ADOBE PHOTOSHOP CS5 AND MATLAB

M. Kruľáková

BERG Faculty, Technical University of Kosice, Slovak

Abstract

This article discusses the possibility of interconnecting programs MATLAB and Adobe Photoshop CS5, where communication between Photoshop and MATLAB uses interface of JavaScript and the MATLAB library interface. These integrations allow you to call functions from the MATLAB command line or editor and view the result of image in the Photoshop.

The aim was to create a graphical user (GUI) interface MATLAB and to search the possibility of using the functions Adobe Photoshop toolbox in the stitching and processing of images.

1 Introduce

A lot of software is used for processing of digital image, but Adobe Photoshop offers link to MATLAB, where of image processing is on high-level technical computing for algorithm development, data visualization, data analysis, and numeric computation.

In this paper is presented possibilities of combine MATLAB commands with Photoshop image-editing features for image processing and stitching image. Combination of these commands used of Image Processing Toolbox together with Photoshop Toolbox gives more possibilities for another new methods and practices for image processing in MATLAB.

2 Principles of integration Adobe Photoshop CS5 and MATLAB

First condition for installation is necessary to have in Windows of 64-bit Photoshop of 64-bit MATLAB, and for 32-bit Photoshop with 32-bit MATLAB [1]. Operating system on PC was installed Windows_NT and for connection were used these soft wares toolbox MATLAB(R2008b) and Adobe Photoshop CS5, that were 32 bit. For the successful run of function testall(), which has had to ensure integration between these programs was needful observe settings in these programs with reference to [1], [2].

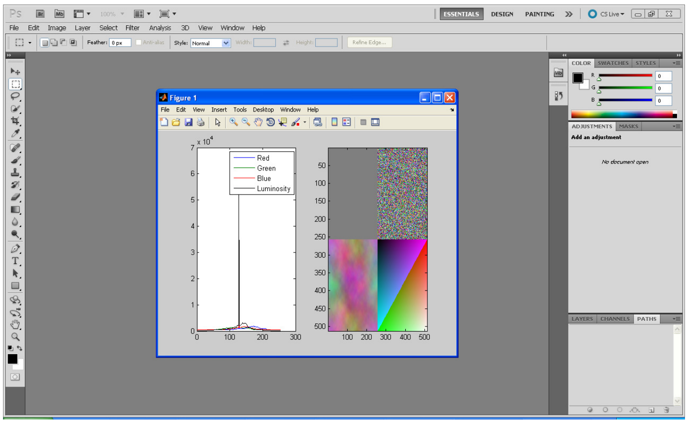


Figure 1: Final report of MATLAB function testall()

After connect to Photoshop from MATLAB (see Fig.1), typing commands into the MATLAB command prompt performs operations in Photoshop. You can run MATLAB algorithms and view the results in your image in Photoshop [1].

Then was possible to connect Photoshop from MATLAB via a function call pslaunch () and disconnect function via psquit ().These functions were called as General function in Photoshop toolbox in MATLAB.

Table 1: ADOBE PHOTOSHOP TOOLBOX - CATEGORICAL LIST OF FUNCTIONS

General functions	psconfig, psjavascript, pslaunch, psquit
Document functions	psclosedoc, psdocinfo, psdocnames, pshistogram, psnewdoc, psnumdocs, psopendoc, pssetastivedoc
Color management functions	pscolorprofiles, pscolorsettings, psconvertprofile
Layer functions	pslayernames, psnumlayers, pssetactivelayer
Channel functions	pschannelnames, psnumchannels, pssetactivechannels
Path functions	psgetpath, psnewpath, psnumpaths, pspathnames
Pixel functions	psgetpixels, psimread, psnewdocmatrix, pssetpixels psnewlayermatrix,
Filter functions	psaddnoise, psaverage, psblur, psblurmore, psboxblur, pscustom, psdustandscratches, psgaussianblur, pshighpass, pslensblur, psmaximum, psmedian, psminimum, psmotionblur, psoffset, psradianblur, pssharpen, pssharpenedges, pssharpenmore, pssmartblur, pssurfaceblur, psunsharpmask

3 Analyze of functions in Photoshop toolbox

These functions (see Tab.1) were written for image processing in MATLAB and view results in Adobe Photoshop CS5. This calling functions of Photoshop toolbox makes possible via communication of JavaScript. JavaScript enable write scripts via script/text editor via application ExtendScript Toolkit CS5 (ESTK) of Adobe. The ExtendScript is Adobe's extended implementation of JavaScript, which offer the entire standard JavaScript features [4].

Adobe provides the ScriptUI component, which works with the ExtendScript JavaScript interpreter to provide JavaScript scripts with the ability to create and interact with user interface elements. It provides an object model for windows and user-interface control elements within an Adobe application.

Every functions in Photoshop toolbox had calling with reference to functions which were used in library Extend Script Toolkit CS5 (ESTK). A shared library is implemented by a DLL in Windows and for MATLAB was implemented by psmatlab.dll, where loadlibrary calls in the Photoshop .m files must use the psmatlab.h file.

If the data type is a constant, the constant appears as a hyperlink to the constant's values. Constant names begin with a prefix that matches the Adobe application's abbreviated name. For example: The prefix **ps** is used for enumerations in Photoshop CS5 [3].

The prefix ps were also used for enumerations in MATLAB for created of Photoshop toolbox. As the example is presented function psopendoc.m, which has following source code.

```
function n = psopendoc(f)
% N = PSOPENDOC(F) F is the full path to the file. N is the resulting
%   Thomas Ruark, 2/23/2006, Copyright 2006 Adobe Systems
f = strrep(f, '\', '\\');
 % build up the JavaScript
pstext = ['var result = "";' ...
    'var errorDetails = "";' ...
    'try {' ...
    '     app.open(File("' f '"));' ...
    '     result = app.activeDocument.name;' ...
    '}' ...
    'catch(e) {' ...
    '     errorDetails = e.toString();' ...
    '     result = "8F6AFB7E-EC1F-4b6f-AD15-C1AF34221EED";' ...
    '}' ...
    'var b = result + errorDetails;' ...
    'b;'];
 psresult = psjavascriptu(pstext);
 lo = strfind(psresult, 'if
(app.document[1].length):app.document[2].length');
 if isempty(lo)
    n = psresult;
else
    error  (psresult(length('8F6AFB7E-EC1F-4b6f-AD15-C1AF34221EED')  +
1:end));
```

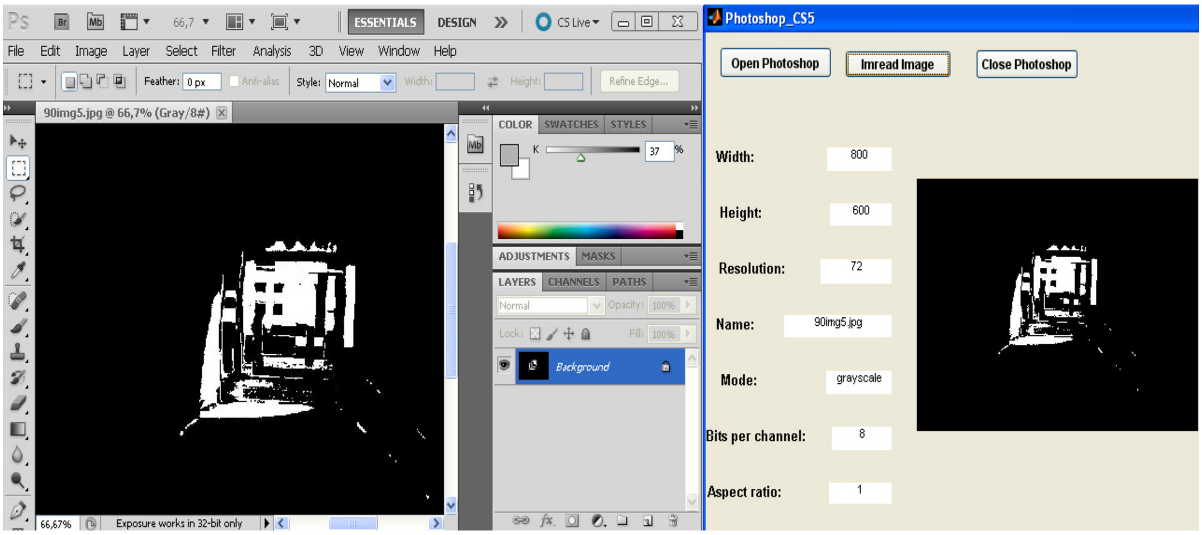
These functions psopendoc() and psdocinfo() were used in function pushbutton2_Callback(), see following source code, for simultaneously opened image in Photoshop and (GUI) MATLAB.

```
function OpenImg_Callback(hObject, eventdata, handles)

% select image from file
[FileName1,PathName1]=uigetfile('*.jpg','pushbutton2');
if isequal(FileName1,0) || isequal(PathName1,0)
    return
end
 I1=imread(fullfile(PathName1,FileName1));
imshow(I1)

% called function for opening image in Photoshop
n1=psopendoc(fullfile(PathName1,FileName1));
axes(handles.axes1);

% called function for viewing properties of image
[W,H, R, N,M,B,A]=psdocinfo();

% insert properties of image to component Edit Text in (GUI) MATLAB
set(handles.W,'String',W);
set(handles.H,'String',H);
set(handles.R,'String',R);
set(handles.N,'String',N);
set(handles.M,'String',M);
set(handles.B,'String',B);
set(handles.A,'String',A);

 handles.image1=I1;
guidata(hObject,handles)
```

Here was presented part of MATLAB's code of Photoshop_CS5.m which describes possibility of using of both toolboxes in image processing. Advantage of some functions of Photoshop toolbox is that provides base information about image, which Image processing toolbox has. For example function psinfodoc() it is resolution, mode, and aspect ratio, etc.

When we want obtain to some information about image we must used more of calling on functions from Image Processing Toolbox in our code via MATLAB command line or editor..

Although for image processing MATLAB used of Image Processing Toolbox. These functions of toolboxes we can use together in body of new function via graphical user (GUI) interface MATLAB (see Fig. 2).



Width: 800

Height: 600

Resolution: 72

Name: 90img5.jpg

Mode: grayscale

Bits per channel: 8

Aspect ratio: 1

Figure 2: The result of pressing pushbutton named Imread Image

Down in the window is presented second created source code in (GUI) MATTLAB. Here is button named StitchingImg which after pressing via function cat() of Matlab   is stitching of two images and via function psnewdocmatrix() is created new document in Photoshop.

```
function StitchingImg_Callback(hObject, eventdata, handles)
% reference of reading first image from calling function imread
dvaI1=handles.obl;
% reference of reading first image from calling function
psgetpixels
dok1=handles.p1;
% reference of reading second image from calling function imread
dvaI2=handles.obr2;
% reference of reading second image from calling function
psgetpixels
dok2=handles.p2;
 [m1 n1]=size (dok1);
 [m2 n2]=size (dok2);
% reference  of calling function cat for stitching two gray level
images
siva=cat(2,dok1(1:m1,1:n1/3),dok2(1:m2,1:n2/3));
% calling function of psnewdocmatrix for create new document in
Photoshop (see Fig. 3)
psnewdocmatrix(siva);
% reference  of calling function cat for stitching two color images
farebna=cat(2,dvaI1,dvaI2);
% calling function of psnewdocmatrix for create new color document
in Photoshop (see Fig. 4)
psnewdocmatrix(farebna);
guidata(hObject,handles)
```

The same size of rows depends on with the use function cat which is able to concatenate matrices. When want vertical concatenate two images, they must have the same number of columns. By horizontal concatenate images they must have the same number of rows. This is condition concatenate image matrix of function cat() [5]. On the Fig.3 and Fig.4 are presented vertically stitching of matrix images. Of course is possible also horizontally stitching of matrix images.

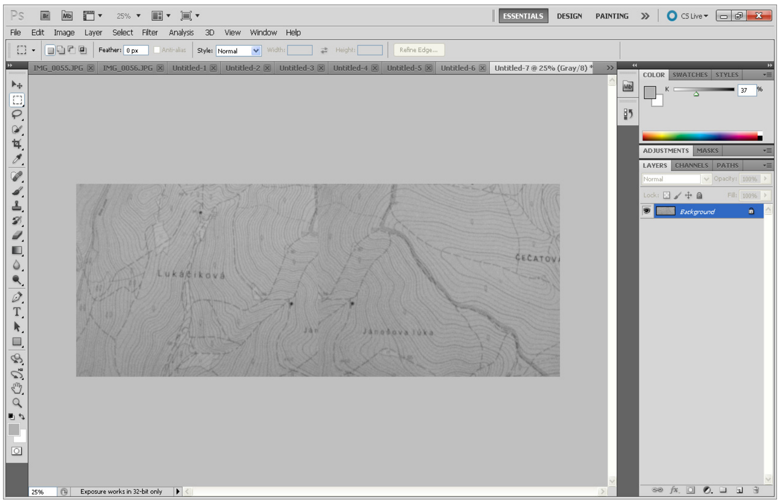

File Edit Image Layer Select Filter Analysis 3D View Window Help
ESSENTIALS DESIGN PAINTING
Lukáčiková
ČEČATOV
Jánošova lúka

Figure 3: Stitching of gray level images created via new document

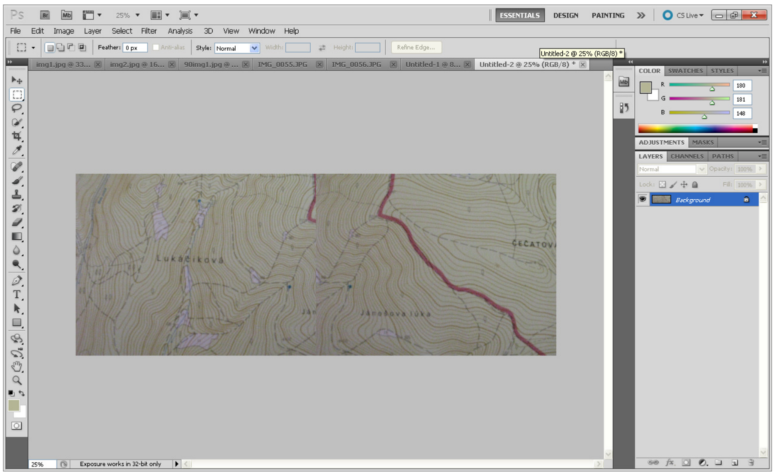

File Edit Image Layer Select Filter Analysis 3D View Window Help
ESSENTIALS DESIGN PAINTING
Lukáčiková
ČEČATOV
Jánošova lúka

Figure 4: Stitching of color images created via new document

4 Results

It follows that of combination these functions and toolboxes make easier our practices of image processing and stitching images. This process stitching of large image into digital form via these toolboxes will have worth for preserving works of art painter and mosaic maps into digital image. Advantage is that these programs communicate via JavaScript. This fact gives possibility of extend set of Photoshop toolbox for creating new .m functions for stitching images into large model.

Acknowledgements

This paper was published thanks to support of VEGA 1/0390/10 grant from Slovak Agency for Science.

References

[1] © 2010 Adobe Systems Incorporated and its licensors, *Using Adobe Photoshop CS5 for Windows and Mac OS.*

[2] © Adobe Systems Incorporated 10/25/2006, *Lame de MATLAB Photoshop, System requirements.*

[3] © Copyright 2009 Adobe Systems Incorporated, *Adobe Introduction to Scripting.*

[4] http://morris-photographics.com/photoshop/tutorials/scripting2.html

[5] M. Kruľáková*, Matrix technique of image processing in Matlab,* ICCC'2010: proceedings of 11th International Carpathian Control Conference, Rekatel, 2010. - ISBN 978-963-06-9289-2. - P. 203-206.