Homework 2: Image Denoising

Realized by: Eya Ghamgui

Chapter 5: Multi_scale DCT Denoising:

Exercice n 5.1 :

We have :

- n: the dimension of the initial image
- k: the dimension of the image at a lower scale.
- u_0: the initial image
- u_1: the image at the lower scale
- DCT_n^{iso} and DCT_k^{iso} are the isometric DCTs at scale n and k respectivelly.
- ZP_k is the zero padding operator extracting the first $k \times k$ subimage from an $n \times n$ image.

The construction of the image at a lower scale is divided into several steps:

1. The first step consists in transforming the initial image u_0 to the frequency space. Thus, we will apply the function DCT_n^{iso} to the initial image. We obtain: $u' = DCT_n^{iso}(u_0)$
2. To construct the resulted image with dimension $k \times k$, we need to apply the zero padding operator. By definition, this operator extracts a subimage from an $n \times n$ image. We obtain: $u'' = ZP_k(DCT_n^{iso}(u_0))$
3. Before we transform the image to the original image space, we need to adjust the scale. Since we have transformed the image from $n \times n$ size to $k \times k$ size, we have:
   - * Numpixel (initial) is the number of pixels in the original image which is equal to $n \times n$.
   - * Numpix (lower) is the number of pixels of the image at the lower scale which is equal to $k \times k$.

Now, we need to scale the image with a factor equal to:

$$\sqrt{\frac{Nupix(layer)}{Nupix(input)}} = \frac{\sqrt{k \times k}}{\sqrt{n \times n}}.$$

We obtain: $u'' = \frac{\sqrt{k \times k}}{\sqrt{n \times n}} ZP_k\left(DCT_n^{iso}(u_0)\right)$

4- The final step consist in transforming the resulted image to the image space. Thus, we obtain:

$$u_1 = IDCT_k^{iso}\left(\frac{\sqrt{k \times k}}{\sqrt{n \times n}} ZP_k\left(DCT_n^{iso}(u_0)\right)\right)$$

$$\Rightarrow u_1 = \frac{\sqrt{Nupix(layer)}}{\sqrt{Nupix(input)}} IDCT_k^{iso}\left(ZP_k\left(DCT_n^{iso}(u_0)\right)\right)$$

We know that the images at different scale contain a white gaussian noise. Therefore, the image u_1 and u_0 can be written as a sum of a denoised image and a white gaussian noise.

$$\Rightarrow u_1 = u + n_1$$

$$\Rightarrow Var(n) = Var(u_1 - u) \text{; with } u \text{ deterministic}$$

$$\Rightarrow Var(n) = Var(u_1)$$

$$= \frac{Nupix(layer)}{Nupix(input)} \underbrace{Var\left(IDCT_k^{iso}\left(ZP_k\left(DCT_n^{iso}(u_0)\right)\right)\right)}_{= \text{ constant (white noise)}}$$

$$= \frac{Nupix(layer)}{Nupix(input)} \sigma^2(u_0)$$

$$\Rightarrow \sigma(u_1) = \sqrt{\frac{Nupix(layer)}{Nupix(input)}} \sigma(u_0)$$

We can say that the standard deviation of the noise is scaled with a factor equal to: $\sqrt{\frac{Nupix(layer)}{Nupix(input)}}$.

Chapter 6 : Image self-similarity and denoising:

Exercice n 6.1 :

- We want to show:

  " The convolution of the image with a gaussian kernel ensures a fixed noise standard deviation reduction factor"

- We have : * $n(i)$; $i \in \mathbb{Z}$ (not necessarily Gaussian)

  * Discrete finite filter $a(i) \geq 0$ supported in $[-m, m]$ such that $\sum_{j} a(j) = 1$.

- $a * n(i) = \sum_{j} n(i-j)\, a(j)$

$$\mathrm{Var}(a * n(i)) = \mathrm{Var}\left(\sum_{j} n(i-j)\, a(j)\right)$$

$$= \sum_{j} \mathrm{Var}\big(a(j)\, n(i-j)\big) + 2\sum_{k<l} \mathrm{cov}\big(a(k)n(i-k), a(l)n(i-l)\big)$$

$$= \sum_{j=-m}^{m} \mathrm{Var}\big(a(j)\, n(i-j)\big) + 2\sum_{k<l} \mathrm{cov}\big(a(k)\, n(i-k), a(l)\, n(i-l)\big)$$

or : $n(i)$ is a white noise. Thus, its term are independent. It has a null mean and a constant variance.

Therefore,
$$\begin{cases} \mathrm{Var}\big(a(j) \cdot n(i-j)\big) = a^2(j)\, \mathrm{var}\big(n(i-j)\big) = \sigma^2 a^2(j) \\ \\ \mathrm{Cov}\big(a(k), n(i-k), a(l)\, n(i-l)\big) = 0 \end{cases}$$

$$\Rightarrow \boxed{\mathrm{Var}(a * n(i)) = \sum_{j=-m}^{m} a^2(j)\, \sigma^2}$$

- We have $a(j) \geq 0$; $\sum_{j=-m}^{m} a(j) = 1$

$$\Rightarrow a(j) \leq 1 \ \forall j.$$

$$\Rightarrow a^2(j) \leq a(j)$$

$$\Rightarrow \sigma^2 a^2(j) \leq \sigma^2 a(j)$$

$$\Rightarrow \quad \sigma^2 \sum_{j=-m}^{m} a^2(j) \leqslant \sigma^2 \sum_{j=-m}^{m} a(j)$$

$$\Rightarrow \quad \sum_{j=-m}^{m} \sigma^2 a^2(j) \leqslant \sigma^2$$

$$\Rightarrow \quad \boxed{Var\left(a * n(i)\right) \leqslant Var(n(i))}$$

- Now, we want to determine the optimal values of $a(i)$.

Let's define the problem:

$$\begin{cases} \text{Min } Var(a * n(i)) \\ \text{st} \\ \displaystyle\sum_{j=-m}^{m} a(j) = 1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \text{Min } \displaystyle\sum_{j=-m}^{m} \sigma^2 a^2(j) \\ \text{st} \\ \displaystyle\sum_{j=-m}^{m} a(j) = 1 \end{cases} \Leftrightarrow \begin{cases} \text{Min } \displaystyle\sum_{j=-m}^{m} a^2(j) \\ \text{st} \\ \displaystyle\sum_{j=-m}^{m} a(j) = 1 \end{cases}$$

To solve this problem, we need to write the lagrangian function:

$$\mathcal{L} = \sum_{j=-m}^{m} a^2(j) + \vartheta\left(1 - \sum_{j=-m}^{m} a(j)\right)$$

Using the KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial a(j)} = 0 \Leftrightarrow 2\,a(j) - \vartheta = 0 \quad \forall j$$

$$\Leftrightarrow a(j) = \vartheta/2 = \underset{= c}{cte} \quad \forall j \text{ and } a(j) \geqslant 0$$

We have $\displaystyle\sum_{j=-m}^{m} a(j) = 1$

$$\Leftrightarrow \sum_{j=-m}^{m} c = 1 \Leftrightarrow c \times (2m+1) = 1$$

$$\Leftrightarrow c = \frac{1}{2m+1} \geqslant 0$$

Conclusion: $\boxed{a(j) = \dfrac{1}{2m+1} \quad \forall j}$