

## 고속 블록정합 움직임 예측을 위한 최적의 탐색 패턴

임 동근, 호요성<sup>o</sup>

광주과학기술원 정보통신공학과

## Optimal Search Patterns for Fast Block-Matching Motion Estimation

Dong-Keun Lim, Yo-Sung Ho<sup>o</sup>

Kwangju Institute of Science and Technology (K-JIST)

dklim@kjist.ac.kr, hoyo@kjist.ac.kr

## 요약

움직임 탐색은 동영상 부호화에서 중요한 역할을 한다. 본 논문에서는 고속 블록정합 움직임 탐색 방법에 적합한 탐색 패턴을 분석하였다. 시간적으로 인접한 영상 블록들의 움직임에 대한 단순한 가정을 세우고, 블록정합 방법을 블록의 모양과 크기의 함수로 삼아, 블록간 상관도의 관점에서 동일 상관도(equi-correlation)를 가지는 탐색 패턴을 유도하였다. 또한 컴퓨터 모의시험을 통하여 제안한 분석의 타당성을 검증하였고, 실제 영상에 움직임 탐색을 적용하여 구한 실제적인 탐색 패턴과 비교하였다. 유도된 탐색 패턴과 실제적인 탐색 패턴은 모두 다이아몬드 형태를 가지고 있었다. 유도된 탐색 패턴에 경계 조건을 적용하였을 때, 다이아몬드형에서 원형으로 변하다가 최종적으로는 블록과 같은 모양으로 수렴하였다. 제안한 접근방법은 MPEG-2 표준화 작업에 제안된 다이아몬드 탐색법과 지역 탐색법의 이론적인 근거를 제시하며, 나아가서는 모든 고속 블록정합 방법에서 채택하고 있는 탐색 패턴의 의미를 설명할 수 있다.

## 1. 서론

최근 무선 통신이나 인터넷을 이용한 실시간 멀티미디어 서비스에 대한 관심이 증가하고 있다. 전송 대역폭이 제한된 채널에서 멀티미디어 데이터를 효과적으로 전송하기 위해 여러 가지 압축 알고리즘이 사용되고 있다. 특히, 영상 신호에는 시간적으로 연속하는 화면 사이에 중복되는 정보가 많으므로, 이러한 중복 정보를 효과적으로 제거하기 위해 움직임 예측 기법을 사용한다.

영상 신호에서 물체의 움직임을 예측하기 위해 가장 많이 사용되는 블록정합 방법에서는 각 화면을 일정한 크기의 사각형 블록으로 나누고, 정합 기준(matching criteria)에 따라 이전 화면의 탐색 영역에서 현재 화면의 각 블록과 가장 비슷한 블록을 찾아낸다. 선택된 블록은 현재 블록을 예측하는데 사용되며, 두 블록 사이의 상대적인 위치 차이는 현재 블록의 움직임 벡터로 표현된다.

전역탐색 방법은 오차 함수값을 최소화하는 움직임 벡터를 구하기 위해 가능한 모든 경우를 탐색하기 때문에 일반적으로 좋은 성능을 나타낸다. 그러나, 탐색 영역 전체에서 적합한 움직임 벡터를 찾기 때문에 많은 계산량이 필요하다. 계산량을 줄이기 위해 여러 종류의 고속 탐색 방법들이 제안되었으나, 대부분의 고속 탐색 방법들은 경험 예측적인

(heuristic) 접근 방법을 사용하여 계산량을 줄이기 때문에 복원된 영상의 화질을 다소 저하시킨다.<sup>1,2</sup>

본 논문에서는 움직임 벡터를 찾는 데 요구되는 계산량을 줄이도록 블록정합 방법을 위한 최적의 탐색 패턴을 수식적으로 유도하고, 실제적인 움직임 정보와 비교하여 제안한 방법이 옳다는 것을 입증하였으며, 기존의 탐색 방법과 비교하였다.

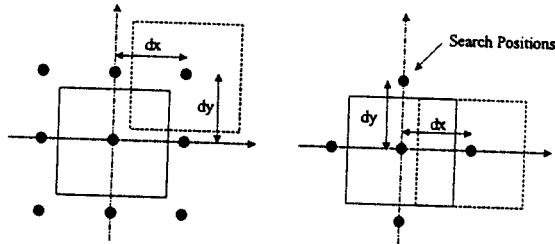
제안된 탐색 패턴의 유도 방법은 사각형 블록 뿐만 아니라, MPEG-4 표준과 같이 임의의 형태를 가진 블록을 사용하는 블록정합 방법에서도 최적의 탐색 패턴을 예측할 수 있다. 최근 많이 사용되는 다이아몬드 탐색법과 지역 탐색법의 블록정합 방법에서 채택하고 있는 탐색 패턴의 형태는 유도된 탐색 패턴의 부분집합으로 볼 수 있다. 따라서, 제안된 방법을 사용하여 고속 블록정합 방법에 적합한 탐색 패턴을 도출하고, 탐색 패턴을 적응적으로 변형시키면 다른 고속 블록정합 방법들보다 계산량은 감소시키고, 움직임 예측 성능은 향상시킬 수 있다.

## 2. 블록정합 방법의 탐색 패턴

고속 블록정합 방법에서 탐색 패턴이란 블록정합을 위해 각 탐색 단계에서 정합 기준값을 검사하는 위치를 의미하며, 이 값들 중에서 최소값을 가지는 위치를 중심으로 다음 단계의 움직임 탐색이 수

행된다. 따라서, 고속 블록정합 움직임 탐색방법에서 사용되는 탐색 패턴의 모양과 크기는 탐색의 속도와 성능을 좌우하는 중요한 요소이다.

그림 1에서는 고전적인 고속 블록정합 방법인 3단계 탐색법(TSS)<sup>1</sup>과 2차원 로그 탐색법(TDL)<sup>2</sup>을 보여주고 있다.



(a) 3단계 탐색법 (b) 2차원 로그 탐색법

그림 1. 고속 블록정합 방법의 탐색 패턴

3단계 탐색법의 1단계에서는 9개의 탐색 위치에 현재 화면의 블록과 이전 화면에 있는 탐색 영역의 블록간의 정합 기준 오차값을 계산한다. 2단계에서는 1단계에서 구한 최소의 정합 기준 오차값을 가지는 블록의 위치를 중심으로 그림 1(a)와 같은 사각형 형태의 탐색 패턴이 위치한 탐색점들에 대하여 탐색을 실시하는데, 탐색점들 사이의 간격이 이전 단계보다는 절반으로 줄어든 형태를 사용한다. 마지막 3단계에서도 이전 단계에서 구한 최소 정합 기준 오차값의 위치를 중심으로 구성된 탐색점들에서 동일한 검사를 실시하는데, 이때 탐색점 사이의 간격은 한 화소 간격이다. 여기서 구한 최소 정합 기준 오차값을 가지는 위치가 최종적으로 해당 블록에 대한 움직임 벡터로 결정된다.

2차원 로그 탐색법은 그림 1(b)와 같이 각 단계에서 5개의 탐색 후보점에 대한 정합 기준 오차값을 검사한다. 이 값들 중 최소값을 가진 위치는 다음 단계에서 탐색의 중심 위치가 된다. 3단계 탐색법과 달리, 탐색점들 사이의 거리는 다음 단계에서 감소하지 않는다. 최소값을 가진 위치가 현재 탐색 패턴의 중심점에서 발견되거나 탐색 영역의 경계에 이르면 탐색점 사이의 간격을 줄여서 한 화소 간격으로 떨어진 주변의 탐색점들에 대하여 정합 기준 오차값을 계산하고, 최소값을 가지는 위치가 최종적으로 움직임 벡터로 결정된다.

그림 1에서 보듯이, 3단계 탐색법의 탐색 패턴은 사각형의 형태로 분포하는 탐색점들로 이루어져 있고, 2차원 로그 탐색법에서의 탐색 패턴은 십자가 형태로 분포하는 탐색점들로 구성되어 있다. 블록정합은 탐색점을 블록의 중심으로하는 탐색 영역안의 이전 블록과 현재 화면의 블록 사이의 상관도나 정합 기준 오차값을 구하게 된다. 그림에서  $dx$ 와  $dy$ 는 탐색 패턴을 구성하는 탐색점 사이의 간격을 의미한다. 일반적으로 많이 사용되는 탐색점 사이의 간격은 표 1과 같다.

표 1. 고속 블록정합 방법에서 탐색점 사이의 간격

3단계 탐색법	1단계	2단계	3단계(최종)
	4 화소간격	2 화소간격	1 화소간격
2차원로그 탐색법	각 단계		최종단계
	2 화소간격		1 화소간격

블록정합을 위한 블록의 크기를  $N \times N$ 으로 설정하고, 현재화면의  $(k, l)$ 위치에 있는 화소의 밝기값을  $I_i(k, l)$ 라 하고 이전 화면의 해당 위치에 있는 화소의 밝기값을  $I_{i-1}(k, l)$ 라고 하자. 블록정합 방법에서 블록정합의 정도를 평가하기 위해 사용되는 대표적인 정합 기준은 다음과 같다.

(a) 상관도(Cross-Correlation Function, CCF)

$$CCF(i, j) = \frac{\sum_{k=1}^N \sum_{l=1}^N I_i(k, l) I_{i-1}(k+i, l+j)}{\left[ \sum_{k=1}^N \sum_{l=1}^N I_i^2(k, l) \right]^{1/2} \left[ \sum_{k=1}^N \sum_{l=1}^N I_{i-1}^2(k+i, l+j) \right]^{1/2}} \quad (1)$$

(b) 평균 제곱 오차(Mean Squared Error, MSE)

$$MSE(i, j) = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N [I_i(k, l) - I_{i-1}(k+i, l+j)]^2 \quad (2)$$

(c) 평균 절대값 오차(Mean Absolute Difference, MAD)

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k+i, l+j)| \quad (3)$$

(d) 정합 화소수(Matching Pel Count, MPC)

$$T(k, l, i, j) = \begin{cases} 1, & \text{if } |I_i(k, l) - I_{i-1}(k+i, l+j)| \leq THS \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$MPC(i, j) = \sum_{k=1}^N \sum_{l=1}^N T(k, l, i, j) \quad (5)$$

이 값들 중에서 상관도는 최대값을 가지는 위치를 움직임 벡터로 결정하고, 나머지 정합 기준들은 최소값을 가지는 위치를 움직임 벡터로서 결정한다. 본 논문에서는 개념적으로 이해하기 쉽고 수식전개에 편리한 상관도(CCF)를 사용하였다.

### 3. 최적의 탐색 패턴

최적의 탐색 패턴을 구하는 목적은 블록 정합을 수행할 때 어떤 형태의 탐색 패턴을 취하는 것이 요구되는 계산량과 정합 기준의 관점에서 유리한지를 예측하기 위한 것이다.<sup>10-11</sup>

블록정합 방법을 간략화시켜 수학적으로 분석하기 위해 다음과 같이 가정한다.

화소값이 0인 배경위에서 화소값이 1인 사각형 물체가 이동한다고 가정한다. 따라서, 상관도 식(1)의 분자를 계산하면 이전 화면의 물체와 현재 화면

의 물체가 겹친 부분은 개별 화소 위치에서 1값을 가지지만 겹치지 않은 부분은 0값을 가지게 된다.

그림 2에서 겹침 영역의 면적은 상관도를 의미하며, 현재 블록과 탐색 영역내의 이전 블록과 겹침이 많을수록 상관도가 크다는 것을 보여준다. 지금 구하려는 것은 움직임 방향 자체가 아니라, 움직임 방향을 찾기 위해 매 단계의 탐색을 수행할 때 어떤 탐색 패턴을 사용하는 것이 최소 개수의 효과적인 탐색 후보점을 검사하여 보다 정확도가 높은 움직임방향 후보를 찾아내는 데 있다.

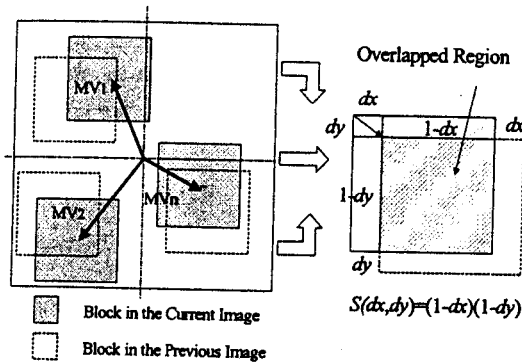


그림 2. 탐색 패턴에 대한 문제 설정

그림 2에서 고려되고 있는 블록은 모든 방향으로 움직일 수 있다. \$MV\_1\$에서 \$MV\_2\$으로 표현된 움직임 벡터는 어느 방향으로든 움직일 수 있다는 것을 표현하고 있다. 즉, 한 블록에 대한 움직임 벡터를 알지 못하는 상태에서 제안된 접근방법은 시작된다. 그림에서 겹침 영역(overlapped region)은 임의의 MV에 있을 때 완전 겹침이 있고, 그 점에서 현재 블록과 이전 블록이 \$(dx, dy)\$만큼 위치를 이동했을 때 상관도는 감소한다. 즉, \$(dx, dy) = (0, 0)\$일 때 최대 상관도를 가지고 현재 블록과 겹침 위치에서 이전 블록이 사방으로 \$(dx, dy)\$만큼 움직일 때, 동일한 상관도를 가지는 등고선들을 구할 수 있다. 이론적으로 동일한 상관도를 가지는 탐색점 후보 중에서 움직임 벡터를 예상하는 것이 의미가 있기 때문에 이 동일상관도(equi-correlation) 곡선이 중요성을 갖는다.

현재 블록의 위치를 중심으로 놓여 있는 이전 화면의 탐색 영역에서 블록의 겹침 영역은 상관도를 나타내며, 정규화된 면적은 다음과 같다.

$$S(dx, dy) = (1 - |dx|)(1 - |dy|) \quad (6)$$

$$0 \leq |dx| \leq 1, 0 \leq |dy| \leq 1$$

그림 2로부터 \$S(dx, dy)\$는 다음과 같은 대칭성(symmetry)을 가진다.

$$S(dx, dy) = S(-dx, dy) = S(dx, -dy) = S(-dx, -dy) \quad (7)$$

식 (6)로부터 \$(dx, dy)\$를 구할 수 있는데, 이것은

동일한 상관도를 가진 등고선을 의미한다.

$$|dx \cdot dy| - |dx| - |dy| + 1 - S(dx, dy) = 0 \quad (8)$$

식 (8)을 풀기 위해 \$dx\$와 \$dy\$의 관계를 여러 가지 고려할 수 있으나, 다음과 같은 선형 관계를 사용할 수 있다.

$$dy = c \cdot dx, \quad c \in \mathbb{R} \quad (9)$$

이 관계식은 블록 내에 있는 모든 위치를 쉽게 표현할 수 있기 때문에 선택하였다.

식 (9)를 식 (8)에 대입하고 정리한 후, 식 (6)과 식 (7)의 조건을 고려하면 다음과 같은 \$(dx, dy)\$의 관계식을 구할 수 있다.

$$dx = \frac{(c+1) - \sqrt{(c-1)^2 + 4c \cdot S(dx, dy)}}{2c} \quad (10)$$

$$dy = \frac{(c+1) - \sqrt{(c-1)^2 + 4c \cdot S(dx, dy)}}{2} \quad (11)$$

블록의 경계 부분에서의 특성을 알아보기 위해, (10)식에 대해 조건 (6)과 (7)을 적용하면 다음과 같은 경우에 대해 경계 부분의 값을 구할 수 있다.

경우 1: \$S(dx, dy) = 0\$, 즉, 겹침이 없는 경우

(a)

$$dx = \frac{(f+1) + \sqrt{(f-1)^2}}{2f}$$

$$= \begin{cases} \frac{f+1+f-1}{2f} = 1, & f > 1, dy = f \cdot dx = f \\ \frac{f+1+1-f}{2f} = \frac{1}{f}, & f < 1, dy = f \cdot dx = 1 \end{cases} \quad (12)$$

(b)

$$dx = \frac{(f+1) - \sqrt{(f-1)^2}}{2f}$$

$$= \begin{cases} \frac{f+1-f+1}{2f} = \frac{1}{f}, & f > 1, dy = f \cdot dx = 1 \\ \frac{f+1-1+f}{2f} = 1, & f < 1, dy = f \cdot dx = f \end{cases} \quad (13)$$

경우 2: \$S(dx, dy) = 1\$, 즉, 완전 겹침이 있는 경우

(a)

$$dx = \frac{(f+1) + \sqrt{(f+1)^2}}{2f} = 1 + \frac{1}{f} \quad (14)$$

$$dy = f \cdot dx = f + 1$$

(b)

$$dx = \frac{(f+1) - \sqrt{(f+1)^2}}{2f} = 0 \quad (15)$$

$$dy = f \cdot dx = 0$$

앞에서 구한 결과 중 식 (12)와 식 (14)만이 주어진 조건을 만족한다. 또한 그 결과로부터 접침이 없는 경우에는 경계선 상에서만 값이 존재하기 때문에 블록의 모양과 같은 사각형의 형태로 수렴한다는 것을 알 수 있다.

4. 실험 결과 및 분석

이론적으로 구한 식의 신뢰도를 검사하기 위해 실제 영상의 움직임 패턴을 구하여 비교하였다. 블록정합을 위해 16x16 크기의 블록을 사용하고, 탐색 영역의 크기는 7로 잡았다. 즉,  $0 < |dx|, |dy| < 7/16 \approx 1/2$ 로 설정하였다. 식 (5)와 식 (6)에서  $S(dx, dy)$ 가 1/2보다 작은 경우에 구한 탐색 패턴을 그림 3에 보였다.

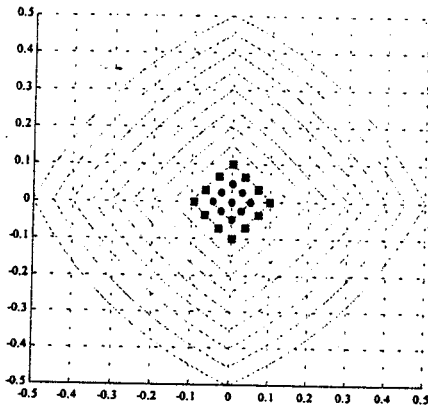
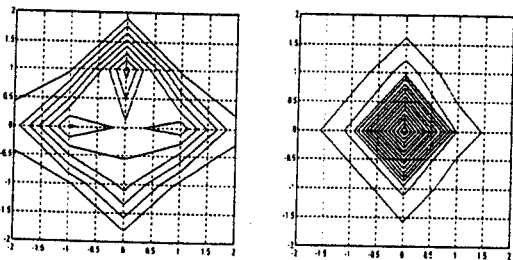


그림 3. 이론적인 적합한 탐색 패턴 (7/16 = 0.5)

실제 동영상으로부터 구한 움직임 벡터의 분포(탐색 패턴)를 살펴보면, 그림 4에 보인 것과 같이, 대부분 중심 부근에 집중되어 있다. 특히, 수평과 수직선상으로 많은 움직임이 있음을 알 수 있다. 실제 탐색 패턴과 이론적인 탐색 패턴은 다이아몬드와 비슷한 형태를 띠는 등고선임을 알 수 있다.

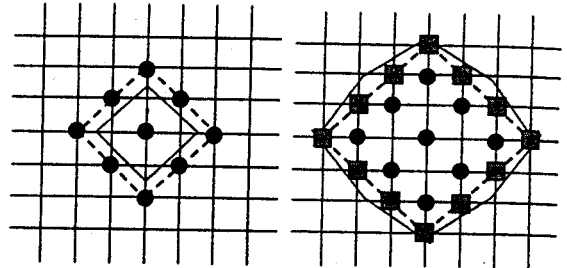


(a) 실제 탐색 패턴 (MISS AMERICA) (b) 실제 탐색 패턴 (SALESMAN)

그림 4. 실제 계산된 탐색 패턴

이론적으로 구한 그림 3의 탐색 패턴으로부터 실

제로 사용되는 여러 가지 탐색 패턴을 구할 수 있다. 그림 5와 같이 사용되는 다이아몬드 패턴은 연속적인 다이아몬드 형태를 실제 이산적인 격자 위치에 맞춰서 가장 가까운 격자 위치들을 선택한다. 그림 5에서 실선으로 표현된 것이 연속적인 다이아몬드 패턴이고, 점선으로 표시된 것이 이산적인 다이아몬드 패턴이다.



(a) 작은 다이아몬드 (b) 큰 다이아몬드

그림 5. 실제 사용할 수 있는 탐색 패턴

그림 5에서는 작은 다이아몬드와 큰 다이아몬드를 구성하는 방법을 보여주고 있다. 즉, 다이아몬드는 여러 가지 크기를 가질 수도 있고, 한 다이아몬드 내에서도 샘플된 위치만을 선택할 수도 있는데, 그 방법은 어떠한 움직임 탐색 방법이 적용되느냐에 따라서 결정될 수 있다. 즉, 사용하는 움직임 탐색 방법과 탐색 패턴 사이에는 여러 가지의 상보점(trade-off)이 존재하게 되며, 적절한 선택이 이루어졌을 때, 움직임 탐색 성능은 향상된다.

실제 탐색 패턴의 결과인 그림 4로부터 대부분의 움직임이 2 화소 간격의 다이아몬드 안에 들어감을 알 수 있다. 그렇지만 필요에 따라서 화소 간격을 증가시켜서 구성된 다이아몬드 형태에 대해 요구되는 정밀도의 탐색 패턴을 추출할 수 있다. 탐색 패턴은 사용하는 응용과 영상 시퀀스의 움직임 특성에 따라 적절한 형태로 만들 수 있다.

5. 기존 방법과의 비교

유도된 탐색 패턴인 다이아몬드 형태는 3단계 탐색법의 사각형 탐색 패턴과 2차원 로그 탐색법의 십자가형 탐색 패턴의 결합된 형태라고도 볼 수 있다. 최근에 개발된 고속블록정합 움직임탐색 방법 중 제안한 것과 비슷한 탐색 패턴을 취하는 것으로는 다음과 같이 크게 2가지를 들 수 있다.

우수한 성능을 가지는 Shan Zhu의 다이아몬드 탐색법(Diamond Search)<sup>7,9</sup>과 Tourapis의 지역 탐색법(Zonal Search)<sup>8,9</sup>은 MPEG-2 표준에 제안되었으며, 이들은 최종적으로는 Informative로 채택되었다.

다이아몬드 탐색법에서는 그림 6과 같이 사각형의 격자가 있을 때 여기에 그려진 원형 영역 안에 있는 샘플된 탐색점의 위치를 살펴보고 탐색 패턴을 결정하였다. 즉, 거리의 관점으로 볼 때, 동일

한 거리에 있는 탐색점을 가지고 탐색 패턴을 구성하였다. 그리고, 이를 바탕으로 그림 7과 같이 움직임의 크기에 따라 두 가지 종류의 탐색 패턴을 정의하였다. 이 방법에서는 매 단계마다 추가되는 탐색점의 개수를 절약하는데 주된 관점을 두었다.

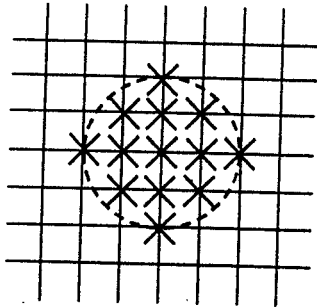
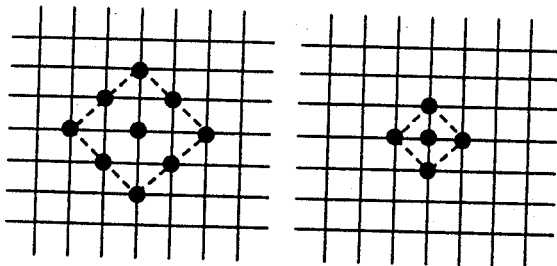


그림 6. 다이아몬드 탐색법의 탐색 패턴 추출 방법



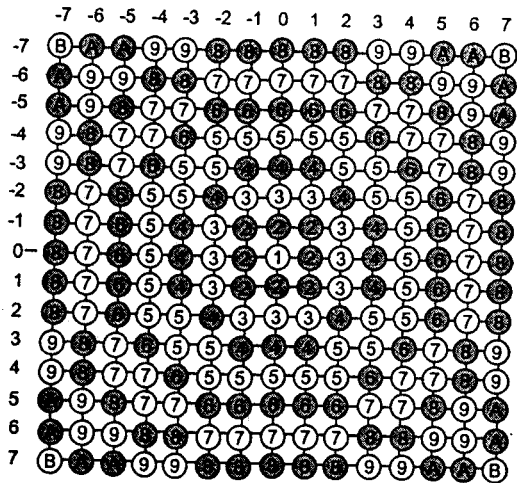
(a) 큰 다이아몬드 패턴 (b)작은 다이아몬드 패턴

그림 7. 다이아몬드 탐색법의 두 가지 탐색 패턴

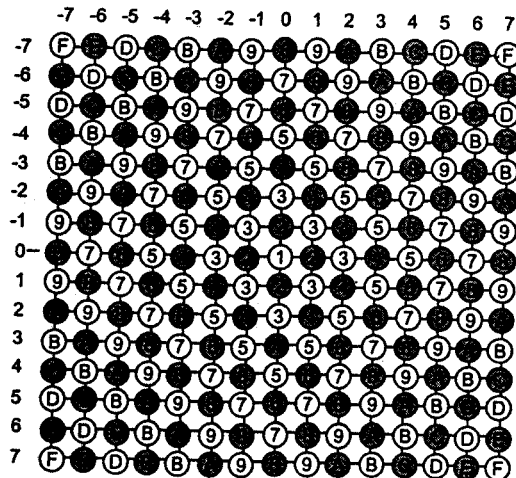
지역 탐색법은 그림 8 과 같이 두 가지 종류의 탐색 패턴을 사용하여 탐색을 수행하며, 각각 원형 지역 탐색법(circular zonal search)과 다이아몬드형 지역 탐색법(diamond zonal search)이라고 불린다. 이 방법은 앞에서 설명한 다이아몬드 탐색법과 거의 동일하다. 다만 매 탐색 단계마다 검사되는 탐색점의 개수를 어떻게 할당할 것인가의 방법의 차이가 있다. 다이아몬드 탐색법은 매 단계마다 적은 수의 탐색점을 사용하려고 하는 반면에, 지역 탐색법은 매 단계에서 다이아몬드 탐색점보다 많은 탐색점을 할당하여 정밀도를 높인다. 이 두 방법은 정밀도와 계산량의 관점에서 각각 상보점(trade-off)을 가지기 때문에 어느 쪽이 반드시 우세하다고 말할 수는 없다.

사실 이 두 방법은 제안한 방법으로 유도된 탐색 패턴의 부분집합임을 알 수 있다. 앞에서 언급된 3단계 탐색법과 2차원 로그 탐색법 뿐만 아니라, 일반적인 블록정합 방법에서 사용하는 탐색 패턴은 유도된 탐색 패턴의 일부분으로 생각할 수 있다.

그림 3에서는 탐색범위를 [-7, 7]로 제한한 경우, 즉  $7/16 \approx 0.5$  인 경우를 보였는데, 탐색범위를 [-15, 15]로 확대한 경우, 즉  $15/16 \approx 1$ 인 경우에 대해 적합한 탐색 패턴의 모양은 그림 9와 같다.



(a) 원형 지역 탐색법



(b) 다이아몬드형 지역 탐색법

그림 8. 지역 탐색법에서 두 가지 탐색 패턴

그림 9의 탐색 패턴은 사각형의 블록을 사용하여 블록정합을 한 경우, 다이아몬드 모양에서 원형으로 변하다가, 결국 사각형 형태로 수렴한다. 이것은 작은 움직임을 가지는 영상에 대해서 다이아몬드 형태의 탐색 패턴이 유리하며, 움직임이 커지면 원 또는 사각형의 형태로 탐색 패턴을 바꾸어 움직임을 탐색하는 것이 유리함을 의미한다. 즉, 초기에는 움직임이 여러 방향으로 나타나지 않고, 수평과 수직 방향으로만 움직임이 많이 분포됨을 알 수 있다. 그런데, 움직임이 커지면 모든 방향으로의 움직임이 나타나게 된다. 원의 형태에서 사각형으로 근접하는 이유는 사각형의 블록을 사용하였기 때문이다. 만약 그렇지 않다면, 원의 형태로 계속 퍼져나갈 것이다. 즉, 최종적으로는 움직임이 거리적으로나 상관도의 관점에서나 균일하게 분포되는 특성을 가지게 될 것이다.

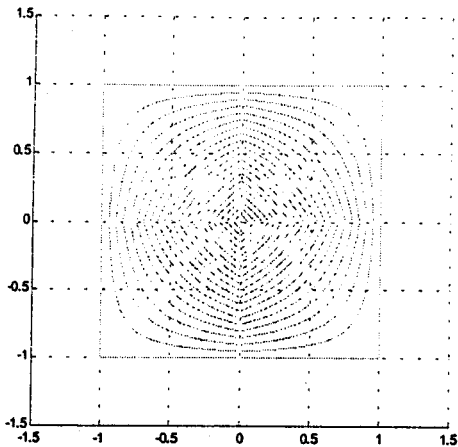


그림 9. 적합한 탐색 패턴(탐색 영역  $15/16 = 1$ )

일반적인 고속 블록정합 방법에서 사용하는 탐색 패턴은 탐색 패턴의 중심에서 동일한 거리 안에 있는 탐색점들을 가진 경우를 고려하였지만, 본 논문에서 사용한 탐색 패턴의 정의 방법은 동일한 상관도를 가지는 위치를 탐색점으로 잡았다.

실제로 블록정합을 수행할 때, 블록 사이의 상관도나 정합기준 값을 계산하고 그 중에서 탐색 오차가 작은 움직임 벡터 후보를 찾는 것이 우선이다. 통계적으로 움직임이 탐색 중심에 많이 분포하기 때문에 탐색 영역의 중심으로부터 가까운 거리에 있는 탐색점을 우선적으로 고려하기 쉽다. 물론, 상관도나 정합기준 값이 비슷한 경우에는 탐색 영역 중심으로부터 거리가 가까운 탐색점을 최종 움직임 벡터로 결정하지만, 우선순위는 정합기준 값이 되어야 한다. 즉, 정합기준 값이 비슷하다고 예상되는 탐색점 위치들 중에서 움직임 벡터의 후보를 결정하는 것이 보다 더 타당하다. 상관도의 관점으로 볼 때, 사각형 블록을 사용하는 블록정합의 경우, 탐색점들의 탐색 영역 중심으로부터 거리와 상관도의 관계는 일치하지 않는다. 이 결과는 블록정합을 단순화시켜 분석한 것이지만, 실제의 경우를 적용하더라도 거리적으로 동일한 위치에 있을 때, 상관도가 같다고 확신할 수 없다.

## 6. 결론

본 논문에서는 시간적으로 움직이는 블록의 상관도를 정합기준 값으로 사용하여 움직임 탐색에 적합한 형태의 탐색 패턴을 유도하였고, 이를 실제 영상에서 구한 탐색 패턴과 비교하였다. 두 탐색 패턴은 모두 다이아몬드 형태를 가지고 있었다. 또한 경계조건을 적용하여 구한 탐색 패턴은 다이아몬드 형에서 원형으로 변하다가 최종적으로는 사용하는 블록과 같은 모양으로 수렴함을 알 수 있었다. 대부분의 고속 탐색 방법에서 사용되는 탐색 패턴들은 제한한 방법을 사용하여 이론적으로 구한 탐색 패

턴의 부분집합이 된다는 것을 확인할 수 있었다. 따라서, 사용하려는 목적과 블록의 모양에 따라 적합한 탐색 패턴을 유도할 수 있고, 이를 사용하여 움직임 탐색의 성능을 개선할 수 있다.

## 감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트워크 연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

## 7. 참고 문헌

- [1] 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.
- [2] R. Srinivasan and K.R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COM-33, no. 8, pp. 888-896, Aug. 1985.
- [3] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-38, pp. 950-953, July 1990.
- [4] H.G. Musmann, P. Pirsh and H.J. Grallert, "Advances in Picture Coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523-548, Apr. 1985.
- [5] L.M. Po and W.C. Ma, "A Novel Four-step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 6, no. 3, pp. 313-317, June 1996.
- [6] 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. Circuit and Systems for Video Technology*, vol. 3, no. 1, Feb. 1993.
- [7] S. Zhu and K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Processing*, vol. 92, pp. 287-290, Feb. 2000.
- [8] A. Tourapis, O. Au, M. Liou and G. Shen, "An advanced zonal block based algorithm for motion estimation," *Internat'l Conf. on Image Processing*, vol. 2, pp. 610-614, Oct. 1999.
- [9] T. Chiang and H. Sun, Report of Ad Hoc Group on Encoder Optimization, ISO/IEC/JTC1/SC29/WG11 MPEG99/M5528, Dec. 1999.
- [10] D.K. Lim and Y.S. Ho, "A Fast Block Matching Motion Estimation Algorithm based on Statistical Properties of Object Displacement," *IEEE Region 10 Annual Conference*, pp. 138-141, Dec. 1998.
- [11] D.K. Lim and Y.S. Ho, "A Fast Block Matching Motion Estimation Algorithm using Optimal Search Patterns," to appear in *SPIE VCIP Proc.*, Jan. 2001.