어셈블리프로그램 설계 및 실습

Term project 제안서

제출일자: 2018년 11월 13일 (화)

학 과: 컴퓨터정보공학부

담당교수: 이형근 교수님

학 번: 2017202087

성 명: 홍 세 정

1. 과제제목

임의의 부동소수점 데이터로 이루어진 20x20 정방행렬에 대해 Bilinear interpolation을 수행하여 행과 열을 각 4배로 확장한 80x80 정방행렬을 계산하라.

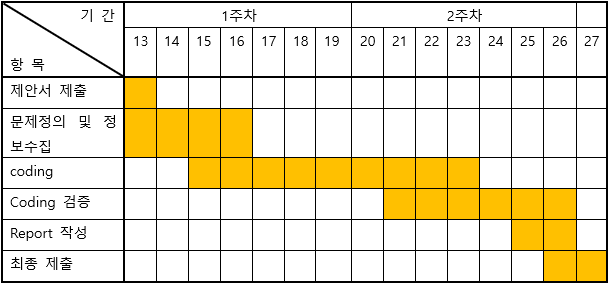
1. 과제목표

Bilinear interpolation(이중선형 보간)을 이용하여 행렬을 확장할 수 있다.

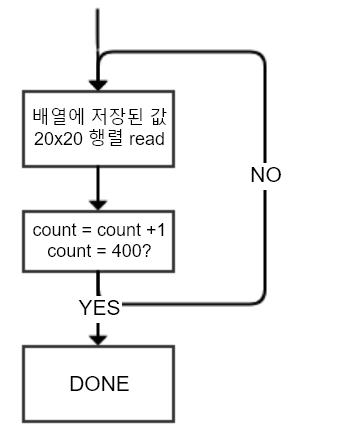
Linear interpolation의 수식을 알고 이해하며 가운데 값을 추정할 수 있다.

성능이 좋은 코드를 구현할 수 있다.

1. 일정



1. 각 function 별 알고리즘

Function

1. Data read

Label : Matrix\_data에 선언되어 있는 -2000 ~ 2000 사이의 부동소수점 값의 데이터들을 읽어온다.

배열에 저장된 값 행렬 20x20을 register로 하나씩 읽어온다.

20x20의 행렬의 숫자 개수는 총 400개

초기 값 count=0, 하나의 행렬의 값을 읽을 때 마다

count는 1씩 증가하게 된다.

Count가 400이 되면 data read함수는 끝.

1. Data store

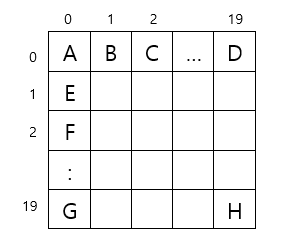
20x20 정방행렬을 행과 열을 각 4배로 확장한 80x80의 행렬을 저장해야 한다.

Bilinear interpolation을 완료한 80x80 행렬의 각 숫자 6400개의 값들을 저장.

데이터를 저장할 때에는 1word 단위로 저장하고, 0x60000000 번지로 통일한다.

Label : Result\_data는 Matrix\_data의 원소들을 이용하여 결과값이 저장되는 데이터 공간의 시작 주소 값이다.

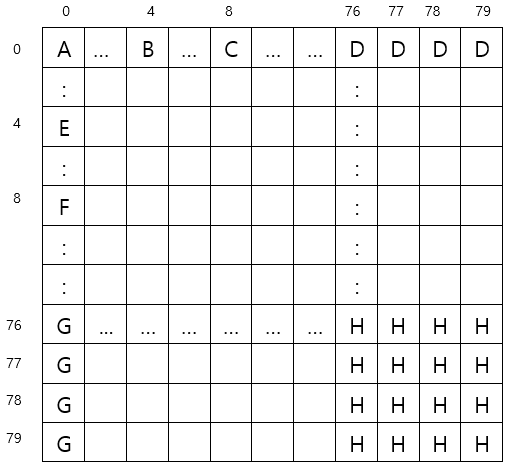
1. Bilinear interpolation



임의의 20x20 행렬이다.

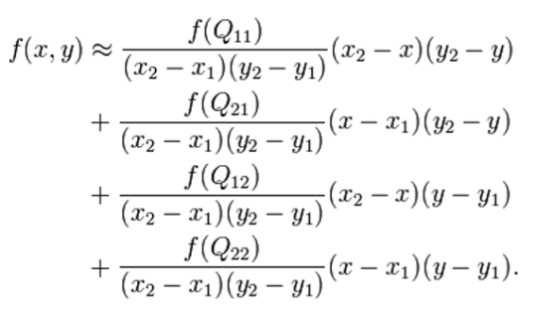
4배로 확장했을 경우 원래 2개의 행렬 사이에 각 3개의 수가 더 들어가야하고 4개의 행렬 사이에는 각 9개의 수가 들어가야 한다. 그렇게 해서 만들어진 행렬은 76x76 행렬이다. Interpolation이 불가능한 외각에 추가된 값들은 현재 위치와 가장 가까운 값으로 복사한다.(=padding)

Bilinear interpolation한 값이 밑에 그림과 같다.



20x20 행렬을 4배 확장

80x80 행렬



각 Bilinear interpolation 수식에 맞게 알맞은 수를 추정하여 값을 넣을 수 있다.

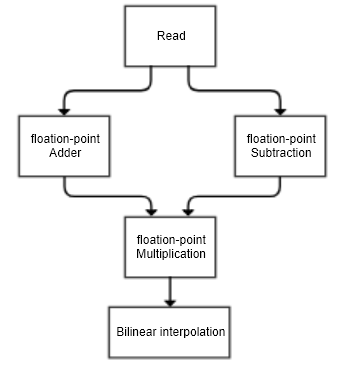
20x20에서 근접해 있는 4개의 수를 받아 5x5행렬을 361개 만들 수 있다. 4의 배수에서는 수가 서로 겹치므로 생략하고 이어 붙여 77x77행렬을 만들 수 있다.

1. Padding

Bilinear interpolation에서 확장할 때 0~76까지는 확장이 가능하지만 그 77~79까지의 외각은 interpolation 수식으로 수의 추정이 불가능하다. 따라서 그 외각의 값들은 가장 가까운 값들로 복사된다. 위의 그림처럼 x: 77~79, y:0 까지는 D, x:0, y: 77~79 까지는 G, x:76~79, y:76~79까지는 H로 padding할 수 있다. 76번째 값을 확인하고 알맞은 수를 추정하여 행렬의 수를 정할 수 있다.

1. Floating-point Adder, Subtraction, multiplication

Bilinear interpolation할 때 floating-point adder, sub, multiplication이 수행 가능해야 한다.



읽어온 값들을 알맞게 Adder, Subtraction하여 multiplication한다.

여기서 Adder와 Subtraction은 전에 했던 실습을 이용한다.

Multiplication도 adder처럼 함수를 만들어 줄 수 있다.

1. 예상되는 문제점

Multiplication에서 mantissa값을 각 24bit와 24bit를 곱해서 결과값 48bit가 나오게 된다. 하지만 어셈블리어에서 사용하는 register은 32bit이다. 그러므로 result 48bit의 값을 두개의 register에 나눠서 저장할 수 있어야한다.

Multiplication에서 MUL 명령어 사용하는 부분을 없애야 한다.

80x80의 행렬을 저장할 때 겹치는 부분을 잘 파악하여 없애고 행과 열의 순서를 맞춰서 저장할 수 있어야 한다.

Bilinear interpolation의 식을 잘 파악하고 Adder와 Subtraction, Multiplication의 순서를 잘 파악하여 수를 추정해야 한다.

1. 검증전략

처음에 정수의 값들로 행렬을 구성하여 정수의 값이 나오는지 확인한다.

각 행렬 사이에 어떤 값이 들어가야 되는지 계산해보고 구현한 함수가 정확한 값이 나왔는지 확인한다.

Multiplication에서 곱해진 값이 정확히 나왔는지 검증한다.

Padding한 결과 값이 잘 나오는지 확인한다.

최종 80x80 정방행렬이 알맞은 수가 들어 갔는지 확인한다.