Multimedia Application Final Project

Maria TROCAN – Dominique YOLIN

March, 2025

TOPIC

The project is to develop a small GIMP-like image editor, with basic functions. The advantages of this project are:

- the modularity, it is easy to separate part of the code for group work.
- using image manipulation
- adaptability, you can implement various functions and features depending on what interests you the most.

As usual in the approach by project, it will be done by group of 3–4 students.
Also, it is mandatory that each group use a GitHub account, so tutors can check the student's code.

1 Project Time Table

This project will be realized during the 6 last weeks of the course (as soon as you have finished your TPs you can start working on your Project)

- Presentation: 26 May
- Final report and code: to be sent before 25/05/2025 11:59 PM.
- You will be evaluated on your code, report, and oral presentation.

The report must explain the OpenCV functions you used, how you exploited them and the image processing technique you implemented. All team members must participate in writing the report, and it must clearly show who did what in the project. No code needs to be in the report. At the oral exam you will present your work and show how the code works (demo and code structure).

More detailed instuctions will be given on the moodle page of the course in the chapter dedicated to the project

2 Functions to use in the project.

These are the functionalities you have to include in the project:

- dilatation / erosion
- resizing
- lighten / darken
- panorama / stitching
- Canny edge detection

2.1 Dilatation / erosion

These functions use the Open CV functions :

- **`cv::dilate`**
- **`cv::erode`**

The erosion size (respectively the dilatation size) should be a parameter the user can set within a given range. So there should be two parameters to this function:

- the input image
- the erosion size

Plus, eventually a parameter to choose between erode and dilate (it is up to you to choose how you want to make that choice available for the user). The exact range of the sizes is not an issue.

2.2 Resizing

This functionality will use the function:

- **`cv::resize`**

It takes as input:

- an image
- either a factor or two dimensions

2.3 Lighten / darken

This function doesn't require any specific function from OpenCV. It can be done with matrix manipulation. So, there should be two parameters to this function:

- the input image
- the lightening / darkening factor

As for the erosion / dilatation, the user should be able to set the magnitude of the transformation in a given range.

2.4 Panorama / stitching

The purpose of this function is to take as parameters the name of several images, or a directory, and try to stitch them together, ie detecting the common part. You can use the cpp sample "stitching from OpenCV" to see how it works.

2.5 Canny edge detection

This functionality will use the function:

- **`cv::Canny`**

It takes as input:

- an image
- two thresholds
- a kernel

3 Interface

No specific interface is required. Command line is acceptable, especially because it won't work otherwise on ISEP computer.

4 Advanced functions

Some more advanced functions must be added (the implementation of the above mentionned functionnalities will be no more than half the grade)
You are free to implement whatever interest you. Here are some propositions but there are a lot more possible (check with your teacher first):

- Video manipulation
- face identification / recognition
- Background separation
- Objects detection
- advanced gui
- …

Each person will have to participate to both basic and advanced features. Your personnal involvement and achiement will have to transpire in your code through well written comments, in your report and during the presentation.
We expect from each of you to present your own work with details during the presentation and be able to anwer questions on it.