

다중 참조 영상을 위한 연속 제거 알고리즘 기반 움직임 추정 기법

김영문, 이재은, 강현수
 중앙대학교 첨단영상대학원 영상공학과
 kymoon@wm.cau.ac.kr, jlee@wm.cau.ac.kr, hskang@cau.ac.kr

Motion Estimation Scheme Based on Successive Elimination Algorithm for Multi-reference

Young-Moon Kim, Jae-Eun Lee, Hyun-Soo Kang
 Graduate School of AIM, CHUNG-ANG University

요 약

본 논문에서는 전역움직임추정방법의 고속 알고리즘인 연속 제거 알고리즘(successive elimination algorithm)을 다중 참조영상(multi-reference frames)에 적용하였을 때의 결과를 분석하고, 이를 바탕으로 계산량 감축 방안을 제안한다. 제안된 방법은 바로 이전 영상에 대한 움직임 벡터는 기존의 연속 제거 알고리즘을 그대로 적용하는 대신, 그 외의 참조영상에 대해서는 이전 영상에서 얻어진 움직임 벡터를 사용하여 외삽(extrapolation)을 수행함으로써 추정하고, 추정된 벡터를 중심으로 탐색범위를 제한함으로써 계산량을 크게 감축하는 방법이다. 실험을 통해 제안된 방법의 성능을 분석하고 검증할 것이다.

I. 서론

움직임 추정 (motion estimation: ME)은 비디오 신호의 시간 방향 잔여성분 (redundancy)를 제거하는데 효과적이므로 H.261, H.263, H.264, MPEG-1, MPEG-2, MPEG-4 등의 비디오 압축 표준에서 뿐만 아니라 움직임 보상 부호화 기법이 적용된 비디오 부호화에 널리 채용되고 있다. 그래서 좋은 복원 화질을 유지하면서, 좀 더 빠르고 정확한 움직임 벡터를 찾을 수 있는 방법이 요구된다. 전역탐색법 (full search algorithm)은 최적의 움직임 벡터를 찾는 반면 많은 계산량이 요구된다. 그래서 이러한 단점을 해결하기 위해, 많은 고속 알고리즘들이 제안되어 왔다. 예를 들어, 2-D logarithmic 탐색법, 3 단계 탐색법, conjugate direct 탐색법, cross 탐색법, 4 단계 탐색법, diamond 탐색법 등이 있다 [1][2][3][4]. 또한, 전역탐색법 자체에 대한 고속화 알고리즘 또한 활발히 연구되었는데, PDE(partial difference elimination algorithm), SEA (successive elimination algorithm), MSEA (multi-level SEA) 등이 대표적이다. PDE는 MPEG의 reference s/w에서 구현되어 있는 방식으로서, SAD (sum of absolute difference)의 계산 과정 중 이전의 최소 SAD를 초과하는 경우 SAD의 계산을 더 이상 수행하지 않는 방식이다[5][6][7]. SEA는 블록의 평균값으로부터 최적 벡터가 될 수 있는지의 여부를 판단함으로써 전체 블록에 대한 SAD의 계산이 불필요한 블록에 대한 계산을 수행하지

않음으로써 계산량을 감축하는 방법이다[8]. MSEA는 SEA의 다계층 접근방식으로 기본적인 아이디어는 동일하나 계산량을 크게 줄이는 방법이다[9][10]. 여기서 언급된 PDE, SEA, MSEA는 모두 전역탐색법과 동일한 성능을 보이면서 계산량을 줄이는 고속 전역탐색법이라고 할 수 있다.

본 논문에서는 SEA를 이용하여 다중 참조영상에 적용하고, 또한 탐색범위별로 그 성능을 분석한다. 구성은 다음과 같다. 2장에서는 전역 탐색과 연속 제거 알고리즘의 다중 참조를 간략하게 기술하였고, 3장에서는 계산량을 줄인 연속 제거 알고리즘의 다중 참조를 제안하고 실험을 통해 제안된 방법의 성능을 평가하였다. 마지막으로, 4장에서는 제안된 방법을 요약 기술하고 결론을 맺는다.

II. 기존 알고리즘

2.1 전역 탐색 알고리즘(full search algorithm)

움직임 벡터를 찾는 가장 간단한 방법은 탐색 범위 내의 모든 위치에 대하여 SAD를 구하고, 그 SAD값이 최소가 되는 위치를 찾는 것이다. 이와 같은 방법은 전역탐색법이라고 한다. 전역탐색방식은 최적의 움직임 벡터를 찾을 수 있지만, 많은 계산량이 요구된다는 단점을 가지고 있다. 탐색 영역 내에 모든 탐색점에 대해 다음 식을 적용함으로써 움직임 벡터를 찾게 된다.