

# 第二次作业报告

PB21010362 汪兆辰

2023 年 3 月 18 日

## 1 算法介绍

程序采用 RBF 算法，通过设定一系列参考点将原图 im 中的像素点插值计算映射到像 im2 中的对应位置。变换函数为

$$f(x) = \sum_{i=1}^N a_i b_i(x) \quad (1)$$

其中  $b_i(x)$  是径向函数，可定义为：

$$b_i(x) = \frac{1}{|x - p_i|^2 + d} \quad (2)$$

式中 N 为参考点  $p_i$  的个数，x 为原图 im 中点的位置， $f(x)$  为像点的位置，d 为参数。

将  $p_i, q_i$  代入可得到方程：

$$Q = PA \quad (3)$$

其中 Q 为参考点为行向量组成的 N\*2 矩阵，P 为 N\*N 矩阵且满足  $P(i, j) = \frac{1}{|p_i - q_j|^2 + d}$ ，A 为系数  $a_i$  为行向量组成的 N\*2 矩阵。由该方程可解出系数矩阵 A。

将 im 中的点遍历并带入  $f(x)$ ，计算结果经四舍五入为 im2 中点的坐标后，就得到了 im 经过 RBF 变换后的图像。

## 2 算法实现

### 2.1 使用幂函数的算法实现

将  $b_i(x)$  定义如上，设置参数  $d=1e6$ ，代码如下：

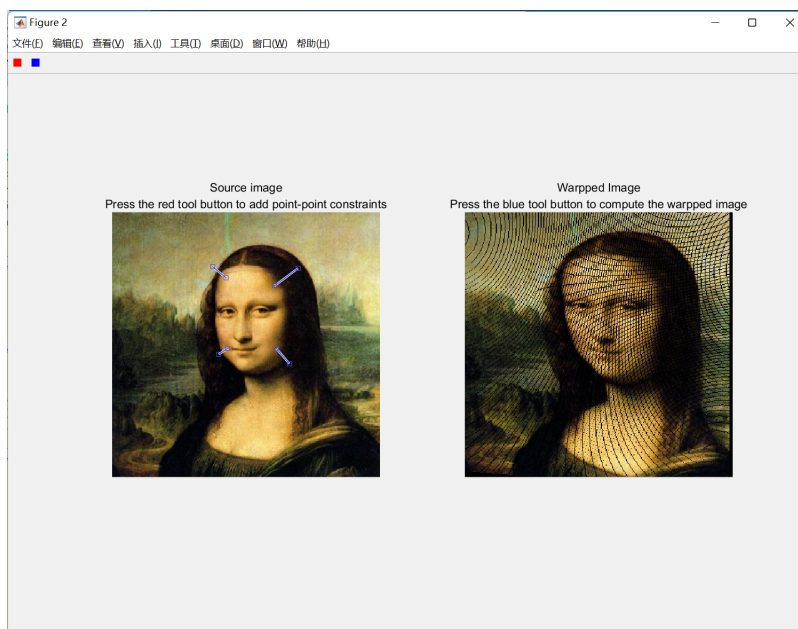
```
N=size(psrc,1);
P=zeros(N);
d=1e5;
for i=1:1:N
    for j=1:1:N
```

```

        p=psrc(i,:);
        q=pdst(j,:);
        P(i,j)=1/(norm(p-q)^2+d);
    end
end
A=P\(pdst-psrc);
for i=1:h
    for j=1:w
        for k=1:dim
            x=[j,i];
            fx=zeros(1,2);
            for m=1:N
                fx=fx+A(m,:)/(norm(x-psrc(m,:))^2+d);
            end
            fx=fx+x;
            if fx(1,1)>0&&ceil(fx(1,1))<=h&&fx(1,2)>0&&ceil(fx(1,2))<=w
                im2(ceil(fx(1,2)),ceil(fx(1,1)),k)=im(i,j,k);
            else
                end
        end
    end
end
end
end
end

```

图像处理的结果如下：



注意到此变换不保持局部性，即整张图都因变换产生较大变化，与预期只有参考点附近变化较大不符，因而考虑采用高斯函数来替代幂函数。

## 2.2 使用高斯函数的算法实现

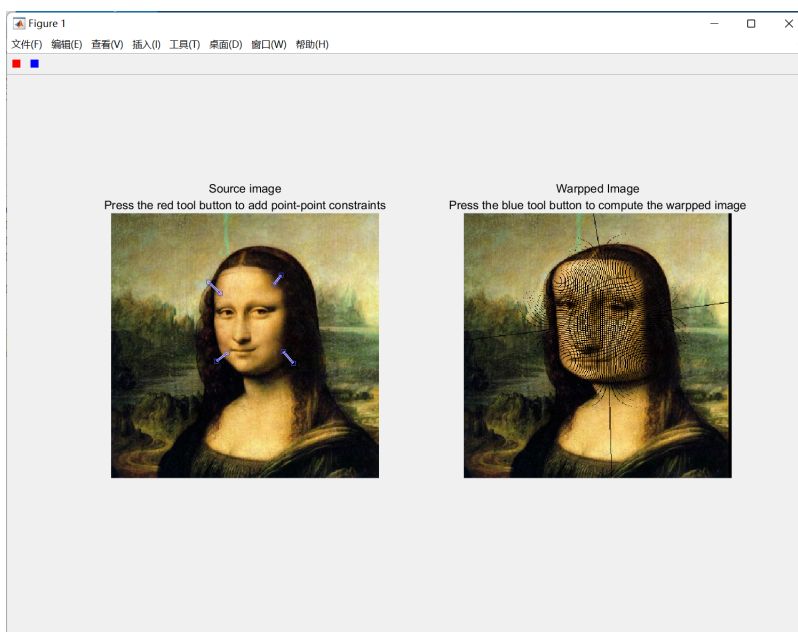
高斯函数的形式为：

$$g(t) = e^{-t^2/\delta^2} \quad (4)$$

可以证明<sup>[1]</sup>，利用高斯函数可以使得离参考点较远的点经变换后还在原位置附近，而离参考点较远的点经变换会发生较大变化。

代码只需将上述代码的  $b_i(x)$  修改成高斯函数  $g_i(x)$ ，不再展示修改后的代码。

取  $\delta^2 = 2200$ ，图像处理的结果如下：



## 3 算法的进一步完善

容易发现按照上述算法进行图像处理后的图像中有大量黑色条纹，这是由于一方面源图像的存储结构为对像素点进行采样后得到的矩阵，本身已经是离散的，经过 RBF 变换后更无法保证像的每个像素点连续；另一方面由于计算得到的坐标并非整数，在赋值过程中经过了取整，导致条纹处的点没有原像。因此需要在已有算法基础上进一步完善。

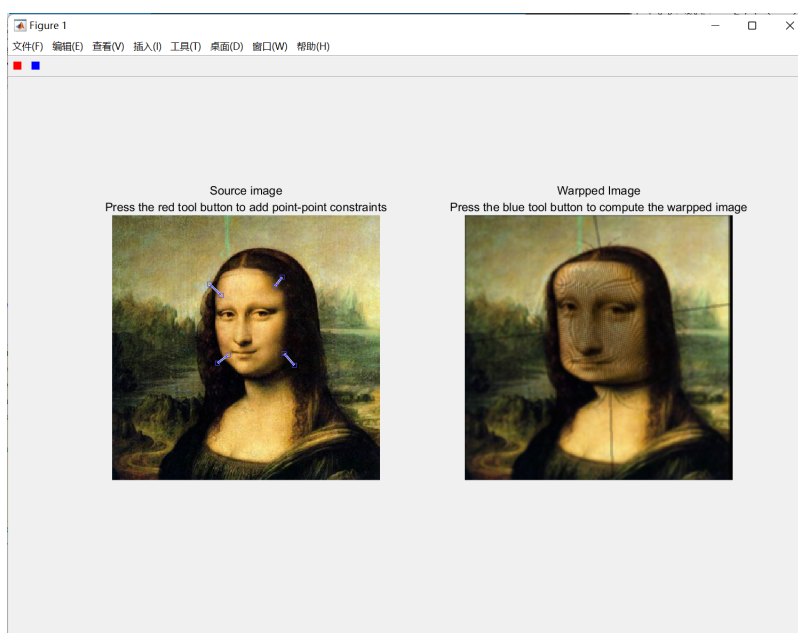
### 3.1 尝试使用卷积

卷积可以使图像模糊或“平滑”，于是尝试用卷积解决上述问题，代码如下：

```
% Create Gauss filter
kernel_size = 5;
sigma = 2;
gauss_kernel = fspecial('gaussian', [kernel_size kernel_size], sigma);

% convolution
im2(:,:,1) = conv2(im2(:,:,1), gauss_kernel, 'same');
im2(:,:,2) = conv2(im2(:,:,2), gauss_kernel, 'same');
im2(:,:,3) = conv2(im2(:,:,3), gauss_kernel, 'same');
```

图像处理的结果如下：



发现卷积并未完全消除黑色条纹，反而由于卷积的平均作用，整张脸的颜色都明显变暗，处理效果并不理想. 而且这也只是对图像的二次处理，并不解决本质问题，需要寻找其他的解决方案.

### 3.2 尝试使用逆变换

RBF 算法的变换是将所有  $im$  中的点经过变换找到  $im2$  上的点与之对应，则黑色条纹中的点也应该可以对应回  $im$  中的某个点，而且应该是多个  $im2$  中的点都对应到  $im$  上一个点. 按照这种思想，我们只需要解非线性方程：

$$y = \sum_{i=1}^N a_i b_i(x) \quad (5)$$

其中  $y$  取遍  $im2$  中点的坐标. 代码如下：

```
for t=1:N
    sum=sum+A(t,:)/(norm(x-psrc(t,:))^2+d);
```

```

end
F=@(x) sum+x-[i,j];
solution=fzero(F,[1,1]);

```

fzero 用数值逼近方程的解，但该方法的计算量过大，还应寻找更好的算法.

### 3.3 一种形式上的“反变换”

原算法出现黑色条纹的一部分原因是因为变换并非满射，为了消除结果的条纹，我们考虑从 im2 出发的一个单射. 但在 3.2 中我们已经说明了直接求得变换的反变换并不容易，甚至无法求出，于是我们转而考虑一种形式上的“反变换”.

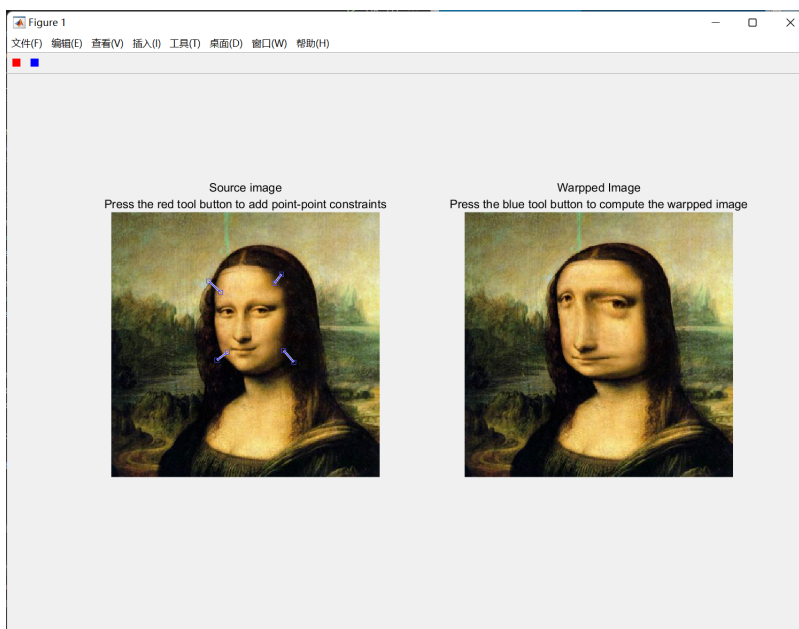
假设原变换的效果是使图像向外“膨胀”，那么“反变换”的效果应该使图像向内“收缩”，反之亦反. 那么我们只需要当 RBF 作用在 im2 时可以得到源图像 im，并在处理时对 p,q 进行调换，就可以得到具有“反变换”效果的一个变换，即若有原变换为：

$$y = x + g(x) \quad (6)$$

那么只需：

$$x = y + g'(y) \quad (7)$$

其中 x,y 分别为 im,im2 中的点. 图形处理的结果如下：



注意这种变换并不是严格意义上的反变换，得到的图形与用之前算法得到的图形之间有差别，用这种变换直接作用在变换后的图像上也无法还原成源图像.

## 参考文献

- [1] Nur Arad and Daniel Reissfeld, Image Warping Using Few Anchor Points and Radial Functions. Computer Graphics Forum, 1995.
- [2] <https://zhuanlan.zhihu.com/p/366929278>
- [3] [https://blog.csdn.net/weixin\\_39702639/article/details/111203275](https://blog.csdn.net/weixin_39702639/article/details/111203275)