Lecture Machine Vision 2019/20

Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie
Dr. Martin Lauer, Christian Kinzig, M.Sc.

Practical Exercises: Color and Segmentation

MATLAB provides several functions to work with different color spaces. When we read a color image with the command `imread`, e.g. `I = imread ('flower.png')`, then `I` contains a 3-dimensional tensor[1] which represents the pixture in RGB color space. The values for R, G and B range between 0 and 255 (8 bit unsigned integers). You can access the 3d-tensor `I` analogously to a matrix, i.e. `I(v,u,1)` yields the R-value at position row v and column u, `I(v,u,2)` yields the G-value at the respective position and `I(v,u,3)` the B-value. You can plot a RGB image with the `image` command. For many purposes, it is more convenient to represent RGB values in the range between 0 and 1. You can transform an image into this form by `Irgb = im2double(I)`. To ensure consistency of calculations, we recommend to transform RGB values always into the range between 0 and 1.

Images can be transformed between different color spaces using the following commands:

command	explanation	example
`rgb2hsv`	calculate HSV image from RGB image	`Ihsv = rgb2hsv(Irgb)`
`hsv2rgb`	calculate RGB image from HSV image	`Irgb = hsv2rgb(Ihsv)`
`rgb2lab`	calculate L*a*b* from RGB[2]	`Ilab = rgb2lab(Irgb)`
`lab2rgb`	calculate RGB from L*a*b*[3]	`Irgb = lab2rgb(Ilab)`

MATLAB uses values between 0 and 1 to represent all three channels of HSV. The luminance channel of L*a*b* is represented by numbers between 0 and 100 while the chrominance channel is represented by values which range approximately between -100 and +100. However, not all combinations in L*a*b* yield valid colors.

- Consider the image file *stack.png*. Load the file into MATLAB and transform it into HSV and L*a*b* color space. Display the three channels of the HSV color space and the luminance channel of the L*a*b* space. Compare the result of the value channel of HSV and the luminance channel of L*a*b*.

  Level of difficulty: easy

- The file *ccl.m* contains an implementation of the connected components labeling algorithm in MATLAB. Apply the CCL algorithm to the image *stack.png*. Vary the color space by using RGB, L*a*b*, just the luminance channel of L*a*b*, or

  Level of difficulty: easy/ medium

[1] a tensor is the extension of the idea of a matrix to more than two dimensions, i.e. a 3d-tensor is a 3-dimensional table of numbers, a matrix is a 2d-tensor and a vector a 1d-tensor

[2] in older MATLAB versions this command is not available. Use `Ilab = applycform(Irgb,makecform('srgb2lab'))` instead.

[3] in older MATLAB versions this command is not available. Use `Ilab = applycform(Ilab,makecform('lab2srgb'))` instead.

just the two chrominance channels of L*a*b*. Vary the threshold to achieve good results. Compare the results qualitatively.

- The file *color_kmeans.m* contains an implementation of k-means clustering for color segmentation. Apply it to the image *stack.png*. Vary the color space by using RGB, L*a*b*, just the luminance channel of L*a*b*, or just the two chrominance channels of L*a*b*. Always use $k = 4$. Compare the results qualitatively.

  Level of difficulty: easy/ medium

- File *segments.png* contains the result of a segmentation. Apply an appropriate morphological operator on it to fill the holes in the segments. MATLAB provides the following operations:

  Level of difficulty: easy/ medium

command	explanation	example
`imerode`	erosion operator	`L1 = imerode (L, S)`
`imdilate`	dilation operator	`L1 = imdilate (L, S)`
`imclose`	closing operator	`L1 = imclose (L, S)`
`imopen`	opening operator	`L1 = imopen (L, S)`

  In all cases, `S` is a binary matrix (a matrix with entries 0 and 1) that specifies the neighborhood to use. Try the following matrices:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

  to model the standard 8-neighborhood and 4-neighborhood, resp.

  Furthermore, try some non-standard neighborhood relationship together with the morphological operations and observe their effect, e.g.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- For quick workers: The files *flower.png*, *boat.png*, and *number.png* contain some more pictures that have been used on the lecture slides. Try to calculate good segmentations with the k-means-algorithm and with CCL.