IB Paper 8: Photo Editing

Lecture 1: Program Framework and Cropping

N G Kingsbury (given by J Lasenby in 2016)

Signal Processing Group,
Engineering Department,
Cambridge, UK

Easter 2016
Contents

• Lecture 1: Sections 1 & 2: Program framework and Cropping.
• Lecture 2: Sections 3 & 4: Resizing and Rotating.
• Lecture 3: Sections 5 & 6: Morphing and Colour.
• Lecture 4: Sections 7 & 8: Histograms, Lighting and Filtering.
• Lecture 5: Section 9: Enhancing.
• Lecture 6: Examples Class
Second and Third Sections of the Course

B – Image Features and Matching Course

Roberto Cipolla

C – Image Searching and Modelling using Machine Learning Methods

Roberto Cipolla
Recommended Textbooks


Matlab code for the Photo Editor is downloadable from NGK’s website:

www-sigproc.eng.cam.ac.uk/~ngk

in the section *Downloadable teaching material.*
Introduction

• Aim: to develop a photo-editor in Matlab

• First task: produce an image display framework and graphical user interface (GUI) to allow the functions to be called and controlled, and to show results.

• Choose Matlab because: allows quite complicated image processing operations to be implemented with relatively small amounts of code, and it has the basic functions for providing a suitable GUI for the user to interact with the program.
The Main Script File:  \texttt{ph\_edit}

- The code for the main script file \texttt{ph\_edit.m} is shown in fig. 1.2. in the notes.
- No variables are passed as arguments when script files are called.
- For GUIs with controls and buttons, it is simplest if calls are created to script files.
- Now look at the code to see how the GUI is set up.
Opening an input file: \texttt{ph\_openfile}

- \texttt{ph\_openfile} is a script called by the menu item \textbf{Open} ‘\textbf{Before}’ (and also at the end of \texttt{ph\_edit} – although there are a few problems with this).

- \texttt{uigetfile} is used to open a GUI window to allow selection of an input image file (.tif, or .jpg, but could be others). Directory path and filename then concatenated into \texttt{infile}.

- \texttt{imread} reads in image into \texttt{xui}. If no output image \texttt{yui}, \texttt{xui} is read into \texttt{yui}.

- \texttt{xui} and \texttt{yui} are displayed as \textbf{Before} and \textbf{After} via \texttt{showimages}. 
Saving an output file:  \texttt{ph\_savefile}

- Script for saving files is given in notes: it is called by the menu item \texttt{Save \textquotesingle{}After\textquotesingle{}} and also when \texttt{Close Editor} is activated.

- Code creates an \texttt{outfile} by adding an \texttt{.a} to the input filename (if the ending is not already \texttt{.a})

- \texttt{uiputfile} then called to allow user to confirm, change or cancel.

- \texttt{imwrite} then saves the \texttt{After} image \texttt{yui} as \texttt{.tif} or \texttt{.jpg}.
Displaying the images: \texttt{showimage}

- Fig 1.5 in the notes shows the code for displaying images and their colour histograms.

- \texttt{newbefore} is a variable which ensures (if it is 0) that the \texttt{Before} image is not updated – this speeds things up.

- \texttt{subplot} and \texttt{image} Matlab functions are used to plot the RGB images.

- Matlab functions \texttt{reshape}, \texttt{barcolour}, \texttt{hist} are used to plot the histograms.

- Zoom is enabled.
Other simple operations within:  \texttt{ph\_edit}

Look at the top 5 menu options:

- **Open ‘Before’:** via a call to \texttt{ph\_openfile}
- **Reopen ‘Before’:** re-reads from input file via: \texttt{xui = imread(infile); showimages}
- **Save ‘After’:** via a call to \texttt{ph\_savefile}
- **Copy ‘Before’ to ‘After’:** via \texttt{yui=xui; newbefore=0; showimages}
- **Copy ‘After’ to ‘Before’:** via \texttt{xui=yui; newbefore=1; showimages}
**Cropping the image:**  \texttt{ph\_crop}

\texttt{ph\_crop} is the first of the 9 script files that perform the main operations within the Photo Editor. Code is given in Fig 2.1.

- Scripts are all controlled by \texttt{mode}: a string variable in the main workspace.
- All options are in the same script file and are selected by \texttt{mode} using the \texttt{switch} syntax in Matlab.
- \textbf{Note:} \texttt{switch} executes only the first matching case, there is no need for \texttt{break} statements.
- The crop area can be selected via entering coordinates of a rectangle or via the cursor (using Matlab \texttt{zoom}).
Summary

• **Section 1** of the notes outlines how the Photo Editor works – ie making use of the GUI facilities in Matlab and adopting a script approach with the `switch` and `mode` functionalities.

• **Section 2**: The first of the 9 main script files, `ph_crop`, was examined, both for its use of switch/mode and for its intrinsic working.

J. Lasenby (Easter 2016)