

前景后景分割

姓名：王浩然

学号：1700012937

代码思路

- 基础：K-聚类图片分割技术
 - 开始的时候用随机数生成分类中心点的初始位置
 - 计算图片中的像素点在RGB空间当中和分类中心点的距离
 - 根据距离进行分类，将分类的结果保存在mask矩阵当中
 - 根据分类的结果对分类的中心点进行调整
 - 递归重复上述步骤直到分类中心点的位置逐渐稳定

```
function [newcen,newmask] = kmeans( r,g,b,oldcen,oldmask )
[h,w,n]=size(oldmask);
tempmask=zeros(h,w,n);
dist=zeros(h,w,n);
tempcen=zeros(3,n);

%计算点到所有的中心点的距离矩阵
for i=1:n
    cr=oldcen(1,i);
    cg=oldcen(2,i);
    cb=oldcen(3,i);
    dist(:,:,i)=cal_dist(r,g,b,cr,cg,cb);
end

%根据距离进行分类,分类结果存在newmask当中
for i=1:h
    for j=1:w
        [x,idx]=min(dist(i,j,:));
        tempmask(i,j,idx)=1;
    end
end

%根据分类将中心点调整
for i=1:n
    tm=double(tempmask(:,:,i));
    tr=tm.*double(r);
```

```

    tg=tm.*double(g);
    tb=tm.*double(b);
    tpnum=sum(tm(:,:));
    cr=sum(tr(:,:))/tpnum;
    cg=sum(tg(:,:))/tpnum;
    cb=sum(tb(:,:))/tpnum;
    tempcen(1,i)=cr(1);
    tempcen(2,i)=cg(1);
    tempcen(3,i)=cb(1);
end

bias=0;
%递归
for i=1:n
    tbias=tempcen(:,i)-oldcen(:,i);
    bsum=(tbias(1)*tbias(1)+tbias(2)*tbias(2)+tbias(3)*tbias(3))^0.5;
    bias=bsum;
end

if bias>1
    [newcen,newmask]=kmeans(r,g,b,tempcen,tempmask);
else
    newcen=tempcen;
    newmask=tempmask;
end

end
end

```

- 利用背景往往占大多数的特性，对分类的mask，进行筛选拼接

```

function outres = kmeansig( inrgb,n )
rgb=double(inrgb);
r=rgb(:,:,1);
g=rgb(:,:,2);
b=rgb(:,:,3);
[h,w,t]=size(rgb);
mask=zeros(h,w,n);
outres=zeros(h,w);
%随机生成类的中心
cen=zeros(3,n);
for i=1:n
    cen(:,i)=randi(255,3,1);
end
[cen,mask]=kmeans(r,g,b,cen,mask);

```

```

summax=0;
maxidx=0;
fixgate=h*w*0.20;

fnin='mask';

%筛选mask, 并拼接出outres上
for idx=1:n
    fn=sprintf('%s%d.jpg',fnin,idx);
    imwrite(mask(:,:,idx),fn);
    sumcurr=sum(sum(mask(:,:,idx),1),2);
    flag=sumcurr>summax;
    if flag
        summax=sumcurr;
        maxidx=idx;
    end
end

for idx=1:n
    thismask=mask(:,:,idx);
    sumcurr=sum(sum(thismask(:,:,2),1));
    flag=sumcurr<summax*0.6 && idx~=maxidx && sumcurr<fixgate;
    if flag
        outres=outres+mask(:,:,idx);
    end
end
end

```

- 加入图片的滤波预处理并重复多次生成进一步去除噪音

```

function voter = buildbg( inrgb,r )
f1 =fspecial('gaussian',20,1000);
rgb=imfilter(inrgb,f1,'replicate');
[h,w,t]=size(rgb);
voter=zeros(h,w);
for idx=1:r
    voter=voter+kmeansig(rgb,15);
end
voter=voter/r;
voter=int8(voter);
voter=double(voter);
imwrite(voter,'voter.jpg');

end

```

实验结果

代码测试方法

```
>> test(300937);
```

%等待运行一分钟左右以后，会在当前的文件夹下产生一个outres文件

在多数的图片上的测试效果都很好，但仍然有一些图片效果不是很理想，后续可以尝试一下用神经网络来实现前景背景的分割算法

