

# 计算机图形学

## 上机作业二

刘国瑞 PB21000145 id: 64

2023 年 3 月 19 日

## 问题描述

1. 使用 RBF 算法实现图像拉伸变形 (image warping)
2. 结果图像中有时会出现白色空洞或条纹, 分析其原因并优化算法, 解决该问题

## 实验原理

注: bonus 作业: 解释原因并优化算法参考下文的分析

### 基础算法

基于老师发布的作业说明中的算法, 进行一些修改 (算法本身是有点问题的):

1.  $d$  需要取的大一点, 我的算法中取  $d = 10000$
2. RBF 算法中要用的函数存为 `f.m` 和 `RBF.m`
3. 取  $g(x) = f(x) + x$ , 点  $x$  实际上是映射到  $g(x)$ , 通过  $f(p_i) = q_i - p_i$  来得到系数  $a_i$
4. 映射得到的点可能超出图像的坐标范围, 且可能不是整点, 所以我使用 `if` 语言进行判断像点是否在图像内部, 并用 `round` 函数对像点坐标取整数
5. 若在图像左下和右上区域画线, 会发现拉伸的结果和理想结果对称, 即反向拉伸的现象。我分析应该这是由于图像坐标的定义和别的一些原因所致。为解决这个问题, 我将 `pdst` 和 `psrc` 点的 `xy` 坐标交换了一下

算法能基本实现图形的拉伸, 但是会出现一些不和谐的条纹, 且  $p_i, q_i$  越多, 拉伸后得到的图像就越与理想结果不符, 甚至出现很大的黑色斑点

## 改进算法

基础算法中出现的条纹是由于数字图像的数学模型是离散的，在映射过程中有一些像空间中的点并没有被映射到，所以那些点的值还是原来的初始值，故放在拉伸后的图中就不具有连续性，显得不和谐。要解决这个问题，就需要让像空间中涉及拉伸的点之间具有连续性，或者说可以找到原像。

改进算法是基于**逆向映射**，将像图当作原空间，原图当作像空间。具体来说，我们知道正的映射是从  $p_i \rightarrow q_i$ ，那像图到原图的映射就是  $p_i \rightarrow p_i$ ，所以可以通过同样的算法，将像图中的点映射回原图中。这个像点不一定是整点，这里我们使用**双线性插值**，对该点存储的颜色数值进行近似 (注意这里近似后的颜色也不一定是整数，和 im 矩阵元素类型不符，故还需要取个整)。

## 其余算法的尝试

根据条纹状的问题分析，最直观的解决方式就是将对应坐标用  $f^{-1}$  映射回原图，然后找到原点给其赋值，这样就能确保连续性。但是这个算法要求数值解，我调查到的 matlab 函数 `fzero` 和 `fsolve` 在求解时的速度都不快，且会在命令窗口打印一些无用的东西，使得整个拉伸过程很慢。并且，在拉伸幅度较大的时候，或者  $p_i, q_i$  较多的时候，像图中仍会出现一些斑点，无法彻底实现拉伸。所以 `bonus` 还是用逆向映射的方法好一点。

## 实验结果

### 基础算法一组约束点

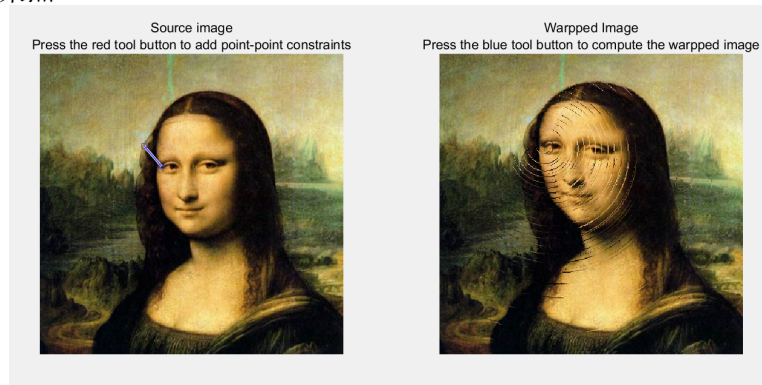


图 1.

### 基础算法多组约束点

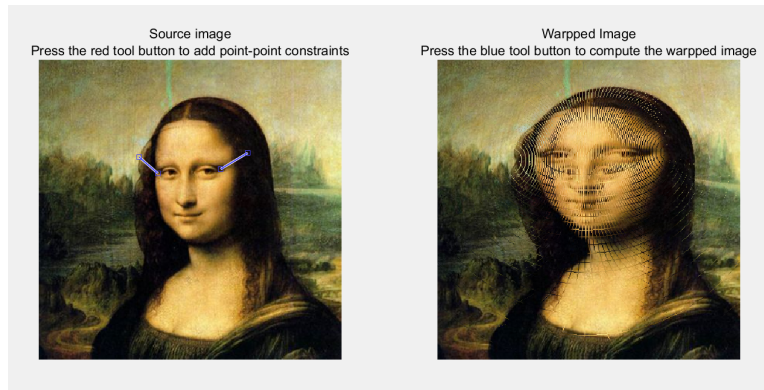


图 2.

改进算法一组约束点

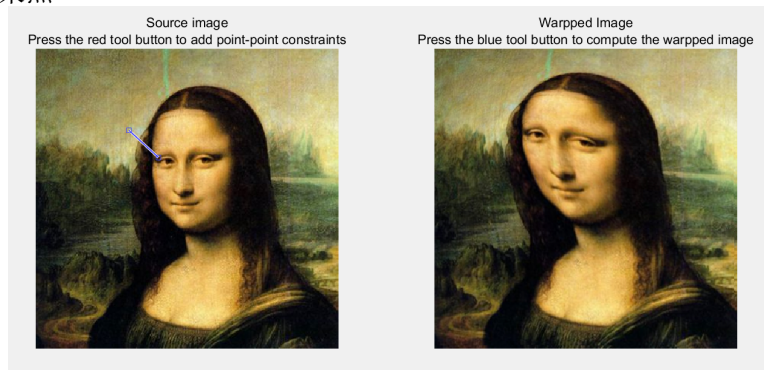


图 3.

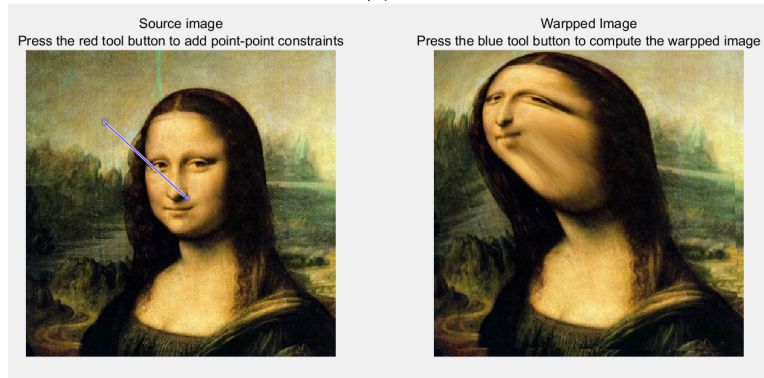


图 4.

改进算法多组约束点

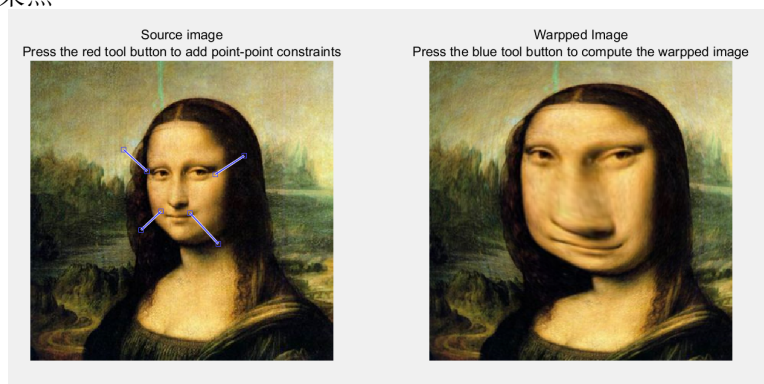


图 5.

## 结果分析

由实验结果可以看到，基础算法在一组约束点时能看得出来有拉伸效果（忽略条纹状），但是多组时就不能看了。

改进算法即使是拉伸幅度很大的时候也可以正常拉伸，且多组约束点的时候表现也良好，像图中没有不和谐的地方。

## 代码说明

改进算法我新开了一个 RBFImageWarp\_optimized.m 文件存储，使用时在 toolWarpCB.m 文件 14 行中用想要的函数即可。

## 致谢

感谢助教的辛苦付出！