INTRODUCTION TO DIGITAL IMAGE PROCESSING

PRACTICE PROBLEMS –ALGEBRAIC OPERATIONS

OBJECTIVES

1. To derive the matrix equation for least-squares fitting of a polynomial surface to the non-uniform background of an image for purposes of background removal.
2. To explore averaging for noise reduction.

PROBLEMS

1. Using the least-squares approach described in class, derive the matrix equation for fitting the surface:

$$B(x,y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy \tag{1}$$

to the background of an image, $I(x, y)$, assuming that the background has been sampled at N pixels with coordinates (x_i, y_i) and gray-level values, $I(x_i, y_i)$. a_i, $i = 0, 1,\ldots,5$, are coefficients that need to be determined. Your answer should be expressed in the form:

$$[C]\{a\} = \{k\}, \quad \{a\} = \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix}.$$

Express the matrix $[C]$ and the vector $\{k\}$ in terms of the following quantities:

$$\begin{array}{cccc}
S_x = \sum_{i=1}^{N} x_i & S_{x2} = \sum_{i=1}^{N} x_i^2 & S_{x3} = \sum_{i=1}^{N} x_i^3 & S_{x4} = \sum_{i=1}^{N} x_i^4 \\
S_y = \sum_{i=1}^{N} y_i & S_{y2} = \sum_{i=1}^{N} y_i^2 & S_{y3} = \sum_{i=1}^{N} y_i^3 & S_{y4} = \sum_{i=1}^{N} y_i^4 \\
S_{xy} = \sum_{i=1}^{N} x_i y_i & S_{x2y} = \sum_{i=1}^{N} x_i^2 y_i & S_{xy2} = \sum_{i=1}^{N} x_i y_i^2 & S_{x2y2} = \sum_{i=1}^{N} x_i^2 y_i^2 \\
S_{x3y} = \sum_{i=1}^{N} x_i^3 y_i & S_{xy3} = \sum_{i=1}^{N} x_i y_i^3 & S_I = \sum_{i=1}^{N} I(x_i, y_i) & S_{xI} = \sum_{i=1}^{N} x_i I(x_i, y_i) \\
S_{yI} = \sum_{i=1}^{N} y_i I(x_i, y_i) & S_{x2I} = \sum_{i=1}^{N} x_i^2 I(x_i, y_i) & S_{y2I} = \sum_{i=1}^{N} y_i^2 I(x_i, y_i) & S_{xyI} = \sum_{i=1}^{N} x_i y_i I(x_i, y_i)
\end{array}$$

Explain all steps in your derivation, e.g., state the cost function you are using.

2. Write a MATLAB function that reads in an image stored in the file with name `‘fname’` and generates `M` noisy images and averages them together. The function should display one of the noisy images along with the averaged image. Try different values of `M`, say `M = 4, 9, 16`. The function header should be:

```
function imAve(fname, M).
```

NOTE: You should convert the images to type `double` and add them together using the + operator. If you use the `imadd` function or if you use the + operator without conversion from type `uint8` to type `double`, you will have the problem that the function also clips the result of each addition to the range 0…255. The + operator does not do this after conversion to type `double`.

You will need to learn about the `imnoise` function in MATLAB. This allows us to simulate a noisy image acquisition process without having to go into the field and acquire the images ourselves.