

VLIW 기반 DSP에서의 개선된 블록매칭 알고리즘 구현

*유희재, 정선태, 정수환
승실대학교 정보통신공학부
e-mail : rovhr@erc.ssu.ac.kr

An Improved Implementation of Block Matching Algorithm on a VLIW-based DSP

*Hui-Jae You, Sun-Tae Chung, Souhwan Jung
School of Electronic Engineering, Soongsil University

Abstract

In this paper, we present our study about the optimization of the block matching algorithm on a VLIW based DSP. The block matching algorithm is well known for its computational burden in motion picture encoding. As supposed to the previous researches where the optimization is achieved by optimizing SAD, the most heavy routine of the block matching, we optimize the block matching algorithm by applying software pipelining technique to the whole routine of the algorithm. Through experiments, the efficiency of the proposed optimization is verified.

I. 서론

임베디드 프로세서의 경우, 멀티미디어 처리 응용의 증대에 따라 프로세서 처리 성능 증가가 요구되고 있다. 현재 임베디드 프로세서의 경우, 소요 전력 및 크기의 제약의 한계 속에서 고성능을 얻기 위해 병렬 처리 구조를 지원하고 있고, 병렬처리 지원 구조로는 VLIW(Very Long Instruction Word)의 채택이 대세를 이루고 있다.

본 논문에서는 VLIW 구조 프로세서에서의 최적화된 동영상 코덱 구현 연구의 일환으로[1,2], 동영상 인코더에서 가장 많은 시간이 소용되는 것으로 잘 알려진 움직임 추정의 블록 매칭 알고리즘을 VLIW 구조 프로세서에서 최적으로 구현하는 연구를 수행하였다. 기존의 연구는 블록 매칭 알고리즘의 주요 핵심 부분인 매크로블록간 SAD 계산 루틴을 최적으로 구현함으로써, 블록 매칭 알고리즘을 최적으로 구현하고자 하였다. 블록 매칭 알고리즘을 보다 최적화 구현하기 위해서는 알고리즘 전체를 살펴보고 이를 VLIW 구조에 맞게 최적화하는 것이 필요하다. 본 논문에서는 VLIW

구조를 지원하는 대표적인 DSP인 TI사의 TMS320C6713에서 기존 연구결과 보다 개선된 성능을 나타내는 블록 매칭 알고리즘 구현에 대해 연구한 결과를 간략히 기술한다.

II. 관련 연구

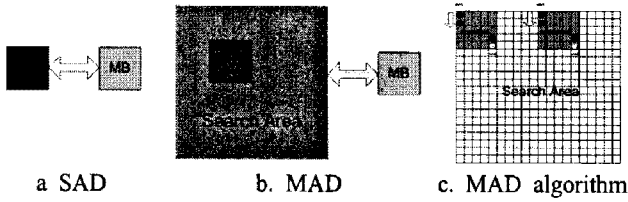
VLIW 프로세서에서 루프의 최적화 구현 기법으로 가장 많이 사용되는 것은 소프트웨어 파이프 라이닝이며, 이에 대해서는 많은 연구가 되어 있다. 소프트웨어 파이프라이닝 스케줄링에는 모듈로 스케줄링 기법[3,4]이 많이 활용되며, 본 논문에서도 이를 사용하였다. 블록 매칭 알고리즘의 구현에 대한 연구는 많으나, 대부분이 알고리즘 그 자체의 효율성에 관한 연구이다[5]. VLIW 프로세서에서의 블록 매칭 알고리즘 최적화 구현 연구로는 [1,2,6] 등이 있으나, 주로 블록 매칭 알고리즘의 주요 루틴인 매크로블록간 SAD 루틴 최적화에 집중되어 있다. TI의 DSP Image Library는 매크로블록간 SAD 계산 알고리즘 뿐만 아니라 블록 매칭 알고리즘의 어셈블리 코드 구현을 제공하고 있다[7]. 본 논문에서 개선한 블록 매칭 알고리즘 구현은 TI의 DSP Image Library에서 제공하는 블록매칭 알고리즘 구현보다 성능이 낫다.

III. 개선된 블록 매칭 알고리즘 구현

3.1. MAD(Minimum Absolute Difference)

기존의 블록 매칭 알고리즘으로서 화두인 SAD는 현재 매크로블록과 참조 매크로블록 하나와만 비교하여 가장 작은 값을 찾아내는 개념이다. 이에 반해 탐색영역 전체에서 비교하여 현재 매크로블록과의 차이가 최소값이 되는 매크로블록을 찾아내는 알고리즘이 MAD(Minimum Absolute Different) 알고리즘이다

<그림 1> 블록 매칭 알고리즘



3.1. MAD 알고리즘 개선

기존 TI사의 MAD 알고리즘은 내부 루프 안의 누적치 값의 활용이 이루어 지지 않고 있다 이를 활용하기 위해 MAD 알고리즘 안의 내부 중첩루프를 단일 루프로 변환하고 8개의 실행 유닛을 가진 TMS320C6713의 VLIW 구조를 최대한 사용하기 위하여 탐색영역을 둘로 나누고 (<그림 1> c. MAD algorithm 참조) 명령어들의 병렬 수행 가능성을 고려하여 스케줄링 한다.

탐색 시 두 개의 시작점에서부터 동시에 매크로블록을 계산해 나가며 각각의 영역에서 이미 조사된 매크로블록의 가중치 보다 더 큰 계산 값이 나오면 조건 분기를 통해 보다 적은 가중치 쪽을 가진 영역을 가려내어 계산 량을 줄일 수 있다.

이러한 알고리즘은 최소값을 가진 매크로 블록이 마지막에 위치한 것처럼 최악의 경우라 하더라도 한 번의 루프에 하나씩 찾는 방식에 비해 루프의 수를 절반으로 줄일 수 있다

IV. 실험 및 결과 검토

4.1 실험 환경

본 논문에서는 Texas Instruments의 통합 개발환경인 Code Composer Studio의 시뮬레이션 툴을 사용하였다. 내부 메모리는 32KB 2way L2 캐쉬로 구성하였으며, heap 사이즈는 7MB, 스택 사이즈는 64KB로 설정하였다.

사용한 H.263 Encoder는 TMN 8.0 에 기반을 둔 UBC(University of British Columbia) H.263 version 2 (H.263+) 구현으로 C 소스 코드의 크기는 17,000 라인 (530KBytes)이다. 실험에서는 기본 모드 (baseline) 만을 테스트하였고 선택한 영상 포맷은 QCIF이며, 테스트 영상은 테스트 영상으로 많이 쓰이는 foreman (foreman.qcif)이다. 또한, 움직임 벡터 추정에는 전탐색 (fast full search) 알고리즘을 선택하였으며, 역 DCT 는 빠른 IDCT (FastIDCT) 알고리즘을 사용하였다. 또한, 컴파일 시에 -o2 옵션(소프트웨어 파이프라이닝 지원)으로 컴파일 하였다.

4.2 실험 결과 및 검토

실험에서는 TMS320C6713 DSK에서 블록 매칭 알고리즘 C 버전을 이용한 H.263 인코더와 블록 매칭 알고리즘을 논문[1]의 SAD 어셈블리 코드를 사용한 블록 매칭, 논문[2]의 SAD 어셈블리 코드를 사용한 블록 매칭, TI Image Library의 블록 매칭 어셈블리 코

드, 본 논문에서 개선한 블록 매칭 어셈블리 코드를 각각 이용한 H.263 인코더 각각에 대해 QCIF 사이즈 영상 프레임 2장을 인트라모드 1장, 인터모드 4장으로 인코딩을 수행하였다.

<표 1> 실험결과

	SNR			소요 사이클수	비율(%)
	Y	Cb	Cr		
C 버전 (-o2 옵션 사용)	31.15	38.46	38.59	223213581	100.00
논문[1]의 SAD 어셈블리 코드	31.15	38.46	38.59	187185533	83.86
논문[2]의 SAD 어셈블리 코드	30.72	37.99	38.17	139495788	62.49
TI 블록매칭 MAD 어셈블리 코드	32.01	38.81	39.21	121844516	54.59
본 논문의 블록매칭 MAD 어셈블리 코드	30.82	38.06	38.24	119888075	53.71

표 1에서 보면, 본 논문에서 제안한 블록 매칭 알고리즘이 성능을 개선하였음을 알 수 있다.

IV. 결론 및 향후 연구 방향

본 논문에서는 VLIW 구조 프로세서에서의 최적화된 동영상 코덱 구현 연구의 일환으로, 동영상 인코더에서 가장 많은 시간이 소용되는 것으로 잘 알려진 움직임 추정의 블록 매칭 알고리즘을 VLIW 구조 프로세서에서 최적으로 구현하는 연구의 결과를 기술하였다. 본 논문의 연구 결과는 VLIW 기반 프로세서에 이식되거나 설계되는 S/W 알고리즘의 최적화 방안에 도움을 줄 것으로 기대된다.

참고문헌

- [1] 최수철, 정선대, 임철수, "VLIW 기반 고성능 DSP에서의 블록 매칭 알고리즘 최적화," 차세대 PC 학회, pp.1 ~ 12, 2005년 12월호.
- [2] 한동훈, 정선대, 정수환 "VLIW 기반 프로세서에서의 블록 매칭 알고리즘 최적화 구현," 2006년도 대한전자공학회 추계종합학술대회 제29권 제2호 pp.703-707, 11.25, 2006.
- [3] S. Lam, "Software pipelining: An effective scheduling technique for VLIW machines," in Proc the ACM SIGPLAN 1988 Conference on Programming Language Design andpp. 318-328, Jun. 1988.
- [4] B. R. Rau, "Iterative Modulo Scheduling," International Journal of Parallel Processing, Vol. 24, No. 1, Feb. 1996
- [5] W. Hwang, Y. Lu, Y. Zeng, "Fast block-matching algorithm for video coding," Letters Volume 33, Issue 10, pp. 833 - 835, May 1997.
- [6] H. R. Sheikh, et al., "Optimization of a Baseline H.263 Video Encoder on the TMS320C6000," Proc. Texas Instruments DSP Educator's Conference, Aug. 2000.
- [7] TMS320C62x Image Library, <http://focus.ti.com/docs/toolsw/folders/print/sprc093.html>