SGN-12006 Basic Course in Image and Video Processing

EXERCISE 11

23.11.2015-25.11.2015

This exercise consists of both lab exercises and homework. Complete the lab exercises and present your results for the TA. Prerequisite for submitting the homework is attendance in an exercise session. Homework should be submitted only online using Moodle2.

Follow the naming format 'ExN_surname_ID.pdf' (N is the number of exercise). Also please clearly write down your full name and student number in the document. The homework report should be no more than 1 page long and it should be done individually (no pairs allowed). Questions on this exercise should be addressed to TA's email address: (firstname.surname@tut.fi).

Topics of the exercise: Video formats, reading images from the folder, creation of video in for loop, filestream in Matlab, header of the video file and how to process it.
The exercise 12 continues the same topic as this exercise, so it is suggested to comment the tasks well.
Commands for the exercises: dir, strcmp, cell, struct, length, size, disp, ops, for, if, imread, imshow, set, load, struct, cat, reshape, subplot, imresize, strfind, if, switch - case

Lab exercises

1) Visualization of the video frames (3 points)

Create a "visualize" function according to phases 1.-4. The function call is following:

[frames, N, frameformat] = visualize(framepath)

Phases:
1. Load (load) formats.mat -file to working memory. It contains the definitions of for different video resolutions (see also [1]). You can search the contents of the variables with ex11.m script.
2. Load images from the folder "frames" to cell variable: read the filenames inside the folder with dir command and create a for -loop where the images are loaded one by one to the cell table.
3. Visualize the read images with two following ways:
   - Figure 1: All images together in one subplot. Define the amount of rows and columns in subplot according to the amount of found images: for example 12 images can be visualized in 3 rows and 4 columns or 2 rows and 6 columns.
   - Figure 2: By updating the new image over the old by using the example code at right. How the "video" looks like?

```
posvec=[50 150 1024 768];
figure('Color',[0 0 0],'Position',posvec)
rgbframe=frames{1};
imagehandle=image(rgbframe,'EraseMode','None');
axis equal; axis off;
for index=2 : max(size(frames))
   rgbframe=frames{index}; % cell-type variable
   set(imagehandle,'Cdata',rgbframe); drawnow;
end
```

4. Combine the earlier codes to one function:
   - the function loads formats.mat file
   - the function loads images to frames cell variable from the path defined in framepath variable and saves the number of images in N.
   - the function finds the format of the video by checking the size of a single frame and comparing the size to information in formats variable. The format type is saved to frameformat string. The function assumes that all frames are from same video and thus having the same size.
   - the function opens the first Figure -window and draws all frames there with subplot.
   - the function opens the second Figure -window and plays the frames as a video with a for-loop.
   - the function has a help which tells what the function does, how it is used and who has created it.

2) Introduction to header (2 points)

a) Load the mat-files into Matlab in memory.
b) Present the contents of videofileinfo variable using for -loop and disp -command (See the 3)-c).
c) Shape the print out to the screen as following (index, header, sequence name):
   '01 ... YUV4MPEG2 W176 H144 F30000:1001 Ip A128:117 ... akiyo_qcif.y4m'

3) Processing of the header string (3 points)

1. Create a struct named "parameters", which has following fields for the header parameters:

   W = width,

   H = Height,

   F = Framerate,

   I = interlacing,

   A = pixel aspect ratio,

   C = chroma subsampling ratio (in these exercises always 4:2:0 or 4:2:0jpeg)
2. Create a for -loop where all headers in videofileinfo are processed and variables are saved to paramters struct (one instance for each header). See the example below:

```
>> header =
YUV4MPEG2 W176 H144 F30000:1001 Ip A128:117
>> parameters(1) =
W: 176
H: 144
F: '30000:1001'
I: 'p'
A: '128:117'
C: '420' ← This is a default value which is not
defined in this header, but still initialized.
```

The order of the information in the header might change so the data must be identified according the letter preceding the parameter. Every field begins with letter and ends with space (or end of the header). In the other words you need to find the letter and the space after the letter and save the content between them to the parameters struct. In the case of W and H the string must be translated to number, but in all other cases (F, I, A, C) the data is saved as a string. If C parameter is not defined in the header it is defined to '420' (as a string). Use the find data.m script as an example for the processing and remember that string can be indexed as vector in Matlab.

3. Test your struct with following code:

```
for n=1:length(parameters)
       disp(['W: ' num2str(parameters(n).W) ' H: ' num2str(parameters(n).H) ' F: ' parameters(n).F '...
           I: ' parameters(n).I ' A: ' parameters(n).A ' C: ' parameters(n).C])
end
```

Homework (2 points)

Explain how colours are defined in Y'CrCb color model and what is the meaning of following chroma sub-sampling patterns: a) 4:4:4, b) 4:2:2, and c) 4:2:0.

Calculate the number of bits per frame required to encode a Full HD (1920 × 1080 pixels per frame) video with these patterns.

Materials

[1] http://en.wikipedia.org/wiki/Display_resolution
[2] http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch6/ch6_color_models.html
[3] http://wiki.multimedia.cx/index.php?title=YUV4MPEG2