

고속 움직임 추정을 위한 적응형 육각 탐색 방법

정희원 전 병 태*, 김 병 천*

An Adaptive Hexagon Based Search for Fast Motion Estimation

Byung Tae Chun*, Byung Cheon Kim* *Regular Members*

요 약

본 논문에서는 일반적인 비디오 압축에 사용되는 블록 움직임 예측 방법을 위한 육각형 탐색 형태를 이용한 적응형 블록 정합 탐색 알고리즘(Adaptive hexagon based search : AHBS)을 제안한다.

제안하는 알고리즘은 다이아몬드 형태를 이용한 초기 탐색 과정과 제안한 두개의 육각 형태를 이용하여 적응적으로 탐색하는 과정으로 이루어져 있으며, 육각형 탐색 형태의 중앙값에 블록 정합을 위한 목적 함수의 최소 값이 발생할 때 그 과정을 종료하는 것으로 구성되어 있다. 또한, 탐색 과정에 있어 적용할 육각형 탐색 형태의 결정은 이전 탐색 과정에서 발생한 최소 값의 위치에 따라 이루어진다.

제안한 알고리즘의 성능측정은 전역탐색 방법을 포함 기존의 다양한 고속 탐색 방법들과 전역 탐색 방법의 결과와의 비교를 통하여 이루어졌다. 기존의 고속 탐색 방법에 비하여 본 논문에서 제안한 방법의 성능이 우수하고 그 수행 속도 또한 개선된 것을 실험 결과 알 수 있다.

Key Words : hexagon based search; motion estimation;

ABSTRACT

An adaptive hexagon based search(AHBS) algorithm is proposed in this paper to perform block motion estimation in video coding.

The AHBS evaluates the value of a given objective function starting from a diamond-shaped checking block, and then continues its process using two hexagon-shaped checking blocks until the minimum value is found at the center of checking blocks. Also, the determination of which checking block is used depends on the position of minimum value occurred in previous searching step.

The AHBS is compared with other fast searching algorithms including full search(FS). Experimental results show that the proposed algorithm provides competitive performance with slightly reduced computational complexity.

I. 서 론

움직임 예측(motion estimation)은 움직임 보상 비디오 코딩(motion compensated video coding)에서 중요한 역할을 하고 있으며, 잘 알려진 비디오 압축 표준인 H.26x와 MPEG-x[1]-[4]에서 반드시 필요한 부분이라고 볼 수 있다. 블록 정합(block matching)

기법이 하드웨어 및 소프트웨어에서 구현하기 쉽기 때문에 H.26x와 MPEG-x[1]-[4]에서 널리 사용되는 방법이다. 비디오 시퀀스를 구성하는 각 프레임들은 작은 블록들로 나누어지며, 각 블록들은 참조 프레임(reference frame)의 탐색 범위에 있는 블록들 중에서 최적의 움직임 벡터를 구할 수 있다.

최근 몇 년 동안 실생활에 사용하기 위한 많은

* 국립 한경대학교 웹정보공학과(chunbt@hnu.hankyong.ac.kr)
논문번호 : 040077-0211, 접수일자 : 2004년 2월 12일

효과적인 블록 정합 알고리즘(Block-Matching Algorithms: BMA)이 연구되고 개발되어 지고 있다.

블록 정합 과정에서 완전 탐색(Full-Search : FS) 방법은 찾고자 하는 블록의 화소들과 참조 프레임의 탐색 영역에 있는 모든 화소들과 전부 비교하는 방법이다. 이 방법은 전체 화소 탐색을 수행하므로 정확도가 대단히 높다고 볼 수 있다. 그러나 전체 영역을 탐색하기 때문에 많은 연산량이 소요되어 시스템에 부당을 주는 문제가 있다. 완전 탐색 방법의 처리 시간의 문제를 해결하기 위하여 탐색 영역 안에서 탐색 점을 줄이기 위한 다양한 블록 정합 알고리즘이 개발되어 왔다.

그 대표적인 방법들은 3단계 탐색(New Three-Step Search : NTSS) 방법[5], 2-차원 로그(2-D Log) 탐색 방법[6], 블록 기반 기울기 감소 탐색(Block Based Gradient Descent Search : BBGDS) 방법[7], 마름모꼴 탐색(Diamond Search : DS) 방법[8], 1차원 기울기 감소(one-Dimensional Gradient Descent Search: 1-DGDS) 탐색 방법[9], 육각형 기반 탐색(Hexagon based search) 방법[10] 등이 있다.

TSS(Three-Step Search) 방법이 통일된 탐색 형태를 기반으로 하기 때문에 많은 응용 분야에 사용되고 있으나 움직임이 작게 발생하는 움직임 탐색에는 효과적이지 못하다. BBGDS, 2-D log, DS 및 1-DGDS 방법들은 특정 탐색 패턴에서 각 점들을 평가한다. 이들의 탐색은 블록 정합에서 최소 왜곡(minimum distortion)을 얻기 위하여 기울기 감소 방향으로 탐색 경로를 움직인다. 그러나 이들 방법들은 빠른 움직임이 있는 연속된 영상에서는 국부 최소점(local minimum)에 쉽게 빠질 수 있다. 가령 DS와 1-DGDS 방법들과 같이 몇몇 방법들은 앞서 언급한 단점을 줄이기 위하여 변화가 심한 블록에서 수렴을 조사한다.

일반적으로 전역 최소점(global minimum)을 찾기 위한 기울기 감소 탐색 방법은 초기의 기울기 방향, 시작점 및 기울기 방향에 매우 민감하다[9][12]. 여기서 전역 최소점을 쉽게 찾는다는 말은 좋은 성능을 얻을 수 있음을 의미하는 것이며, 전역 최소점은 초기에 적절한 감소 방향(descend direction) 및 시작점을 선택함으로써 이룰 수 있다. 뿐만 아니라 기울기 감소 방법의 단점을 극복하기 위하여 탐색 점들이 균일 분포 형태로 이루어진 탐색 패턴을 사용한다.

본 논문에서는 기울기 감소 탐색을 기반으로 하는 육각형 탐색 형태(hexagon search pattern)를 사용하여 효과적인 움직임 예측 방법을 제안하고자 한다. 제2장에서는 새로이 제안된 탐색 형태와 전략에 대하여 자세히 설명한다. 제3장에서는 실험 결과에 대하여 설명하고, 제4장에서는 향후 연구 방향에 대하여 논하겠다.

II. 적응형 육각 탐색 형태에 기반한 블록 움직임 예측 방법

2.1 적응형 육각 탐색의 형태

일반적으로 대부분의 블록 움직임 예측 알고리즘은 각 영상을 $N \times N$ 작은 블록으로 나눈다. MPEG-1[3], MPEG-2[4] 및 H.263[2]에서 사용하는 블록 크기는 보통 16×16 로 사용하고 있다. 블록 정합을 위한 탐색 영역은 보통 15×15 로 구성되며, 이것은 255 개의 후보 탐색 점들을 갖는다는 의미한다. 이러한 탐색에서 고려해야 할 주된 2가지는 탐색 영역 안에서 적당한 후보 점들을 선택하는 방법과 최소 계산 시간이 소요되는 블록 정합 방법을 고려해야 한다.

그림1은 본 논문에서 제안한 2개의 육각 탐색 형태를 보여주고 있다. 각 형태를 보면 8개의 후보 탐

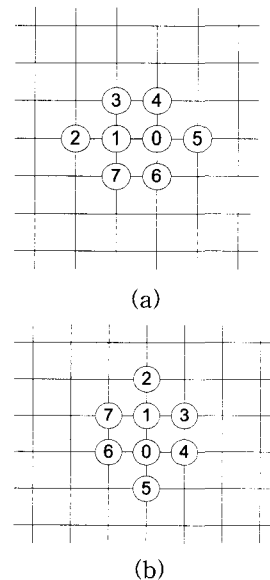


그림 1. 기본 육각에 기반한 탐색 형태
(a) 편향 수평 (b) 편향 수직

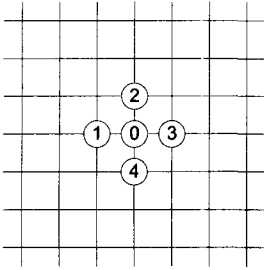


그림 2. 시작을 위한 탐색 형태

색 점들로 구성되어 있다. 각 점들은 수평 및 수직 방향으로 편향(bias)된 형태로 구성되어 있으며, 그림1(a)에서는 수평축에서 후보 영역을 조사하기 위하여 수평축에 기반한 형태(Horizontal biased Pattern : (HBP))를 이용하고, 그림1(b)에서는 수직축에 기반한 형태(Vertical biased Pattern(VBP))를 보여주고 있다. 그림 1과 같이 수평 및 수직 방향으로 편향된 형태는 기울기 감소 방향에 가중치를 두어 전역 최소점으로 이동하기 위한 목적을 반영하고 있다고 볼 수 있다. 각 형태에 구성된 점들은 정점(vertex), 표면(surface) 및 내부 점들(inner points)로 나뉘어 진다. 정점은 "2"와 "5" 위치가 되며, 표면은 "3","4","6","7" 위치가 된다. 내부 점들은 "0"과 "1"이 된다. 특히, "0"은 중심점으로 정의한다.

적응형 육각에 기반한 기울기 감소 탐색 (Adaptive Hexagon Based Search : AHBS) 알고리즘을 객관적으로 설명하기 위하여, 블록 탐색을 위한 객관적 함수로 사용되고 있는 블록 왜곡 측정 (block distortion measurement)을 정의해야 한다. 본 논문에서는 사용된 객관적 함수는 MAD(Mean Absolute Difference)를 사용한다.

$$MAD(u, v) = \sum_{i=1}^N \sum_{j=1}^M |I_i(i, j) - I_{t+1}(i+u, j+v)| \quad \dots\dots(1)$$

여기서 I_t and I_{t+1} 는 비교될 현재 프레임(t th frame), 참조 프레임($t+1$ th frame)에 있는 블록들을 말하며, (u, v) 는 후보 움직임 벡터의 위치이며, N, M 은 블록의 크기를 나타낸다.

그림 3은 다음 단계에서 적용되어야 할 패턴을 선택하기 위한 전략을 보여주고 있다. 다음 단계에

대한 탐색 형태는 현재 탐색 상태에서 최소 MAD 값을 갖는 위치가 선택되어진다. 예를들어, HBP의 경우 1(a)에서 보여지는 것과 같이 최소 MAD값이 2와 5에서 발생했다면 다음 탐색 단계에 대한 탐색 형태는 HBP가 된다. 이와 반대로 3, 4, 6, 7에서 발생했다면다음 탐색 형태는 HBP에서 VBP로 변경되어 진다. 그림 3은 수평 및 수직에서 발생할 수 있는 최소 MAD 값의 위치에 따라서 탐색 형태를 선택하는 전략을 설명해 놓은 그림이다. 이러한 선택 전략의 목적은 현재의 탐색 과정에서 사용된 탐색 형태에 따라 발생하는 탐색 영역의 협소함을 보완하기 위한 것이다. 탐색 과정에서 탐색 영역은 균일한 분포를 이루도록써 후보 점들의 선택 확률을 균등하게 배분해야 하며, 이를 위하여 탐색 형태를 원(circle)의 형태로 구성하는 것이 이상적이다.

그러나 제안하는 AHBS 알고리즘에서 사용하고 있는 편향 수직, 수평 탐색 형태에 의하여 탐색 영역의 비 균등성이 발생함으로 이를 보완하기 위하여 현재 탐색 과정에서 제외된 탐색 영역에 대하여 다음 탐색 과정에서 이를 포함시켜 탐색을 실시한다. 또한, 그 탐색 계획은 기울기 감소 방향과 관련을 갖고 있다. 최소 MAD 값이 발생되어진 위치

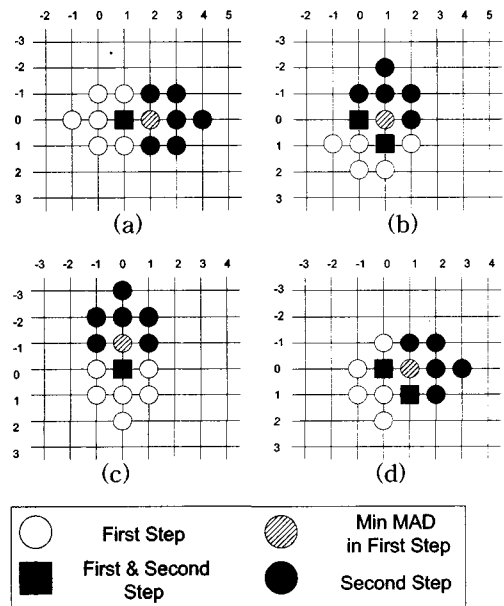


그림 3. 각 위치에 대한 HBP와 VBP의 탐색 계획의 예. (a) HBP 계속 탐색 (b) HBP에서 VBP로 탐색 형태의 변경 (c) VBP 계속 탐색 (d) VBP에서 HBP로 탐색 형태 변경

는 감소 방향의 결정에 사용되며, 이러한 기울기 감소 방향을 고려하여 다음 탐색 단계에 적용한다. 각 패턴에 있는 정점들(vertex points)은 기울기 감소 방향을 의미하며, 이들 정점들은 현재 탐색 단계의 기울기 감소 방향과 동일 하다. 이와 반대로 한 패턴에서 기울기 감소 방향으로 나타나 있는 표면 점들(surface points)은 현재 방향으로 변경되어진다.

2.2 AHBS에 기반한 블록 움직임 예측 방법

탐색 과정에 있어서 다음 탐색 단계에 대한 패턴을 결정하기 위한 탐색 절차는 패턴의 모양에 따라 HBP와 VBP 2가지로 나뉘어 진다. 탐색 패턴의 선택 방법은 MAD 값이 어디서 최소가 발생하였는가에 따라 결정 되어진다. 다음 탐색 단계에 사용되어 질 형태를 결정하기 위한 2가지 다른 전략이 있다.

한 가지는 정점 탐색(vertex searching) 방법이고 다른 한 가지는 표면 탐색(surface searching) 방법이다. 정점 탐색은 양 패턴의 정점과 관련이 있고, 표면 탐색은 그들의 표면과 관련이 있다.

AHBS에서 사용된 제안된 알고리즘을 요약하면 다음과 같다.

Step 1) 그림 2에 나타난 초기 탐색 형태(SSP)을 미리 지정된 탐색 영역내의 중심 위치(0,0)에 위치하여 탐색 형태내의 각 후보점들에 대하여 MAD 값을 계산한다. 계산된 MAD 값중에서 최소 MAD 값이 SSP의 중심에 위치할 경우, 현재 탐색 과정을 중지하며, Step 4)로 이동한다. 최소 MAD 값이 SSP의 1 혹은 3의 위치에서 발생할 경우 HBP를 탐색 패턴으로 선정하고 Step 2)로 이동하여, SSP의 2 혹은 4의 위치에서 발생할 경우 VBP를 탐색 패턴으로 선정하고 Step 3)으로 이동한다.

Step 2) (HBP의 선택시) 이전 탐색 과정에서 최소 MAD 값이 발생한 위치를 HBP의 중심값으로 선택하고 HBP를 구성하다. 구성된 HBP에 의한 각 후보점들의 MAD값을 계산하며, 최소 MAD값이 발생한 위치에 따라 아래와 같은 과정으로 탐색 과정을 실시한다.

- 내부점에서 발생시 : 탐색 과정을 종료되며, Step 4)로 이동한다.
- 정점에서 발생시 : 다음 탐색 과정은 Step 2)를

반복한다. 단, HBP의 5번 위치에서 최소 MAD 값이 발생했을 경우, 다음 탐색 과정에서 사용되는 HBP의 중심값은 수평 방향으로 +1 증가한 위치로 선택한다.

- 표면점에서 발생시: 다음 탐색 과정은 Step 3)으로 이동한다. 단, HBP의 3 혹은 4번 위치에서 최소 MAD 값이 발생했을 경우, 다음 탐색 과정에서 사용되는 VBP의 중심값은 수직 방향으로 1 감소한 위치로 설정한다.

Step 3) (VBP의 선택시) 이전 탐색 과정에서 최소 MAD 값이 발생한 위치를 VBP의 중심값으로 선택하고 VBP를 구성하다. 구성된 VBP에 의한 각 후보점들의MAD값을 계산하며, 최소 MAD값이 발생한 위치에 따라 아래와 같은 과정으로 탐색 과정을 실시한다.

- 내부점에서 발생시 : 탐색 과정을 종료되며, Step 4)로 이동한다.
- 정점에서 발생시 : 다음 탐색 과정은 Step 2)를 반복한다. 단, VBP의 2번 위치에서 최소 MAD 값이 발생했을 경우, 다음 탐색 과정에서 사용되는 HBP의 중심값은 수직 방향으로 -1 감소한 위치로 선택한다.
- 표면점에서 발생시: 다음 탐색 과정은 Step 3)으로 이동한다. 단, VBP의 3 혹은 4번 위치에서 최소 MAD 값이 발생했을 경우, 다음 탐색 과정에서 사용되는 VBP의 중심값은 수평 방향으로 +1 증가한 위치로 설정한다.

Step 4): 현재 최소 MAD 값이 발생한 위치를 최종 움직임 벡터로 선택한다.

최소값이 발생된 위치는 예측된 움직임 벡터로서 선택이 되어지고, 현재 블록의 정합(matching) 과정은 완료되어 진다.

그림 4는 AHBS를 사용한 탐색 경로의 예를 보여주고 있다.현재 탐색을 실행하는 블록의 실제 움직임 벡터가 $(mx, my) = (4, -2)$ 으로 가정한다면, 탐색 영역의 중심인 $(0, 0)$ 에서 시작 한다. 5개의 후보 점들 각각에 대하여 MAD를 계산하고 비교되어야 한다. 만약 최소점이 $(0,0)$ 에 위치한다면, 탐색 과정은 종료되어 지며 최종 움직임 벡터는 $(0,0)$ 이 된다.반면 다른 4개의 점에서 최소가 발생되어 지면 탐색은 계속 되어져야 할 것이다. 만약 그림2에서 3으로 표시된 $(1,0)$ 에서 최소값이 발생하면, 다음

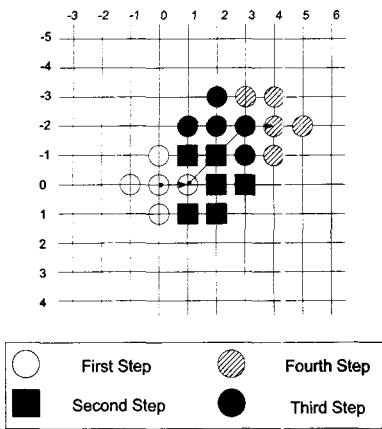


그림 4. AHBS 설명을 위한 예. 움직임 벡터는 (4,-2)

탐색 형태는 HBP이고 HBP의 중앙점은 (2,0)에 위치하게 된다. 밝은 회색 동그라미는 2번째 탐색 단계에 대한 패턴의 위치를 표현해 놓은 것이다. 위 예제에서 최적의 정합 후보 점을 찾기 위해 4 번의 탐색 단계를 요구하고 있으며, 각 단계는 밝은 동그라미와 어두운 동그라미로 표시되었다. 최종 탐색 단계에 대해서는 줄 무늬 모양의 동그라미로 표시해 놓았다. 또한 각 단계의 최소 MAD 값을 갖는 점들은 화살표에 의하여 경로를 표시했다.

III. 실험 결과

본 실험에서는 5가지(FS 방법, BBGDS 방법, DS 방법, TSS 방법, AHBS 방법) 블록 정합 방법들에 대하여 실험을 하였다. 동영상(foreman, paris, student 동 영상)에 150 프레임, football 동영상에서 125 프레임, garden 동영상에서 110 프레임을 본 실험에서 실험에 사용했다.

Garden 동영상은 움직이지 않은 객체를 포함하고 있으나 빠른 카메라 움직임(패닝(panning) 동작)이 발생하고 있다.

Football 동영상은 주로 빠르고 큰 객체 움직임이 발생하고 있다. 다른 동영상들은 비디오 영상 회의와 같이 제한된 객체 움직임과 고정된 배경들로 구성되었다.

블록의 크기는 16x16이고, 탐색 영역은 15화소로 고정되어 있다. 본 논문에서는 사용된 영상은 소스 코드 "mpeg2encode" [11]을 기반 한 MPEG2 encoder로 인코딩하여 사용 했으며, 실험은

Pentium-IV 1.4Gh 상에서 실험했다. MSE(Mean Square Error)는 각 알고리즘의 성능을 비교하기 위하여 사용되어졌다. MSE는 다음과 같이 계산되어진다.

$$MSE = \frac{\sum_{i=1}^N \sum_{j=1}^M (I_t(i, j) - \hat{I}_t(i, j))^2}{N \times M} \dots\dots (2)$$

여기서 프레임 크기는 $N \times M$, $I_t(\cdot, \cdot)$ and $\hat{I}_t(\cdot, \cdot)$ 는 각각 원 프레임과 움직임 보상(the motion compensated) 프레임이다.

표 1. 각기 다른 영상에 대한 FS, BBGDS, DS, TSS, AHBS의 성능 비교

Sequ- ence	GARDEN		FOOTBALL		FOREMAN		PARIS		STUDENT	
	MSE	Sec.	MSE	Sec.	MSE	Sec.	MSE	Sec.	MSE	Sec.
FS	187.3	46	114.89	51	13.85	66	33.56	53	9.17	54
BBGDS	357.01	26	135.37	29	19.63	41	38.70	40	9.74	39
DS	288.81	30	124.99	32	15.64	45	34.28	43	9.22	42
TSS	248.51	28	134.76	30	18.95	43	82.12	41	18.1	41
AHBS	243.18	30	119.82	33	14.49	45	33.9	42	9.18	40

그림 5는 각 알고리즘의 성능을 보여주고 있으며, 이것은 화소 당 MSE 값에 대한 실험 동영상의 결과값을 보여주고 있다. 표1에서 보여주는 것과 같이 본 논문에서 제안한 AHBS 방법이 다른 시험 영상보다도 성능이 우수함을 볼 수 있다. 특히 Paris, students 동영상에서는 AHBS의 MSE는 FS의 MSE에 매우 근접함을 볼 수 있다. 다른 동영상들에서는 AHBS 방법이 다른 알고리즘보다 가장 낮은 MSE 값을 가지고 있으며, FS의 MSE와 매우 근접함을 볼 수 있다.

표2와 표3은 속도의 성능 향상을 보여주고 있다. 표에서 보느바와 같이 AHBS는 FS 방법을 제외한 다른 비교된 탐색 알고리즘보다 성능이 우수함을 볼 수 있다. 특히 AHBS 방법의 성능은 FS의 성능에 매우 근접할 뿐 아니라, student 동영상에서는 놀랄만한 속도 향상을 볼 수 있다. 특히 student 영상의 경우 제안하는 AHBS 방법은 FS 방법과 비교할 시 성능은 거의 동일함과 동시에 매우 향상된 계산 속도를 가짐을 볼 수 있다. 실험 결과는 AHBS 방법이 일반적으로 보다 효과적임을 볼 수 있으며, 이것은 카메라의 팬닝(panning), 작거나 큰

움직임의 움직임에 불구하고 다른 빠른 탐색 알고리즘보다도 안정된 성능을 보유하고 있다는 것을 의미한다.

본 논문에서는 AHBS의 성능과 그것의 속도 사이의 상보 관계(trade off)를 조사를 했다. 표2와 표3은 AHBS의 상보 관계를 정의하기 위한 중요한 정보를 제공하고 있다. foreman, Paris, and student 동영상에서 AHBS 방법이 어떤 고속 탐색 알고리즘보다도 우수한 성능을 보여주고 있는 반면에 처리 소요 시간에 있어서 복잡도의 증가를 볼 수 있다.

이러한 결과는 기존의 고속 탐색 알고리즘과 비교하여 제안하는 AHBS 알고리즘이 비디오 회의와 같이 적은 움직임이 발생하는 응용 부분에 있어 계산 속도의 증가를 부담하나 좀 더 선명한 영상의 획득이 요구될 시 적합함을 알 수 있다. 또한 AHBS 방법이 Football, Garden 영상에서 실험 알고리즘들 중 최소의 블록 왜곡(block distortion) 현상을 보여주고 있으나, 다른 탐색 알고리즘 보다 약간의 처리 시간이 더 소요됨을 볼 수 있었다. AHBS 방법은 BBGDS, TS 방법보다도 블록 왜곡 측정에 약 10% 정도의 시간이 더 소요됨을 볼 수 있었으나, 블록 왜곡 현상은 10% 이상 감소함을 볼 수 있었다. Football 영상에서 FS의 블록 왜곡은 AHBS 방법과 매우 근접한 값을 갖는 것을 볼 수 있었다. Garden 영상에서 BBGDS, DS 보다도 평균 15%의 블록 왜곡이 적음을 볼 수 있었다.

표 2. AHBS 대 다른 탐색 알고리즘 왜곡 감소의 비율(%)
표시는 증가를 나타냄

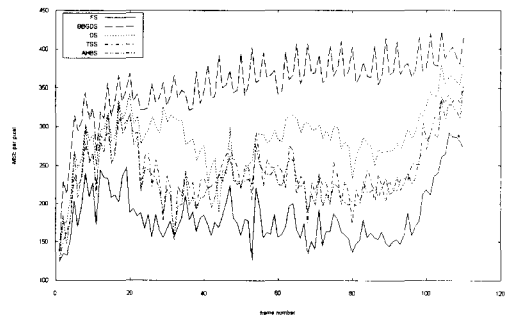
Sequence	GARDEN	FOOTBALL	FOREMAN	PARIS	STUDENT
FS	-29.8	-4.3	-4.6	-1.01	-0.1
BBGDS	31.8	11.4	26.2	12.4	5.7
DS	15.8	4.1	7.4	1.1	0.4
TSS	2.1	11.1	23.5	58.7	49.2

표 3. AHBS 대 다른 탐색 알고리즘의 속도 향상 비율(%)

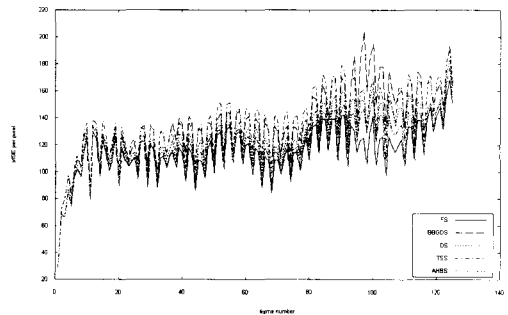
Sequence	GARDEN	FOOTBALL	FOREMAN	PARIS	STUDENT
FS	34.8	35.3	31.8	20.7	25.9
BBGDS	-15.3	-13.7	-9.7	-5	-2.5
DS	1	-3.1	1	2.3	4.8
TSS	-7.1	-10	-4.6	-2.4	2.4

실험 결과는 기존의 고속 블록 정합 알고리즘에 비하여 제안된 AHBS 방법이 계산 량에 있어서 약간 증가함을 볼 수 있으나 국부 최소점(Local minima)에 빠지지 않고 블록 왜곡을 최소화하는 효과적인 방법이 될 수 있음을 알 수 있다. 즉, 제안하는 AHBS 알고리즘은 미세한 양의 속도 증가 부담을 가지면서, 기존의 고속 탐색 알고리즘보다 훨씬 향상된 성능이 필요한 경우에 적절함을 보여주고 있으며, 선명한 영상의 획득에 효과적으로 사용될 수 있음을 알 수 있다.

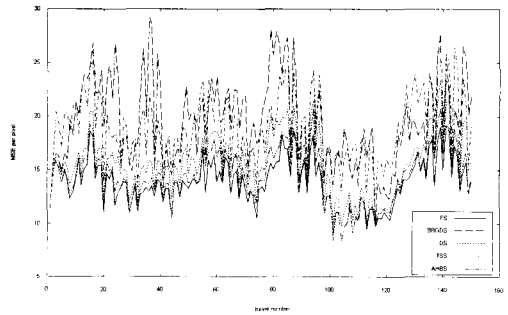
그림5는 프레임 단위의 각 시험 탐색 알고리즘의 실제 성능을 보여주고 있다. AHBS 방법이 블록 왜곡에 있어서 좋은 성능을 보여주고 있음을 볼 수 있다.



(a)

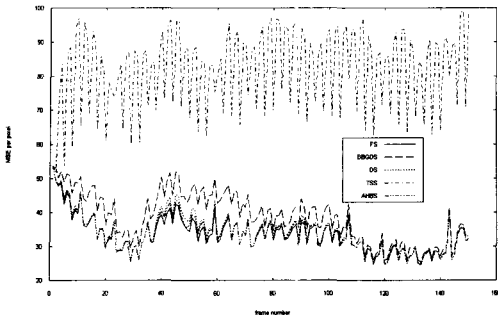


(b)

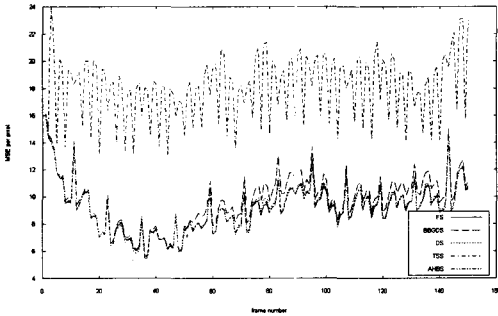


(c)

참고문헌



(d)



(e)

그림 5. MSE 값은 16 x 16 화소의 블록 크기와 7 에서 +7 까지의 범위에서 FS, BBGDS, DS, TSS, AHBS를 적용한 원영상과 움직임 보상 프레임 사이의 값 (a) 정원 (b) 측구 (c) 공장장 (d) 파리 (e) 학생

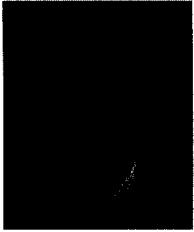
IV. 결론

본 논문에서는고속 블록 기반 움직임 예측에 있어서 적응형 육각형 기반 탐색 방법(adaptive hexagon based search(AHBS)을 제안했다. 사용된 2개의 탐색 형태는 수직 및 수평 방향에 기반하고, 이것들은 각각 최소 블록 왜곡 값을 갖는 위치를 찾게 된다. 또한, 적응적으로 두 개의 육각형 탐색 형태를 이용함으로써 원(circle)의 형태를 가지는 탐색 과정과 유사하게 균일하게 분포된 선택 확률을 가지는 탐색 영역의 구성이 가능함을 알 수 있었다. 실험 결과 AHBS 방법이 보다 효과적이고 효율적이며 기존의 고속 탐색 알고리즘(BBGDS, DS, and TS방법) 과 비교할 때 강건함(robust)함을 볼 수 있다. 또한 AHBS는 비디오 회의 응용 분야 및 빠르고 큰 국부 움직임을 갖고 있는 비디오 영상에 적용시 계산 속도 개선 및 성능의 강건함을 알 수 있었다.

- [1] T. Koga, K. Iinima, A .Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video sequences", in *Proc. National. Telecommunication Conf.*, 1981, pp. C9.6.1-9.6.5
- [2] "Video Coding for Low Bitrate Communication" International Telecommunication Union, ITU-T Recommendation H.263, July 1995
- [3] MPEG-1 CommitteeDraft, ISO/IEC JTC1/SC29/WG11, ISO/IEC CD 11172: Information Technology, Dec. 1991
- [4] MPEG-2 CommitteeDraft, ISO/IEC JTC1/SC29/WG11, ISO/IEC CD 13818: Information Technology, Dec. 1993
- [5] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 6, pp. 313-317, June 1994
- [6] J.R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1981
- [7] L. 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. 412-422, Aug. 1996.
- [8] J.Y. Tham, S. Ranganath, M. Ranganath, and A.A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 369-377, Aug. 1998
- [9] O. T.-C. Chen, "Motion estimation using a one-dimensional gradient descent search", *IEEE Trans. Circuits syst. Video Technol.*, vol. 10, pp.608-616, Jun. 2000
- [10] C. Zhu, X. Lin, and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.* Vol. 12, pp. 349-355, May 2002
- [11] MPEG Software Simulation Group, MPEG2 video codec version 1.2
- [12] R. Fletcher, *Practical Methods of Optimization.* New York: Wiley, 1991

전 병 태(Byung Tae Chun)

정회원



1986년 2월 : 한남대학교

전산과 학사

1989년 8월 : 숭실대 대학원

전산과 석사

2001년 2월 : 고려대 대학원

컴퓨터학과 박사

1989년 8월~2004년 2월

한국전자통신연구원

선임연구원

2004년2월 ~현재 : 국립 한경대학교 웹정보공학과

<관심분야> 멀티미디어 영상처리, 영상 압축 및 복원, 생체 인식 분야

김 병 천(Byung Cheon Kim)

정회원



1988년 2월 : 한남대학교

전산과 학사

1990년 8월 : 숭실대 대학원

전산과 석사

1999년 2월 : 명지대 대학원

컴퓨터공학과 공학박사

1991년2월 ~현재 : 국립

한경대학교 웹정보공학과

부교수

<관심분야> 기계학습, 컴퓨터 비전 에이전트 시스템 분야