

《数字图像处理》小作业 5

药 3 李宇晖 2013012470

实验目的

- 1) 了解图像彩色图像原理的基本模型和原理
- 2) 能对彩色图像进行切割、虚化等处理

实验过程

题目一、对人物图像(messi.jpg)的背景做更大程度虚化（尝试多种尺寸的滤波器）。

首先通过手工交互式的方式勾出前景人物的轮廓，matlab 中 `ginput()`和 `roipoly()`函数可以方便地达到目的。代码如下：

```
function [ binary_image ] = Get_FrontGround( image )
    close all;
    imshow(image);
    [x,y] = ginput;
    binary_image = roipoly(image,x,y);
    imwrite(binary_image,'binary.bmp');
    binary_image = 1 - binary_image;
    imwrite(binary_image,'without_front.bmp');
end
```

得到二值图如下：



在 main 函数中，将原图点乘 1-二值图即可得到前景，点乘二值图即可得到背景。实现代码如下：

```
messi_image = imread('messi.jpg'); %read image
[m,n,c] = size(messi_image);
%binary_image = Get_FrontGround(messi_image); %get binary front image
for k = 1:3,
    binary_image(:, :, k) = imread('without_front.bmp')/255;
end
front = uint8(messi_image).* uint8((1-binary_image)); %get front
back = uint8(messi_image).* uint8(binary_image); %get back
```

之后，使用循环的方式采用不同大小的滤波器对彩色图像进行平滑，`imfilter()`函数可以达到目的。同时在 RGB 通道对图像进行平滑，和在 HIS 空间下对图像的亮度进行平滑，继而进行

比较。代码如下：

```
hsize_list = [20,40,60,80,100];
for k = 1:5,
    hsize = hsize_list(k); % size of averaging filter
    h = ones(hsize,hsize)/(hsize*hsize); %filter

    J = im2double(back);
    J_hsv = rgb2hsv(J); %get hsv image
    J2_hsv = J_hsv;
    J2_hsv(:,:,3) = imfilter(J_hsv(:,:,3),h); %smooth intensity of HSI
    J2 = hsv2rgb(J2_hsv);

    J1 = im2double(messi_image); %initilize
    for i = 1:3
        J1(:, :, i) = imfilter(J(:, :, i), h); %smooth RGB channel
    end

    figure(1), clf;
    J1 = im2double(front)+J1;
    J2 = im2double(front)+J2;
    subplot(2,2,1), imshow(messi_image,[]), title('original image');
    subplot(2,2,2), imshow(J1,[]), title('smooth RGB channels');
    subplot(2,2,3), imshow(J2,[]), title('smooth intensity of HSI only')
    saveas(figure(1), [num2str(k) '.jpg']);
    close all;
end
```

结果如下：

滤波器大小 20*20

original image



smooth RGB channels



smooth intensity of HSI only



滤波器大小 40*40

original image



smooth RGB channels



smooth intensity of HSI only



滤波器大小 60*60

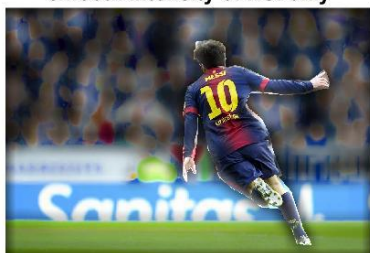
original image



smooth RGB channels

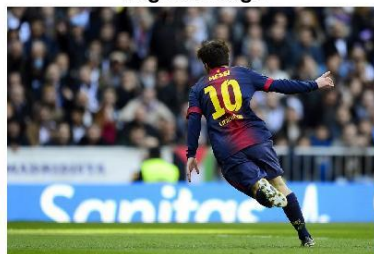


smooth intensity of HSI only



滤波器大小 80*80

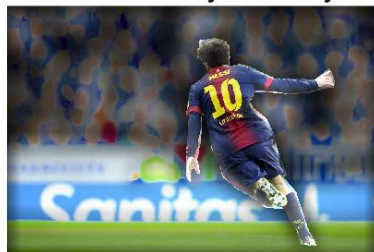
original image



smooth RGB channels



smooth intensity of HSI only



滤波器大小 100*100



可以看出，滤波器大小越大，图像的虚化效果越好。RGB 模型下对图像的模糊是全局的，而在 HIS 模型下，近景的图像模糊效果并不如远景。

题目二、对斗牛图像(bull.jpg)在 HSI 颜色空间中进行彩色切割增强。在输出图像中，除红色斗篷外，其他区域以灰度显示。提示：可进行简单交互（例如手工标注一个矩形）

首先使用 `rgb2hsv()`函数将图像转到 HIS 颜色空间中，并分别作出色调、饱和度、亮度的图像。
代码如下：

```
I=imread('bull.jpg');  
I = im2double(I);  
J = rgb2hsv(I); %get hsi image  
J1=J;  
M = size(J,1);N = size(J,2);  
figure(6),clf; %show the HSI  
subplot(1,3,1),imshow(J1(:,:,1),[]);  
subplot(1,3,2),imshow(J1(:,:,2),[]);  
subplot(1,3,3),imshow(J1(:,:,3),[]);
```

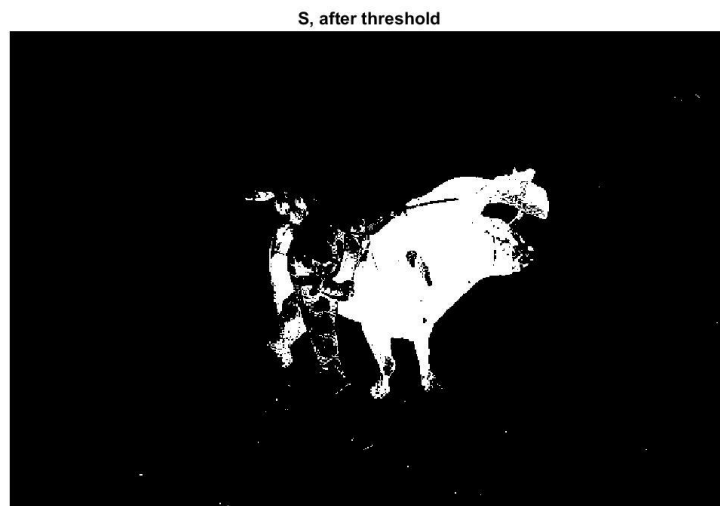
得到的结果如下：



可以发现，色调部分最容易看出红色斗篷与其他部分的差别，饱和度可以将红色斗篷和牛与其他外界噪声分开。所以首先使用阈值处理饱和度图像产生二值模板，代码如下：

```
threshold = 0.85; %the threshold for saturation
index = find(J(:, :, 2) > threshold);
roi = false(size(J(:, :, 2)));
roi(index) = 1;
J(:, :, 2) = J(:, :, 2) & roi;
figure(1), imshow(J(:, :, 2)), title('S, after threshold');
```

结果如下：



接下去，再分别使用 `impoint` 函数交互地选中色调图像中红色斗篷和牛的部分，采用阈值处理色调图像，同样生成二值图像，代码如下：

```
figure(2), imshow(J(:, :, 1)), title('picking point');
% choose point of interest
h = impoint;
pos = wait(h);
pos = round(pos);
a = J(pos(2), pos(1), 1);

h = impoint;
```

```

pos = wait(h);
pos = round(pos);
b = J(pos(2),pos(1),1);

threshold = (b-a)*2/3+a; %the threshold for hue
index = find(J(:,1)>threshold);
roi = false(size(J(:,1)));
roi(index) = 1;
J(:,1) = J(:,1)&roi;
figure(3), imshow(J(:,1)),title('H after threshold');

```

结果如下：



将这两个二值图像进行点击，就可以得到较为理想的红色斗篷范围。代码如下：

```

X = J(:,1).*J(:,2); %image for choose
figure(4),imshow(X);

```

之后使用 `imrect` 函数交互地画出感兴趣的矩形边界，与上述得到的模板合并。而后将原图的三个通道的色彩图像的这部分色彩保留，剩下变为灰色图像。代码如下：

```

h = imrect; %square the position
pos = wait(h);
pos = round(pos);
roi = false(size(X));
roi(pos(2):pos(2)+pos(4),pos(1):pos(1)+pos(3)) = 1;
mask = X&roi; %get the mask

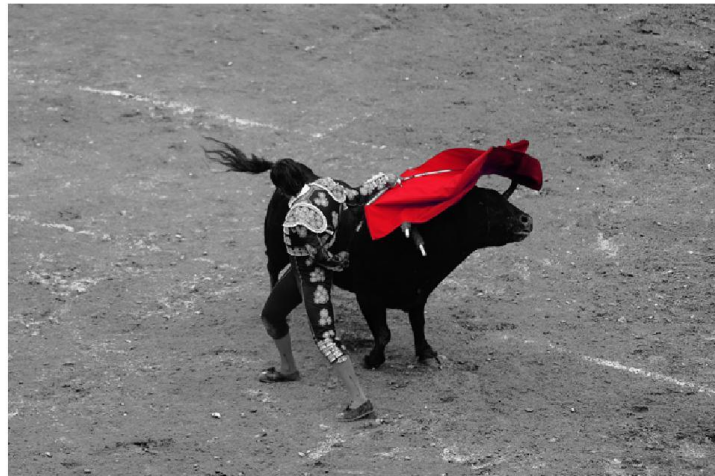
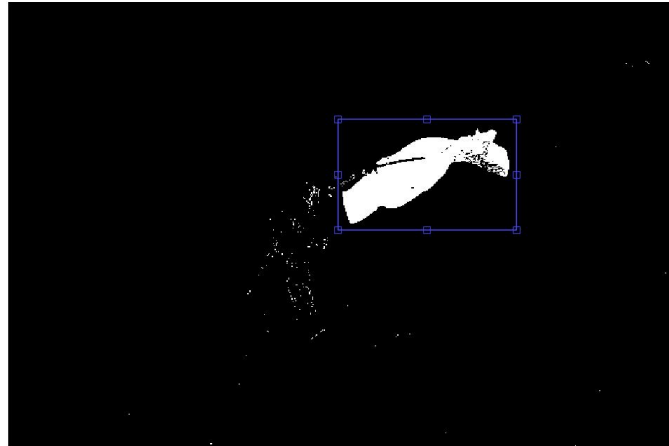
g = rgb2gray(I);
G = I;
G(:,1) = g;
G(:,2) = g;
G(:,3) = g;

```



```
index = find(mask);  
G(index) = I(index);  
G(index+M*N) = I(index+M*N);  
G(index+M*N*2) = I(index+M*N*2);  
  
figure(5),imshow(G);
```

结果如下：



可以看出，在 **HIS** 空间内可以简单通过阈值的方法交互地达到彩色图像切割增强的效果。但是，仍然会有小的误差，通常在 **RGB** 的空间内会有更好的效果。在 **RGB** 空间内，对于色彩可以采用欧式距离的方式进行度量，从而和 **HSI** 颜色空间相似的方法得到结果。代码如下：

```
I=imread('bull.jpg');
```



```

I = im2double(I);
M = size(I,1);
N = size(I,2);
figure(1),imshow(I);

% choose point of interest
h = impoint;
pos = wait(h);
pos = round(pos);
a = [I(pos(2),pos(1),1) I(pos(2),pos(1),2) I(pos(2),pos(1),3)];
R = 100/255;% radius

D = (I(:, :, 1)-a(1)).^2+(I(:, :, 2)-a(2)).^2+(I(:, :, 3)-a(3)).^2;
mask = D<=R*R;

% choose ROI
figure(2),imshow(mask);
h = imrect;
pos = wait(h);
pos = round(pos);
roi = false(size(mask));
roi(pos(2):pos(2)+pos(4),pos(1):pos(1)+pos(3)) = 1;
mask = mask & roi;

g = rgb2gray(I);
J = I;
J(:, :, 1) = g;
J(:, :, 2) = g;
J(:, :, 3) = g;
idx = find(mask);
J(idx) = I(idx);
J(idx+M*N) = I(idx+M*N);
J(idx+M*N*2) = I(idx+M*N*2);
figure(3),imshow(J,[]);

```

结果如下：

