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

SLIDES

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

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

Excercise 1.

Spatial Convolution Filter

Background

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.

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. $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

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
```

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.

Excercise 2.

Median Filtering

Background

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 2.2. 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
```

Excercise 3.

Point Operations

Background

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

$$v = f(u)$$

Contrast Stretching

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

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.

Clipping and Thresholding

Clipping: a special case of contrast stretching where $\alpha=\gamma=0$

Thresholding: a special case of clipping where a=b and the output becomes binary

Gray Scale Reversal (Digital Negative)

$$v = L - u$$

Intensity Level Slicing

Without background:

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

With background:

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

Range Compression

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

where c is a scaling constant.

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
```