Integrated Image Processing Functions using MATLAB GUI Article Info

Preprint · March 2013

DOI: 10.13140/RG.2.2.10073.16486

2 authors:

Nassir H. Salman
University of Baghdad

Gullanar M. Hadi
Salahaddin University - Erbil

Journal of Advanced Computer Science and Technology Research, Vol.3 No.1, March 2013, 31-38

Integrated Image Processing Functions using MATLAB GUI

Nassir H. Salman[a], Gullanar M. Hadi[b],
Faculty of Computer science, Cihan university,Erbil, Iraq
Faculty of Engineering-Software Engineering, Salaheldeen Univ., Erbil, Iraq
nahu64@yahoo.com, gullanarm@yahoo.com

Article Info

Received: 29th April 2012
Accepted: 23rd September 2012
Published online: 1st March 2013

ISSN: 2231-8852

ABSTRACT

Graphic User Interface (GUI) was designed to integrate many functions in image processing (IP) field to callback such that it can perform actions of IP functions such as image segmentation, skin detection, levelset technique, object extraction, separating color image components and combining gray images to get color one, adding different noises and preprocessing operations such as different filtering to remove these noises, and thresholding. All these functions are built and tested using MATLAB 7.6.0(R2008a) environment. This paper presents a GUI development in Matlab for image processing.

Keywords: **MATLAB GUI, image segmentation, skin detection, levelset , object extraction, color image, image processing.**

1. Introduction

A graphic user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes , sliders, menus, and so forth. There are three principle elements required to create a MATLAB graphic user interface: (1)-Components: like pushbuttons, labels, edit box …,(2)-figures: to arrange the components of a GUI, (3)-callbacks : integrated IP functions to perform an actions) , for more details see Ch-10 in (Stephen,2002).

In this paper many functions in image processing field were integrated to callback. For example; vehicle number plate segmentation and extraction, where according to the image segmentation, the shape of an object can be described either in terms of its boundary or in terms of the region it occupies. The objective of segmentation technique in this paper is to partition a given image into regions or components for extracting objects from an image (i.e vehicle number plate segmentation and extraction) based on examining an image on a pixel-by-pixel basis and on utilizing the image information in a prescribed neighborhood to established boundaries between regions in order to obtain good segmentation results (Pitas,

2000). See other functions in the next section. in (Nassir,2008) it built image processing toolbox using MATLAB menu for some user and built functions. Mathwork group built many functions in MATLAB that are especially useful for image processing, such as imread, imfinfo, and imwrite. And others from the online image processing toolbox, User's Guide-"Getting Started" - "What is Image Processing Toolbox?". The Image Processing Toolbox provides a set of tools, which allow you to view and manipulate images. A few classic things that the Image Processing Toolbox allows you to do are: histogram equalization (imhist), filtering (imfilter), fast Fourier transform (fft2), converting color images to grayscale (rgb2gray), edge detection (edge), reading, displaying and saving images.
This article was distributed as follows: 1) introduction, 2) GUI callback IP functions, 3) GUI design steps, 4) results and 5) conclusion.

2. Theoretical Considerations

2.1 GUI callback functions

In this paper a set of MATLAB files were built and then used to implement a set of functions in image processing field (in addition to image segmentation explained above) as seen in figures 1 and 2.

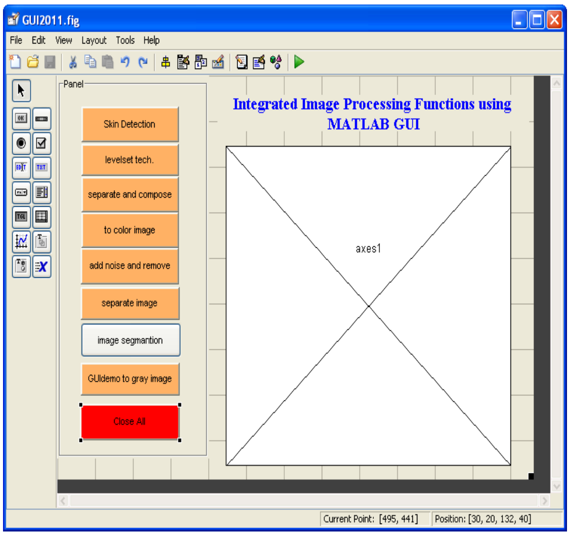

Fig.1. Shows the design of the GUI

Fig. 2. Shows the GUI figure after running the main function.

These functions were modified and tested through GUI. Following a short description for these functions:
1-Level set method: In mathematics, a level set of a real-valued function f of n variables is a set of the form:

$$\{ (x1,...,xn) \mid f(x1,...,xn) = c \} \quad (1)$$

where c is a constant; that is, it is the set where the function takes on a given constant value. When the number of variables is two, this is a level curve (contour line), if it is three (in 3-D; Surface (Interface) evolution), this is a level surface, and for higher values of n the level set is a level hypersurface. Level set methods are used for the implementation of curve/interface evolution under various forces. In this method many modified functions were used (BarisSumengen, 2005 & Matlab 7.6.0.324,2008).

a) function evolve2D () was used which is a high level function that takes an input, evolves it N iterations and returns the result. b) Function contour (Z) is a contour plot of matrix Z treating the values in Z as heights above a plane. A contour plot is the level curves of Z for some values V. The values V are chosen automatically. The contours are normally colored based on the current color map. c). Function imcontour; where imcontour(I) was used to create contour plot of image data and to draw a contour plot of the intensity image I. It was found that 50 iterations are very good for some images, and the evolution type depend upon the parameters values in function Phi (BarisSumengen, 2005 & Matlab 7.6.0.324,2008):see Fig.3.

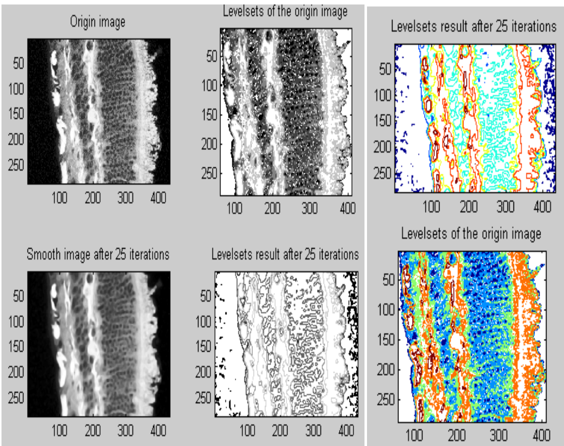

Fig.3. The final results of levelset segmentation method to brain image and elapsed time of processing(28.5 sec)

Fig.4. The vehicle number plate and extraction results

phi= evolve2D(phi,dx,dy,0.5,25,[],[],0,[],0,[],[],1,b); also the processing time depend upon iteration numbers.

2- Object extraction method: This is a program for extracting objects from an image. It was written for vehicle number plate segmentation and extraction for example. The input image is color input image; median filtering the image was used to remove noise, then finding edges, smoothing image to reduce the number of connected components. Then calculating all connected components in the image and displaying them in one figure. Finally storing the extracted image in an array (JenyRajan,2006).see figures(4 & 9).

3-In skin detection method (JenyRajan,2006): at first, Load an RGB image, convert the image data to doubles and compute skin likelihood for each pixel; show skin-likelihood image; threshold the likelihood to detect skin; show the original image, but replace the red band or blue band with the binary skin image, see Fig5. The detector works as follows:

- gathering a collection of images and mark each pixel as "skin" or "non skin"
- creating an RGB histogram for "skin pixels" and another one for "non-skin pixels" (they are 32x32x32 in size)
- For a particular bin (i.e. pixel colour), the log likelihood of it being skin is: log(H(R,G,B) / h(R,G,B))

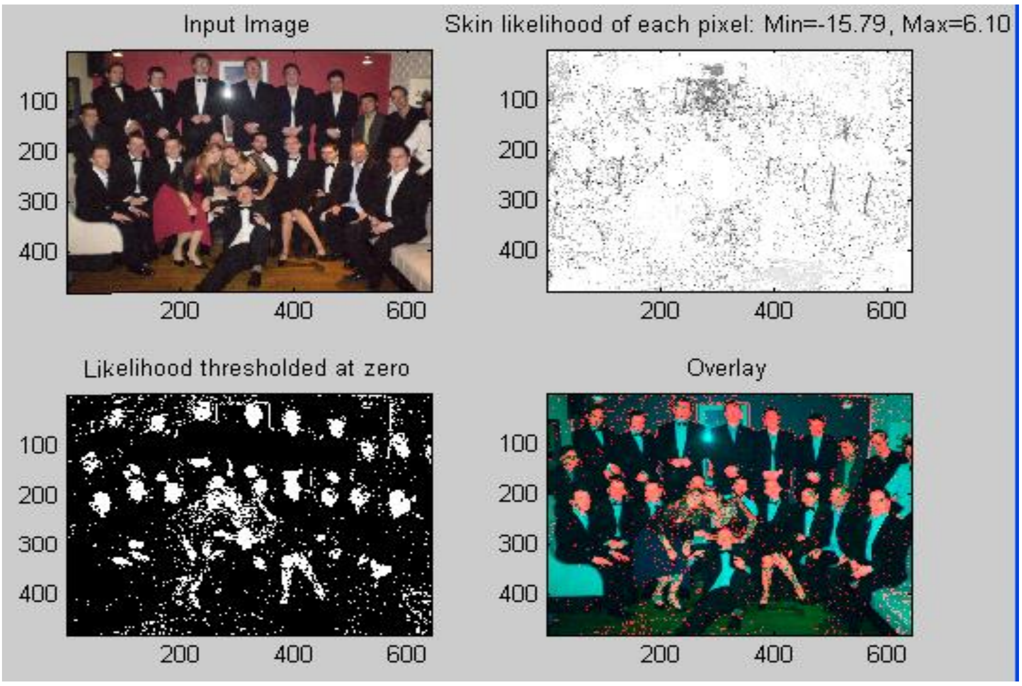

Fig.5. Skin detection result, see the face and the hand in the image (the processing time is 0.935 sec)

where, H is the skin histogram, and h is the non-skin histogram. For a new image, the log likelihood of each pixel was computed and then threshold the result to decide skin/non-skin.(Ciarán, Noel & Alan ,2007) ,where testing the black skins has problems in the decision.

4-In add and remove noise: this action is to explore noise reduction in images (e.g. salt &pepper(0.1 default intensity value , Guassian(the default mean value is 0.0 and the default variance value is 0.005 , Speckels (the default variance value is 0.1...) using linear and nonlinear filtering techniques [noise removal filter(e.g. median and adaptive(the default filter neighborhood is [3 3] , the output image class is of as the input image class. It is easy to add averaging filter, (Nassir, 2008), see Fig.6.

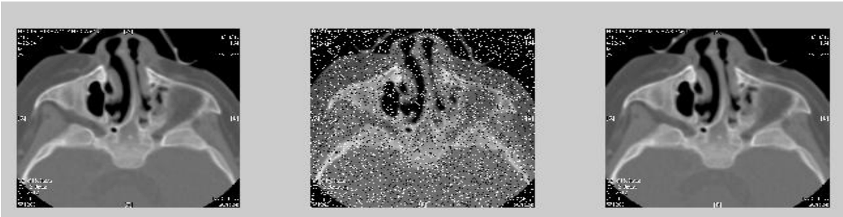

Fig.6. shows original brain image (left), the noisy image (mid), and the filtered image (right)

5- In GUIdemo to gray image call, in this function another GUI interface was called (took it from (Dr.Yingzi, 2009) and modified) through the GUI2011 to implement image brightness and convert color image to gray image tasks. See Fig 7.

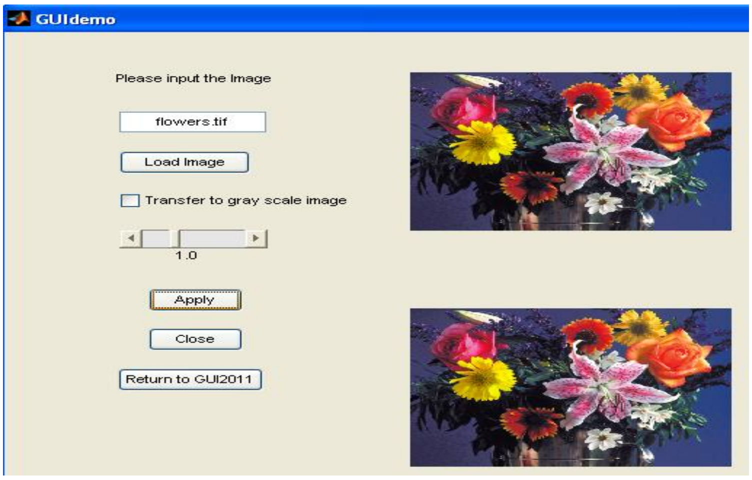

Fig.7. GUI demo to gray image call and control

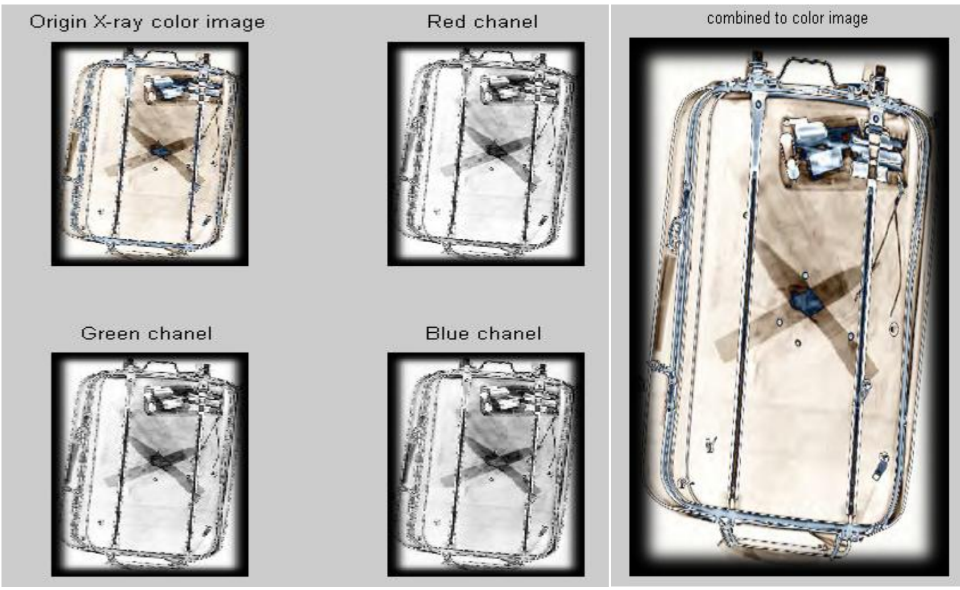

Fig.8. Split color image into its components (RGB) then combined them to get color one.

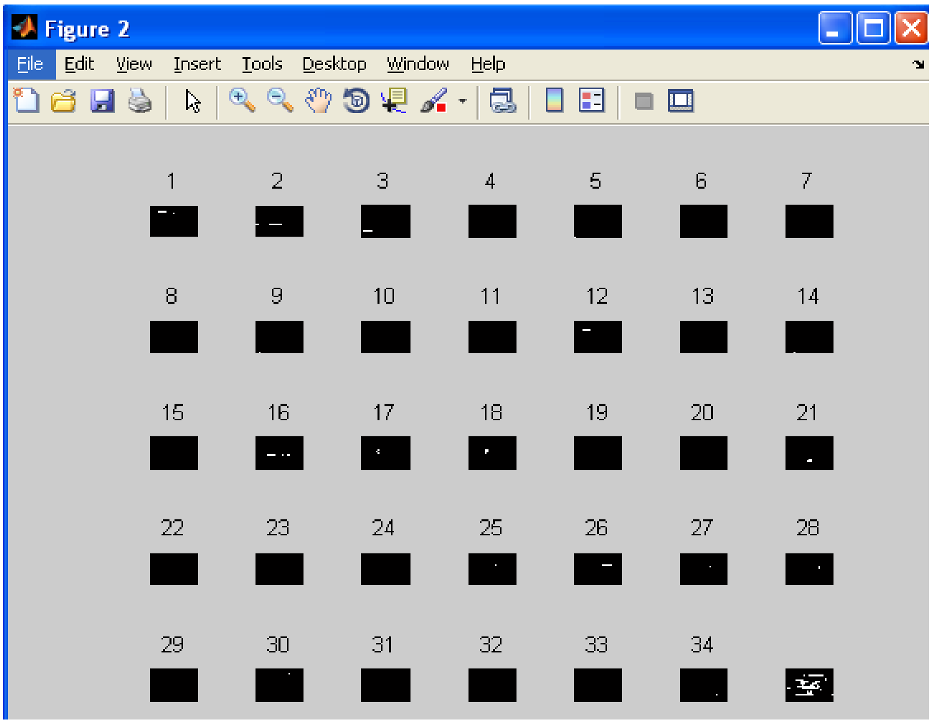

Fig.9. The vehicle number plate connected components

3. GUI Design steps

3. 1 The Blueprint of a GUI figure

At first, it is necessary to know what is the objective to have in the GUI figure (the final user interface), such as how many input/output images/plots, how many other kinds of control objects, and how to arrange it.

Second, it needs to come up the names for each control objects (including the axes to show images/plots). Usually, the names (tags) of the objects have two parts: the function of the object (such as "skin detection"), and the type of the object (such as "Push Button").see the main figure design in Fig 1.

3.2 Design the GUI figure based on the blueprint

Suppose that Fig. 1 is the GUI Figure we would like to design (i.e., using guide menu then property inspector menu to indicate the name of the objects we would like to have in the program). It includes following types of unicontrols: the Edit Box, the Static Text, the Push Button and the Axes.

To access to the object properties, select the object, right click the mouse, and select "property inspector". And to change the names (tags) of the objects, in the properties, we change the tags of the objects. The strings can be changed (the words that show the user of the objects, such as "levelset" and "skin detection") from the properties of the objects.

3.3 Designing the functions to fulfill the goals

If the designed GUI figure was saved as: GUI2011 (or whatever you want). The MATLAB will automatically save it as GUI2011.fig. And the MATLAB will also automatically generate the file GUI2011.m associate with this GUI figure. (Therefore, it need to carefully avoid saving the figure to be some name existed in your directory). Now, if in the MATLAB command line input “GUI2011” or whatever name you give to this GUI program, you would see Fig.2. As shown in Fig.2, It is just an empty GUI program. Some codes will add in the follows. If this m-file is open, you will find there are some predefined functions:

functionvarargout = GUIdemo(varargin)

This function is to set up the whole figure. Here, we do not need to edit it. (Keep it as it is.): function GUI2011_OpeningFcn(hObject, eventdata, handles, varargin)

This function is to initialize the GUI figure when it is opened at the first time. Here, we would like to show a default image in the axes1 with the name in the opening function is SKY.JPG as logo image. We can add the code as follows:

Here, if the GUI program executes, the following code will get:

functionclosePush_Callback(hObject, eventdata, handles)

Usually a push button named “close” will add in each GUI figure. The function of this button is to close the whole GUI program.Please add one line in this function: close all;

And all the other functions are built separately then integrated in one file called GUI2011.m. After running that file , the GUI2011.fig is display and from this GUI figure all the hidden functions are called to perform our tasks in image processing (such as skin detection , level set, segmentation ,…and so on , as shown in the Fig. (2) , left.

4. Conclusion

A graphic user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The designed GUI is easy to use and program. The results show that many image processing functions can be integrated in one file to be called and then get actions. There are very clear results, where the origin image and the processed result appeared in one figure to compare easy. The levelset technique and extraction an object method give very accurate and clear results. Finally it is easy using MATLAB programming to get useful image processing toolbox.

References

Stephen J. Chapman,(2002). Matlab programming for engineers, (2nd ed.) (pp.391-425).brooks/cole Thomson learning .CA 93950 USA.

I. Pitas .(2000). Digital image processing algorithms and applications. (PP-275-82).John Wiley &Sons , Inc. U.S.A.

BarisSumengen,Citing websites.A Matlab toolbox implementing Level Set methods., vision research lab at UC Santa Barbara.Retreived 2005,from http://barissumengen.com/level_set_methods/

The MathWorks, Inc. Citing Websites.MATLAB R2008a 7.6.0.324 .Retrieved 2008, from http://www.fileguru.com/user/the_mathworks__inc_

JenyRajan. Citing websites. Image segmentation & extraction, Retrieved 2006, from http://www.mathworks.com/matlabcentral/fileexchange/8031-image-segmentation extraction/content/seg.m

Ciarán Ó Conaire, Noel E. O'Connor and Alan F. Smeaton,(2007). Detector adaptation by maximising agreement between independent data sources. International Workshop on Object Tracking and Classification beyond the Visible Spectrum, IEEE.

Nassir H. Salman,(2008b,15-18 Dec. 2008). New image processing toolbox using MATLAB codes.paper presented at image processing ,2008. International Arab conference on information technology, ACIT2008.

Dr. Yingzi (Eliza) Du.Citing Websites. Matlab GUI tutorial. Retrieved May, 2009 , from http://www.engr.iupui.edu/~yidu/

SALMAN, N. H, born in 1960. He received his B.Sc. and M.Sc. degrees from Al-Mustansyriah University / Collage of Science in 1983 and 1989 respectively. He received his PhD degree in Image Processing and Pattern Recognition from Shanghai Jiao Tong University-China in 2002. His research interests include remote sensing, image processing and image analysis based on image segmentation and edge detection techniques. Also he is interesting in computer programming languages, Matlab programming. From 1982-1998, he worked at Space Research Center -Remote Sensing Dept. in Baghdad –IRAQ, also he was an assistant professor (from 2002) in the Dept. of Computer Science-Zarqa Private University-Jordan. Now he is head of Computer Science dept .Cihan University, Erbil-Iraq.

Gullanar M. Hadi, born in 1964. She received her B.Sc. and M.Sc. degrees from Collage of Science, Al-Mustansyriah University, Baghdad –IRAQ in 1985 and 1989 respectively. She received his PhD degree in Opto-Electronics engineering from Shanghai Jiao Tong University-China in 2004. Her research interests include image processing and nano-materials. From 1985-2006, she worked at Al-Mustansyriah University , collage of Science, Baghdad –IRAQ, from 2006-2009 she worked as a lecturer in several Jordanian University, from 2009 to present she was lecturer in Software Engineering dept. , Salahaddin University, Erbil-Iraq.