Mathematical methods for Image Processing

Hands-on session
4

Dequantizing by H^1 regularization

During this Hands-on sesssion, we consider an optimization method to de-quantize images. Let $\tau > 0$, we consider the quantization of an image $u \in I\!R^{N^2}$ as defined by an image $v \in I\!R^{N^2}$, whose coordinates are :

$$v_{i,j} = q_\tau(u_{i,j}), \qquad \forall (i,j) \in \{0,\dots,N-1\}^2$$

where for all $t \in I\!R$,

$$q_\tau(t) = \tau \left[\frac{t}{\tau} + \frac{1}{2} \right]$$

where $[x]$ represents integer floor function applied to x (i.e. : the largest integer small than x). We denote $Q : I\!R^{N^2} \longrightarrow I\!R^{N^2}$, the (non-linear) quantization operator and write

$$v = Q(u).$$

During the process, we have lost many gray levels and the result v contains large areas with a constant value.

As in the preceding Hands-on session, we denote

$$\mathcal{C} = \{w \in I\!R^{N^2}, Q(w) = v\}$$

To restore (or simply improve the visual aspect) of v, we propose to solve the following optimization problem

$$(P) : \begin{cases} \text{minimize } E(w) \\ \text{under the constraint } w \in \overline{\mathcal{C}}, \end{cases}$$

for a well chosen regularization criterion E (n.b. $\overline{\mathcal{C}}$ is the closure of $\mathcal{C}$).

In this Hands-on session, we consider the regularization criterion

$$E(w) = \sum_{i,j=0}^{N-1} |\nabla w_{i,j}|^2$$

for $w \in I\!R^{N^2}$, with

$$\nabla w_{i,j} = \begin{pmatrix} (D_x w)_{i,j} \\ (D_y w)_{i,j} \end{pmatrix} = \begin{pmatrix} w_{i+1,j} - w_{i,j} \\ w_{i,j+1} - w_{i,j} \end{pmatrix}$$

for all $(i,j) \in \{0,\dots,N-1\}^2$ (we assume that w is periodized outside $\{0,\dots,N-1\}^2$).

Exercice 1. *(1)*

- *Go to the web-page*
  http ://www.math.univ-toulouse.fr/~fmalgouy/index.html
  and navigate to reach the course web-page.
- *Download and unzip the archive* tp2.zip.

The program deQuantifieImage *approximates a solution of* (P)*, by a penalization method. More precisely, it minimizes the cost function*

$$F_\lambda(w) = E(w) + \lambda \sum_{i,j=0}^{N-1} \varphi_\tau\left(w_{i,j} - v_{i,j}\right),$$

where, for all $t \in I\!R$,

$$\varphi_\tau(t) = \left(sup\left(|t| - \frac{\tau}{2}, 0 \right) \right)^2.$$

(2) *Calculation concerning the cost function F_λ.*

  (a) *Calculate the derivatives of $\varphi_\tau(t)$. (You may distinguish the three cases : $t \leq -\frac{\tau}{2}$, $-\frac{\tau}{2} < t < \frac{\tau}{2}$ and $\frac{\tau}{2} \leq t$.)*

  (b) *Deduce from the previous question and the Hands-on session 1-2, the gradient of F_λ.*

  (c) *Make precise connections between the gradient algorithm with Armijo step-size rule for minimizing F_λ and the programm* deQuantifieImage*.*

(3) *Launch the script for different values of λ. For every value of λ, you will adjust the number of iterations required by the algorithm to converge. Comment on the results obtained in terms of : -image properties ; -the convergence of the optimization algorithm.*

Exercice 2. *When considering (P), it is possible to compute the projection onto $\overline{\mathcal{C}}$. We can therefore implement a projected gradient algorithm (i.e. a proximal gradient algorithm).*

(1) *Let $w \in \mathbb{R}^{N^2}$. What are the coordinates of $\Pi(w)$ (the projection of w onto $\overline{\mathcal{C}}$). (Begin with a proof arguing that we can compute N^2 projections : We project independently every $w_{i,j}$ onto the interval $[v_{i,j} - \frac{\tau}{2}, v_{i,j} + \frac{\tau}{2}]$.)*

(2) *Copy* deQuantifieImage.sci *in a file* deQuantifieImage1.sci *and modify it to obtain a projected gradient algorithm whose result approximates a solution of (P).*

(3) *Compare the results of* deQuantifieImage.sci *and* deQuantifieImage1.sci *for different values of λ.*