Digital Image Processing Exercises in the Spatial Domain on Vector Computer CONVEX 3820

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

Dr. Rezső Nagy, Associate Professor
Etelka Kovács, Senior Lecturer

Introduction

The image processing algorithms in the three exercises serve as examples for understanding vectorization and parallelization as well as optimization. The exercises involve simple algorithms working in the spatial domain.

The sample programs in these exercises are nearly as simple as possible. They work with simple, program-generated 'pictures', i.e. matrixes of pixel values.The intention is that they should be easy to understand and the students could modify them several times during the laboratory class of about two hours.

Image Representation in the Programs

The digital image can be conveniently represented by an N*M matrix I of the form:

```
          i (1,1)  i(1,2)   ...      i (1,M)
          i (2,1)  i(2,2)   ...      i (2,M)
i =         .
            .
            .
          i (N,1) i(N,2)   ...      i (N,M)
```

The matrix elements (image pixels) are integers in the range [0,...,255] for 8 bit images. Therefore, they can be represented as characters in FORTRAN language. In certain cases, when complex arithmetic is involved, image pixels must be represented as complex numbers in FORTRAN

```
character a (N,M)
complex   b (N,M)
```

The matrix elements have the form a (i,j), b (i,j), 1 <= i <= N, 1 <= i <= M.

Excercise 1.

Spatial Convolution Filter

Background

This technique is based on spatial operations performed on local neighbourhoods of input pixels. The image is convolved with a finite impulse response filter called *spatial mask*.
Each pixel is replaced by a weighted average of its neighbourhood pixels, that is,

$$v(m,n) = \sum_{(k,l)\in W} \sum a(k,l) y(m-k, n-l)$$

where y(m,n) and v(m,n) are the input and output images, respectively, W is a suitably chosen window, and a(k,l) are the filter weights.

Depending on the filter matrix a(k,l) spatial convolution can perform different tasks, e.g.:

smoothing filter, e.g. $\begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{2}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}$

sharpening filter, e.g. $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Laplace operator, e.g. $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

gradient operator, e.g. $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Exercise 1.1. Study and comment the following program implementing a simplified case of smoothing filter:

```
      program spconvol
      parameter (m=8, n=8, fd=3, gr=64)
      integer imo(m,n),imn(m,n),his(gr), fmx(fd,fd)
      integer d,i,j,k,l
      real work
c
      do j=1,n
       do i=1,m
     imo(i,j)=i
     imn(i,j)=i
       end do
      end do
      imo(3,3)=64
```

```
      imn(3,3)=64
c
      do j=1,fd
       do i=1,fd
     fmx(i,j)=1
       end do
      end do
      fmx(2,2)=2
c
      d=(fd-1)/2
c
      do l=2,n-1
     do k=2,m-1
c
       work=0
       do i=-d,d
         do j=-d,d
          work=work+imo(k+j,l+i)*fmx(j+d+1,i+d+1)
         end do
       end do
       imn(k,l)=int(work/10)
     end do
      end do
c
      do i=1,gr
       his(i)=0
      end do
      do j=1,m
       do i=1,n
     work=imn(i,j)
     his(work)=his(work)+1
       end do
      end do
c
      do i=1,n
       print 100,(imo(i,j), j=1,m)
      end do
      print *
      do i=1,n
       print 100,(imn(i,j), j=1,m)
      end do
c      read '(I0)',work
      print *
      print 100,(his(i),i=1,gr)
100   format (1X, 16I4)
      stop
      end
```

Explanation of the program:
The program begins with the picture generation (lines 7 to 14), then it follows the generation of the filter matrix. The two nested loops using the iteration variables l and k step through the rows and columns of the picture. For sake of simplicity, the filter does not affect the border pixels of the picture.
The two nested loops beginning in lines 28 and 29 perform the calculation of the new pixel value, which will be put to its place in line 33.
The next 9 lines of the program computes the histogram of the new picture.
From line 47 the program prints the pixel values of the original picture, the new picture and the histogram values of the new picture.

Exercise 1.2. Compile and run the program! Analyse the result.

You will get the following result when running the program:

```
1    1    1    1    1    1    1    1
2    2    2    2    2    2    2    2
3    3   64    3    3    3    3    3
4    4    4    4    4    4    4    4
5    5    5    5    5    5    5    5
6    6    6    6    6    6    6    6
7    7    7    7    7    7    7    7
8    8    8    8    8    8    8    8

1    1    1    1    1    1    1    1
2    8    8    8    2    2    2    2
3    9   15    9    3    3    3    3
4   10   10   10    4    4    4    4
5    5    5    5    5    5    5    5
6    6    6    6    6    6    6    6
7    7    7    7    7    7    7    7
8    8    8    8    8    8    8    8

8    5    5    5    8    8    8   11    2    3    0    0    0    0    1    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

Exercise 1.3. Compile the program with option -O2. Analyse the optimizitation report.
You will get the following optimizitation report:

```
            Optimization for Procedure SPCONVOL

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7         1  J         FULL VECTOR Inter           No Strip
    8         2  I         Scalar

   16         3  J         FULL VECTOR Inter           No Strip
   17         4  I         Scalar

   25         5  L         Scalar
   26         6  K         Scalar
   29         7  I         FULL VECTOR Inter           No Strip Reduction
   30         8  J         Scalar

   38         9  I         FULL VECTOR                 No Strip

   41        10  J         Scalar
   42        11  I         Scalar

   48        12  I         Scalar
   49        13  J         Scalar

   52        14  I         Scalar
   53        15  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7         1  J       Interchanged to innermost
   16         3  J       Interchanged to innermost
   25         5  L       Unable to distribute
   26         6  K       Unable to distribute
   29         7  I       Interchanged to innermost
   42        11  I       Insufficient vector code
   48        12  I       Unable to distribute
   49        13  J       Insufficient vector code
   52        14  I       Unable to distribute
   53        15  J       Insufficient vector code
```

Why the compiler did not vectorize neither loop 10 nor loop 11?
Why did the compiler interchange loops of lines 7 and 8?
Why the compiler did not vectorize printing loops?

Exercise 1.4. Compile the program with option -O3. Analyse the optimizitation report.
You will get the following optimizitation report:

```
            Optimization for Procedure SPCONVOL

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7         1  J         FULL VECTOR Inter           No Strip
    8         2  I         PARALLEL

   16         3  J         FULL VECTOR Inter           No Strip
   17         4  I         PARALLEL

   25         5  L         PARALLEL
   26         6  K         Scalar
   29         7  I         FULL VECTOR Inter           No Strip Reduction
   30         8  J         Scalar

   38         9  I         PARA/VECTOR

   41        10  J         Scalar
   42        11  I         Scalar

   48        12  I         Scalar
   49        13  J         Scalar

   52        14  I         Scalar
   53        15  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7         1  J       Interchanged to innermost
   16         3  J       Interchanged to innermost
   25         5  L       Unable to distribute
   26         6  K       Unable to distribute
   29         7  I       Interchanged to innermost
   38         9  I       Parallel outer strip mine loop
   42        11  I       Insufficient vector code
   48        12  I       Unable to distribute
   49        13  J       Insufficient vector code
   52        14  I       Unable to distribute
   53        15  J       Insufficient vector code
```

Explain transformation PARA/VECTOR of loop 9.
Explain transformation PARALLEL of loop 2.

Exercise 1.5. Change the filter matrix and the program for performing sharpening filter
Exercise 1.6. Change the filter matrix and the program for performing Laplace operator
Exercise 1.7. Change the filter matrix and the program for performing the two components of a gradient operator

Exercise 1.8. Run under CXpa the program compiled with option -O2, then that compiled with O3. Analyse and compare the CXpa reports. Which version is faster, and which one requires more total CPU time?
Some parts of the CXpa report with option -O2:

```
================================================================================
                          Loop Performance Analysis
                                For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count          CPU Time
 Line                     Exec     Min     Max     Avg   (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
     7                          0       0       0       0           0.000000  u
     8                          1       8       8       8           0.000006
    16                          0       0       0       0           0.000000  u
    17                          1       3       3       3           0.000002
    25                          1       6       6       6           0.000238
    26                          6       6       6       6           0.000230
    29                          0       0       0       0           0.000000  u
    30                         36       3       3       3           0.000148
    38                          0       0       0       0           0.000000  u
    41                          1       8       8       8           0.000054
    42                          8       8       8       8           0.000041
    48                          1       8       8       8           0.003740
    49                          8       8       8       8           0.002484
    52                          1       8       8       8           0.003578
    53                          8       8       8       8           0.002542

Details:
            Optimization  Times      Iteration Count          CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 S                 1       8       8       8           0.000006
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    17    0 S                 1       3       3       3           0.000002
    16    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    25    0 S                 1       6       6       6           0.000008
    26    1 S                 6       6       6       6           0.000082
    30    2 S                36       3       3       3           0.000148
    29    3 V,I             N/A     N/A     N/A     N/A                N/A  u

    38    0 V               N/A     N/A     N/A     N/A                N/A  u

    41    0 S                 1       8       8       8           0.000013
    42    1 S                 8       8       8       8           0.000041

    48    0 S                 1       8       8       8           0.001256
    49    1 S                 8       8       8       8           0.002484

    52    0 S                 1       8       8       8           0.001036
    53    1 S                 8       8       8       8           0.002542
```

		Static Profile			Estimated Mflops (less inner loops)		Mflops (plus inner loops)		
Line	NL	Vector Spills	Vector Flops	Chime Count	Avg	Peak	Avg	Peak	PS
8	0	0	0	0	0.000	0.000	0.000	0.000	
7	1	0	0	0	N/A	N/A	0.000	0.000	u
17	0	0	0	0	0.000	0.000	0.000	0.000	
16	1	0	0	0	N/A	N/A	0.000	0.000	u
25	0	0	0	0	0.000	0.000	0.454	20.000	
26	1	0	0	0	0.000	0.000	0.470	20.000	
30	2	0	1	3	0.730	20.000	0.730	20.000	
29	3	0	0	0	N/A	N/A	0.000	0.000	u
38	0	0	0	0	N/A	N/A	0.000	0.000	u
41	0	0	0	0	0.000	0.000	0.000	0.000	
42	1	0	0	0	0.000	0.000	0.000	0.000	
48	0	0	0	0	0.000	0.000	0.000	0.000	
49	1	0	0	0	0.000	0.000	0.000	0.000	
52	0	0	0	0	0.000	0.000	0.000	0.000	
53	1	0	0	0	0.000	0.000	0.000	0.000	

(u) Loop(s) contains 1 or more uninstrumentable points.

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

Some parts of the CXpa report with option -O3:

```
================================================================================
                          Loop Performance Analysis
                                For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count         CPU Time
 Line                     Exec     Min     Max     Avg   (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
     7                          0       0       0       0           0.000000  u
     8                          1       8       8       8           0.000028
    16                          0       0       0       0           0.000000  u
    17                          1       3       3       3           0.000006
    25                          1       6       6       6           0.000201
    26                          6       6       6       6           0.000174
    29                          0       0       0       0           0.000000  u
    30                         36       3       3       3           0.000129
    38                          1      64      64      64           0.000008
    41                          1       8       8       8           0.000056
    42                          8       8       8       8           0.000041
    48                          1       8       8       8           0.003754
    49                          8       8       8       8           0.002456
    52                          1       8       8       8           0.003259
    53                          8       8       8       8           0.002286

Details:
            Optimization  Times      Iteration Count         CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 P                 1       8       8       8           0.000028
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    17    0 P                 1       3       3       3           0.000006
    16    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    25    0 P                 1       6       6       6           0.000027
    26    1 S                 6       6       6       6           0.000045
    30    2 S                36       3       3       3           0.000129
    29    3 V,I             N/A     N/A     N/A     N/A                N/A  u

    38    0 P/V               1      64      64      64           0.000008

    41    0 S                 1       8       8       8           0.000015
    42    1 S                 8       8       8       8           0.000041

    48    0 S                 1       8       8       8           0.001298
    49    1 S                 8       8       8       8           0.002456

    52    0 S                 1       8       8       8           0.000973
    53    1 S                 8       8       8       8           0.002286
```

```
                Static Profile                Estimated  Mflops
              Vector  Vector  Chime  (less inner loops)  (plus inner loops)
 Line    NL Spills  Flops   Count     Avg       Peak      Avg       Peak    PS
------- --- ------ ------- ------- --------- --------- --------- --------- --
     8    0      0       0       0     0.000     0.000     0.000     0.000
     7    1      0       0       0       N/A       N/A     0.000     0.000  u

    17    0      0       0       0     0.000     0.000     0.000     0.000
    16    1      0       0       0       N/A       N/A     0.000     0.000  u

    25    0      0       0       0     0.000     0.000     0.537    15.000
    26    1      0       0       0     0.000     0.000     0.621    15.000
    30    2      0       1       4     0.837    15.000     0.837    15.000
    29    3      0       0       0       N/A       N/A     0.000     0.000  u

    38    0      0       0       0     0.000     0.000     0.000     0.000

    41    0      0       0       0     0.000     0.000     0.000     0.000
    42    1      0       0       0     0.000     0.000     0.000     0.000

    48    0      0       0       0     0.000     0.000     0.000     0.000
    49    1      0       0       0     0.000     0.000     0.000     0.000

    52    0      0       0       0     0.000     0.000     0.000     0.000
    53    1      0       0       0     0.000     0.000     0.000     0.000

(u) Loop(s) contains 1 or more uninstrumentable points.

================================================================================
                     Parallel Region Performance Analysis
                                  For: MAIN_
================================================================================
Summary:
           Times            All Regions          Process Virtual
 Line      Exec       CPU Time   Wall Clock Time    Time (PVT)     CPU/PVT  PS
-------  --------   ------------ --------------- ---------------  --------- --
     8          1       0.000028        0.000485        0.000193      0.145
    17          1       0.000006        0.000100        0.000047      0.128
    25          1       0.000201        0.001127        0.000883      0.228
    38          1       0.000008        0.000113        0.000050      0.160

Details:
                            By Region
 Line                 CPU Time   Wall Clock Time    Chore Count             PS
-------              ------------ ---------------   -------------           ——
     8                   0.000028        0.000091              8
    17                   0.000006        0.000019              3
    25                   0.000201        0.001035              6
    38                   0.000008        0.000017              2

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

Excercise 2.

Median Filtering

Background

Here the input pixel is replaced by the median of the pixels contained in a window around the pixel, that is,

$$v(m,n) = median\{y(m-k, n-l), (k,l) \in W\}$$

where W is a suitably chosen window. The algorithm for median filtering requires arranging the pixel values in the window in increasing or decreasing order and picking the middle value. Generally the window size is chosen so that N_W is odd. If N_W is even,the median is taken as the average of the two values in the middle.

Exercise 2.1. Study and comment the following program implementing a simplified case of median filtering:

```
      program medianfilter
      parameter (m=8, n=8, fd=3, gr=64)
      integer imo(m,n),imn(m,n),his(gr)
      integer d,dd,h,i,j,k,l
      real temp,work(fd*fd)
c
      do j=1,n
       do i=1,m
     imo(i,j)=i
     imn(i,j)=i
     end do
       end do
      imo(3,3)=64
      imn(3,3)=64
c
      dd=(fd*fd-1)/2
      d=(fd-1)/2
c
      do l=2,n-1
     do k=2,m-1
c
       h=1
       do i=-d,d
         do j=-d,d
          work(h)=imo(k+j,l+i)
          h=h+1
         end do
       end do
c
```

```
        i=0
        do while (i .eq. 0)
         i=1
         do h=1,fd*fd-1
          if (work(h) .gt. work(h+1)) then
           i=0
           temp=work(h+1)
           work(h+1)=work(h)
           work(h)=temp
          endif
         end do
        end do
        imn(k,l)=int(work(dd+1))
      end do
       end do
c
       do i=1,gr
        his(i)=0
       end do
       do j=1,m
        do i=1,n
      temp=imn(i,j)
      his(temp)=his(temp)+1
        end do
       end do
c
       do i=1,n
        print 100,(imo(i,j), j=1,m)
       end do
       print *
       do i=1,n
        print 100,(imn(i,j), j=1,m)
       end do
c      read '(I0)',temp
       print *
       print 100,(his(i),i=1,gr)
100    format (1X,16I4)
       stop
       end
```

Explanation of the program:
The program begins with the picture generation (lines 7 to 14). The two nested loops using the iteration variables l and k step through the rows and columns of the picture. For sake of simplicity, the filter does not affect the border pixels of the picture.
The two nested loops beginning in lines 28 and 29 perform the calculation of the new pixel value, which will be put to its place in line 33.
The next 9 lines of the program computes the histogram of the new picture.
From line 47 the program prints the pixel values of the original picture, the new picture and the histogram values of the new picture.

Exercise 2.2. Compile and run the program! Analyse the result
You will get the following result when running the program:

```
1    1    1    1    1    1    1    1
2    2    2    2    2    2    2    2
3    3   64    3    3    3    3    3
4    4    4    4    4    4    4    4
5    5    5    5    5    5    5    5
6    6    6    6    6    6    6    6
7    7    7    7    7    7    7    7
8    8    8    8    8    8    8    8

1    1    1    1    1    1    1    1
2    2    2    2    2    2    2    2
3    3    3    3    3    3    3    3
4    4    4    4    4    4    4    4
5    5    5    5    5    5    5    5
6    6    6    6    6    6    6    6
7    7    7    7    7    7    7    7
8    8    8    8    8    8    8    8

8    8    8    8    8    8    8    8    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

Exercise 2.3. Compile the program with option -O2. Analyse the optimizitation report.
You will get the following optimizitation report:

```
            Optimization for Procedure MEDIANFILTER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         Scalar

   19        3  L         Scalar
   20        4  K         Scalar
   23        5  I         Scalar
   24        6  J         FULL VECTOR                 No Strip
   31        7  *NONE*    Scalar
   33        8  H         Scalar

   46        9  I         FULL VECTOR                 No Strip

   49       10  J         Scalar
   50       11  I         Scalar

   56       12  I         Scalar
   57       13  J         Scalar

   60       14  I         Scalar
   61       15  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7        1  J       Interchanged to innermost
   19        3  L       Inner loop has induction value with varying base or step
   20        4  K       Inner loop has induction value with varying base or step
   23        5  I       Inner loop has induction value with varying base or step
   31        7  *NONE*  No induction variable
   33        8  H       Insufficient vector code
   50       11  I       Insufficient vector code
   56       12  I       Unable to distribute
   57       13  J       Insufficient vector code
   60       14  I       Unable to distribute
   61       15  J       Insufficient vector code
```

Why the compiler did not vectorize neither loop 10 nor loop 11?

Exercise 2.4. Compile the program with option -O3. Analyse the optimizitation report. You will get the following optimizitation report:

```
            Optimization for Procedure MEDIANFILTER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         PARALLEL

   19        3  L         Scalar
   20        4  K         Scalar
   23        5  I         Scalar
   24        6  J         FULL VECTOR                 No Strip
   31        7  *NONE*    Scalar
   33        8  H         Scalar

   46        9  I         PARA/VECTOR

   49       10  J         Scalar
   50       11  I         Scalar

   56       12  I         Scalar
   57       13  J         Scalar

   60       14  I         Scalar
   61       15  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7        1  J       Interchanged to innermost
   19        3  L       Loop carried dependency
   20        4  K       Loop carried dependency
   23        5  I       Inner loop has induction value with varying base or step
   31        7  *NONE*  No induction variable
   33        8  H       Insufficient vector code
   46        9  I       Parallel outer strip mine loop
   50       11  I       Insufficient vector code
   56       12  I       Unable to distribute
   57       13  J       Insufficient vector code
   60       14  I       Unable to distribute
   61       15  J       Insufficient vector code
```

Explain transformation PARA/VECTOR of loop 9.
Explain transformation PARALLEL of loop 2.

Exercise 2.5. Run under CXpa the program compiled with option -O2, then that compiled with O3. Analyse and compare the CXpa reports. Which version is faster, and which one requires more total CPU time?

Some parts of the CXpa report with option -O3:

```
parents:  fclose for$clos2

================================================================================
                          Loop Performance Analysis
                                For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count         CPU Time
 Line                     Exec     Min     Max     Avg   (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
     7                          0       0       0       0           0.000000  u
     8                          1       8       8       8           0.000004
    19                          1       6       6       6           0.001335
    20                          6       6       6       6           0.001327
    23                         36       3       3       3           0.000114
    24                          0       0       0       0           0.000000  u
    31                         36       4       6       5           0.001093
    33                        181       8       8       8           0.000722
    46                          0       0       0       0           0.000000  u
    49                          1       8       8       8           0.000043
    50                          8       8       8       8           0.000033
    56                          1       8       8       8           0.003342
    57                          8       8       8       8           0.002243
    60                          1       8       8       8           0.003342
    61                          8       8       8       8           0.002345

Details:
            Optimization  Times      Iteration Count         CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 S                 1       8       8       8           0.000004
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    19    0 S                 1       6       6       6           0.000008
    20    1 S                 6       6       6       6           0.000120
    23    2 S                36       3       3       3           0.000114
    24    3 V               N/A     N/A     N/A     N/A                N/A  u
    31    2 S                36       4       6       5           0.000371
    33    3 S               181       8       8       8           0.000722

    46    0 V               N/A     N/A     N/A     N/A                N/A  u

    49    0 S                 1       8       8       8           0.000010
    50    1 S                 8       8       8       8           0.000033

    56    0 S                 1       8       8       8           0.001099
    57    1 S                 8       8       8       8           0.002243

    60    0 S                 1       8       8       8           0.000997
    61    1 S                 8       8       8       8           0.002345
```

	Static Profile			Estimated Mflops (less inner loops)		Mflops (plus inner loops)		
Line	NL	Vector Spills	Vector Flops	Chime Count	Avg	Peak	Avg	Peak	PS
8	0	0	0	0	0.000	0.000	0.000	0.000	
7	1	0	0	0	N/A	N/A	0.000	0.000	u
19	0	0	0	0	0.000	0.000	0.000	0.000	
20	1	0	0	0	0.000	0.000	0.000	0.000	
23	2	0	0	2	0.000	0.000	0.000	0.000	
24	3	0	0	0	N/A	N/A	0.000	0.000	u
31	2	0	0	0	0.000	0.000	0.000	0.000	
33	3	0	0	0	0.000	0.000	0.000	0.000	
46	0	0	0	0	N/A	N/A	0.000	0.000	u
49	0	0	0	0	0.000	0.000	0.000	0.000	
50	1	0	0	0	0.000	0.000	0.000	0.000	
56	0	0	0	0	0.000	0.000	0.000	0.000	
57	1	0	0	0	0.000	0.000	0.000	0.000	
60	0	0	0	0	0.000	0.000	0.000	0.000	
61	1	0	0	0	0.000	0.000	0.000	0.000	

(u) Loop(s) contains 1 or more uninstrumentable points.

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

Some parts of the CXpa report with option -O3:

```
================================================================================
                          Loop Performance Analysis
                                 For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count         CPU Time
 Line                     Exec     Min     Max     Avg   (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
     7                          0       0       0       0           0.000000  u
     8                          1       8       8       8           0.000024
    19                          1       6       6       6           0.001352
    20                          6       6       6       6           0.001343
    23                         36       3       3       3           0.000120
    24                          0       0       0       0           0.000000  u
    31                         36       4       6       5           0.001101
    33                        181       8       8       8           0.000737
    46                          1      64      64      64           0.000006
    49                          1       8       8       8           0.000053
    50                          8       8       8       8           0.000037
    56                          1       8       8       8           0.003695
    57                          8       8       8       8           0.002405
    60                          1       8       8       8           0.003676
    61                          8       8       8       8           0.002522

Details:
             Optimization  Times      Iteration Count         CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 P                 1       8       8       8           0.000024
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    19    0 S                 1       6       6       6           0.000009
    20    1 S                 6       6       6       6           0.000122
    23    2 S                36       3       3       3           0.000120
    24    3 V               N/A     N/A     N/A     N/A                N/A  u
    31    2 S                36       4       6       5           0.000364
    33    3 S               181       8       8       8           0.000737

    46    0 P/V               1      64      64      64           0.000006

    49    0 S                 1       8       8       8           0.000016
    50    1 S                 8       8       8       8           0.000037

    56    0 S                 1       8       8       8           0.001290
    57    1 S                 8       8       8       8           0.002405

    60    0 S                 1       8       8       8           0.001154
    61    1 S                 8       8       8       8           0.002522
```

```
                Static Profile                Estimated  Mflops
             Vector  Vector  Chime  (less inner loops)  (plus inner loops)
 Line    NL Spills  Flops   Count     Avg       Peak      Avg       Peak    PS
------- --- ------ ------- ------- --------- --------- --------- --------- --
      8    0      0       0       0     0.000     0.000     0.000     0.000
      7    1      0       0       0       N/A       N/A     0.000     0.000  u

     19    0      0       0       0     0.000     0.000     0.000     0.000
     20    1      0       0       0     0.000     0.000     0.000     0.000
     23    2      0       0       2     0.000     0.000     0.000     0.000
     24    3      0       0       0       N/A       N/A     0.000     0.000  u
     31    2      0       0       0     0.000     0.000     0.000     0.000
     33    3      0       0       0     0.000     0.000     0.000     0.000

     46    0      0       0       0     0.000     0.000     0.000     0.000

     49    0      0       0       0     0.000     0.000     0.000     0.000
     50    1      0       0       0     0.000     0.000     0.000     0.000

     56    0      0       0       0     0.000     0.000     0.000     0.000
     57    1      0       0       0     0.000     0.000     0.000     0.000

     60    0      0       0       0     0.000     0.000     0.000     0.000
     61    1      0       0       0     0.000     0.000     0.000     0.000

(u) Loop(s) contains 1 or more uninstrumentable points.

================================================================================
                     Parallel Region Performance Analysis
                                  For: MAIN_
================================================================================
Summary:
           Times            All Regions          Process Virtual
 Line      Exec       CPU Time   Wall Clock Time    Time (PVT)     CPU/PVT  PS
-------  --------   ------------ --------------- ---------------  --------- --
      8          1       0.000024        0.000399        0.000167      0.144
     46          1       0.000006        0.000098        0.000042      0.143

Details:
                            By Region
 Line                 CPU Time   Wall Clock Time    Chore Count             PS
-------              ------------ ---------------   -------------           __
      8                  0.000024        0.000084               8
     46                  0.000006        0.000015               2

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

Exercise 2.6. Compare median filtering algorithm to the spatial convolution algorithm from the point of view of vectorizability.

Exercise 2.7. Prevent the compiler from interchanging loops. Compile the program with option -O3 and analyse the optimization and Cxpa reports.

A possible solution:

```
      program medianfilter
      parameter (m=8, n=8, fd=3, gr=64)
      integer imo(m,n),imn(m,n),his(gr)
      integer d,dd,h,i,j,k,l
      real temp,work(fd*fd)
c
      do j=1,n
       do i=1,m
     imo(i,j)=i
        imn(i,j)=i
     end do
       end do
      imo(3,3)=64
      imn(3,3)=64
c
      dd=(fd*fd-1)/2
      d=(fd-1)/2
C$DIR FORCE_PARALLEL
      do l=2,n-1
     do k=2,m-1
c
          h=1
          do i=-d,d
         do j=-d,d
             work(h)=imo(k+j,l+i)
             h=h+1
           end do
          end do
c
       i=0
       do while (i .eq. 0)
        i=1
        do h=1,fd*fd-1
         if (work(h) .gt. work(h+1)) then
             i=0
             temp=work(h+1)
          work(h+1)=work(h)
          work(h)=temp
            endif
        end do
          end do
          imn(k,l)=int(work(dd+1))
        end do
      end do
c
```

```
      do i=1,gr
       his(i)=0
      end do
      do j=1,m
       do i=1,n
        temp=imn(i,j)
        his(temp)=his(temp)+1
       end do
      end do
c
      do i=1,n
       print 100,(imo(i,j), j=1,m)
      end do
      print *
      do i=1,n
       print 100,(imn(i,j), j=1,m)
      end do
c      read '(I0)',temp
      print *
      print 100,(his(i),i=1,gr)
100   format (1X,16I4)
      stop
      end
```

You will get the following optimizitation report for option -O3:

```
fc: Warning on line 24.6 of mp.f: WARNING: IGNORING DEPENDENCE FROM 'H' TO 'H' CAN
CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 25.14 of mp.f: WARNING: IGNORING DEPENDENCE FROM 'WORK' TO
'WORK' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 34.12 of mp.f: WARNING: IGNORING DEPENDENCE FROM 'WORK' TO
'WORK' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 34.25 of mp.f: WARNING: IGNORING DEPENDENCE FROM 'WORK' TO
'WORK' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!

            Optimization for Procedure MEDIANFILTER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         PARALLEL

   19        3  L         PARALLEL
   20        4  K         Scalar
   23        5  I         Scalar
   24        6  J         FULL VECTOR                 No Strip
   31        7  *NONE*    Scalar
   33        8  H         Scalar

   46        9  I         PARA/VECTOR

   49       10  J         Scalar
   50       11  I         Scalar

   56       12  I         Scalar
   57       13  J         Scalar

   60       14  I         Scalar
```

```
   61       15  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------------
    7         1  J       Interchanged to innermost
   20         4  K       Inner loop has induction value with varying base or step
   23         5  I       Inner loop has induction value with varying base or step
   31         7  *NONE*  No induction variable
   33         8  H       Insufficient vector code
   46         9  I       Parallel outer strip mine loop
   50        11  I       Insufficient vector code
   56        12  I       Unable to distribute
   57        13  J       Insufficient vector code
   60        14  I       Unable to distribute
   61        15  J       Insufficient vector code
```

Evaluate the warning messages at the beginning of the report.

Excercise 3.

Point Operations

Background

Point operations are zero memory operations where a given gray level u([0,L] is mapped into a gray level v([0,L] according to a transformation

$$v = f(u)$$

Proably the most frequently used point operation is contrast stretching.

Contrast Stretching
Low-contrast images occur often due to poor or nonuniform lighting conditions or due to nonlinearity or small dynamic range of the imaging sensor.

In general, contrast stretching can be expressed as

$$v = \begin{cases} \alpha u, & 0 \le u \langle a \\ \beta(u-a) + v_a, & a \le u \langle b \\ \gamma(u-b) + v_b, & b \le u \langle L \end{cases}$$

The slope of the transformation is chosen greater than unity in the region of stretch. The parameters a and b can be obtained by examining the histogram of the image. For example, the gray scale intervals where pixels occur most frequently would be stretched most to improve the overall visibility of a scene.

Clipping and Thresholding
A special case of contrast stretching where $\alpha=\gamma=0$ is called clipping. This is useful for noise reduction when the input signal is known to lie in the range [a,b].

Thresholding is a special case of clipping where a=b and the output becomes binary. For example, a seemingly binary image, such as a printed page, does not give binary output when scanned because of sensor noise and background illumination variations. Thresholding is used to make such an image binary.

Gray Scale Reversal (Digital Negative)
A negative image can be obtained by reverse scaling of the gray levels according to the transformation

$$v = L - u$$

Intensity Level Slicing
Without background:

$$v = \begin{cases} L, & a \le u \le b \\ 0, & otherwise \end{cases}$$

With background:

$$v = \begin{cases} L, & a \le u \le b \\ u, & otherwise \end{cases}$$

These transformations permit segmentation of certain gray level regions from the rest of the image. This technique is useful when different features of an image are contained in different gray levels.

Range Compression
Sometimes the dynamic range of the image data may be very large compared to the possibilities of the displaying device. The dynamic range can be compressed using "contrast stretching" with slope of the transformation smaller than or via the logarithmic transformation

$$v = c \log_{10}(1 + |u|)$$

where c is a scaling constant. This transformation enhances the small magnitude pixels compared to those pixels with large magnitudes.

Exercise 3.1. Study and comment the following program implementing a simplified case of contrast stretching:

```
      program pointoper
      parameter (m=8, n=8, gr=64)
      integer imo(m,n),imn(m,n),hiso(gr),hisn(gr)
      integer i,j,k,l
      real mul,work,shft
c
      do j=1,n
       do i=1,m
     imo(i,j)=int(gr/3+i)
     imn(i,j)=int(gr/3+i)
       end do
      end do
c
      mul=3
      shft=-50
c
      do l=1,n
       do k=1,m
     imn(k,l)=int(mul*imo(k,l)+shft)
       end do
      end do
c
      do i=1,gr
       hiso(i)=0
      end do
c
      do j=1,m
       do i=1,n
     work=imo(i,j)
     hiso(work)=hiso(work)+1
       end do
      end do
c
      do i=1,gr
       hisn(i)=0
      end do
c
      do j=1,m
       do i=1,n
     work=imn(i,j)
     hisn(work)=hisn(work)+1
       end do
      end do
c
      do i=1,n
       print 100,(imo(i,j), j=1,m)
      end do
```

```
      print *
      do i=1,n
       print 100,(imn(i,j), j=1,m)
      end do
c      read '(I0)',work
      print *
      print 100,(hiso(i),i=1,gr)
      print *
      print 100,(hisn(i),i=1,gr)
100   format (1X, 16I4)
      stop
      end
```

Explanation of the program:
The program begins with the picture generation (lines 7 to 12).
The two nested loops using the iteration variables l and k step through the rows and columns of the picture.
The two nested loops beginning in lines 17 and 18 perform the calculation of the new pixel value.
The next 21 lines of the program computes the histogram of the original and the new picture.
From line 44 the program prints the pixel values of the original picture, the new picture and the histogram values of the original and the new picture.

Exercise 3.2. Compile and run the program! Analyse the result

You will get the following result when running the program:

```
22  22  22  22  22  22  22  22
23  23  23  23  23  23  23  23
24  24  24  24  24  24  24  24
25  25  25  25  25  25  25  25
26  26  26  26  26  26  26  26
27  27  27  27  27  27  27  27
28  28  28  28  28  28  28  28
29  29  29  29  29  29  29  29

16  16  16  16  16  16  16  16
19  19  19  19  19  19  19  19
22  22  22  22  22  22  22  22
25  25  25  25  25  25  25  25
28  28  28  28  28  28  28  28
31  31  31  31  31  31  31  31
34  34  34  34  34  34  34  34
37  37  37  37  37  37  37  37

 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   8   8   8   8   8   8   8   8   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   8
 0   0   8   0   0   8   0   0   8   0   0   8   0   0   8   0
 0   8   0   0   8   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Exercise 3.3.. Compile the program with option -O2. Analyse the optimizitation report.
You will get the following optimizitation report:

```
           Optimization for Procedure POINTOPER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         Scalar

   17        3  L         FULL VECTOR Inter           No Strip
   18        4  K         Scalar

   23        5  I         FULL VECTOR                 No Strip

   27        6  J         Scalar
   28        7  I         Scalar

   34        8  I         FULL VECTOR                 No Strip

   38        9  J         Scalar
   39       10  I         Scalar

   45       11  I         Scalar
   46       12  J         Scalar

   49       13  I         Scalar
   50       14  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7        1  J       Interchanged to innermost
   17        3  L       Interchanged to innermost
   28        7  I       Insufficient vector code
   39       10  I       Insufficient vector code
   45       11  I       Unable to distribute
   46       12  J       Insufficient vector code
   49       13  I       Unable to distribute
   50       14  J       Insufficient vector code
```

Exercise 3.4. Compile the program with option -O3. Analyse the optimizitation report.
You will get the following optimizitation report:

```
           Optimization for Procedure POINTOPER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         PARALLEL

   17        3  L         FULL VECTOR Inter           No Strip
   18        4  K         PARALLEL

   23        5  I         PARA/VECTOR

   27        6  J         Scalar
   28        7  I         Scalar

   34        8  I         PARA/VECTOR

   38        9  J         Scalar
   39       10  I         Scalar

   45       11  I         Scalar
   46       12  J         Scalar

   49       13  I         Scalar
   50       14  J         Scalar

Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------
    7        1  J       Interchanged to innermost
   17        3  L       Interchanged to innermost
   23        5  I       Parallel outer strip mine loop
   28        7  I       Insufficient vector code
   34        8  I       Parallel outer strip mine loop
   39       10  I       Insufficient vector code
   45       11  I       Unable to distribute
   46       12  J       Insufficient vector code
   49       13  I       Unable to distribute
   50       14  J       Insufficient vector code
```

Explain transformation PARA/VECTOR of loop *.
Explain transformation PARALLEL of loop *.

Exercise 3.5. 8. Run under CXpa the program compiled with option -O2, then that compiled with O3. Analyse and compare the CXpa reports. Which version is faster, and which one requires more total CPU time?

Some parts of the CXpa report with option -O3:

```
================================================================================
                          Loop Performance Analysis
                                For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count          CPU Time
 Line                     Exec     Min     Max     Avg   (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
     7                          0       0       0       0           0.000000  u
     8                          1       8       8       8           0.000004
    17                          0       0       0       0           0.000000  u
    18                          1       8       8       8           0.000013
    23                          0       0       0       0           0.000000  u
    27                          1       8       8       8           0.000046
    28                          8       8       8       8           0.000036
    34                          0       0       0       0           0.000000  u
    38                          1       8       8       8           0.000051
    39                          8       8       8       8           0.000042
    45                          1       8       8       8           0.003590
    46                          8       8       8       8           0.002434
    49                          1       8       8       8           0.003459
    50                          8       8       8       8           0.002421

Details:
             Optimization  Times      Iteration Count          CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 S                 1       8       8       8           0.000004
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    18    0 S                 1       8       8       8           0.000013
    17    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    23    0 V               N/A     N/A     N/A     N/A                N/A  u

    27    0 S                 1       8       8       8           0.000010
    28    1 S                 8       8       8       8           0.000036

    34    0 V               N/A     N/A     N/A     N/A                N/A  u

    38    0 S                 1       8       8       8           0.000009
    39    1 S                 8       8       8       8           0.000042

    45    0 S                 1       8       8       8           0.001156
    46    1 S                 8       8       8       8           0.002434

    49    0 S                 1       8       8       8           0.001038
    50    1 S                 8       8       8       8           0.002421
```

```
                Static Profile                Estimated  Mflops
              Vector  Vector  Chime  (less inner loops)  (plus inner loops)
 Line     NL Spills  Flops   Count     Avg       Peak      Avg       Peak    PS
------- --- ------ ------- ------- --------- --------- --------- --------- --
      8    0      0       0       0     0.000     0.000     0.000     0.000
      7    1      0       0       0       N/A       N/A     0.000     0.000  u

     18    0      0       2       2     1.231    60.000     1.231    60.000
     17    1      0       0       0       N/A       N/A     0.000     0.000  u

     23    0      0       0       0       N/A       N/A     0.000     0.000  u

     27    0      0       0       0     0.000     0.000     0.000     0.000
     28    1      0       0       0     0.000     0.000     0.000     0.000

     34    0      0       0       0       N/A       N/A     0.000     0.000  u

     38    0      0       0       0     0.000     0.000     0.000     0.000
     39    1      0       0       0     0.000     0.000     0.000     0.000

     45    0      0       0       0     0.000     0.000     0.000     0.000
     46    1      0       0       0     0.000     0.000     0.000     0.000

     49    0      0       0       0     0.000     0.000     0.000     0.000
     50    1      0       0       0     0.000     0.000     0.000     0.000

(u) Loop(s) contains 1 or more uninstrumentable points.

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

Some parts of the CXpa report with option -O3:

```
================================================================================
                            Loop Performance Analysis
                                  For: MAIN_
================================================================================
Summary:
                          Times      Iteration Count          CPU Time
 Line                     Exec     Min     Max     Avg    (plus inner loops) PS
-------                  -------- ------- ------- ------- ------------------ --
      7                         0       0       0       0           0.000000  u
      8                         1       8       8       8           0.000036
     17                         0       0       0       0           0.000000  u
     18                         1       8       8       8           0.000024
     23                         1      64      64      64           0.000007
     27                         1       8       8       8           0.000045
     28                         8       8       8       8           0.000034
     34                         1      64      64      64           0.000006
     38                         1       8       8       8           0.000055
     39                         8       8       8       8           0.000040
     45                         1       8       8       8           0.003743
     46                         8       8       8       8           0.002489
     49                         1       8       8       8           0.003622
     50                         8       8       8       8           0.002515
```

```
Details:
            Optimization  Times      Iteration Count         CPU Time
 Line    NL    Report     Exec     Min     Max     Avg   (less inner loops) PS
------- --- ----------- -------- ------- ------- ------- ------------------ --
     8    0 P                 1       8       8       8           0.000036
     7    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    18    0 P                 1       8       8       8           0.000024
    17    1 V,I             N/A     N/A     N/A     N/A                N/A  u

    23    0 P/V               1      64      64      64           0.000007

    27    0 S                 1       8       8       8           0.000011
    28    1 S                 8       8       8       8           0.000034

    34    0 P/V               1      64      64      64           0.000006

    38    0 S                 1       8       8       8           0.000015
    39    1 S                 8       8       8       8           0.000040

    45    0 S                 1       8       8       8           0.001254
    46    1 S                 8       8       8       8           0.002489

    49    0 S                 1       8       8       8           0.001107
    50    1 S                 8       8       8       8           0.002515

(u) Loop(s) contains 1 or more uninstrumentable points.

================================================================================
                     Parallel Region Performance Analysis
                                  For: MAIN_
================================================================================
Summary:
           Times            All Regions          Process Virtual
 Line      Exec       CPU Time   Wall Clock Time    Time (PVT)     CPU/PVT  PS
-------  --------   ------------ --------------- ---------------  --------- --
     8          1       0.000036        0.000598        0.000217      0.166
    18          1       0.000024        0.000122        0.000074      0.324
    23          1       0.000007        0.000091        0.000041      0.171
    34          1       0.000006        0.000095        0.000044      0.136

Details:
                           By Region
 Line                CPU Time   Wall Clock Time    Chore Count             PS
-------             ------------ ---------------   -------------           ——
     8                  0.000036        0.000303               8
    18                  0.000024        0.000048               8
    23                  0.000007        0.000014               2
    34                  0.000006        0.000016               2

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

Exercise 3.6. - Compare point operations to the median filtering algorithm and to the spatial convolution algorithm from the point of view of vectorizability.
Exercise 3.7. -Modify the program for decreasing the contrast of a picture the pixel values of which cannot be diplayed on a given device because of its limited gradational resolution.

Exercise 3.8. -Modify the program in such a way that the two histograms would be computed parallel with each other. Analyse the optimizitation reports. Run the modified program under CXpa. Analyse the CXpa reports.

A possible solution:

```
      program pointoper
      parameter (m=8, n=8, gr=64)
      integer imo(m,n),imn(m,n),hiso(gr),hisn(gr)
      integer i,j,k,l
      real mul,work,work1,shft
c
      do j=1,n
       do i=1,m
     imo(i,j)=int(gr/3+i)
        imn(i,j)=int(gr/3+i)
       end do
      end do
c
      mul=3
      shft=-50
c
      do l=1,n
       do k=1,m
        imn(k,l)=int(mul*imo(k,l)+shft)
       end do
      end do
C$DIR BEGIN_TASKS
      do i=1,gr
       hiso(i)=0
      end do
C$DIR NEXT_TASK
      do j=1,gr
       hisn(j)=0
      end do
C$DIR END_TASKS
C$DIR BEGIN_TASKS
      do j=1,m
       do i=1,n
        work=imo(i,j)
        hiso(work)=hiso(work)+1
       end do
      end do
C$DIR NEXT_TASK
      do j=1,m
       do i=1,n
        work1=imn(i,j)
        hisn(work1)=hisn(work1)+1
       end do
      end do
```

```
C$DIR END_TASKS
      do i=1,n
       print 100,(imo(i,j), j=1,m)
      end do
      print *
      do i=1,n
       print 100,(imn(i,j), j=1,m)
      end do
c      read '(I0)',work
      print *
      print 100,(hiso(i),i=1,gr)
      print *
      print 100,(hisn(i),i=1,gr)
100   format (1X, 16I4)
      stop
      end
```

You will get the following optimizitation report for option -O3:

```
fc: Warning on line 24.8 of pp.f: WARNING: IGNORING DEPENDENCE FROM 'HISO' TO
'HISO' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 28.8 of pp.f: WARNING: IGNORING DEPENDENCE FROM 'HISN' TO
'HISN' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 35.20 of pp.f: WARNING: IGNORING DEPENDENCE FROM 'HISO' TO
'HISO' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!
fc: Warning on line 42.21 of pp.f: WARNING: IGNORING DEPENDENCE FROM 'HISN' TO
'HISN' CAN CAUSE GENERATION OF INCORRECT OBJECT CODE !!

            Optimization for Procedure POINTOPER

Line      Id    Iter.     Reordering        New       Optimizing / Special
Num.      Num.  Var.      Transformation    Loops     Transformation
-------------------------------------------------------------------------
    7        1  J         FULL VECTOR Inter           No Strip
    8        2  I         PARALLEL

   17        3  L         FULL VECTOR Inter           No Strip
   18        4  K         PARALLEL

   22        5  *VAR*     PARALLEL
   23        6  I         FULL VECTOR                 No Strip
   27        7  J         FULL VECTOR                 No Strip

   31        8  *VAR*     PARALLEL
   32        9  J         Scalar
   33       10  I         Scalar
   39       11  J         Scalar
   40       12  I         Scalar

   46       13  I         Scalar
   47       14  J         Scalar

   50       15  I         Scalar
   51       16  J         Scalar
```

```
Line      Id    Iter.   Analysis
Num.      Num.  Var.
-------------------------------------------------------------------------------
    7        1  J       Interchanged to innermost
   17        3  L       Interchanged to innermost
   33       10  I       Insufficient vector code
   40       12  I       Insufficient vector code
   46       13  I       Unable to distribute
   47       14  J       Insufficient vector code
   50       15  I       Unable to distribute
   51       16  J       Insufficient vector code
```

Evaluate the warning messages at the beginning of the report..

Bibliography

1. CONVEX Architecture Reference Manual (C Series)
/Sixth Edition, April, 1992 – Convex Computer Corporation/
/Convex Press, Richardson, Texas United States of America – DHW-300/

1. CONVEX FORTRAN Optimization Guide
/Fourth Edition, November, 1992 – Convex Computer Corporation/
/Convex Press, Richardson, Texas United States of America – DSW-034/

CONVEX CXpa User's Guide 3rd edition March 1993
CONVEX Press
Richardson, Texas USA

CONVEX CXpa Reference 1st edition March 1993

I. Pitas: DIGITAL IMAGE PROCESSING ALGORITHMS
Prentice Hall 1993.

Larry Nyhoff- Sanford Leestma: FORTRAN 77
and Numerical Methods for Engineers and Scientists
Prentice Hall 1995.

A. K. Jain: Fundamentals of Digital Image Processing
Prentice Hall 1989.

Etelka Kovács - Ágnes Burián:
Architecture and Operation of Vector Computer CONVEX 3820
TEMPUS Project S_JEP 8333-94

Dr. Rezső Nagy - Magda Kovács:
Optimization and Profiling on Vector Computer CONVEX 3820
TEMPUS Project S_JEP 8333-94

Magda Kovács - Ágnes Burián:
Digital Image Processing Exercises in the Fourier Domain using VECLIB on
Vector Computer CONVEX 3820
TEMPUS Project S_JEP 8333-94