

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

一、 实现课上所讲述的 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

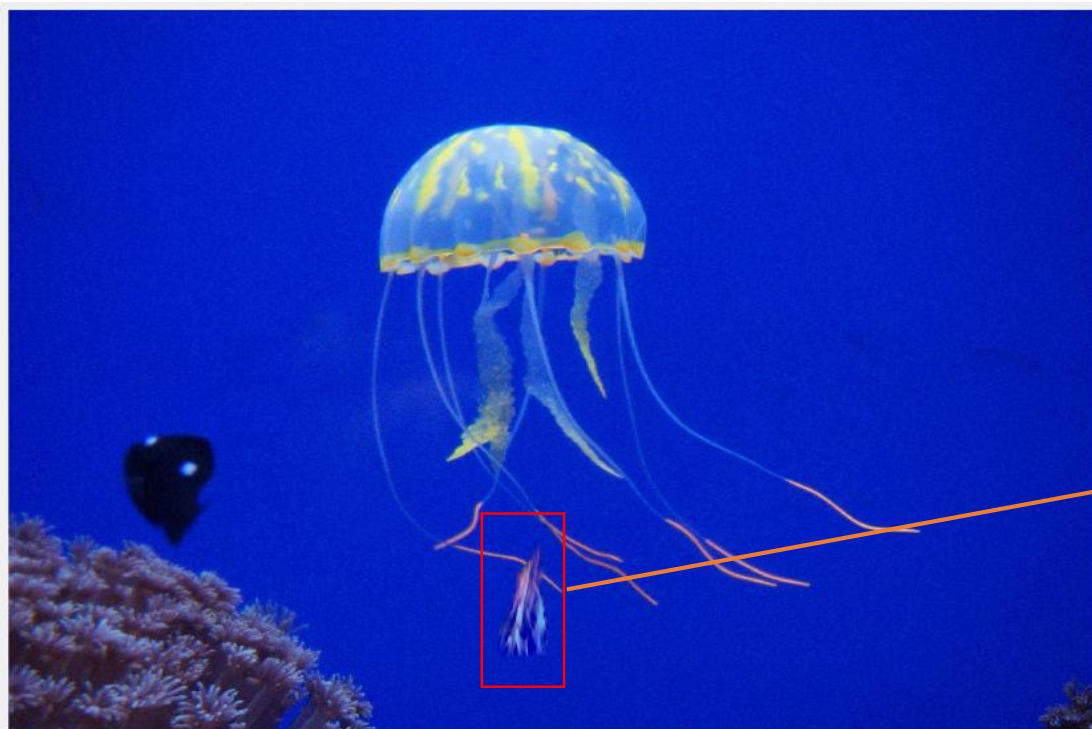
background_pad = double(background_pad);
verse_target_Mask_pad = (target_Mask_pad - 1) * (-1);
result = target_res + background_pad .* logical(verse_target_Mask_pad);
figure(1),imshow(uint8(result));

```

二、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

