



数字图像处理

基于颜色空间的人脸检测

RGB颜色空间 施天予

HSV颜色空间 孙奥远

YCbCr颜色空间 覃浩南



Outline



1 RGB颜色空间人脸检测

2 HSV颜色空间人脸检测

3 YCbCr颜色空间人脸检测



中山大學

SUN YAT-SEN UNIVERSITY

Part1

RGB颜色空间人脸检测

施天予



➤ 原理

RGB 颜色空间的人脸检测，即给定一张图片，将其中的人脸框出来。我实现的方法主要基于 RGB 颜色空间人脸检测的阈值，将原图像做阈值变换转为二值图，对应人脸的部分为 1，其他部分为 0。在用方框把人脸框出来。

在 RGB 颜色空间中的人脸检测需要满足以下条件：

$$\left\{ \begin{array}{ll} R > 95 \wedge G > 40 \wedge B > 20 \wedge (\max\{R, G, B\} - \min\{R, G, B\}) > 15 \\ \wedge |R - G| > 15 \wedge R > G \wedge R > B & \text{均匀照明} \\ R > 20 \wedge G > 210 \wedge B > 170 \wedge |R - G| < 15 \wedge R > G \wedge R > B & \text{潜在照明} \end{array} \right.$$

整体流程包括以下几步：

1. 图像预处理
2. 图像颜色空间的转换
3. 基于阈值的图像分割
4. 连通分量分析
5. 标注出人脸并显示

```

1 - close all;clear all;clc;
2 -
3 - I1 = imread('1.jpeg');
4 - BW_rgb1 = rgb_face(I1);
5 - bb_rgb1 = get_bb(BW_rgb1);
6 - I2 = imread('2.jpeg');
7 - BW_rgb2 = rgb_face(I2);
8 - bb_rgb2 = get_bb(BW_rgb2);
9 - I3 = imread('3.jpeg');
10 - BW_rgb3 = rgb_face(I3);
11 - bb_rgb3 = get_bb(BW_rgb3);
12 -
13 - figure, subplot(1,2,2);
14 - imshow(BW_rgb1)
15 - title('分割图');
16 - subplot(1,2,1);
17 - imshow(I1);
18 - rectangle('Position',bb_rgb1,'EdgeColor','r');
19 - title('基于RGB的人脸检测');
20 -
21 - figure, subplot(1,2,2);
22 - imshow(BW_rgb2)
23 - title('分割图');
24 - subplot(1,2,1);
25 - imshow(I2);
26 - rectangle('Position',bb_rgb2,'EdgeColor','r');
27 - title('基于RGB的人脸检测');
28 -
29 - figure, subplot(1,2,2);
30 - imshow(BW_rgb3)
31 - title('分割图');
32 - subplot(1,2,1);
33 - imshow(I3);
34 - rectangle('Position',bb_rgb3,'EdgeColor','r');
35 - title('基于RGB的人脸检测');
36 -

```

```

37 - function bw = rgb_face(I)
38 -     [m,n,c] = size(I); % 获取图像
39 -     BW = zeros(m,n);
40 -     for i = 1:m
41 -         for j = 1:n
42 -             R = I(i,j,1);
43 -             G = I(i,j,2);
44 -             B = I(i,j,3);
45 -             v = [R,G,B];
46 -             if (R > 95 && G > 40 && B > 20) && ((max(v) - min(v) > 15 && abs(R-
47 -                 % if R > 20 && G > 210 && B > 170 && abs(R-G) < 15 && R > G && % R
48 -                     BW(i,j) = 1;
49 -             end
50 -         end
51 -     end
52 -     bw = BW;
53 - end
54 -
55 - function g = get_bb(BW)
56 -     L = bwlabel(BW,8); % 返回和BW相同大小的8连通矩阵
57 -     BB = regionprops(L,'BoundingBox'); % 返回最小矩阵的结构体
58 -     BB1 = struct2cell(BB); % 将结构体转换为元胞数组
59 -     BB2 = cell2mat(BB1); % 元胞数组转为矩阵
60 -     % 二值图所以连通只有1,大小1*4n,每4个[x,y,w,h]代表矩形的从点(x,y)开始绘制一个
61 -
62 -     [s1,s2] = size(BB2);
63 -     max_area = 0;
64 -     j = 3;
65 -     for k = 3:4:s2-1
66 -         area_bb = BB2(1,k) * BB2(1,k+1); % 计算值矩阵大小
67 -         if area_bb > max_area && (BB2(1,k) / BB2(1,k+1)) < 1.8
68 -             max_area = area_bb;
69 -             j = k;
70 -         end
71 -     end
72 -     g = [BB2(1,j-2),BB2(1,j-1),BB2(1,j),BB2(1,j+1)];
73 - end

```



➤ 实验结果展示

基于RGB的人脸检测



分割图





➤ 实验结果展示

基于RGB的人脸检测



分割图



➤ 实验结果展示

基于RGB的人脸检测



分割图





中山大學

SUN YAT-SEN UNIVERSITY

Part2

HSV颜色空间人脸检测

孙奥远



➤ 题目描述

在HSV颜色空间下进行基于肤色的亚洲人的人脸检测

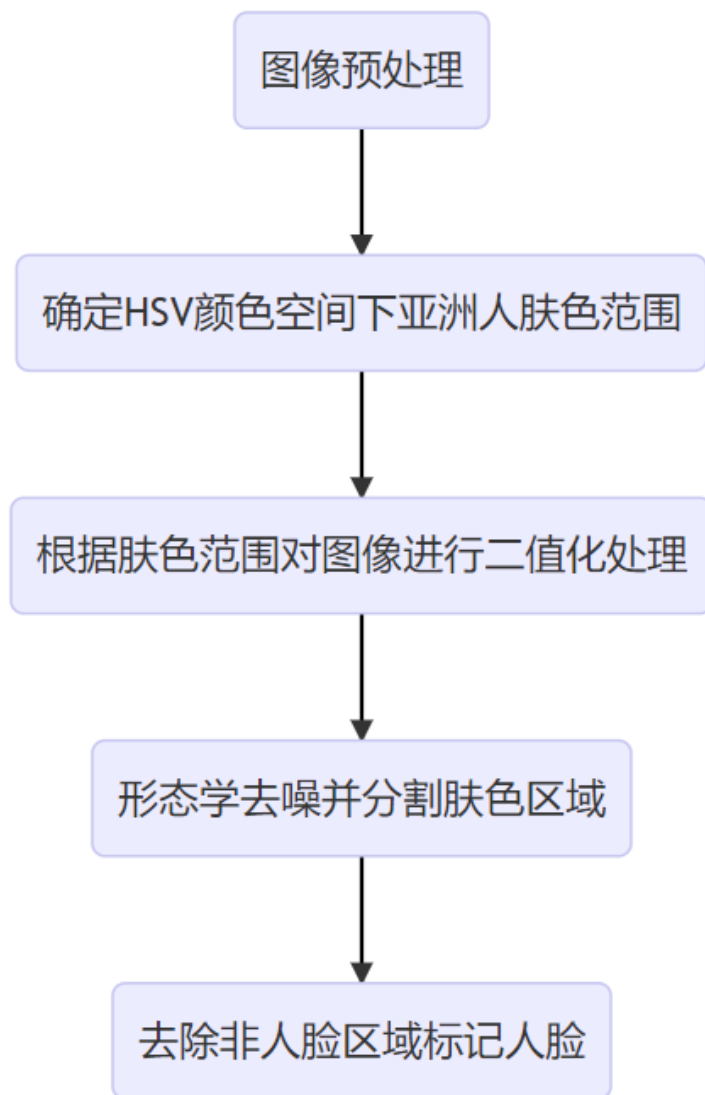
➤ 原理

为了能够利用肤色进行人脸检测，就希望找到一个颜色空间，使得人类的肤色能够聚成密集分布的一类，即使得人的肤色在该颜色空间中的颜色特征有聚类特性

在HSV颜色空间下，亚洲人脸颜色的范围为

$0 \leq h \leq 0.25, 0.23 \leq s \leq 0.68, v \geq 0.4$

因此，只要将h, s, v三分量取值在该范围内的像素标记为1，其他像素标记为0，就可以将人脸提取出来



```
image = imread('figure1.bmp');%RGB图像%
[m,n,~] = size(image);
image2 = WhiteBalance(image);%对图像进行白平衡
image_hsv = rgb2hsv(image2);%将RGB转换为HSV图像

BW = zeros(m,n);
h = image_hsv(:, :, 1);
s = image_hsv(:, :, 2);
v = image_hsv(:, :, 3);

BW((h<=0.25)&(h>=0)&(s>=0.23)&(s<=0.68)&v>=0.4) = 1;%当0.25>=h>=0, 0.23<=s<=0.68, v>=0.4时认为是人脸区域,得到二值化图像
se = strel('disk',5);%形态学结构元素
BW2 = imerode(BW,se); %腐蚀操作
BW3 = imdilate(BW2,se);%膨胀操作

figure;
subplot(2,4,1);imshow(image);title('原图');
subplot(2,4,2);imshow(image2);title('预处理');
subplot(2,4,3);imshow(BW);title('二值化图');
subplot(2,4,4);imshow(BW2);title('腐蚀');
subplot(2,4,5);imshow(BW3);title('膨胀');
subplot(2,4,6);imshow(image);title('人脸检测图');
pos = DrawRect(BW3);%在图像上找到人脸区域并标记
for i =1:size(pos,1)
    rectangle('Position',pos(i,:), 'edgecolor','r');
end
```

HSV颜色空间



中山大學

SUN YAT-SEN UNIVERSITY

%计算人脸图像的R, G, B三个通道的均值avgR, avgG, avgB和R, G, B的总均值avg。

%计算R, G, B三通道的调整系数，a1=avg/avgR, a2=avg/avgG, a3=avg/avgB

%. 将原图像的R, G, B三通道的取值分别乘以a1, a2, a3。超过255的按照255计算。

function rgbimage = WhiteBalance(image)%对图像进行光照补偿

```
[m,n,~] = size(image);
```

```
r = image(:, :, 1);
```

```
g = image(:, :, 2);
```

```
b = image(:, :, 3);
```

```
rmean = sum(r)/(m*n);
```

```
gmean = sum(g)/(m*n);
```

```
bmean = sum(b)/(m*n);
```

```
avgGray = (rmean+gmean+bmean)/3;
```

```
Kr = avgGray/rmean;
```

```
Kg = avgGray/gmean;
```

```
Kb = avgGray/bmean;
```

```
r = r*Kr;
```

```
g = g*Kg;
```

```
b = b*Kb;
```

```
r(r>255) = 255;
```

```
g(g>255) = 255;
```

```
b(b>255) = 255;
```

```
rgbimage = cat(3, r, g, b);
```

```
end
```



```
function result = DrawRect(image)%在image上找到人脸区域并标记
    [L,num] = bwlabel(image,8);%找到8连通区域,L:与image相同大小的标记矩阵,num:连通区域数目
    [m,n] = size(image)
    temp = regionprops(L,'area','boundingbox');%返回L中包含连通区域的最小矩形框的位置和大小的数组
    p = 1;
    result = zeros(p,4)
    for i = 1:num
        box = temp(i).BoundingBox;
        x = box(1);
        y = box(2);
        w = box(3);
        h = box(4);
        subimage=image(int64(y):min(int64(y+h)-1,m),int64(x):min(int64(x+w)-1,n));%取出二值图像,计算疑似人脸的面积
        ratio = h/w;%矩形框高与宽的比值,在[1,1.8]之间认为是人脸区域
        lambda = sum(sum(subimage))/(w*h);%疑似人脸的面积/矩形框的面积,在[0.5,0.85]之间认为是人脸区域
        if ratio >= 1 && ratio <= 1.8 && lambda >= 0.5 && lambda <= 0.85
            result(p,:) = [x y w h];%取出坐标值,宽度,高度
            p = p+1;
        end
    end
end
```



➤ 实验结果展示

原图



人脸检测图





中山大學

SUN YAT-SEN UNIVERSITY

Part3

YCbCr颜色空间人脸检测

覃浩南



➤ 原理

YCbCr 有的时候会被写作为 YCBCR，是色彩空间的一种。它通常会用于影片中的影像连续处理，或是数字摄影系统中。

Y 为颜色的亮度 (luma) 成分，而 CB 和 CR 则为蓝色和红色的浓度偏移量成分。

YCbCr 颜色空间的人脸检测方法类似 RGB：首先将原图像根据 YCbCr 的参数条件做阈值变换转化为二值图，人脸部分的像素为 1，其他部分为 0，最后用方框把人脸框出来。

在网上查询相关资料得出，人脸部分在 YCbCr 颜色空间需要满足的参数条件为：

$$140 < cr < 160$$

注意要确保图片读入格式为 RGB 三通道格式，否则图片无法转换为 YCbCr 格式。



```
close all;clear all;clc;
```

```
I1 = imread('./fig/test1.jpg');  
BW_ycbcr1 = YCbCrTransform(I1);  
bb_ycbcr1 = get_bb(BW_ycbcr1);
```

```
I2 = imread('./fig/test2.jpg');  
BW_ycbcr2 = YCbCrTransform(I2);  
bb_ycbcr2 = get_bb(BW_ycbcr2);
```

```
I3 = imread('./fig/test3.jpg');  
BW_ycbcr3 = YCbCrTransform(I3);  
bb_ycbcr3 = get_bb(BW_ycbcr3);
```

```
figure, subplot(1,2,1);  
imshow(BW_ycbcr1)  
title('二值图');  
subplot(1,2,2);  
imshow(I1);  
rectangle('Position',bb_ycbcr1,'EdgeColor','r');  
title('基于YCbCr的人脸检测');
```

```
figure, subplot(1,2,1);  
imshow(BW_ycbcr2)  
title('二值图');  
subplot(1,2,2);  
imshow(I2);  
rectangle('Position',bb_ycbcr2,'EdgeColor','r');  
title('基于YCbCr的人脸检测');
```

```
function bw = YCbCrTransform(I)
    I_y = rgb2ycbcr(I);
    [m,n,c] = size(I_y);           % 获取图像大小
    BW = zeros(m,n);               % 二值图
    for i = 1:m
        for j = 1:n
            y = I_y(i,j,1);
            cb = I_y(i,j,2);
            cr = I_y(i,j,3);
            if (140 < cr && cr < 160)
                BW(i,j) = 1;
            end
        end
    end
    bw = BW;
end
```

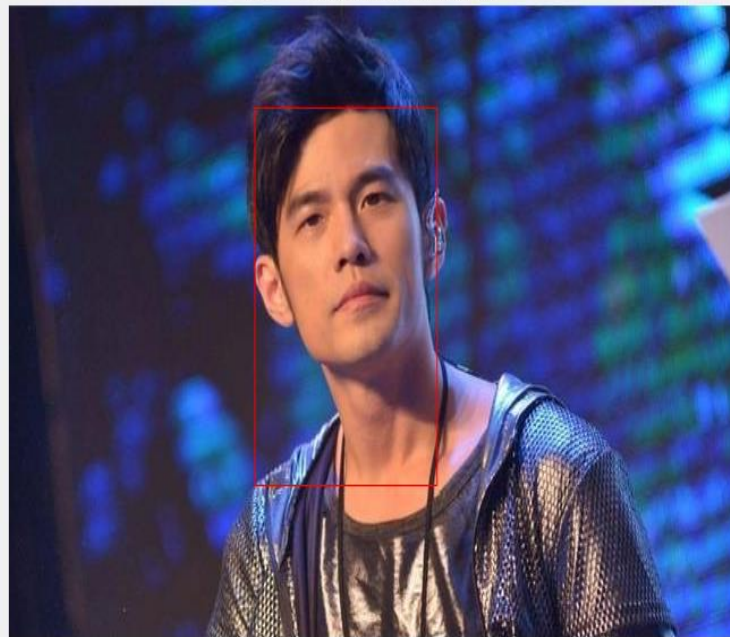


➤ 实验结果展示

二值图



基于YCbCr的人脸检测

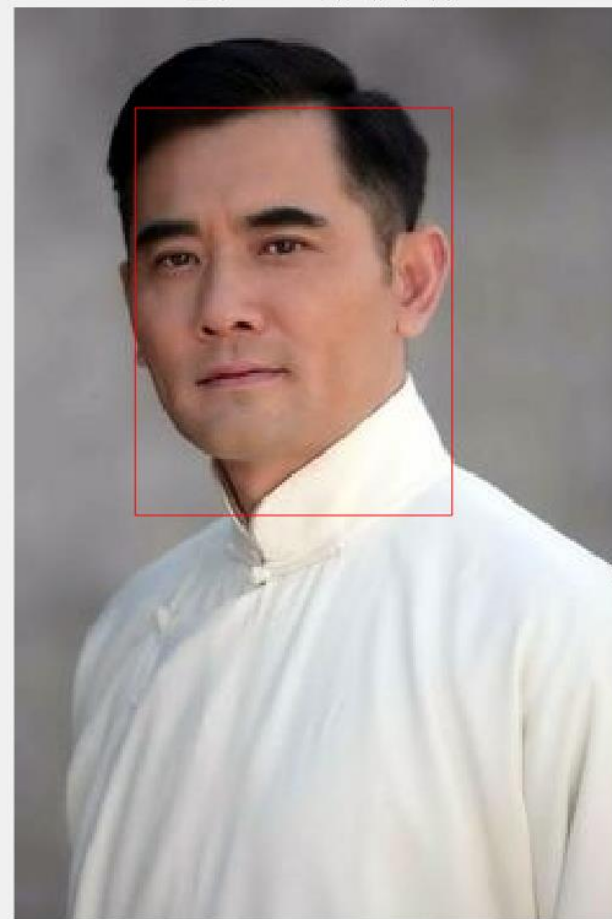




二值图

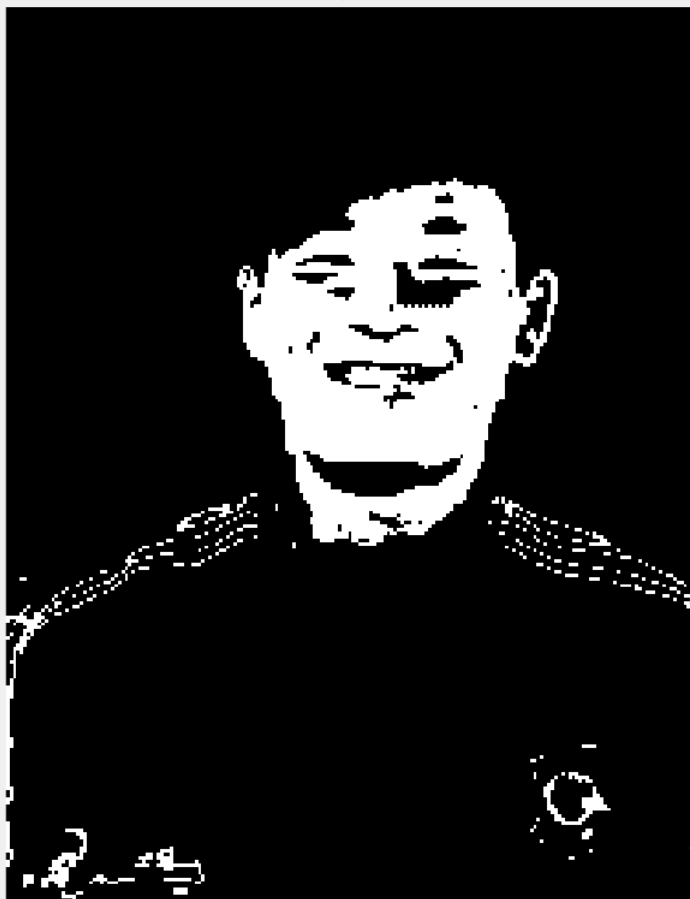


基于YCbCr的人脸检测





二值图



基于YCbCr的人脸检测

