데이터 재사용을 이용한 프랙탈 영상압축을 위한 효율적인 일차원 VLSI 어레이

이희진 · 이수진 · 우종호

부경대학교

Efficient One-dimensional VLSI array using the Data reuse for Fractal Image Compression

Hee-jin Lee · Su-jin Lee · Chong-ho Woo

Pukyong National University

E-mail : hjin@mail1.pknu.ac.kr, sjlee@woongbi.pknu.ac.kr, chwoo@pknu.ac.kr

요 약

본 논문에서는 프랙탈 영상압축 알고리즘의 고속처리를 위한 효율적인 일차원 VLSI 어레이를 설계한다. 프랙탈 영상압축 알고리즘은 영상을 정의역블럭과 치역블럭을 나누고, 블럭간의 자기유사성을 비교하여 데이터를 압축시킨다. 이때 정의역블럭은 치역블럭의 두배크기로 인접한 정의역블럭과 50% 중첩시켜 분할한다. 인접한 두 정의역 블럭의 픽셀들이 중첩되므로, 이 픽셀들을 재사용하여 데이터의 입력수를 줄이고 처리속도를 향상시킬수 있다. 이 결과 최소한의 레지스터와 MUX, DEMUX의 추가만으로 약 25%의 처리속도 향상을 얻을 수 있다.

ABSTRACT

In this paper, we designed one-dimensional VLSI array with high speed processing in Fractal image compression. Fractal image compression algorithm partitions the original image into domain blocks and range blocks then compresses data using the self similarity of blocks. The image is partitioned into domain block with 50% overlapping. Domain block is reduced by averaging the original image to size of range block. VLSI array is trying to search the best matching between a range block and a large amount of domain blocks. Adjacent domain blocks are overlapped, so we can improve of each block's processing speed using the reuse of the overlapped data. In our experiment, proposed VLSI array has about 25% speed up by adding the least register, MUX, and DEMUX to the PE.

키워드

프랙탈, 재사용, 시스톨릭 어레이, VLSI

Ⅰ. 서 론

컴퓨터 시스템의 성능이 향상됨에 따라 정보의 효율적인 표현을 위해 문자뿐만 아니라 이미지, 동영상 등의 멀티미디어 데이터를 많이 이용한다. 그러나 이러한 영상데이터는 문자데이터에 비해 많은 용량을 필요로 한다. 따라서 영상데이터의 효율적인 저장과 전송을 위해 영상압축기술이 필수적이다.

프랙탈 영상압축은 영상을 구성하는 각 부분의 자기 유사성을 이용하여 데이터의 양을 압축하는 방법이다. 프랙탈 영상압축기법은 코드북이 필요없으며, 압축율이 높고 복호화 속도가 빠르다. 또한 해상도 독립성을 가지며 다른 영상의 부호화 기술과의 조합 등의 우수한 특성을 지닌다.[1-4]

그러나 프랙탈 영상압축 알고리즘은 부호화과정에서 정의역블럭과 치역블럭간 비교에 많은 계산량을 필요로 하는 단점을 갖는다. 따라서 순차머신으로 프랙탈 압축기법을 실시간에 가깝게 처리하는 것은 불가능하다.

각 블럭들에 대한 정합여부를 결정하는 프랙탈 영상

압축 알고리즘은 데이터 흐름이 규칙적이고 병렬성이 대단히 높다. 이러한 병렬성을 이용하여 대량의 데이터를 병렬로 처리할 수 있는 VLSI 어레이를 구현하여 프랙탈 영상압축을 고속으로 처리할 수 있다.

기존의 연구에서 프랙탈 영상압축을 위한 일차원 VLSI 어레이를 제안했다.[5] 고정블럭 분할 방식의 알고리즘을 블럭의 크기에 따라 반복적용시킴으로서 화질을 유지하면서 압축율이 높은 쿼드-트리 분할방식의 효과를 얻었다. 또한 치역블럭과 정의역블럭의 입력된 핀을 공유하여 데이터 입력핀의 수를 줄였다. 그러나 인접한 두 정의역블럭에 공통으로 사용하는 데이터들을 재사용하지 않고, 재입력받아 연산하는 단점이 있다. 본 논문에서 인접한 두 정의역블럭의 중첩되는 데이터를 재사용하여 하드웨어의 추가비용이 거의 없이 처리속도를 약 25% 향상시키는 일차원 VLSI 어레이를 제안한다.

II. 프랙탈 영상압축을 위한 VLSI 어레이

1. 프랙탈 영상압축을 위한 일차원 시스톨릭 어레이 구조

프랙탈 영상압축을 위한 Fisher의 순차 알고리즘은 원영상을 치역블럭과 정의역블럭으로 나누고, 각 치역블럭에 대해 자기 유사성이 가장 큰 최적의 정합된 정의역블럭을 찾기 위해서 전체 영역을 탐색한다.[4]

블럭사이의 유사성 정도를 표현하기 위해 식(1)과 같은 rms(root mean square)를 이용한다.

$$rms = \sum_{i=1}^{B^2} (s \cdot a_i + o - b_i)^2 \qquad (1)$$

정의역블럭과 치역블럭과의 rms가 최소가 되는 블럭의 쌍을 정합되는 블럭으로 선택한다. 이때의 IFS(Iterated Function System)를 구성하는 식(2)의 s와 식(3)의 o를 압축의 결과로 저장한다.

$$s = \frac{n\sum_{i=1}^{n} D_i R_i - \sum_{i=1}^{n} D_i \sum_{i=1}^{n} R_i}{n\sum_{i=1}^{n} D_i^2 - (\sum_{i=1}^{n} D_i)^2} \qquad (2)$$

$$o = \frac{\sum_{i=1}^{n} R_i - s\sum_{i=1}^{n} D_i}{n} \qquad (3)$$

식(1)에서 유도된 일차원 VLSI 어레이는 그림 1의 구조를 갖는다.[5] 그림 1의 일차원 시스톨릭 어레이는 정의역블럭 및 치역블럭의 픽셀값을 누적하기 위한 PE0~PE15의 16개의 처리요소, s를 연산하고 양자화를 위한 PE16~PE18, o를 연산하고 양자화를 위한 PE19~PE21, 그리고 mse를 연산하고 최적의 블럭을 선택하기 위한 PE22~PE24로 총 25개의 처리요소들로 구성된다.

PE0~PE15는 블럭크기, 치역블럭의 픽셀값, 정의역블럭의 픽셀값을 입력받는 핀들이있다. 이 입력핀들을 공유하여 입력핀의 수를 줄이고 있다. 따라서 치역블럭과 정의역블럭의 픽셀값을 번갈아 입력한다.

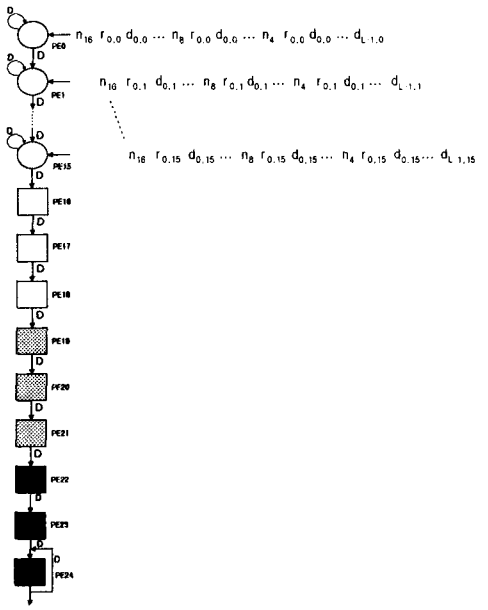

그림 1. 일차원 시스톨릭 어레이

2. 데이터 재사용을 이용한 일차원 시스톨릭 어레이

그림 1의 구조에서는 인접한 정의역블럭에 공통으로 포함된 데이터를 재사용하지 않는다. 따라서 같은 값을 다시 입력해서 계산하는 단점이 있다.

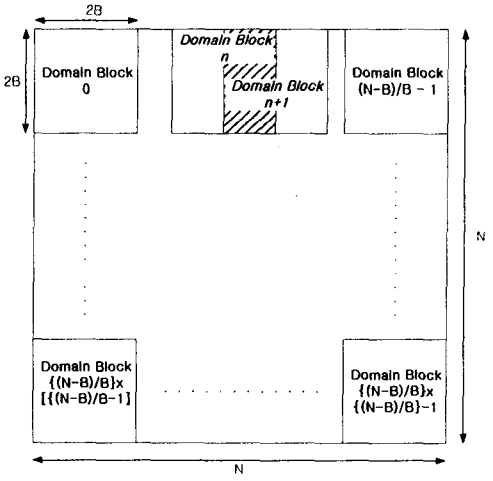

그림 15 정의역블럭

그림 2는 영상의 크기가 N×N이고 블럭크기가 B×B일 때 영상에서 정의역블럭을 분할한 것이다. 정의역블럭은 원 영상을 블럭크기의 2B×2B로 50%씩 중첩하여 분할하고, B×B 크기로 축소한 픽셀값들이므로 그림 2와 같이 임의의 정의역블럭은 인접한 정의역블럭과 블럭크기 1/2의 데이터를 공유한다. 그러므로 중복되는 정의역블럭의 데이터를 재사용함으로써 입력수를 줄일 수 있고 전체 처리 시간을 줄일수 있다.

그림 2에서 빗금친 부분은 처리요소내부의 레지스터에 저장해 두어 재사용한다.

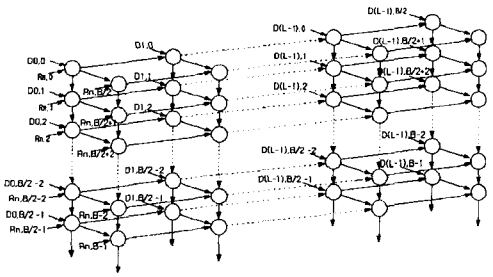

그림 3. m번째 치역블럭에 대한 정의역블럭의 데이터의존그래프

그림 3은 m번째 치역블럭과 정의역블럭에 대한 데이터 의존그래프이다. n번째 정의역블럭의 (0, B/2)~(B-1, B-1)에 해당하는 영역은 n+1번째 정의역블럭 (0,0)~(B-1,B/2-1)과 같은 값을 갖는다. n번째 정의역블럭의 (0, B/2)~(B-1, B-1)의 값을 한번 입력한 것으로 n+1번째 정의역블럭에 대한 연산까지 수행한다. 따라서 정의역블럭의 데이터를 우선 입력받고 대응되는 치역블럭의 데이터를 각각 입력받는 것으로 전체 데이터의 입력횟수를 감소시킨다.

데이터의 재사용을 위해서 각 처리요소에는 정의역블럭의 데이터를 저장하기 위한 레지스터가 추가되어야 한다. PE15에서는 n번째와 n+1번째 정의역블럭에 대한 연산하고 누적된 값을 저장하는 각각의 레지스터가 존재해야 한다.

그림 4는 정의역블럭의 데이터를 재사용한 어레이의 구조와 데이터들의 입력순서를 나타낸 것이다. 먼저 블럭의 크기를 입력받고 정의역블럭의 데이터를 입력받아 레지스터에 저장한다. 1/2로 분할된 치역블럭의 값들을 차례로 입력 받아서 연산한다. 그러므로 데이터는 정의역블럭, 치역블럭(0, B/2)~(B-1, B-1), 치역블럭(0,0)~(B-1,B/2-1)의 각각 한 픽셀씩 순차적으로 입력된다.

III. 처리요소의 설계

본 논문에서 제안한 어레이의 기본구조는 기존

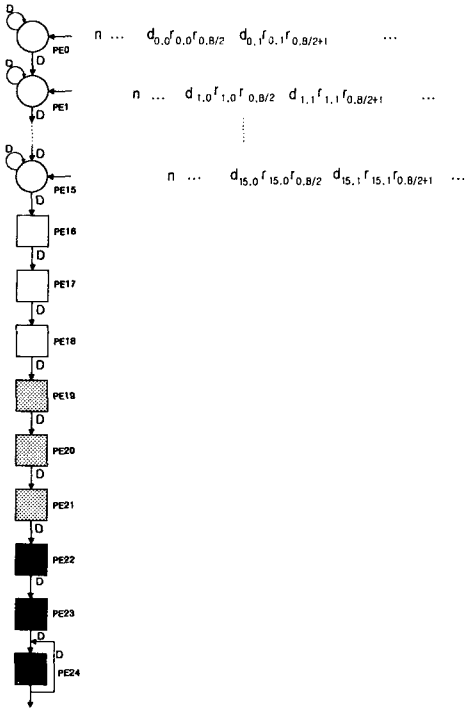

그림 17. 데이터를 재사용한 일차원 시스톨릭 어레이

의 구조와 유사하다.[5] 제안한 일차원 시스톨릭 어레이는 정의역 및 치역의 픽셀값과 제곱의 값을 각각 누적하는 16개의 처리요소와 s, o, mse를 연산하고 최적의 값을 기억하는 9개의 처리요소로 구성된다.

PE0~PE14에는 호스트 컴퓨터와 이전의 처리요소로부터 값을 전달받고 연산하여 그 결과값을 각각의 레지스터에 저장한다. 치역블럭과 정의역블럭의 데이터를 재사용하기 위한 레지스터가 각각 추가된다.

그림 6은 PE15의 내부구조를 나타낸다. 16개의 처리요소로 16×16, 8×8 블럭크기에 대한 연산도 수행해야 하므로 PE15에서 여러번 연산값을 누적한다. PE15는 데이터를 재사용하기 위한 레지스터, 정의역블럭과 치역블럭을 연산하는 장치, MUX, DEMUX, 그리고 그 결과를 해당 정의역블럭에 따라 저장하는 레지스터들로 구성된다.

PE15에서는 PE14로부터 전송된 연산값들과 호스트 컴퓨터로부터 전달받은 값들을 연산하고, 기존의 연산결과를 현재의 연산결과와 누적한다. 연산결과들은 하나의 정의역블럭과 인접한 정의역블럭의 값이 교대로 입력이 되므로, 이들 결과값은 각각의 레지스터에 분리해서 저장해야 한다. 그러므로 기존 어레이의 PE15에 비해 연산 결과값을 저장하는 레지스터(T1~T5)가 추가된다.[5] 치역블럭에 대응되는 정의역블럭의 연산값들이 DEMUX를 통해서 S1~S5 레지스터 또는 T1~T5 레지스터에 저장되고, 한 블럭에 대한 연산이 완

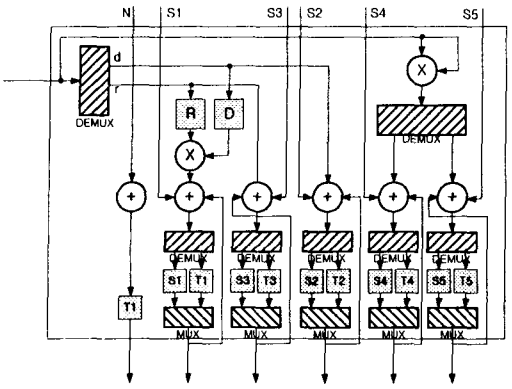

그림 6. PE15

전히 끝날 때까지 MUX를 통해서 해당 정의역블럭에 따라 S1~S5 또는 T1~T5에 누적된다. 이 값들은 s와 o를 연산하기 위한 처리요소로 전달된다.

Ⅳ. 결과 및 고찰

데이터를 재사용하지 않은 그림1의 구조와 정의역블럭의 중첩되는 데이터를 재사용한 그림4의 구조에 입력되는 데이터의 입력횟수는 각각 식(4)와 식(5)와 같다. 이때 N은 영상의 크기를 나타내며, B는 치역블럭의 크기를 나타낸다.

$$\left[\frac{B^2}{16}\times 2\times (N-B)/B\right]\times (N-B)/B \quad (4)$$

$$\left[\frac{B^2}{32}\times 3\times ((N-B)/B+1)\right]\times (N-B)/B \quad (5)$$

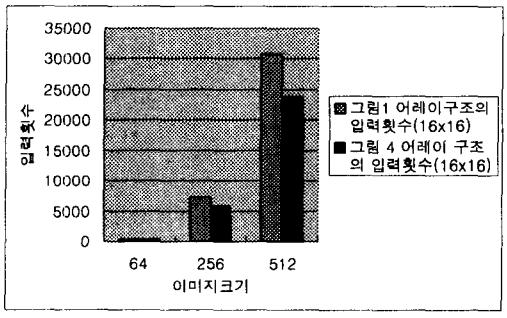

그림 7. 그림 1의 어레이 구조와 그림 4의 어레이 구조의 입력횟수 비교

512×512, 256×256, 64×64 각 영상에 대한 16×16 치역블럭을 이용할 경우 두 구조의 영상압축을 위한 데이터 입력횟수는 그림 7과 같다.

표 1은 비교할 연산의 수가 N일 때 두 구조를 이용한 경우 연산시간을 비교한 것이다. 중첩하는 정의역블럭의 데이터를 재사용할 경우 약 25%의 성능향상이 있는 것을 알 수 있다.

표 1. 어레이의 성능 비교 (N: 연산 횟수)

	데이타 재사용하지 않은 경우	데이타를 재사용한 경우
처리시간	{25+2×16×16×(N-1)}	{25+3×16×8×N}
상대적 처리시간 T_R	1	0.75

Ⅴ. 결 론

본 논문에서는 프랙탈 영상압축 알고리즘의 고속처리를 위한 일차원 VLSI 어레이를 설계하였다. 제안한 VLSI 어레이는 인접한 정의역블럭이 50%가 중첩되는 점을 이용하여 처리능력을 향상시켰다. 중첩되는 정의역블럭의 데이터들을 재사용하므로써 전체 연산에 필요한 데이터의 입력수를 감소시키고 전체 처리 시간을 줄였다. 어레이를 구성하는 처리요소의 일부에 소수의 레지스터, MUX, DEMUX의 추가만으로 전체 처리 성능을 약 25%정도 향상시켰다.

본 논문의 연구결과는 프랙탈 영상압축의 실시간 처리를 위한 VLSI 시스템의 구현에 활용될 수 있을 것이며, 향후 어레이를 구성하는 회로를 간략화하는 연구를 진행할 것이다.

참고문헌

[1] K. P. Aiken, M. J. Irwin, and R. M. Owens, "A parallel ASIC architecture for efficient fractal image coding," Journal of VLSI Signal Processing 19, pp. 97-113, 1998.

[2] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," Elsevier Science Publishers B. V, Signal Processing, Vol. 29, pp. 251-263, 1992.

[3] A. E. Jacquin, "Fractal image coding: A review," Proceedings of the IEEE, Vol. 81, pp.1451-1465, Oct., 1993.

[4] Y. Fisher, "Fractal image compression: Theory and application," Springer-Verlag, Berlin, 1995.

[5] 성길영, 우종호, "고속 프랙탈 영상압축을 위한 최적의 파이프라인 주기를 갖는 VLSI 어레이 구조 설계", 한국통신학회 논문지, 제25권, 제5A호, pp. 702-708, May, 2000