课程实践 5：图象几何变换

实验目的：

纹理映射，将二维图像投影到三维物体表面上

实验要求：

将此图片 Colorful Rose 投影在抛物面 $z = x^2 + y^2$ $(x \geq 0, 10 \leq z \leq 100)$ 上

实验环境： MATLAB R2018a

原理和方法：

图像变形的实质是，在二维平面中，将一个有界区域得图像按照一组映射规则，将源图像像素值映射到目的图像中，从而产生另一幅数字图像的过程。根据图像像素值的离散表示，图像可以表示为一个二维矩阵，矩阵中元素表示该位置处的像素值。因此图像变形的实质是，找到一个合适的变换 f(I), 将源图像 Src 映射到 Dest = f(Src)即可得到变形后得目标图像。

映射分为前向映射和后向映射。前向映射是从源域向目的域进行，由于源域的信息是已知的，该过程根据源像素信息，直接投影获得目的域中对应的目的像素位置，计算 d=f(s)即可。与之相反，后向映射是从目的域向源域进行，后向映射的思路是针对每个未知的目的像素，通过在源域中对源像素的搜寻和匹配来完成。

前向映射(Forward Mapping)由原始图像 I_s 通过变形函数 f 直接获得目标图像 I_d。包括采样和重建两步。首先对原始图像采样，即扫描原始图像 I_s 的每一个像素；然后重建，即通过变形函数，依次计算每个像素对应到目标图像 I_d 中的位置，将每个原始图像像素的 R,G,B 值依次复制到目标图像中的指定位置。如果一个输入像素被映射到若干个输出像素之间的位置上，那么目标图像的像素值就按插值算法在若干个输出像素之间进行分配。

后向映射(Inverse Mapping)是从目标图像 I_d 出发，通过变形函数 f 求解目标图像上每一个像素点位置在原始图像 I_s 上的对应位置。后向映射同样包括采样和重建两个步骤，跟前一种方法正好相反，在这种方法中输出像素一次一个地映射回到输入图像中，以便确定其源像素值，假如一个输出像素被映射到若干个输入像素之间，那么它的像素值由这若干个输入像素的像素值插值决定。后向映射法是前向映射法的逆。

前向映射的显著优点是计算简单，只需要根据变换 f 在定义域上的信息计算出值域信息；然后简单地将像素 R,G, 值赋给目标点或者通过插值算法得到目标点像素值即可。前向

映射的缺点包括冗余映射(像素重叠)，映射空洞，映射出界等问题。在后向映射中这些问题基本上得到了有效的解决。

1) 后向映射的效果要好于前向映射，因为目标图像的每个像素都能被扫描，获得适当的 R,G,B 值，避免了前向映射中输出图像的某些点可能没有被绘制而出现空洞的情况。

2) 由于许多输入像素可能映射到输出图像的边界之外，所以前向映射法浪费了部分像素信息，即映射出界。后向映射解决了这个问题，因为后向映射不用考虑目标图像区域外的点。

3) 一个输出像素可能由多个输入像素来决定，因而涉及到多次计算，即冗余映射，出现重叠现象。而后向映射也较好地解决这个问题，根据后向变换的定义，一个目标像素点仅计算一次。

①获取坐标

• 法一：直接横纵坐标等距采样，然后 meshgrid 获取网格，再带入抛物面公式获取，但由于题目中的坐标范围是圆环，所以要将 z<10 和 z>100 的区域赋值为 NaN。

• 法二：极坐标转三维笛卡尔坐标，这里可以直接对 θ 和 r 的范围作出限制获得圆环域，公式如下：

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \\ z = r^2 \end{cases} \qquad -\frac{\pi}{2} \le \theta \le \frac{\pi}{2}, \sqrt{10} \le r \le 10$$

②纹理映射

纹理映射（Texture Mapping），又称纹理贴图，是将纹理空间中的纹理像素映射到屏幕空间中的像素的过程。简单来说，就是把一幅图像贴到三维物体的表面上来增强真实感，可以和光照计算、图像混合等技术结合起来形成许多非常漂亮的效果。

• 法一：直接使用 MATLAB 自带函数 warp，采用双线性插值方法将图像的像素值赋给三维表面，这里对三维形状的采样点个数没有要求。

•法二：先用 surf 函数画图，再用 set 语句更改图像的 CData 属性，对于矩形范围 0≤x≤10，-10≤y≤10，即直角坐标中，横坐标采样点数=图像的列数，纵坐标采样点数=图像的行数；将图像映射到该矩形区域，再剔除不属于定义域 $x^2+y^2<10$ & $x^2+y^2>100$ 的点，然后投影到三维空间上。

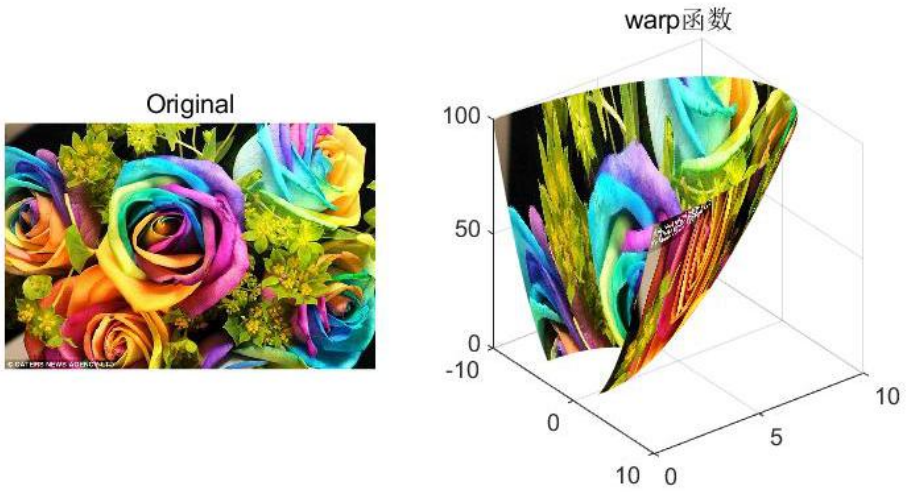
• 法三：先用 surf 函数画图，再用 set 语句更改图像的 CData 属性，这里要注意由于是将图片的像素点一一对应到抛物面采样点上，因此抛物面坐标的采样点的个数要与图像尺寸一致，即直角坐标中，横坐标采样点数=图像的列数，纵坐标采样点数=图像的行数；极坐标中，角度采样点数=图像的列数，半径采样点数=图像的行数。

实验代码和结果：

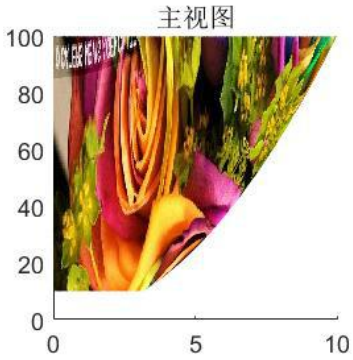
法一 warp函数

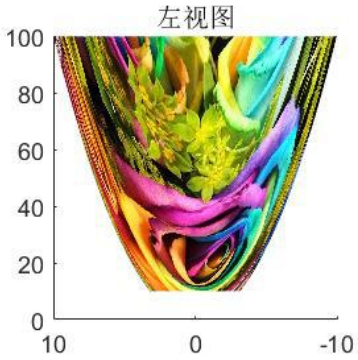
```
clear all
A = imread('Colorful Rose.jpg');
xx = 0:0.01:10;
yy = -10:0.01:10;
[x y] = meshgrid(xx,yy);
z = x.^2 + y.^2;
z(z<10) = NaN;
z(z>100) = NaN;

figure()
subplot(1,2,1),imshow(A)
title('Original')
subplot(1,2,2),warp(x,y,z,A);
view(3)
axis square
grid on
title('warp函数')

figure()
subplot(2,2,1),warp(x,y,z,A)
title('主视图')
view(0,0)
axis square
subplot(2,2,2),warp(x,y,z,A)
title('左视图')
view(90,0)
axis square
subplot(2,2,3),warp(x,y,z,A)
title('俯视图')
view(2)
axis square
```

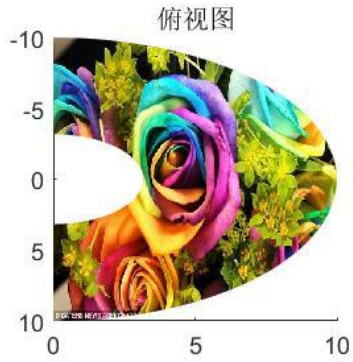
贴图效果如下：



三视图如下：





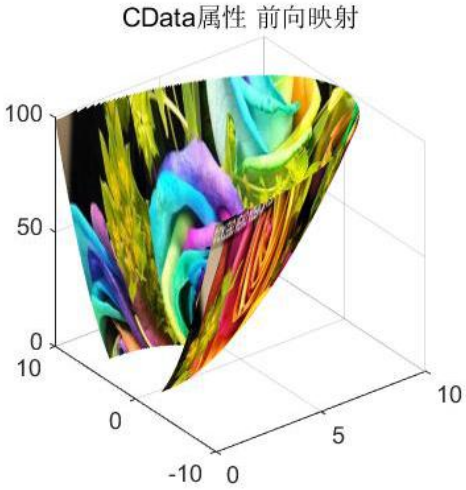


法二 CData属性 前向映射

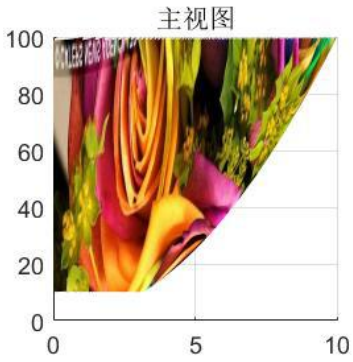
```
clear all
A = imread('Colorful Rose.jpg');
[m n ~] = size(A);
xx = linspace(0,10,n);
yy = linspace(10,-10,m);
[x1 y1] = meshgrid(xx,yy);
z1 = x1.^2 + y1.^2;
B = z1<=100 & z1>=10;
z1(B == 0) = NaN;

figure()
subplot(1,2,1),imshow(A)
title('Original')
subplot(1,2,2)
h=surf(x1,y1,z1);
shading interp
set(h,'CData',A); % 纹理贴图
view(3)
axis square
title('CData属性 前向映射')
```

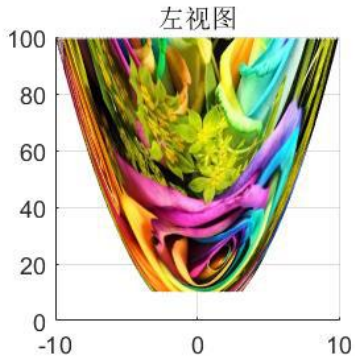
贴图效果如下：

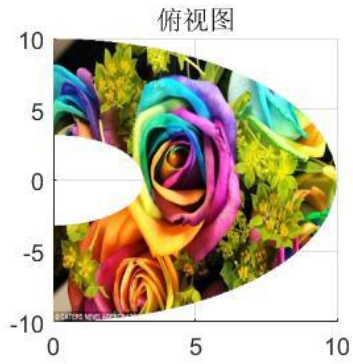

Original


CData属性 前向映射
100
50
0
10
0
-10
0
5
10

三视图如下：


主视图
100
80
60
40
20
0
0
5
10

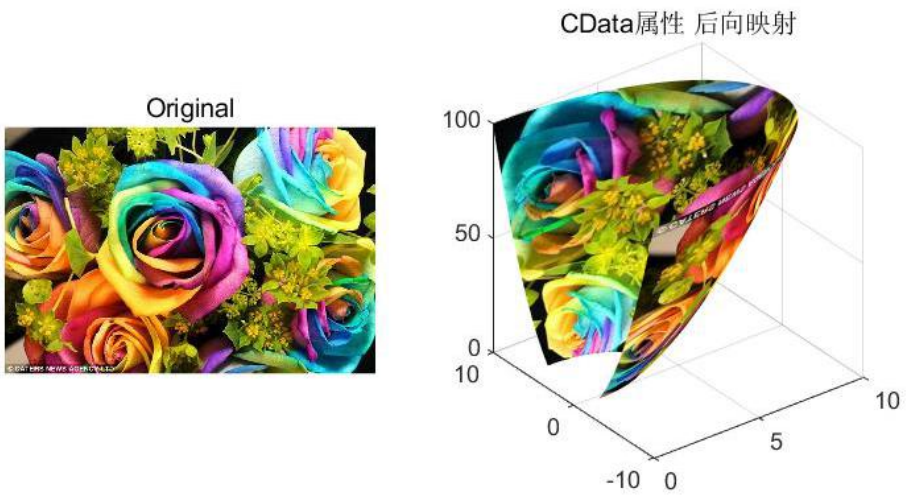

左视图
100
80
60
40
20
0
-10
0
10


俯视图
10
5
0
-5
-10
0
5
10

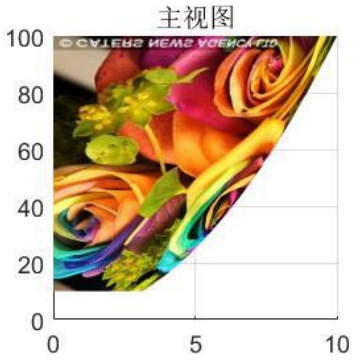
法三 CData属性 后向映射

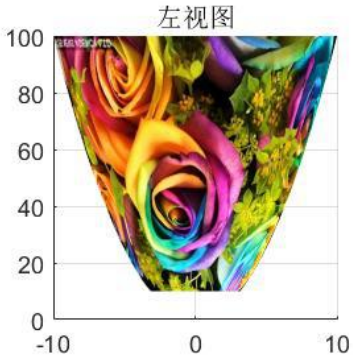
```
clear all
A = imread('Colorful Rose.jpg');
[m n ~] = size(A);
theta = linspace(-pi/2,pi/2,n) ;
r = linspace(sqrt(10),10,m);
[theta r] = meshgrid(theta,r);
x = r.*cos(theta) ;
y = r.*sin(theta) ;
z = x.^2 + y.^2;

figure()
subplot(1,2,1),imshow(A)
title('Original')
subplot(1,2,2)
h=surf(x,y,z);
shading interp
set(h,'CData',A); % 纹理贴图
view(3)
axis square
title('CData属性 后向映射')
```

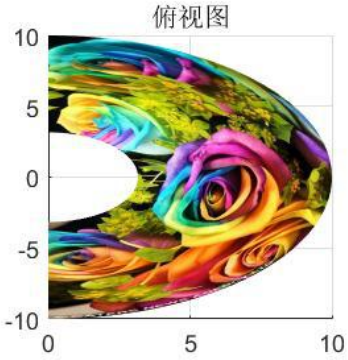
贴图效果如下：



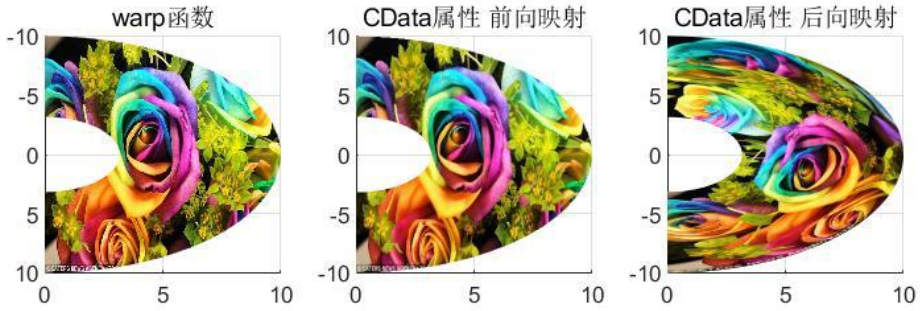
三视图如下：



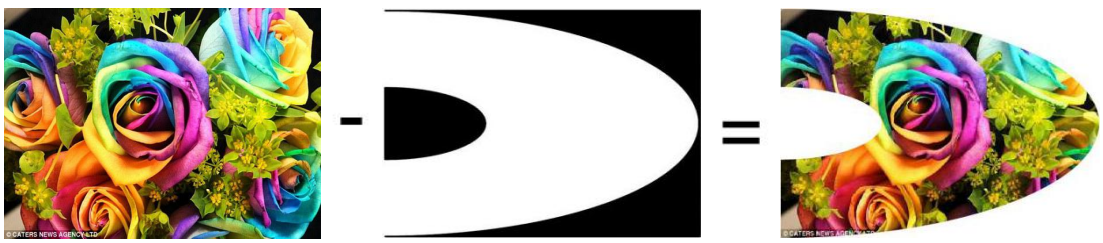


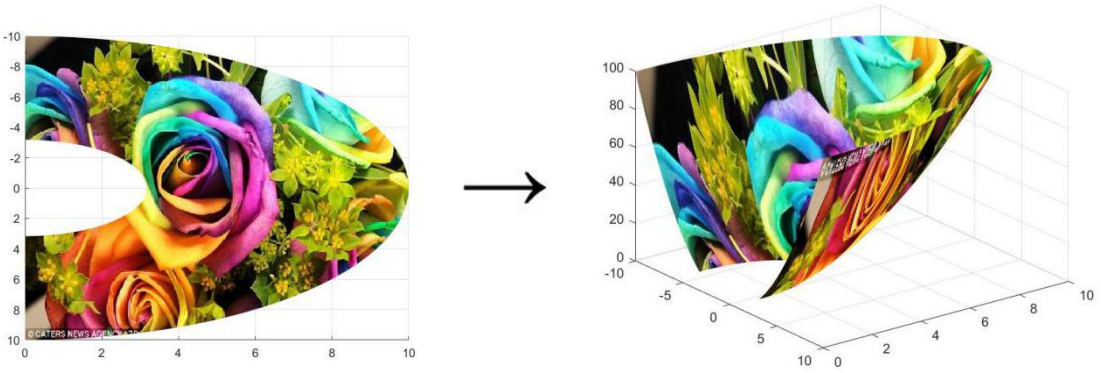


对比三种做法的俯视图（如下图），可以看出：

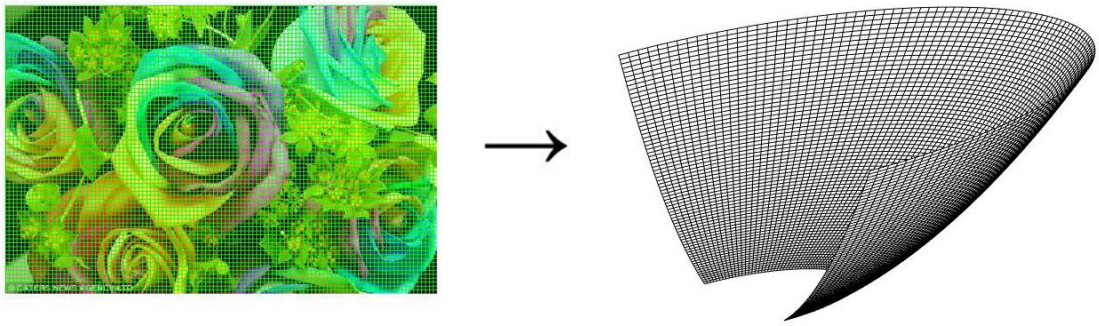


warp 函数应当是插值映射，像是先把图像按曲面的定义域（xOy 平面上）形状裁剪了之后再对应到曲面上的点，本质还是二维→二维线性映射，因此看俯视图中图像的比例没有大变化，就是观察如何二维映射的，这样子的结果就是，有些点不会被映射，因此不是原图像和曲面不是一一对应的，个人觉得本质上是非等分辨率的前向映射。使用 CData 函数进行了验证。





而由极坐标定义的映射则是直接二维→三维点的映射，等角度等层间距的非线性映射，因此是进行了一定的拉伸扭曲，图像各像素点之间的距离比例已经完全不同，但是原图像每个像素点都能在曲面上找到对应的映射点，本质上是等分辨率的后向映射。



参考文献：

[1] 唐亮亮, 基于前向映射的图像变形算法研究,上海交通大学硕士学位论文, 2009-12