

구속조건을 적용한 다이아몬드 탐색 알고리즘에 의한 고속블록정합움직임추정

홍성용*

Fast Block-Matching Motion Estimation Using Constrained Diamond Search Algorithm

Sung-Yong, Hong *

요 약

영상 시퀀스로부터 추정된 움직임벡터분산에 관한 연구에 기초하여 본 논문에서는 고속블록정합움직임 추정을 위한 구속조건을 적용한 다이아몬드탐색(DS) 알고리즘을 제안한다. 영상 시퀀스에서의 움직임벡터가 수직 또는 수평방향으로 2개화소이내의 거리에서 탐색되어지는 점을 고려하여, DS 알고리즘은 수직 또는 수평방향의 움직임벡터의 추정을 강조함으로써 새로운 3단계 탐색알고리즘에 비하여 오차율 면에서 유사한 결과를 나타내지만 계산량을 줄일 수 있었음을 확인하였다. 또한, DFD를 구속조건으로 DS 알고리즘에 적용함으로써 움직임이 없는 안정된 블록에서의 움직임벡터의 계산에 소요되는 계산량을 줄일 수 있을 뿐 아니라 국부 해를 예측할 가능성을 감소시킬 수 있었다. 따라서 제안된 구속조건을 적용한 DS 알고리즘은 기존의 DS 알고리즘뿐 만 아니라 4단계 탐색, 블록기반 기울기-감소 탐색에 비해 평균자승오차 또는 필요한 탐색 점의 수에서 우수한 결과를 나타냄을 알 수 있었다.

Abstract

Based on the studies on the motion vector distributions estimated on the image sequences, we proposed constrained diamond search (DS) algorithm for fast block-matching motion estimation. By considering the fact that motion vectors are searched within the 2 pixels distance in vertically and horizontally on average, we confirmed that DS algorithm achieves close performance on error ratio and requires less computation compared with new three-step search (NTSS) algorithm. Also, by applying displaced frame difference (DFD) to DS algorithm, we reduced the computational loads needed to estimate the motion vectors within the stable block that do not have motions. And we reduced the possibilities falling into the local minima in the course of estimation of motion vectors by applying DFD to DS algorithm. So, we knew that proposed constrained DS algorithm achieved enhanced results as aspects of error ratio and the number of search points to be necessary compared with conventional DS algorithm, four step search (FSS) algorithm, and block-based gradient-descent search algorithm

▶ Keyword : Fast Block Matching Algorithm, Diamond Search Algorithm

움직임추정, 다이아몬드 탐색 알고리즘, 세부사항, 제안한 방법, 및 실험결과 언급하였다.

I. 서론

비디오압축의 측면에서 고려해 볼 때, 비디오시퀀스에서 이웃프레임간의 시간적인 측면에서의 군더더기(temporal redundancy)라고 불리는 많은 양의 시간적인 상관관계를 갖는 요소들을 식별하여 이를 제거함으로써 적절하게 압축된 결과를 얻을 수 있다.

블록정합움직임추정(block-matching motion estimation, BMME)이라고 불리는 효과적이고 많이 사용되어지는 방법들이 CCITT, H.261, H.263, MPEG-1, MPEG-2, 그리고 MPEG-4등과 같은 다양한 비디오압축표준에서 널리 채택되어졌다. 그러므로 빠르고 정확한 블록기반탐색기술은 처리지연을 줄일 수 있을 뿐 아니라 향상된 재구성된 영상의 화질을 유지하기 위하여 그 필요성이 요구되어지고 있다.

완전탐색(full search) 알고리즘과 같이, 탐색 창(search window)내의 모든 후보블록들의 움직임을 추정하는 소모적인 방법은 탐색 창 내 최소정합오차를 갖는 점의 추정과 같은 포괄적인 최적 해를 구하는 것이 가능하지만 많은 양의 계산을 필요로 하게 된다. 이와 같은 단점을 극복하기 위하여 2차원 로그탐색(2-D logarithmic search, LOGS)[2], 3단계 탐색(Three-Step Search, TSS)[3], 공액방향탐색(Conjugate Direction Search, CDS)[4], 교차탐색(Cross Search, CS)[5], 새로운 3단계탐색(New Three-Step Search, NTSS)[6], 4단계 탐색(Four-Step Search, FSS)[7], 블록기반기울기감소탐색(Block-Based Gradient-Descent Search, BBGDS)[8] 등과 같은 많은 고속 블록정합알고리즘(Block-Matching Algorithm)들이 연구되어졌다. 이 고속 블록정합알고리즘들은 완전탐색과 비교할 때 탐색 점의 수를 급격히 감소시킬 수 있는 최적 움직임 벡터를 추정하기 위하여 각각 서로 다른 탐색패턴과 탐색방법을 연구하여 적용하였다.

본 논문에서는 효율적이고 간단한 고속블록정합알고리즘인 다이아몬드 탐색(Diamond Search) 알고리즘에 구속조건을 적용함으로써 오차 면에서 DS알고리즘이나 NTSS, FSS알고리즘과 유사한 결과를 나타내면서도 계산량을 감소시킬 수 있는 방법을 제안하였다.

본 논문은 II, III, IV, V, 그리고 VI절에 각각 블록정합

II. 블록정합움직임벡터추정

기본적으로, 고속 움직임 추정에 있어서 향상된 탐색속도와 그 처리오차를 구하기 위하여 탐색패턴의 형태와 크기에 관하여 연구되었다. 탐색 창 내에서 블록의 왜곡 또는 블록정합오차를 구함으로써 최소오차를 갖는 최적의 정합을 나타내는 움직임벡터를 추정할 수 있는데, 이와 같은 정합오차는 많은 움직임요소들을 포함하는 영상 시퀀스에 대하여 다중국부최소점(multiple local minimum point)을 갖게 된다. 그러므로 BBGDS[8]와 같이 3×3과 같은 크기의 작은 탐색패턴을 이용한 움직임 추정의 경우 많은 움직임요소들을 포함하는 비디오 시퀀스에 대하여 국부최소점을 추정함으로써 재구성된 영상의 질을 저하시킬 수 있다. 한편 TSS와 같이 9×9의 큰 탐색패턴을 이용한 움직임 추정의 경우 산재된 탐색패턴으로 인하여 잘못된 방향으로 탐색을 함으로써 최적 해를 추정하지 못할 수 있다.

실제 비디오 시퀀스의 경우 최적 해는 탐색창의 중심점에 해당하는 0의 움직임을 갖는 분포를 나타내기 때문에 중심으로 편향된 탐색패턴을 적용하는 NTSS는 기존의 블록정합방법에 비하여 적은 수의 탐색 점을 적용하면서도 향상된 결과를 나타낼 수 있다. 그러나 NTSS는 TSS에 비하여 복잡한 탐색패턴을 갖는다. NTSS와 FSS는 검색과정에서 검색 점을 중복하여 적용함으로써 계산량을 감소시킨다. 그러나 9개의 검색 점만을 고려하는 BBGDS에 비하여 FSS는 여전히 안정된 움직임을 갖는 영역의 경우 17개의 검색 점에서의 움직임을 확인하여야만 한다. 또한 BBGDS를 제외한 기타 블록정합움직임추정알고리즘 즉, TSS, NTSS, 그리고 FSS의 경우, 검색 창 크기를 15×15로 국한시키고 있다.

Chun-Ho Cheung 등은 평균자승오차를 정합기준으로 하였을 때 완전탐색알고리즘(FSA)을 적용하여 검색창의 중심으로부터 특정거리만큼 떨어진 곳의 움직임벡터의 분산확산확률을 연구하였다[6]. 예측프레임(predicted frame)을 원래의 이전프레임으로부터 추정했다고 할 때, 움직임벡터

의 52.76%에서 98.70%가 검색 창 중심으로부터 2개 화소이내에 위치하고 있다는 것을 확인하였다. 또한 실제 비디오시퀀스에서의 블록의 움직임은 다양한 방향으로 발생되어질 수 있으나, 주로 수직 또는 수평방향으로 움직임이 발생되어진다는 것을 알 수 있었다.

이와 같은 근거를 바탕으로 2개 화소를 반지름으로 하는 원내에 위치하는 검색 점들로 구성되어진 움직임패턴을 적용하는 다이아몬드검색알고리즘을 사용함으로써 기존의 고속블록정합알고리즘에 비해 적은 탐색 점만을 사용함으로써 최적의 움직임벡터를 추정하는 것이 가능하다.

III. 다이아몬드검색알고리즘

다이아몬드검색(Diamond Search, DS)알고리즘은 그림 1과 같이 두 가지의 검색패턴을 적용한다. 첫 번째는 큰 다이아몬드검색패턴(Large Diamond Search Pattern, LDSP)으로서, 가운데 점을 중심으로 9개의 검색 점으로 구성되어져 있으며, 두 번째 패턴은 작은 다이아몬드검색패턴(Small Diamond Search Pattern, SDSP)으로서 5개의 검색 점으로 구성되어져 있다.

DS 알고리즘의 검색과정은 중심에 위치한 검색 점에서 최소블록왜곡(Minimum Block Distortion, MBD)값이 구해질 때까지 LDSP를 반복적으로 적용하고, 만약 중심에 위치한 검색 점에서 최소블록왜곡 값이 구해질 경우 마지막 단계로 SDSP를 적용한다. SDSP의 5개의 검색 점들 중에서 최소블록왜곡 값을 갖는 검색 점을 그 블록의 움직임벡터로 고려한다.

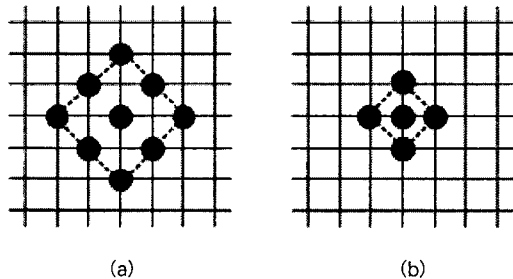
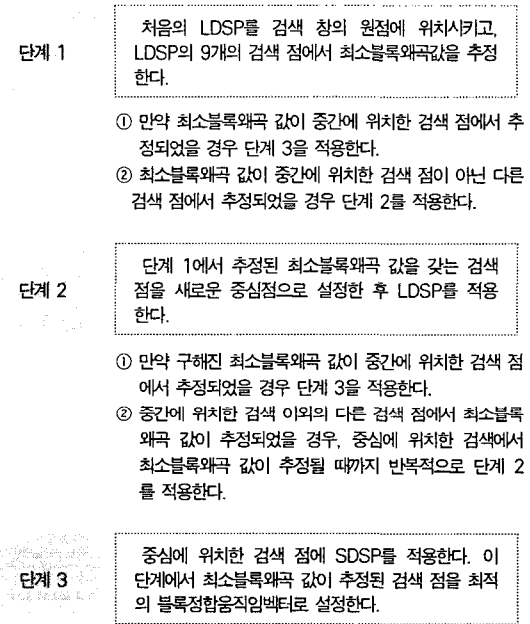


그림 1. 다이아몬드탐색알고리즘에서 적용되어지는 탐색패턴
(a) 큰 다이아몬드탐색패턴(LDSP)

(b) 작은 다이아몬드탐색패턴(SDSP)
Fig. 1 Search pattern employed in the Diamond Search algorithm
(a) Large Diamond Search Pattern(LDSP)
(b) Small Diamond Search Pattern(SDSP)

DS 알고리즘을 다음과 같이 요약할 수 있다.



IV. DS 탐색알고리즘 세부사항

LDSP 또는 SDSP를 적용한 탐색패턴이 탐색창의 경계 부근에 도달했을 경우, 탐색 창 외부의 탐색 점들은 모두 제외시키고 다이아몬드탐색알고리즘을 적용시킴으로써 탐색 영역을 제한시킨다. 또한 필요할 경우 다이아몬드탐색알고리즘을 임의의 큰 탐색 창에 적용시키는 것이 가능하다.

둘째로, 다이아몬드탐색알고리즘은 본질적으로 탐색단계의 수를 제한하지 않는다.

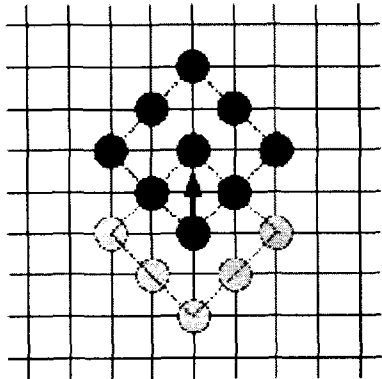
이론적으로는 탐색단계가 반복됨에 따라 탐색경로가 순환되어지는 경우도 발생할 수 있으나, 실제적으로는 이와 같은 경우는 발생되어질 경우는 드물다.

또한 이와 같은 경우가 발생한다고 하더라도 간단한 제약조건을 적용한다면 다이아몬드탐색알고리즘은 항상 실제의 해에 수렴하는 움직임벡터의 추정이 가능하다.

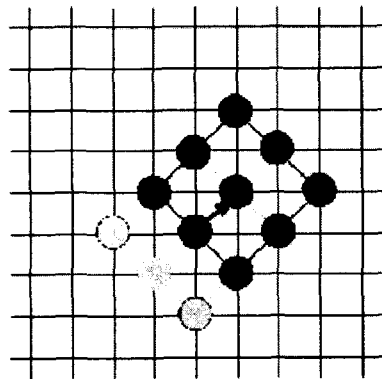
셋째로, LDSP를 적용할 경우, 임의의 단계의 탐색 점들은 직전 또는 직후의 탐색단계의 탐색 점들과 부분적으로 중첩되어진다. 각각의 단계에서 탐색 점들이 중첩되어질 수 있는 경우를 그림 3에 나타내었다.

그림 2(a) 또는 (b)과 같이, 직전의 단계에서 추정된 최소블록왜곡 값을 갖는 탐색점이 LDSP의 모퉁이에 위치하거나 또는 가장자리에 위치할 경우 임의의 단계에서는 단지 5 또는 3개의 탐색 점에서 최소블록왜곡 값을 추정한다.

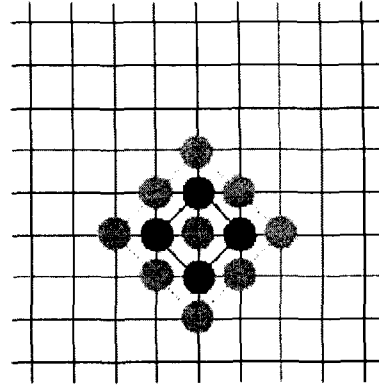
만약 임의의 단계에서 LDSP의 가운데 탐색 점에서 최소블록왜곡 값이 추정되어질 경우, SDSP를 탐색패턴으로 적용한다.



(a)



(b)



(c)

그림 2. LDSP를 적용하였을 경우 발생 가능한 탐색 점의 중첩의 3가지 경우. 이전탐색단계에서 최소블록왜곡 값이
(a)모퉁이에 위치한 탐색 점에서 추정된 경우,
(b)가장자리에 위치한 탐색 점에서 추정된 경우,
(c) 가운데 탐색 점에서 추정되었을 경우.

단, 검은 점이 현재 탐색단계에서의 탐색 점을 나타낸다.

Fig. 2 Three cases of checking points overlapping in LDSP. In the case of the point having minimum block distance is located at

- (a) one of the corner points
- (b) one of the edge points.
- (c) the center point.

탐색경로에 따른 탐색 점의 중첩을 나타내기 위하여 15×15의 탐색 창에서 다이아몬드탐색알고리즘을 적용하였을 때 발생 가능한 탐색경로의 한 예를 <그림 3>에 나타내었다.

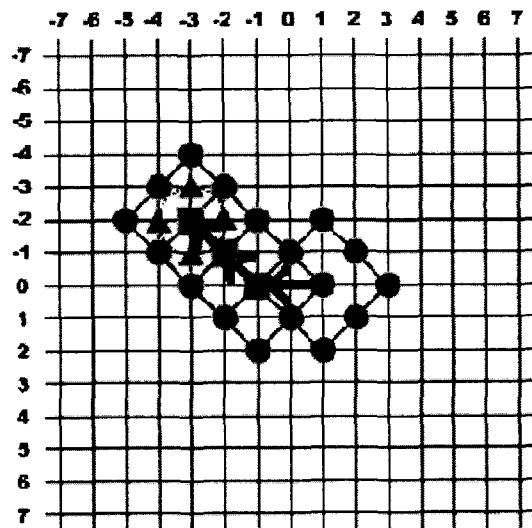


그림 3. 다이아몬드탐색알고리즘을 적용한 움직임벡터 추정의 한 예
Fig. 3. Example of motion vector estimation using diamond search algorithm

V. 구속조건을 적용한 다이아몬드탐색 알고리즘

본 논문에서 제안한 방법은 다이아몬드탐색알고리즘에 치환프레임차분(Displaced Frame Difference, DFD)를 적용한 것으로서, 프레임에 현재프레임과 다음 프레임으로부터 DFD 값을 추출하여 DFD값이 발생한 블록에만 선택적으로 다이아몬드탐색알고리즘을 적용하여 움직임벡터를 추정하고, DFD값이 존재하지 않은 블록은 움직임벡터의 값을 0으로 설정한 것이다. 제안한 방법을 <그림 4>와 같이 흐름도로 나타낼 수 있다.

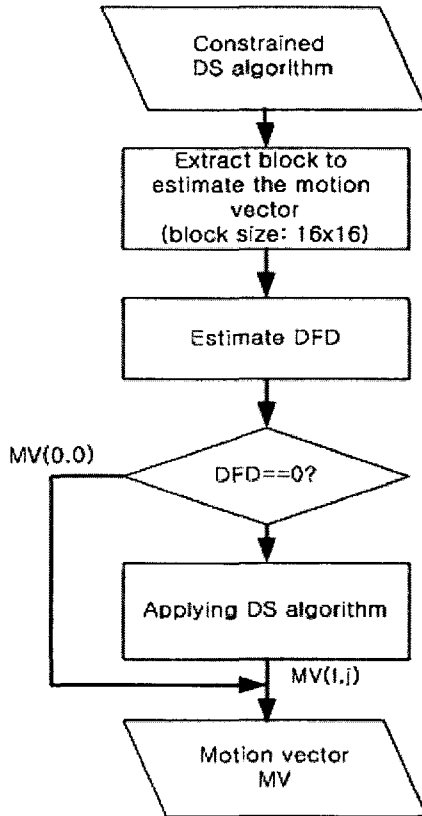


그림 4. 제안된 구속조건을 적용한 다이아몬드탐색알고리즘
Fig. 4 Proposed constrained diamond search algorithm

<그림 4>로부터, 현재 프레임에서 추출된 블록과 다음 프레임의 동일 위치의 블록간의 차를 구한 후 그 결과가 0 일 경우 움직임이 없는 안정된 블록으로 고려하여 움직임벡터의 값을 (0,0)로 설정하고, 0이 아닌 값일 경우 다이아몬드탐색알고리즘을 적용시킨다. 제안한 방법을 영상 시퀀스에 적용할 경우 움직임 벡터를 탐색하는 과정에서 발생할 수 있는 부분 해의 추정에 대한 오차를 극복할 수 있을 뿐 아니라 계산 량도 줄일 수 있을 것으로 생각되어진다.

VI. 실험결과

본 실험은 블록의 크기를 16×16로 설정하였다. 기존의 방법에 의한 결과와 비교하기 위하여 15×15 즉, 수평 또는 수직으로 ±7의 화소의 탐색 창 내에서 블록정합을 실시하였다. 그리고 예측프레임과 원래의 프레임간의 프레임 간격은 2로 설정하였다. 블록정합 시 계산 량을 줄이기 위하여 평균절대거리(Mean Absolute Distance, MAD)를 정합 기준으로 정하였다.

<그림 5>에 "Missa"영상 시퀀스에 대해 제안한 방법으로 움직임벡터를 추출한 후 과거프레임으로부터 현재프레임을 복원한 결과를 나타내었다. <그림 5>(a)와 (c)를 비교한 오차영상을 (d)에 나타내었는데, 결과로부터 입술부분을 제외한 대부분의 부분의 움직임이 비교적 양호하게 추출된 것을 시각적으로 확인할 수 있다.

<표 1> ~ <표 4>에 "Missa"와 "Claire"를 영상 시퀀스로 사용하였을 경우, 제안한 방법과 기존의 고속탐색알고리즘간의 오차 및 탐색 점의 수를 비교하였다. 실험결과로부터 움직임이 적은 영상 시퀀스에서 움직임벡터를 추정하였을 때, 제안한 방법을 적용하였을 경우 적은 수의 탐색 점만을 사용하더라도 향상된 오차 값을 갖는 움직임벡터를 추정하는 것이 가능하였지만, 움직임이 큰 영상 시퀀스에서는 기존의 방법에 비해 불리한 오차 값을 갖는 해를 추정하였다.

이것은 탐색거리가 기존의 블록정합에 비하여 가운데 탐색 점에 집중된 탐색패턴을 갖기 때문이며, 움직임의 크기에 따라 탐색패턴의 길이를 가변 시킬 경우 움직임이 다소 큰 시퀀스에서도 기존의 방법보다 향상된 결과를 나타낼 수 있을 것으로 예상된다.

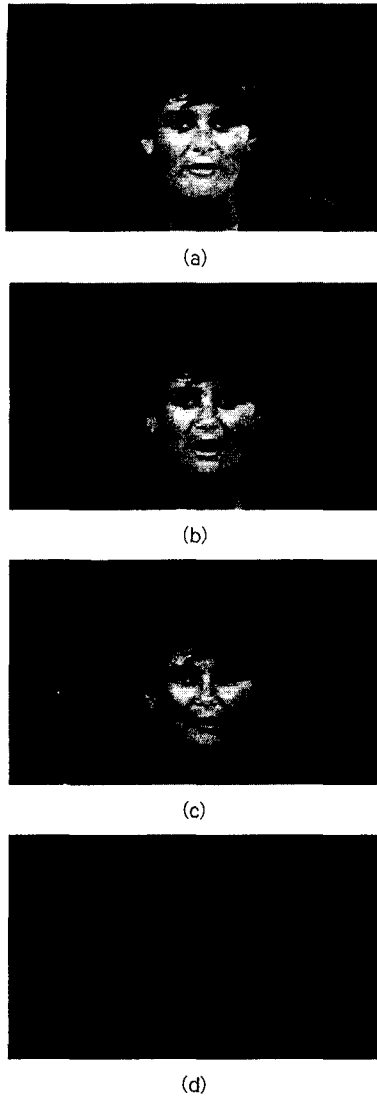
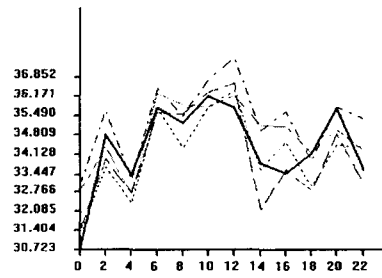


그림 5. "Missa" 영상 시퀀스에 제안한 구속조건을 적용한 다이아몬드탐색알고리즘을 적용하여 움직임벡터를 추출하였을 경우.

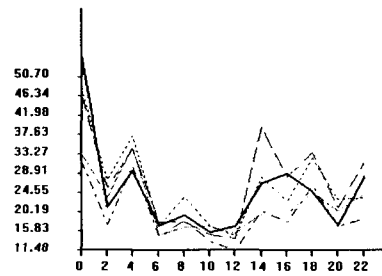
- (a) 24번 프레임 영상
 - (b) 22번 프레임 영상
 - (c) 22번 프레임영상으로부터 추정된 움직임벡터로부터 보상된 영상
 - (d) 24번 프레임 원래 영상과 보상된 영상간의 오차영상
- Fig. 5 In the case of block-matching motion estimation using proposed algorithm for "Missa" image sequences
- (a) current frame(24th frame)
 - (b) previous frame(22nd frame)
 - (c) reconstructed frame using motion vectors estimated 22nd frame.
 - (d) error image between current frame and reconstructed frame.

〈그림 6〉는 "Missa" 영상시퀀스에 제안한 구속조건을 적용한 DS알고리즘과 기존의 고속블록정합움직임추정알고리즘에 대한 오차와 탐색 점의 수를 비교한 것이다. 〈그림 5〉(a)와 (b)의 경우는 첨두신호대잡음비(Peak signal to noise ration, PSNR)와 평균자승오차(mean square error, MSE)를 비교한 것으로서, 영상프레임마다 다소 차이는 있으나 제안한 방법과 기존의 방법 간에 유사한 결과를 나타냄을 알 수 있었다.

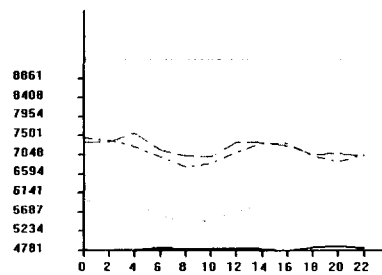
〈그림 5〉(c)는 움직임벡터의 추정에 필요한 탐색 점의 수를 비교한 것으로서, 제안한 방법의 경우프레임 당 평균 탐색 점의 수가 4700~4800개로서 5400~6000개의 탐색 점을 필요로 하는 기존의 DS알고리즘에 비해 적을 뿐 아니라 기존의 알고리즘에 비하여 적은 탐색 점을 필요로 하는 것을 확인할 수 있었다. 이와 같은 결과를 "Claire"영상과 "Missa" 영상에 대하여 〈표 1-4〉와 같이 나타내었다.



(a)



(b)



(c)

-----	Full Search Algorithm
-----	Three Step Search Algorithm(TSS)
-----	New Three Step Search Algorithm(NTSS)
-----	Four Step Search Algorithm(FSS)
-----	Diamond Search Algorithm(DS)
-----	Proposed Algorithm

그림 6. "Missa" 영상 시퀀스를 사용했을 경우, 기존의 블록정합알고리즘과 다이아몬드탐색알고리즘, 그리고 제안된 알고리즘의 비교

(a) PSNR, (b) MSE, (c) 사용되어진 탐색 점의 수 비교
Fig. 6 Comparison of PSNR, MSE, and the number of candidate points for "Missa" image sequence

표 1. "Claire" 영상 시퀀스를 사용했을 경우 제안한 방법과 기존의 블록정합알고리즘간의 첨두 신호대 잡음비의 비교(dB)
Table 1. PSNR comparison for "Claire" image sequence between proposed method and the conventional block-matching algorithms

	DS	FSA	FSS	NTSS	TSS	Prop.
0	34.46	36.76	37.10	32.75	32.98	35.37
2	34.32	38.21	37.95	33.16	34.82	35.26
4	32.75	36.75	36.53	32.27	33.60	32.80
6	32.75	35.02	35.01	31.77	32.48	34.61
8	33.77	39.10	39.03	35.18	36.37	35.33
10	35.77	38.68	38.57	34.29	35.56	38.58
12	33.40	38.68	35.76	31.44	34.36	35.09
14	33.00	37.49	37.32	33.20	34.89	34.62
16	30.51	35.14	35.05	31.32	32.47	31.52
18	33.80	36.31	36.23	34.01	33.95	36.09
20	37.15	39.92	39.97	39.62	38.82	37.05
22	34.06	38.03	37.99	32.04	34.22	34.83

표 2. "Claire" 영상 시퀀스를 사용했을 경우, 제안한 방법과 기존의 블록정합방법과의 움직임벡터추출과정에서 사용되어진 탐색점의 수 비교

Table 2. Comparison of the number of candidate points for "Claire" image sequence between proposed method and the conventional block-matching motion estimation algorithms

	DS	FSA	FSS	NTSS	TSS	Prop.
0	4449	67500	5631	5559	8100	5015
2	4431	67500	5629	5562	8100	4971
4	4557	67500	5704	5685	8100	4992
6	4557	67500	5728	5616	8100	5024
8	4389	67500	5579	5514	8100	5017
10	4362	67500	5497	5553	8100	4980
12	4584	67500	5774	5730	8100	5058
14	4428	67500	5642	5565	8100	5029
16	4506	67500	5725	5850	8100	5050
18	4563	67500	5760	5733	8100	5061
20	4221	67500	5424	5427	8100	5031
22	4362	67500	5520	5460	8100	4972

표 3. "Missa" 영상 시퀀스를 사용했을 경우 제안한 방법과 기존의 블록정합알고리즘간의 첨두신호대잡음비의 비교(dB)

Table 3. PSNR comparison for "Missa" image sequence between proposed method and conventional block-matching motion estimation algorithms.

	DS	FSA	FSS	NTSS	TSS	Prop.
0	30.74	32.86	33.13	31.46	31.33	30.70
2	34.84	34.40	35.68	34.02	33.72	34.84
4	33.44	32.70	33.26	32.76	32.40	33.44
6	35.84	36.33	36.46	35.61	35.77	35.73
8	35.23	35.88	35.51	35.58	34.38	35.25
10	36.24	35.80	36.79	36.35	35.80	36.12
12	35.81	36.21	37.53	36.64	36.40	35.87
14	33.83	35.17	34.99	32.16	33.63	33.82
16	33.49	35.14	35.62	33.62	34.57	33.52
18	34.16	34.22	33.94	32.84	33.00	34.21
20	35.74	35.02	35.80	34.86	34.50	35.76
22	33.64	34.27	35.39	33.16	34.41	33.64

표 4. "Missa" 영상 시퀀스를 사용했을 경우, 제안한 방법과 기존의 블록정합방법과의 움직임벡터추출과정에서 사용되어진 탐색 점의 수 비교

Fig. 4 Comparison of the number of candidate points for "Missa" image sequence

	DS	FSA	FSS	NTSS	TSS	Prop.
0	5988	77625	7457	7365	9315	4823
2	5931	77625	7389	7362	9315	4781
4	5913	77625	7269	7578	9315	4823
6	5622	77625	7003	7182	9315	4892
8	5445	77625	6791	7032	9315	4830
10	5511	77625	6860	7008	9315	4846
12	5703	77625	7124	7353	9315	4858
14	5910	77625	7332	7362	9315	4840
16	5841	77625	7321	7284	9315	4791
18	5775	77625	7036	7074	9315	4885
20	5640	77625	6902	7107	9315	4915
22	5700	77625	7077	7032	9315	4855

VII. 결론

본 논문에서는 구속조건을 적용한 다이아몬드탐색알고리즘을 적용한 움직임벡터의 추정방법을 나타내었고, 기존의 블록정합알고리즘을 적용한 움직임 추정의 결과들과 비교하

여 제안한 방법의 우수성을 나타내었다. 다이아몬드탐색알고리즘은 기존의 블록정합알고리즘에 비하여 탐색단계뿐 아니라 탐색 창의 크기가 제한되어있지 않기 때문에 우수한 움직임 추정결과를 나타낼 수 있을 뿐 아니라 영상 시퀀스가 갖는 다양한 움직임에 따라서 적응 적으로 탐색패턴을 설정하여 적용하는 것이 가능하다. 그러므로 제안한 구속조건을 적용한 다이아몬드탐색알고리즘은 MPEG-4의 비디오 압축에도 적용이 가능할 것으로 생각되어지며, 적은 계산 량으로 향상된 압축효과를 얻을 수 있을 것으로 사료되어진다.

- [7] Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, and Ashrat Ali Kassim, "A Novel Unrestricted Center-Biased Diamond Search Algorithm For Fast Block-Matching Motion Estimation," IEEE Trans. Circuits and Systems for Video Technology, vol.8, no.4, August 1998
- [8] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm For Fast Block-Matching Motion Estimation," IEEE Trans. Image Processing, vol.9, no.2, February 2000

참고문헌

- [1] Jyi-Chang Tasi, Chaur-Heh Hsieh, Shiu-Ku Weng, and Mao-Fu Lai, "Block-Matching Motion Estimation Using Correlation Search Algorithm," Signal Processing: Image Communication 13(1998), pp119-133
- [2] Borku Furht, Joshua Greenberg, and Raymond Westwater, "Motion Estimation Algorithm for Video Processing," Kluwer Academic Publishers, pp50-81
- [3] Yao Wang, Jorn Ostermaun, and Ya-Qin Zhang, "Video Processing and Communications," Prentice-Hall, pp141-216
- [4] Renxiang Li, Bing Zeng, and Ming L. Lion, "New Three-Step Search Algorithm for Block Motion Estimation," IEEE Circuits and Systems for Video Technology, vol.4, no.4, August 1994
- [5] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," IEEE Circuits and Systems for Video Technology, vol.6, no.3, June 1996, pp.313-317
- [6] Chun-Ho Cheung and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm For Fast Block Motion Estimation," IEEE, Trans. Circuits and Systems for Video Technology, vol.12, no.12, December 2002

저자소개



홍성웅
영남대학교 대학원 전자공학과
(공학박사)
현재 대구과학대학 컴퓨터정보과
교수