SGN-12006 Basic Course in Image and Video Processing

EXERCISE 11

20.11.2015-02.12.2015

This exercise consists of both lab exercises only. Complete the lab exercises and present your results for the TA.

Topics of the exercise: reading of y4m-video file, filestream, using header information and videoplayback from the file. YUV to RGB transformation in 4:2:0 video format.

1) YUV to RGB transformation (3 points)

a) Load the file yuvdata.mat to Matlab.
b) Check the size of the variables
c) Compute the sive of image (data (rows*cols) and compare it to the variable sizes
d) Reshape the yy component to an image (reshape)
e) Reshape the uu and vv components to images. Note that the data is in 4:2:0 format, where chromatic components are having half the size against Y component (cols/2, rows/2)
f) Visualize the components in the figure (figure, imshow, subplot)
g) Scale the U and V component images to the same size as Y (imresize) and move the intensity of the image around the zero subtracting 127 from both scaled images
h) Combine the components as one 230400x3 matrix for the RGB transformation: YUV = cat(2,Y(:),U(:),V(:))
i) Multiply the UYV matrix with the RGB transformation matrix: RGB = yuvToRgb * YUV';
j) Reshape every image vector separately back to the original sized images (reshape) and combine the RGB components as one RGB image (cat)
k) Transform the image to uint8 format and see the results (imshow)

```
YuvToRgb = [1   0         1.402   ;
            1  -0.34413  -0.71414;
            1   1.772     0      ];
```

2) YUV to RGB transformation (3 points)

This exercise combines the knowledge from exercise 11 for the y4m-video file reading.
Open the example code and familiarize yourself with the code. Get your solutions from exercise 11 ready for this exercise. You will need it, but if you don't have it ask it from the assistant.
Phases of the work (Phases written in *italics* are pre-written in the example code):

a) Define the path of the video file.
b) Read the header from the videofile
c) Process the header string with the codes from exercise 11. **(0.5 point)**
d) Define the parameters according to the header (cols, rows) **(0.5 point)**
e) Compute the length of the image data according to the header information and generate the indexes for the first 50 frames. Read the data of the first frame from the file and process the separated Y, U and V components.
f) Process the RGB frame from the YUV data with the codes of Task I and visalize the frame. **(1 point)**
g) Create a function based on earlier phases: **(2 points)**

[rgbframe]=readframe(fid,location, length), which reads and processes one frame from the file stream pointed by fid from the defined location and data length. The function returns uint8 RGB frame ready for visualization.

h) Similarly as in Exercise 10, create a video in the Figure window with a for -loop using the readframe-function. Now the frame is aquired from the readframe-function instead of cell table of Exercise 10. View the first 50 frames of the videofile using the computed frame locations and the length of image data. **(1 point)**

You can also test some other videos from [1].

Image nr 607/1253 from film sintel_trailer_2k_480p24.y4m YUV4MPEG2 C420jpeg W854 H480 F24:1 Ip A1:1

3) Computations (2 points)

Compute how much memory you would need if you load all 420 coded videoframes of sintel_trailer_2k_480p24.y4m (above) once in the memory instead of streaming the data frame by frame from the file? **(1 point)**

In the task II the maximum number of frames was defined by hand to 50. Explain how you could play the whole video automatically without defining the maximum number of frames beforehand. No matlab code is needed, only an idea how the code should be designed. **(1 point)**

Materials

[1] http://media.xiph.org/video/derf/