Error Diffusion Dithering*

Leonardo Rezende Costa - RA262953**

I. Introduction

This report describes the work done related to the first project proposed on class. The aim of this project is to implement an error diffusion dithering algorithm to reduce an image depth.

II. Script execution

This section describes how to execute the script and it's dependencies.

The ***t1.py*** file has the script used in this work to generate the results. It's made statically to process the images in the project's folder. It can be executed by ***python t1.py*** in an environment with all the dependencies listed on subsection II-A

A. Dependencies

The dependencies to run the python3 script are:

- Numpy
- Opencv-python

III. Implementation

This section describes how the algorithm works and what was the approach to develop the script.

A. Diffusion error approach

This approach aims to change the intensity to 0 (minimum) or 1 (maximum) and then propagate the difference from the original value to the pixel's neighbors. This way, a region with more 0 pixels will be perceived by human eye as a dark region; a region with more 1 pixels will be perceived as a brighter region; a region with the same amount of black and white pixels distributed equally will be perceived as gray, and this logic continues for the proportion between black and white in the image's regions.

B. Algorithm

First, the Floyd and Steinberg script presented at page 132 of [1] was implemented. As the operation may seem like a convolution, it has fundamental differences, but some concepts can be applied in order to make a more generic algorithm. The approaches looks like a mask, with a field that is the current pixel and other fields at the sides and bottom. So, an algorithm that takes a mask and pass it through the image was developed. For each pixel, the value is approximated (to 1 or 0) in the result image, and the error (that is, the difference between the original value and the approximated) is distributed to it's neighbors on the original image. When a cell reaches an address out of the image, it's ignored.

C. Image scanning approaches

The two approaches chosen to scan the image was: **left to right** and **zigzag**. For this, a parameter called *alternate* is passed to the method, and when it's true, the horizontal range changes to right-left in the odd rows. Additionally, the kernel is flipped horizontally to propagate the errors in the right direction (never changing a pixel that was already rounded on the output matrix).

D. Colored images

To process RGB images, each channel was dithered separately, and then mounted as a three dimension array again.

IV. Results

For each of the approaches, the images represented on Fig. 1 was used as input[1]. As the approaches have different shapes, the pattern in the result changes along the tests. In Fig. 2 can be seen the difference between **Stevenson and Arce** and **Floyd and Steinberg** results at the same location of the same input.

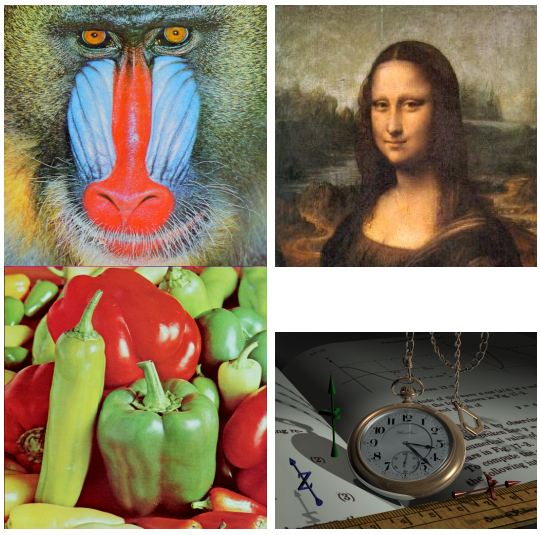

Fig. 1. Original images used as input

*Report of MO443 - Introdução ao processamento de imagens digitais

**Instituto de Computação - Mestrado, Universidade Estadual de Campinas

[1]All those images was get at <https://www.ic.unicamp.br/ helio/imagens_coloridas/>, accessed in 11/09/2019

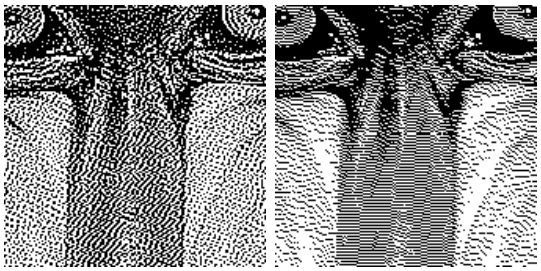

Fig. 2. Difference between patterns generated with different approaches. In the left, Stevenson and Arce. In the right, Floyd and Steinberg

At Fig. 3 can be seen the difference between the two scan approaches using **Floyd and Steinberg**. as the mask reaches just some of 8-neighbors pixels and the weight is higher at the right side, when a pixel has an intensity near to the threshold, it propagates too much error to its right neighbor, creating a result with more horizontal lines. When the zigzag approach is used, this horizontal lines (the black and white ones) are replaced by a pattern with more dots. This difference is less evident in the other approaches with a bigger mask, as the pixels are more influenced by more distant pixels that have different intensities and that distributes complementary errors.

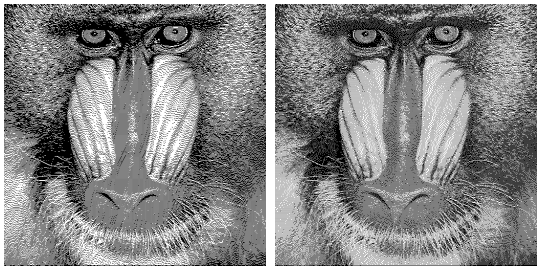

Fig. 3. Difference between left-to-right approach (left image) and zigzag approach (right image)

Fig. 4 shows the result of Burkes approach with zigzag scan in the *peppers* input. As each channel are converted to a binary image, the combination of the three channels result on colors: **red**, **green**, **blue**, **cyan**, **magenta**, **yellow**, **white** and **black**.

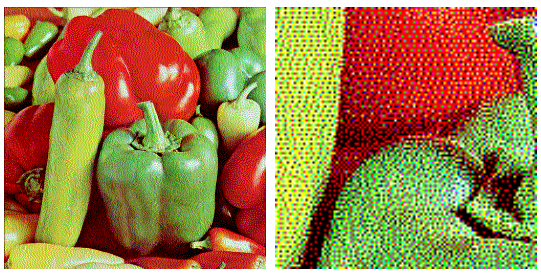

Fig. 4. Result from Burkes applied to *peppers* input

On Fig.s 5 and 6 can be seen the the result of all approaches suggested on the lab on *watch* and *monalisa* input using zigzag method. All of them looks like the input when plotted with a small size. That's one of the purposes of the technique, but with a zoom in (like did in Fig. 4) it can be seen the absolute pixel values and the difference between the approaches. Fig. 7 highlights the dark region of **Jarvis, Judice and Ninke** and **Floyd and Steinberg** results. On the right one, pretty much all texture around the arrow was lost and became pure black. On the left one, can be seen that more pixels of the paper at the back of the clock appear after dithering, but when looked this close, it just seem like noise.

Fig. 5. *Watch* dithering results. From Up-left to bottom-right: Floyd and Steinberg, Steverson and Arce, Burkes, Sierra, Stucki and Jarvis, Judice and Ninke

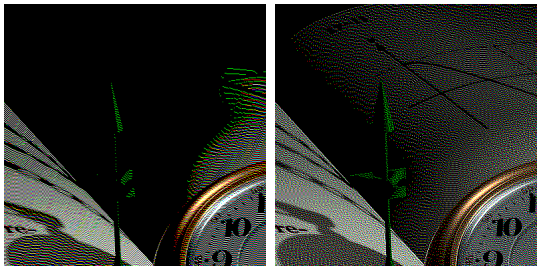

Fig. 7. Difference between **Jarvis, Judice and Ninke** and **Floyd and Steinberg** results from *watch* input using left-to-right approach.

V. Conclusion

The changes of the approaches leads to very different results when looked close, but the main purpose of this technique is to plot high resolution (high dimension in a small area) in context with low resource, like old newspaper printing (with monochrome images) or low cost magazine printing (with colorful images), so the result achieve the proposed goals. Another interest detail is that, as the depth of the image changes 1/8 per channel, this half-tone technique can be used to reduce an image weight, losing information but achieving good results, depending on the context. Another use of this technique is in led panels, that mix dithering with

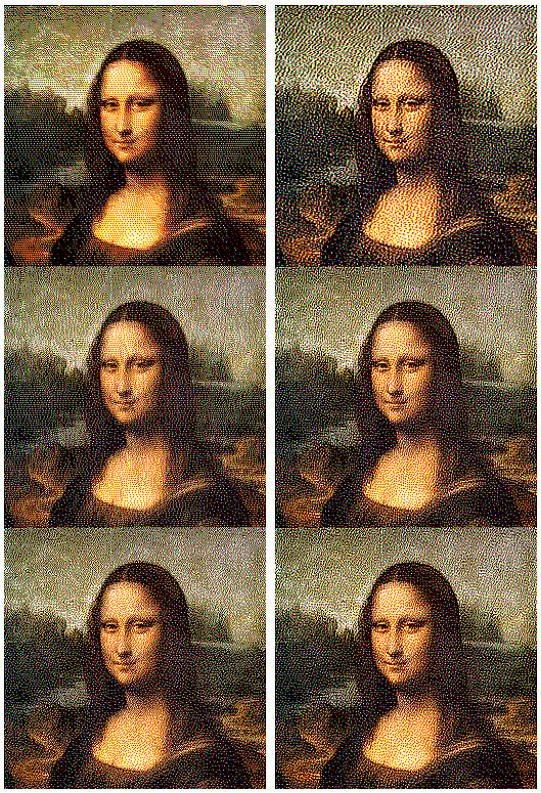

Fig. 6. *Monalisa* dithering results. From Up-left to bottom-right: Floyd and Steinberg, Steverson and Arce, Burkes, Sierra, Stucki and Jarvis, Judice and Ninke

a blink technique, so more frames can be processed at time and lesser micro-controller ports is used to print something on the screen.

APPENDIX

The following images are grouped by approach and then by scan technique.

REFERENCES

[1] H. Pedrini, Introdução ao Processamento Digital de Imagem, Slide - Realce, 2019.

Fig. 8. All results from **Floyd and Steinberg** approach, using left to right scan (first four) and zigzag scan (last four)

Fig. 9. All results from **Stevenson and Arce** approach, using left to right scan (first four) and zigzag scan (last four)

Fig. 10. All results from **Burkes** approach, using left to right scan (first four) and zigzag scan (last four)

Fig. 11. All results from **Sierra** approach, using left to right scan (first four) and zigzag scan (last four)

Fig. 12. All results from **Stucki** approach, using left to right scan (first four) and zigzag scan (last four)

Fig. 13. All results from **Jarvis, Judice and Ninke** approach, using left to right scan (first four) and zigzag scan (last four)