Design of user interfaces

Image thresholding

2018/19 course - Practice 5

David Bohmann
Petr Lukašík

October 13, 2018

Contents

1. Assigment

Some applications allow a large number of functionalities among which the user can choose. The realization of an interface with the components seen so far would make the design not very functional since the large number of buttons, text boxes, sliders, etc. It would be very hard and difficult to understand. In this type of applications, the choice is made over the menus that allow grouping of the different functionalities available to the application and that are "hidden" until the user activates them. This practice therefore aims to introduce the Swing components related to the menu bar (classes JMenuBar, JMenu, JMenuItem, ...) and some types of dialog box (JFileChooser and JOptionPane classes) so that the user can provide information to the application which can show information to the user. The objective of the practice will be for the student to implement an application to perform a thresholding process on an image that the user can select from a folder on the computer. The developed application should have a title in the main window as well as show the following options grouped in different menus of a menu bar:

- Allow the user to open an image from a folder for using a thresholding.
- Allow the user to save the image resulting from the thresholding process in a folder.
- Apply the thresholding process by requesting the threshold value through a dialog box.
- Show both the original image when it is loaded and the result of the thresholding process.
- Have a "Help" menu that shows information "About" the application with information about it.
- Include the option to exit the program and the closing of the window in both cases you must ask the user for confirmation.

1.1 Implementation notes

The thresholding of a gray level image consists in assigning a value of 1 (or 0) to all pixels with a value higher than a fixed threshold, and a value of 0 (or 1) to all pixels with a lower value or equal to the threshold. As the implementation of this type of methods is not the subject of this matter, the code that carries out the process is attached below. Because the OpenCV library is used, it is necessary to include the opencv.3.4.3.jar module in the Netbeans project and to define the path to the shared library as options in the virtual machine (Properties - Run). Djava.library.path = path-to-library-shared-opencv_java343.dll.

```
private Mat umbralizar(Mat imagen_original, Integer umbral) {
       // crear dos imagenes en niveles de gris con el mismo
          tamano que la original
       Mat imagenGris = new Mat(imagen_original.rows(),
          imagen_original.cols(), CvType.CV_8U);
       Mat imagenUmbralizada = new Mat(imagen_original.rows(),
          imagen_original.cols(), CvType.CV_8U);

       // convierte a niveles de grises la imagen original
       Imgproc.cvtColor(imagen_original, imagenGris, Imgproc.
          COLOR_BGR2GRAY);

       // umbraliza la imagen:
       //     pixeles con nivel de gris > umbral se ponen a 1
       //     pixeles con nivel de gris <= umbra se ponen a 0
       Imgproc.threshold(imagenGris, imagenUmbralizada, umbral,
           255, Imgproc.THRESH_BINARY);
       // se devuelve la imagen umbralizada
       return imagenUmbralizada;
   }
```

To load the shared library, it must be included in the class that implements the application:

```
  static {
       System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
   }
```

To convert an image in Mat format from OpenCV to BufferedImage you can use the toBufferedImage method of the HighGui module:

```
  BufferedImage imagen = (BufferedImage) HighGui.
     toBufferedImage(imagen_mat);
```

1.2 Examples of thresholding an image

(a) Original image (b) Threshold = 150 (c) Threshold = 200

Figure 1.1: Thresholding examples

2. Implementation

Main window (shown in pic 2.1) is created by free area for image and a menu bar. As requested in the assigment, there are no buttons, textfields or sliders that disrupt the user clarity in the application. The menu bar contains file and help menu. Help menu contains only "About" option to show a modal window (JOptionPane) with informations about the application (pic 2.3a). In the file menu (ref. 2.1) is present choices for: open image, save image, threshold image and exit application. These options are allowed to be executed by keybind. These options are disabled and enabled by current state of application e.g. you can't save a image if there is no image loaded. Therefore first you have to open and image from a folder, which is shown in picture 2.3b. After image is loaded, it is shown in the main area of the application. Afterwards you can either save this image or threshold the image. Choosing threshold option opens a new modal window (JOptionPane) with textbox to choose a threshold value (pic 2.2a). If input is wrong, application will show error window with response. If input is right there will be applied thresholding filter onto image (as shown in 2.2b). Afterwards the save option is again avalabile to save the thresholded image. If user wants to exit the application by closing the window or choosing the exit option in the file menu he is asked if he really wants to do so (by JOptionPane pic 2.3c).

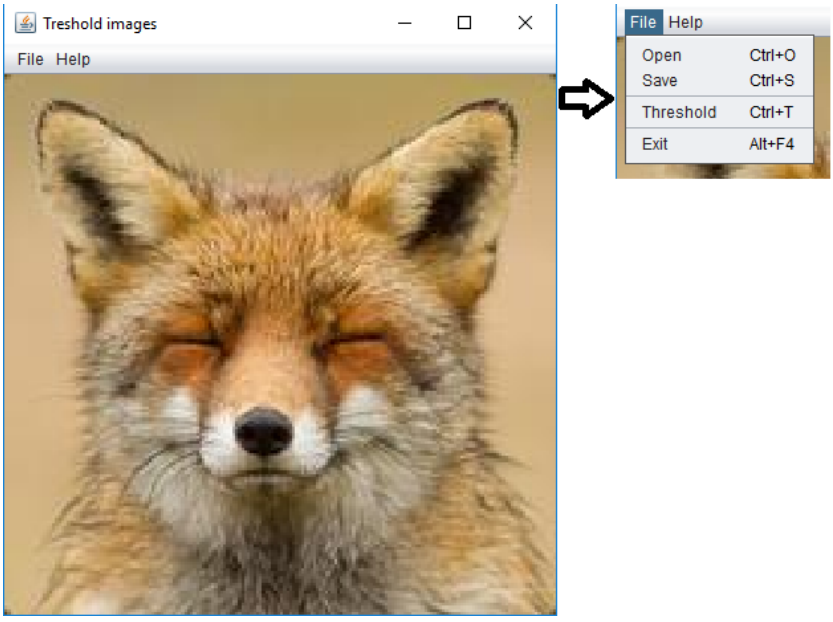
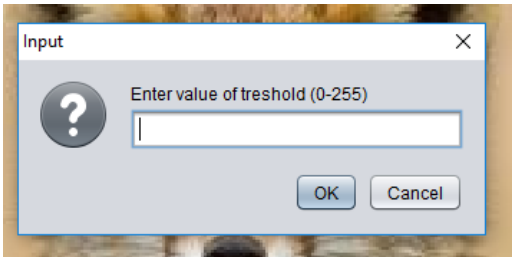


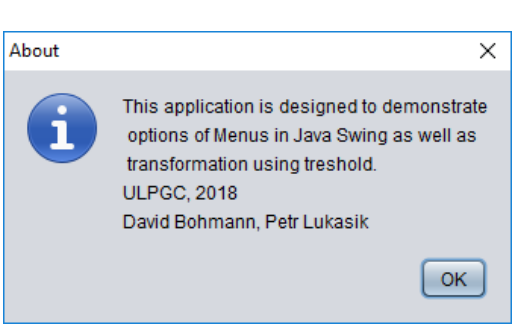
Figure 2.1: Main application area and dropped file menu



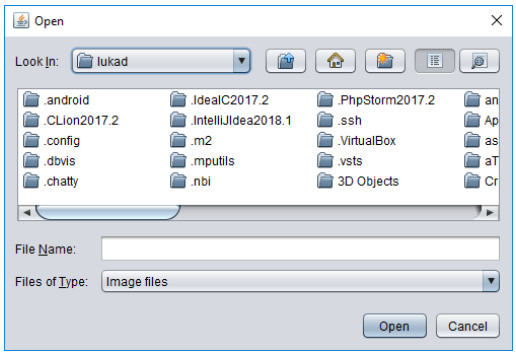
(a) Thresholding options window

(b) Thresholded image

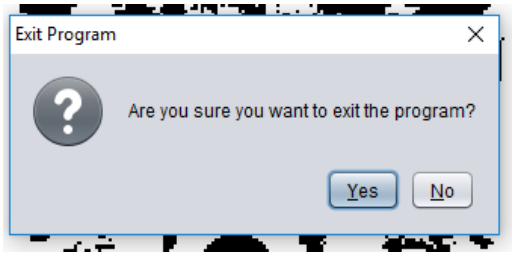
Figure 2.2: Image thresholding application



(a) About window



(b) File chooser for opening a image



(c) Exit prompt window

Figure 2.3: Dialog windows in this application

3. Shneiderman and Plaisant principles

- **Consistency**
  There is no disrupting or unnecessary button which could confuse the user. Every option is in menu bar in the upper side of the application.
- **Universal usability**
  Every option avalabile has its own key shortcut for more experienced users to cut time between operations.
- **Informative feedback**
  Any action done by user is shown in the main area of the application or new option window is shown.
- **Design dialogue to yield closure**
  Inputs are made only by filechooser which has filter only for images. When exiting application user is asked if he really wants to quit the program.
- **Simple error handling.**
  Only when trying to threshold image with wrong numbers (or wrong format) there is immediately shown an error window with information about what range of numbers is only allowed.
- **Permit easy reversal of actions**
  There are not implemented any reversal functions. User needs to reload the old image if he wants to apply thresholding with different threshold.
- **Support internal locus of control**
  The user is in full control of choosing and saving image. Every action is instantly applied and shows results.
- **Reduce short-term memory load**
  For such a small application our application is fast enough for human needs.