

图像的几何变换

李东嵘

16342080

所用平台: windows10, MATLAB R2016a

1.任务描述

在本报告中,我们将使用图像的几何变换技术,将下述图像 colorful rose 几何变换到抛物面 $z = x^2 + y^2, x \geq 0, 10 \leq z \leq 100$ 上。



2.问题分析

在具体操作前,我们先考虑想要映射到的曲面的值域。稍加分析可以知道,上述曲面在 uOv 平面上对应的区域为一个扇形 $G = \{(u,v) \mid 10 \leq u^2 + v^2 \leq 100, v \geq 0\}$ 。将一副二维的矩形图像直接蒙到三维曲面上是困难的,但是我们可以考虑先将这一曲面通过映射作用到曲面的定义域上,然后通过垂直抬升的方法将之蒙到相应的曲面上。

换句话说,假设原图像在 xOy 平面上是一个左下角位于原点的矩形 F (为了方便数学推导,我们假设图像的左下角为原点),那么我们第一步的目标是希望找到一个映射,能将 F 映照为 G ,即将一个矩形映射为一个扇形。用数学语言叙述即为 $f : xOy \rightarrow uOv, s.t. f(F) = G$ 。

现在，我们约定记矩形长为a，宽b，围成扇形的同心圆半径分别为 R 与 \sqrt{R} （这是符合上述 G 的定义的）。设 $f(x, y) = (u, v)$ 。为了保持变换前后图形内部结构关系的一致性，一个自然的想法是我们将 F 按其宽来“分层”，将每一层映照到扇形对应半径的同心圆上，即令 x 正比于 (u, v) 与 v 轴夹角 θ ，令 y 正比于 (u, v) 到原点距离 r 。即：

$$\theta/\pi = x/a,$$

$$r/(R - \sqrt{R}) = y/b$$

直观上来说，这一映射将平行于 x 轴的直线 $y = y_0$ 映照成了半圆 $u^2 + v^2 = r^2, (u, v) \in G$ 。由于 x, y, R, a, b 已知，因此我们可以求出 r, θ ，进而由圆的参数方程我们得到 $u = r\cos(\theta) = \frac{y}{b}(R - \sqrt{R})\cos(\pi \frac{x}{a}), v = r\sin(\theta) = \frac{y}{b}(R - \sqrt{R})\sin(\pi \frac{x}{a})$ 。由此我们就成功将图像由矩形扭曲为扇形。接下来，我们只需要将该图像垂直抬升即可完成全部任务。

算法描述如下：

输入：图像 F

输出：几何变换后抬升至三维空间的图像 H

1. 用上述变换，将 F 变换为 G
2. $w = u^2 + v^2, (u, v) \in G$,
3. 绘制 H 并对每一对 $(u_1, v_1), w_1$ ，令 w_1 颜色等于 (u_1, v_1) 的颜色。

3.图像的几何变换

接下来我们将用作业中的图片作为例子实践该算法的效果。首先，我们导入图片：

```
image=imread('c:/users/a/desktop/flowers.jpg');
```

然后，通过上述算法，我们将 F 变换到 G ，得到扭曲后的图像colormap，代码见附录1。

在变换前，我们的图像如下：



变换后得到的图像colormap如下所示：



我们看到，图像被扭曲到了一个扇形区域中，这对应的即为扇形区域G。但是美中不足的是，图像上有大量的黑色点，这是由数值计算的精度问题造成的：我们采用的变换是非线性变换，而变换前后的坐标只能是整数，因此在变换中，colormap图片中的部分点由于舍入误差原因没有原像，这就导致了该点为黑。为了解决填补扇形区域内的黑色点，考虑到大部分黑色点均单个出现，并没有成片黑色点的情况，我们使用中值滤波器对图片进行中值滤波，给黑色点填色。这一步骤需要额外实现中值滤波器sortfilter()，滤波过程与滤波器实现均在附录中给出。

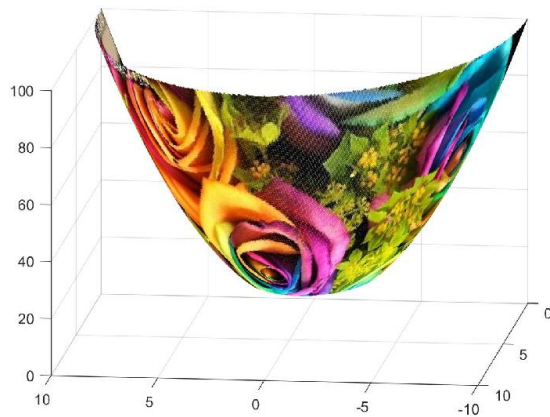
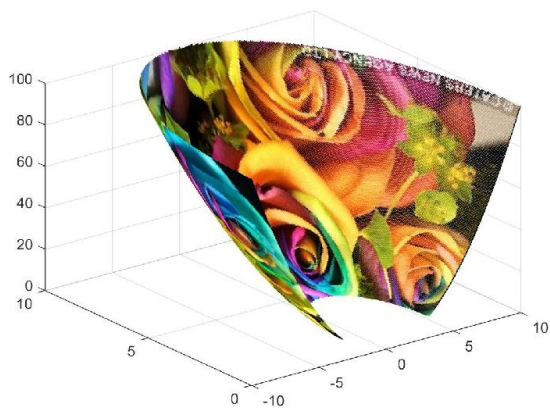
中值滤波后的colormap 如下图所



可以看到，绝大多数的黑点都被成功填补了，被扭曲后的图像呈现出了较为完整的结构。现在我们使用 `mesh` 函数绘制被图像覆盖的三维曲面：

```
mesh(u,v,w,colormap)
```

得到的三维曲面图如下所示：



效果十分之优秀，原来的矩形图像区域被成功地形变到了一块三维曲面上。

4.APPENDIX

4.1 图像扭曲

```
image=im2double(image);
[u,v]=meshgrid(-10:0.04:10,0:0.04:10);
w=u.^2+v.^2;

%构造出题目所求曲面
for i=1:length(u(1,:))
    for j=1:length(v(:,1))
        if u(1,i).^2+v(j,1).^2<10 | u(1,i).^2+v(j,1).^2>100
            w(j,i)=nan;
        end
    end
end

%半径
R=length(u(1,:))/2;

[m,n,k]=size(rose);

%初始化colormap
colormap=zeros(length(v(:,1)),length(u(1,:)),3);

%几何变换, 扭曲图像
for i=1:n
    for j=1:m
        x=i;
        y=j;

        r=sqrt(R)+(y/(m))*(R-sqrt(R));
        theta=pi*x/(n);
        ux=r*cos(theta);
        vy=r*sin(theta);

        ux=R+ux;
        vy=vy;

        ux=round(ux)+1;
        vy=round(vy)+1;

        ux
        vy

        colormap(vy,ux,1)=rose(j,i,1);
        colormap(vy,ux,2)=rose(j,i,2);
        colormap(vy,ux,3)=rose(j,i,3);
    end
end

%归一化
for i=1:3
    colormap(:,:,i)=colormap(:,:,i)/max(max(colormap(:,:,i)));
end

%由于循环避免出现0坐标的原因导致colormap出现了一个像素的溢出, 因此重新调整colormap大小, 删去溢出的一个像素
colormap=colormap(1:251,1:501,:);
```

4.2 图像的中值滤波

4.2.1 中值滤波器

```
%function output=sortfilter(x)
x=x(:);
output=median(x);
%end
```

4.2.2 对colormap中值滤波

```
%edge
[h,g,at]=size(colormap);

background=zeros(h+2,g+2,3);

background(2:h+1,2:g+1,:)=colormap;

for i=2:h+1
    for j=2:g+1
        %检测黑点是否在G内以及是否为需要除去黑点
        r=((i-1)^2+(j-1-R)^2);
        if r>=R & r<=R^2
            tmp=background(i,j,:);
            if sum(tmp)<0.01
                %对各个频道中值滤波
                background(i,j,1)=sortfilter(background(i-1:i+1,j-1:j+1,1));
                background(i,j,2)=sortfilter(background(i-1:i+1,j-1:j+1,2));
                background(i,j,3)=sortfilter(background(i-1:i+1,j-1:j+1,3));
            end
        end
    end
end

%输出滤波后的colormap
colormap=background(2:h+1,2:g+1,:);
```