

Genetic NTSS 기법을 이용한 움직임 추정

(Motion Estimation using Genetic NTSS Method)

박지영[†] 백순화^{**} 전병민^{***}

(Jee-Young Park)(Soon-Hwa Baek)(Byoung-Min Jun)

요약 기존의 블록 정합 알고리즘인 FS(Full Search)알고리즘은 정확한 움직임 벡터를 구할 수 있으나 요구되는 계산량이 많다. 반면에 국부 탐색을 하는 고속 블록 정합 알고리즘은 FS보다 빠른 탐색을 할 수 있으나 FS 보다 정합 오차가 크다.

본 연구는 전역탐색을 하는 유전자 알고리즘에 빠른 탐색을 하는 블록 정합 알고리즘인 NTSS(New Three Step Search)알고리즘을 제안한다. 제한한 방법에서 각 탐색체는 움직임 벡터를 표현하며 초기 탐색체는 탐색 공간의 중심 탐색점 가까이에 고정적으로 발생시키고 각 탐색체는 MSE(Mean Square Error)값으로 평가된다. 평가된 탐색체 중 작은 MSE값을 가지는 탐색체가 NTSS의 탐색점 수만큼 다음 세대의 탐색점으로 선택된다. 선택된 탐색체는 세대를 거치면서 돌연변이 연산과 교배연산이 행해지고 이때 돌연변이 연산의 크기는 NTSS의 탐색 단계 크기가 된다. 제한한 세대 수 만큼 반복 후 최소의 MSE 값을 가지는 유전자가 해당 블록의 움직임 벡터가 된다.

시뮬레이션 결과 제안한 방법은 가장 우수한 성능을 가지는 FS와 유사한 MSE 값을 얻을 수 있었고 동시에 FS에서 요구되는 계산량에 비해 많은 계산량을 줄일 수 있었다.

Abstract Block-based FS(Full Search) algorithm for motion estimation can find optimal motion vector. However, a heavy computational load is demanded. On the other hand, fast BMAs(Block Matching Algorithm) in local search algorithm take shorter search time than FS to compute motion vector. But the MSE(Mean Square Error) of these fast BMAs is poorer than that of FS.

In this paper, GA(Genetic Algorithm) using NTSS(New Three Step Search) is proposed for global and fast search of motion. The GA is algorithm for searching global maxima and minima and the NTSS is algorithm for fast search. In proposed method, each chromosome represents a motion vector. Initial chromosomes are selected from fixed locations near the central point of the search space. Each chromosome is evaluated by MSE. The chromosomes with smaller MSE are selected as the search points of NTSS in the next generation. And selected chromosome is performed crossover operation and mutation operation. The mutation step size is the same as the search step size of NTSS. After proposed generation, the chromosome with the minimum MSE is the motion vector of the block.

The simulation result shows the proposed method has similar performance to FS in term of MSE. In addition, computational complexity required by the proposed method is lower than computational load of FS.

1. 서론

영상 부호화의 목적은 적은 양의 정보로 원 영상을 충실히 표현하고 재생하는데 있다. 일반적으로 동영상의 특징은 프레임내(intraframe)의 중복되는 정보와 프레임간(interframe)의 중복되는 정보가 많다는 것인데 이러한 중복성을 줄이는 것이 동영상 압축의 기본 원리이다. 시간에 따라 변화하는 동영상은 시간 상관계수가 공간 상관 계수보다 훨씬 크기 때문에 공간 상관성을 이용하는 프레임내의 부호화보다 시간 상관성을 이용하는 프

[†] 비 회 원 : 삼성통신 CSF 연구원
noela@kebi.com

^{**} 정 회 원 : 천안외국어대학 컴퓨터정보과 교수
back@mail.chonan-c.ac.kr

^{***} 정 회 원 : 충북대학교 컴퓨터공학과 교수
gmjun@cbucc.chungbuk.ac.kr

논문접수 : 2000년 1월 13일
심사완료 : 2000년 8월 23일

레이미간의 부호화가 효율적이다. 프레임간 중복성을 제거하는 방법으로는 움직임 추정기가 있다. 움직임 추정[1] 물체의 움직임 변화를 추정하는 방법으로 연속적인 영상 신호에서 현재 프레임(frame)의 화소(pixel)들이 이전 프레임에 비해 어느 정도 움직였는지를 나타내는 움직임 벡터를 추정하여 전체 영상의 전송 대신 움직임 벡터를 전송함으로써 전송 데이터 량을 줄이는 방법이다.

움직임 추정에 사용되는 방법으로는 수행시간이 적고 하드웨어 구현이 간단한 블록 정합 알고리즘이 많이 사용된다. BMA(Block Matching Algorithm)[2][3]의 가장 일반적인 방법인 전역 탐색은 우수한 성능을 가지고 있으나 추정 블록당 많은 계산량이 소요된다. 이러한 계산량 과다를 해결하기 위해, three-step search(3SS)[4], new three-step search(NTSS)[5][6], 그리고 four-step search(4SS)[7]와 같은 다양한 방법의 고속 알고리즘이 제안되었다. 그러나 이들 방법은 탐색 영역을 줄여 가는 국부 탐색의 움직임 추정을 하므로 전역 탐색에 비해 예측 오차가 크고 움직임 벡터의 신뢰성이 떨어지는 문제점을 내포하고 있다.

본 연구에서는 기존의 전역 탐색이 가지는 계산량 문제를 해결하고 고속 알고리즘이 갖는 국부적 최적해 문제를 해결하기 위해 유전자 알고리즘에 NTSS를 적용한 움직임 추정을 제안한다.

GA(Genetic Algorithm)[8][9]는 여러 개의 탐색 집(군:population)을 동시에 탐색하고 확률적 연산자를 사용함으로써 전역적인 최적의 경로를 탐색하기 위한 방법으로 사용된다. 고속 블록 정합 알고리즘 중 NTSS는 큰 움직임과 더불어 주위의 작은 움직임도 탐색할 수 있는 장점이 있다. 이 때 GA의 탐색점 수를 NTSS의 탐색점 수로 함으로써 전역 탐색보다 적은 수의 탐색점으로 보다 빠르게 탐색할 수 있다. 동시에 전역적 탐색을 하는 GA를 사용하여 고속 알고리즘의 국부 최소화 문제[10]를 해결함으로써 보다 정확한 움직임 추정을 기대할 수 있다.

2장에서는 기존의 움직임 추정 방법인 블록 정합 알고리즘과 기본적인 유전자 알고리즘에 대해서 설명한다. 3장에서는 제안한 방법인 NTSS를 이용한 수정된 유전자 알고리즘을 설명하고 4장에서는 제안한 알고리즘의 구현과 실험 결과에 대해서, 5장에서는 결론 및 향후 연구 방향에 대해 설명한다.

2. 블록정합 알고리즘과 유전자 알고리즘

2.1 기존의 연구 블록정합 알고리즘

블록 정합 알고리즘은 움직임 벡터를 구하고자 하는 현재 프레임 블록과 동일한 위치의 탐색 영역을 이전 프레임에 설정한다. 현재 프레임의 탐색 블록과 이전 프레임의 탐색 영역 내 블록과의 유사도를 화소단위로 이동해가면서 정합 오차가 가장 작은 블록의 위치를 구한다. 이 때 구해진 블록과 현재 프레임의 탐색 블록과의 위치 차가 움직임 벡터가 된다.

정합 오차를 계산하는 함수로는 MSE(Mean Square Error)[14]나 MAE(Mean Absolute Error)[14] 등이 있다.

$$MSE(u, v) = \frac{1}{mn} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} [I(k, l, t) - I(k+u, l+v, t-\tau)]^2 \quad (1)$$

$$MAE(u, v) = \frac{1}{mn} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} |I(k, l, t) - I(k+u, l+v, t-\tau)| \quad (2)$$

블록 크기는 $m \times n$ 화소, u, v 는 움직임 벡터, k, l 은 x, y 좌표를 나타내고 t 는 시간을 나타낸다.

전역 탐색기법(FS)은 전체 탐색 영역 내에서 최소의 MSE값을 가진 블록을 찾기 때문에 블록 정합 알고리즘으로 얻을 수 있는 가장 정확한 움직임을 찾을 수 있다. 전역 탐색 방법은 한 블록 당 탐색 횟수가 프레임간의 최대 움직임을 d 화소라 했을 때 $(2d+1)^2$ 이 되며 한 프레임에서 (영상의 전체 블록갯수) $\times (2d+1)^2$ 이 되므로 상당히 많은 계산량이 요구된다. 따라서 3단계 탐색(3SS), 새로운 3단계 탐색(NTSS), 4단계 탐색(4SS)과 같은 여러 고속 블록 정합 알고리즘이 제안되었다. 인접한 프레임사이에서 최대 움직임이 7 화소이면 하나의 블록에 대해 FS는 225개, 3SS는 25개(9+8+8=25)의 탐색점을 가지며, NTSS는 33개 (17+8+8=33)의 탐색점을, 그리고 4SS는 27개 (9+5+5+8=27)의 탐색점을 갖는다.

2.2 기본적인 유전자 알고리즘

유전자 알고리즘은 70년대 Holland가 제안한 최적화 탐색 알고리즘[8]의 일종으로 다른 알고리즘과는 달리 방대하고 복잡한 함수에 대해 전역적인 최적화가 가능하다. 유전자를 갖는 가상의 생물 집단, 즉 개체군 중에서 환경에 대한 적합도가 낮은 개체는 도태시키고 높은 적합도를 갖는 개체는 증식시키는 세대교체를 수행한다. 유전자 알고리즘은 실제의 생물생식에서도 발생하는 유전자의 교차나 돌연변이 등의 조작을 수행하여 다음 세대의 개체군을 생성한다. 유전자 알고리즘의 기본적인 흐름도는 다음과 같다.

- ① 모집단내의 두 개의 염색체를 초기화한다.
- ② 모집단내에서 두 개의 염색체를 임의로 선택한다.
- ③ 두 개의 염색체를 교배하여 새로운 염색체를 생성한다.

- ④ 새로운 염색체내 인자의 값을 돌연변이[8][11] 시킨다.
- ⑤ 이러한 염색체들이 새로운 모집단을 형성한다.
- ⑥ 정지조건을 만족하면 최적의 염색체를 선택하고 그렇지 않으면 ②과정으로 돌아간다.

2.2.1 유전자 알고리즘의 구성요소[15]

유전자 알고리즘은 풀고자 하는 문제의 변수값을 이진수의 나열인 염색체(chromosome)로 표현하고 이 염색체들의 모임의 집합을 군(population)이라고 한다. 군을 초기화 한 후 유전자 알고리즘의 연산자인 선택, 교배, 돌연변이를 사용하여 현재의 군으로부터 주어진 조건에 적합도가 높은 새로운 세대를 만드는 일을 반복한다. GA는 생성된 군 가운데 가장 높은 적합도를 가지는 염색체의 값이 원하는 정도에 도달하거나 정해진 세대의 반복 횟수를 넘을 경우 종료된다.

최적화의 대상인 적합도 함수에 의해 생성되는 염색체의 함수값을 그 염색체의 적합도(fitness)라고 한다.

유전자 알고리즘에서 사용하는 기본적인 연산자로는 선택, 교배, 돌연변이가 있다. 선택은 가장 높은 적합도를 갖는 개체가 다음 세대에 살아 남을 가능성이 높다는 자연 선택 현상을 모델링한 것으로 다음과 같은 선택법들이 있다. 기본 선택의 룰렛 선택(roulette selection)[8][13]은 각 개체 s_i 의 적합도 $f(s_i)(>0)$, $i = 1, \dots, N$ 의 총합을 구해 각 개체의 선택 확률을

$$p_i = \frac{f(s_i)}{\sum_{i=1}^N f(s_i)} \quad (3)$$

와 같이 정하는 방법이다.

기대치 선택[8][13]은 적합도에 대한 각 개체의 확률적인 재생 개체수를 구하여 선택하는 방법이고 순위 선택[8][3](ranking selection)은 적합도 순위에 따라 다음 세대에 자손을 남길 확률을 결정하는 방법이다. 토너먼트 선택(tournament selection)은 개체군 중 일정수의 개체를 임의로 선택하여 최고의 적합도를 가지는 개체를 다음 세대에 남기는 방법으로 이것은 다음 세대의 개체수가 모두 결정될 때까지 반복된다. 엘리트 보존 선택(elitist preserving selection)은 교배 및 돌연변이 결과로 우수한 해가 소실되는 것을 막기 위해 가장 우수한 해를 보존하여 다음 세대에 남기는 방법이다. 일반적으로 다른 선택 방법과 융합하여 사용한다.

교배[8]연산자는 선택된 두 개체간의 염색체를 부분적으로 서로 바꿈으로써 새로운 개체를 생성하는 역할을 한다. 이는 탐색의 관점에서 보면 제한된 공간에서 탐색의 공간을 넓히는 것을 의미한다. 염색체의 길이가

l 이라면 교배가 일어날 수 있는 곳은 $0 \sim l-1$ 사이가 된다. 돌연변이는 개체의 각 유전자에 대하여 일정한 돌연변이 확률[8][9]을 적용하여 대립 유전자의 값으로 바꾸는 것이다. 염색체의 비트 하나를 0은 1로, 1은 0으로 바꾸는 것을 의미하는데, 이것을 사용하여 국부적인 최소화 상태에서부터 벗어날 수 있으며 잃어버린 유전형질을 복구하는 수단으로도 사용된다.

GA연산자를 이용하여 새로운 세대를 생성하였을 때 부모 세대보다 더 낮은 적합도를 가지는 염색체가 생성될 수 있다. 그러나 세대를 거듭하면서 적합도가 낮은 것은 염색체끼리의 경쟁에서 도태되어 사라지므로 낮은 적합도를 갖는 염색체가 다음 세대에 생기더라도 문제가 되지 않는다.

3. 제안하는 NTSS 유전자 알고리즘을 이용한 움직임 추정

NTSS알고리즘은 3단계 탐색알고리즘의 수렴된 버전으로 주위의 작은 움직임을 찾기 위해 그림 1에서와 같이 첫 단계에서 주위 8 화소를 추가적으로 탐색하게 된다.

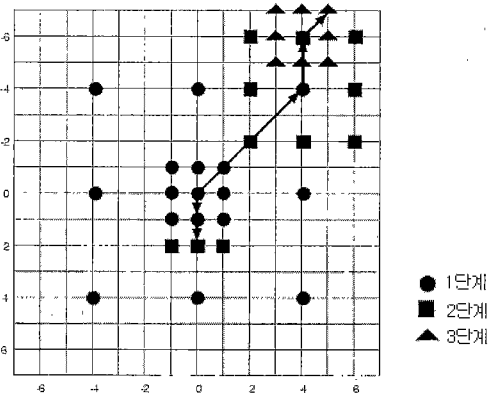


그림 1 New three-step search (NTSS)

기존의 FS는 정확한 움직임을 찾을 수 있지만 탐색 시간이 오래 걸린다는 단점이 있다. 반면에 고속 블록정합의 국부 탐색은 탐색 시간은 빠르지만 정확한 움직임 추정은 기대할 수가 없다. 유전자 알고리즘은 전역 탐색인 반면 계산량이 많은 단점을 가진다. 따라서 본 논문에서는 전역적 탐색을 하면서도 빠른 탐색을 할 수 있도록 전역탐색의 유전자 알고리즘에 빠른 탐색이 가능한 NTSS를 적용함으로써 FS보다 빠르면서 정확한 움직임을 찾고자 한다.

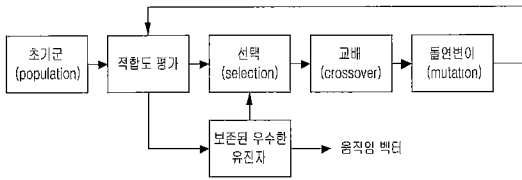


그림 2 GNTSS 움직임 탐색과정

3.1 수정된 염색체 구조

GA개체는 염색체(chromosome)로 표현되며 본 논문에서의 염색체는 움직임 벡터가 된다.

$$C_i = \{ X_i, Y_i \} \tag{4}$$

여기서 C_i 는 염색체이고 X_i 는 x축 움직임 벡터, Y_i 는 y축 움직임 벡터를 나타낸다.

움직임 벡터는 기본적인 유전자 알고리즘의 이진수 배열이 아닌 십진수로 표현되며 염색체의 크기는 최대 블록 범위가 되고 염색체 군(population size)은 NTSS의 탐색점 수가 된다.

3.2 염색체의 초기 군 생성

탐색의 초기 시점에서는 어느 염색체가 적합한지를 알 수 없기 때문에 난수 발생으로 초기 염색체 군을 생성한다. 그러나 본 논문에서는 연속된 영상의 움직임이 중심에 높게 분포한다는 사실을 반영하여 초기의 염색체군을 중심점 부근으로 고정시킴으로써 높은 적합도 값을 가지는 염색체를 생성하고자 한다.

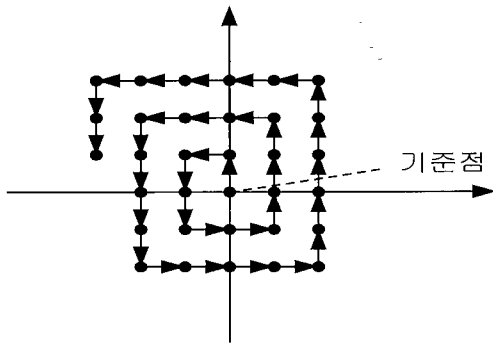


그림 3 고정된 초기 군(33개)생성

다음 식에 의해 x, y의 초기 염색체군을 고정시킨다.

$$\hat{x}_i = (-1)^i \left([i - K(l+1)] \cdot [1 - (\lfloor 2\sqrt{i} \rfloor \bmod 2)] + \lfloor \frac{1}{2} l \rfloor \right) \tag{5}$$

$$\hat{y}_i = (-1)^i \left([i - K(l+1)] \cdot [\lfloor 2\sqrt{i} \rfloor \bmod 2] - \lfloor \frac{1}{2} l \rfloor \right) \tag{6}$$

$$(l = \lfloor \sqrt{i} \rfloor) \tag{7}$$

$$i = 0, 1, 2, \dots, \text{population size} - 1$$

3.3 적합도 평가

본 논문에서는 적합도 함수로 MSE를 사용하고 이때 MSE값이 작을수록 적합도가 높은 보다 정확한 움직임 벡터로 평가된다.

MSE는 비교되는 블록간의 화소값의 평균 차를 구하는 것으로 그 값이 작을수록 블록은 서로 유사성이 많은 것으로 평가되고 이를 이동된 블록으로 본다. 따라서 염색체, 즉 움직임 벡터를 MSE를 계산하는 식 (1)에 대입했을 때 MSE값을 가장 작게 하는 염색체를 진정한 움직임 벡터로 평가한다.

처음 단계에서 고정된 초기군의 염색체를 식 (1)에 움직임 벡터(u, v)로 대입하여 최소의 MSE값을 가지게 하는 염색체를 선택한다. 또 유전 연산자를 적용한 후 최소의 MSE값을 가지게 하는 염색체를 찾는다. 이와 같이 MSE는 각 단계에서 행해지며 최적의 움직임 벡터를 찾는 평가함수가 된다.

3.4 재생을 위한 염색체 선택

본 논문에서는 평가된 적합도 값에 기초한 새로운 개체 집단을 선택하는 과정으로 기본 선택법인 룰렛(roulette wheel)선택 방법과 엘리트 보존 선택법[8]을 사용한다.

룰렛 선택법은 염색체 각각의 적합도 값을 계산하고 전체 염색체 적합도에 대한 자신의 염색체가 가지는 선택확률(p_i)을 계산한다. 그리고 각 염색체(V_i)는 선택확률의 누적 확률(q_i)을 구한다. 0 과 1사이의 임의의 수(r)를 발생하여 $q_{i-1} < r \leq q_i$ 이면 i 번째 염색체 V_i ($1 \leq i \leq \text{popsize} - 1$)를 선택한다.

엘리트 보존 선택법은 한 세대를 거친 최고의 염색체를 보존하여 다음 세대의 우수한 염색체들과 비교 평가하는 방법이다. 본 논문에서는 이와 같은 엘리트 보존 선택법을 사용함으로써 최고의 염색체가 세대를 거치면서 돌연변이, 교배와 같은 유전 연산자에 의해 소멸되는 것을 막고자 하였다.

3.5 염색체의 교배 연산

수정된 유전자 알고리즘을 사용하는 진화 전략[8]은 초기에 돌연변이 연산자만을 사용하였으며 교배 연산자는 그 후에 도입되었다. 본 논문에서는 산술 교배[8]를 사용하여 초기 진화 전략보다 더 나은 최적의 해를 구하고자 한다.

s_v^t 와 s_w^t 가 교배될 때 그 다음세대(s_v^{t+1} , s_w^{t+1})는

$$s_v^{t+1} = a \cdot s_w^t + (1-a) \cdot s_v^t \tag{8}$$

$$s_w^{t+1} = a \cdot s_v^t + (1-a) \cdot s_w^t \tag{9}$$

가 된다. 이 때 a 는 교배율을, s_v^t, s_w^t 는 교배될 두 벡터를 나타낸다.

따라서 본 논문에서의 산술적 교배 벡터 $s_v^t = \{V_x^t, V_y^t\}$, $s_w^t = \{W_x^t, W_y^t\}$ 는 다음과 같이 적용된다. V_x, W_x 는 x 축 움직임 벡터를, V_y, W_y 는 y 축 움직임 벡터를 나타낸다.

$$V_x^{t+1} = a \cdot W_x^t + (1-a) \cdot V_x^t \quad (10)$$

$$V_y^{t+1} = a \cdot W_y^t + (1-a) \cdot V_y^t \quad (11)$$

$$W_x^{t+1} = a \cdot V_x^t + (1-a) \cdot W_x^t \quad (12)$$

$$W_y^{t+1} = a \cdot V_y^t + (1-a) \cdot W_y^t \quad (13)$$

3.6 염색체의 돌연변이 연산

돌연변이 연산은 NTSS의 탐색 기법을 이용한다. NTSS의 탐색 거리는 각 세대에서의 돌연변이 단계 크기(step size)로 한다. 총 제안한 세대수는 4세대로 초기 탐색 단계는 1세대와 2세대가 되고 두 번째 탐색 단계는 3세대, 세 번째 탐색 단계는 4세대가 된다.

1세대는 추가적으로 탐색되는 주위 8개 탐색점을 가지며 탐색 거리는 1이 되고 2세대의 탐색 거리는 4, 3세대의 탐색거리는 2, 4세대의 탐색거리는 1이 된다.

1세대는 최고의 적합도 값을 가진 염색체를 별도로 저장해 둔다. 저장된 1세대 염색체는 돌연변이 연산을 한 번만 행하며 나머지 2, 3, 4세대는 세대를 거치면서 돌연변이 연산이 반복적으로 행해진다. 1세대의 최고 적합도 값을 가진 염색체는 2, 3, 4세대를 거친 최고의 염색체와 비교되어 적합도 값이 더 높은 염색체를 최종 움직임 벡터로 결정한다.

돌연변이 연산자[11]는 다음의 서로 다른 9개 연산자 $i=0$ 에서 $i=8$ 까지 수를 발생시켜 새로 선택된 집단에 각 해당 연산자를 적용한다.

$$X_{mutate} = (-1)^m \left([m - \lfloor (l+1) \cdot [1 - (\lfloor 2\sqrt{m} \rfloor \bmod 2)] + \lfloor \frac{1}{2} l \rfloor] \right) \quad (14)$$

$$Y_{mutate} = (-1)^m \left([m - \lfloor (l+1) \cdot [1 - (\lfloor 2\sqrt{m} \rfloor \bmod 2)] - \lfloor \frac{1}{2} l \rfloor] \right) \quad (15)$$

$$m = i \bmod 9 \quad (16)$$

$$l = \lfloor \sqrt{m} \rfloor \quad i = 0, 1, 2, \dots, \text{population size} - 1 \quad (17)$$

X_{new}, Y_{new} 는 돌연변이 된 새로운 염색체[12]를, X_{select}, Y_{select} 는 선택과정에서 우수한 염색체로 선택된 염색체를 나타낸다.

$$X_{new} = X_{select} + X_{mutate} \times \text{탐색거리} \quad (18)$$

$$Y_{new} = Y_{select} + Y_{mutate} \times \text{탐색거리} \quad (19)$$

4. 실험 및 결과고찰

본 논문에서 제안한 움직임 추정 방법의 성능을 비교

평가기 위해 제안한 GNTSS와 기존의 탐색 알고리즘인 3SS, 4SS 그리고 가장 우수한 성능을 가지는 FS, 유전자 알고리즘을 이용한 4GS (Four-step Genetic Search)[12]를 펜티엄 II 400MHz에서 Visual C++을 사용하여 구현하였다.

4GS는 유전자 알고리즘의 돌연변이 단계에 4SS를 적용한 방법으로 유전자 선택에 있어서 순위 선택법을 사용하였으며 유전 연산자로는 돌연변이 연산자만을 사용한 방법으로 우수한 성능을 가진다.

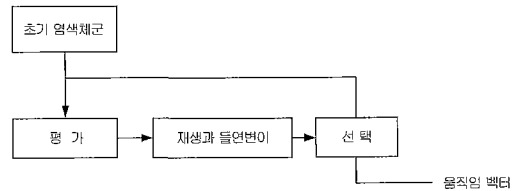


그림 4 4GS의 움직임 탐색 과정

제안한 알고리즘의 성능 평가 기준으로는 프레임간의 블록 정합 오차를 계산하는 MSE와 움직임 탐색의 소요 시간을 사용하였다.

각 프레임에서 움직임 추정을 위해 사용된 블록 크기는 16x16 화소, 블록의 최대 움직임 범위는 7화소로 하였다. 제안한 방법 GNTSS에서 실험영상 football, table tennis, salesman, garden의 교배율은 0.2, claire의 교배율은 0.3으로 하였다. 만일 교배율에 따라 심한 MSE값의 차가 생긴다면 영상의 특성에 따라 교배율을 정해야 하는 문제가 발생하게 된다. 그러나 교배율에 따른 MSE값의 변화를 실험한 결과 큰 차이의 값을 가지지 않기 때문에 이 중 가장 우수한 교배율을 경험적으로 사용하였다.

일반적으로 유전자 알고리즘을 사용한 4GS와 제안한 GNTSS는 기존의 블록 정합 알고리즘 3SS와 4SS보다 더 나은 성능을 가짐을 볼 수 있다. 특히 salesman과 claire와 같이 움직임이 적은 영상에서는 제안한

표 1 실험 동영상 정보

실험 동영상	프레임 수	프레임 크기
football	89	352x240
garden	80	352x240
table tennis	80	352x240
salesman	89	352x288
claire	80	352x288

표 2 교배율에 의한 MSE값의 변화

실험동영상	교 배 율				
	0.2	0.4	0.6	0.8	교배하지 않았을 경우 (교배율 0)
football	173.70	176.20	177.10	176.95	177.12
table tennis	126.60	127.94	130.39	129.27	126.72
claire	5.07	5.09	5.11	5.12	5.08
salesman	21.10	21.40	21.54	21.8	21.67

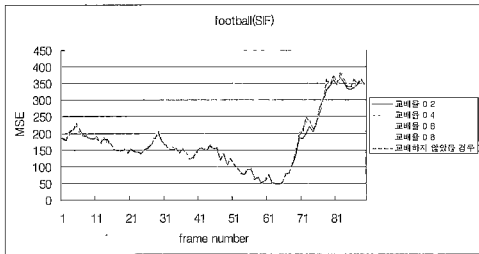


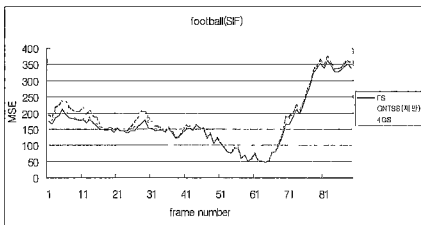
그림 5 football영상의 교배율에 의한 MSE값의 변화

표 3 기존의 움직임 추정 알고리즘과 제안한 방법의 평균 제곱 오차(MSE) 비교

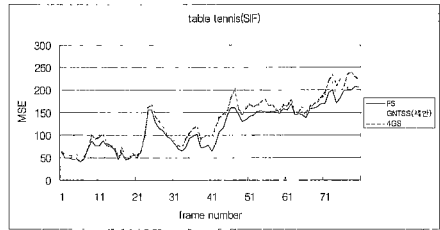
실험 동영상	기존의 움직임 추정 방법				제안한 방법
	블록 정합 알고리즘			유전자 알고리즘과 블록 정합알고리즘	유전자 알고리즘과 블록정합 알고리즘
	FS	3SS	4SS	4GS	GNTSS
football	168.63	195.31	190.96	179.67	173.70
table tennis	116.36	160.37	153.93	130.85	126.60
garden	293.32	342.17	320.54	312.74	310.89
salesman	20.78	22.45	21.79	21.25	21.10
claire	5.081	5.21	5.30	5.089	5.087

GNTSS가 FS와 매우 유사한 MSE값을 갖는 성능을 보여준다. 반면에 움직임이 많고 빠른 파노라마식의 움직임을 갖는 garden영상은 성능 평가의 기준이 되는 FS에 비교하여 MSE값이 큼을 볼 수 있다.

그림 8은 이전 프레임과 비교하여 이동된 움직임 벡터를 현재 프레임에 흰 색의 선으로 나타내었다.

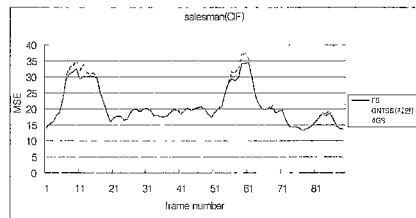


(a) football

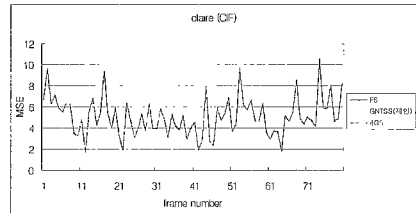


(b) table tennis

그림 6 SIF(352x240)영상에 대한 MSE 비교

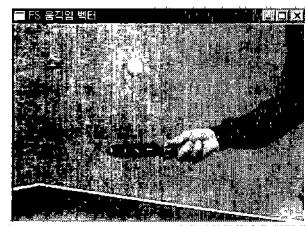


(a) salesman

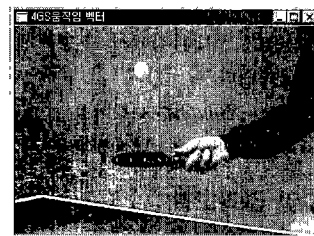


(b) claire

그림 7 CIF(352x288)영상에 대한 MSE 비교



(a) Full Search (FS)



(b) Genetic four-step search(4GS)



(c) 제안한 Genetic 이용한 New Three-Step Search (GNTSS)

그림 8 table tennis의 움직임 벡터

표 4 움직임 탐색 시간 (sec)

실험 동영상	기존의 움직임 추정 방법			제안한 방법			
	블록 정합 알고리즘			유전자 알고리즘 & 블록 정합 알고리즘		유전자 알고리즘 & 블록 정합 알고리즘	
	FS	3SS	4SS	4GS		GNTSS	
				시간	개선율 (%)	시간	개선율 (%)
football	18.388	6.096	5.812	13.381	27.23	15.795	14.10
table tennis	18.738	6.890	5.432	14.051	25.01	15.176	19.01
garden	18.143	6.976	5.332	13.427	25.99	15.899	12.37
salesman	19.757	7.099	7.001	13.842	29.94	15.197	23.08
claire	16.557	5.432	5.109	11.890	28.19	13.221	20.15

위 실험 결과에서 FS 탐색 시간 보다 4GS는 약 27%, 제안한 탐색 방법은 약 17%의 시간 절감 효과를 얻을 수 있음을 보였다. 3SS와 4SS는 빠른 탐색을 하는데 이것은 국부적 탐색에 의해 조기 수렴하는 최소화 문제에 빠지기 때문이다.

5. 결론

본 논문에서는 전역 탐색을 하면서도 빠른 탐색이 가능하도록 전역탐색의 유전자 알고리즘에 빠른 탐색의 NTSS를 적용함으로써 FS보다 빠르면서 정확한 움직임을 찾기 위한 움직임 추정 방법을 제안하였다.

제안한 방법은 움직임 벡터를 십진수의 염색체로 표현하고 염색체의 초기 개체 집단을 블록의 시작점 주위에 고정적으로 발생시킨다. MSE를 계산하여 우수한 염색체를 선택하고 선택된 염색체는 산술적 교배와 돌연변이 연산을 행한 후 적합도를 계산한 후 높은 적합도를 가지는 염색체가 움직임 벡터가 된다.

실험 결과 제안한 방법은 기존의 3SS, 4SS와 같은 블록 정합 알고리즘 보다 유전자 알고리즘을 이용한 4GS와 제안한 GNTSS가 더 우수한 성능을 가짐을 알

수 있었고 제안한 GNTSS방법은 FS와 가까운 MSE값을 가지며 특히 claire와 salesman영상과 같이 움직임이 적은 영상에 대해 거의 동일한 성능을 볼 수 있었다. 그리고 football, table tennis, garden, salesman, claire 모든 영상에서 산술적 교배와 돌연변이 연산을 행한 GNTSS가 돌연변이 연산만을 행한 4GS보다 더 적은 MSE값을 가지며, 제안한 탐색 방법은 FS의 탐색시간에 비해 약 17%의 더 빠른 효과를 얻을 수 있었다. 이와 같이 제안한 알고리즘은 전역 탐색과 유사한 움직임 추정의 성능을 가지면서 탐색 시간을 줄일 수 있었다.

유전자 알고리즘의 특징 중의 하나가 변수 성분이 많다는 점이다. 그러나 이 값들을 결정하는 일정한 방법이 없어 시행 착오적 경험에 의존하고 있다. 따라서 개체의 수, 교배율, 돌연변이율 등의 결정에 최적화기법을 도입하면 유전자 알고리즘의 결정을 해결할 수 있을 것으로 생각된다.

참고 문헌

- [1] Nobuyuki Yagi, Seiki Inoue, Masaki Hayashi, Makoto Okui and Seiichi Goushi, "C Gengo de Manabu Jissen Dijitaru Eizoushori." Ohmsha, Ltd. 1995
- [2] 조한욱, 서정욱, 정재창 "변형된 비용함수를 이용한 움직임 추정 기법" 방송공학회논문지, 제 3권 제 1호, 1998년
- [3] 정일화, 이대영 "차영상과 4진트리 구조를 이용한 가변 블록정합 알고리즘에 관한 연구" 한국통신학회 논문지 제 21권, 1996년
- [4] Yiu-Hung Fok, Oscar C. Au, "A fast block matching algorithm in feature domain," Proc. of IEEE workshop on visual signal processing and communications '93, pp.199-202, Sept 21-23 1993
- [5] J.S. Kim, R.H. Park, "Feature-based block matching algorithm using integral projections," IEEE Electron. Vol. 25. pp.29-30, Jan. 1989
- [6] R.Li, B.Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Transactions On Circuits and System For Video Technology, 4(4):438- 442, Aug.1994
- [7] Ken Sauer, Brian Schwartz, "Efficient block motion estimation using integral projections," IEEE Transactions on circuits and systems for video technology, Vol. 6 n.5, Oct. 1996
- [8] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures=Evolution Programs." Springer-Verlag, 1996
- [9] J.H.Holland. "Adaption in Natural and Artificial Systems," University Michigan Press. Ann Arbor,

1975

- [10] Keith Hung-Kei Chow and Ming L. Liou "Genetic Motion Search Algorithm for Video Compression." IEEE Transactions on Circuits and System For Video Technology, 3(6):440-445, Dec 1993
- [11] Chun-Hung Lin and Ja-Ling Wu, "A Lightweight Genetic Block-Matching Algorithm for Video Coding," IEEE Transactions on circuits and systems for video technology, Vol. 8. NO.4, Aug. 1998
- [12] Man F.So and Angus Wu, "Four-Step Genetic Search For Block Motion Estimation," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.3, May. 1998
- [13] D.E.Goldberg, Genetic Algorithm in Search, Optimization & Machine Learning. Reading, MA: Addison-Wesley, 1989
- [14] J.L.R. Filho and P.C Treleaven, " Genetic-algorithm programing environments," IEEE Comput. Mag, pp 28-43, June 1994
- [15] 배대규, 한준희 "유전자 알고리즘을 이용한 윤곽선 추출" 정보과학회논문지(B)제24권 제 10호, 1997년



전 병 민

1976년 한국항공대학교 전자공학과 공학사. 1978년 연세대학교 전자공학과 공학석사. 1988년 연세대학교 전자공학과 공학박사. 1978 ~ 1982년 공군사관학교 전자공학과 전임강사. 1982 ~ 1986년 중앙공업전문대학 통신과 조교수. 1992 ~ 1993년 미시간대학교 교환교수. 1986년 ~ 현재 충북대학교 컴퓨터공학과 교수. 관심분야는 영상신호처리, 패턴인식



박 지 영

997년 충북대학교 컴퓨터공학과 학사. 2000년 충북대학교 컴퓨터공학과 석사. 현재 삼성 톨슨 CSF. 관심분야는 영상신호처리, 동영상 압축, 움직임 추정



백 순 화

1985년 계명대학교 전자계산학과 공학사. 1993년 호서대학교 대학원 전자계산학과 이학석사. 2000년 충북대학교 대학원 컴퓨터공학과 공학박사. 1999년 ~ 현재 천안외국어대학 컴퓨터정보과 전임강사. 관심분야는 영상처리, 신호처리, 패턴인식

식