

Discontinuity 특성을 줄이기 위한 블럭 기반 움직임 추정 알고리즘

A Simple Block-based Motion Estimation Algorithm for Discontinuity Blocks

裴 皇 植*, 鄭 正 和*

Hwang-Sik Bae*, Jong-Wha Chong*

요 약

본 논문은 블럭 기반 움직임 추정 알고리즘들에서 나타나기 쉬운 불연속성(discontinuity)을 줄이기 위한 알고리즘을 제안한다. 제안된 알고리즘은 정합 과정 중, 한 블럭에 대한 부분 SAD 값을 이용하여 쉽고 간단하게 불연속성 특성을 갖는 블럭을 찾아낼 수 있으며, 이러한 블럭들에 대해서만 부분 블럭에 대한 움직임 벡터를 추가 생성해낸다. 제안된 알고리즘은 기존의 여러 대표적인 움직임 추정 알고리즘들과의 성능 비교를 통하여, 일반 블럭에 대해서는 다른 알고리즘과 대등한 연산량을 나타내며, 불연속성을 갖는 블럭에 대해서는 화질의 개선을 보여주었다.

Abstract

In this paper, we propose a motion estimation algorithm for the discontinuity blocks. The algorithm uses sub-SAD value (i.e. the sum of absolute difference for a quarter of a block) to identify the discontinuity region, and produces additional motion vectors for these sub-blocks if necessary. We show with experimental results that, in comparison with some conventional motion estimation algorithms, the proposed algorithm achieves quality enhancement for the sequences with discontinuity blocks, and also shows the same computational quantity as to normal algorithms for sequences with less discontinuity.

Key words : block based motion estimation, discontinuity, multi-resolution motion estimation

1. 서 론

블럭 기반 움직임 추정 기법은 데이터의 흐름이 규

칙적이고, 간단하기 때문에 알고리즘 설계와 하드웨어 구현까지 널리 사용되고 있다.[1] 이러한 블럭 기반 움직임 추정 기법은 한 블럭 내에 있는 모든 영역이 동일한 움직임을 갖는다는 전제하에 이루어진다.[1][2][4][5][7] 그러므로, 한 블럭 내에 여러 물체가 존재한다 하더라도, 이러한 알고리즘은 수식적인 계산에 의해서 오차가 가장 적은 곳을 한 블럭에 대한 움

* 漢陽大學校 電子電氣컴퓨터工學部
(Department of Electronic & Electrical and Computer
Engineering, Hanyang University)
接受日:2002年 5月1日, 修正完了日:2002年7月28日

직업 벡터로 결정하게 되며, 이는 서로 다른 움직임을 갖는 여러 물체가 같은 움직임을 갖는 벡터로 결정됨으로써 화질의 저하를 초래할 수 있다. 이러한 움직임 벡터에 의해서 재생된 블록(reconstructed block)은 인접한 재생 블록들과 서로 다른 움직임을 나타낼 수 있으며, 이러한 블록들은 불연속(discontinuity)되어 보일 수 있다.

이러한 불연속성으로 인해 발생하는 문제를 최소화하기 위해서는 모든 물체에 대해서, 각각의 움직임 벡터를 구해야 한다. 그러나 이러한 방법은 블록 기반 움직임 추정에서 블록 단위로 나누어 연산을 수행하는 것에 알맞지 않을 뿐 아니라, 이로 발생하는 연산량의 증가도 무시할 수 없다.

본 논문에서는 이러한 불연속성을 띄는 블록들에 대해서, 한 블록을 여러 부분으로 나누어 각 부분에 대해서 발생하는 SAD(sum of absolute difference)값을 이용하여, 간단하게 불연속성의 여부를 판단하여 필요한 경우 일부분에 대한 움직임 벡터를 추가로 발생시켜주는 알고리즘을 제안한다. 제안되는 알고리즘은 SAD나 유사한 오차 기준식을 이용하여 정합을 수행하는 대부분의 움직임 추정 알고리즘들에게 적용될 수 있다.[1][2][3][4][5] 본 논문에서는 기존의 대표적인 알고리즘들에 제안되는 알고리즘을 적용하여, 성능 평가를 수행한 결과 기존의 알고리즘보다 우수한 화질의 향상을 얻을 수 있었다.

II. 불연속성을 위한 움직임 추정

이 절에서는 블록 기반 움직임 추정 과정에서, 각 블록의 불연속성 여부를 검사하는 방법과, 이를 이용해 추가의 움직임 벡터를 구하는 방법에 대하여 기술된다.

2.1 불연속성 블록(discontinuity blocks)

일반적인 블록 기반 움직임 추정 기법들은 정합이 수행되는 하나의 블록 내부에 존재하는 모든 물체들은 동일한 움직임을 갖는다는 가정을 전제로 한

다.[1][2][4][5][7] 그러나 이러한 사실은 실제에서는 어긋나는 경우가 많으며, 특히 움직임의 변화가 비교적 큰 동영상의 경우 많이 발생한다. 다수의 물체가 서로 다른 방향으로 움직이거나, 배경과 달리 움직이는 물체의 경우 등은 서로 다른 움직임을 갖는 물체들이 한 블록 내에 존재하게 된다.

실제로 블록 기반 알고리즘들이 블록을 단위로 나눈 영역은 움직이는 물체를 중심으로 구분한 것이 아니고, 일정한 영역의 크기대로 규칙적으로 나눈 것이므로, 한 블록에는 여러 개의 물체가 존재할 수 있으며, 각각의 물체는 각각의 움직임을 나타낼 수 있다. 대부분의 블록 기반 움직임 추정 알고리즘들은 각 블록 단위로 현재 프레임과 이전 프레임에서의 해당되는 하나의 블록 단위의 정합을 수행하여 정합 오차가 가장 적게 발생하는 곳을 움직임 벡터로 결정한다. 그러나 서로 다른 움직임을 갖는 여러 물체가 있는 블록의 경우, 이렇게 계산되어진 움직임 벡터는 정확한 움직임 추정을 하지 못하게 되고, 이는 화질 저하의 결과를 초래한다.

이처럼, 하나의 블록 내부에 서로 다른 움직임을 갖는 다수의 물체를 갖는 블록을 “불연속성 블록(discontinuity block)”이라 정의한다. 이와 같은 불연속성 블록에 대한 예는 그림1.에서 확인할 수 있다.

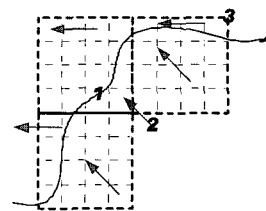


그림 1. 불연속성 블록(discontinuity block)의 예
Fig. 1 an example of discontinuity blocks

그림1.에서 예로 보이는 3개의 블록에서 '1'로 표기되어있는 경계를 중심으로 '2'로 표기된 움직임과, '3'으로 표기된 움직임을 볼 수 있다. 이와 같은 경우, '1'의 표시가 되어 있는 곳의 블록은 2개의 움직임이 있는 것을 알 수 있다. 그러나 일반적인 블록 기반 움

직임 추정 알고리즘을 통해 정해지는 움직임 벡터는 이 2가지 움직임과는 무관한, 정합 오차가 수식적으로 최소가 되는 곳으로 결정되게 된다.

2.2 블럭의 불연속성 판정

한 블럭에 존재하는 다수의 움직임은 블럭 내에 존재하는 일부분의 각각의 움직임을 나타내게 된다. 이러한 일부분에 대한 모든 움직임을 찾는 것은 계산량의 증가와, 블럭 기반 움직임 추정 방식의 규칙적이고, 간단한 구현 방식을 해치는 것이므로, 모든 움직임을 찾기보다는, 구현에 알맞도록 일부분에 대한 움직임을 찾아야 한다. 본 논문에서 제안된 알고리즘은 이러한 일부분에 대한 움직임을 찾는 것을 그림 2. 에 서와 같이 하나의 블럭을 작은 4개의 부분블럭(sub-block)으로 나누어 처리했다.

일반적인 블럭 기반 알고리즘은 정합되었음을 판단하는 기준으로 식1.과 같은 두 화소 x 와 y 의 차를 누산하는 SAD(sum of absolute difference)식을 많이 사용한다.

$$SAD(p, q) = \sum_{r,c=0}^{16} |x(r, c) - y(r + p, c + q)| \quad (1)$$

$$|p, q| \leq search_range$$

이는 정합을 위해 사용되는 두 블럭에 대해서, 두 블럭간의 화소들의 차이를 구할 목적으로, MAD(mean absolute difference)를 구하는 것이 원칙이나, 이 값보다는 하드웨어로 구현하기에 보다 쉽고 간단한 SAD를 사용하는 것이다. 이러한 SAD를 이용해서 "정합되었다(matched)"라고 판단을 하는 것은 이 SAD값이 최소가 될 때를 의미한다. 그런데, 그림 2.와 같이 불연속성 블럭의 경우 SAD를 구성하는 값이 부분에 따라서 다르게 분포될 수 있으며, 이를 이용하여, 불연속성의 여부를 판단할 수 있다. 이러한 방법은 SAD를 사용하는 일반적인 블럭 기반 방식의 알고리즘이라면, SAD를 구하는 중간 과정의 하나로써, 블럭의 불연속성 여부를 판단할 수 있게 되므로, 추가의 과정 없이 구현될 수 있다.

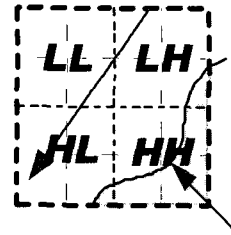


그림 2. 불연속성 블럭 내부의 부분블럭
Fig. 2 sub-blocks in a discontinuity block

그림 2. 는 하나의 블럭을 4개의 sub-block 으로 나누어 표시한 것이며, 각 sub-block들을 위치에 따라서, 각각 LL, LH, HL, HH 라고 명명하였다. 그림 2. 와 같은 예는 2개의 움직임이 1개의 블럭 내에 존재하는 경우로, 일반적인 블럭 기반 알고리즘에 의하여 "정합(matched)"된 위치에서의 SAD를 살펴 보면, sub-block HH가 다른 sub-block 보다 큰 오차 값을 갖는 것을 알 수 있다. 이는 현재의 위치가 정합된 곳이므로 가장 적은 SAD값을 갖는 곳인데, 전체 블럭에 대해서 일부분의 다른 움직임을 갖는 HH가 더 큰 오차를 갖게 되기 때문이다. 이러한 sub-block HH는 정합된 현재 블럭에 대해서, 서로 다른 움직임을 갖게 되므로, 이 부분만을 따로 정합 과정을 수행하여, 현재의 정합된 곳보다 적은 정합 오차를 갖는 곳을 탐색한다면, 보다 적은 정합 오차를 갖는 블럭을 재생(reconstruction)할 수 있다.

본 알고리즘은 4개로 나누어진 sub-block 들의 평균 오차를 불연속성 여부를 판단하기 위한 기준으로 삼았으며, 이 값을 다양하게 변화시켜 가며 실험을 통해서 성능 평가를 수행하였다. 이 실험 결과는 III절에서 제시되었다. 이렇게 설정된 기준값보다 더 큰 오차를 갖는 sub-block HH의 경우, 이 블럭을 "불연속성 블럭"이라 판정한다.

2.3 불연속성을 위한 움직임 추정

불연속성을 판정하고, 이를 움직임 추정에 적용하여 sub-block에 대한 움직임 벡터를 산출하기 위한 과정이 본 절에서 기술될 것이다.

2.3.1 불연속성을 고려한 후단계 처리(post processing)

로서의 움직임 추정

불연속성을 고려한 움직임 추정을 위한 방법은 2.2 절에서 기술된 바에 의한 불연속성의 판정에 필요한 절차를 거치게 된다. 일반적인 블록 기반 움직임 추정 알고리즘이라면, 정합을 위한 기준을 대부분 SAD, 혹은 이와 유사한 기준식을 사용하므로, 2.2절의 판정 절차를 쉽게 적용할 수 있다. 이때, 이러한 판정을 위해 추가되는 연산은 sub-block에 의한 기준식과의 비교만으로 끝나게 되므로 아주 적다고 할 수 있다. 그러므로 본 불연속 판정 알고리즘은, 블록 단위로 이루어지는 대부분의 움직임 추정 알고리즘에 쉽게 추가 적용 될 수 있다.

본 알고리즘을 적용하려면 대부분의 블록 정합에 사용되는 SAD와 같은 정합 기준식을 적용할 때, 하나의 블록 전체에 대해서 값을 저장하는 것 대신에, sub-block 단위로 구분하여 저장하고, 블록 전체에 대해 값의 저장이 끝났을 경우, 하나의 블록 당 2.2절에서 제시한 여러 판정 방법에 의해서 비교 연산을 한번 수행하면 된다. 이와 같은 연산 구조는 하드웨어로 설계가 될 경우에도 쉽게 구성이 될 수 있다. 움직임 추정기의 하드웨어 구현은 대부분, 블록 단위로 처리가 이루어지는 처리 소자(processing element)로 구현되는데, 본 논문의 알고리즘은 이러한 처리 소자의 최종 출력에 대한 비교 연산부를 추가하여 쉽게 구성될 수 있다.

비교 연산에 의한 불연속성 판정 결과, 현재 정합 중인 블록이 불연속성을 띄지 않는다고 판정될 경우, 일반적인 블록 기반 알고리즘과 다를 것 없이 동일한 추정 과정이 진행되며, 만약, 불연속성 판정이 될 경우는 불연속 판정을 받은 sub-block 에 대한 추가 정합을 수행한다. 이러한 과정은 일반적인 움직임 추정 이후의 판정 결과에 의해서, 추가적인 연산이 필요할 경우에만, 움직임 추정 이후 후단계(post processing)의 형태로 구현될 수 있다.

2.3.2 Multi-Resolution 움직임 추정에서의 적용

본 논문에 제안된 알고리즘은 대부분의 블록 기반 움직임 추정 알고리즘에 후단계(post processing)의

형태로 적용되는 것이 가능하다. 본 절에서는 이러한 예를 multi-resolution 움직임 추정에 적용시켜 설명을 한다.

하나의 블록을 sub-block으로 나누어 SAD를 계산하는 본 알고리즘은 블록 기반의 대부분의 움직임 추정 알고리즘에 쉽게 적용될 수 있지만, 특성상 multi-resolution 움직임 추정에 적용될 경우, 특히 sub-block 별로 나누어 처리하기도 용이하고, 구현도 또한 간단하다.

MRME(multi-resolution motion estimation)은 각 level별로 서로 다른 해상도의 이미지를 구축하여 저해상도의 이미지로부터 추정을 시작하여, 고해상도의 이미지로 추정을 옮겨가면서, 점차 정확한 추정을 해나가는 방법으로, level이 높아지면서 보통 2:1 로 sub sampling이 이루어져 저해상도의 이미지가 만들어진다. 정합의 기준으로 사용되는 SAD는 MRME에서 각 level별로 다음과 같이 식.2 로 기술된다.

$$SAD_{(p,q)} = \sum_{r=0}^{16/2^l-1} \sum_{c=0}^{16/2^l-1} |X'(r,c) - Y'(p+r,q+c)| \quad (2)$$

where

- X : luminance at current frame
- Y : luminance at previous frame
- l : level which is from 0 to 2
- (r,c) : location at previous frame
- (p,q) : location at current frame

MRME는 상위 level에서 추정된 후보 움직임 벡터(candidate motion vector)는 하위 level에서 조정된 탐색 범위 내에서 좀 더 정확한 값을 갖도록 보정되면서 가장 낮은 level에서의 움직임 벡터를 현재 정합 중인 블록의 최종 움직임 벡터로 결정한다.

MRME의 경우에도 그림 2.에서와 같이 불연속성 블록(discontinuity block)이라고 판정을 받은 블록은 sub-block에 대한 추가의 움직임 추정 과정을 거치게 되는데, 불연속성이라고 판정 받은 sub-block들에 대한 움직임 추정을 위해, 일반 블록과 같은 많은 level의 과정을 거친다면, 너무나 많은 연산량이 발생하고, 또한, sub-block 자체가 이미 2:1로 sub sampling되어

있는 상태이므로, 일반적인 MRME의 최상위 level 보다 한 단계 낮은 level로 부터 추정을 시작한다. 만약 이러한 sub-block들도 최상위 level부터 추정을 시작한다면, 이미 sub sampling 되어 있는 sub-block들이 더 많은 sub sampling이 되어 추정을 행하게 되므로, 추정을 올바로 행하기에 부족한 화소의 개수로 인해 성능 저하를 나타낼 수 있다.

정합 과정 중 발생하는 연산량을 줄이기 위해, 움직임 벡터들 사이의 위치적 연관성과, 정합을 도중에 중단(half-stop strategy)하는 방식을 적용한다. 인접한 블록들과의 움직임 벡터의 연관성[1][2][4]은 그림 3.에서 보는 바와 같이 탐색을 시작하기 위한 위치를 결정짓기 위해서 주변의 인접 블록들에서 발생한 움직임 벡터를 참조로 한다. 이는 하나의 움직이는 물체는 여러 블록에 걸쳐서 위치하여도 동일하거나, 유사한 방향과 크기의 움직임 벡터를 갖는다는 특성을 이용한 것이다.

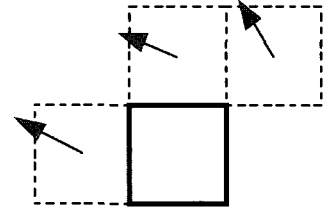


그림 3. 움직임 벡터의 공간적 연관성
Fig. 3 spatial MV correlation

대부분의 경우 이러한 연관성을 이용한 처음 탐색 위치에서, 주변의 탐색 위치에서보다 적은 정합 오차를 갖게 되며, 이는 도중 중단(half-stop strategy)과 같이 적용되었을 때, 연산량 감소의 효과를 크게 볼 수 있다. 도중 중단 방식은 현재 정합중인 블록에서, 각 화소의 정합 오차를 누산(accumulation)하는 동안에, 현재까지의 누산 오차 값이, 현재 블록에 대한 탐

표 1. 블록 기반 알고리즘과의 화질 성능 비교
Table 1. COMPARISONS OF PERFORMANCE

	FS	TSS	MR	MRD	MRD0_5	MRD0_7	MRD1_5
Flower	24.026	21.810	19.397	20.657	20.600	20.939	19.262
Football	22.936	22.009	22.092	22.422	22.731	22.754	22.129
TableTennis	29.329	27.913	28.224	28.127	28.592	28.680	28.029
average	25.430	23.911	23.238	23.735	23.974	24.125	23.140
Sales	35.567	35.295	35.469	35.469			
MissAmerica	37.712	37.130	36.982	36.982			

표 2. 블록 기반 알고리즘과의 정합 연산량 비교
Table 2. COMPARISONS OF COMPUTATIONAL COMPLEXITY

	FS	TSS	MR	MRD	MRD0_5	MRD0_7	MRD1_5
Flower	76,395,520	2,408,928	4,791,857	14,586,545	20,467,564	19,518,030	3,201,641
Football	82,258,432	2,574,610	5,209,799	16,134,061	23,423,785	22,112,759	4,200,912
TableTennis	76,395,520	2,395,479	4,475,012	13,945,008	20,337,498	19,844,422	1,840,644
average	78,349,824	2,459,672	4,825,556	14,888,538	21,409,616	20,491,737	3,081,066
Sales	99,847,168	3,103,744	5,113,211	5,113,211			
MissAmerica	99,847,168	3,136,936	6,710,227	6,710,227			

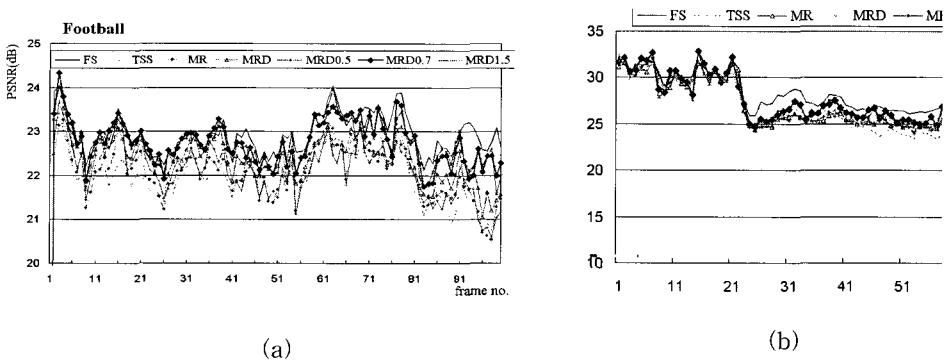


그림 4. 불연속성이 있는 영상들에 대한 화질 성능 비교: (a) Football (b) Tabletennis
 Fig. 4 Comparison of performance for the sequences with discontinuity: (a) Football (b) Tabletennis

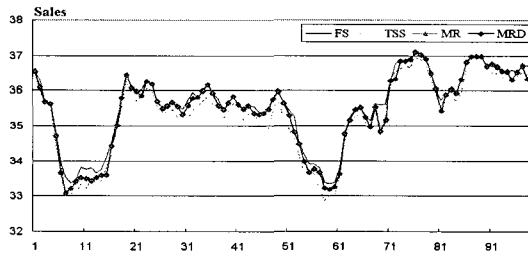


그림 5. 불연속성이 적은 영상에 대한 화질 성능 비교
 Fig. 5 Comparison of performance for the sequence with less discontinuity (less discontinuity makes MRD operate in normal mode without additional discontinuity matching.)

색 범위에 있는 다른 블록들에 대한 정합 오차들의 최소 값보다 커지게 되는 순간, 현재 블록에 대한 정합을 포기하고 다음 블록에 대한 정합 과정을 진행시킨다. 이는 움직임 벡터는 가장 적은 정합 오차를 갖는 위치에서 결정되기 때문에, 가장 적은 정합 오차를 갖는 곳보다 큰 오차를 갖는 곳은 움직임 벡터로 결정될 확률이 없으므로, 더 이상의 연산이 필요하지 않기 때문이다.

III. 실험 결과

본 절은 제안된 알고리즘을 기존의 여러 블록 기반

움직임 추정 알고리즘에 적용하여 성능 평가를 수행한 실험 결과에 대해 기술한다.

본 실험에 사용된 테스트 동영상들은 "Football" (352x240, 100 frames), "Flower Garden" (352x224, 100 frames), "Tabletennis" (352x224, 100 frames), "MissAmerica" (352x288, 100 frames), "Sales" (352x288, 100 frames) 등과 같이 다양한 종류의 움직임을 갖고 있는 영상들이다. 처음 3개의 이미지 시퀀스들은 비교적 다양한 움직임을 갖고 있는 영상들이고, "Football"의 경우 가장 복잡하고, 많은 움직임을 갖고 있는 영상이며, "Flower garden"의 경우 카메라 이동(panning)의 기술이 적용된 영상으로, 이 시퀀스는 풀이나 잔디의 움직임으로 인한 복잡하고 임의적인 움직임을 갖고 있는 영상이다. "Table tennis"는 다양

한 움직임 이외에도, 축소 확대(zooming) 등의 기술이 적용되어 있는 영상이다. 기술된 바와 같이 다양하고, 임의적인 움직임을 갖는 이상의 영상들은 불연속성 블럭(discontinuity)을 포함하고 있을 가능성이 큰 영상들로; 이러한 특성으로 인해 실험 결과 화질의 개선을 얻을 수 있었다.

나머지 뒤의 "MissAmerica", "Sales"와 같은 이미지 시퀀스들은 뉴스 캐스터의 장면과 같은 비교적 적은 움직임을 갖는 영상들로, 이러한 영상들은 불연속성 블럭을 포함할 확률이 매우 적다.

본 실험에서 사용된 MRME는 2.2절에서 기술된 불연속성 판단 기준을 여러 가지로 설정하여 실험 되었는데, 이 기준식에 따라, 본래의 기준식을 적용한 것을 MRD(multi-resolution for discontinuity)라 명명하였으며, 기준식 값에 따라 MRDxxx와 같은 명명을 하였다. 이는 기존의 대표적인 추정 알고리즘인 FS(full search), TSS(three step search), MR(multi-resolution search)등과 성능 비교를 하여 실험 결과를 Table I, II에 나타내었다.

Table I은 제안된 알고리즘을 통한 테스트 영상들에 대한 화질 개선에 대한 실험 결과를 화질 평가의 기준으로 널리 쓰이고 있는 평균 PSNR 값을 통해 보여주고 있다. 본 실험에서는 제안된 알고리즘을 MRME에 적용하여 결과를 도출하였는데, 이 중 MRD0_7에 대한 실험 결과는, MR은 물론이고 블럭 기반 움직임 추정 중에 가장 정확도가 높은 것으로 평가되고 있는 Full-search 블럭 정합 알고리즘에 거의 유사한 수준의 화질 개선을 이루었으면서도 연산량은 보다 적음을 확인 할 수 있다. 이는 Table II에서 다시 확인할 수 있는데, 이러한 실험 결과는, 물론 일반적인 MRME보다도 화질이 개선되었음을 나타낸다. 주의할 것은 Table I,과 Table II에서 구분되어 있는 선을 중심으로 불연속성이 있을 확률이 큰 영상과, 적은 영상이 나뉘어 있는데, 각각의 실험 결과를 통해, 본 알고리즘은 일반 블럭 기반 알고리즘들처럼 똑같이 동작을 할 수도 있고, 불연속성이라고 판정을 받은 sub-block들에 대해서만 필요할 경우 추가적인 정합을 실행하는 것을 알 수 있다.

각 실험 결과 table에서, 불연속성이 거의 없는

Sales나, MissAmerica등의 영상들은 일반적인 블럭 기반 알고리즘과, 불연속성을 위한 알고리즘 사이에 거의 차이가 없는 것을 알 수 있으며, 이는 특히 MRME의 경우 Table II에서의 수치와 같이 MRME와 MRDxxx들의 수치가 같은 것을 통해서 확인할 수 있다. MRDxxx에 해당하는 나머지 항목도 모두 이와 같은 수치를 나타낼 것이므로 항이 비어 있음에 유의한다. Table II에 나타난 MRD0_5, MRD0_7, MRD1_5의 결과를 살펴보면, 기준식에 의해 불연속성이 있는 것으로 판단된 영상들에 대해 기준값을 조정하면서 제안된 알고리즘을 적용한 결과 기준식에 따라 연산량을 충분히 조정할 수 있음을 알 수 있으며, 이는 성능과 연산량간에 서로 조정할 수 있음을 나타낸다. 만일, 2.2 절에서 기술된 불연속성을 판단하는 기준을 시간적인 연관성을 고려하여, 프레임 간에 서로 다른 기준식을 적용하면, 불연속성을 위한 추가적인 연산량을 더욱 줄일 수 있을 것이다.

그림 4와 그림 5는 각 알고리즘에 따른 영상의 frame별 PSNR의 변화를 나타낸 것으로, 그림 4에서는 전술된 table에서 MRD라 명명된 알고리즘이 기존의 다른 블럭 기반 알고리즘들 보다 우수한 성능을 나타내어, FS 정합 알고리즘에 가까운 화질 성능을 보이는 것을 알 수 있다. 이는 특히 불연속성이 많이 포함된 영상에서 뚜렷하게 보임을 알 수 있다. 그림 5에서는 Sales와 같이 불연속성이 거의 없는 영상들은 일반 블럭 기반 알고리즘들과 별 차이 없이 동작함을 알 수 있다.

IV. 결론과 고찰

본 논문에서는 일반적인 블럭 기반 움직임 추정 알고리즘들에 적용시킬 수 있는 불연속성 블럭을 위한 움직임 정합 알고리즘을 제안하였다. 이는 불연속성이라고 판정받은 블럭에 대해서만, 추가적인 정합을 시행하여 보다 정확한 움직임 추정을 하게 한다. 본 알고리즘은 불연속성이 없는 일반 블럭들에 대해서는, 일반적인 블럭 기반 알고리즘들과 동일하게 동작한다.

불연속성을 판정하기 위한 추가적인 연산은 SAD를 이용하여 간단하게 수행하며, 이 SAD는 이미 일반적인 블럭 기반 알고리즘들에서 정합시 구해지는 값이다.

제안된 알고리즘은 대부분의 블럭 정합 움직임 추정 알고리즘들에 간단히 추가되어 합당한 추가적인 움직임 추정 과정을 통해 화질 향상을 하고자 하는 곳에 쓰일 수 있다.

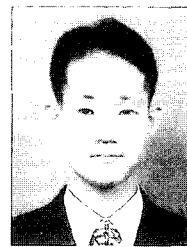
불연속성을 판정하는 기준에 대한 조정을 하면, 더욱더 화질 면과 연산량 면에서의 성능 향상을 할 수 있을 것이며, sub-block을 나누는 방법과 크기, 또 각 level에서의 탐색 범위 등을 조정해도 보다 나은 성능 향상을 할 수 있을 것이다. 여기에, 시간적 연관성을 고려한 움직임 추정을 수행한다면, 더욱더 불연속성 블럭을 판정하고, 정합하는데 성능 향상을 할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] J.H.Lee, et al. , "A Fast Multi-Resolution Block Matching Algorithm and its LSI Architecture for Low Bit-Rate Video Coding," IEEE trans.Circuits Systems.Video Technol. ,vol.11, pp.1289-1301,Dec. 2001.
- [2] J.Chalidabhongse and C.C.J.Kuo,"Fast motion vector estimation using multiresolution-spatio-temporal correlations," IEEE trans.Circuits Systems.Video Technol. ,vol.7, pp.477-478, June 1997.
- [3] X.Song,T.Chiang,X.Lee, and Y.-Q.Zhang,"New fast binary pyramid motion estimation for MPEG2 and HDTV encoding," IEEE trans.Circuits Systems.Video Technol. ,vol.10, pp.1015-1028,Oct. 2000.
- [4] S.Zafar,Y.-Q.Zhang, and J.S.Baras, "Predictive block-matching motion estimation for TV coding-Part I:Inter-block prediction," IEEE trans.Broadcast.. ,vol.37, pp.97-101,Sept. 1991.
- [5] Y.-Q.Zhang, and S.Zafar, "Predictive block-matching motion estimation for TV coding-Part II : Inter-frame prediction," IEEE trans.Broadcast.. ,vol.37, pp.102-105,Sept. 1991.
- [6] W.Li and E.Salari, "Successive elimination algorithm for motion estimation," IEEE trans.Image Processing.. ,vol.4, pp.105-107,Jan. 1995.
- [7] L.W.Lee, J.F.Wang, J.Y.Lee, and J.D.Shie, , "Dynamic search window adjustment and interlaced search for block-matching algorithm," IEEE trans. Circuits Systems.Video Technol. ,vol.3, pp.85-87, Feb. 1993.

저 자 소 개

裴 皇 植 (正會員)



1972년 7월 15일생
1995년 2월 한양대학교 전자공학과 공학사
1997년 2월 한양대학교 대학원 전자공학과 공학석사
1997년 ~ 현재 한양대학교 대학원 전자공학과 박사 과정
관심분야 : 동영상 압축, 전송 및 하드웨어 설계 기법

鄭 正 和 (正會員)

제6권 제1호 논문 02-01-10참조
현재 한양대학교 정보통신대학 교수