POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Energy Management for IoT

Lab 2 Report

Gabriel GANZER
271961

A.Y. 2020/21

Table of Contents

1. Image Manipulation

The main goal of this laboratory section is to demonstrate how image manipulation can be used in Organic Light-Emitting Diode (OLED) displays as a mean of reducing power consumption. The scripts *main1.m* applies different techniques over the extracted image-set, evaluating the trade-off between power consumption and image distortion.

1.1. Color Reduction

The first transformation evaluated consisted of a simple pixel-wise color reduction implemented by the MATLAB function *color_manipulation.m*, subtracting a fixed value to the power-hungry red and blue channels.

Figure 1.1 illustrates the effects of this color reduction strategy. The original image has a particularly large power consumption due to its prevalence of red and blue colors. Notice the substantial MSSIM decrease with a color reduction higher than 30%. Indeed, the image starts to become greenish, since the green channel remains unchanged. Reduction rates higher than 50% yield poor-quality images, therefore, they were neglected during the experiments.

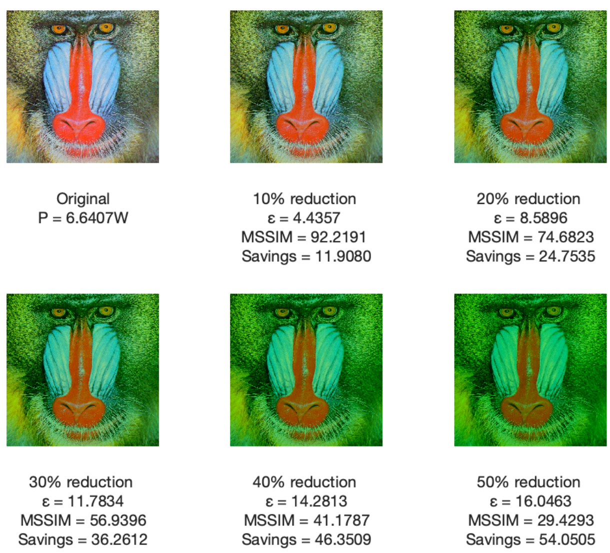
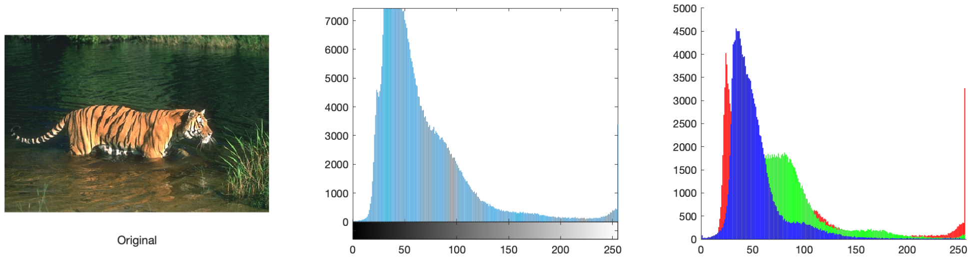


Figure 1.1: Example of color reduction.

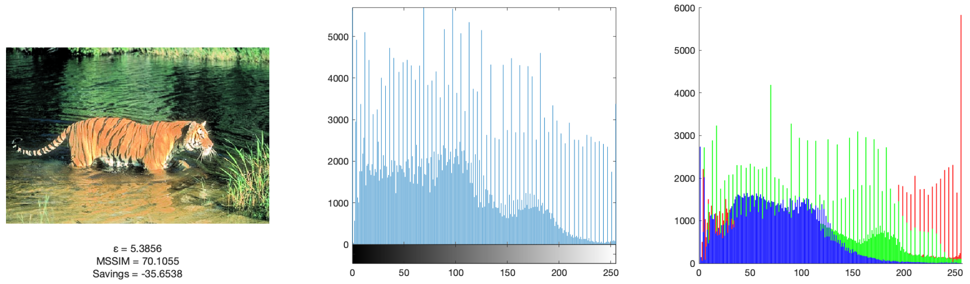
1.2. Histogram Equalization

Histogram equalization is a transformation that adjusts the intensity values. The function *histogram_equalization.m* first converts the image into the HSV color space. The command *histeq*, provided by MATLAB, matches the vector V to a flat histogram with 64 bins by default.

Figure 1.2 portrays the before and after of an image sample that has undergone histogram equalization. Note the increase in brightness and the color redistribution. In fact, the power consumption increased, a negative impact caused by the intensity equalization.



(a) Before



(b) After

Figure 1.2: Histogram equalization.

1.3. Contrast-limited Adaptive Histogram Equalization

The Contrast-limited Adaptive Histogram Equalization (CLAHE) operates on small regions in the image, called tiles, rather than the entire image. Each tile's contrast is enhanced, so that the histogram of the region approximately matches an uniform distribution. The neighboring tiles are then combined using bi-linear interpolation to eliminate artificially induced boundaries.

The script *adaptive_histogram_equalization.m* converts the image into the LAB color space. The L channel must be scaled to the range expected by the *adapthisteq* function, [0 1]. Subsequently, CLAHE is performed over L, and the values are scaled back to the range used by the LAB color space. The new image is then converted back to RGB.

In Fig. 1.3 the same image sample as before has undergone this transformation strategy. In this case, a large difference in contrast can be perceived. Note how smoother the histograms are in comparison with those presented by Fig. 1.2 b).

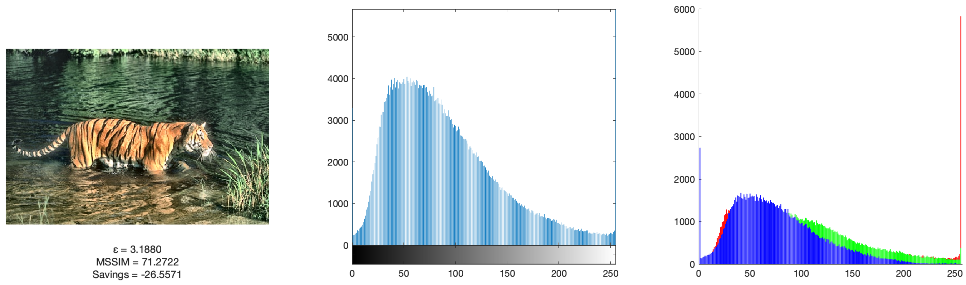


Figure 1.3: Contrast-limited adaptive histogram equalization.

1.4. Results and Discussion

Color reduction has been demonstrated to be an effective strategy for reducing power consumption. Table 1.1 presents the average measurements for each iteration. Although these results vary throughout the entire image-set, negative savings were not observed.

The savings diagram depicted in Fig. 1.4 demonstrates that histogram equalization has been unsuccessful in obtaining any amount of savings in almost half the image-set. In some cases the power consumption drastically increased. Even though the average distortion was slightly better, being 3.9110%, the average saving was −4.3082%.

Color reduction [%]	Savings [%]	Distortion [%]	MSSIM [%]
10	15.36	4.3908	81.1393
20	29.48	8.1371	57.8759
30	40.79	10.8011	41.2836
40	49.98	12.7267	29.6337
50	56.59	13.9472	22.4242

Table 1.1: Color reduction average measurements.

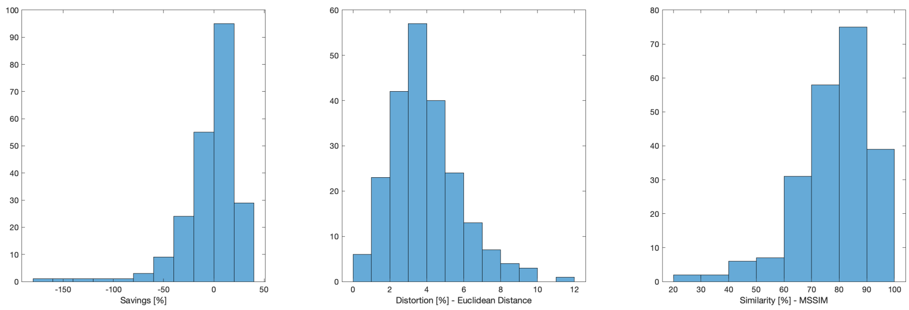


Figure 1.4: Results obtained with conventional histogram equalization.

Finally, the results obtained with the contrast-limited adaptive histogram equalization are shown in Fig. 1.5. The power consumption actually worsen in comparison with the conventional equalization, proving that even an optimized version of this technique is not efficient also. The average saving equal to −9.0236% confirms that, despite displaying an average distortion of 2.7566%, the lowest from all techniques.

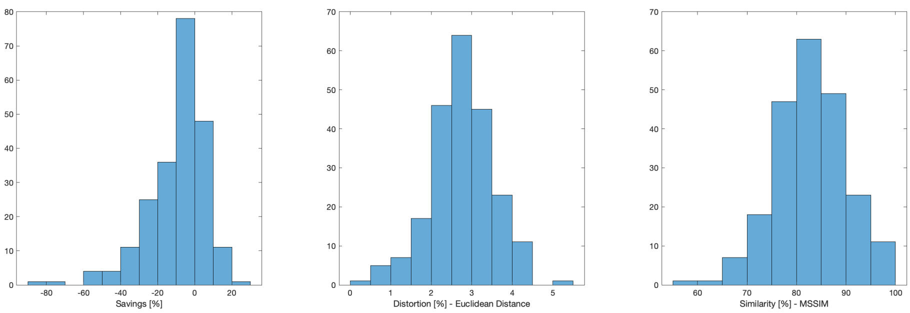


Figure 1.5: Results obtained with contrast-limited adaptive histogram equalization.

2. Dynamic Voltage Scaling

The second part of this lab aims at experimenting with Dynamic Voltage Scaling (DVS) and evaluating software-based compensation techniques. The simulation flow is managed by the script *main2.m*. At each iteration the voltage supply is decreased by 5%, down to 75% of the default V_{DD}. Lower rates were not considered due to their impact on quality.

2.1. Brightness compensation

The function *brightness_compensation.m* takes as input the original voltage supply value as well the new voltage after scaling. For increasing brightness, a coefficient b is added to each pixel intensity value in the HSV color space. Figure 2.1 depicts an image sample after undergoing both brightness compensation and voltage scaling.

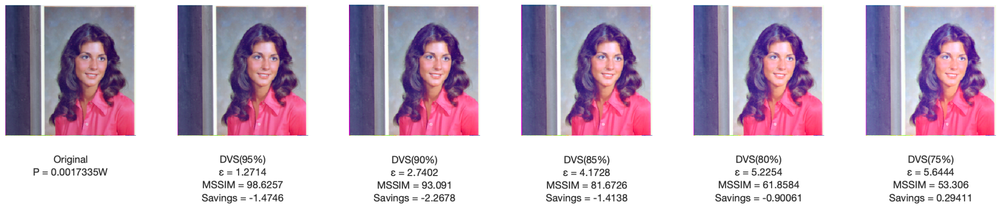


Figure 2.1: Brightness compensation effects.

2.2. Contrast enhancement

The function *contrast_enhancement.m* works similarly to the previous one, except that in this case the intensity value is multiplied by b. Figure 2.2 displays the effects of this strategy, where the image is indeed rather sharper than brighter.

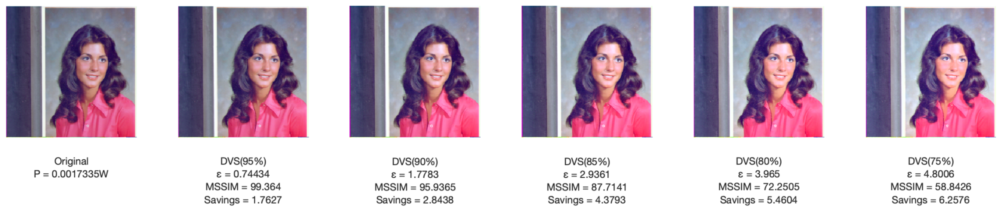


Figure 2.2: Contrast enhancement effects.

2.3. Concurrent brightness and contrast compensation

The function implemented by *concurrent_compensation.m* uses the brightness coefficient gl as a lower boundary, while the contrast coefficient gu serves as the upper boundary. The intensity value is calculated through equation 2.1. Figure 2.3 illustrates the effects of this compensation technique.

$$[!h]t(x) = \begin{cases} 0, & 0 \leq x < gl \\ cx + d, & gl \leq x \leq gu, \text{ where } c = \frac{1}{gu-gl}, \text{ and } d = \frac{-gl}{gu-gl} \\ 1, & x > gu \end{cases} \tag{2.1}$$

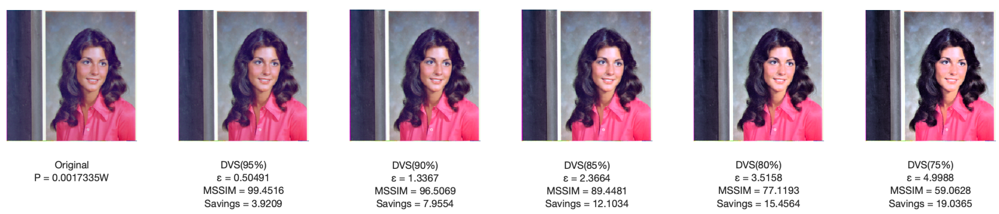
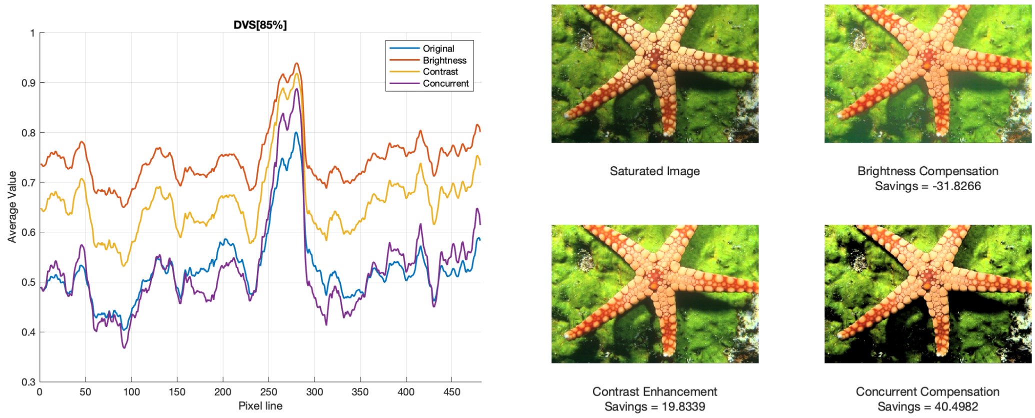


Figure 2.3: Concurrent brightness compensation and contrast enhancement effects.

2.4. Results and Discussion

First, the effects of each compensation strategy on luminosity can be seen in Fig. 2.4. This plot correlates the intensity values for each pixel in the image sample depicted on right. Notice how the concurrent compensation curve almost overlaps the original one.



(a) Luminance **(b)** Image Sample

Figure 2.4: Concurrent brightness compensation and contrast enhancement effects.

The average saving achieved with each technique is presented in table 2.1. As expected from the previous experiments, the brightness compensation actually increased power consumption. Overall, the concurrent enhancement not only improved the image quality after voltage scaling as it saved more energy than the saturated image without any kind of compensation.

DVS [%]	Saturated [%]	Brightness [%]	Contrast [%]	Concurrent [%]
75	28.1271	-17.6505	17.7549	49.0929
80	21.9046	-20.3604	11.7088	40.6037
85	16.1468	-19.1584	7.7741	31.7599
90	10.5752	-15.0558	4.6518	22.3627
95	5.1873	-8.5468	2.1247	12.3448

Table 2.1: Average savings achieved with each compensation technique.

Lastly, table 2.2 displays the average distortion for each transformation. Both contrast enhancement and concurrent compensation presented similar distortion rates. Moreover, none of the strategies exhibited a distortion rate higher than 15% throughout the whole image-set.

DVS [%]	Saturated [%]	Brightness [%]	Contrast [%]	Concurrent [%]
75	2.7715	6.6886	4.7789	5.2299
80	1.7296	5.7018	3.7340	3.8649
85	0.9957	4.4434	2.7120	2.6903
90	0.4892	3.0366	1.7160	1.6529
95	0.1679	1.5001	0.7933	0.7481

Table 2.2: Average distortion resultant of each compensation technique.

Bibliography

[1] Mian Dong, Yung-Seok Kevin Choi, and Lin Zhong. «Power-saving color transformation of mobile graphical user interfaces on OLED-based displays». In: *ISLPED*. 2009.

[2] D. Jahier Pagliari, S. Di Cataldo, E. Patti, A. Macii, E. Macii, and M. Poncino. «Low-Overhead Adaptive Brightness Scaling for Energy Reduction in OLED Displays». In: *IEEE Transactions on Emerging Topics in Computing* (2019), pp. 1–1. DOI: 10.1109/TETC.2019.2908257.

[3] D. Jahier Pagliari, E. Macii, and M. Poncino. «LAPSE: Low-Overhead Adaptive Power Saving and Contrast Enhancement for OLEDs». In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4623–4637. DOI: 10.1109/TIP.2018.2844722.

[4] G. Yadav, S. Maheshwari, and A. Agarwal. «Contrast limited adaptive histogram equalization based enhancement for real time video system». In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2014, pp. 2392–2397. DOI: 10.1109/ICACCI.2014.6968381.

[5] D. Kim, W. Jung, and H. Cha. «Runtime power estimation of mobile AMOLED displays». In: *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2013, pp. 61–64. DOI: 10.7873/DATE.2013.027.

[6] Wei-Chung Cheng and M. Pedram. «Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling». In: *IEEE Transactions on Consumer Electronics* 50.1 (2004), pp. 25–32. DOI: 10.1109/TCE.2004.1277837.