西安交通大学

**计算机视觉与**

**模式识别**

计算机53班

龙思宇

2150500103

1. 实现课上所讲诉的Gradient Blending的内容

源代码

%% 清空工作区和命令行窗口

clc;clear;

%% 读入两幅图像，一幅是background，另一幅是target图像，利用Matlab的roipoly函数标记target一个多边形的区域

background = imread('grass.jpg');

background\_size = size(background);

target = imread('cat\_2.jpg');

target\_size = size(target);

%% 利用roipoly从target图片中选择感兴趣区域

figure(1);

[BW,xi,yi] = roipoly(target);

save('target.mat','BW','xi','yi');

close(figure(1));

target\_Mask = load('target.mat','BW');

target\_Mask = target\_Mask.BW;

target\_Mask = double(target\_Mask);

%% padding操作

if(size(background,3) == 1),background\_size(3) = 1;end

if(size(target,3) == 1),target\_size(3) = 1;end

final\_size = max([background\_size(:) target\_size(:)],[],2);

if(target\_size(1) < final\_size(1))

target\_pad = vertcat(target,zeros(final\_size(1) - target\_size(1),target\_size(2),target\_size(3)));

target\_Mask\_pad = vertcat(target\_Mask,zeros(final\_size(1) - target\_size(1),target\_size(2)));

else

target\_pad = target;

target\_Mask\_pad = target\_Mask;

end

if(target\_size(2) < final\_size(2))

target\_pad = horzcat(target\_pad,zeros(size(target\_pad,1),final\_size(2) - target\_size(2),target\_size(3)));

target\_Mask\_pad = horzcat(target\_Mask\_pad,zeros(size(target\_Mask\_pad,1),final\_size(2) - target\_size(2)));

end

if(target\_size(3) < final\_size(3))

target\_pad = repmat(target\_pad,[1 1 3]);

end

if(background\_size(1) < final\_size(1))

background\_pad = vercat(background,zeros(final\_size(1) - background\_size(1),background\_size(2),background\_size(3)));

else

background\_pad = background;

end

if(background\_size(2) < final\_size(2))

background\_pad = horzcat(background\_pad,zeros(size(background\_pad,1),final\_size(2) - background\_size(2),background\_size(3)));

end

if(background\_size(3) < final\_size(3))

background\_pad = repmat(background\_pad,[1 1 3]);

end

%% 获得位置信息

figure(1);

imshow(background\_pad);

[xshift,yshift] = ginput(1);

close(figure(1));

%% 获取原始图像中

maskPoints = load('target.mat','xi','yi');

xshift = (xshift - mean(maskPoints.xi));

yshift = (yshift - mean(maskPoints.yi));

%% 平移target以及它的mask

target\_pad = imtranslate(target\_pad,[xshift,yshift]);

target\_Mask\_pad = imtranslate(target\_Mask\_pad,[xshift,yshift]);

% figure(1);imshow(uint8(target\_pad));

% figure(2);imshow(uint8(target\_Mask\_pad)\*255);

%% 对target\_pad做拉普拉斯算子卷积

laplacian\_cal = [0 -1 0;-1 4 -1;0 -1 0];

target\_pad\_gradient = zeros(final\_size(1),final\_size(2),final\_size(3));

for path\_con = 1 : final\_size(3)

target\_pad\_gradient(:,:,path\_con) = target\_Mask\_pad .\* conv2(target\_pad(:,:,path\_con),laplacian\_cal,'same');

end

% figure(1);imshow(uint8(target\_pad\_gradient));

%% 给像素编号

target\_label = zeros(final\_size(1),final\_size(2));

label = 1;

% for row\_con = 1 : target\_size(1)

% for col\_con = 1 : target\_size(2)

% if target\_Mask\_pad(row\_con + round(yshift),col\_con + round(xshift)) == 1

% target\_label(row\_con + round(yshift),col\_con + round(xshift)) = label;

% label = label + 1;

% end

% end

% end

for row\_con = 1 : final\_size(1)

for col\_con = 1 : final\_size(2)

if target\_Mask\_pad(row\_con ,col\_con ) == 1

target\_label(row\_con ,col\_con) = label;

label = label + 1;

end

end

end

% figure(1);imshow(uint8(target\_label),[]);

%% 构造矩阵A和b

dim = label - 1;

nei = [-1 0;0 1;1 0;0 -1];

f = zeros(dim,1,3);

for path\_con = 1 : final\_size(3)

A = sparse(dim,dim);

b = zeros(dim,1);

cur = 0;

for row\_con = 1 : final\_size(1)

for col\_con = 1 : final\_size(2)

if target\_Mask\_pad(row\_con ,col\_con ) == 1

cur = cur + 1;

for nei\_con = 1 : 4

if target\_Mask\_pad(row\_con - nei(nei\_con,1),col\_con - nei(nei\_con,2)) == 1

A(cur,target\_label(row\_con - nei(nei\_con,1),col\_con - nei(nei\_con,2))) = -1;

else

b(cur,1) = b(cur,1) + background\_pad(row\_con - nei(nei\_con,1),col\_con - nei(nei\_con,2),path\_con);

end

end

b(cur,1) = b(cur,1) + target\_pad\_gradient(row\_con,col\_con,path\_con);

A(cur,cur) = 4;

end

end

end

f(:,:,path\_con) = A \ b;

end

%% 回填

target\_res = zeros(final\_size(1),final\_size(2),final\_size(3));

for row\_con = 1 : final\_size(1)

for col\_con = 1 : final\_size(2)

if target\_Mask\_pad(row\_con ,col\_con ) == 1

target\_res(row\_con,col\_con,1) = f(target\_label(row\_con,col\_con),1,1);

target\_res(row\_con,col\_con,2) = f(target\_label(row\_con,col\_con),1,2);

target\_res(row\_con,col\_con,3) = f(target\_label(row\_con,col\_con),1,3);

end

end

end

background\_pad = double(background\_pad);

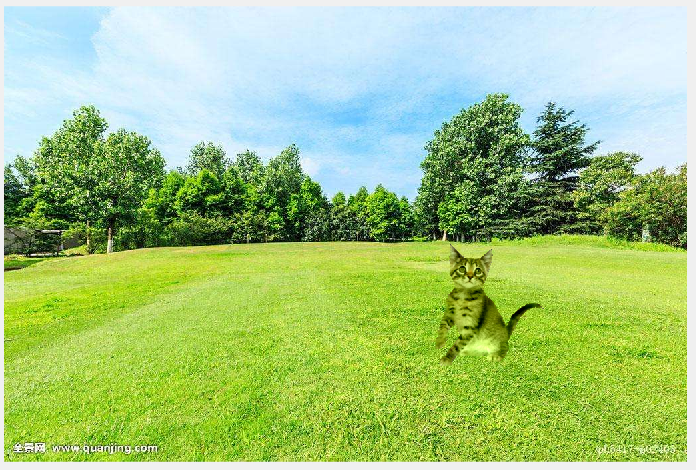
verse\_target\_Mask\_pad = (target\_Mask\_pad - 1) \* (-1);

result = target\_res + background\_pad .\* logical(verse\_target\_Mask\_pad);

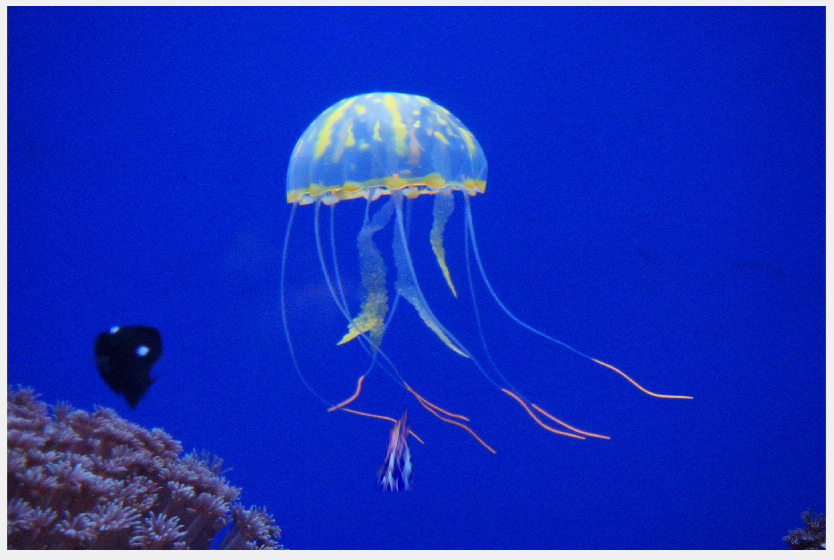
figure(1),imshow(uint8(result));

1. 找到自己感兴趣的4组照片，从中选择物体进行融合。

结果







这是一团火

