

Thierry CANTENOT

REPORT

Digital Image Processing

« Assignments »



2014-2015

Summary

A	Histogram Equalization	1
A.1	Problem statement	1
A.2	Python implementation	1
A.3	Figure 1	1
A.3.1	Histogram	1
A.3.2	Histogram equalization	1
A.4	Figure 2	1
A.4.1	Histogram	1
A.4.2	Histogram equalization	1
B	Spatial enhancement methods	6
B.1	Problem statement	6
B.2	Python implementation	6
B.3	Results	6
B.3.1	Original image	6
B.3.2	3x3 Laplacian ($A = 0$)	6
B.3.3	3x3 Laplacian ($A = 1$)	6
B.3.4	3x3 Laplacian ($A = 1.7$)	6

A. Histogram Equalization

A.1 Problem statement

1. Write a computer program for computing the histogram of an image.
2. Implement the histogram equalization technique.
3. Your program must be general to allow any gray-level image as its input.

A.2 Python implementation

Usage : `python problem1.py [-h] image_path`

A.3 Figure 1

A.3.1 Histogram

Original image : [A.1](#) | Original image's histogram : [A.2](#)

A.3.2 Histogram equalization

Enhanced image : [A.3](#) | Enhanced image's histogram : [A.4](#)

A.4 Figure 2

A.4.1 Histogram

Original image : [A.5](#) | Original image's histogram : [A.6](#)

A.4.2 Histogram equalization

Enhanced image : [A.7](#) | Enhanced image's histogram : [A.8](#)

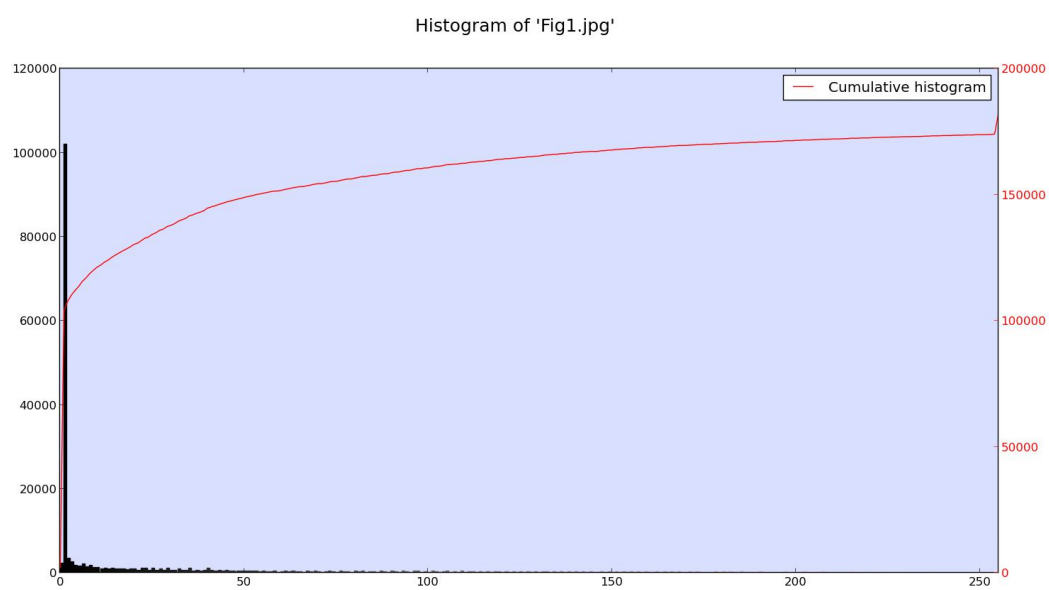
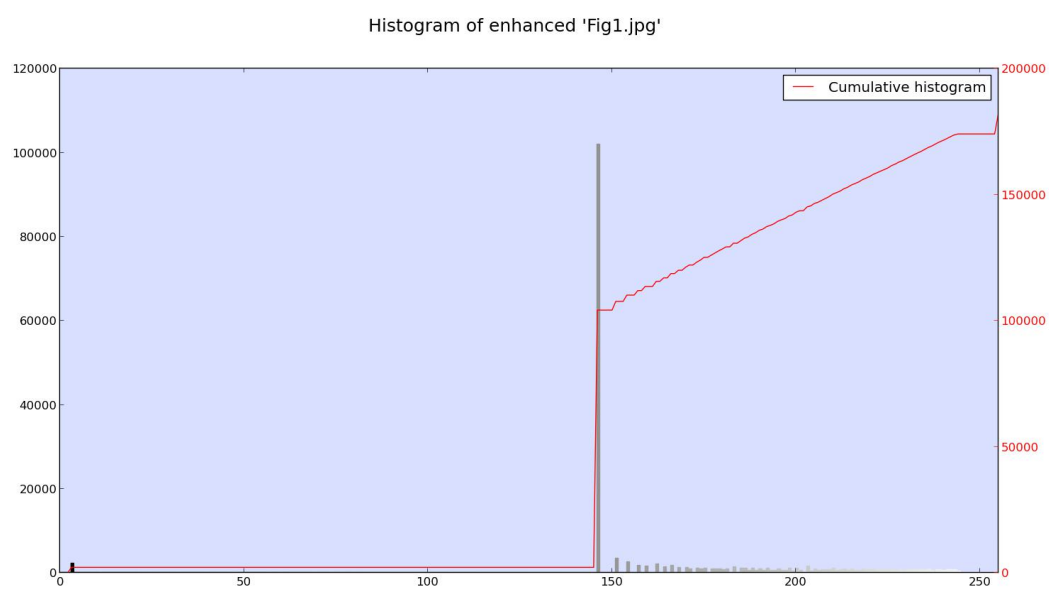
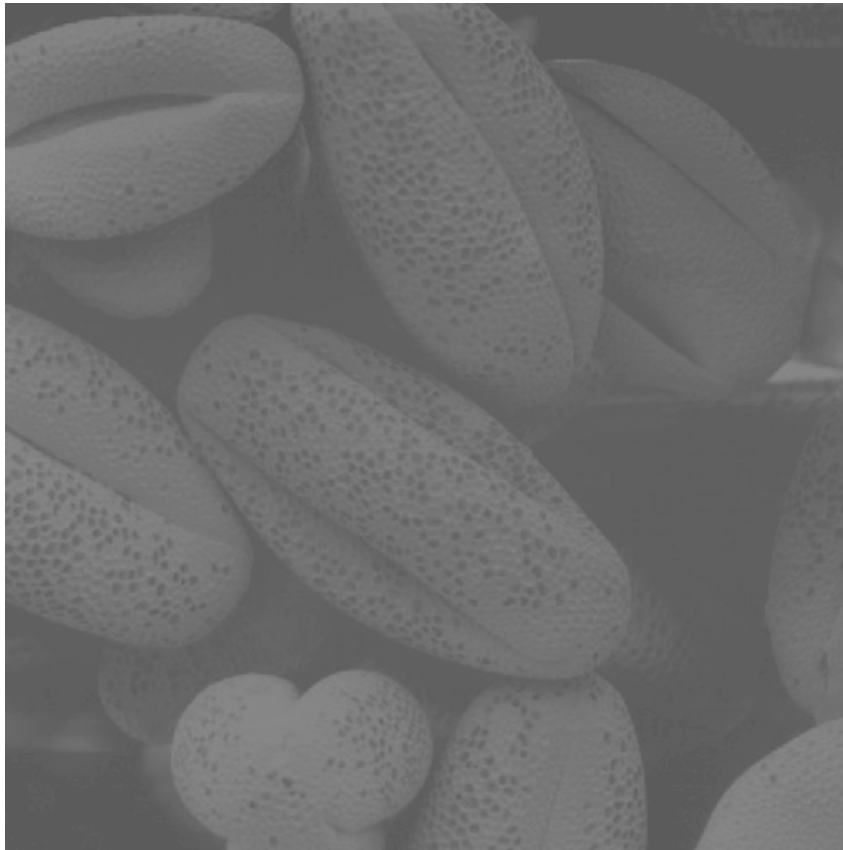
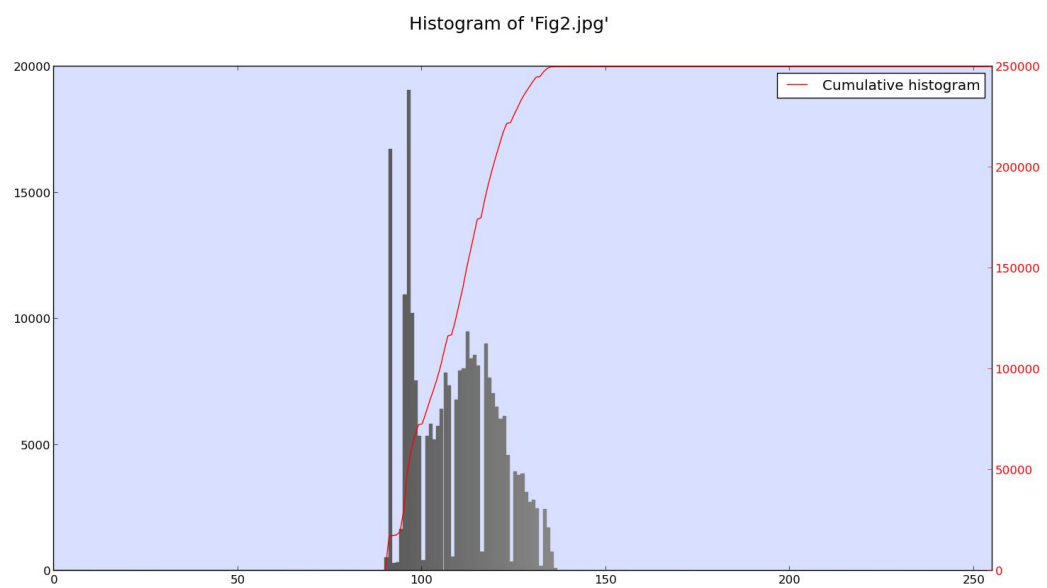
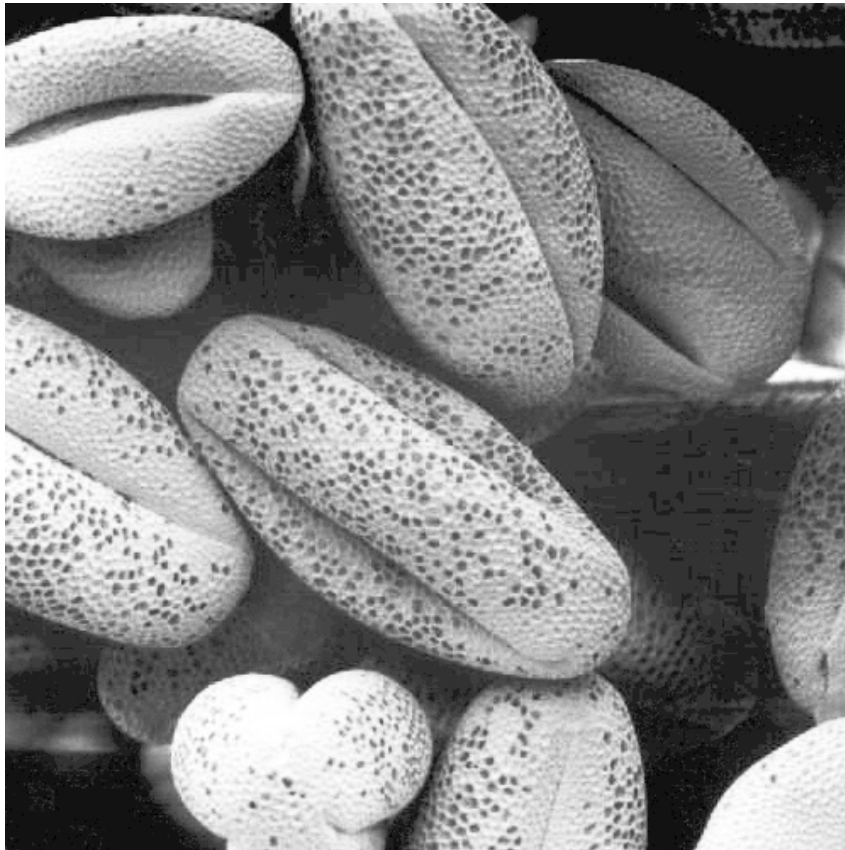
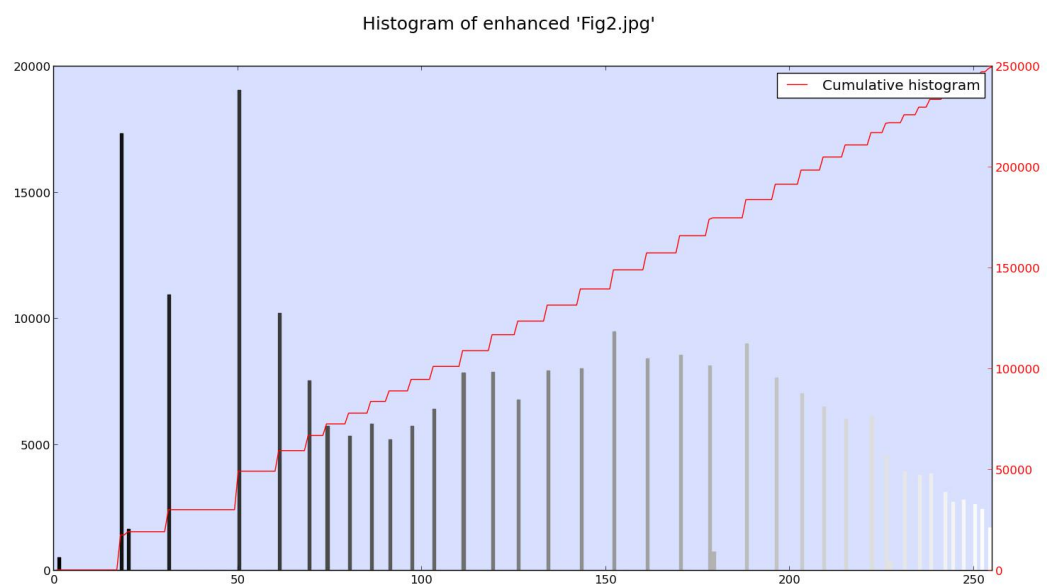
FIGURE A.1 – Original *Fig1.jpg*FIGURE A.2 – Histogram of *Fig1.jpg*

FIGURE A.3 – Enhanced *Fig1.jpg*FIGURE A.4 – Equalized histogram of *Fig1.jpg*

FIGURE A.5 – Original *Fig2.jpg*FIGURE A.6 – Histogram of *Fig2.jpg*

FIGURE A.7 – Enhanced *Fig2.jpg*FIGURE A.8 – Equalized histogram of *Fig2.jpg*

B. Spatial enhancement methods

B.1 Problem statement

Implement the image enhancement task of Section 3.7 (Fig 3.43) (Section 3.8, Fig 3.46 in our slides).

The image to be enhanced is *skeleton_orig.tif*.

You should implement all steps in Figure 3.43.

B.2 Python implementation

Usage : `python problem2.py [-h] [-laplacian] [-sobel] [-a A] image_path`

For example, to use a 3x3 Laplacian filter with $A = 1.7$, and then a Sobel, type :
`python problem2.py -laplacian -a 1.7 -sobel skeleton_orig.tif`

B.3 Results

B.3.1 Original image

Original image : [B.1](#)

B.3.2 3x3 Laplacian ($A = 0$)

Original image's laplacian : [B.2](#) | Original image's sharpened laplacian : [B.3](#)

B.3.3 3x3 Laplacian ($A = 1$)

Original image's laplacian : [B.5](#) | Original image's sharpened laplacian : [B.6](#)

B.3.4 3x3 Laplacian ($A = 1.7$)

Original image's laplacian : [B.8](#) | Original image's sharpened laplacian : [B.9](#)

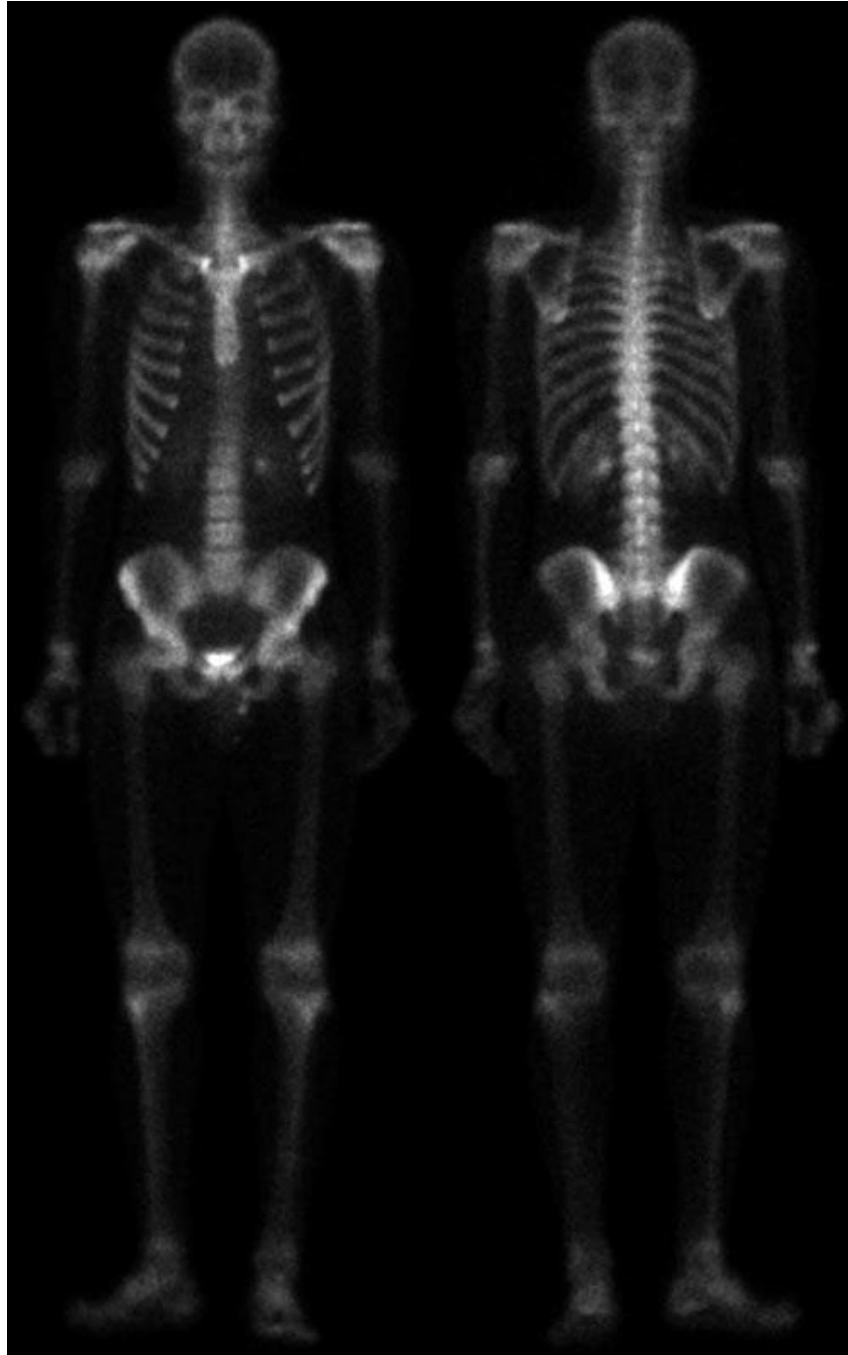


FIGURE B.1 – Original *skeleton_orig.tif*

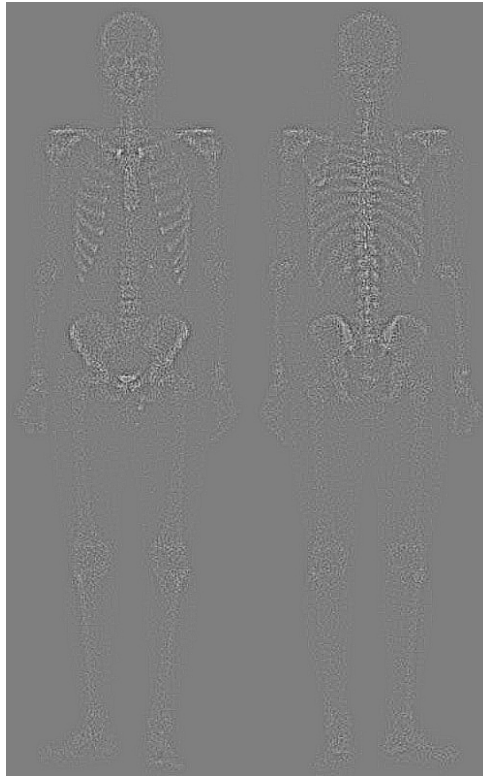


FIGURE B.2 – Laplacian ($A=0$)

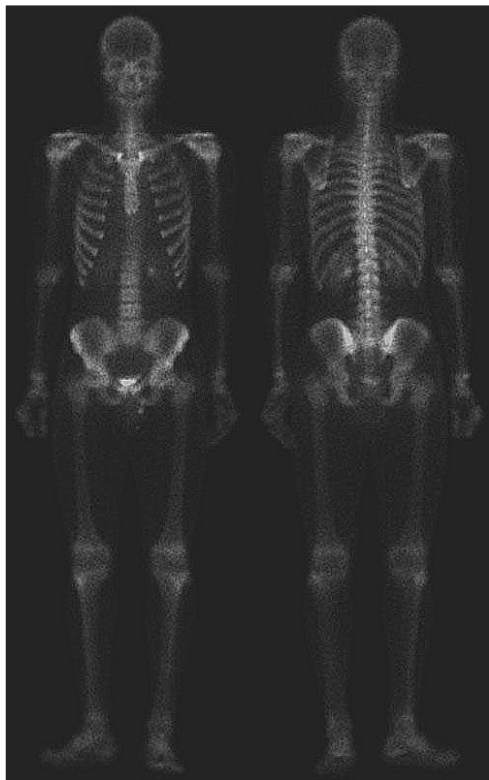


FIGURE B.3 – Sharpened image

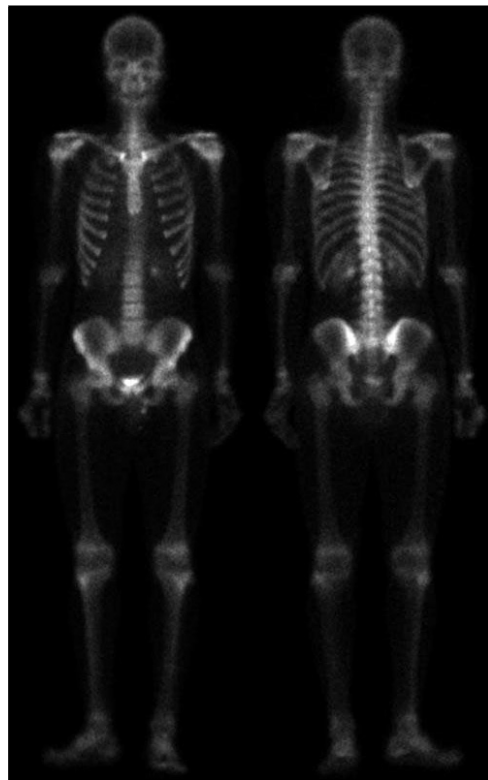


FIGURE B.4 – Original image



FIGURE B.5 – Laplacian ($A=1$)

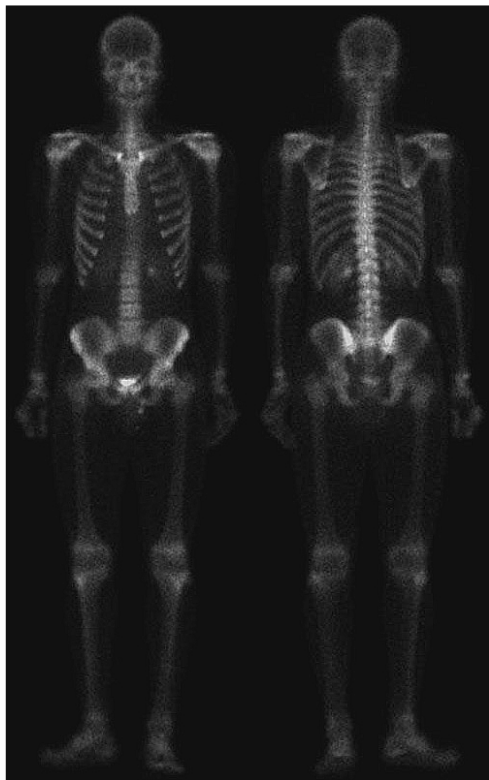


FIGURE B.6 – Sharpened image

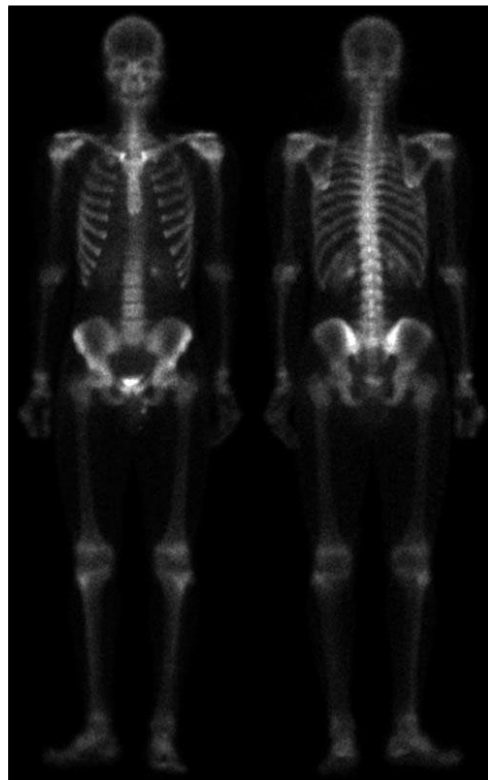


FIGURE B.7 – Original image

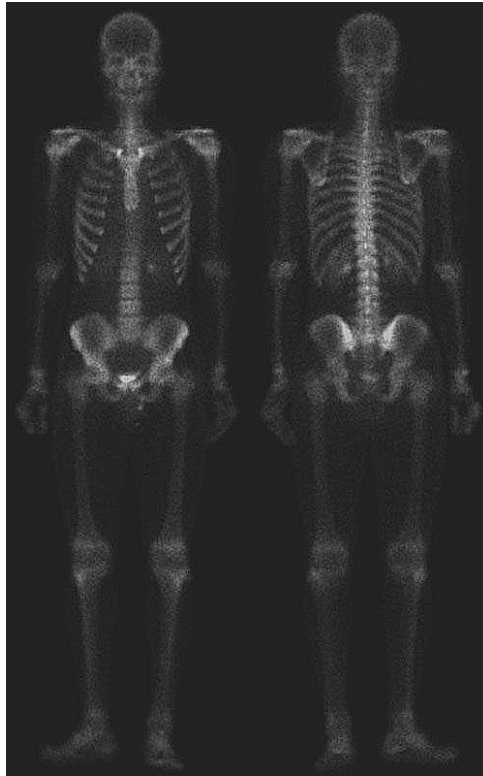
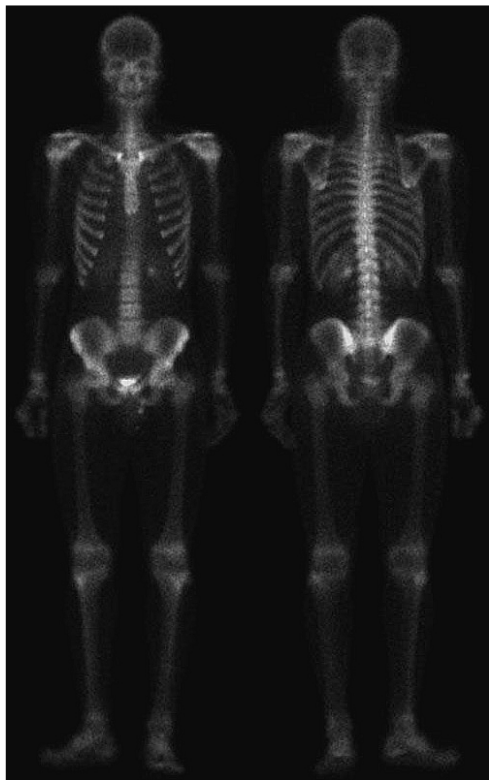
FIGURE B.8 – Laplacian ($A=1.7$)

FIGURE B.9 – Sharpened image

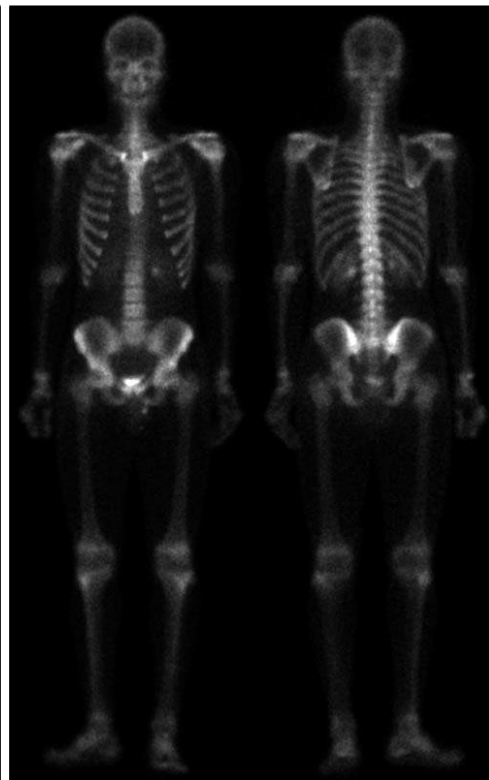


FIGURE B.10 – Original image