

EE433 Real-Time Applications of Digital Signal Processing

Project

This project is about the implementation of an image processing task in C and call this function from LabVIEW to display it in the user menu. Two functions will be written. The first function is to convert RGB image to grayscale and the other is to convert grayscale image to RGB. Note that the second function requires a color map where grayscale is mapped to RGB color. You can use jet colormap for this purpose. Also you should note that grayscale to RGB conversion is non-unique since the information regarding to color is lost during RGB to grayscale conversion. RGB to grayscale conversion is realized by the following relation

$$G = 0.299R + 0.587G + 0.114B$$

where R,G,B, are the red, green and blue color values and G is the grayscale value. When you convert from grayscale to RGB, you can divide grayscale values into regions and map these regions to the corresponding color fields.

Your image processing functions will be implemented in C and called from LabVIEW as a shared library function "image-func.dll". Note that this implementation is done on a PC which usually has Windows OS. Hence the shared library is "*.dll". If the OS is Linux then the shared library should be compiled as "*.so". myRIO is Linux based and if you would implement it in myRIO, your shared library would be "image_func.so". Assuming that you would be implementing this project in a PC, you can create your "image-func.dll" by using a development environment called as LabWindows. LabWindows is an ANSI C programming environment developed by NI. You will install LabWindows to your PC and write your C functions to compile them as shared library "image-func.dll". Then in LabVIEW, you will acquire an image (Ex. image.jpg) and feed it to the shared library by appropriately adjusting the input and output parameters as described in [1],[2]. Once the image is processed, the result will be received from the shared library block and displayed in the user interface window.

The user interface should include displays of the input and output images. It can include the user parameters such as RGB to grayscale and grayscale to RGB conversion selection. You should also display the time it takes for the conversion process. This should be compared with the LabVIEW function for RGB to grayscale conversion [3]. The time difference should also be displayed in the user interface. In addition, you should be able to change the color map used in the grayscale to RGB conversion in user interface. As an example, this can be done by implementing a slider bar. The image size can be fixed or you can make it as another variable. The input image can be read from a file or it can be read from the computer camera. If you can process the video input and output in real-time, that is counted as a positive item, which is then favorably graded.

The project report print-out together with the codes should be zipped and sent to etuncer@metu.edu.tr before the demo session which will be announced later. The project report should include the theoretical background which should be in accordance with the implemented code. It should also include all the results, plots, etc. The report should also be brought to the demo session.

Note that a genuine effort is expected from you so that you should have all the knowledge about your implemented code which will be tested in the demo session.

- [1] <https://forums.ni.com/t5/LabVIEW/Procesing-image-in-Labview-using-a-dll-in-c/m-p/3946407?profile.language=en>
- [2] <https://forums.ni.com/t5/Machine-Vision/Use-Imaq-in-C-dll-to-be-called-in-LabVIEW/td-p/4006579?profile.language=en>
- [3] <https://forums.ni.com/t5/Example-Code/IMAQ-Color-Image-to-Grayscale/ta-p/3535022?profile.language=en>
- [4] <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YGvGCAW&l=en-TR>
- [5] <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YHR7CAO&l=en-TR>
- [6] <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YHR7CAO&l=en-TR>
- [7] https://documentation.help/NI-Vision-LabView/IMAQ_GetImagePixelPtr_Example.html