Digital Image Processing Exercises in the Fourier Domain using VECLIB on Vector Computer CONVEX 3820

SLIDES

Kandó Kálmán Polytechnic
Institute of Computer Technology, Székesfehérvár
and
University of Athens

Magda Kovács, Senior Lecturer
Ágnes Burián, Senior Lecturer

Fourier Domain Operations

Background

Digital Image Transform

Represented by the transform matrices A:

$$X = A\,x$$

where x, X is the original and the transformed image respectively.

In most cases, the transform matrices are unitary:

$$A^{*} = A$$

Commonly used tranforms:

Discrete Fourier Transform (DFT)
Discrete Cosine Transform (DCT)
Walsh transform
Haar transform
Slant transform
Discrete Sine Transform (DST)

Noise generators for digital image processing

Digital images are corrupted by noise either during image acquisition or during image transmission.

The image acquisition noise is photoelectronic noise or film-grain noise. It can be proven that in both cases the noise is signal-dependent.

Digital image noise is removed by using noise filtering techniques. In many cases, digital images must be corrupted artifically in order to assess the performance of various filtering techniques.

Such noise generators that can be produce additive noise or multiplicative noise:

$$g(i,j) = f(i,j) + n(i,j)$$
$$g(i,j) = f(i,j)\, n(i,j)$$

or impulsive noise:

$$g(i,j) = \begin{cases} z(i,j), & \text{with probability } p \\ f(i,j), & \text{with probability } 1-p \end{cases}$$

Digital image filtering

Digital enhancement and restoration operations can be thought as two-dimensional digital filters. They can be distinguished in two large classes:

linear digital filters
nonlinear digital filters.

Linear digital image filter can be designed and / or implemented either in the spatial domain (spatial operation) or in the frequency domain (transform operations.

Inverse filter implementation

The aim of digital image restoration is the recovery of an image that has been degraded by the sensing instruments.

Degradation sources usually produce motion blur, defocus blur and/or noise introduced during sensing.

Such a model is described by the following equation:

g (n1,n2) = f (n1,n2) ** h (n1,n2) + n (n1,n2)

where
f is the original image,
g is the observed image ,
h is the blur point spread function
n is the sensing noise

The problem of image restoration is to estimate f from the observed image g, possibly based on some a priori knowledge about the blur point spread function and on the statistics of the noise n and of the images g, f.

Let us suppose that the formation noise n is negligible. The image formation model takes the following format in the frequency domain::

$$G(\omega 1, \omega 2) = F(\omega 1, \omega 2)\, H(\omega 1, \omega 2)$$

If the degradation function H ((ω1, ω2) is a priori known, the simplest approach to digital image restoration is inverse filtering:

$$F(\omega 1, \omega 2) = G(\omega 1, \omega 2) / H(\omega 1, \omega 2)$$

The most straightforward implementation of the inverse or the pseudoinverse filter is by using the Discrete Fourier Transform:

$$f(n1,n2) = \mathrm{IDFT}\,[G(k1,k2)/ H(k1, k2)$$

Where G(k1,k2) is the DFT of the observed image g(n1,n2) and H(k1,k2) is the DFT of the blur point spread function h(n1,n2).

CONVEX VECLIB

CONVEX VECLIB (and VECLIB8) is a collection of FORTRAN-callable subprograms optimized for use on the CONVEX family of supercomputers. This library provides mathematical software and computational kernels for application programs involving arrays.

These libraries contain subprograms for:

- dense vector operations, including the Basic Linear Algebra Subprograms(BLAS)
  sparse vector operations, including the Sparse Blas
- matrix operations, including the Level 2 and Level 3 BLAS
- linear eqation solution, including LINPACK
- eigensystem solution, including EISPACK
- sparse symmetric linear equation solutions
- sparse symmetric ordinary and generalized eigensystem solutions
- skyline linear eqations
- discrete Fourier transforms
- convolution and correlation
- linear recurrences
- miscellaneous tasks, such as manipulating dynamic memory, sorting and generatingrandom numbers

A VECLIB and VECLIB8 libraries are compiled subprograms ready for you to incorprate into your programs with the linker. Simply include the appropriate declaration and CALL statements in your FORTRAN source program and specify that VECLIB or VECLIB8 be used as an object library at link time, by using the –l option on the fc command line, as follows:

fc [options] file –lveclib

or

fc [options] file –lveclib8

```
=================================================================
                    Loop Performance Analysis
                       For: convolutions
=================================================================
Summary:
                    Times     Iteration Count        CPU Time
 Line                 Exec    Min     Max     Avg   (plus inner loops) PS
-------              -------- ------- ------- ------- ------------------ --
  131                    1    256     256     256          0.006755
  132                    2    256     256     256          0.006748

Details:
          Optimization  Times     Iteration Count        CPU Time
 Line    NL    Report    Exec    Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
  131    0 SM,Hs            1    256     256     256       0.000007
  132    1 S               2    256     256     256          0.006748

             Static Profile              Estimated  Mflops
          Vector  Vector  Chime  (less inner loops)  (plus inner loops)
 Line    NL Spills  Flops   Count     Avg       Peak      Avg       Peak   PS
------- --- ------ ------- ------- --------- --------- --------- --------- --
  131    0      0       0       0    0.000     0.000     0.455    60.000
  132    1      0       6       6    0.455    60.000     0.455    60.000
```

Compiling command: fc -O2 -lveclib -pa

```
=====================================================================
                    Loop Performance Analysis
                       For: convolutions
=====================================================================
Summary:
                    Times     Iteration Count        CPU Time
 Line                Exec    Min     Max     Avg   (plus inner loops) PS
-------             -------- ------- ------- ------- ------------------ --
   131                 256     256     256     256          0.006904
   132                   1     256     256     256          0.007929

Details:
          Optimization  Times     Iteration Count        CPU Time
 Line    NL    Report    Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
   132    0 P                1     256     256     256          0.001025
   131    1 SM,I           256     256     256     256          0.006904

             Static Profile              Estimated  Mflops
          Vector  Vector  Chime  (less inner loops)  (plus inner loops)
 Line    NL Spills  Flops   Count     Avg       Peak      Avg       Peak   PS
------- --- ------ ------- ------- --------- --------- --------- --------- --
   132    0      0       0       0     0.000     0.000    49.592    60.000
   131    1      0       6       6    56.955    60.000    56.955    60.000

=====================================================================
                 Parallel Region Performance Analysis
                       For: convolutions
=====================================================================
Summary:
         Times          All Regions          Process Virtual
 Line     Exec       CPU Time   Wall Clock Time    Time (PVT)     CPU/PVT  PS
-------  --------   ------------ --------------- ---------------  --------- --
   132         1      0.007929       0.016354       0.010977      0.722

Details:
                       By Region
 Line              CPU Time   Wall Clock Time    Chore Count            PS
-------           ------------ ---------------   -------------          __
   132               0.005620       0.016246            185
                    0.002309       0.008744             71

=====================================================================
                   Basic Block Performance Analysis
                 No data was gathered for basic block(s)
=====================================================================
```

Compiling command: fc -O3 -lveclib -pa