ECE 6310: INTRODUCTION TO COMPUTER VISION

Lab 1 Report

C14109603

Rahil Modi
rmodi@clemson.edu

1. Normal 7x7 filter:

1.1 Introduction:

In this part I will modifying the image smoothing code provided to incorporate a 7x7 filter and apply it on a 512x512 PPM image.

1.2 Snippet of the code:

```
for (r=0; r<ROWS; r++)
for (c=0; c<COLS; c++)
    {
    sum=0;
    if (r<3 || r>=ROWS-3)
    {
        sum=0;
    }
    else if (c<3 || c>=COLS-3)
    {
        sum=0;
    }
    else
    {
    for (r2=-3; r2<=3; r2++)
    for (c2=-3; c2<=3; c2++)
        sum += image[(r+r2)*COLS+(c+c2)];
    smoothed[r*COLS+c]=sum/49;
    }
    }
```

Figure 1 Normal 7x7 Filter

1.3 Average Time:

Normal 7x7 filter
23931200
23934600
26961200
25931900

23937100
32911900
25913000
30915600
30914200
32909200
Avg: 27825990

1.4 Output Image:

Figure 2 Output of normal 7x7 filter

2. Separable filters:

2.1 Introduction:

In this part I will modifying the image smoothing code provided to incorporate a 1x7 and 7x1 separable filters and apply it on a 512x512 PPM image.

2.2 Snippet of the code:

```
for (r=0; r<ROWS; r++)
for (c=0; c<COLS; c++)
{
    sum=0;
    if (c<3 || c>=COLS-3)
    {
        sum=0;
    }
    else
    {
    for (c2=-3; c2<=3; c2++)
    sum+=image[(r)*COLS+(c+c2)];
    smoothed[r*COLS+c]=sum;
    }
}
    /* Vertical filter (7x1) */
for (r=0; r<ROWS; r++)
for (c=0; c<COLS; c++)
{
    sum=0;
    if (r<3 || r>=ROWS-3)
    {
        sum=0;
    }
    else
    {
    for (r2=-3; r2<=3; r2++)
    sum+=smoothed[(r+r2)*COLS+(c)];
    smoothed_2[r*COLS+c]=sum/49;
    }
}
```

Figure 3 Separable filters

2.3 Average Time:

Separable filters
10978400
10971700
10964000
10988900
17985700
10005800
10014300
10969300
10970400
11950400
Avg: 11579890

2.4 Output Image:

Figure 4 Output of separable filters

3. Separable filters with sliding window:

3.1 Introduction:

In this part I will modifying the image smoothing code provided to incorporate a 1x7 and 7x1 separable filters with sliding window and apply it on a 512x512 PPM image.

3.2 Snippet of the code:

```
for (r=0; r<ROWS; r++)
{
    sum = 0;
    for (c=0; c<COLS; c++)
    {
        if (c<3 || c>=COLS-3)
        {
            sum=0;
        }
        else
        {
            for (c2=-3; c2<=3; c2++)
            {
                if (c == 3)
                {
                    sum+=image[(c+c2)+r*COLS];
                }
                else
                {
                    sum = sum - image[(c2+c-1)+r*COLS] + image[(c+c2+6)+r*COLS];
                    break;
                }
            }
            smoothed[(r)*COLS+c]=sum;
        }
    }
}
```

```
for (c=0; c<COLS; c++)
{
    sum = 0;
    for (r=0; r<ROWS; r++)
    {
        if (r<3 || r>=ROWS-3)
        {
            sum=0;
        }
        else
        {
            for (r2=-3; r2<=3; r2++)
            {
                if(r==3)
                {
                    sum+=smoothed[(r+r2)*COLS+c];
                }
                else
                {
                    sum = sum - smoothed[(r+r2-1)*COLS+c] + smoothed[(r+r2+6)*COLS+c];
                    break;
                }
            }
            smoothed_2[(r)*COLS+c]=(sum/49);
        }
    }
}
```

Figure 4 Separable filters with sliding window

3.3 Average Time:

Separable filters with Sliding Window
3988300
3989200
4015900
6443600
4017100
3992300
4023300
3988600

3990800
3966800
Avg: 4241590

3.4 Output Image:

Figure 5 Output of separable filters with sliding window

4. Checking for any differences in the files

4.1 Introduction
The equivalent of the diff command of Linux for Windows is to use FC [filename1] [filename2]. Using this command, I am checking for any differences between all the 3 images.

4.2 Output of the FC command:

Figure 6 Output of the difference command

As you can see in the image there are no differences between all the 3 images which proves that all the 3 images output is same.

5. Comparison of amount of time each filter takes to run:

Normal 7x7 filter	Separable filters	Separable filters with Sliding Window
23931200	10978400	3988300
23934600	10971700	3989200
26961200	10964000	4015900
25931900	10988900	6443600
23937100	17985700	4017100
32911900	10005800	3992300
25913000	10014300	4023300
30915600	10969300	3988600
30914200	10970400	3990800
32909200	11950400	3966800
Avg: 27825990	**Avg: 11579890**	**Avg: 4241590**

Looking at the amount of time each filter takes to run and taking the average of the 10 samples the time in milliseconds is what was expected, the normal 7x7 filter took the highest time to process, separable filters were an improvement over normal filter and the separable filters with sliding window was the fastest. And all the 3 filters had the same output.