Bachelor in Science - Computer Vision

Computer Science – QT and C++

Project guidelines, Assessment and deliverables

If you already submitted a proposal which has been accepted, directly go to the Requirements section. Once you will have created your github repository and trello board, clearly recall your project's goals and objectives.

1 – Project topics

1.1. Shared functionalities

You will have to propose and implement a Qt/C++ application related to color in the general sense. The project is composed of a common set of tasks upon which you can propose two different types of applications. The common functionalities of the project can be summarized as follows: pixelise an image, which requires:

- The ability to load and display an image located anywhere on your hard drive, and to save any processed image.
- The provided software should allow to transform the loaded image into a second one such that the pixels' color of the second image is computed according to various methods (average, median, most represented color, etc.) so that the image is pixelised. For instance, the image below shows an example of the expected result (using pixel filters in Gimp):

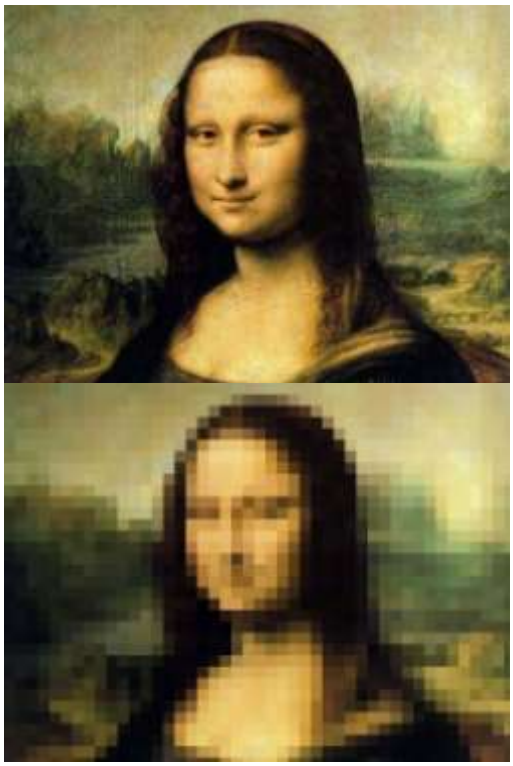

Figure 1: Example of expected results

1.2. Project branches

Once you can pixelise an image, you can choose between two types of applications for your project, namely Pixel Art Rendering or a Nonogram game

1.2.1. Pixel Art rendering

In this context, you will transform the pixelized image into a third one in which its "big and blurred" pixels are now represented with images from a set of images of your choice. You can choose as many images as you want, the only restriction being that these images should not be included as resources of the project, so that your application can automatically load any "database" of provided images from your drive without recompiling the entire programme. Eventually, you should be able to represent the final images in various ways, such as, for instance using painted wood bricks, as showed below and available online here:

Or maybe you prefer to use Lego bricks, as available from here…

1.2.2. Nonogram game

If you are not interested in Pixel Art, the second choice offered is to implement a nonogram game. If you do not know nonograms, please refer to http://www.nonograms.org/ for illustrations and games of black and white or color nonograms. In a nutshell, a nonogram can be understood as a colored 2D variation of the crosswords game in which you know, along each row and each colum, the number and the colors of the tiles to be placed. For the detailed rules and principles of nonograms, please see the Wikipedia page available here.

The objective is simple: propose a QT Gui application for playing nonograms in which the seeked image is obtained after pixelization, as explained in the Shared functionalities section. You can strongly inspire from the behavior of the web application available in http://www.nonograms.org/, but you are also totally free to update, modify or suppress functionalities.

2 – Requirements

To be as concise as possible, this section is split in two halves: **what you must do**, and **what you must not do**.

The project must

- **Be illustrative of your ability to develop graphical applications** using Qt, C++ and good oriented object practices.
- **Be restricted to most of the notions studied this year**. In other words, you are not allowed to use any other libraries other than Qt libraries (no OpenCV, no Eigen, no OpenGL, etc.)
- **Focus on OOP practices** (constructors, destructors, access rights, inheritance, template classes, etc). Are the classes correctly designed? Why such strategy has been implemented?
- **Focus on simplicity of the interface**. Is it user friendly? Are the menu well designed. Is the user interaction (through the cursor, keyboard, or other) carefully considered? Etc.
- Focus on your ability to **use numerous Qt widgets and objects and be illustrative or your ability to construct rich GUI applications** with menus, drawing, shortcuts, tabs, proper signal and slot connections, etc.
- Be **fully commented and documented** (see assessment and deliverables section). If you use some code found online, cite your sources otherwise this will be considered as plagiarism.
- Be **easily compilable**, otherwise provide the necessary instructions

The project must NOT

- **Implement tricky and unnecessarily complex algorithms**. If nothing works because there is a tricky bug in a non linear min-square minimization, this will be of your responsibility because no one ever asked you to tackle such problems (and you will see them next year anyway). **Forget advanced mathematics in this project because this is simply not the objective**. If you have a doubt, just ask for clarification.
- Be a console application only. **Graphical user interface is mandatory.**
- Be developed using other IDEs than Qt and be a mix of various languages (avoid QML). You can tweak Qt CSS.
- Use Matlab engine, mex, or anything related to other programming languages: stick to C++

- Have anything related to network, web, SQL, etc.

3 – Assessment and deliverables

From now and until the end of the academic year (May the 26th for the defense), you will be assessed though 3 marks

1. The quality of your implementation (equivalent to a "Labs" mark), noted CL. The labs are designed so that you will learn the necessary techniques for your project. This means that you have to prepare the Labs at home and join the Labs cessions with questions and issues to be worked with your professor and classmates.
2. The project management (continuous control), noted CC. This will be evaluated from your trello board and github repository
3. The exam mark, which wraps up the defense, the report, and the demo, noted CE

Your result for the semester will simply be computed as CF = (CL+CC+2*CE) / 4

In order to fairly assess the three previous items, please find below the corresponding deliverables with their respective deadlines:

1. Dully commented source code (and compilation instructions if needed). Deadline: Thursday May the 25th , project defense will be held on May the 26th
2. Continuous progress reports. Create a github repository (https://github.com/) and a trello board (https://trello.com/) and invite me to both so that I can see all you commits, concerns, and can interact with you on a daily basis
3. Report in pdf format (no later than Sunday May the 25th) + presentation and demo during the defense day on May the 26th

4 – General remarks and comments

1. If you have any question or doubt, please do not hesitate to communicate through Edmodo, so all your classmates will benefit of the discussion.
2. I am your project tutor, thus consider me as a partner of your team: ask me what I think about your project. Do that quite often and never hesitate to seek for some intermediate feedback.
3. Do not waste time and start working on the project now!
4. Enjoy!