

어셈블리프로그램 설계 및 실습

Term Project 보고서



담당교수	이형근 교수님
실습분반	화요일 (2분반)
소속	컴퓨터정보공학부
학번·성명	2021202085 전한아솔
제출일	2024년 12월 02일

1. Problem Statement

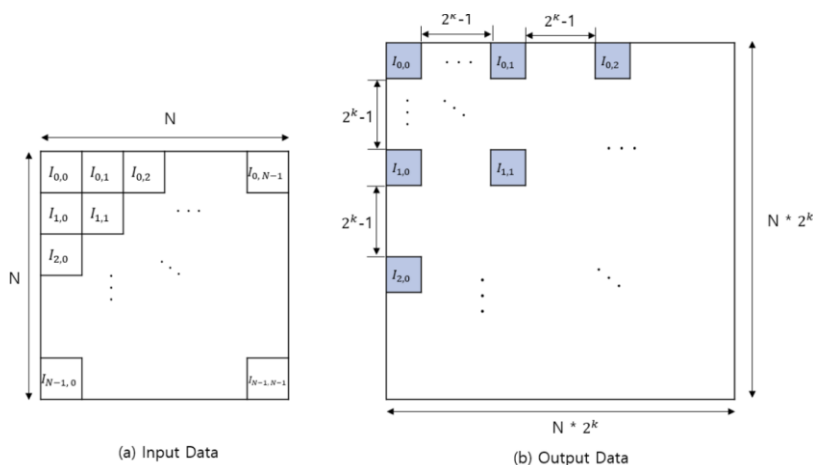
이번 Term project는 Bilinear interpolation 알고리즘을 사용하여 20x20 크기의 픽셀 값을 80x80의 크기로, 4배 확대하는 어셈블리 코드를 작성하는 프로젝트이다.

주어지는 각 픽셀값은 IEEE 754 Single precision으로 구성되므로 20x20 크기의 픽셀 값을 4배 확대할 때, floating point 연산이 필요하다.

따라서 이번 프로젝트는 어셈블리 코드를 통한 Floating point의 연산을 구현함으로써 floating point에 대한 개념과 이론을 좀 더 알아갈 수 있고, 메모리나 Stack을 통한 매개변수 전달 방법을 좀 더 숙지할 수 있으며 Bilinear interpolation 알고리즘을 더 알아가는 목적을 가진다.

2. Algorithm

과제 코드를 작성하기 위한 설계 방법, 접근 방식, 전략 등을 작성합니다.



제안서에 나와있는 Output Data의 사진을 보게되면, 이번 과제에서는 4배 확대를 진행하므로 $N=20$, $k=2$ 라고 생각할 수 있다. 따라서 20x20의 데이터는 왼쪽 상단(0, 0)부터 3칸(1칸은 4 bytes)의 간격을 두고 하나씩 배치되므로 Text file에서 가져오는 20x20의 데이터를 80x80의 결과 값을 저장하는 메모리부터 3칸의 간격을 두고 저장했다.

하나의 Word를 각 칸이라고 표현하고, 각 행을 i , 각 열을 j 라고 표현하며 80x80을 좌표로 생각하고 (i, j) 의 형태로 설명하겠다.

20x20의 데이터를 80x80의 메모리에 배치한 후, 각 값을 이용하여 중간에 위치한 값의 평균을 구하여 저장하는 방식으로 보간을 구현했다. 예를 들면 (0, 0)과 (0, 4)의 평균 값을 구해서 (0, 2)에 저장하는 방식으로 수평 보간을 진행하고 (0, 0)과 (4, 0)의 평균 값을 구해서 (2, 0)에 저장하는 방식으로 수직 보간을 진행했다.

먼저, 80x80의 수평 보간을 진행하는데, 20x20의 저장된 데이터는 0, 4, 8, ... ,72, 76의 행에 3칸의 간격으로 저장되어있다. 따라서 4의 배수인 행만 0~76까지 수평 보간을 진행한다.

한 행은 80개의 칸(0~79)으로 이루어져 있으므로 320 bytes, 140(hex)이다. 따라서 현재 행에 140(hex)를 곱한 값을 결과의 주소를 나타내는 r5에 더해서, 현재 내가 보간을 진행할 행의 첫 메모리 주소를 r5에 저장했다.

J는 처음에 0이므로 j와 j+4칸에 존재하는 값을 가져와, Floating point의 덧셈 연산을 수행한다. 연산이 끝난 후 결과의 EXP(지수부)에 1을 빼주게 되면, 2진수의 base는 2이므로 2로 나눈 결과와 동일하다. 이는 곧, 두 Floating point number의 평균이므로 해당 값을 j+2에 저장한다. 이러한 방식을 j와 j+2, j+2와 j+4에 수행하여 한 레이블에서 j+1, j+2, j+3의 값을 모두 저장해서 4칸의 보간을 완료한다. 한 행에서 0~76열까지 이런 방식으로 모두 보간 후, 77, 78, 79번째 값은 76의 값을 Padding해야하므로 j가 76이 되면, 79열까지 76열의 값을 복사했다.

이렇게 0~76행에 4의 배수인 행만 수평 보간이 끝나면, 0~79의 열을 모두 수직 보간하여 80x80의 값을 모두 채운다.

수직 보간을 수평 보간에서 i와 j의 역할을 바꿔주면 된다. 달라지는 부분은 한 열의 수직 보간이 끝나고 다음 열로 넘어갈 때, 열 사이에는 4bytes의 차이가 존재하므로 r5에 현재 열에 4를 곱하여 내가 수직 보간 하고싶은 메모리의 주소를 r5에 저장한다. 또한 수평 보간에서는 4 열의 차이로 보간을 진행하므로 16 bytes의 차이만 존재했지만, 수직 보간에서는 4 행의 차이로 보간을 진행하므로 500(140x4, hex)의 차이가 있다. 이 차이를 고려하여 수직 보간도 한 열에서 i와 i+4의 평균 값을 구하여 i+2에 저장, i와 i+2의 평균을 구하여 i+1에 저장, i+2와 i+4의 평균을 구하여 i+3에 저장하고 i가 76이 된다면, 나머지 77, 78, 79의 값을 76번째 값으로 Padding을 하는 방식으로 구현했다. 수직 보간은 0~79의 열을 모두 수행하므로 모든 열에 수직 보간을 수행하면 80x80의 값이 모두 채워진다.

3. Result

<20x20의 입력 값을 80x80에 저장>

0x10000000:	1	0	0	0	0	0	0	0
0x10000020:	0	0	0	0	0	0	0	0
0x10000040:	0	0	0	0	0	0	0	0
0x10000060:	0	0	0	0	0	0	0	0
0x10000080:	0	0	0	0	0	0	0	0
0x100000A0:	0	0	0	0	0	0	0	0
0x100000C0:	0	0	0	0	0	0	0	0
0x100000E0:	0	0	0	0	0	0	0	0
0x10000100:	0	0	0	0	0	0	0	0
0x10000120:	0	0	0	0	0	0	0	0

80x80의 결과를 저장하는 주소(0x10000000)의 0행에 4칸씩 20x20의 입력 값을 저장한 모습

어셈블리프로그램 설계 및 실습

```
0x10005F00: 0 0 0 0 0 0 0 0
0x10005F20: 0 0 0 0 0 0 0 0
0x10005F40: 0 0 0 0 0 0 0 0
0x10005F60: 0 0 0 0 3 0 0 0
0x10005F80: 2 0 0 0 0 0 0 0
0x10005FA0: 3 0 0 0 3 0 0 0
0x10005FC0: 0 0 0 0 0 0 0 0
0x10005FE0: 0 0 0 0 0 0 0 0
0x10006000: 0 0 0 0 0 0 0 0
0x10006020: 0 0 0 0 0 0 0 0
```

5F00(hex)=24320(dec)=1280x19이다. 즉, 4행씩 19번을 저장하면 마지막에 결과 값은 76번째 행인 0x10005F00부터 저장된다.

80x80을 모두 저장하면 0x10006400전까지 모든 word가 채워진다. 20x20의 결과 값만 4행씩 저장하면 0x10006040부터 0x10006400전까지는 Padding이 채워질 공간이므로 0으로 비어있다.

```
0x10006040: 0 0 0 0 0 0 0 0
0x10006060: 0 0 0 0 0 0 0 0
0x10006080: 0 0 0 0 0 0 0 0
0x100060A0: 0 0 0 0 0 0 0 0
0x100060C0: 0 0 0 0 0 0 0 0
0x100060E0: 0 0 0 0 0 0 0 0
0x10006100: 0 0 0 0 0 0 0 0
0x10006120: 0 0 0 0 0 0 0 0
0x10006140: 0 0 0 0 0 0 0 0
0x10006160: 0 0 0 0 0 0 0 0
0x10006180: 0 0 0 0 0 0 0 0
0x100061A0: 0 0 0 0 0 0 0 0
0x100061C0: 0 0 0 0 0 0 0 0
0x100061E0: 0 0 0 0 0 0 0 0
0x10006200: 0 0 0 0 0 0 0 0
0x10006220: 0 0 0 0 0 0 0 0
0x10006240: 0 0 0 0 0 0 0 0
0x10006260: 0 0 0 0 0 0 0 0
0x10006280: 0 0 0 0 0 0 0 0
0x100062A0: 0 0 0 0 0 0 0 0
0x100062C0: 0 0 0 0 0 0 0 0
0x100062E0: 0 0 0 0 0 0 0 0
0x10006300: 0 0 0 0 0 0 0 0
0x10006320: 0 0 0 0 0 0 0 0
0x10006340: 0 0 0 0 0 0 0 0
0x10006360: 0 0 0 0 0 0 0 0
0x10006380: 0 0 0 0 0 0 0 0
0x100063A0: 0 0 0 0 0 0 0 0
0x100063C0: 0 0 0 0 0 0 0 0
0x100063E0: 0 0 0 0 0 0 0 0
```

<4행씩 수평 보간>

```
0x10000000: 1 0.75 0.5 0.25 0 0 0 0
0x10000020: 0 0 0 0 0 0 0 0
0x10000040: 0 0 0 0 0 0 0 0
0x10000060: 0 0 0 0 0 0 0 0
0x10000080: 0 0 0 0 0 0 0 0
0x100000A0: 0 0 0 0 0 0 0 0
0x100000C0: 0 0 0 0 0 0 0 0
0x100000E0: 0 0 0 0 0 0 0 0
0x10000100: 0 0 0 0 0 0 0 0
0x10000120: 0 0 0 0 0 0 0 0
```

아까는 중간에 3칸이 비어있었지만 수평 보간을 진행한 후, 한 행이 모두 보간된 모습을 볼 수 있다. 마지막 3칸은 76번째 열의 값을 그대로 복사하여 Padding을 채웠다.

```
0x10000140: 0 0 0 0 0 0 0 0
0x10000160: 0 0 0 0 0 0 0 0
0x10000180: 0 0 0 0 0 0 0 0
0x100001A0: 0 0 0 0 0 0 0 0
0x100001C0: 0 0 0 0 0 0 0 0
0x100001E0: 0 0 0 0 0 0 0 0
0x10000200: 0 0 0 0 0 0 0 0
0x10000220: 0 0 0 0 0 0 0 0
0x10000240: 0 0 0 0 0 0 0 0
0x10000260: 0 0 0 0 0 0 0 0
```

0~76까지 4행씩 건너뛰어서 보간을 진행했으므로 해당하지 않는 행은 아직 비어있다.

<1열씩, 모두 수직 보간>

0x10000140:	0.75	0.5625	0.375	0.1875	0	0	0	0
0x10000160:	0	0	0	0	0	0	0	0
0x10000180:	0	0	0	0	0	0	0	0
0x100001A0:	0	0	0	0	0	0.0625	0.125	0.1875
0x100001C0:	0.25	0.375	0.5	0.625	0.75	0.8125	0.875	0.9375
0x100001E0:	1	1	1	1	1	1	1	1
0x10000200:	1	0.9375	0.875	0.8125	0.75	0.5625	0.375	0.1875
0x10000220:	0	0	0	0	0	0	0	0
0x10000240:	0	0	0	0	0	0	0	0
0x10000260:	0	0	0	0	0	0	0	0
0x10000280:	0.5	0.375	0.25	0.125	0	0	0	0
0x100002A0:	0	0	0	0	0	0	0	0
0x100002C0:	0	0	0	0	0	0	0	0
0x100002E0:	0	0	0	0	0	0.125	0.25	0.375
0x10000300:	0.5	0.75	1	1.25	1.5	1.625	1.75	1.875
0x10000320:	2	2	2	2	2	2	2	2
0x10000340:	2	1.875	1.75	1.625	1.5	1.125	0.75	0.375
0x10000360:	0	0	0	0	0	0	0	0
0x10000380:	0	0	0	0	0	0	0	0
0x100003A0:	0	0	0	0	0	0	0	0
0x100003C0:	0.25	0.1875	0.125	0.0625	0	0	0	0
0x100003E0:	0	0	0	0	0	0	0	0
0x10000400:	0	0	0	0	0	0	0	0
0x10000420:	0	0	0	0	0	0.1875	0.375	0.5625
0x10000440:	0.75	1.125	1.5	1.875	2.25	2.4375	2.625	2.8125
0x10000460:	3	3	3	3	3	3	3	3
0x10000480:	3	2.8125	2.625	2.4375	2.25	1.6875	1.125	0.5625
0x100004A0:	0	0	0	0	0	0	0	0
0x100004C0:	0	0	0	0	0	0	0	0
0x100004E0:	0	0	0	0	0	0	0	0

0행과 4행에 채워진 값으로 1~3행의 비어있던 값이 모두 보간됨을 볼 수 있다.

0x10005F00:	0	0	0	0	0	0	0	0
0x10005F20:	0	0	0	0	0	0	0	0
0x10005F40:	0	0	0	0	0	0	0	0
0x10005F60:	0	0.75	1.5	2.25	3	2.75	2.5	2.25
0x10005F80:	2	1.5	1	0.5	0	0.75	1.5	2.25
0x10005FA0:	3	3	3	3	3	2.25	1.5	0.75
0x10005FC0:	0	0	0	0	0	0	0	0
0x10005FE0:	0	0	0	0	0	0	0	0
0x10006000:	0	0	0	0	0	0	0	0
0x10006020:	0	0	0	0	0	0	0	0
0x10006040:	0	0	0	0	0	0	0	0
0x10006060:	0	0	0	0	0	0	0	0
0x10006080:	0	0	0	0	0	0	0	0
0x100060A0:	0	0.75	1.5	2.25	3	2.75	2.5	2.25
0x100060C0:	2	1.5	1	0.5	0	0.75	1.5	2.25
0x100060E0:	3	3	3	3	3	2.25	1.5	0.75
0x10006100:	0	0	0	0	0	0	0	0
0x10006120:	0	0	0	0	0	0	0	0
0x10006140:	0	0	0	0	0	0	0	0
0x10006160:	0	0	0	0	0	0	0	0
0x10006180:	0	0	0	0	0	0	0	0
0x100061A0:	0	0	0	0	0	0	0	0
0x100061C0:	0	0	0	0	0	0	0	0
0x100061E0:	0	0.75	1.5	2.25	3	2.75	2.5	2.25
0x10006200:	2	1.5	1	0.5	0	0.75	1.5	2.25
0x10006220:	3	3	3	3	3	2.25	1.5	0.75
0x10006240:	0	0	0	0	0	0	0	0
0x10006260:	0	0	0	0	0	0	0	0
0x10006280:	0	0	0	0	0	0	0	0
0x100062A0:	0	0	0	0	0	0	0	0
0x100062C0:	0	0	0	0	0	0	0	0
0x100062E0:	0	0	0	0	0	0	0	0
0x10006300:	0	0	0	0	0	0	0	0
0x10006320:	0	0.75	1.5	2.25	3	2.75	2.5	2.25
0x10006340:	2	1.5	1	0.5	0	0.75	1.5	2.25
0x10006360:	3	3	3	3	3	2.25	1.5	0.75
0x10006380:	0	0	0	0	0	0	0	0
0x100063A0:	0	0	0	0	0	0	0	0
0x100063C0:	0	0	0	0	0	0	0	0
0x100063E0:	0	0	0	0	0	0	0	0
0x10006400:	0	0	0	0	0	0	0	0

0x10005F00부터 0x10006040전까지 줄은 76번째 행이므로 이미 수평 보간을 진행하는 과정에서 채워졌으나, 0x10006040부터 0x10006400까지 행들(77, 78, 79행)은 Padding이 채워져야하는 부분이므로 76줄의 행을 그대로 복사하여 Padding값을 채웠다. 이 과정으로 80x80의 값이 모두 채워졌으므로 0x10006400부터는 값이 없다.

4. Discussion and Conclusion

과제를 수행하며 느낀 점, 개선 방향, 과제 수행 중 발생한 문제점, 또는 과제에서 잘못 주어진 오류 등에 대한 분석과 결론을 작성합니다.

이번 과제에서 구현한 코드를 완성한 후 프로그램을 실행해본 결과, 아무런 숫자도 나오지 않고 검은색 바탕만 나오는 사진이 만들어졌다. 이전에 한번 프로그램이 수정되었고, 모두 0이 채워져 있는 .hex 파일을 가지고도 PSNR 수치가 나와서 내 코드에 이상이 있다고 판단하고 메모리를 확인하여 보간의 문제점을 찾아보았으나 찾지 못했다. 따라서 내가 구현한 Bilinear interpolation 알고리즘에 문제가 있다고 판단하여 새로운 코드를 다시 만들어서 프로그램을 동작시켰으나, 똑같은 결과가 나타났다.

이번 프로젝트 github의 issue탭에 다른 학우께서 .hex 파일을 이용하여 png를 만드는 프로그램이 Little-endian을 인식하지 못하는 에러가 있고, 다른 코드를 이용하면 된다고 올려주셔서 그 코드를 가지고 파이썬 코드를 바꾸어 프로그램을 동작시켜보니 올바른 결과가 나타났다.

과제에 대한 bug를 빠르게 인식하고 해결해주시면 원할하게 과제를 수행하는데에 큰 도움이 될 것이라고 생각했다.

(Optional) Acknowledgement

컴퓨터정보공학부 21학번 김경목: 보간을 하는 과정에서 두 값을 더하고 평균을 구하는 방식으로 구현했는데, Floating point의 나눗셈을 사용하지 않고 두 값을 더한 결과의 EXP부분을 -1하면 /2의 효과를 얻는다는 사실을 설명해주어서, 해당 방법을 사용하여 이번 프로젝트에서는 Floating point의 add 연산만을 사용하여 코드를 구성했다.

Reference

어셈블리프로그램 설계 및 실습-실습강의자료 6. Floating_Point

어셈블리프로그램 설계 및 실습-실습강의자료 8. Floating_Point_Multiplication

어셈블리프로그램 설계 및 실습-실습강의자료 9. Term project – Bilinear interpolation

명예서약서 (Honor code)

“꿈을 가지십시오. 그리고 정열적이고 명예롭게 이루십시오.”

본인은 [어셈블리프로그램설계및실습] 수강에 있어 다음의 명예서약을 준수할 것을 서약합니다.

1. 기본 원칙

- 본인은 공학도로서 개인의 품위와 전문성을 지키며 행동하겠습니다.
- 본인은 안전, 건강, 공정성을 수호하는 책임감 있는 공학도가 되겠습니다.
- 본인은 우리대학의 명예를 지키고 신뢰받는 구성원이 되겠습니다.

2. 학업 윤리

- 모든 과제와 시험에서 정직하고 성실한 태도로 임하겠습니다.
- 타인의 지적 재산을 존중하고, 표절을 하지 않겠습니다.
- 모든 과제는 처음부터 직접 작성하며, 타인의 결과물을 무단으로 사용하지 않겠습니다.
- 과제 수행 중 받은 도움이나 협력 사항을 보고서에 명시적으로 기재하겠습니다.

3. 서약 이행

본인은 위 명예서약의 모든 내용을 이해하였으며, 이를 성실히 이행할 것을 서약합니다.

학번 | 2021202085 | 이름 | 전한아솔 | 서명 | 전한아솔