

An image processing Demo

Theodoros Giannakopoulos ^{1,2}

¹*Department of Informatics and Telecommunications
University of Athens, Greece*

²*Computational Intelligence Laboratory (CIL)
Institute of Informatics and Telecommunications (IIT)
NCSR DEMOKRITOS, Greece*

email: *tyiannak@gmail.com*
web: *www.di.uoa.gr/~tyiannak*

February 4, 2011

1 Introduction

The presented Matlab code demonstrates a set of basic image processing functionalities. Furthermore, it provides a simple Graphical User Interface, also implemented in Matlab, using the GUIDE tool. **For more details and possible questions, please refer to my Mathworks File Exchange web site or my personal web site.**

2 Demo functionalities

This demo provides some very basic image processing functionalities. In Figure 1, a screenshot of the implemented GUI is shown. In general, the GUI plots the original (loaded) image, as long as a secondary (processed)

image. In addition, on the left of the GUI, the RGB histograms of both images are plotted.

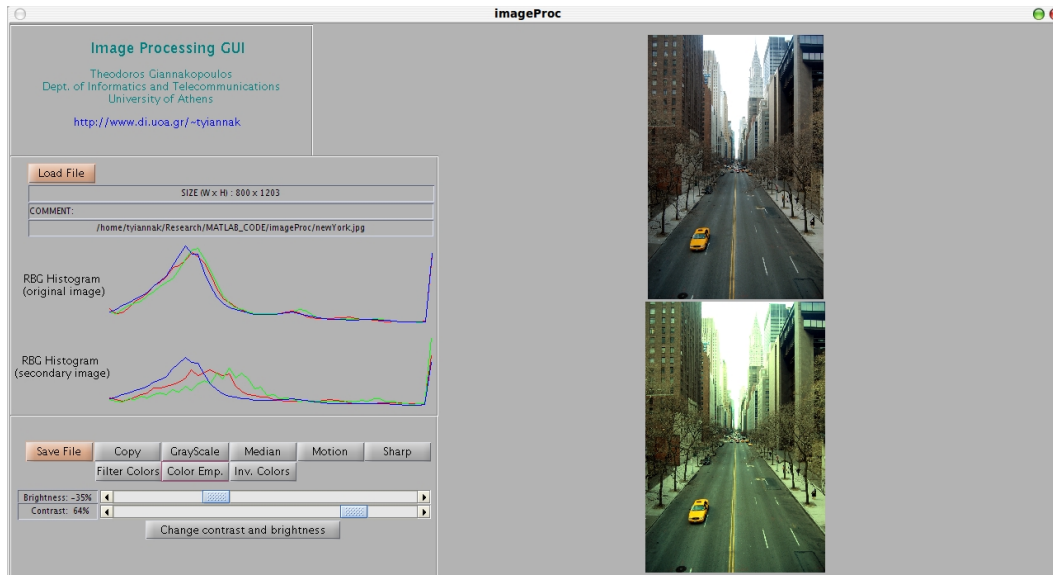


Figure 1: Screenshot of the provided Graphical User Interface

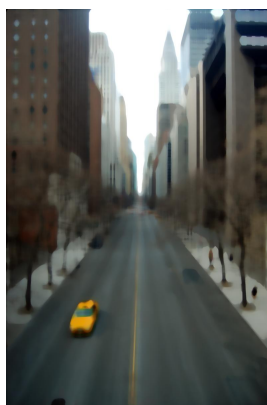
In particular, the following functionalities are provided via the GUI buttons:

- **Load File:** Use this button to load an image file. In the current version, only JPEG, GIF, TIFF and BMP image formats are supported.
- **Save File:** Use this button to save the current secondary image.
- **Copy:** This button copies the original image in the secondary one.
- **Grayscale:** This button generates the grayscale version of the original (loaded) image.
- **Median:** This button applies a median filtering in the original image. Towards this end, the Matlab build-in function `medfilt2` is called. Also, the user is asked to provide the filter mask size, not in pixels, but in percentage of the original image's dimensions (e.g. 2% of the width by 3% of the height). In Figure 2(b) an example of median filtering is presented (filter size was selected to be equal to 3% of the image's dimensions).

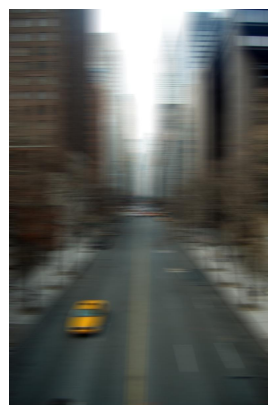
- **Motion:** This functionality applies a filter that approximates the linear motion of the camera. For generating the appropriate filter mask, the function `fspecial` is called, with the "motion" property value. This time, the user has to provide the motion direction (in degrees, $[0..360]$), along with the motion length (not in pixels, but in percentage of the image's larger dimension). For applying the generated mask, the `imfilter` function is used. In Figures 2(c) and 2(d) we present the result of a motion filtering procedure for 0 and 90 degrees motion direction respectively.
- **Sharp:** This button executes a sharpening of the original image. In order to generate the sharpening mask, the `fspecial` function has been used. Furthermore, the `imfilter` function is used for applying the adopted image filter. An example of this procedure is given in Figure 2(e).
- **Filter Colors:** This functionality creates a gray-scaled version of the original image with colored areas. When the respective button is pressed, the user is asked to provide 3 thresholds (in the range $[0..255]$) for each one of the R, G and B coefficients. The larger the thresholds are, the "looser" the color thresholding becomes (i.e., in general, larger thresholds lead to more colored areas in the image). Then, the user is prompt with a copy of the original image and is asked to select 5 points on the image using the mouse. Then, the average color of these colors is computed and a simple thresholding step (using the thresholds provided in the beginning of the procedure) is applied on the image. In Figure 2(f) we have selected 5 points from the yellow color (on the cab), while the color thresholds were selected to be equal to 130, 130 and 30 for the R, G and B coefficients respectively.
- **Color Emphasis:** This functionality lets the user emphasize particular coefficients of the RGB space. In Figure 2(g), we have emphasized the R coefficient by 150%.
- **Invert Colors:** This button is used for inverting the colors of the image. An example of this functionality is given in Figure 2(h).
- **Change Contrast and brightness:** Use this button after having set the contrast and brightness factors (using the provided sliders). An example of brightness and contrast adjustment is shown in Figure 2(i).



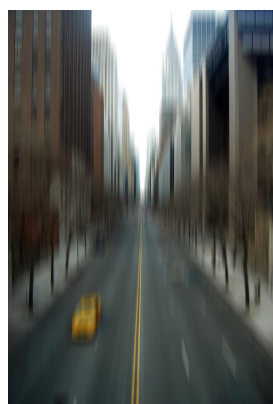
(a) Original image



(b) 3x3 Median



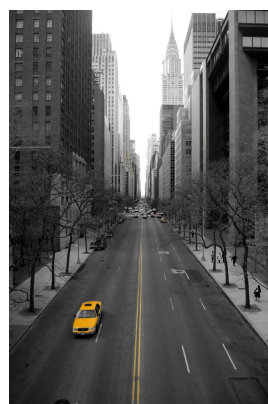
(c) Motion 0 deg



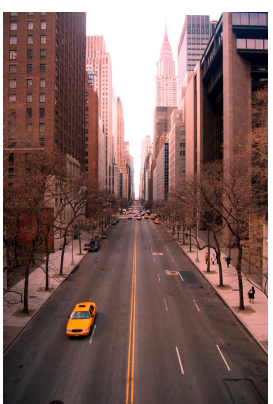
(d) Motion 90 deg



(e) Sharp



(f) Color Filtering



(g) Color Emphasis



(h) Invert Colors



(i) Contrast - Brightness

Figure 2: Some examples of a processed image using the provided GUI