論文95-32B-11-6

단위 처리기를 조기 은퇴시키는 완전탐색 블록정합 알고리듬

(A Full-Search Block-Matching Algorithm with Early Retirement of Processing Elements)

南 基 哲 *, 蔡 洙 翊 *

(Ki-Chul Nam and Soo-Ik Chae)

요 약

본 논문에서는 빠른 움직임 추정을 위해 1 차원 시스톨릭 어레이에 적용될 수 있는 단위 처리기를 조기 은퇴시키는 완전탐색 블록정합 알고리듬을 제안한다. 제안하는 알고리듬에서는 각 단위 처리기에서 현재까지의 차의 절대치 누적합이 이미 구한 최소값보다 같거나 크면, 그 단위 처리기는 조기 은퇴된다. 만약 모든 단위 처리기가 은퇴되면, 탐색 영역에서 현재 배열 위치의 연산은 끝내고 다음 배열 위치의 새로운 연산을 시작한다. 이 방법으로 전체 연산을 하지 않고도 최적의 움직임 벡터를 구할 수 있으며, 전체 연산 싸이클 수를 약 60%로 줄이고 단위 처리기에서의 전력 소모도 40~60% 정도로 줄일 수 있음을 모의실험을 통하여 확인하였다.

Abstract

In this paper, we propose a full-search block-matching algorithm with early retirement, which can be applied to a 1-D systolic array of processing elements (PE's) for fast motion estimation. In the proposed algorithm, a PE is retired when its current accumulated sum is equal to or larger than the current minimum MAD. If all PE's are retired, the MAD calculation is stopped for the current array position and is started for the next one in the search window. Simulation results show that the optimum motion vector is always found with less computation, the total computation cycles for motion estimation are decreased to about 60%, and the power dissipation in the PE's is reduced to about 40~60%.

I. 도 입

화상회의 또는 고화질 TV 등의 동영상에서는, 연속적인 프레임 사이의 상관관계는 매우 크며, 이 상관관계는 움직임 추정을 통해서 매우 효율적으로 줄일 수 있다.[1] 움직임 추정에는 한 프레임을 블록 단위로 분할하여 각 블록 단위로 가장 유사한 블록을 찾아 움직임 벡터를 구하고 이를 부호화하는 블록 정합 알고리듬(Block Matching Algorithm)과 화소 단위로 움직임 벡터를 구하는 재귀 화소 알고리듬(Pel-Recursive Algorithm)이 있다. MPEG2 부호화기 등 움직임 추정을 실시간 처리하기 위하여 배열 구조(array architecture)를 갖는 블록 정합 알고리듬이 주로 사용되고 있다.[2-4]

* 正會員, 서울大學校 半導體共同研究所 및 電子工學科
(Inter-Univ. Semiconductor Research Center & Dept. of Elec. Eng. Seoul Nat. Univ.)

接受日字: 1995年 5月10日, 수정완료일: 1995年10月31日

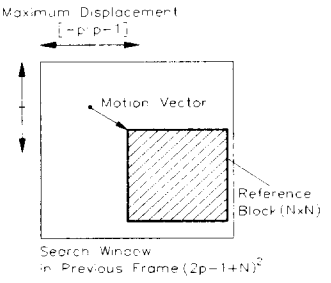


그림 1. 블록 정합 알고리듬
Fig. 1. Block Matching Algorithm.

블록 정합 알고리듬은 최적의 움직임 벡터를 찾기 위한 판단 기준으로 하드웨어 구현이 비교적 간단한 차의 절대치 평균(mean-of-the-absolute-differences)을 사용한다. 이것은 다음 수식과 같이 표현하며, 본 논문에서는 이제부터 "차의 절대치 합"이라 칭한다.

$$MAD(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{N}|a(i,j)-b(i+u,j+v)|, \quad -p \le u,v < p$$

여기서 a(i,j)는 (i,j)에 위치하는 NxN 블록 크기의 기준 블록(Reference Block)이고, b(i,j)는 이전 프레임의 탐색 영역(Search Window)내 후보 블록(Candidate Blocks)들을 나타낸다. p는 기준 블록의 위치로부터 이전 프레임에서의 탐색 영역을 제한하는 최대 변위를 의미하며, 이 차의 절대치 합이 최소가 되는 위치를 움직임 벡터(Motion Vector)라 정의한다(그림 1). 따라서 이전 프레임에서 탐색 영역의 범위는 $(2p-1+N)^2$이고, 후보 블록들의 수는 $q=(2p)^2$로 계산된다.

블록 정합 알고리듬은 완전 탐색 알고리듬(Full Search Algorithm)과 계층적인 탐색 알고리듬(Hierarchical Search Algorithm)으로 구분할 수 있다.[5] 완전 탐색 알고리듬에서는 후보 블록을 완전 탐색을 하므로 많은 연산이 필요하다는 문제가 있으나, 화질이 좋으며 규칙적인 데이타 흐름을 시스톨릭 어레이에 적용하여 병렬·파이프라인 처리를 할 수 있으므로 하드웨어 구현이 용이하다. 반면, 3-단계 탐색(3-Step Search)과 같은 계층적인 탐색 알고리듬에서는 계층적으로 추정 위치를 좁혀가므로 후보 블록의 수를 줄여 빠른 속도를 가지지만, 규칙적인 데이타의 흐름이 깨져 병렬·파이프라인 처리를 위한 복잡한 제어가 필요하다. 그리고 국부적인 최소점(local minimum)에 빠질 가능성이 있으므로 완전 탐색 알고리듬에 비하여 화질이 저하된다.

본 논문에서는 많은 연산량을 줄이기 위하여 단위 처리기를 조기 은퇴시키는 완전 탐색 알고리듬을 제안한다. II장에서는 기존의 알고리듬 구조를 고찰하고 III장에서는 제안하는 알고리듬을 설명한다. IV장에서 기존 알고리듬 구조와 제안한 구조를 모의 실험을 통해 비교하며, 마지막으로 V장에서는 결론을 맺는다.

II. 기존의 알고리듬

연산량이 매우 많은 완전 탐색 알고리듬을 하드웨어로 구현하기 위해서는 뺄셈, 절대치 그리고 누적 등의 명령들을 가진 단위 처리기(Processing Elements)를 기본 셀로 한 배열 구조를 이용한다.[2] 이 배열 구조에는 단위 처리기의 효율성을 극대화시키고 병렬 처리가 가능하도록 시스톨릭 어레이를 이용한 구조가 널리 사용되고 있다.

참고문헌 [3]에서 Yang은 N개의 단위 처리기를 갖는 1 차원 시스톨릭 어레이를 사용하여 병렬처리가 가능하도록 움직임 추정기를 설계하였고, 이 움직임 추정기의 데이타 흐름과 효율성을 설명하였다. 한 기준 블록의 움직임 벡터를 찾기 위해 "차의 절대치 합 구하기"와 "비교하기"로 구분하여, "차의 절대치 합 구하기"에서 각 단위 처리기는 기준 블록과 탐색 영역내의 한 후보 블록에 대한 차의 절대치 합을 연산하고, N개의 단위 처리기가 서로 이웃한 위치의 차의 절대치 합을 동시적으로 구한다. "비교하기"에서는 연산이 끝난 단위 처리기부터 그 결과를 순차적으로 비교하여 차의 절대치 합의 최소값과 그때의 움직임 벡터를 찾게 된다. 블록 크기가 16^2(N=16), 추정할 최대 변위가 8(p=8)이고 16개의 단위 처리기를 사용하였을 경우, i 번째 단위 처리기는 i 싸이클부터 16x16+i 싸이클 동안 차의 절대치 합이 계산되고 0번째 단위 처리기로부터 계산된 결과를 순차적으로 출력하여 비교하게 된다. 이 과정으로 입력의 지연(latency)을 포함하여 한 블록을 처리하는데 소요되는 시간은 (N^2+1)xN 싸이클이다.

참고문헌 [4]에서는 2 차원 시스톨릭 어레이를 이용한 경우, 탐색 영역 내에서 q개 후보 블록들의 차의 절대치 합을 연속적으로 계산하면서 연산이 끝난 차의

절대치 합부터 차례로 비교하면서 최소값을 갖는 움직임 벡터를 찾게 된다. 블록 크기가 16^2, 추정할 최대 변위가 8이고 16^2개의 단위 처리기를 사용하였을 경우, 입력의 지연을 포함하여 한 블록을 처리하는데 소요되는 시간은 $(N+2p-1)^2+1$ 싸이클이다. 표 1에서는 1개의 단위 처리기, 1 차원 시스톨릭 어레이 그리고 2 차원 시스톨릭 어레이를 이용한 경우에 대하여 요약하였다.

표 1. 움직임 추정기의 비교 (N=16, p=8 그리고 q=$(2p)^2$)

Table 1. Comparison of motion estimators (N=16, p=8 and q=$(2p)^2$).

	한 개의 단위 처리기	기존의 1-D systolic array[2]	기존의 2-D systolic array[4]
기준 블록의 크기 N^2	16×16	16×16	16×16
탐색영역의 크기$(2p-1+N)^2$	31×31	31×31	31×31
추정 최대 변위(p=8)	-8~+7	-8~+7	-8~+7
단위 처리기의 수	1	16	16×16
한 블록을 처리하는데 소요되는 싸이클 수	$N^2\times q$ =65,536	$(N^2+1)\times N$ =4,112	$(N+2p-1)^2+1$ =962

1 차원 시스톨릭 어레이를 이용한 구조는 매우 높은 효율에도 불구하고, 완전 탐색 알고리듬에서의 많은 연산으로 인한 처리 속도(throughput)가 제한되며, 2 차원 시스톨릭 어레이를 이용한 구조는 높은 처리 속도와 효율을 가지지만 면적의 낭비가 크다는 단점이 있다.

Ⅲ. 조기 은퇴 알고리듬

II 장에서 움직임 벡터를 찾기 위해 "차의 절대치 합 구하기"와 "비교하기"로 구분하여 시스톨릭 어레이를 이용하는 방법을 설명하였다. 여기에 비교기의 사용을 최대로 하여 보다 빠르며 전력 소모도 줄일 수 있는 단위 처리기를 조기 은퇴시키는 1 차원 시스톨릭 어레이를 이용한 움직임 추정을 제안한다.

1. 기본 개념

차의 절대치 합의 최소값을 갖는 블록을 찾는 과정에서, 각 단위 처리기에서 현재까지의 차의 절대치 누적합이 이미 구한 최소값과 비교하여 같거나 더 큰 값을 가지면, 그 단위 처리기를 조기 은퇴시킨다. 만약 1 차원 시스톨릭 어레이에서의 모든 단위 처리기가 은퇴되었거나, 기준 블록에 대한 차의 절대치 합 연산이 끝나면, 현재 탐색 영역에서 다음 배열 위치의 차의 절대치 합 연산을 시작한다.

2. 전체 구조

블록 크기가 $16^2(N=16)$, 추정할 최대 변위가 8(p=8)이고 16개의 단위 처리기를 사용하였을 경우, 그림 2에서 보듯이 제안하는 구조는 Yang의 1 차원 시스톨릭 어레이를 이용한 구조에 단위 처리기를 은퇴시킬 수 있도록 단위 처리기 정지 로직(PE-STALL Logic)을 추가하였다. 그리고 단위 처리기가 은퇴되었을 때, 단위 처리기의 연산을 막기 위하여 단위 처리기 정지 신호(PE-STALL signal)가 필요하다. 단위 처리기 정지 신호는 단위 처리기 연산의 대부분을 차지하는 덧셈/뺄셈을 정지시키거나 클록을 차단(clock-gating)하도록 구현될 수 있다. 그림 2에서 기준 블록의 입력데이타 C는 공통 버스를 통해 래치(latch)로 입력되어 매 싸이클마다 다음의 단위 처리기에 전파되어지며, 이전 프레임의 후보 블록들의 입력데이타인 P와 P'은 2 대 1 멀티플렉서를 통해 각 단위 처리기에 의해 선별되어 입력된다.

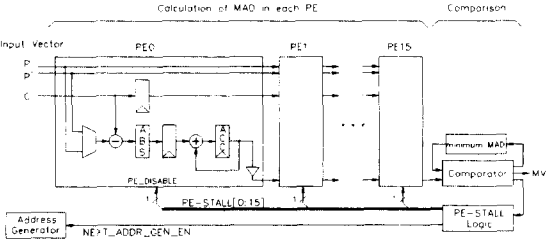


그림 2. 1 차원 시스톨릭 어레이를 이용한 제안하는 구조

Fig. 2. Proposed architecture using 1-D systolic array.

그림 3에서는 1 차원 시스톨릭 어레이의 단위 처리기들이 N개 위치의 한 행을 파이프라인 처리함을 보이며, 차의 절대치 합 연산이 끝난 경우 다음 행으로의 이동을 보인다.

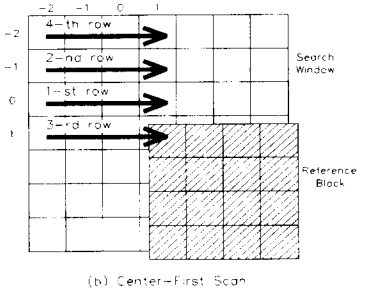


그림 3. 중앙 행에서 멀어지는 스캔 방향 (p=2이고 N=4인 경우)

Fig. 3. Center-first scan direction(p=2 and N=4).

여기서는 조기 은퇴 조건에 적합하도록 스캔 순서가 중앙 행에서 멀어지는 방향으로 잡았으며 IV장에서 그 이유를 언급하겠다. 그리고 한 행에 대하여 입력데이타의 흐름을 그림 4에서 보이며, 매 싸이클마다 비교기로 입력되는 부분 누적합을 나타내기 위해 굵은 선으로 표시하였다. 비교기에서는 16x15 싸이클동안은 단위 처리기를 조기 은퇴시키기 위한 비교를 하고, 마지막 16 싸이클동안은 차의 절대치 합의 최소값을 찾기 위한 비교를 한다. 그림 5에서는 각 단위 처리기가 조기 은퇴되거나 차의 절대치 합의 새로운 최소값을 구하였을 경우의 타이밍도를 Yang의 알고리듬을 이용하였을 경우와 비교하기 위해 나타내었다. 그림 5 (a)에서 보듯이, Yang의 알고리듬은 매 16x16 싸이클마다 16 싸이클동안 각 단위 처리기의 누적합이 순차적으로 비교기로 입력되어 차의 절대치 합의 최소값을 찾는다.

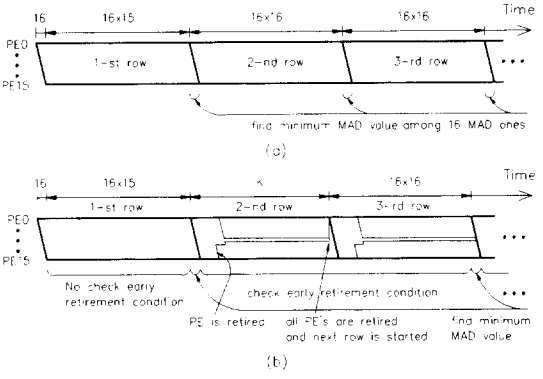


그림 5. 타이밍도 다이어그램: (a) Yang의 알고리듬 (b) 제안하는 알고리듬

Fig. 5. Timing diagram: (a) Yang's algorithm and (b) proposed algorithm.

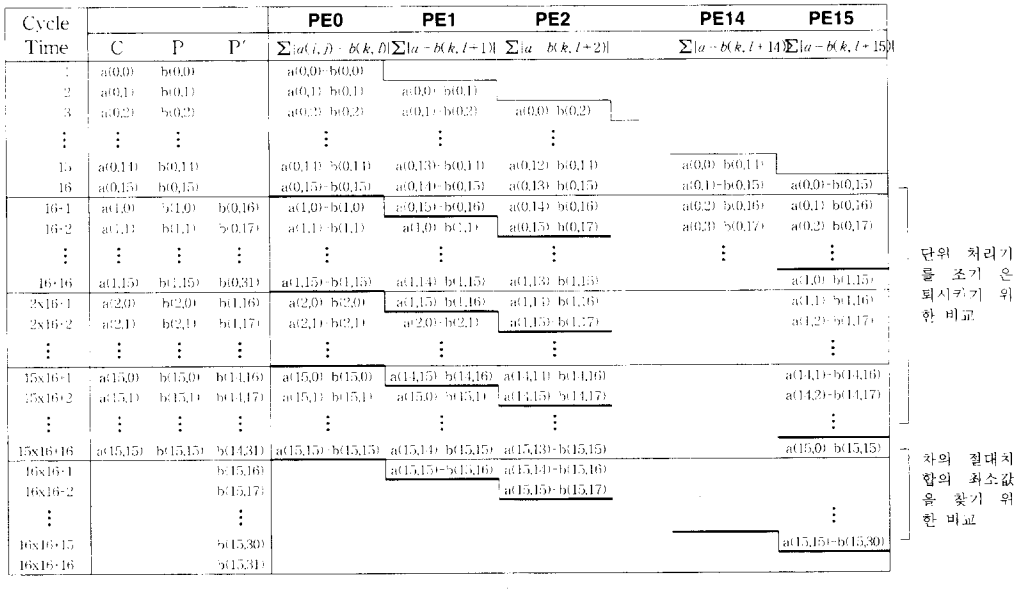


그림 4. 한 행에 대한 배열의 데이타 흐름도 (굵은 선:현재까지의 누적합이 비교기에 입력)

Fig. 4. Data flow in the array for a row (bold line: current accumulated sum is inputted to a comparator).

제안하는 알고리듬에서는 조기 은퇴 조건을 이용하기 위하여 차의 절대치 합의 현재 최소값을 알아야 한다. 그러므로 그림 5 (b)에서 보듯이, 처음 행의 16x16 싸이클동안은 조기 은퇴 조건을 확인하지 않으며 차의 절대치 합을 계산하고, 다음 16 싸이클동안 각 단위 처리기의 누적합이 순차적으로 비교기에 입력되어 최소값을 찾아 레지스터(minimum MAD register)에 저장한다. 단위 처리기의 누적합이 비교기로 입력됨과 동시에 다음 행 연산을 위하여 단위 처리기는 리셋되며 다음 행의 파이프라인 처리가 시작된다. 그리고 다음 16x17 싸이클부터는 조기 은퇴 조건을 확인하면서 차의 절대치 합을 계산한다. 즉, 16x17+i 싸이클에 i 번째 단위 처리기의 부분 누적합이 비교기에 입력되며, 매 16 싸이클마다 반복된다. 비교기에 입력되는 부분 누적합은 차의 절대치 합의 현재 최소값과 비교하여 같거나 더 큰 값을 가지면, 단위 처리기 정지 신호(PE-STALL [i] signal)를 발생하여 이 단위 처리기는 조기 은퇴되며 단위 처리기 내 누적을 위한 레지스터(ACC register)는 리셋된다. 특히 그림 5 (b)의 두 번째 행은 K(<16x15) 싸이클만에 모든 단위 처리기들이 은퇴된 경우로, 단위 처리기 정지 로직에서 다음 주소 발생 신호(NEXT_ADDR_GEN_EN signal)를 발생하여 다음 행의 연산을 시작하기 위한 새로운 위치의 주소를 발생한다. 그림 5 (b)의 세 번째 행에서처럼, 적어도 1개 이상의 단위처리기가 조기 은퇴되지 않았다면 세 번째 행에 대한 16x16 싸이클부터 다음 16 싸이클동안 은퇴되지 않은 단위 처리기의 누적합은 현재 최소값과 비교되어 최소값을 찾고 새로운 움직임 벡터를 갱신한다. 그리고 조기 은퇴되지 않은 단위처리기가 존재하는 경우에는 처음 행의 경우와 동일하게 다음 행의 파이프라인 처리가 시작된다.

Ⅳ. 모의 실험 및 결과

단위 처리기를 조기 은퇴시키는 1 차원 시스톨릭 어레이를 이용한 구조는 Yang이 제안한 구조보다 시스템 자원의 효율성을 향상시키게 되며 추가되는 하드웨어의 부담은 상대적으로 작고, 확률적으로 높은 처리 속도를 얻을 수 있었다. 테스트 영상으로 각각 87 프레임을 갖는 CIF 포맷의 "Claire", "Foreman", "Carphone" 그리고 "Miss America"에 대하여 모의 실험하였다. 표 2에서 기존의 완전 탐색 알고리듬(FBMA)을 기준으로 한 단위 처리기를 조기 은퇴시키는 완전 탐색 알고리듬(ER_FBMA)에서 필요한 연산 싸이클 수의 백분율을, 표 3에서는 FBMA를 기준으로 한 ER_FBMA에서 소모된 전력의 백분율을 나타내어 제안한 구조의 성능을 비교하였다. 그림 6은 FBMA를 기준으로 한 ER_FBMA에서 필요한 연산 싸이클 수의 백분율에 대한 확률 밀도 함수를 보인다.

표 2. FBMA를 기준으로 한 ER_FBMA에서 필요한 연산 싸이클 수의 백분율

Table 2. Percentage of required computation cycles in ER_FBMA with respect to those in FBMA.

(a) 단위 처리기의 수에 따른 비교 (30프레임/초, 중앙 행에서 멀어지는 방향으로의 스캔 사용)

(a) Comparisons according to the number of PE's (When frame rates are 30 frames/sec with center-first scan).

단위 처리기의 수	8	16	32
Claire	53.40 %	58.05 %	61.96 %
Foreman	50.52 %	57.90 %	61.97 %
Carphone	53.00 %	58.37 %	62.27 %
Miss America	74.42 %	78.70 %	81.31 %

(b) 프레임율과 스캔 방향에 따른 비교(16개 단위 처리기 사용시)

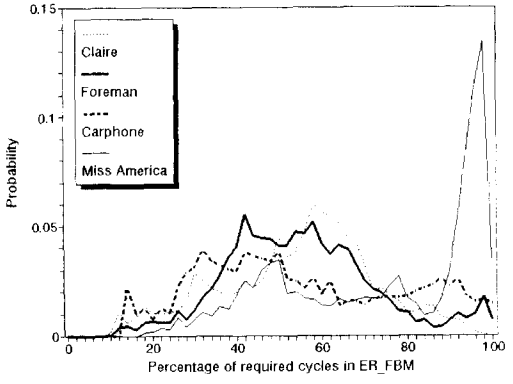
(b) Comparisons according to frame rates and scan directions (When 16 PE's are used).

프레임율	10	30	30
스캔 방향	center-first	center-first	top-down
Claire	64.68 %	58.05 %	80.29 %
Foreman	66.04 %	57.90 %	77.43 %
Carphone	62.26 %	58.37 %	78.54 %
Miss America	80.15 %	78.70 %	89.39 %

표 3. FBMA를 기준으로 한 ER_FBMA에서 소모된 전력의 백분율 (16개 단위 처리기와 중앙 행에서 멀어지는 방향으로의 스캔 사용)

Table 3. Percentage of required power consumption in ER_FBMA with respect to that in FBMA (When 16 PE's are used with center-first scan).

프레임율	10	30
Claire	49.98 %	44.99 %
Foreman	47.41 %	40.79 %
Carphone	46.95 %	43.78 %
Miss America	65.43 %	64.35 %



Percentage of required cycles in ER_FBMA with respect to those in FBMA(%)

그림 6. 확률 밀도 함수
Fig. 6. Probability Density Function.

표 2 (a)에서 보듯이, 16개의 단위 처리기를 사용하여 조기 은퇴를 시키는 완전 탐색 알고리듬을 이용했을 때, 연산 싸이클 수가 대부분의 영상에 대하여 평균적으로 대략 60%정도로 감소하였다. 특히 "Miss America"의 경우에는 다른 영상이미지보다 싸이클 수의 감소비가 약 78%정도로 다소 높은 편인데 "Miss America"의 영상 중 배경의 변화가 거의 없는 경우로 부분 누적합을 계산할 때 값의 변화가 적어 차의 절대치 합의 최소값과 비교하여 은퇴될 가능성이 줄어들었다고 판단된다. 따라서 이것은 제안한 알고리듬의 성능이 영상에 따라 달라진다는 것을 의미한다. 그리고 단위 처리기의 사용 개수가 증가함에 따라 연산 싸이클 수의 감소가 줄어듬을 알 수 있다.

표 2 (b)에서는 4개의 영상에 대하여 공통적으로 프레임율이 높을수록 제안한 구조의 성능이 향상됨을 알 수 있으며, 이것은 프레임율이 증가할수록 차의 절대치 합의 최소값이 작아지므로 단위 처리기가 은퇴될 가능성이 높아짐을 의미한다. 같은 이유로 차의 절대치 합을 처음에 구하는 곳도 전체 성능에 중요한 요인이 될 수 있다. 즉, 움직임 추정은 움직임이 매우 작을 것이라는 가정으로부터 시작한 알고리듬이므로 탐색 영역의 중심에서 차의 절대치 합이 최소일 가능성이 높다. 그러므로 1 차원 시스톨릭 어레이가 탐색 영역의 위 행에서부터 아래 행로 순차적으로 추정을 하는 방법(top-down scan)과 중앙 행에서 멀어지는 방향으로 추정을 하는 방법(center-first scan)을 비교하면 중앙 행에서 멀어지는 방향으로 추정하는 것이 나은 결과를 보였다.

표 2의 결과를 표 1에서 요약한 것과 비교하면, N^2 개의 단위 처리기를 갖는 2 차원 시스톨릭 어레이를 사용한다고 하여 처리 속도도 1 차원 시스톨릭 어레이를 사용한 경우의 N배가 되는 것은 아니며 조기 은퇴시키는 1 차원 시스톨릭 어레이를 사용한 경우 처리 속도의 차이는 더욱 줄어든다. 즉, 2N 또는 3N개의 단위 처리기를 조기 은퇴시키는 1 차원 시스톨릭 어레이를 이용하면 2 차원 시스톨릭 어레이를 사용한 움직임 추정기와 거의 비슷한 성능을 가짐을 알 수 있다.

표 3에서는 단위 처리기에서의 연산이 움직임 추정기의 전력 소모 대부분을 차지할 것이라는 가정으로부터 각 단위 처리기들이 은퇴될 때까지 연산 싸이클 수의 합으로 계산한 것이다. 이러한 가정 하에 전력 소모는 약 40~60%정도로 줄어들었다.

V. 결 론

본 논문에서는 1 차원 시스톨릭 어레이의 단위 처리기를 조기 은퇴시키는 완전 탐색 알고리듬을 이용한 움직임 추정기를 제안하였다. 제안하는 알고리듬은 1 차원 시스톨릭 어레이를 사용하여 병렬·파이프라인 처리가 용이하며, 기존의 완전 탐색 알고리듬과 동일한 움직임 벡터를 가지면서도 전체 연산 싸이클 수를 약 60%로 줄이므로 처리속도를 높일 수 있다. 또한, 전체 연산 싸이클 수가 줄어들고 단위 처리기를 조기 은

퇴시키므로 단위 처리기에서의 전력 소모는 40~60% 정도로 줄일 수 있다.

제안하는 알고리듬에서 매 블록마다 필요한 연산 싸이클 수는 가변적이며, 영상에 따라서도 가변적이라는 것을 알고 있다. 이러한 문제를 해결하기 위해 첫째로 영상 코덱에서 이러한 변화에 유연한 스케줄링을 수행하거나, 둘째로 각 응용분야에 따라서 매 블록마다 필요한 연산 싸이클 수로 제한할 수 있으며 이에 따른 화질의 저하가 존재하므로 이에 대한 연구가 지속되어야 할 것이다.

참 고 문 헌

[1] H.G. Musmann, P. Pirsch and H.J. Grallert, "Advances in Picture Coding," *Proc. IEEE*, vol. 73, pp. 523-548, Apr. 1985.

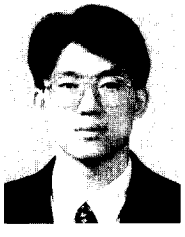
[2] T. Komarek and P. Pirsch, "Array architectures for blockmatching," *IEEE Trans. Circuits and Systems*, vol. 36, pp. 1301-1308, Oct. 1989.

[3] K.M. Yang, M.T. Sun and L. Wu, "A family of designs for motion compensation block-matching algorithm," *IEEE Trans. Circuits and Systems*, vol. 36, pp. 1317-1325, Oct. 1989.

[4] C.H. Hsieh and T.P. Lin, "VLSI Architecture for Block-Matching Motion Estimation Algorithm," *IEEE Trans. Circuits and Systems Video Technol.*, vol. 2, no. 2, pp. 169-175, Jun. 1992.

[5] Y.S. Jehng, L.G. Chen and T.D. Chiueh, "An Efficient and Simple VLSI Tree Architecture for Motion Estimation Algorithms," *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 889-900, Feb. 1993.

저 자 소 개



南 基 哲(準會員)

1969年 12月 4日生. 1993年 8月 KAIST 전기 및 전자공학과 졸업 (공학사). 1994년 3월 ~ 현재 서울대 전자공학과 석사과정 재학중. 주관심분야는 영상처리용 집적회로 설계 등임.

蔡 洙 翊(正會員) 第 31卷 B編 第 5號 參照

서울대학교 전자공학과 및 반도체 공동연구소 교수