FH Vorarlberg
University of Applied Sciences

Color based face detection

Digital Signal Processing Project

University of Applied Sciences Vorarlberg
Master in Mechatronics

Submitted to

Prof. (FH) DI Dr. Reinhard Schneider

Handed in by

Burtscher Tobias
Stark Stefan

Dornbirn, 07.12.2016

Table of Contents

List of Figures

Bibliography

Hjelmas, E. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–247.

Kumar, P. and M, S. (2014). Real time detection and tracking of human face using skin color segmentation and region properties. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, 6(8):40–46.

Marius, D., Pennathur, S., and Rose, K. (2000). Face detection using color thresholding, and eigenimage template matching. *Stanford University*.

Singh, S. K., Chauhan, D. S., Vatsa, M., and Singh, R. (2003). A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, 6(4):227–234.

ZHAO, W., CHELLAPPA, R., P.J.Phillips, and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):339–458.

1. Problem description

1.1 Overview

According to the articel *Face recognition: A literature survey* from ZHAO et al. (2003), face recognition can be segmented into three key steps, shown in figure 1.

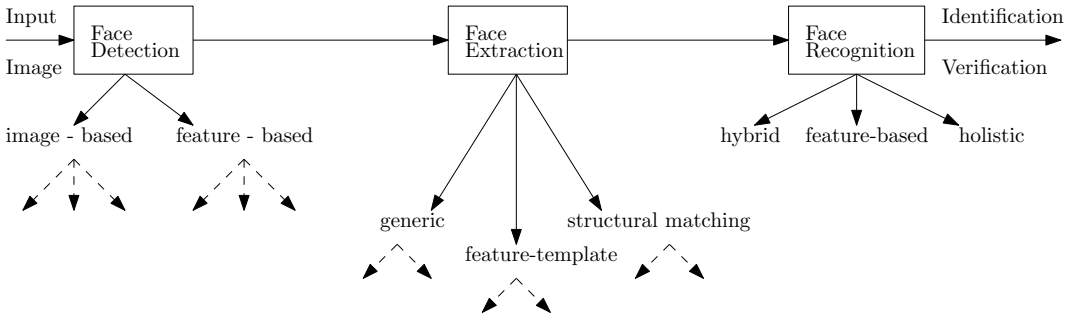


Figure 1: Face Recognition Progress

Face Detection is responsible for a rough normalization (like face tracking) and use for this task different approaches.
Face Extraction generates a more accurate normalization (like human emotions). The different approaches to get this emotions are shown in figure 1. Face detection and face extraction approaches can use the same feature-based-method (like informations out of color, Motion, ...)so they can perform simultaneous.
Face Recognition is the last step to identify/verify a picture. For a verification/identification several methods are available.

1.2 Face Detection

We decided to have a closer look on the face detection process because for the processes afterwards we need a detected face, which is not available without any effort.
To find an approach which we can study, implement and test we made further researches in this segment. The article *Face detection: A survey* from Hjelmas (2001) gives a good overview of the topic face detection. The figure 2 (out of Hjelmas (2001)) represents the different approaches to detect faces in a picture.

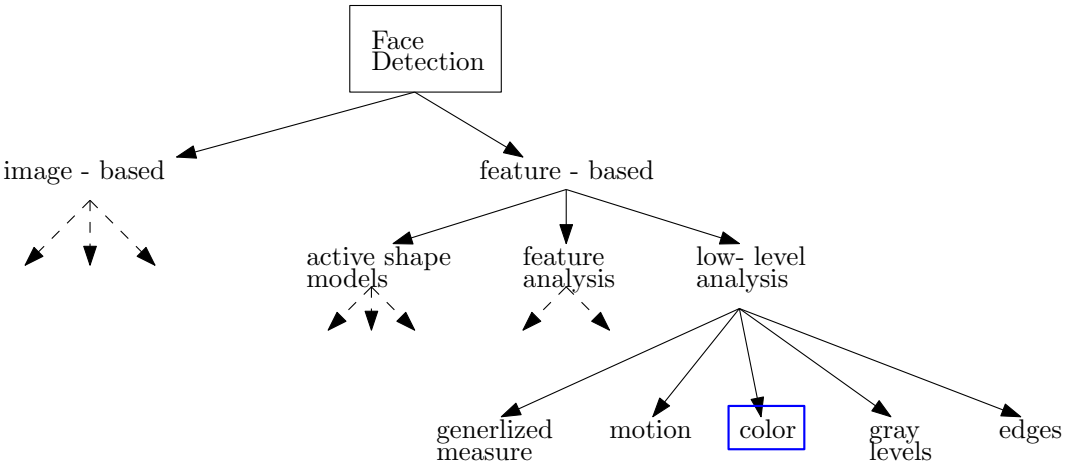


Figure 2: Face Detection divided into approaches (more detailed from Hjelmas (2001)).

According to Hjelmas (2001) are **Image-based approaches** the moste robust techniques for gray images, but on the other side they need a lot of computation time by multiresolution window scanning.
The **feature-based approaches** were the first attemps in the face detection history. They are built up simple and so they need less computation time, this enables these approaches access to real-time applications.

The most interesting approach for us was *Face detection based on color likelihood* approach (in figure 2 marked as *Color*). Argumentation for this algorithm can be found in the section 1.3;

1.3 Color based face detection

According to the article *A Robust Skin Color Based Face Detection Algorithm* of Singh et al. (2003) following points argue for the color based case detection

- Color processing is much faster than other facial features
- Color based algorithm is orientation invariant, that means that a motion estimation is much easier.
- Color based algorithm is often the first step for detection, this algorithm is popular.

An application for the simple and real-time capable algorithmus face detection with color can be found in the articel *Face recognition: A literature survey* from ZHAO et al. (2003)

> *In video conferencing systems, there is a need to automatically control the camera in such a way that the current speaker always has the focus. One simple approach to this is to guide the camera based on sound or simple cues such as motion and skin color.*

2. Literature Analysis

The literature analysis began with the topic selection (see chapter 1). The supervisor told us that the initial chosen topic *Face Detection* is to big to treat within one semester, so the first literature research was done to find a specific topic to handle.
The second literature research was done to find information about the chosen topic.

2.1 Approach

All interesting literature which were found and marked as interesting (by scanning the abstract) were saved in a list on the Ilias project space. This articles were read in a more detail afterwards.
The structure of the table (see figure 3 make additional sorting (by exporting/copying into an EXCEL) possible and the implementation on ILIAS makes it possible to get access easily to the actual table.

Literature Research (LR)

ID	Date	Topic	Source	site	comment
1	22.10.2016	general	olav: face recogintion database	Science direct - On internal representations in face recognition systems	analysis of face recognition systems; mentoined databases: FERET and Face database info MIT
2	27.10.2016	general	google: face recoginition overview	Face Recognition: A Literature Survey	nice overview about face recognition (split into Detection, extraction and Recognition) -> Useful to search a more detailed topic.
3	27.10.2016	Face detecion	olav: face detecion	ScienceDirect - Computer Vision and Image Understanding - Face Detection: A Survey	Good overview about different approaches to detect faces
4	27.10.2016	Face detecion - color	olav: face detecion	ScienceDirect - Pattern Recogintion - Face detection based on skin color likelihood	Face detection based on color likelihood - approach of: Face detection -> Feature-based approaces -> low level analysis -> color

Figure 3: Literature reasarech table - 30.10.2016.

All literatures which were mentioned in this document are also listed in the bibliography.

2.2 LR color based face detection

2.2.1 General Information about Face detection/Face recognition

The articles in subsections gives an overview of the topic face recognition and face detection. These articles were used to define the chosen topic: "Color based face detection".

- *Face Detection: A Survey* from Hjelmas (2001)
- *Face Recognition: A Literature Survey* from ZHAO et al. (2003)

2.2.2 Color based face detection

The article *A Robust Skin Color Based Face Detection Algorithm* from Singh et al. (2003) compares the three different color models (RGC, YCbCr and HSI). According to this article the color model YCbCr is widely used in the digital video domain and is more accurate (in detecting faces) than the other to color models. These two arguments are the reason why the color model YCbCr was chosen for this project.

A article which describes step by step how an approach of the color based face detection can be implemented can be found in the article *Face Detection Using Color Thresholding, and Eigenimage Template Matching* from Marius et al. (2000). In this article thresholds for the Cb and Cr are defined.

A different approach to morphological operations can be found in the article *Real Time Detection and Tracking of Human Face using Skin Color Segmentation and Region Properties* from Kumar and M (2014). This article defines also thresholds for the Cb and Cr.

3. Test scenario

3.1 Target

The target of the implementation is to use the video from a web cam to test the implemented color based face detection. The implemented code should as simple as possible, to achieve a real-time capable

3.2 Implementation steps

Following steps will be done to test if the implemented solution is real-time capable.

3.2.1 Color based face detection on a picture

The first step is to test the algorithms on different pictures. For this following steps are scheduled:

1. Transform picture into the YCbCr color space.
2. Find suitable threshold ranges for the YCbCr.
3. Make Thresholding on the YCbCr to get a binary picture.
4. Detecting faces out of skin regions.
5. Draw boxes to identify the faces on the picture.

3.2.2 Color based face detection on a video

Use a web cam and test the implemented color based algorithm.

3.2.3 Real-time Color based face detection on a picture

There are three possibilities to test if the implemented solution is real-time capable.

1. Test it on a notebook and a web cam (same as section 3.2.2).
2. Calculate if the computation time is less enough for a micro controller like a Arduino.
3. Implement the algorithm on a micro controller which is connected with a web cam.

The decision which method will be implemented/tested depends on the time reserve after the points before were implemented.

A. Appendix

If not otherwise noted pictures are taken from *Wikicommons*. http://commons.wikimedia.org

A.1 YCbCr-Color-Space

A.1.1 Color Spaces

A color space is a mathematical model to represent color information out of a picture. There were several color models available:

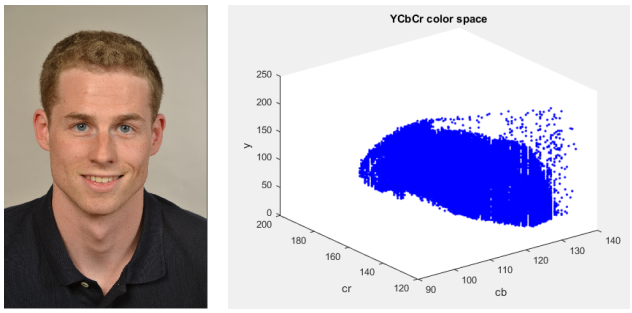
- RGB based color space (RGB, normalized RGB)
- Hue based color space (HSI, HSV and HSL)
- Luminance based color space (YCbCr, YIQ and YUV)
- Perceptually uniform color space (CIEXYZ, CIELAB, and CIELUV)

Luminance based color space has the split the image into intensity (luminance) informations and color (chrominance) informations. The YCbCr space was chosen because several articles recommended this color space for video applications (because of its speed). The letters of YCbCr represents:

- Y: luminance
- Cb: blue-yellow chrominance
- Cr: red-green chrominance

A.1.2 YCbCr skin color

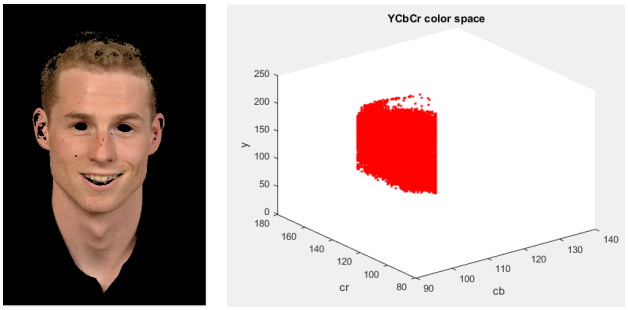
A color picture leads to an YCbCr color space like in figure 4.



(a) Original (b) YCbCr color space

Figure 4: YCbCr - without Thresholding - MEM3 student

By applying the determined thresholds(see section A.2) the skin pixels can be seen clustered in a region of the YCBCR space (see figure 5).

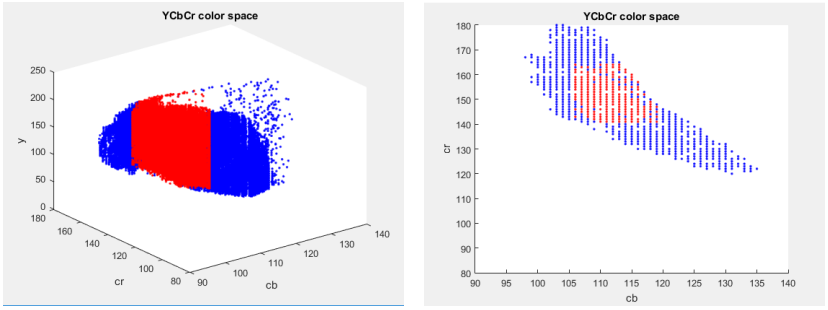


(a) Thresholded picture (b) YCbCr color space

Figure 5: YCbCr - with Thresholding - MEM3 student

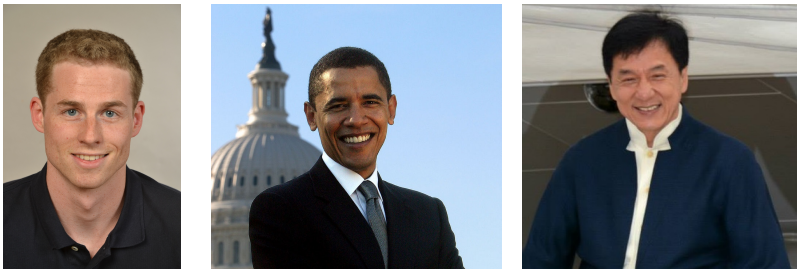
A comparison between the whole coloured picture and the threshold picture (see figure 6) shows that the relevant information of the skin color is in the chrominance (Cb and Cr) and independent of the luminance (Y).

Interesting is that independent of the skin type (white, black or yellow) the relevant skin pixels are always at the same region. To show this the pictures of figure 7 were thresholded and compared see figure yy.



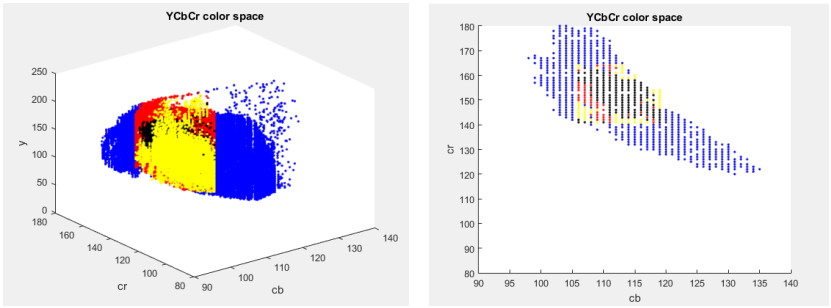
(a) Thresholded picture (b) YCbCr color space

Figure 6: YCbCr - comparison between the full colord picture (blue dots) and the thresholded picture (red dots) - MEM3 student



(a) MEM3 student (b) Barack Obama (c) Jackie Chan

Figure 7: Test pictures



(a) Thresholded picture (b) YCbCr color space

Figure 8: YCbCr - comparison between the full colord picture of MEM3 student (blue dots), thresholded MEM3 student (red dots), thresholded Jackie Chan (yellow dots) and thresholded Barack Obama (black dots).

A.2 Color threshold

A.2.1 Procedure

According to different literature (Real Time Detection and Tracking of Human Face using Skin Color Segmentation and Region Properties Kumar and M (2014), Face Detection Using Color Thresholding, and Eigenimage Template Matching Kumar and M (2014) and A Robust Skin Color Based Face Detection Algorithm Singh et al. (2003)) the thresholds must be found with an try and error procedure.

The three mentioned articles have chosen different thresholds, in this project the thresholds were chosen with the MATLAB application colorThresholder (from the Image Processing Toolbox).

To run this application the command *colorThresholder* must be entered into the command window of MATLAB.

After loading an image and choosing the color space (in this case the YCbCr space - see figure 9) the thresholds can be set (see figure 12).

The next step is to use the find values for Cb and Cr:

- Cb: $105 > \text{Cb} < 120$
- Cr: $140 > \text{Cr} < 165$

to threshold the image (values above the over limit and pixel values under the lower value will be set to black, all pixels within the limit will set to white). This can be done with the matlab commands:

```
% Thresholding -> binary
thresh_cb = cb > 105 & cb < 120;     % thresholding for cb values
thresh_cr = cr > 140 & cr < 165;     % thresholding for cr values
binary_pic = thresh_cb&thresh_cr;    % create binary picture
```

The result looks like figure 11.

The next steps is to modify the binary picture (by removing the small black pixels in the face and the small white pixels out of the face) to make face detection more efficient.

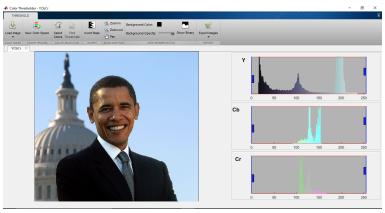


Figure 9: colorThresholder - Barrack Obama - without thresholding

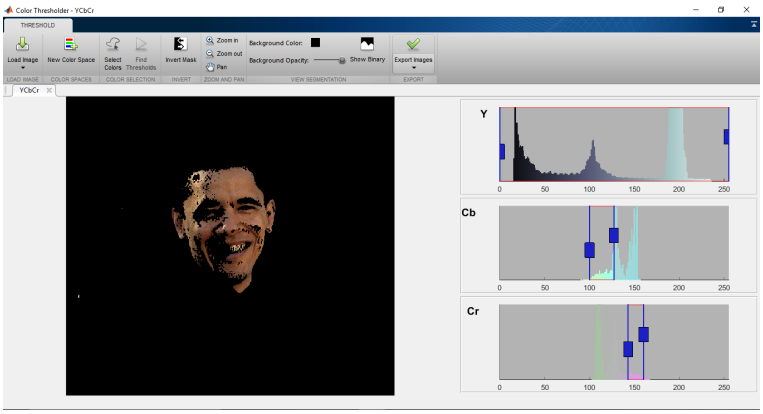


Figure 10: colorThresholder - Barrack Obama - with thresholds : $105 > cb < 120$ and $140 > cr < 165$

Figure 11: colorThresholder - Barrack Obama - with thresholds - Binary

A.2.2 MATLAB implementation

Listing A.1: Color thresholding

```
I=imread('../pictures/obama.jpg');  % load image from workspace

% RGB -> YCbCr
YCBCR = rgb2ycbcr(I);               % transform image into YCbCr space
y = YCBCR(:,:,1);                   % extract Y value out of matrix
cb = YCBCR(:,:,2);                  % extract Cb value out of matrix
cr = YCBCR(:,:,3);                  % extract Cr value out of matrix

% Thresholding -> binary
thresh_cb = cb > 105 & cb < 120;    % thresholding for cb values
thresh_cr = cr > 140 & cr < 165;    % thresholding for cr values
binary_pic = thresh_cb&thresh_cr;   % create binary picture

% show binary picture:
figure
imshow(binary_pic);
```

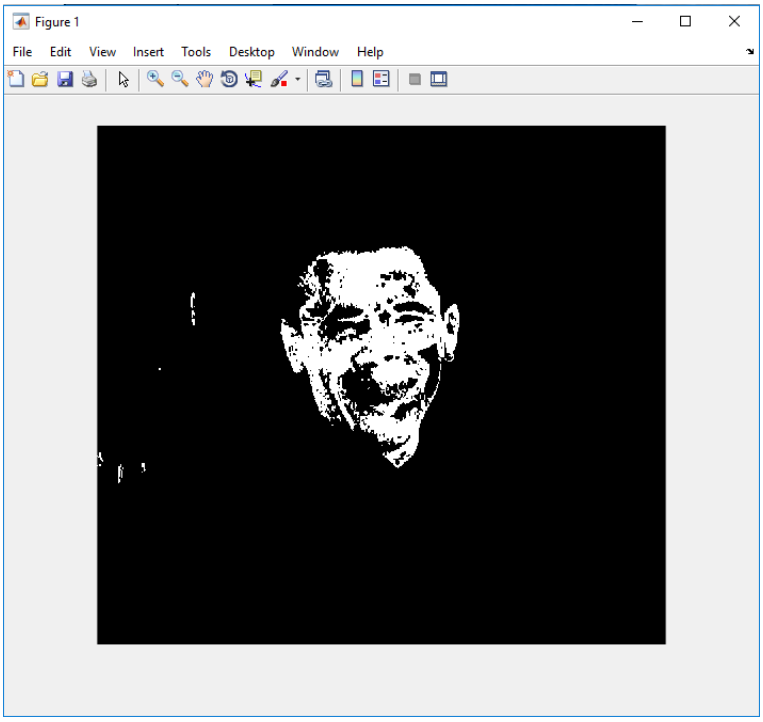


Figure 12: Binary Picture - Barrack Obama - figure out of matlab code A.1

A.2.3 Examples

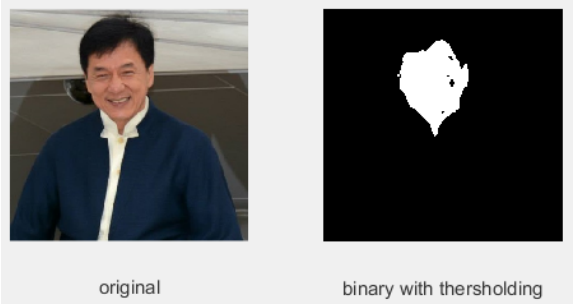


Figure 13: Chackie Jan

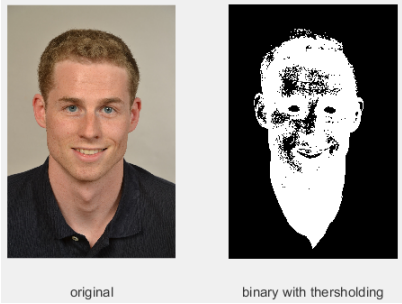


Figure 14: MEM student (private picture)

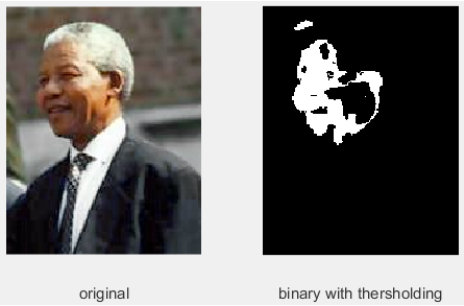


Figure 15: Nelson Mandela

B. MATLAB scripts

B.1 Show pictures in YCbCr space

B.1.1 Main script

```
% University of Applied Science Vorarlberg
% Master of Mechatronics
% ------------------------------------------------------------------------
% Course:       Sensor Systems
% ------------------------------------------------------------------------
% Author:       Tobias Burtscher and Stefan Stark
% Date:         07.12.2016
% Description:  Script which shows picture (original and thresholded)
%               informaion in YCbCr space.

%%
clear all, close all, clc;          % clean up

%%
tLI = tic;                          % start a stopwatch timer
I = imread('pictures/me1_mod.jpg'); % load image
tLoadImage = toc(tLI)               % stop timer to see how much time is
                                    % necessary to load an image

tTI = tic;                          % start a stopwatch timer
YCBCR = rgb2ycbcr(I);               % transfare image into the ycbcr space
y = YCBCR(:,:,1);                   % seperate variable
cb = YCBCR(:,:,2);                  % seperate variable
cr = YCBCR(:,:,3);                  % seperate variable
tTransformImage = toc(tTI)          % stop timer

% Schow image in YCBCR color space
figure
plot3(cb,cr,y,'b.')
hold on;                            % to draw the thresholded values into
```

```
31                                      % the same figure
32
33 %% draw into the same figure the thresholds of different pictures
34 % load image from disk and save rgb values as matrix
35 image1 = imread('pictures/me1_mod.jpg');
36 image2 = imread('pictures/JackieChan.jpg');
37 image3 = imread('pictures/obama.jpg');
38
39 % define colors
40 colo = ['r' 'y' 'k'];
41
42 % define thresholds
43 cb_low = 105; cb_high = 120;
44 cr_low = 140; cr_high = 165;
45
46 % run function which
47 %   calculates cb and cr values,
48 %   make thresholding and
49 %   create YCbCr plot;
50 createYCbCrPlot3(rgb2ycbcr(image1),cb_low,cb_high,cr_low,cr_high,colo(1))
51 createYCbCrPlot3(rgb2ycbcr(image2),cb_low,cb_high,cr_low,cr_high,colo(3))
52 createYCbCrPlot3(rgb2ycbcr(image3),cb_low,cb_high,cr_low,cr_high,colo(2))
53
54
55 hold off;
56 xlim([90 140]);ylim([80 180]);          % scale axis for better view
57 xlabel('cb');ylabel('cr');zlabel('y');  % label axis
58 title('YCbCr color space');             % create title of figure
```

B.1.2 functions

```
1  % University of Applied Science Vorarlberg
2  % Master of Mechatronics
3  % ------------------------------------------------------------------------
4  % Course:       Sensor Systems
5  % ------------------------------------------------------------------------
6  % Author:       Tobias Burtscher and Stefan Stark
7  % Date:         07.12.2016
8  % Description:  Function which seperate Y, Cb and Cr values out of
9  %               transformed picture; threshold the picture and show
10 %               thresholded picture pixels in an existing figure
11
12 function createYCbCrPlot3( YCBCR,cb_low,cb_high,cr_low,cr_high,colo)
13     y = YCBCR(:,:,1);                   % seperate variable
14     cb = YCBCR(:,:,2);                  % seperate variable
15     cr = YCBCR(:,:,3);                  % seperate variable
16     % Thresholding the image
```

```
17      thresh_cb = cb > cb_low & cb < cb_high;
18      thresh_cr = cr > cr_low & cr < cr_high;
19
20      %define color
21      colo = sprintf('%s.',colo);
22      % plot thresholded picture into YCbCr figure
23      plot3(cb.*(uint8(thresh_cb)),cr.*(uint8(thresh_cr)),y,colo);
24
25  end
```