

ECE 285 – Assignment #1

Watermarking

Written by Charles Deledalle on April 15, 2019.

This assignment focuses on Watermarking. Let me introduce you watermarking through the following scenario. Bob is really proud of his pictures and he wants to make sure that nobody is going to claim ownership of them. To do so, he has developed a technique known as watermarking to identify his own pictures. According to Wikipedia: “A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as an audio, video or image data. It is typically used to identify ownership of the copyright of such signal.” Nevertheless, Alice is skeptical about Bob’s technique. She believes that a simple software could be able to remove his watermark. Bob disagrees. In this exercise, we are going to check who is right.

Download `ece285_IVR_assignments.zip`, unzip it in the location of your choice, start a Jupyter Notebook, go into the subdirectory `ece285_IVR_assignments` (rename it if you want), and create a new notebook `assignment1_watermarking.ipynb` and import

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import imageio as im

%matplotlib notebook
```

Note that `imageio` is an homemade package that already contains a few functionalities and that we will embellish in the next assignments.

For the following questions, please write your code and answers directly in your notebook. Organize your notebook with headings, markdown and code cells (following the numbering of the questions).

1 Part I - Bob’s watermarking technique

The simple watermarking technique of Bob can be summarized as follows. Given an image $x \in \mathbb{R}^n$, the watermarked image $y \in \mathbb{R}^n$ will be defined, for all elements k , as

$$y_k = \lambda_k x_k \quad \text{with} \quad \lambda_k = \frac{1}{(1-p)m_k + p} \quad (1)$$

where $p > 0$ and $m \in \mathbb{R}^n$ is a mask (called watermark) such that $0 \leq m_k \leq 1$. We consider that y , x and m are RGB images, thus $n = 3n_1n_2$ and the index k scans each channel of each pixel. See Figure 1.

Here the mask m is supposed to be known by Alice, but Bob’s genius idea is that by hiding the value of p , Alice won’t be able to invert the equation by just doing $x_k = y_k/\lambda_k$.

1. In this assignment, we are going to use the files:

- `assets/img_bob.png`
- `assets/img_alice.png`



(a) Original image x



(b) Mask m



(c) Watermarked image y

Figure 1: Bob's watermarking (here, only m and y are unknown to us)

- `assets/mask.npy`

The two `png` images are RGB images that you can load using the following command

```
xalice = plt.imread('assets/img_alice.png')
ybob = plt.imread('assets/img_bob.png')
```

The mask can be loaded using the command

```
mask = np.load('assets/mask.npy')
```

What are the shapes of these arrays? What are the ranges of their values?

2. Display the array `xalice` as follows

```
plt.figure()
plt.imshow(xalice)
```

Explain what are the 6 buttons appearing below the picture.

3. Display the array `ybob` but by using our homemade function `im.show`. Explain how this function differs from `imshow`.
4. Display the three arrays as follows

```
fig, axes = plt.subplots(ncols=3, figsize=(7, 2))
im.show(xalice, ax=axes[0])
im.show(ybob, ax=axes[1])
im.show(mask, ax=axes[2])
fig.show()
```

Explain what does each command do.

5. Implement equation (1) as a function:

```
def watermark(x, mask, p):
    ...
    return y
```

6. Call the function `watermark` for `xalice` and display the obtained watermarked image `yalice` for $p = .1$. Make sure that your results are consistent with the ones in Figure 2



(a) Original image x



(b) Watermarked image x with $p = 0.1$

Figure 2: Alice's watermarking (here, we know everything: x , m , p and y)

7. Run the following code snippet

```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(7, 3),
                        sharex='all', sharey='all')
for k in range(6):
    im.show(watermark(xalice, mask, p=(k+1)/3), ax=axes[k % 2][int(k / 2)])
fig.show()
```

Zoom and pan axis on one of the images. What do you observe? Explain what does each command do. What is the effect of p ?

2 Part II - Alice's watermarking removal technique

Alice thinks that an estimate \hat{p} of p can be easily made, and subsequently, an estimate \hat{x} of the sought image x can be obtained as

$$\hat{x}_k = [(1 - \hat{p})m_k + \hat{p}]y_k. \quad (2)$$

Her idea is that among all estimates \hat{x}_k obtained for different p , the one corresponding to the sought image x should be the least correlated to the mask m . Alice suggests to use as a measure of correlation between x and m , the following quantity

$$E(p) = \left[\frac{1}{n} \sum_{k=1}^n (m_k - \bar{m})(\hat{x}_k - \bar{\mu}) \right]^2 \quad \text{where} \quad \bar{m} = \frac{1}{n} \sum_{k=1}^n m_k \quad \text{and} \quad \bar{\mu} = \frac{1}{n} \sum_{k=1}^n \hat{x}_k. \quad (3)$$

8. Implement equation (2) as a function:

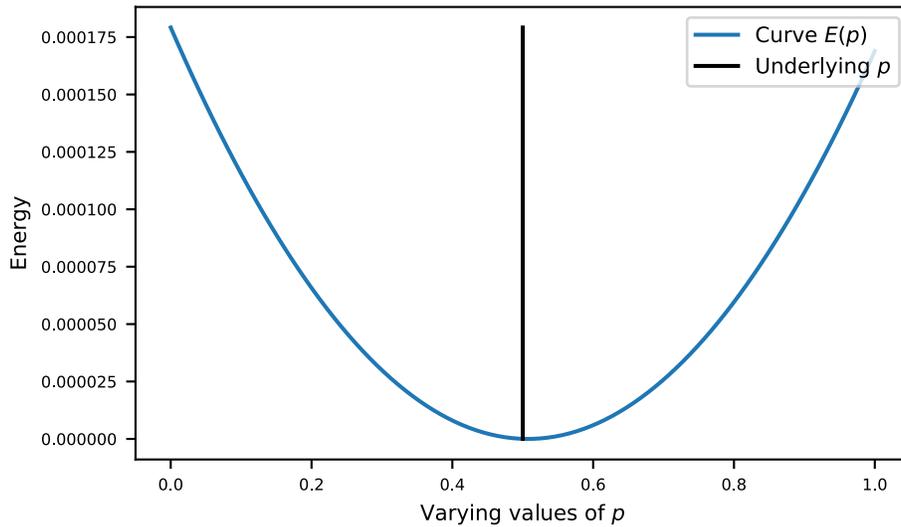
```
def iwatermark(y, m, p):
    ...
    return x
```

9. Write a function that evaluates equation (3):

```
def energy(y, m, p):
    ...
    return e
```

Make sure that if p is a 1d array of size q , then e is a 1d array of size q containing the result of (3) for each value in p .

10. Given `xalice`, call the function `watermark` to create a watermarked image `y` with $p = .5$, and call `energy` for `pctest = np.linspace(0, 2 * p, q)` with $q=100$. Display the curve of energy and check Alice's assumption (don't forget to include a legend and the axis names). Make sure your graph is consistent with the following one:



11. Show that the expression of \hat{p} reads as

$$\hat{p} = \frac{\sum_{k=1}^n \alpha_k y_k}{\sum_{k=1}^n \beta_k y_k} \quad (4)$$

where you have to determine the expression of α_k and β_k .

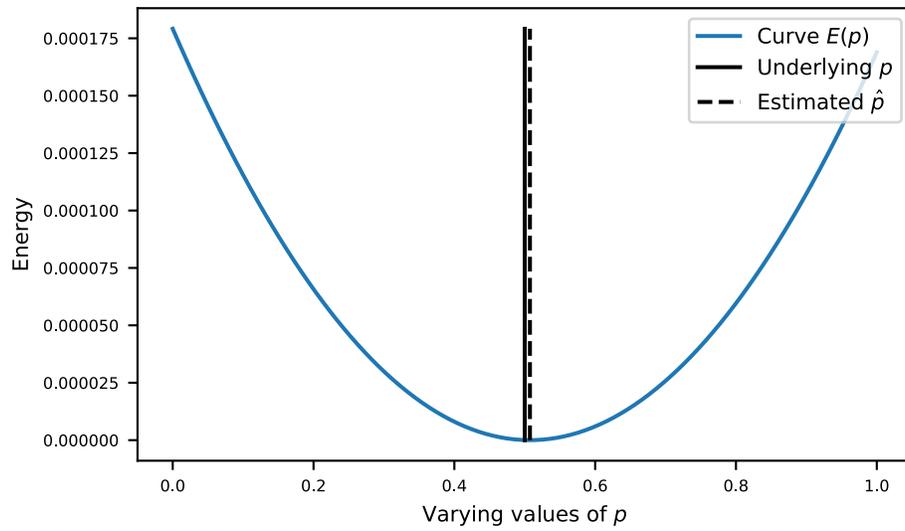
Hint: Show first that $\sum_k (m_k - \bar{m})(\hat{x}_k - \bar{\mu}) = \sum_k (m_k - \bar{m})\hat{x}_k$ and that

$$E(p) = \left[\frac{1}{n} \sum_k m_k (m_k - \bar{m}) y_k + p \frac{1}{n} \sum_k (1 - m_k) (m_k - \bar{m}) y_k \right]^2$$

12. Implement equation (4) as a function:

```
def estimate_p(y, m):
    ...
    return p
```

Plot the result of this function applied to `yalice`. Make sure your graph is consistent with the following one:



13. Write the code to remove the watermark in the image `ybob`. According to Alice, what is the value of p that Bob has been using?
14. Display next to each other `ybob` and your estimation of `xbob`. Link their axis and zoom on the figures.
15. Who is right Alice or Bob?