MPEG-4 의 형상 정보 부호화에 사용되는 움직임 추정부의 고속 알고리즘

유 동 훈, 장 성 규, 나 종 범
한국과학기술원 전자전산학과 전기 및 전자공학 전공

A Fast Motion Estimation Algorithm for MPEG-4 Shape Coding

Donghoon Yu, Sung Kyu Jang, and Jong Beom Ra
Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
E-mail: dhyu@issserver.kaist.ac.kr and jbra@ee.kaist.ac.kr

요 약

본 논문에서는 MPEG-4 의 형상 정보 부호화에 사용되는 움직임 추정부의 고속 알고리즘을 제안한다. 형상 정보 부호기에서 사용되는 움직임 추정부는 기존의 텍스처 기반의 움직임 추정부와는 다른 특성을 가지는데 형상 정보 추정기에서 사용되는 움직임 추정부는 CAE (Context-based Arithmetic Encoding)에서 사용될 컨텍스트를 만들기 위해 수행된다는 점과 움직임 벡터의 공간적 상관성, 그리고 형상정보가 이진성을 가진다는 점이 그것이다. 이러한 세가지 특성을 사용한 제안된 알고리즘은 움직임 추정부의 수행 속도를 비약적으로 향상시킨다. 실험 결과에 의하면 계산량은 최악의 경우에도 10% 이하로 떨어지는 것을 볼 수 있다. 따라서 본 논문에서 제안한 알고리즘은 실시간 소프트웨어의 구현에 적합한 알고리즘이라고 할 수 있다.

I. 서론

MPEG-4 는 MPEG (Moving Pictures Expert Group)에 의해 개발되고 있는 ISO/IEC 표준이다. 버전 1 은 표준 제정이 완료되었고 버전 2 는 계속 작업 중에 있다. MPEG 에 의해 개발된 동영상 압축에 관한 표준은 지금까지 MPEG-1 과 MPEG-2 가 있는데 이는 모두 동영상 CD 나 고화질 텔레비전(HDTV)과 같이 사각형의 화면을 효과적으로 압축하기 위한 표준이었다. 그러나 인터넷의 발달과 함께 멀티미디어의 중요성이 부각되면서 객체중심의 동영상 압축이 중요하게 되었고 이를 표준화한 것이 바로 MPEG-4 이다. MPEG-4 는 그래픽, 정지영상, 동영상, 합성 영상 등을 다루기 위한 여러 가지 도구들을 제공하고 전송률도 64kbps 에서 38.4Mbps 까지 프로파일 (profile)에 따라 제공하고 있다. 따라서 그 응용분야도 인터넷 통신, 화상 회의, 이동 통신, 무선 멀티미디어 서비스까지 다양하다. 또한 객체 중심의 부호화를 기본 개념으로 하기 때문에 각 동영상 객체 평면 (VOP : Video object plane) 은 임의 형상 정보를 저장하는 알파평면 (Alpha plane)을 가지고 있다. 이에 따라 임의의 형상 정보를 부호화하는 것이 필요하다.

형상 정보의 부호화를 수행할 때 움직임 추정을 사용한다.. 움직임 추정은 형상정보의 부호화를 하지 않는 이전의 압축 표준에서도 텍스처 정보를 부호화하기 위해 널리 이용되는 방법 중 하나이다. 동영상에 존재하는 시간적 상관성을 제거하기 위한 좋은 방법으로 적용되고 있는 것이다. 그러나 그 효과에도 불구하고 복잡성이 너무 커서 실시간 부호기의 설계에 많은 방해가 되어 왔다. 많은 연구들이 움직임 추정부분을 빠르게 하는 데 초점이 맞추어져 행해져 왔고 지금도 많은 연구들이 행해지고 있다. 그러나 많은 연구에도 불구하고 여전히 실시간 부호기의 설계에 많은 장애로 작용하고 있고 결국 움직임 추정부는 하드웨어로 구현하는 것이 좋은 방법으로 사용되고 있다. 그런데 MPEG-4 에서 형상정보를 부호화하는데 한번 더 움직임 추정을 사용하고 이는 실시간 부호기의 설계에 다시 한번 어려움을 안겨주게 되었다 [4].

이에 본 논문에서는 형상 정보 부호기의 특징을 이용하여 움직임 추정부의 계산량을 소프트웨어적인 방법만으로 실시간 부호기에 적합하도록 하는 방법을 제안하고자 한다. 형상 정보의 부호화는 크게 세 가지의 특징을 가지고 있는데 첫 번째로 형상 정보는 컨텍스트를 기반으로 하는 CAE (Context-based Arithmetic Encoding: 컨텍스트 기반의 산술 부호기)를 이용하여 부호화 하는데 이 컨텍스트를 얻기 위하여 움직임 추정을 한다는 점이다. 두 번째는 각 이진 알파 블록 (BAB : Binary Alpha Block)이 가지는 움직임 벡터 (MVs : Motion vector for shape)는 주변 BAB 의 움직임 벡터와 공간적인 상관성을 가진다는 점이다. 마지막 특성은 알파 평면에 표현된 형상 정보는 이진성을 가진다는 것이다. 이 세가지 특징을 이용한 제안한 알고리즘은 움직임 추정의 계산량에서 아주 많은 속도 향상을 보여준다. 실험 결과는 가장 나쁜 경우에도 VM (Verification Model) [1]에 소개된 방법에 비하여 0.6%의 복잡도만 가지는 것을 보여준다.

논문의 구성은 다음과 같다. II 장에서 VM 에서 권장하는 움직임 추정부의 방법을 소개하고, III 장에서 제안된 알고리즘에 대하여 설명한다. IV 장에서 실험 결과를 보이고, V 에서 결론을 맺는다.

II. VM 에서의 움직임 추정부

지금까지의 표준안은 복호기 부분만을 정의했지만 MPEG-4 에서는 부호기 부분도 설명하고 이에 대한 VM (Verification Model) [1]을 프로그램으로 만들고 있다. 이 VM 에서 소개하고 있는 움직임 추정부의 방법에 대해 알아본다.

형상 정보를 부호화할 때 사용하는 변수로 AlphaTH 가 있다. 이는 부호화된 형상정보의 질을 결정하는 변수로써 원래의 BAB 과 부호화된 BAB 사이에 달라도 되는 화소의 개수라고도 표현할 수 있다. 조금 더 자세히 표현하면 16x16 의 BAB 을 16 개의 4x4 화소블럭으로 나누어 각 화소블럭 단위로 SAD (Sum of Absolute Difference)와 16×AlphaTH 를 비교한다. 따라서 AlphaTH 를 0 으로 놓는 것은 형상 정보 부호화에서 전혀 손실이 없도록 하겠다는 것을 의미한다. 움직임 추정을 하고 난 후에는 형상정보에서의 움직임 벡터 (MVs; Motion Vector for shape)을 얻을 수 있는데 이는 MVPs (Motion Vector Predictor for shape)과 MVDs (Motion Vector Difference for shape)의 합으로 구성된다.

II.1. MVPs 의 선정

MVPs 는 현재 부호화하고자 하는 BAB 의 주위 BAB 의 움직임 벡터들과, 현재 BAB 에 해당되는 텍스처 매크로블럭 (MB: Macroblock)과 그 주위의 MB 에서의 움직임 벡터들이 후보가 된다. 이 후보 움직임 벡터들을 일정한 순서에 따라 나열한 후 이 중 유효한 첫번째 후보를 MVPs 로 선정한다. 움직임 벡터의 유효 여부는 VM [1]에서 정의하고 있다. 만일 유효한 움직임 벡터가 하나도 없으면 MVPs 는 (0,0)으로 놓는다.

II.2. MVs 의 결정

MVPs 가 결정되면 이를 바탕으로 MVs 의 결정을 하게 된다. 순서는 다음과 같다.

먼저 MVPs 를 이용하여 움직임 보상 (MC; Motion Compensation)을 수행하여 이 때의 SAD 를 구한다. 각 4x4 화소블럭에서의 SAD 를 16×AlphaTH 와 비교해 이보다 작으면 움직임 벡터를 MVPs 로 결정하고 과정을 마친다. 그러나 만약 위의 조건을 만족하지 않으면 MVPs 를 중심으로 SAD 를 가장 작게 하는 움직임 벡터의 탐색을 하고 된다. 탐색 영역은 MVPs 를 중심으로 수직 수평 방향으로 ±16 화소 크기이다. SAD 를 가장 작게 하는 움직임 벡터는 MVs 로 선정되고 MVPs 와의 차이, 즉 MVDs (=MVs - MVPs)를 부호화하게 된다.

만일 두개 이상의 움직임 벡터가 같은 SAD 값으로 최소의 SAD 를 가진다면 MVDs 를 부호화했을 때 발생하는 비트량을 최소로 하는 움직임 벡터를 선정한다. 만일 MVDs 를 부호화했을 때 발생되는 비트량도 같다면 수평방향의 움직임 벡터 값이 작은 것을, 만일 이것도 같다면 수직방향의 움직임 벡터 값이 작은 것을 선정한다.

이와 같은 방법으로 SAD 를 최소로 하는 움직임 벡터가 결정되고 이 움직임 벡터에 의해 움직임 보상이 된 BAB 은 CAE 에서 필요한 컨텍스트의 결정에 사용된다. 이 점은 텍스처에서 사용하는 움직임 추정기와의 가장 큰 차이가 되는데 바로 이 점 때문에 SAD 를 최소로 하는 움직임 벡터가 최고의 압축 효율을 보장하지는 않게 된다. MVPs 를 중심으로 움직임 벡터를 탐색할 때 기본적으로 전역탐색을 수행한다. 그러나 이러한 탐색방법은 여러 탐색 방법 중 하나의 예일 뿐이라고 표준안에서 말하고 있다.

III. 제안된 알고리즘

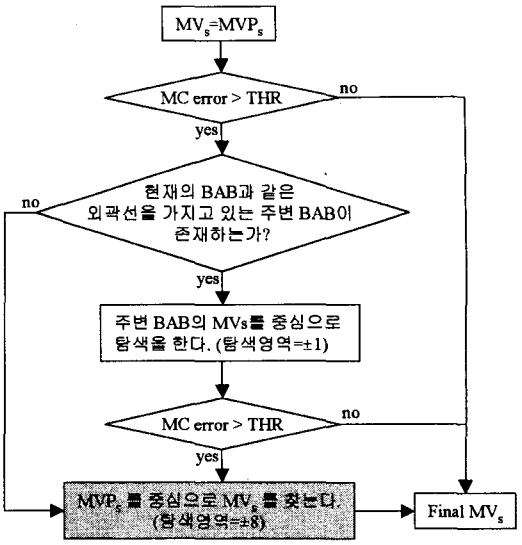

그림 1. 제안된 알고리즘의 흐름도

그림 1 에 제안한 알고리즘의 전체적인 흐름도를 보이고 있다. 움직임 추정에서 많은 계산량으로 문제가 되는 MVPs 를 중심으로 실제 탐색을 하는 BAB 의 개수를 줄이기 위한 두 번의 기회를 가지는데, 이때 형상정보 부호화의 두 특성을 사용한다. 첫째는 형상정보에서 사용하는 움직임 추정은 CAE 를 수행할 때의 컨텍스트를 결정하기 위해서 한다는 것이다. 둘째는 주변 BAB 과의 움직임 벡터의 상관성이다. 그리고 탐색을 하게 될 때는 SAD 를 계산하여 기준으로 삼는데 이때 마지막으로 형상정보의 이진성을 사용한다.

III.1. 컨텍스트 기반의 움직임 추정기

텍스처에서 사용되는 움직임 추정기는 SAD 를 기준으로 가장 적은 SAD 를 갖는 움직임 벡터를 찾는다. 이때 부호화는 움직임 보상된 MB 과 현재의 MB 사이의 에러 이미지를 부호화하기 때문에 최소의 SAD 는 최대의 압축률을 보장하게 된다. 그러나 형상 정보에서 사용하는 움직임 추정기는 다른 특성을 가진다. 형상 정보에서의 움직임 추정기는 형상 정보 부호화중 CAE 에서 사용할 컨텍스트를 결정하기 위한 부분이다. 이 컨텍스트는 CAE 의 압축률을 결정하는데 SAD 와 압축률은 대략적으로는 비례 관계를 가지지만 최소의 SAD 가 최대의 압축률을 보장하지는 않는다. 따라서 이러한 특성을 이용하여 새로운 값 THR 을 사용하여 MVPs 를 움직임 벡터로 생각하고 구한 SAD 값을 THR×255 와 비교하여 SAD 가 THR×255 보다 작으면 최종 MVs 를 MVPs 로 결정하고 탐색은 수행하지 않아도 된다. 이렇게 탐색을 수행하지 않아도 되는 BAB 의 개수를 늘리면 움직임 추정부의 속도는 빨라지게 된다. 이러한 방법으로 탐색을 수행하지 않게 되는 BAB 은 최악의 경우에도 30%가량이 된다.

III.2. 주변 BAB 과의 공간적 상관성

각 BAB 은 주변 BAB 과의 공간적 상관성을 가지고 있고 이를 이용하면 계산량을 더욱 줄일 수 있다. 형상정보는 객체와 배경으로 나뉘어질 수 있는데, 객체의

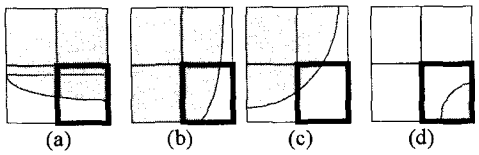

그림 2. 객체의 외곽선 연결에 따른 네 가지 분류

외곽선을 따라 서로 비슷한 움직임 벡터를 가지고 있을 것이라는 것은 충분히 예상할 수 있다. 먼저 형상 정보의 연결에 따라 네 가지로 분류할 수 있는데 그림 2.에서 그 분류를 보이고 있다. 그림 2.(a)는 현재 BAB의 외곽선이 왼쪽 BAB의 외곽선과 연결되어 있다. 따라서 현재 BAB의 움직임 벡터는 왼쪽 BAB의 움직임 벡터와 상관성을 가질 것이므로 왼쪽 BAB의 움직임 벡터를 이용한다. 그림 2.(b)는 위쪽 BAB과 움직임 벡터가 비슷할 것이고 그림 2.(c)는 위쪽과 왼쪽 BAB이 모두 현재 BAB과 외곽선을 공유하고 있으므로 두 BAB의 움직임 벡터를 모두 사용할 수 있다. 그림 2.(d)는 현재 BAB과 외곽선이 연결된 BAB이 존재하지 않으므로 이용할 주변 BAB이 존재하지 않고 이 경우에는 움직임 벡터의 공간적 상관성을 이용하지 못하고 그냥 탐색을 수행하게 된다. 주변 BAB의 움직임 벡터를 사용하여 약간의 미세조정 (탐색 영역=±1) 을 하고 난 후 SAD를 역시 THR과 비교하여 탐색을 해야 하는 BAB의 개수를 줄여 계산량을 줄인다. 이 방법을 사용하여 대략 10%정도의 BAB이 추가로 탐색을 수행하지 않게 된다.

III.3. 형상 정보의 이진성

형상 정보는 알파 평면을 통해서 표현되는데 이때 알파 평면에 하나의 화소는 하나의 변수로 표현된다. 객체는 255, 배경은 0으로 표현되는데 이러한 알파 평면을 이진알파평면이라고 한다. 이에 비하여 객체가 0에서부터 255까지의 값을 가지는 알파 평면도 있는데 본 논문에서 관심을 가지는 움직임 추정을 수행할 때는 이 역시 0과 255로만 구성된 이진 알파 평면으로 바꾸어 수행하게 된다. 즉 움직임 추정을 할 때는 언제나 알파 평면은 이진알파평면이 된다. 그런데 이렇게 이진의 값을 가지는 화소는 하나의 비트만으로 표현해도 충분하므로 이를 이용하면 SAD를 계산하는 시간을 줄일 수 있다. 하나의 화소를 하나의 비트로 표현하면 하나의 바이트 변수에 8개의 화소를 동시에 표현할 수 있다. 이렇게 8개의 화소를 하나의 변수에 묶어 표현된 두 BAB에 대하여 각 변수를 XOR한 후 미리 만들어 놓은 표를 이용하면 가장 많은 계산량이 드는 SAD 계산하는 부분의 시간을 많이 줄일 수 있다. 실험적으로 대략 계산량이 1/6로 줄어드는 것을 확인할 수 있었다. 반면 이런 과정을 수행하기 위해 필요한 메모리의 양의 증가는 무시할 만큼 충분히 적다.

III.4. 최종적인 탐색 알고리즘

위에서 제안한 방법들을 사용하여 탐색을 수행하는 BAB의 개수를 줄이기는 했지만 결국 탐색을 수행해야 하는 BAB은 존재하게 된다. 그러므로 이 탐색 과정에서 사용할 알고리즘 또한 수행 속도에 영향을 미치게 된다. 이 탐색 과정에는 지금까지 텍스처 움직임 추정에 사용된 어떤 알고리즘이라도 사용될 수 있다. 기존의 텍스처에서 사용된 움직임 추정 방법은 TSS (three step search), NTSS (new three step search), OTS (one-at-a-time search), 4SS (four step search), SES (simple and efficient search), UCBDS (unrestricted center-biased diamond search), BBGDS (block-based gradient descent search) 등의 방법이 있다. 제안 된 알고리즘을 통하여 MVDs가 (0,0)을 가지는 많은 수의 BAB이 탐색을 수행하지 않아도 되게 되었지만 여전히 MVDs는 (0,0)을 중심으로 분포하는 특성을 가지고 있다. 또한 SAD의 분포가 하나의 극소값 (국부 최소값)을 가지는 경우가 대부분이므로 BBGDS의 알고리즘이 가장 좋은 결과를 보일 것이다. 따라서 BBGDS 알고리즘을 사용하여 실험을 하였다.

IV. 실험결과

실험은 Children, Bream, Hall monitor, Akiyo, Container, News 의 총 6개의 동영상을 사용하였다. 크기는 QCIF (176x144)이고 총 300개의 VOP들을 10Hz와 30Hz로 부호화 하였다. 형상 정보는 무손실 부호화를 수행하였다.

표에서 #BAB은 탐색을 수행하는 BAB의 개수를 의미한다. 제안된 알고리즘에서 사용한 세가지 특성 중 두가지 특성은 탐색을 수행하는 BAB의 개수를 줄이는데 사용된다. #S_bits/VOP는 하나의 VOP에 해당되는 형상을 부호화 한 후 생성된 비트량의 평균값이다. 이를 통하여 제안된 알고리즘을 적용함으로써 변화하는 비트량을 알 수 있다. #SP는 탐색점의 개수를 나타내는데 이는 탐색을 수행하는 BAB의 개수와 탐색에서 사용하는 알고리즘의 종류에 따라 달라지게 된다. 즉 이 #SP가 알고리즘의 속도와 비례관계를 가진다. 단 여기에는 형상정보의 이진성을 이용한 속도 향상은 포함되어 있지 않다. 그리고 표에서 사용된 퍼센트는 기존의, VM에서 사용된 방법에 대한 비율을 표시한 것이다.

결과를 보면 Container, Akiyo, Weather와 같이 움직임이 적은 비디오에 대해서는 탐색을 수행하는 BAB의 개수가 매우 적다는 것을 알 수 있다. 반면 Hall monitor의 경우 가장 결과가 나쁘게 나왔는데, 탐색 방법에 관계없이 탐색을 수행하는 BAB의 개수는 전체의 60%정도가 되고 여기에 형상정보의 이진성까지 사용하면 계산량은 10% 이하로 떨어지게 된다. 만일 탐색에 BBGDS 알고리즘을 적용하면 #SP는 3.6%까지 줄어들고 이진성을 고려하면 계산량은 0.6%까지 떨어지게 된다.

V. 결론

본 논문에서는 형상 부호화 기법의 특성과 형상정보의 특징을 이용하여 MPEG-4 형상정보 부호화기에 사용되는 움직임 추정부의 고속 알고리즘을 제안하였다 제안된 알고리즘은 하드웨어를 쓰지 않고도 소프트웨어적으로 실시간 부호기를 가능하게 한다. 실험 결과에 따르면 제안된 알고리즘은 최악의 경우에서도 형상정보 부호화에 사용되는 움직임 추정부의 계산량을 VM에서 사용한 알고리즘에 비하여 10% 이하로 줄인다. 여기에 기존의 고속 탐색 알고리즘, BBGDS를 사용하면 계산량은 0.6%까지 떨어지게 된다.

본 연구는 산자부 지원에 의해 수행되었음.

참고 문헌

[1] ISO/IEC JTC1/SC29/WG11, MPEG99/5477, "MPEG-4 Video Verification Model version 14.2," Dec. 1999.

[2] L.-K. Liu, and E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 419-422, Aug. 1996.

[3] T. Koga, K. Iinuma, A. Hirano, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Nat. Telecommun. Conf.*, pp. G5.3.1-G.5.3.5, 1981.

[4] D.Gong, and Y. He, "Computation complexity analysis and VLSI architectures of shape coding for MPEG-4," in Proc. SPIE Visual Communication and Image Processing 2000, vol. 4067, 2000, pp. 1459-1470.

Table 1. Simulation results (30Hz, about 64kbps)

Sequence		Full search						BBGDS					
		#MB	(%)	#S_Bits /VOP	(%)	#SP	(%)	#MB	(%)	#S_Bits /VOP	(%)	#SP	(%)
Children	VM	10237		977.5		2958493		10237	100.0	980.2	100.3	121520	4.1
	Proposed	2883	28.2	980.9	100.3	833187	28.2	2897	**28.3**	985.7	100.8	44040	**1.5**
Bream	VM	8736		814.9		2524704		8722	99.8	810.4	99.4	109913	4.4
	Proposed	2541	29.1	813.0	99.8	734349	29.1	2556	**29.3**	816.7	100.2	38754	**1.5**
Hall monitor	VM	2852		276.3		824228		2850	99.9	274.9	99.5	35354	4.3
	Proposed	1216	42.6	275.8	99.8	351424	42.6	1213	**42.5**	275.3	99.6	15661	**1.9**
Akiyo	VM	1759		152.4		508351		1738	98.8	151.9	99.7	16759	3.3
	Proposed	1	0.1	152.1	99.8	289	0.1	2	**0.1**	152.2	99.9	65	**0.0**
Containe r	VM	1943		196.2		561527		1922	98.9	195.3	99.5	18802	3.3
	Proposed	10	0.5	192.5	98.1	2890	0.5	10	**0.5**	192.5	98.1	90	**0.0**
News	VM	1759		152.4		508351		1738	98.8	151.9	99.7	16759	3.3
	Proposed	1	0.1	152.1	99.8	289	0.1	2	**0.1**	152.2	99.9	65	**0.0**

Table 2. Simulation results (10Hz, about 64kbs)

Sequence		Full search						BBGDS					
		#MB	(%)	#S_Bits /VOP	(%)	#SP	(%)	#MB	(%)	#S_Bits /VOP	(%)	#SP	(%)
Children	VM	3794		1257.1		1096466		3794	100.0	1260.9	100.3	55320	5.0
	Proposed	1763	46.5	1259.3	100.2	509507	46.5	1765	**46.5**	1264.8	100.6	32125	**2.9**
Bream	VM	3138		1010.5		9068882		3141	100.1	1012.0	100.1	47951	5.3
	Proposed	1562	49.8	1016.3	100.6	451418	49.8	1566	**49.9**	1020.9	101.0	29121	**3.2**
Hall monitor	VM	971		305.1		280619		972	100.1	304.6	99.8	14653	5.2
	Proposed	604	62.2	307.2	100.7	174556	62.2	602	**62.0**	307.0	100.6	10013	**3.6**
Akiyo	VM	981		204.3		283509		967	98.6	202.8	99.3	9763	3.4
	Proposed	2	0.2	200.9	98.3	578	0.2	3	**0.3**	201.4	98.6	99	**0.0**
Container	VM	1278		292.4		369342		1272	99.5	291.1	99.6	13230	3.6
	Proposed	18	1.4	283.5	97.0	5202	1.4	18	**1.4**	283.5	97.0	174	**0.0**
News	VM	981		204.3		283509		967	98.6	202.8	99.3	9763	3.4
	Proposed	2	0.2	200.9	98.3	578	0.2	3	**0.3**	201.4	98.6	99	**0.0**