

521466S Machine Vision

Assignment #2

Color spaces

Spring 2016

This assignment explores the HSV color space. It highlights the differences with the RGB color space and how to calculate meaningful distances and means in HSV space. We will implement a skin detector based on simple color comparisons. The detector should mark the pixels that are the same color as a reference skin patch, as in Figure 1.



Figure 1: We wish to extract all regions that are skin-coloured.

Theory

Color spaces

The RGB and HSV color spaces are both very common in computer vision. They serve different purposes. RGB is a linear space and directly represents the colors as recorded by most cameras and as displayed by most screens. HSV on the other hand is a non-linear space and it correlates better to how we humans perceive colors. In other words, compared to RGB, the distances calculated between two colors in HSV space more closely resemble the distance that a human would assign them.

Matlab represents both RGB and HSV colors as a vector with three components, each component ranging from 0 to 1. Thus, a color lies inside a 3D cube of unit size. Figure 2 shows a plot of the color cubes for RGB and HSV in Matlab units.

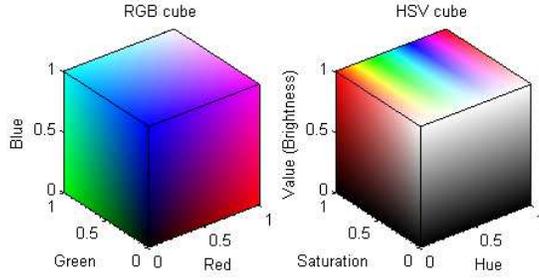


Figure 2: The RGB and HSV color spaces in Matlab units.

Distances

Since RGB is a linear space, the distance between two colors is trivially obtained by the L2-norm:

$$\begin{aligned}
 d_{\text{rgb}} &= |\mathbf{c}_a - \mathbf{c}_b| \\
 &= \sqrt{(r_a - r_b)^2 + (g_a - g_b)^2 + (b_a - b_b)^2}
 \end{aligned} \tag{1}$$

However, the HSV space is non-linear and applying the L2-norm directly in Matlab units leads to an incorrect distance. Instead of a cube, a better representation for the HSV space is a cone, as in Figure 3. If we translate Matlab's HSV units to a point inside the cone, this transformed space is linear and we can apply normal operators to the transformed color values (e.g. L2-norm, mean).

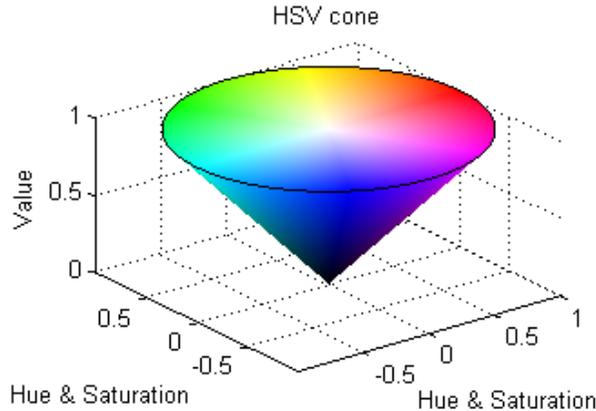


Figure 3: The HSV cone, L2-norm between points in the cone is a good distance.

To transform the HSV color to this linear space the hue indicates the angle around the vertical axis, the saturation indicates the distance from the axis, and the value the distance along the axis. The equations to transform the HSV coordinates (h,s,v) to the linear space (a,b,c) are:

$$a = sv \cos(2\pi h), \quad b = sv \sin(2\pi h), \quad \text{and} \quad c = v \tag{2}$$

Once the color has been transformed to this linear space, the L2-norm gives a good distance between points. Conversely, the equations to transform the point back to HSV

space are:

$$h = \frac{\text{atan2}(b, a)}{2\pi}, \quad s = \frac{\sqrt{a^2 + b^2}}{c}, \quad \text{and} \quad v = c \quad (3)$$

Note that the hue h is an angular quantity and is therefore periodic. To make sure that the resulting value is on the desired range $[0 \dots 1]$ we can use $h = \text{mod}(h, 1)$.

Skin detection

Our skin detector will be very simple. It will be trained to learn the mean color of the skin and what is the maximum distance that a pixel can be from this to be classified as skin. For each pixel, it will calculate the distance between the learned mean and the pixel's color, and then compare this distance to the learned threshold.

Instructions

1. Download the code and images

This assignment comes matlab functions, a matlab data file, and a testing image. They can be downloaded from:

http://www.ee.oulu.fi/research/imag/courses/Machine_Vision/assignments/

The assignment also has a main script that should guide you through the steps: `main.m`. It is useful to use `Ctrl+Enter` in the editor window to execute only a single section of the main script. The numbering of the instructions here corresponds to the numbering in the main script. You don't have to modify the main script.

2. Visualize the color spaces

Run the function `visualizeColorSpaces()` to view the color spaces in Matlab. Rotate the cubes and get a feeling for what the dimensions mean. Notice the non-linearities of the HSV space: (Q1) how does the hue affect the color when saturation is zero? (Q2) what is the difference between a color with hue 1 and hue 0?

3. Load the training data

Execute step 3 from the main script. This loads an image and extracts a region of pixels to use as training data. The image and region are displayed on screen. The training colors are saved in two variables: `colorsRgb` and `colorsHsv`. They are the same colors but in different color spaces.

Note that you can modify the code in step 3 to select the region yourself.

4. Plot the training colors

Plot the training colors in both color spaces (`colorsRgb` and `colorsHsv`). You should obtain a plot similar to the one in Figure 4.

Observe the selected region from step 3. Notice that all the pixels in this region are "skin pixels" yet the actual colors can be very different. The pixels in RGB space cover a large volume and are hard to model by a single representative color. (Q3) Why are the pixels in HSV space split into two different groups?

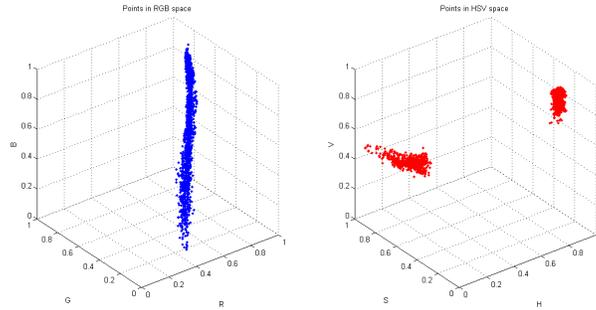


Figure 4: The training colors plotted on the two color spaces.

5. Transforming HSV to a linear space

We will now transform the points from Matlab's HSV units to a linear space (i.e. the cone of Figure 3). Modify the provided function `hsvToLinearSpace()` to implement Equation (2). Once you have modified the function, execute step 5 to visualize the transformed space and the transformed points. You should get a plot similar to Figure 5.

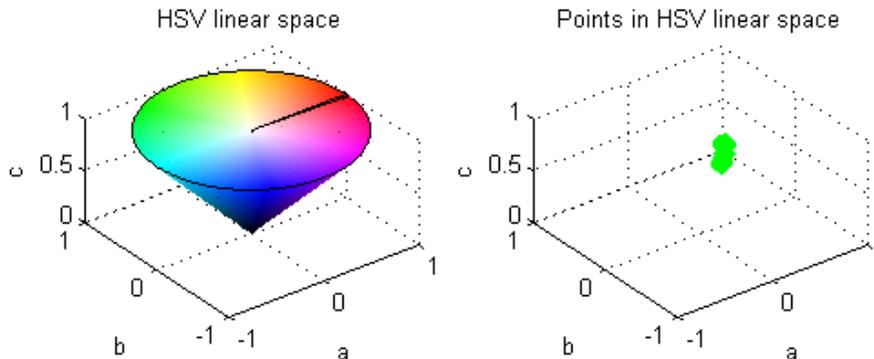


Figure 5: Distances in HSV space.

6. Calculating distances in HSV space

The provided function `hsvDistance(a,b)` should calculate a meaningful distance between two HSV colors. The provided code directly computes the L2-norm on Matlab's HSV space. Modify the function to apply the L2-norm on the transformed linear space (hint: use the `hsvToLinearSpace()` function). Once you have modified the function, execute step 6 to visualize the difference between applying L2-norm in HSV space and applying it in the linear space. You should get a plot similar to Figure 6

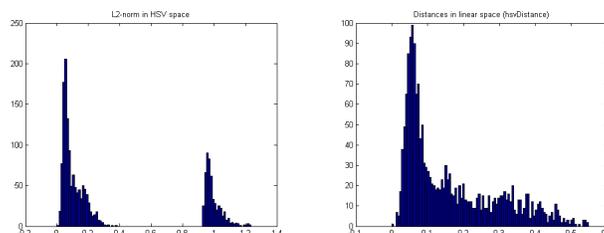


Figure 6: Distances in HSV space.

Notice how applying L2-norm on the original HSV space creates artificially big distances for points that cross the hue discontinuity. If `hsvDistance` is implemented correctly, the distances should be continuous and tightly packed, as in Figure 6.

7. Calculate a mean in HSV space

We will now calculate a mean color from the set of training colors. This mean color will be the average color of a skin pixel.

The provided function `calculateHsvMean()` should calculate a meaningful mean for a group of colors in HSV space. The provided code directly computes the mean on Matlab's HSV space. This produces wrong results due to the hue discontinuity. Modify the function to obtain the correct mean color. *Hint:* Transform all colors to linear space using Eq. (2), calculate the mean of the transformed colors, then convert the mean back to HSV space using Eq. (3).

Once you have modified the function, execute step 7 of the script and analyse the results. Notice how the mean in HSV space produces a value very far away from the cluster of points. Meanwhile, the mean in linear space takes into account the discontinuity in the hue channel and produces a mean that represents the cluster well. Notice how the distances of all points to the mean in linear space are tightly packed and close to zero, as in Figure 7.

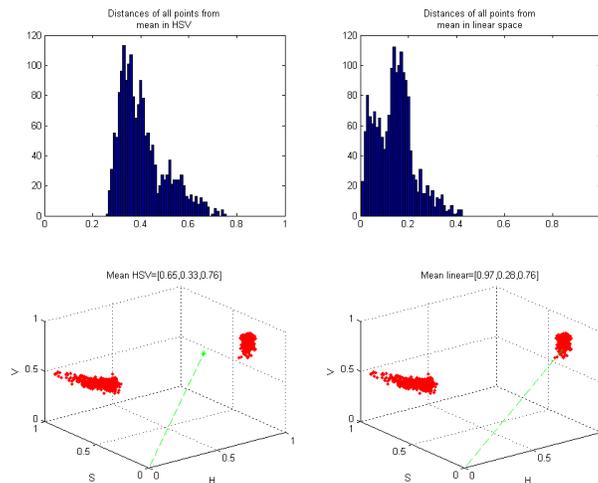


Figure 7: Distances in HSV space.

8. Select a threshold on the color distance

We are now ready to calculate a threshold distance for skin detection. The skin detector will compare a color with the previously selected mean skin color. If the distance between them is lower than the selected threshold, it will classify the pixel as skin. We will compare the performance of detection in RGB space and HSV space (using the correct distance previously developed).

The provided function `findDistanceThreshold()` selects a threshold that classifies 95% of the training data as skin. Execute step 8 of the script to select a threshold in both RGB and HSV space.

You will see histogram of the distances of the training colors to the mean, and a threshold for each color space. Notice how the HSV threshold (0.296) is much lower

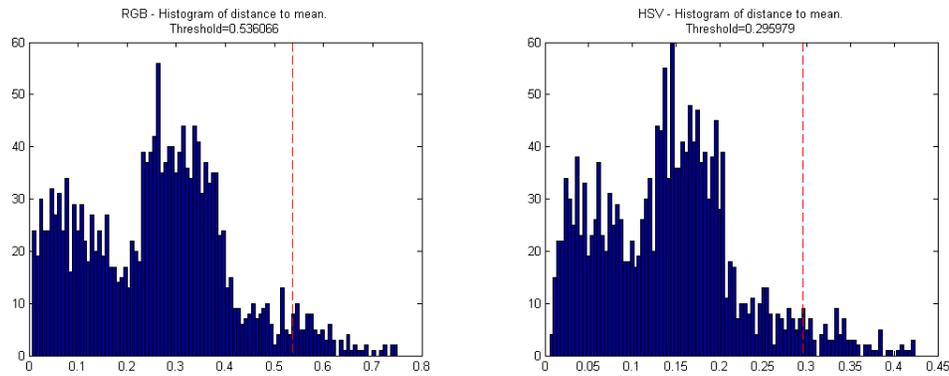


Figure 8: The obtained distance thresholds. Note that the HSV threshold is significantly lower and the distribution is more compact.

than the RGB threshold (0.536). The distances in the HSV space are also smaller, meaning that the model fits better to the training data.

9. Visualize your results

Execute step 9 of the script. This will detect the skin pixels in the training image using both RGB and HSV color spaces. The HSV segmentation should look like one in Figure 1. Notice how RGB misclassifies many pixels.

Deliverables

Send an email to janne.mustaniemi@ee.oulu.fi with the subject line MV-A2-[student number] (e.g. MV-A2-1234567) and a zip file of the same name with the following files:

1. `Answers.txt` with the answers to questions Q1, Q2, and Q3. (0.25 points)
2. `hsvToLinearSpace.m` (0.75 points)
3. `hsvDistance.m` (0.25 points)
4. `calculateHsvMean.m` (0.75 points)

Deadline of the assignment **2.2.2016**