Università
della
Svizzera
italiana

**Faculty
of Informatics**

Computational Fabrication (Spring 2023)

Assignment 2: Halftoning (10 points) May 9, 2023

1   Introduction

In this assignment you will be implementing two image halftoning techniques: Dithering Algorithm and Error Diffusion. You are also asked to prepare a short report summarizing your implementation and experiments.

<div align="center">Original picture   Thresholding   Dithering   Error diffusion</div>
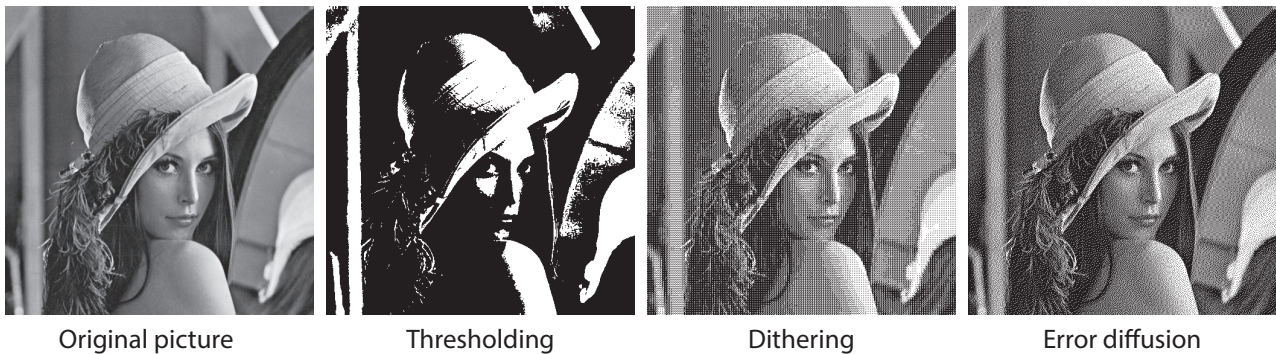
<div align="center">Figure 1: Example input and output for the assignment.</div>

To start you will be provided with an initial code that can read and write images in PGM format, an example of simple halftoning technique, and two example images for testing your solution. The solution should be implemented in C++.

2   Getting Started

We provide a C++ implementation of a simple console program that takes as an input one PGM file, a path to the output file, and the name of the method that should be used for halftoning. Current implementation takes care of reading and writing images. It has also implemented a simple threshold method to make it easier for you to start experimentation. Your task is to fill in the gaps of two functions `convert_using_dithering` and `convert_using_error_diffusion`. After compiling the C++ file you can run the program using the following command:

<div align="center">$ Halftoning InputFile OutputFile Method,</div>

where *Method* can be: Thresholding, Dithering, or ErrorDiffusion.

When you compile the source code that is provided with the assignment you should be able to run the Thresholding method using command:

<div align="center">$ Halftoning test1.pgm result.pgm Thresholding</div>

3   Bonus (5 points)

You can earn extra 5 points if you implement an improved version of error diffusion technique. As discussed during the lecture different schemes of distributing the error perform differently depending on the grey value they try to approximate. In particular, we discussed a method that uses different strategies for shadows, midtones, and highlights. Another improvement can be potentially achieved if instead of simple zigzag order of processing the pixels a Hilbert curve is used. To get the extra 5 points, please implement either of the two improvements and discuss the results on two test images provided with the assignment. To extend the current code please add an additional possible value of the parameter *Method* - ErrorDiffusionExtension.

4   Submission Instructions

Submit your assignment via iCorsi by May 24, 2023. The archive must include a C++ file with your solution as well as a report with results of your experiments.

Please use only standard C++ libraries. In fact, you do not need anything more than what is currently used.