

Report of HW6

夏浩 19307130268

1.编程实现

(1) 空间变换算法，
(2) 并将上述空间变换应用于实现基于反向的图像变换过程，从而实现图像甲到图像乙的图像变换。

作业的基本算法内容参考课堂上讲解和课件。

开发环境：Matlab

Code:

```
r = 5;
A=imread('baboon.png');
B=imread('pig.jpg');
[n,m,~]=size(A);
imshow(A);
set(gcf,'outerposition',get(0,'screensize'));
[x0,y0] = ginput; %模板图上选点
x0=x0/m;
y0=y0/n;
imshow(B);
set(gcf,'outerposition',get(0,'screensize'));
[x1,y1]=ginput; %目标图上选点
while size(x1,1)~=size(x0,1)
    if size(x1,1)>size(x0,1)
        fprintf('选点太多，重新选取\n');
    else
        fprintf('选点不够，重新选取\n');
    end
    imshow(B);
    [x1,y1]=ginput;
end
close(gcf); %关闭当前激活窗口
x1=fix(x1);
y1=fix(y1);
[n,m,p]=size(B);
```

```

%      t=size(x1,1);
x2=fix(x0*m);
y2=fix(y0*n); %实际位置
b=[x1-x2,y1-y2]; %平移变换
C=zeros(n,m,p);
for i=1:m
    for j=1:n
        dist=((x2-i).^2+(y2-j).^2).^0.5;
        dist=dist-r;
        k=find(dist<=0);
        if k
            x=[i;j]+b(k,:)';
        else
            dist=dist/norm(dist,'inf');
            dist=dist.^(-1);
            dist=dist.^2;
            dist=dist/sum(dist);
            x=[i;j]+b'*dist;
        end
        if x(1)>=1 && x(1)<=m && x(2)>=1 && x(2)<=n
            C(j,i,:)=get_pixel(B,x);
        end
    end
end
C=uint8(C);
imwrite(C, 'res.png');
function y=get_pixel(A,x)
    [n,m,~]=size(A);
    x1=floor(x(1));x2=x1+(x1<m);
    y1=floor(x(2));y2=y1+(y1<n);
    s=x(1)-x1;
    r=x(2)-y1;
    y=(1-r)*(1-s)*A(y1,x1,:)+r*(1-s)*A(y2,x1,:)...
        +(1-r)*s*A(y1,x2,:)+r*s*A(y2,x2,:);
end

```

Analysis:

考虑到模板图和待变换图尺寸可能不同，所以我们对图片尺寸进行归一化；

为了处理简便，将局部区域设定为圆，并可以对半径进行改变，当半径为1时则退化为点；

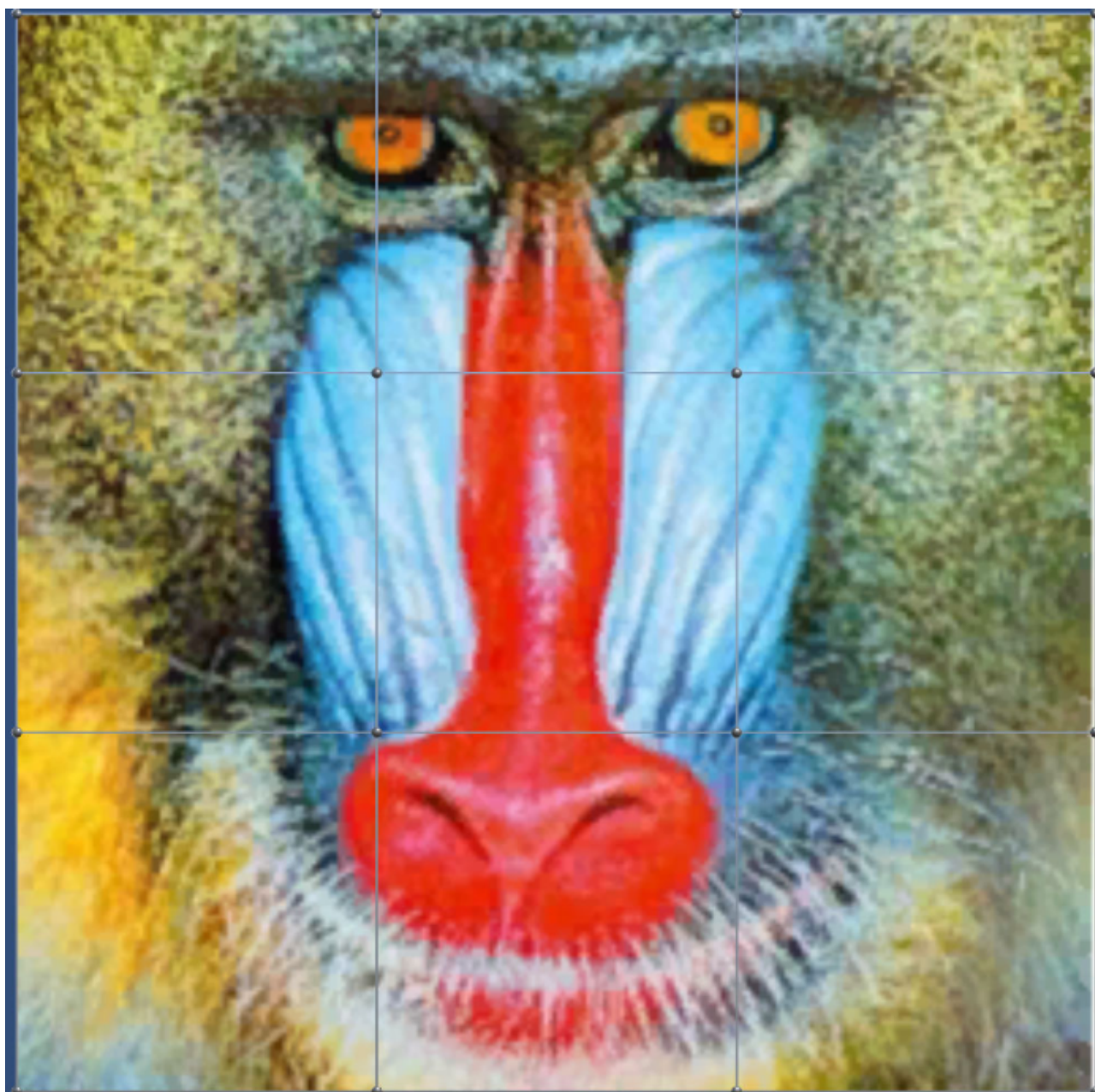
`get_pixel`函数利用双线性插值，得到原始图像上对应位置的像素值，输入参数为原始图像，以及坐标值。

Result:

原图:



目标图:



结果:

