信号，该信号在时域上是重叠的，系统需要判断混合声音中的两个不同声音的频率，并且将两个声音信号分裂，最后分别输出两个声音的模拟信号。

1.2 作业 1 功能需求

1. 拥有可视化操作界面。
2. 从本地选择混合音频文件，可以试听文件，并展示原始语音信号采样后的时域波形、频谱图。
3. 可以设置滤波器的阶数、截止频率，定义输出分离后语音文件的路径。
4. 可以试听分离的信号文件，展示分离信号的时域波形和频谱图。

1.3 作业 2 系统需求

设计一个系统，拥有以下两个功能：

1. 选择一张图片，进行傅里叶变换，给出变换后的频域图形。

2. 分别设计两个不同截止频率的低通滤波器和两个不同截止频率的高通滤波器，对图像进行处理，处理后进行傅里叶反变换，给出得到对应的图像。

1.4 作业 2 功能需求

1. 拥有可视化操作文件
2. 从本地选择源图片，展示源图片、图像频谱、幅度谱、相位谱
3. 选择高通/低通滤波器，自由设置截止频率
4. 使用滤波器转换后，展示源图像、未滤波的逆变换图像、滤波器和变换后的图像

2. 作业 1——声音分离系统设计

2.1 过程设计

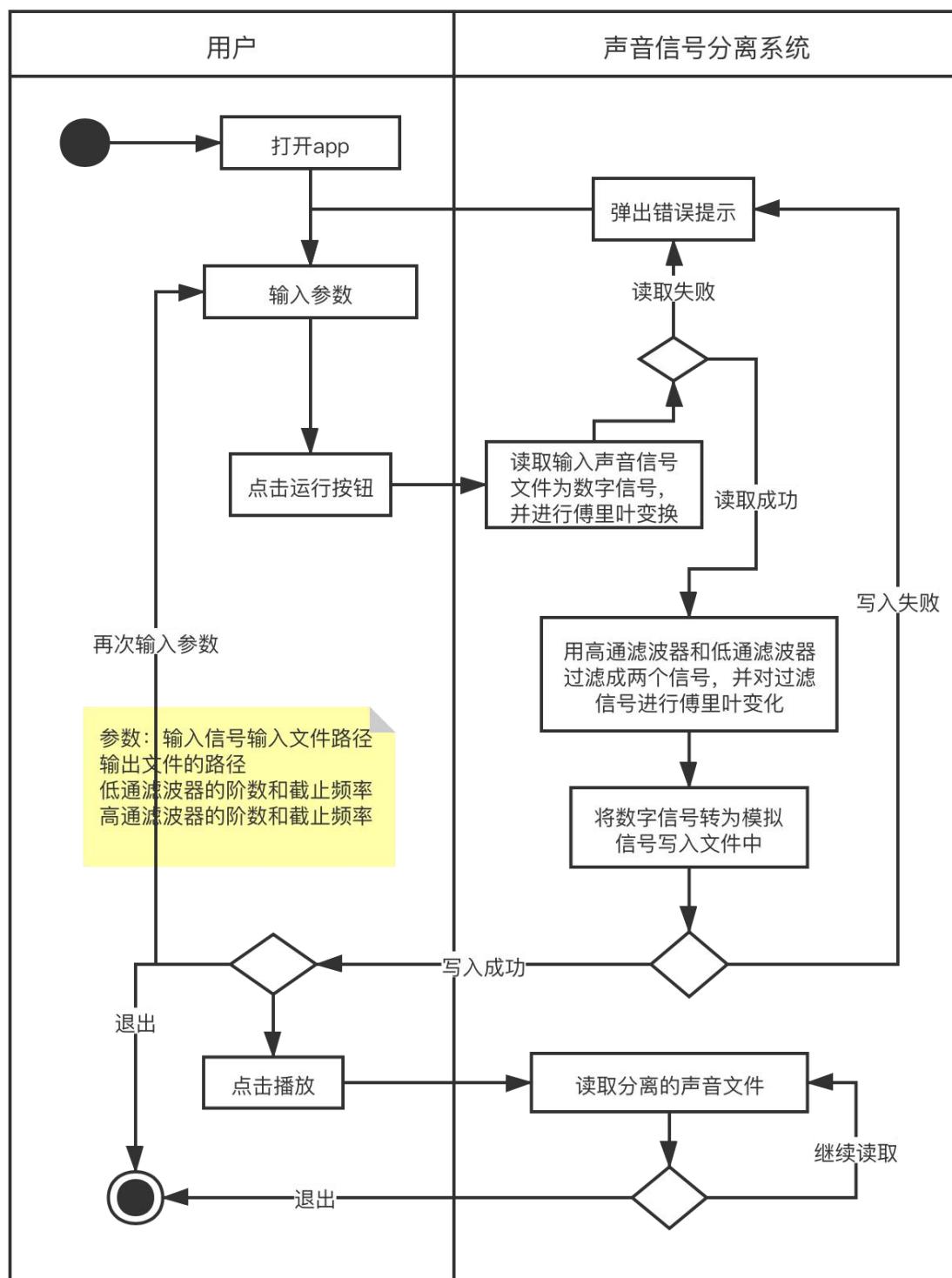


图 2-1 声音分离系统活动图

2.2 层次结构设计

根据声音信号分离系统的需求，将系统划分为读入声音信号、信号过滤分离、信号写入文件、播放声音文件、时域信号转频域、打开音频文件等多个模块。如图 2-2 所示。

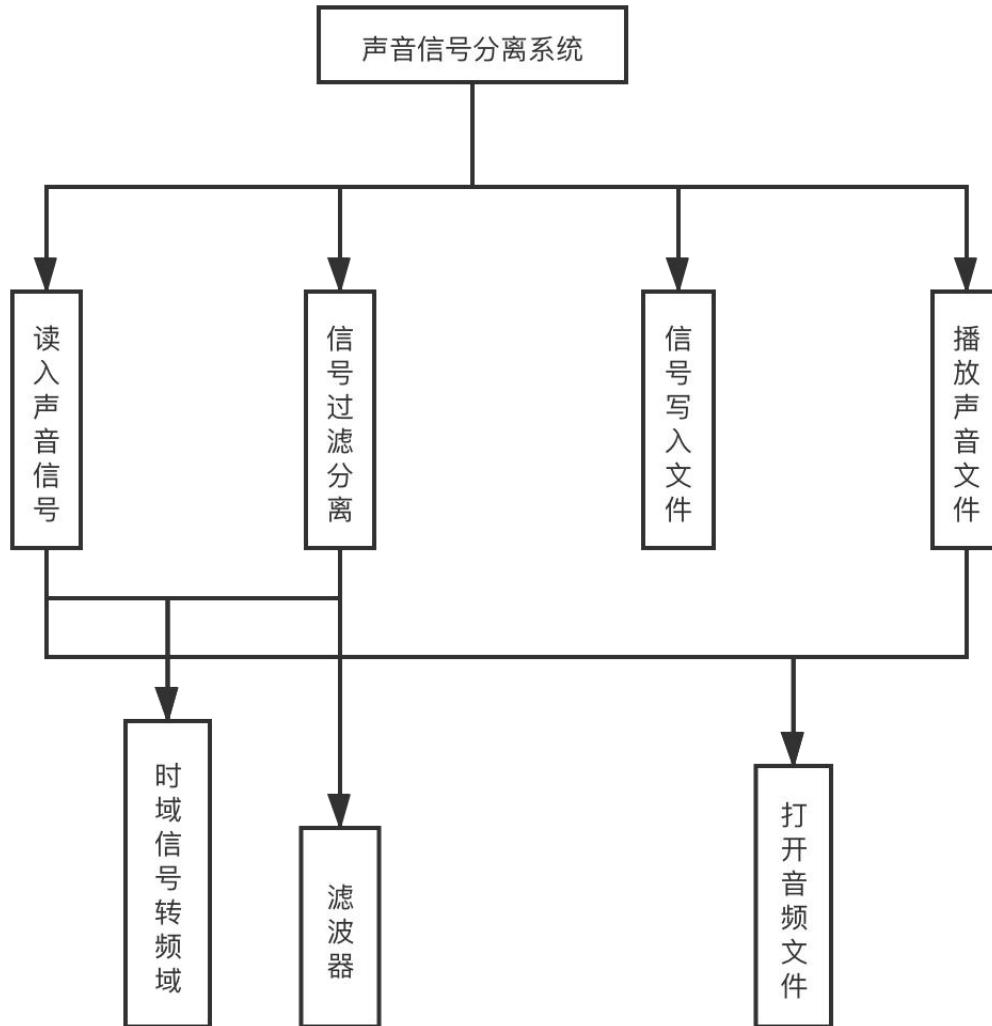


图 2-2 声音分离系统模块层次图

2.3 系统功能模块划分说明

对上图各个模块进行详细说明：

1) 读入声音信号模块

A. 描述：声音模拟信号读入为数字信号。包括打开音频文件和时域信号转频域信号。

B. 输入参数：

- file_name 文件名（待读取的声音模拟信号文件）

C. 输出结果：

- fs 采样频率（将模拟信号进行采样的频率）

- s 时域信号（模拟信号采样获得的时域数字信号）

- S 频域信号（获得的时域信号做傅里叶反变换对应的频域信号）

2) 信号过滤分离模块

A. 描述：时域重叠信号通过过滤分裂成信号。过滤原理为使用高通滤波器和低通滤波器，可以在调用模块的时候输入滤波器的阶数和截止频率的参数从而自定义调整滤波器。包括滤波器模块和时域信号转频域信号。

B. 输入参数：

- s （声音数字信号）

- N1 （低通滤波器阶数）

- wc1 （低通滤波器截止频率）

- N2 （高通滤波器阶数）

- wc2 （高通滤波器截止频率）

C. 输出结果：

- s1 输入信号经过低通滤波器过滤后的时域信号
- S1 输入信号经过低通滤波器过滤后的时域信号对应的频域信号
- s2 输入信号经过高通滤波器过滤后的时域信号
- S2 输入信号经过高通滤波器过滤后的时域信号对应的频域信号

3) 信号写入文件模块

- A. 描述：数字信号存为音频文件，数字信号需要根据原始声音模拟信号的采样率转换为声音模拟信号。
- B. 输入参数：
 - s1 分离出的时域信号 1
 - file_name1 分离出的信号 1 要写入的文件路径
 - s2 分离出的时域信号 2
 - file_name2 分离出的信号 2 要写入的文件路径
- C. 输出结果：
 - 异常：如果写入的文件路径不合法，将会抛出异常

4) 播放声音文件模块

- A. 描述：读取声音模拟信号文件并进行播放。
- B. 输入参数：
 - file_path 待读取的音频文件的路径
- C. 输出结果：
 - 声音

5) 时域信号转频域模块

- A. 描述：将时域信号做傅里叶变换变为频域信号，并画出频谱图
- B. 输入参数：
 - s 需要做傅里叶变换的时域信号
- C. 输出结果：
 - S 根据输入的时域信号做傅里叶变换得到的频谱信号

2.4 系统开发

2.4.1 开发环境

软件环境:

- 开发工具: MATLAB_R2020a
- 可视化界面开发工具: APP Designer of MATLAB
- 代码版本管理工具: GIT
- 项目管理工具: GitHub
- 绘图工具: ProcessOn

硬件环境:

- 操作系统: Windows / macOS

2.4.2 功能实现

下面将给出实现关键模块的主要方式:

1) 读入声音信号模块

`audioread(file_name)` 读取声音并得到采样频率

`x/max(abs(x))` 归一化

`fft` 时域信号转频域

将采样读取的声音信号放大, 可以得到如图 3-1 的时域信号坐标图。进行傅里叶变换后得到对应的频谱图。

```
% read original voice
function [s,fs,S] = read_voice(file_name)
[s ,fs] = audioread(file_name); % fs 是频率
s = s/max(abs(s)); % 归一化
S=t_convert_f(s); % 进行傅里叶变换到频域
end
```

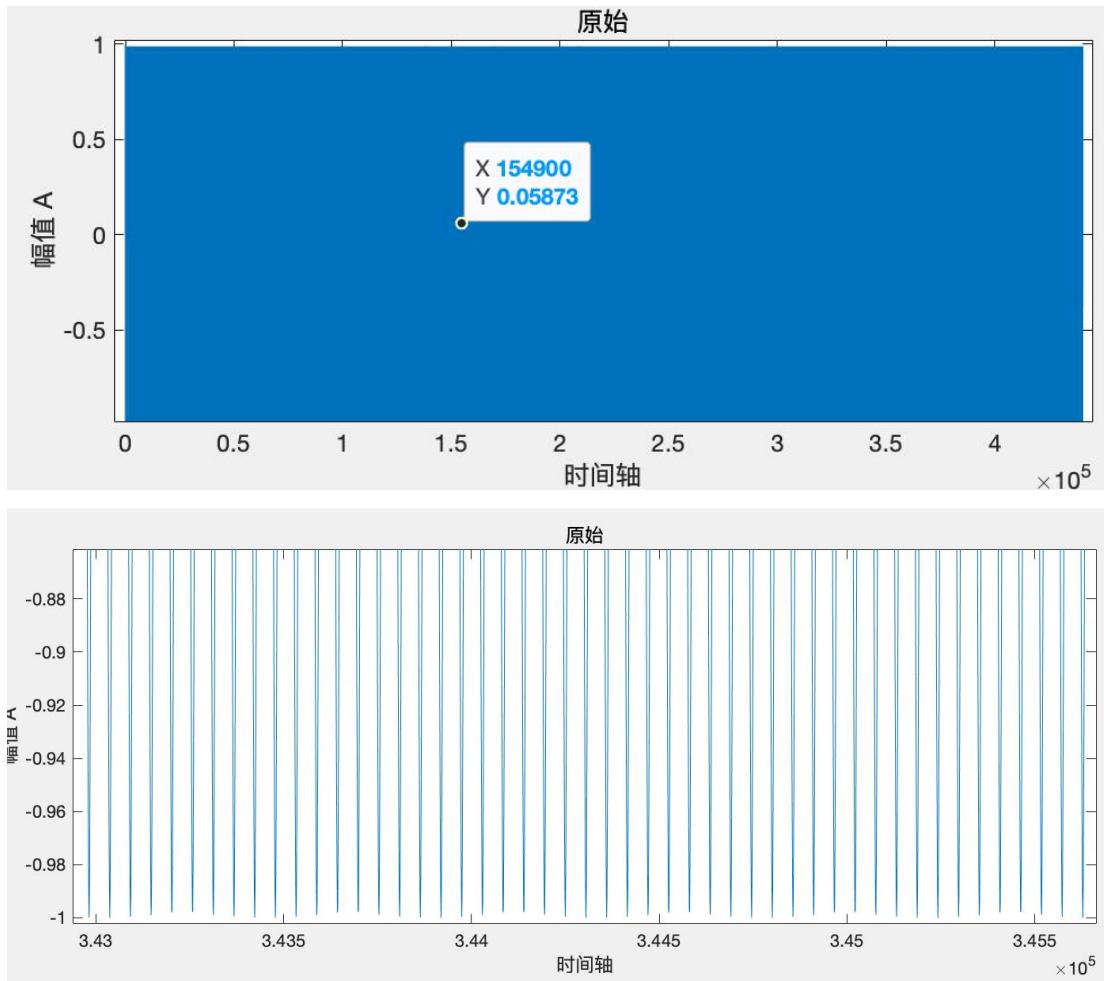


图 2-3 原始声音采样的时域信号图

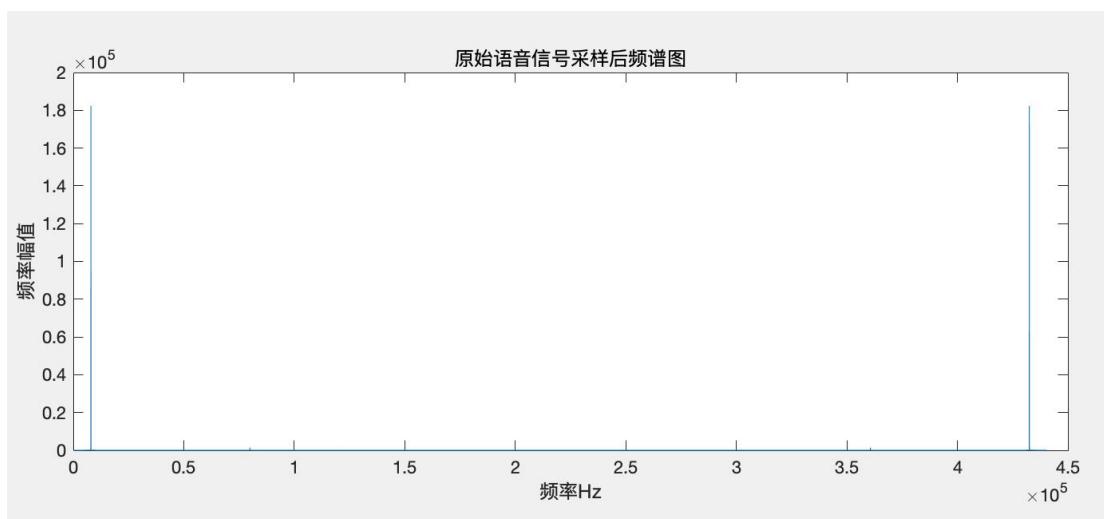


图 2-4 原始数字信号转成频域信号

2) 信号过滤分离模块

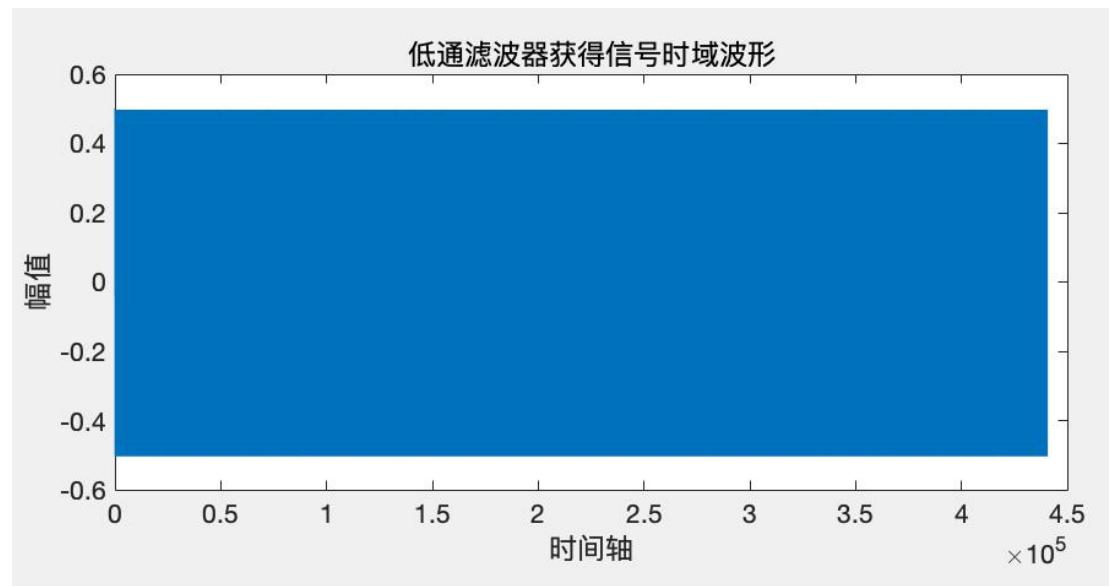
butter 滤波器系数

filter 根据滤波器系数对信号进行过滤

时域重叠信号使用高通滤波器和低通滤波器过滤出两个信号，可以在调用模块的时候输入滤波器的阶数和截止频率的参数从而自定义调整滤波器。获得两个信号如下图 2-5 所示。

```
% filter sound
function [h,z,H,Z] = filter_sound(y,N,wc,N2,wc2)
% 低通滤波器
[b,a]=butter(N,wc,'low');
h=filter(b,a,y);
H = t_convert_f(h);

% 高通滤波器
[b2,a2]=butter(N2,wc2,'high');
z=filter(b2,a2,y);
Z = t_convert_f(z);
```



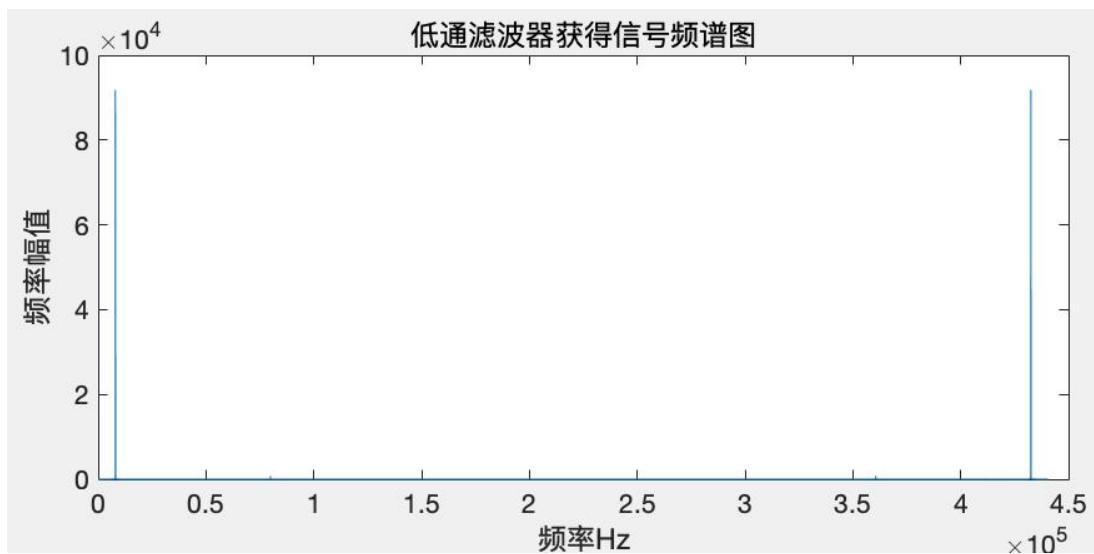


图 2-5 低通滤波器获得信号

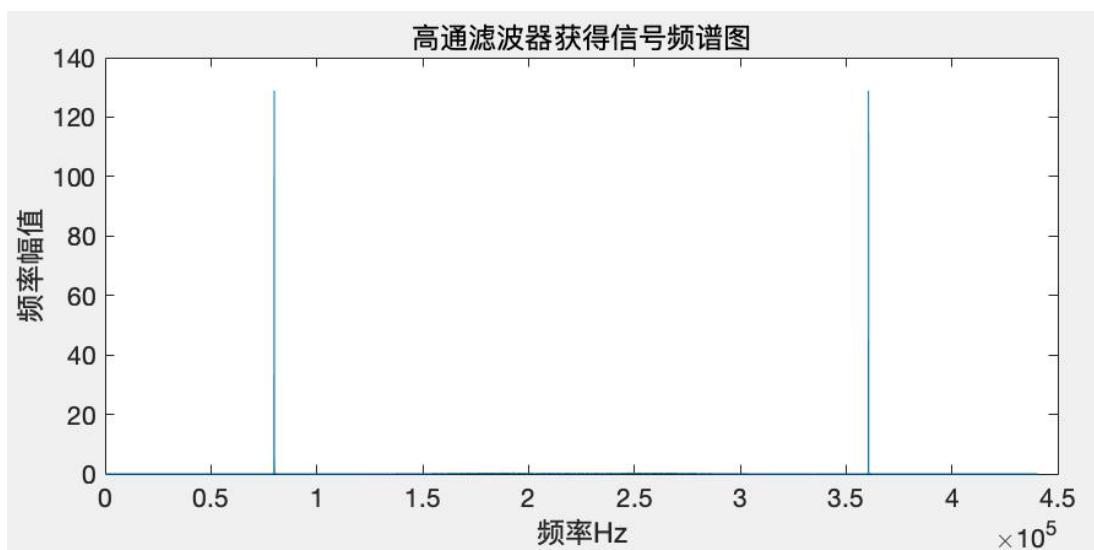
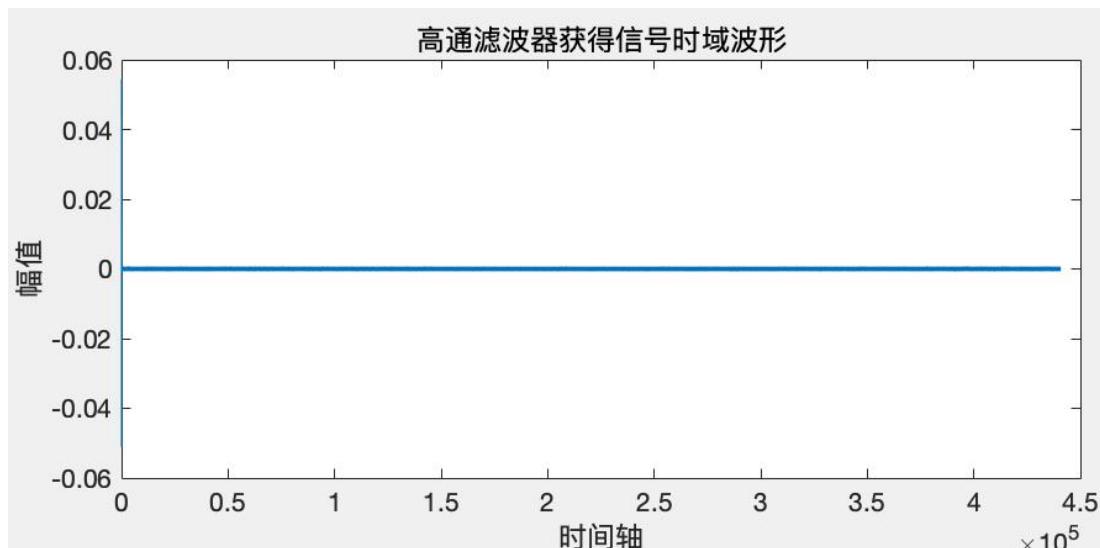


图 2-6 高通滤波器获得信号

3) 信号写入文件模块

audiowrite 数字信号存为音频文件, 数字信号需要根据原始声音模拟信号的采样率转换为声音模拟信号。

```
% write two voice
function [] = write_voice(h,s1_name,z,s2_name,fs)
try
    % 把数字信号转成模拟信号
    % write voice1
    audiowrite(s1_name,h,fs);
    % write voice2
    audiowrite(s2_name,z,fs);
catch ErrorInfo
    throw(ErrorInfo);
end
end
```

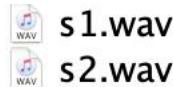


图 2-7 写入的音频文件

4) 播放声音验证

audioread 打开声音文件

sound 通过声音的数字信号和采样频率播放声音

```
% open file of voice and play
function []= open_sound(file_name)
% open file
[s ,fs] = audioread(file_name);
% play voice by mathematical signal and fs of voice
sound(s,fs);
end
```

5) 可视化界面

利用 MATLAB 的 App Designer 扩展实现系统功能的可视化系统。此系统功能详见 4 系统使用手册中的 4.1 作业 1——声音信号分离系统。

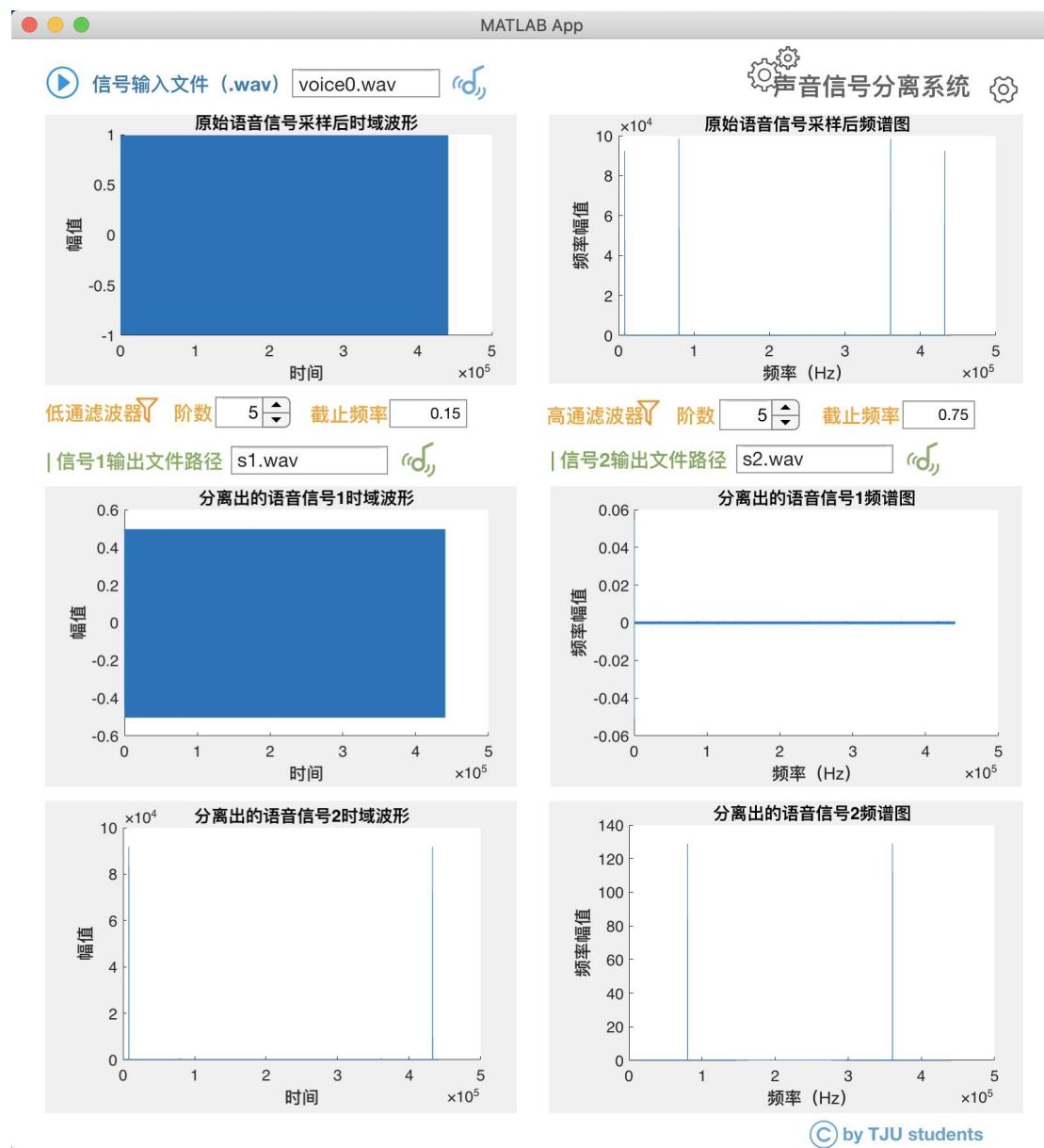


图 2-8 声音信号分离系统界面

3. 作业 2——图片转换系统设计

3.1 过程设计

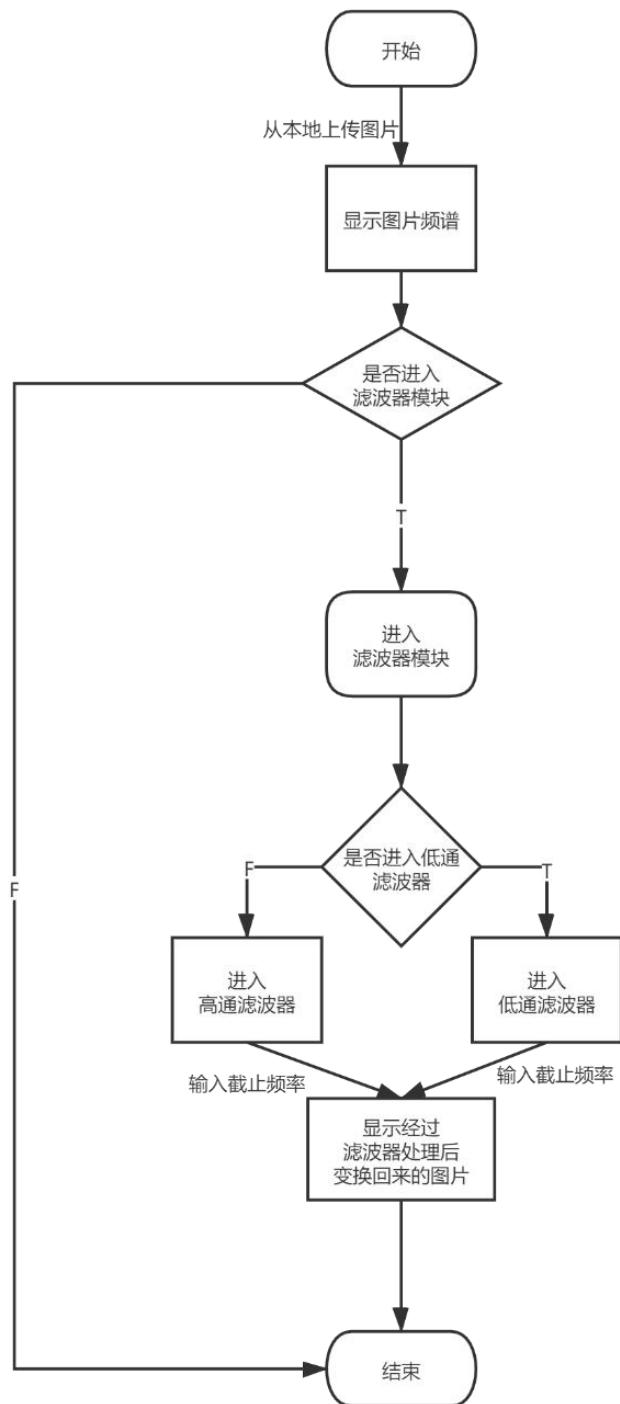


图 3-1 图片转换系统流程图

3.2 逻辑层架构设计

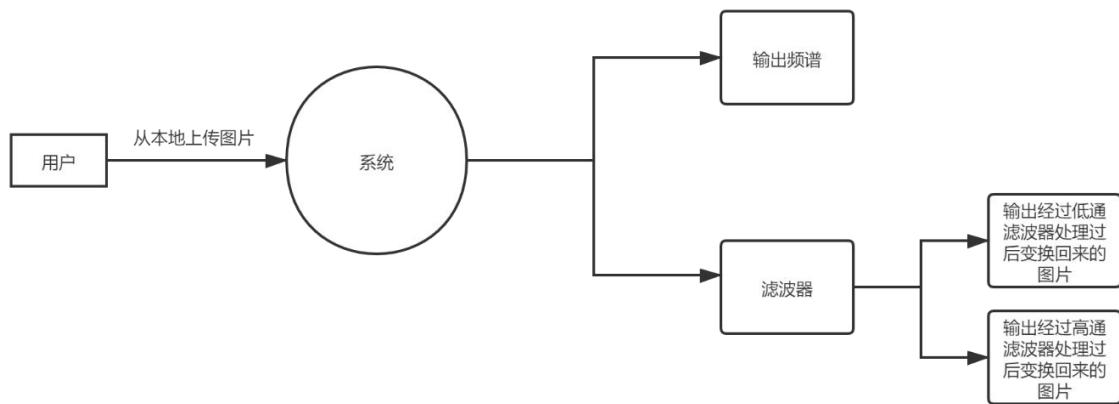


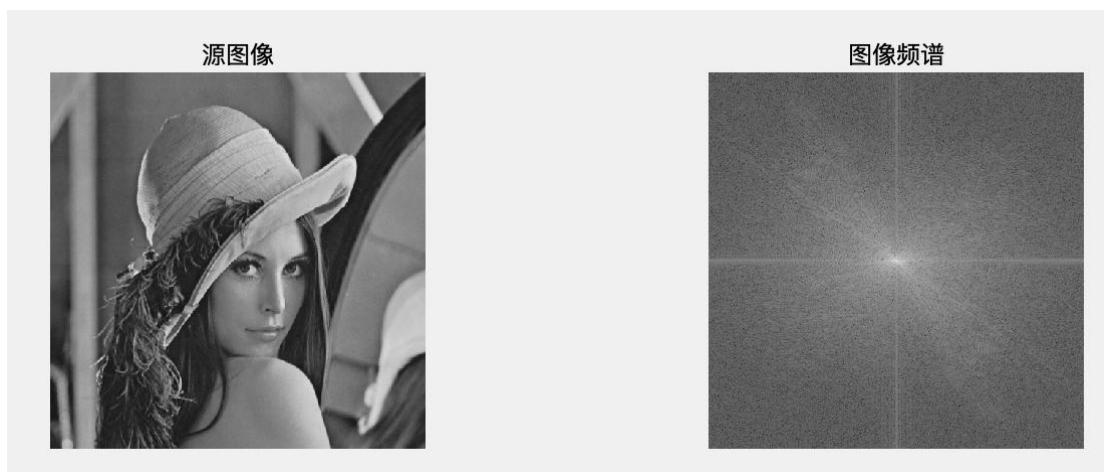
图 3-2 图片转换系统逻辑层架构设计

3.3 模块设计与实现

1. 对图像进行傅里叶变换

- (1) 先读取文件
- (2) 使用 fft2() 和 fftshift() 函数画出图像频谱

效果如图 3-3：



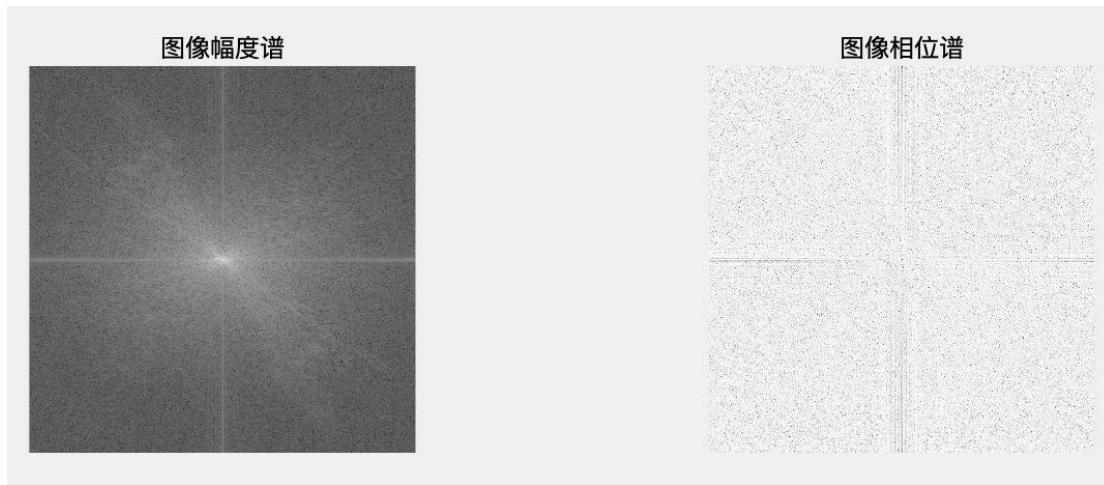


图 3-3 图像进行傅里叶变换效果图

2. 设计两个不同截止频率的低通滤波器

- (1) 先读取文件
- (2) 画出未滤波的逆变换图像

```
A=imread('image3.jpg');
subplot(3,2,1),imshow(A);
title('原图像');
A=rgb2gray(A);
B=fftshift(fft2(A));
E= ifft2(ifftshift(B));
subplot(3,2,2),imshow(log(abs(E)), [ ]);
title('未滤波的逆变换图像');
```

- (3) 设计两个不同频率的低通滤波器。

`freq` 为截止频率, `H1` 为大小和原图片大小相同的矩阵。通过处理变换 `H1`, 再让经过傅里叶变换的 `B` 和 `H1` 相乘。`LOWPASS1` 即为第一个低通的频谱。可以自己更改 `freq` 截止频率。

```
[m,n]=size(A);
%m1=m/2;n1=n/2;
H1=ones(m,n);
freq=80;
%H1(m1-freq:m1+freq, n1-freq:n1+freq)=1;
```

```

for i=1:m
    for j=1:n
        temp=(i - (n - 1) / 2) ^ 2 + (j - (m - 1) / 2) ^ 2;
        if temp >= freq ^ 2
            H1(i, j) = 0;

        end
    end
end
LOWPASS1=B.* H1;
size(LOWPASS1)

```

设计低通滤波器原理:

在频谱中，低频主要对应了图像在平滑区域的总体灰度级分布，而高频对应图像的细节部分例如边缘噪声。因此图像平滑可以通过衰减图像频谱中的高频部分来实现，这就建立起了空间域图像平滑和频域低通滤波之间的对应关系。最直接的方法就是设置一个截止频率，高于此频率的频谱成分置为 0，低于的保存，这就是理想的低通滤波器，如果图像的宽为 M,高为 N,其 $H(u,v)$ 的表现形式为。

$$H(u, v) = \begin{cases} 1, & [(u - \frac{M}{2})^2 + (v - \frac{N}{2})^2] \leq D_0^2 \\ 0, & [(u - \frac{M}{2})^2 + (v - \frac{N}{2})^2] > D_0^2 \end{cases}$$

(4) 处理后进行傅里叶反变换，得到对应图像。

```

C=ifft2(ifftshift(LOWPASS1));

subplot(3,2,3),imshow(log(abs(LOWPASS1)), [ ]);
title('低通1');
subplot(3,2,4),imshow(log(abs(C)), [ ]);
title('低通1变换图像');

```

(5) 描述和分析不同截止频率的低通滤波器对图像的处理效果，分析不同频率在图片中的表现。

截止频率为 80:

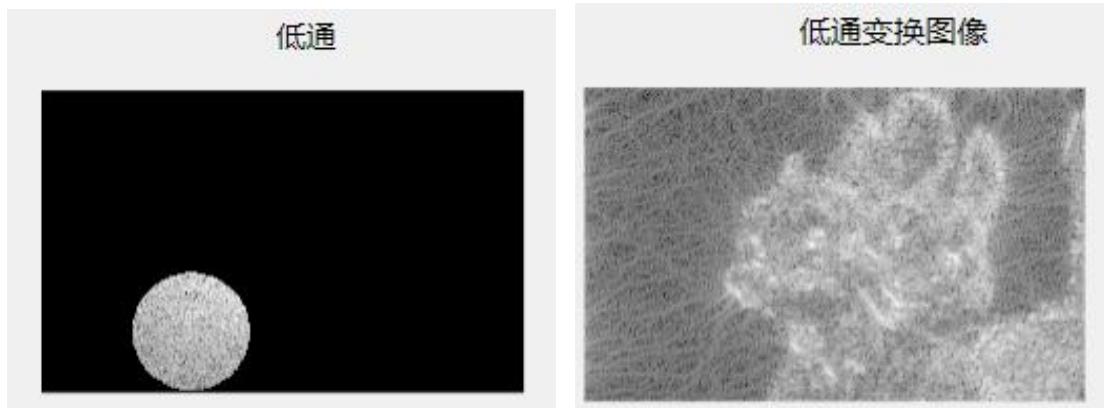


图 3-4 截止频率为 80 的效果图

截止频率为 30:

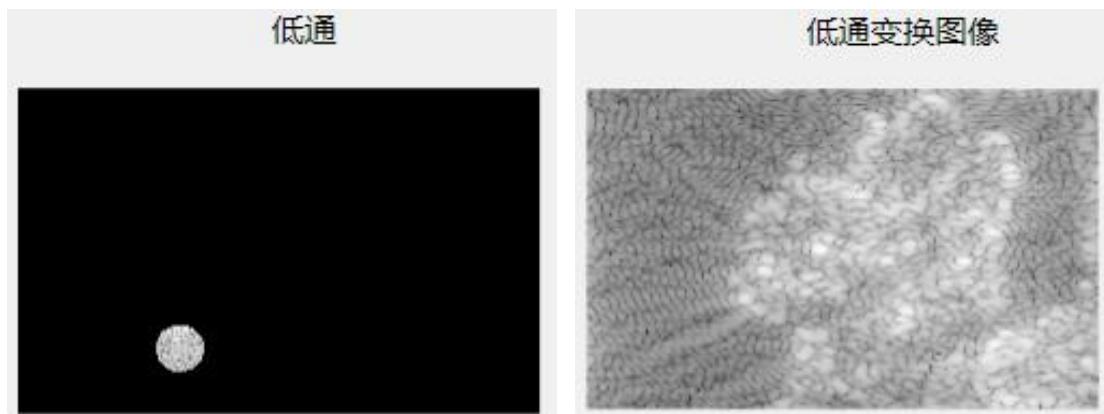


图 3-5 截止频率为 30 的效果图

可以看到第一个截止频率为 80，而第二个截止频率为 30，低通滤波器保留低频率波，随着截止频率的减小，经过傅里叶反变换的图像更加模糊。

3. 设计两个不同截止频率的高通滤波器

(1) 先读取文件

(2) 画出未滤波的逆变换图像

```
A=imread('image3.jpg');
subplot(3,2,1), imshow(A);
title('原图像');
A=rgb2gray(A);
```

```

B=fftshift(fft2(A));
B1=B;
E= ifft2(ifftshift(B));
subplot(3,2,2),imshow(log(abs(E)), [ ]);
title('未滤波的逆变换图像');

```

(3) 设计两个不同频率的高通滤波器。

freq 为截止频率，通过处理变换 B，再对 B 做傅里叶变换。可以自己更改 freq 截止频率。

```

freq=90;

for i=1:m
    for j=1:n
        temp=(i - (m - 1) / 2) ^ 2 + (j - (n - 1) / 2) ^ 2;
        if temp < freq ^ 2
            B(i, j) = 1;
        end
    end
end

```

(4) 处理后进行傅里叶反变换，得到对应图像。

```

C=ifft2(ifftshift(B));
subplot(3,2,3),imshow(log(abs(B)), [ ]);title('高通1');
subplot(3,2,4),imshow(log(abs(C)), [ ]);title('高通1变换图像');

```

(5) 描述和分析不同截止频率的低通滤波器对图像的处理效果，分析不同频率在图片中的表现。

截止频率为 90:

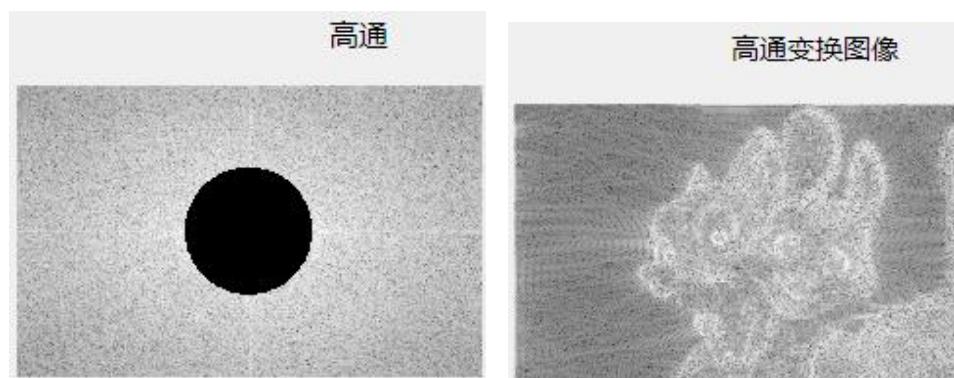


图 3-6 截止频率为 90 的效果图

截止频率为 10:

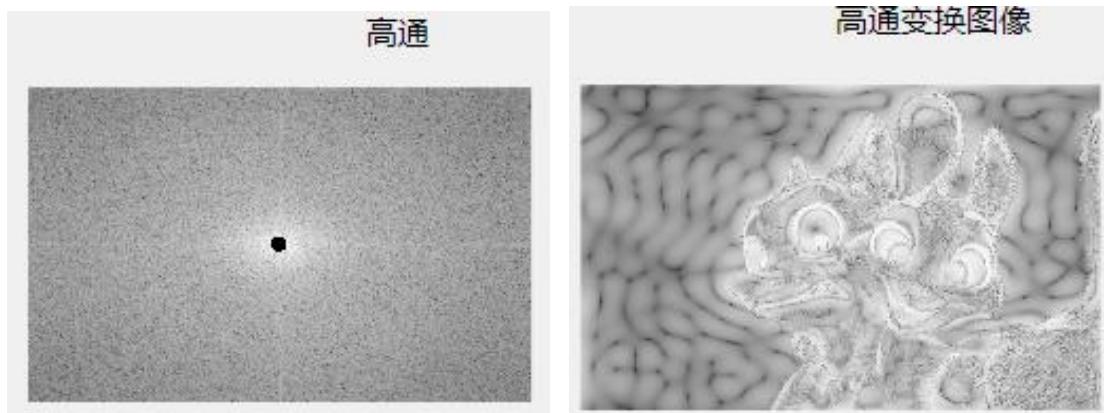


图 3-4 截止频率为 10 的效果图

我们可以看到第一个截止频率为 90, 而第二个截止频率为 10, 随着截止频率的减小, 经过傅里叶反变换的图像更加清晰。高通滤波器保留高频率, 显示的是图片的轮廓。

注: 高通低通处理过后的 4 张图片位于 system2 中的 result 文件夹中。

(6) 可视化界面

此系统功能详见 4 系统使用手册中的 4.2 作业 2——图片转换系统。



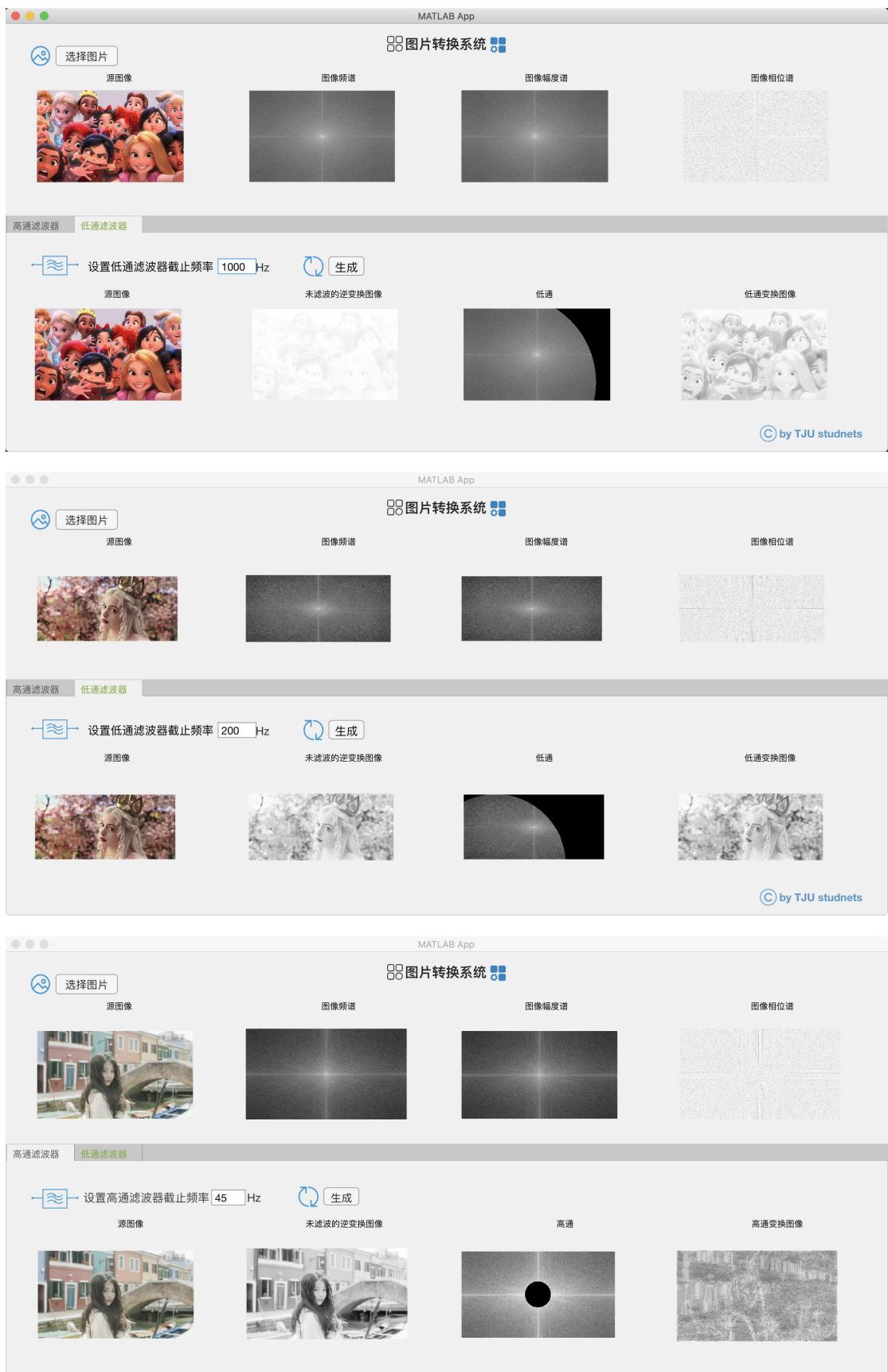


图 3-5 图片转换系统界面

4. 系统使用手册

4.1 作业 1——声音信号分离系统

系统界面：

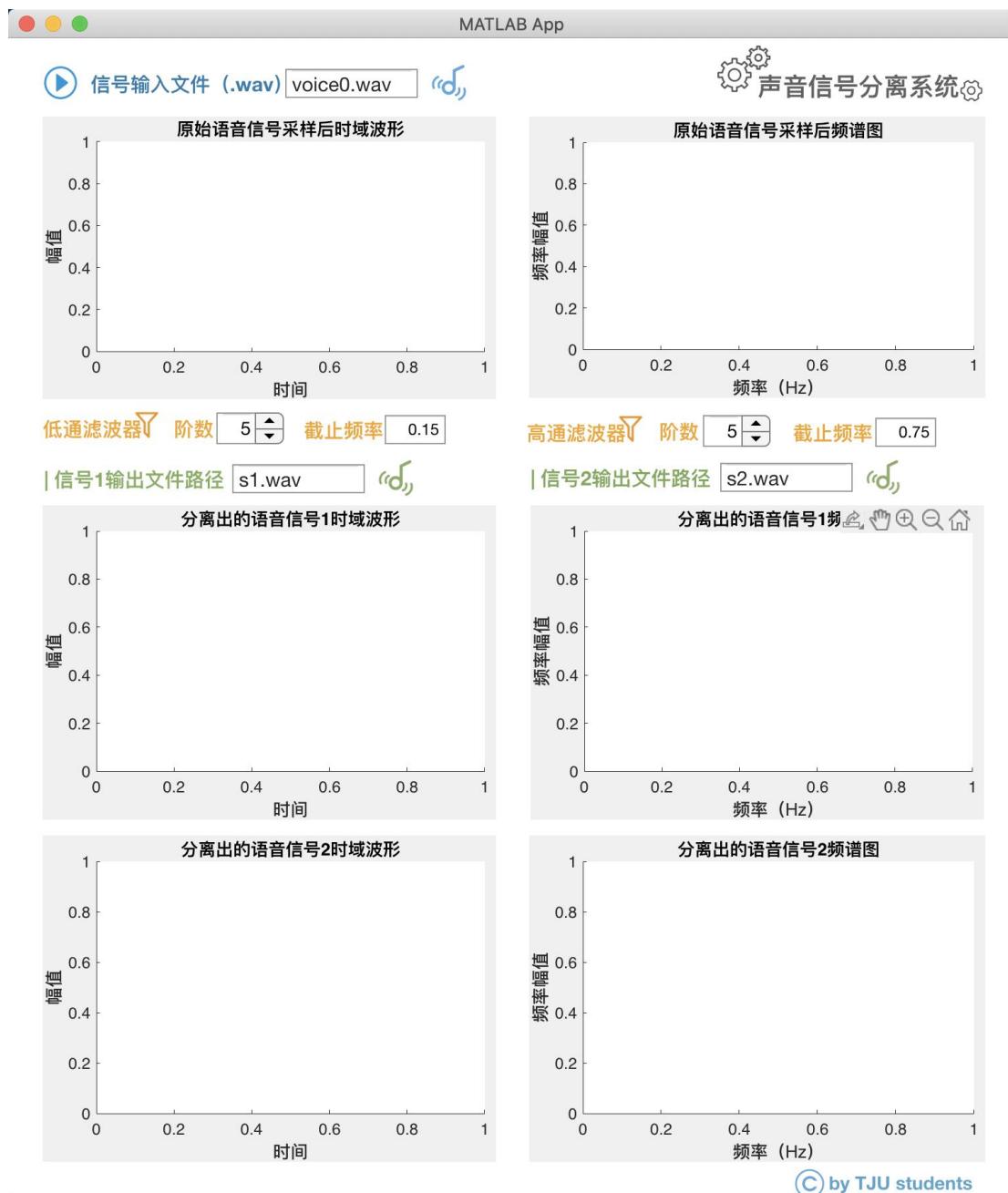


图 4-1 声音信号分离系统界面

1. 选择输入音频文件：在上方输入框输入音频文件相对于程序根

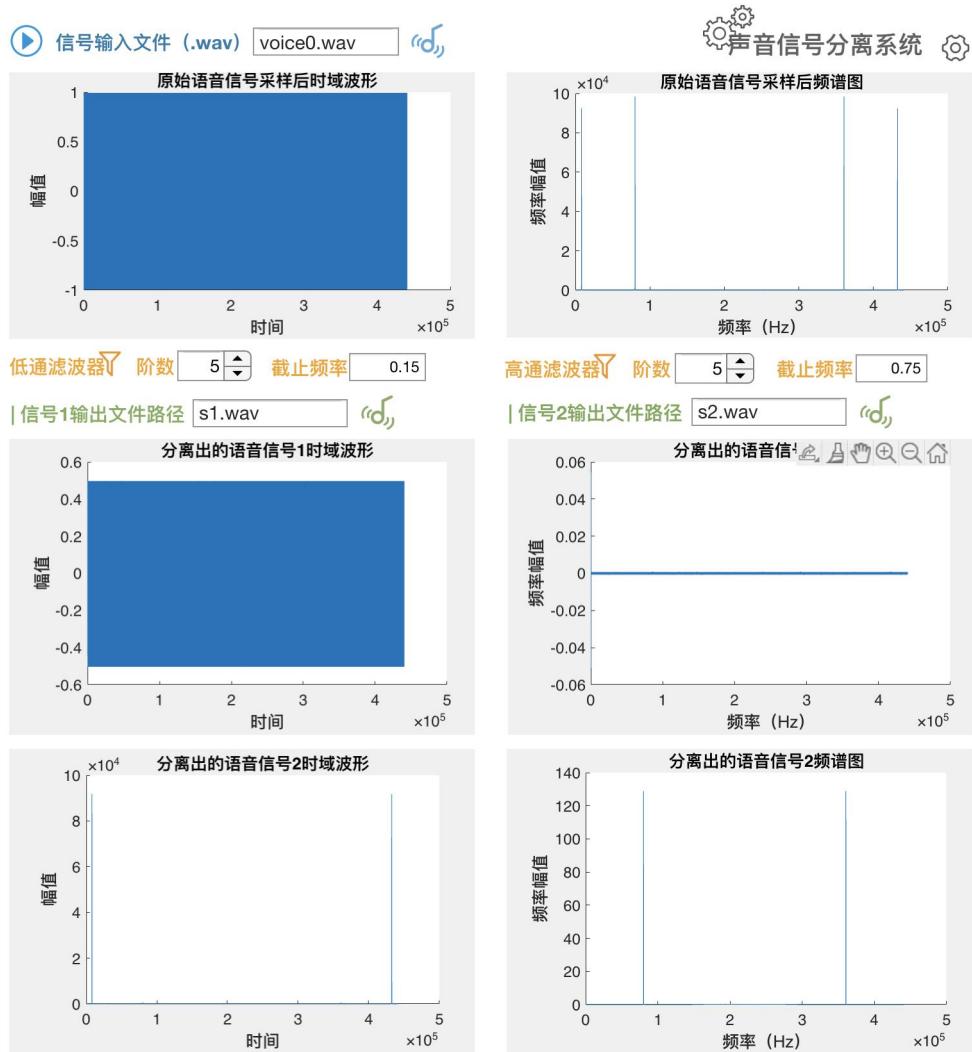
目录的相对路径。点击右侧音符可以试听输入音频文件。

2. 设置滤波器参数：设置滤波器阶数和截止频率。

低通滤波器 阶数 截止频率 高通滤波器 阶数 截止频率

3. 设置分离信号的输出文件路径：在输入框中输入输出文件相对于程序根目录的相对路径。

4. 点击左上角播放按钮，开始转换，分别显示原信号波形和结果。
输入路径、输出路径和滤波器参数均有默认值。点击输出文件路径旁边音符可以试听输出音频。



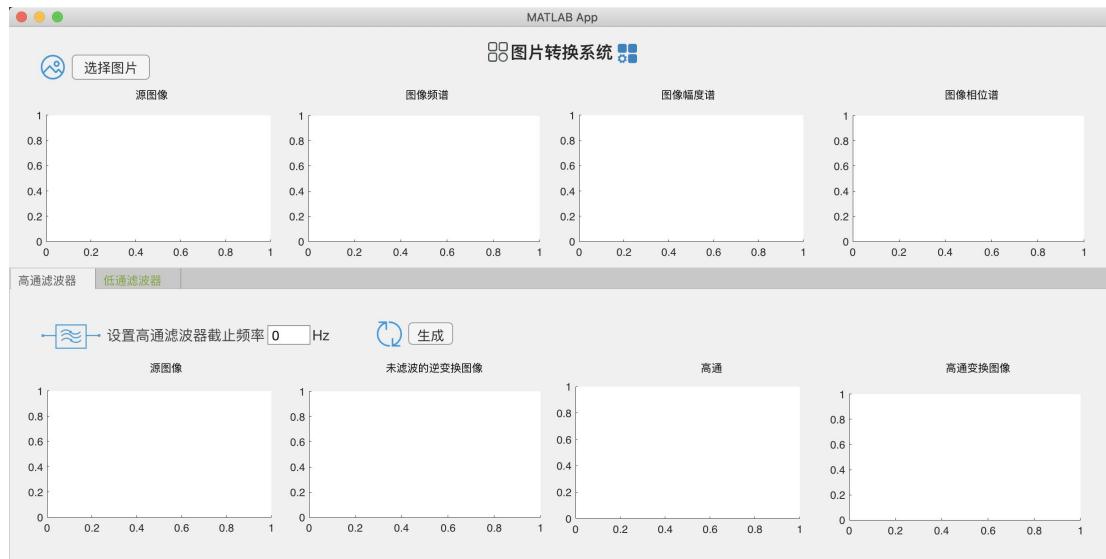
© by TJU students

图

4-2 声音信号分离系统使用效果图

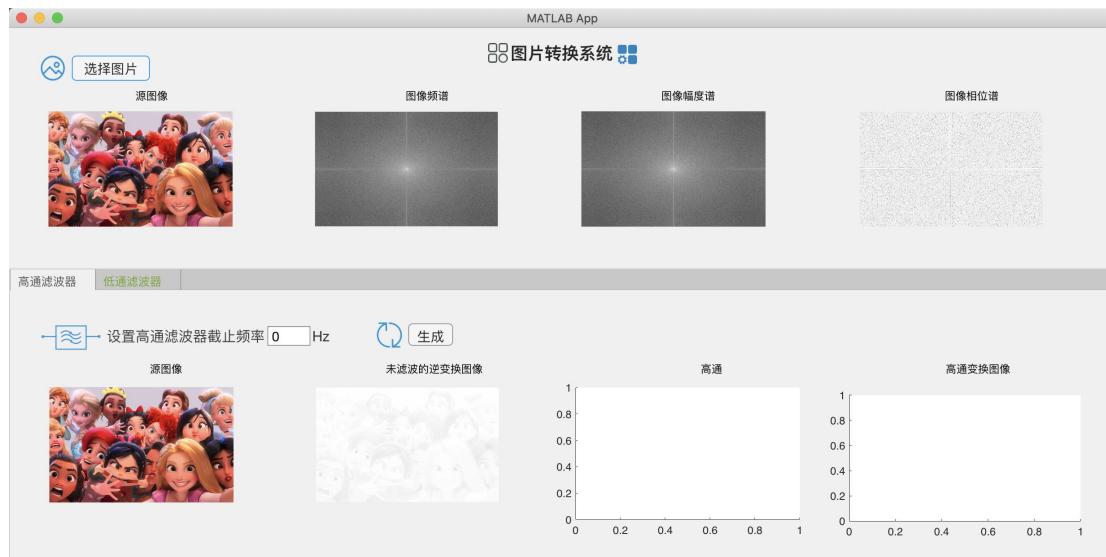
4.2 作业 2——图片转换系统

系统界面：



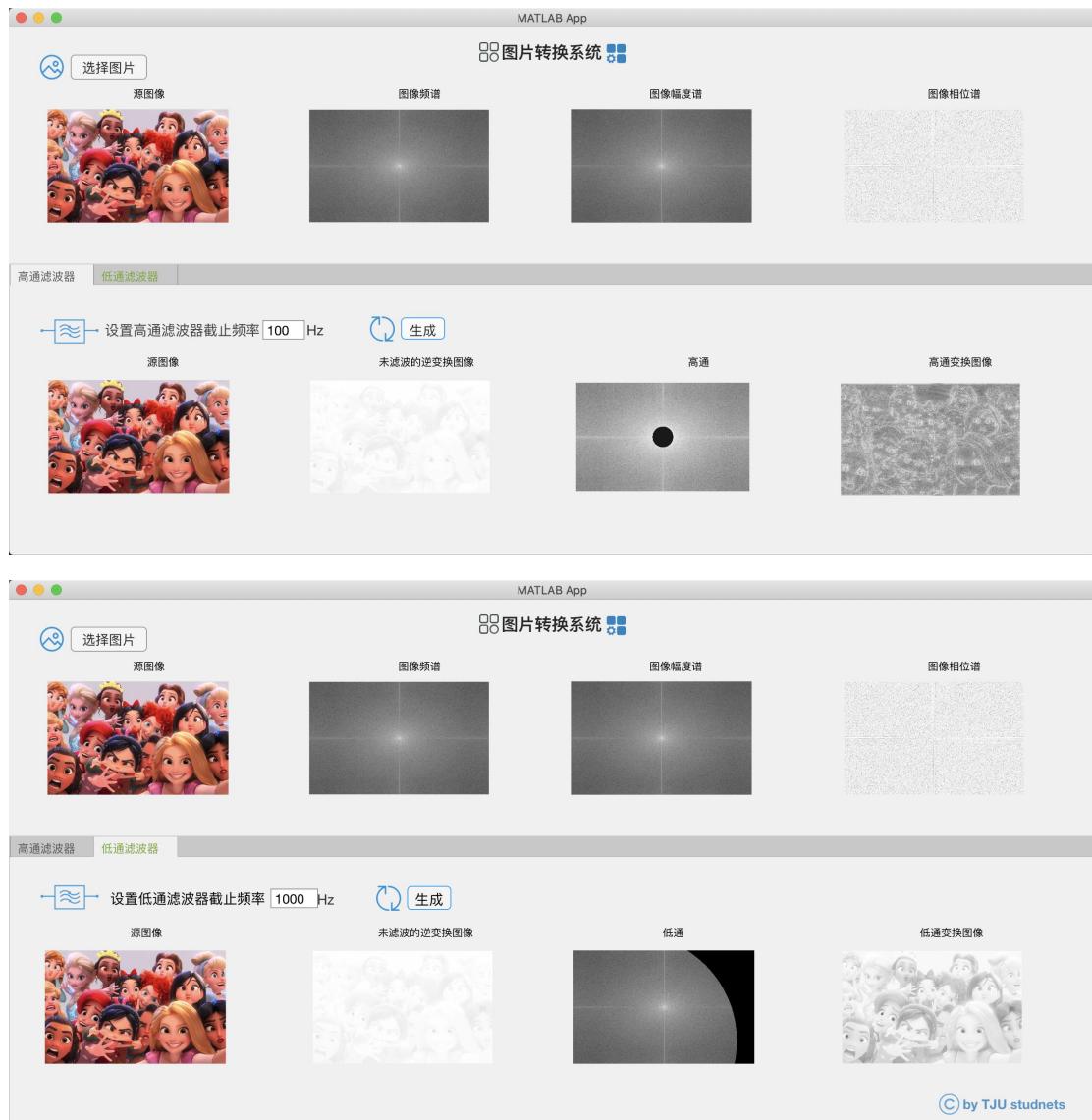
4-3 图片转换系统界面

1. 选择图片：点击左上方“选择图片”按钮，从本地选择需要转换的图片，点击确定后，会自动展示源图像、图像频谱、图像幅度谱、图像相位谱、未滤波的逆变换图像



4-4 图片转换系统使用效果图

2. 切换高通/低通滤波器：点击下方选项卡切换高通、低通滤波器
3. 滤波器转换：输入截止频率，点击生产，可以生成滤波器和变换图像



4-5 滤波器使用效果图

4. 更换图片：无论何时，点击“选择图片”，重新上传图片可以更换图片。

5. 系统测试

5.1 测试目的

1. 测试系统能否正常读取输入
2. 测试系统是否正确完成需求规格中的功能
3. 测试系统输出的正确性、输出文件路径是否正常
3. 测试系统资源占用情况、是否存在崩溃等运行时错误

5.2 测试环境

1. 硬件环境

- 计算机: DELL Inspnion 5468
- CPU: Intel i7 7500U 2.9GHz
- RAM: 4GB
- 磁盘: 256GB SSD
- 显示卡: AMD Readon R7 M340
-

2. 软件环境

- 操作系统: WINDOWS 10 64-bit 专业版
- 软件: MATLAB R2020a 试用版

5.3 作业 1——声音信号分离系统功能测试

5.3.1 被测特性:

| 序号 | 被测需求 |
|-----|--------------------|
| 001 | 选择输入文件 |
| 002 | 使用默认滤波器设置、输出路径运行转换 |
| 003 | 修改滤波器参数 |
| 004 | 输出文件路径 |

5.3.2 测试用例设计

➤ 001: 选择输入文件

在上方输入音频文件名: voice2.wav, 点击右侧试听文件是否成功输入。

➤ 002: 使用默认滤波器设置、输出路径运行转换

在 001 的基础上, 点击左上角运行, 观察六个输出图像, 并点击输出文件路径旁边试听输出。

➤ 003: 修改滤波器参数

修改滤波器 1 阶数为 4, 截止频率为 0.1, 滤波器 2 阶数为 6, 截止频率为 0.8, 点击左上角运行, 观察输出图像、试听输出。

➤ 004: 修改输出文件路径

在 001 基础上, 修改输出文件路径为 s3.wav, s4.wav 点击左上角运行, 观察是否生成正确的音频文件。

5.3.3 测试结果:

➤ 001：选择输入文件



声音播放正常

➤ 002：使用默认滤波器设置、输出路径运行转换

输出声音播放正常

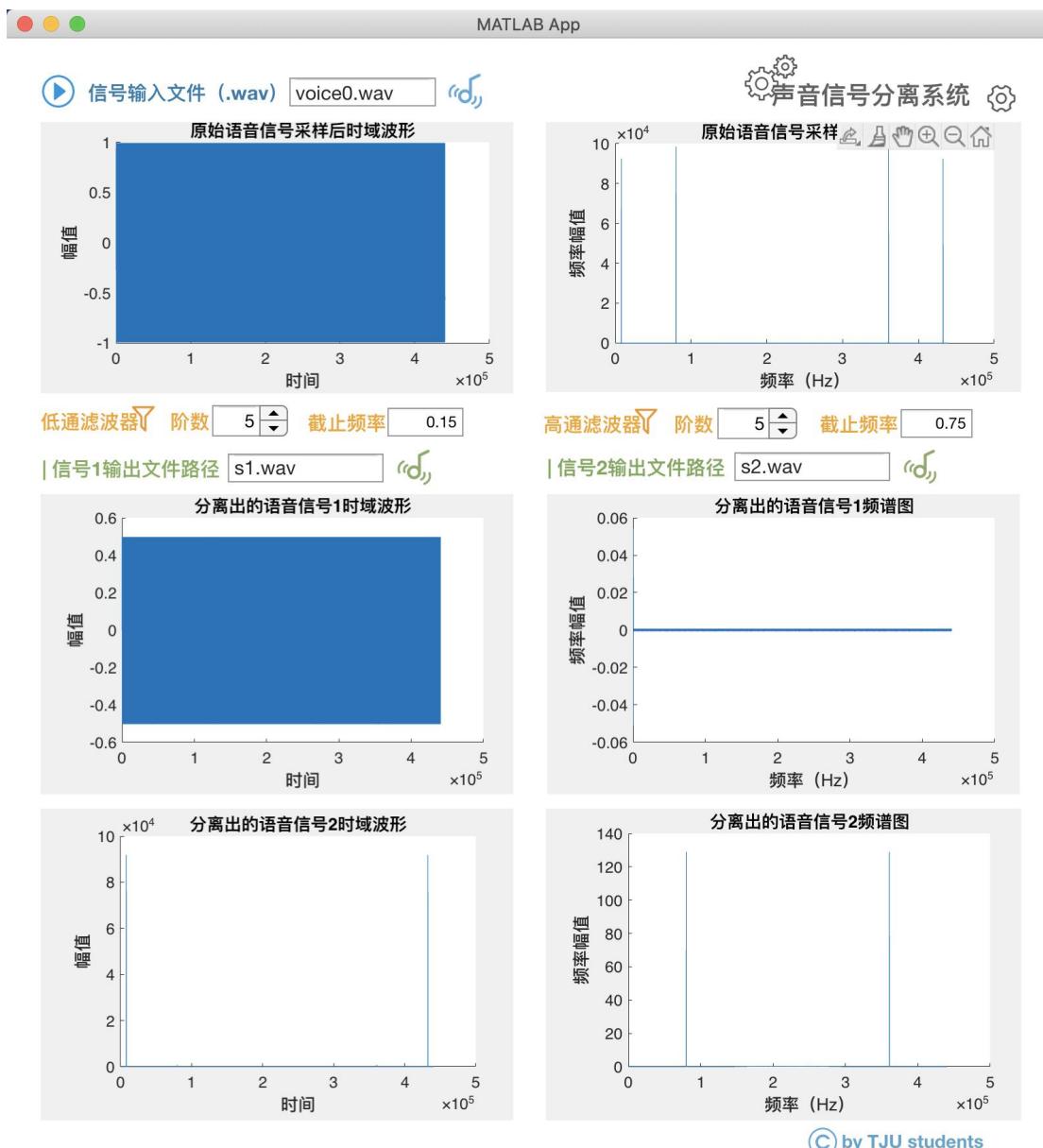


图 5-1 声音信号分离系统测试效果 1

➤ 003: 修改滤波器参数

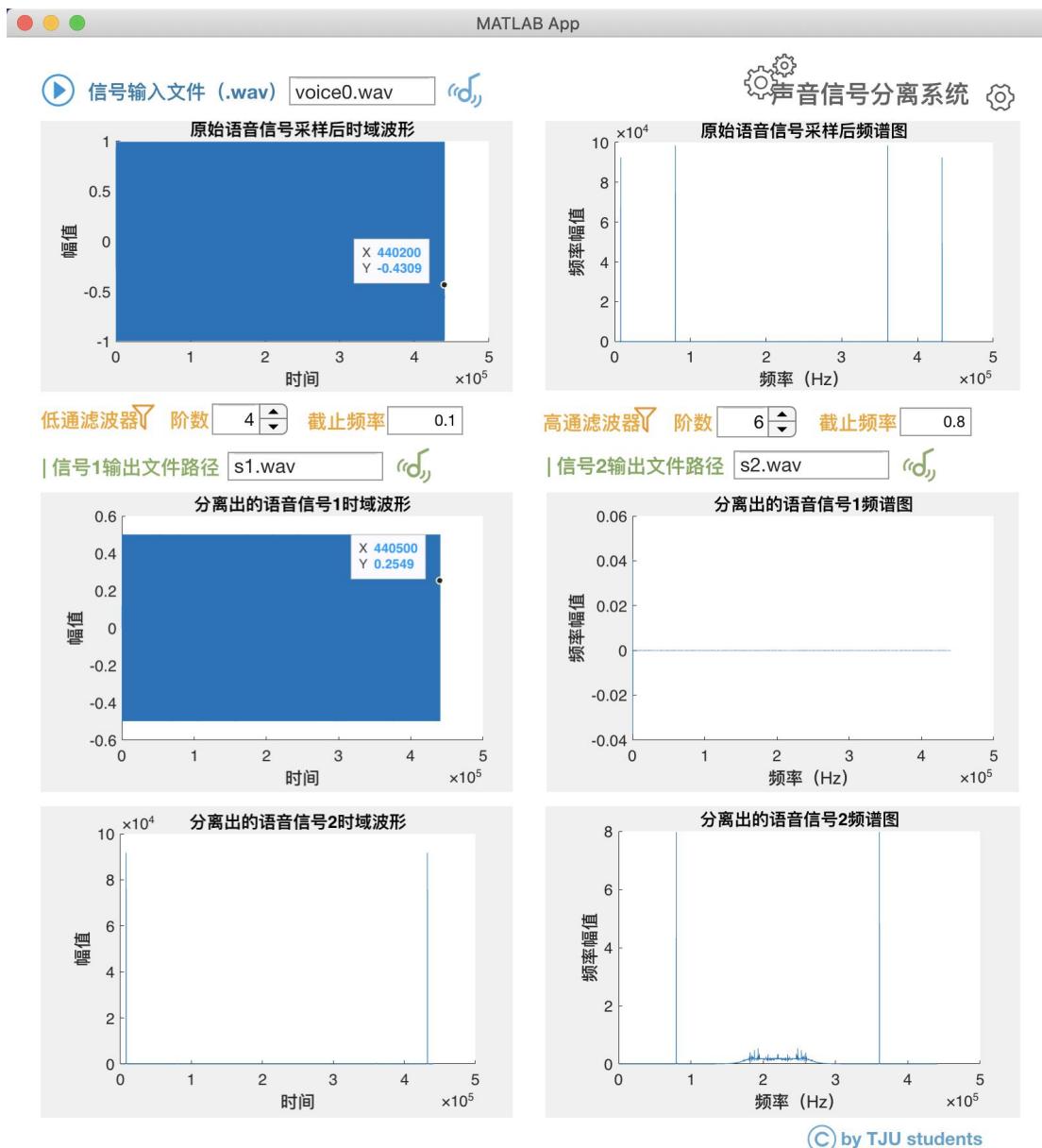


图 5-2 声音信号分离系统测试效果 2

➤ 004: 修改输出文件路径:

| 信号1输出文件路径 | s3.wav | 信号2输出文件路径 | s4.wav |
|-----------|-------------------------|-----------|--------|
| s3.wav | 2020/4/1 星期三 22:12 音频文件 | s4.wav | 861 KB |
| s4.wav | 2020/4/1 星期三 22:12 音频文件 | | 861 KB |

成功输出音频文件

5.4 作业 2——图像转换系统测试

5.4.1 被测特性

| 序号 | 被测需求 |
|-----|--------------|
| 001 | 选择输入图片 |
| 002 | 使用默认高通滤波器转换 |
| 003 | 使用默认低通滤波器转换 |
| 004 | 修改高通滤波器频率后转换 |
| 005 | 修改低通滤波器频率后转换 |
| 006 | 修改输入图片 |

5.4.2 测试用例设计

➤ 001: 选择输入图片

点击左上角选择图片按钮，选择一张图片，点击确定，观察生成的六个图像。

➤ 002: 使用默认高通滤波器转换

在 001 基础上，选项卡选择高通滤波器，点击生成，观察生成的图像

➤ 003: 使用默认低通滤波器转换

在 001 基础上，选项卡选择低通滤波器，点击生成，观察生成的图像

➤ 004: 修改高通滤波器频率后转换

在 002 基础上，修改截止频率为 40Hz，观察图像的变化

- 005：修改低通滤波器频率后转换

在 003 基础上，修改截止频率为 1500Hz，观察图像的变化

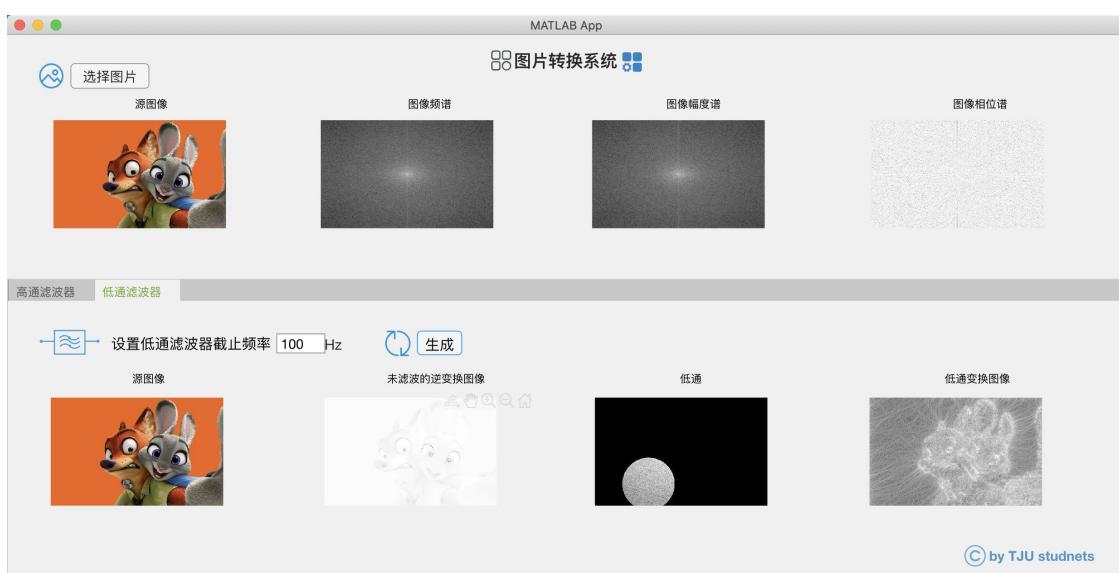
- 006：修改输入图片

在 004 基础上，点击选择图片按钮，重新选择一张图片，观察图
像是否改变

5.4.3 测试结果

- 001：选择输入图片

点击选择图片，选择 image3.jpg，点击确定



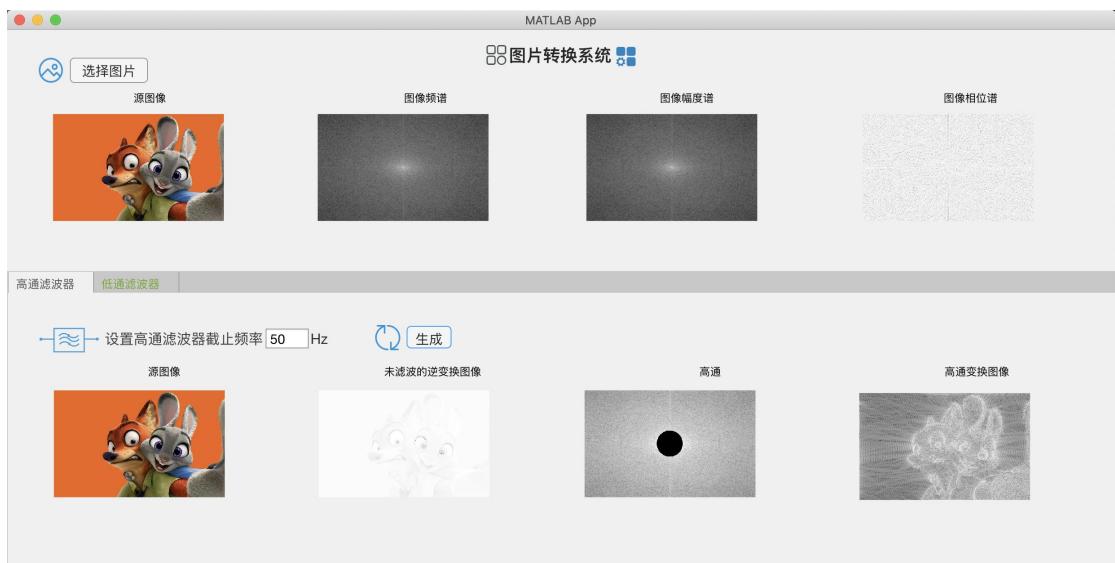


图 5-3 图像转换系统测试效果 1

➤ 002：使用默认高通滤波器转换

选择高通滤波器，使用默认频率 100Hz，点击生成

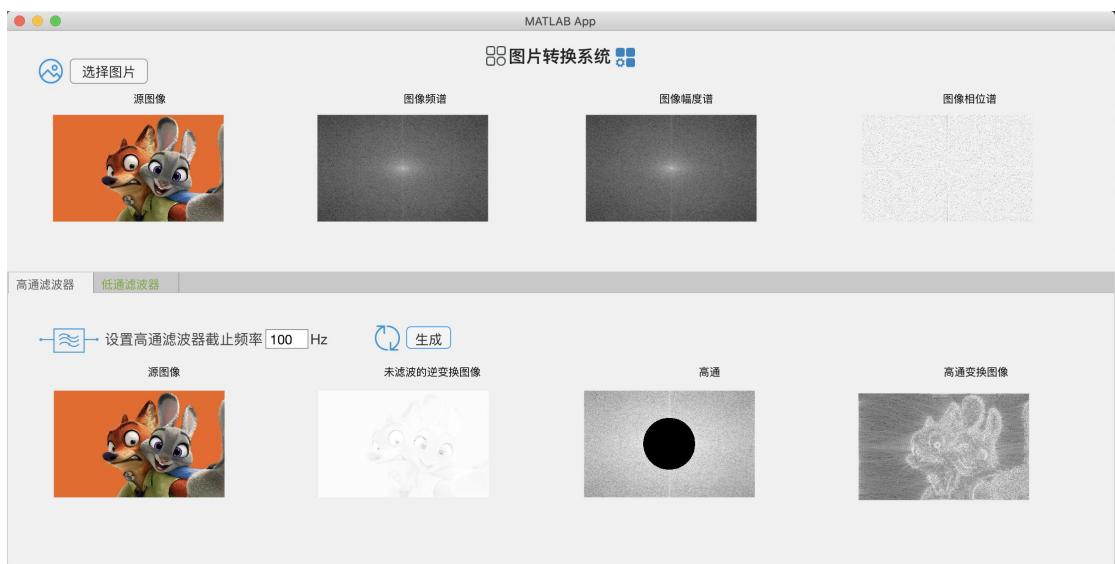


图 5-4 图像转换系统测试效果 2

➤ 003：使用默认低通滤波器转换

选择低通滤波器，使用默认频率 1000Hz，点击生成

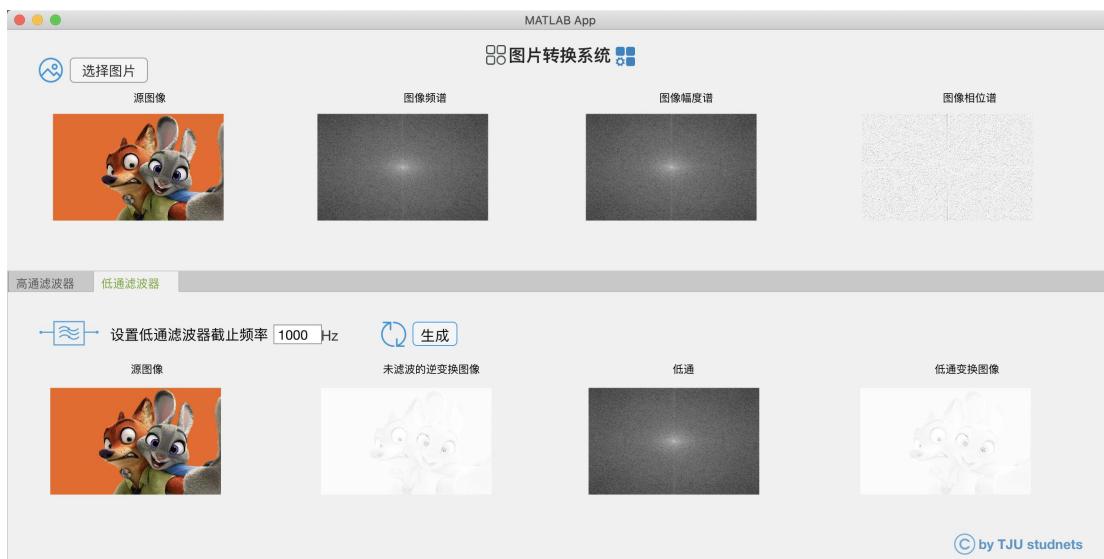


图 5-5 图像转换系统测试效果 3

➤ 004：修改高通滤波器频率后转换

选择高通滤波器，修改截止频率为 40Hz，点击生成

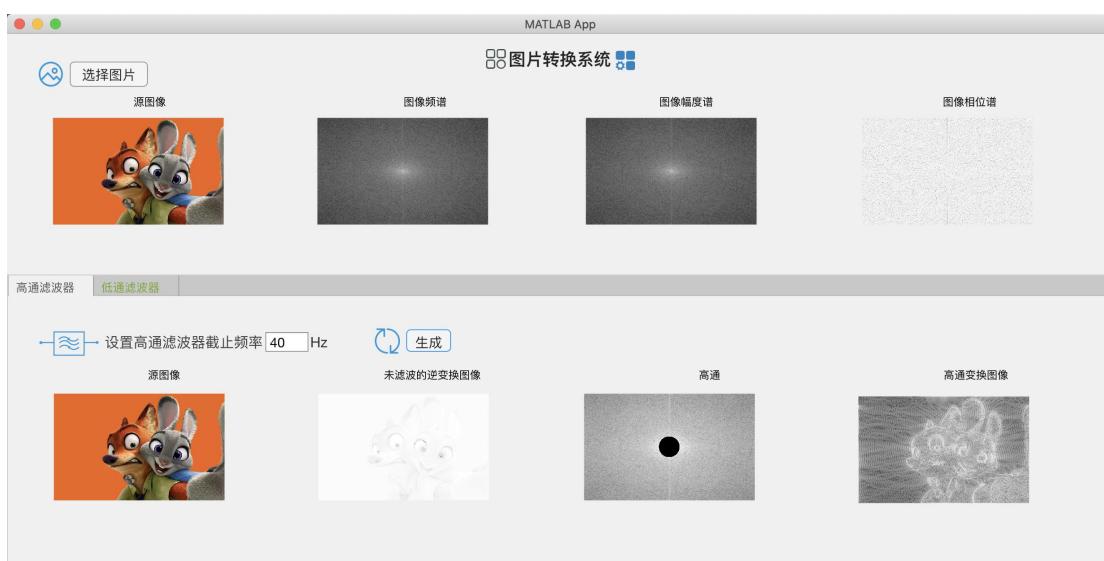


图 5-6 图像转换系统测试效果 4

➤ 005：修改低通滤波器频率后转换

选择低通滤波器，修改截止频率为 1500Hz，点击生成



图 5-7 图像转换系统测试效果 5

➤ 006：修改输入图片

点击左上角选择图片，选择 image4.jpg，点击确定



图 5-8 图像转换系统测试效果 6

附录

附录 1 人员分工

陈姝宇:

作业 1 的模块设计。完成数字信号转化为模拟信号模块代码、信号过滤分离、时域信号与频域信号转换等模块的代码编写。作业 1 可视化界面代码编写。完成实验报告中作业 1 的系统设计以及实验报告文档排版和整合。

刘兴宇:

大作业分工安排。滤波器的设计、音频信号的播放代码编写。进行样本输入收集来完成作业 1 和 2 的输入输出测试和调试。作业 2 可视化界面代码编写。完成实验报告中作业 1 和作业 2 的系统需求、功能需求、系统使用手册和系统测试。

乔雨菲:

作业 2 的模块设计。作业 2 中的频谱、低通高通滤波器设计和作用分析、相位谱和幅度谱代码实现。可视化界面设计与优化。完成实验报告中作业 2 的系统设计。滤波器设计和作用分析