

이항 분포를 이용한 제한된 1비트 변환 움직임 예측의 고속 블록 정합 알고리즘

박한진^{a)}, 최창렬^{a)}, 정제창^{a)‡}

Fast block matching algorithm for constrained one-bit transform-based motion estimation using binomial distribution

Hanjin Park^{a)}, Changryoul Choi^{a)}, and Jechang Jeong^{a)‡}

요약

움직임 예측 분야에서 많은 고속 블록 정합 알고리즘들은 불필요한 움직임 후보 블록들을 고유한 조건식으로 필터링하는 방법, 즉 탐색 포인트의 수를 줄이는 방법으로 연산의 복잡도를 줄이고 있다. 비록 많은 고속 블록 정합 알고리즘들이 기존의 전역 탐색 알고리즘과 비교하여 연산량을 상당 부분 줄일 수 있다 하더라도, 각 조건식의 특성에 의해 때때로 어느 정도의 정합 오차를 감수해야 한다는 단점이 있다. 본 논문에서는 제한된 1비트 변환 움직임 예측을 위한 새로운 고속 정합 알고리즘을 제안 하며, 이는 전역 탐색 알고리즘 대비 화질의 열화를 최소화 하면서도 움직임 블록 예측시의 연산량을 현저하게 줄이는 것에 목적을 둔다. 기존의 고속 블록 정합 알고리즘들과는 달리 제안된 알고리즘은 연산량을 줄이는데 있어서 새로운 접근 방법을 보여준다. 그것은 1비트 변환 후의 이진 평면이 오직 0 과 1이라는 두 개의 성분만으로 이루어진다는 사실에 기초하여 이항 분포 (binomial distribution)를 활용한 접근 방법이다. 모의실험 결과 제안된 알고리즘은 기존의 전역 탐색 기법을 적용한 제한된 1비트 변환 움직임 예측과 비교하여 PSNR (Peak signal-to-noise ratio) 성능은 매우 근접하게 유지하면서도 연산량은 획기적으로 줄여주는 효과를 보여 준다.

Abstract

Many fast block-matching algorithms (BMAs) in motion estimation field reduce computational complexity by screening the number of checking points. Although many fast BMAs reduce computations, sometimes they should endure matching errors in comparison with full-search algorithm (FSA). In this paper, a novel fast BMA for constrained one-bit transform (C1BT)-based motion estimation is proposed in order to decrease the calculations of the block distortion measure. Unlike the classical fast BMAs, the proposed algorithm shows a new approach to reduce computations. It utilizes the binomial distribution based on the characteristic of binary plane which is composed of only two elements: 0 and 1. Experimental results show that the proposed algorithm keeps its peak signal-to-noise ratio (PSNR) performance very close to the FSA-C1BT while the computation complexity is reduced considerably.

Keyword : Constrained one-bit transform, motion estimation, early termination, binomial distribution, block matching algorithm

a) 한양대학교 전자통신컴퓨터공학과

Department of Electronics Computer Engineering, Hanyang University

‡ 교신저자 : 정제창 (jjeong@ece.hanyang.ac.kr)

· 접수일(2011년8월8일), 수정일(2011년9월16일), 게재확정일(2011년9월23일)

1. 서론

디지털 동영상 압축 시스템에서 움직임 예측 기법은

연속된 영상 프레임 사이의 시간적 중복성을 없애줌으로써 전송 데이터의 양을 획기적으로 줄일 수 있는 기반을 마련해 준다는 의미에서 핵심적인 역할을 하고 있다. 일반적으로 움직임 예측은 동영상 부호화 시스템 전체 연산량의 80%를 차지하는 만큼 이 과도한 연산량을 줄이기 위한 요구는 끊임없이 계속되어지고 있다. 움직임 예측 분야에서 가장 흔하게 사용되어지는 기술은 블록 정합 알고리즘 (Block Matching Algorithm : BMA)^[1,2]이다. 이는 각각의 영상 프레임을 정방향 블록으로 나눈 후 가장 적합한 정합 블록을 이전 프레임의 탐색 윈도우 안에서 찾아내는 방법이다. 여러 블록 정합 알고리즘들 중에서도 최적의 움직임 벡터를 찾는 데 있어서 가장 적합한 방법으로 알려져 있는 것은 전역 탐색 알고리즘 (Full Search Algorithm : FSA)이다. 하지만 전역 탐색 알고리즘의 방대한 연산량은 언제나 실시간 동영상 부호화기의 구현에 큰 어려움을 주어 왔다. 그래서 이러한 전역 탐색 알고리즘의 문제점을 해결하기 위하여 많은 고속 블록 정합 알고리즘들, 특히 1비트 변환 (one-bit transform : 1BT)^[3], 2비트 변환 (two-bit transform : 2BT)^[4] 그리고 제한된 1비트 변환 (constrained one-bit transform : C1BT)^[5,6]과 같은 여러 이진 블록 정합 움직임 예측 알고리즘들이 고안 되었다. 상기 이진 블록 정합 움직임 예측 알고리즘들은 연산의 복잡도를 줄이기 위해 서로 다른 정합 기준을 사용한 방법이며, 하드웨어적으로 여러 비트를 한 번에 처리함으로써 연산량을 효과적으로 줄일 수 있는 장점 때문에 널리 사용 되어지고 있다. 하지만 1BT의 경우 정합 오차가 가질 수 있는 범위가 좁아지기 때문에 부정확한 움직임 벡터를 구할 확률이 증가 하게 되는 단점을 가지고 있다. 그래서 2BT와 C1BT등 개선된 이진 블록 정합 움직임 예측 알고리즘들이 1BT의 부정확한 정합 오차를 개선하기 위하여 추가로 제안 되었다. 우선 본 논문에서 제안하는 알고리즘을 상세하게 기술하기 전에, 제안 되는 알고리즘의 선행 연구 기술인 1BT와 C1BT의 기본적인 이론을 알아보도록 하자.

1. 1비트 변환 (1BT) 움직임 예측 알고리즘

1BT 움직임 예측 알고리즘은 우선 원 영상 프레임들 식

(1)에서 기술하는 17X17 크기를 가지는 멀티 대역 통과 필터를 적용하여 새로운 영상 프레임을 만들어내고, 그 새롭게 만들어낸 출력과 원 영상을 비교하여 멀티 비트로 이루어진 원 영상을 이진 표현으로 변환하게 된다. 그 후 기존의 움직임 예측 방법인 전역 탐색 알고리즘을 수행하여 최적의 움직임 벡터를 찾게 된다. 1BT에 필요한 대역 통과 필터는 식 (1)과 같이 정의한다.

$$K(i,j) = \begin{cases} 1/25, & \text{if } i,j \in [0,4,8,12,16] \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

그리고 K를 거친 후 최종 이진 영상인 B(i,j)는 다음 식 (2)를 이용해 만들어진다.

$$B(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq I_F(i,j) \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

여기서 I(i,j)는 영상의 원본 프레임이고 I_F(i,j)는 필터 K가 적용된 후의 영상 프레임을 나타낸다. 즉 I_F(i,j)는 이진 영상을 얻기 위한 일종의 픽셀 단위의 문턱 값이 되며 이렇게 멀티 비트의 프레임을 이진 표현으로 바꾸는 전체 과정을 1비트 변환이라고 한다. 1BT 프레임 (B(i,j))를 구한 후 움직임 벡터는 기존의 전역 탐색 알고리즘에서 널리 사용 되어지고 있는 SAD (Sum of Absolute Differences) 기법 대신에 불리언 Exclusive-OR (XOR) 연산자를 사용한 NNMP (Number of Non-Matching Points)를 이용하며 그 정의는 다음과 같다.

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B^t(i,j) \oplus B^{t-1}(i+m,j+n) \quad (3) \\ -s \leq m, n \leq s$$

여기서 (m,n)은 움직임 후보 블록 (Candidate Motion Block : CMB)의 변위를, NXN 은 정방향 움직임 블록의 크기를 나타내며, s는 탐색 윈도우의 크기를 결정하게 된다. 또한 ⊕는 XOR 연산을 의미하며, B^t는 현재 (current) 영상의 이진 프레임을 그리고 B^{t-1}는 이전 (reference) 영상의 이진 프레임을 각각 나타낸다. 모든 움직임 후보 블록

중 최소의 $NNMP$ 값을 가지는 후보 블록의 변위가 (i, j) 위치의 움직임 블록의 최종 움직임 벡터가 된다. $NNMP$ 의 이해를 돕기 위해 기존에 사용 되는 SAD 를 식 (4)에 추가로 기술 한다. 여기서 f 는 멀티 비트를 가지는 화소의 크기를 의미한다.

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f^t(i, j) - f^{t-1}(i+m, j+n)| \quad (4)$$

$$-s \leq m, n \leq s$$

2. 제한된 1비트 변환 (C1BT) 움직임 예측 알고리즘

C1BT 움직임 예측 알고리즘은 1BT의 단점인 부정확한 PSNR (peak signal-to-noise ratio) 성능을 개선하기 위해 제안 되었다. PSNR의 정확도를 올리기 위해 C1BT는 제한된 마스크 (Constraint Mask : CM)라는 또 다른 이진 평면을 도입했으며 다음과 같이 정의된다.

$$CM(i, j) = \begin{cases} 1, & \text{if } |I(i, j) - I_F(i, j)| \geq D \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

식 (5)에서 주목해야 할 것은 복잡한 연산 과정을 통해 또 다른 정합 조건을 구하는 2BT와는 달리 C1BT는 특별한 연산 과정 없이, 즉 기존 1BT에서 사용 중인 $I_F(i, j)$ 를 이용하여 손쉽게 또 다른 정합 조건인 CM을 만들어낸다는 점이다. 간단하게 추가된 상기 정합 조건을 통하여 C1BT는 1BT대비 더욱 향상된 PSNR 성능을 보여주게 된다. 본 논문에서 문턱 값 D 는 실험적으로 최적화된 값 10이 적용되었으며, 1BT에서의 $NNMP$ 와 마찬가지로 C1BT에서는 $CNNMP$ (Constrained $NNMP$)라는 더욱 진보된 블록 정합 판별 조건이 만들어지며 정의는 식 (6)과 같다. 여기서 $\|$ 와 \cdot 는 각각 이진 OR와 AND 연산자이며, 나머지 항목들은 1BT의 항목들과 동일한 값을 가진다.

본 논문에서는 1BT의 단점인 부정확한 PSNR 특성을

개선하기 위해 제안된 C1BT 움직임 예측 알고리즘의 연산량을 줄이기 위하여 이항 분포 (Binomial distribution)를 이용한 새로운 조기 중단 알고리즘이 기술되며, 모의실험 결과 제안된 알고리즘은 기존의 전역 탐색 기법을 적용한 C1BT 움직임 예측과 비교하여 PSNR 성능은 매우 근접하게 유지하면서도 연산량은 획기적으로 줄여주는 효과를 보여 준다.

사실 제안되는 알고리즘의 핵심 아이디어는 이미 이전 논문^[7]에서 1BT의 연산량을 개선하기위한 목적으로 이미 제안된 적이 있다. 하지만 알고리즘의 특성상 이전 논문에서는 1BT의 연산량은 효과적으로 줄여주었지만 여전히 부정확한 PSNR 특성을 보여주는 단점이 있었다. 그래서 전역 탐색 기법을 적용한 기존 1BT의 광대한 연산량과 열악한 PSNR 특성 모두를 향상시키기 위해 이항분포를 이용한 상기 조기 중단 알고리즘을 C1BT에 확장 적용하여 1BT의 연산량과 PSNR 특성 모두를 함께 개선하는 것이 본 논문의 목적이다. 이제 본 논문에서 제안하는 기본 아이디어 및 핵심 항목들에 대해 이전 논문에서 기술 하였지만 다음 장에서 다시 한 번 언급하기로 한다.

II. 이전 알고리즘

이번 장에서는 1BT의 연산량을 개선하기 위해 이전 논문에서 제안된 이항 분포에 기반한 새로운 조기 중단 알고리즘에 대해서 다시 한 번 기술하고자 한다. 핵심 아이디어는 1BT 변환 후 현재와 이전 영상 프레임 각각의 움직임 블록이 오직 0과 1로 이루어진 이진 평면으로 이루어졌다는 사실에 기반하여 베르누이 시행 (Bernoulli trials)을 따르는 이항 분포로 모델링 할 수 있다는 데 있다. 물론 이항 분포를 적용하기 위해선 각 이진 평면의 값들이 서로 독립적 (Independent)이어야 하지만 조기 중단 판별식을 구하기 위해 이진 평면상의 값들은 서로 독립적인 베르누이 시행

$$CNNMP(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{ [CM^t(i, j) \| CM^{t-1}(i+m, j+n)] \cdot [B^t(i, j) \oplus B^{t-1}(i+m, j+n)] \} \quad (6)$$

$$-s \leq m, n \leq s$$

을 따른다고 가정한다.

이진 평면인 B^t 나 B^{t-1} 의 각 픽셀이 1이 나올 수 있는 확률을 성공 확률 p 그리고 움직임 블록 사이즈 (NXN)를 시행 횟수 n 으로 가지는 이항 분포로 모델링 할 수 있다는 가정 하에 NNMP의 기댓값을 나타내는 $NNMP_{EX}$ 는 이항 분포의 특성 (8)에 의해 n 과 p 의 곱으로 쉽게 구하여 진다. 또한 각 움직임 후보 블록의 $NNMP_{EX}$ 를 기준으로 실제 NNMP를 계산 할지 말지를 판단하기 위한 판별 조건으로 사용될 평균 (μ)과 표준 편차 (σ)는 모두 현재 영상의 이진 프레임 (B^t)에서 동일한 방법으로 유도된다. 이전 논문에서 1BT의 연산량 개선을 위해 제안되었던 조기 중단 알고리즘을 정리하면 다음과 같다. 즉, 후보 블록의 실제 NNMP를 계산하기 전에 우선 $NNMP_{EX}$ 를 μ 와 σ 로 이루어진 판별식과 비교하게 된다. 위 결과에 따라 확률적으로 최적의 움직임 벡터가 될 수 없다고 판단되는 불필요한 탐색 지점은 조기 중단하게 되고 반면에 확률적으로 최적의 움직임 벡터가 될 가능성이 있는 움직임 후보 블록은 실제 NNMP를 계산하여 전체적으로 연산량을 줄일 수 있게 된다. 이제 본 논문의 이론적 배경이 되는 이항분포의 개략적인 특성과 핵심 항목들인 $NNMP_{EX}$, μ 그리고 σ 에 대하여 순서대로 알아보도록 하자.

1. 이항 분포

제안 하는 알고리즘의 이해를 돕기 위해 이항 분포의 정의와 특징을 다음과 같이 간략히 기술하도록 한다. 만약 어떤 독립적인 시행에서 그 사건이 일어날 확률을 p , 일어나지 않을 확률을 q ($=1-p$) 라 하고, 이 시행을 독립적으로 n 번 되풀이할 때, 이 실험은 이항 분포를 따른다고 정의하며, 상기 조건의 이항 분포를 따르는 확률 변수를 V 라고 한다면, V 는 다음과 같이 정의한다.

$$V \sim binomial(n, p) \tag{7}$$

또한 확률 변수 V 의 평균, 분산, 표준 편차는 순서대로 하기와 같이 이항 분포의 특성인 다음의 간단한 공식에 의

해 쉽게 계산되어 진다.

$$\begin{aligned} \mu &= E[V] = np \\ \sigma^2 &= Var[V] = np(1-p) \\ \sigma &= stdev[V] = \sqrt{np(1-p)} \end{aligned} \tag{8}$$

그리고 만약 n 이 충분히 크고 분포의 스큐 (skew)가 크지 않다면, 이항 분포는 정규 분포 (normal distribution)로 근사화 될 수 있다.

$$binomial(n, p) \sim normal(np, npq) \tag{9}$$

2. $NNMP_{EX}(m, n)$

이진 평면인 B^t 와 B^{t-1} 이 이항 분포를 따른다고 가정하면, 두 이진 평면의 XOR 연산의 결과 ($B^t \oplus B^{t-1}$) 또한 이항 분포를 따른다고 가정 할 수 있다. 이 특성을 이용하여 조기 중단을 위한 지표로 사용될 NNMP의 기댓값 $NNMP_{EX}$ 를 구할 수 있다. $NNMP_{EX}$ 를 유도하기 위해 우선 \underline{x} , \underline{y} 와 \underline{z} 라는 세 개의 벡터를 정의한다. \underline{x} 를 16X16의 크기를 가지는 현재 움직임 블록의 이진 벡터라고 하면 \underline{x} 는 다음과 같이 정의 된다.

$$\underline{x} = (x_0, x_1, \dots, x_{255}) \tag{10}$$

이제 움직임 블록의 크기인 16X16에서 유도된 각각의 모든 픽셀 값, 즉 총 256번의 시도 중에서 1이 나올 전체 경우의 수를 성공 확률로 가지는 확률 변수 X 를 정의 하면 가정에 의해 이항 분포를 따르는 다음 식으로 정의 된다.

$$X \sim binomial(256, \frac{w_H(\underline{x})}{256}) \tag{11}$$

여기서 $w_H(\cdot)$ 는 0이 아닌 수의 개수를 의미하는 해밍 웨이트 (Hamming weight)를 나타낸다. 같은 방법으로 이전 움직임 블록의 이진 벡터 \underline{y} 와 확률 변수 Y 도 다음과 같이

정의 된다.

$$\begin{aligned} \underline{y} &= (y_0, y_1, \dots, y_{255}) \\ \mathbb{Y} &\sim \text{binomial}(256, \frac{w_H(\underline{y})}{256}) \end{aligned} \quad (12)$$

마지막으로 \underline{x} 와 \underline{y} 두 벡터의 XOR 연산의 결과 값을 각 원소로 가지는 16×16 의 크기의 이진 벡터 \underline{z} 와 확률 변수 \mathbb{Z} 를 정의하면 다음과 같다.

$$\begin{aligned} \underline{z} &= (x_0 \oplus y_0, x_1 \oplus y_1, \dots, x_{255} \oplus y_{255}) \\ \mathbb{Z} &\sim \text{binomial}(256, A) \end{aligned} \quad (13)$$

여기서 A 는 \underline{x} , \underline{y} 두 벡터의 각 픽셀간의 XOR 연산의 결과가 1이 나올 확률을 의미한다. 이제 $NNMP_{EX}$ 를 유도하기 전에 주목해야 될 것은 $NNMP_{EX}$ 는 식 (3)의 정의에 의해 현재와 이전 영상 프레임의 두 이진 움직임 블록 (\underline{x} , \underline{y}) 간의 서로 다른 전체 픽셀 개수들의 기댓값이라는 점이다. 여기서 서로 다른 픽셀이 나올 확률은, 즉 $NNMP$ 에 사용되는 XOR 연산자의 특성상 이는 두 이진 움직임 블록의 각 픽셀의 XOR 연산의 결과가 1이 될 확률과 동일하다. 결론적으로 $NNMP_{EX}$ 를 구하기 위한 서로 같지 않은 픽셀이 나올 확률은 A 와 동일하고, 확률 변수 \mathbb{Z} 의 기댓값을 구하는 것이 $NNMP_{EX}$ 를 구하는 것이 된다. A 를 구하기 위해 우선 1이 될 모든 경우의 수를 살펴보면 다음과 같이 오직 두 가지 가능성만이 존재 하게 된다.

$$x_i \oplus y_i = (0 \oplus 1) = (1 \oplus 0) \quad (14)$$

그래서 위 두 경우의 수를 고려한 1이 될 확률 A 는 식 (15)와 같이 계산되어진다.

$$\begin{aligned} A &= \left(\frac{256 - w_H(\underline{x})}{256} \right) \times \left(\frac{w_H(\underline{y})}{256} \right) + \\ &\left(\frac{w_H(\underline{x})}{256} \right) \times \left(\frac{256 - w_H(\underline{y})}{256} \right) \end{aligned} \quad (15)$$

계산 측면에서 식 (15)를 살펴보자. 만약 0부터 255까지 모든 정수 256개에 해당하는 확률 즉 $w_H(\cdot)/256$ 과 $(256 - w_H(\cdot))/256$ 의 값을 미리 계산하여 LUT (Look Up Table)에 저장하여 사용한다면 그로 인해 약간의 메모리가 필요하겠지만 A 는 단지 두 번의 곱셈과 한 번의 덧셈만으로 쉽게 구할 수 있다. A 를 구했다면 확률 변수 \mathbb{Z} 의 기댓값 즉 $NNMP_{EX}$ 는 이항분포의 특성 (8)에 의해 다음과 같이 쉽게 구할 수 있다.

$$NNMP_{EX} = n \times A = 256 \times A \quad (16)$$

여기서 n 은 전체 베르누이 시행 횟수를 의미하며 256은 16×16 크기를 가지는 움직임 블록의 전체 픽셀 개수에서 유도되었다.

3. $\mu(i, j)$ and $\sigma(i, j)$

여러 움직임 후보 블록 중 실제 $NNMP$ 계산이 필요할지 여부를 결정하기 위해, 앞서 $NNMP_{EX}$ 라는 새로운 판별 지표를 유도하였다. 이제 이 $NNMP_{EX}$ 의 유효성을 감정하기 위한 새로운 판별 조건 또한 필요하다. 물론 (m, n) 번째 움직임 후보 블록의 $NNMP$ 를 대신하는 판별 지표인 $NNMP_{EX}$ 가 이항분포를 통한 확률로 얻어졌기 때문에 그 판별 조건 또한 동일한 확률 조건에서 만들어 져야 한다. 새로운 판별 조건을 유도하기 위해 다음과 같이 중요하면 서도 타당한 가정을 세울 수 있다. 즉 이전 프레임의 수많은 움직임 후보 블록 중 가장 적합한 움직임 후보 블록은 분명히 현재 움직임 블록 (\underline{x})의 특성과 거의 유사하거나 동일할 것이라는 가정이다. 여기서 특성이란 전체 256개의 픽셀 중 전체 1의 개수를 의미한다. 위 가정을 통해 \underline{x} 와 동일한 해밍 웨이트를 가지는 새로운 슈도 (pseudo) 벡터 \underline{x}' 을 정의한다. 즉 \underline{x}' 은 찾고자 하는 \underline{x} 의 이상적인 움직임 후보 블록이라 간주한다. 이제 \underline{x}' 이 정의 되었으니 앞에서 언급한 새로운 판별 조건인 (i, j) 번째 현재 움직임 블록의 평균 값 (μ)와 표준 편차 (σ)를 $NNMP_{EX}$ 와 동일한 방법으로

유도 할 수 있다. 두 값을 구하기 위해 16X16의 크기를 가지는 \underline{x} 와 \underline{x}' 의 XOR 결과를 값으로 가지는 이진 벡터 \underline{w} 와 확률 변수 \mathbb{W} 를 정의하면 다음과 같다.

$$\begin{aligned} \underline{w} &= \underline{x} \oplus \underline{x}' \\ \mathbb{W} &\sim \text{binomial}(256, B) \end{aligned} \tag{17}$$

식 (17)에서 B 는 A 를 구할 때와 마찬가지로 \underline{x} 와 \underline{x}' 두 벡터의 각 픽셀간의 XOR 연산의 결과가 1이 나올 확률을 의미하며, 동일한 방법으로 확률 B 는 다음과 같이 계산할 수 있다.

$$\begin{aligned} B &= \left(\frac{256 - w_H(\underline{x})}{256} \right) \times \left(\frac{w_H(\underline{x}')}{256} \right) + \\ &\left(\frac{w_H(\underline{x})}{256} \right) \times \left(\frac{256 - w_H(\underline{x}')}{256} \right) \end{aligned} \tag{18}$$

또한 식 (18)은 $w_H(\underline{x}) = w_H(\underline{x}')$ 의 조건하에 식 (19)로 간략화 될 수 있다.

$$B = \frac{512 \times w_H(\underline{x}) - 2 \times w_H(\underline{x})^2}{256^2} \tag{19}$$

확률 B 를 구했으므로 $NNMP_{EX}$ 를 유도 했을 때와 마찬가지로 μ 와 σ 도 이항분포의 특징인 식 (8)을 이용하여 쉽게 구할 수 있다.

$$\begin{aligned} \mu &= n \times B = 256 \times B \\ \sigma^2 &= n \times B \times (1 - B) \\ \sigma &= \sqrt{\sigma^2} \simeq 15 + 0.0125 \times \sigma^2 \end{aligned} \tag{20}$$

여기서 식 (20)의 세 번째 근사식은 참고 문헌 [4]에서 인용 되었다.

최적의 움직임 블록은 대부분 μ 근처에 위치하며, 또한 식 (9)에서 언급 하였듯이 이항 분포의 시행 횟수가 충분히 클 때 이항 분포는 정규 분포로 근사화 할 수 있다는 두

가지 가정 하에 조기 중단 방법을 통해 1BT의 연산량을 개선하기 위한 새로운 판별 조건은 다음 식 (21)로 정의 된다.

$$(\mu - k\sigma) \leq NNMP_{EX}(m, n) \leq (\mu + k\sigma) \tag{21}$$

여기에서 k 는 σ 의 계수로서 판별 조건 범위의 상 하한치를 결정하는 팩터이며 표준 편차의 특성에 따라 0부터 3사이의 값을 가지게 된다. 참조로 표 1은 누적 분포 함수(cumulative distribution function : cdf)를 통해 k 의 값에 따라 $NNMP_{EX}$ 가 범위 안에 존재할 확률을 보여 준다.

표 1. k 의 변화에 따라 $NNMP_{EX}$ 가 판별 범위 안에 존재할 확률
Table 1. The probability which the $NNMP_{EX}$ is within the criterion range with variations of k

k	2	1	0.5	0.25	0.16	0.08
%	95.4	68.2	38.4	19.74	13	9.8

표 1을 통해 k 의 값이 커질수록 판별 조건이 더욱 완화되어 PSNR 성능은 좋아지는 반면에 연산량은 그에 반해 늘어나리라는 사실을 직관적으로 알 수 있다. 다른 한편으로 만약 k 의 값이 0에 가까워진다면, 즉 판별 조건이 더욱 가혹해진다면 PSNR 성능은 감소하겠지만 반면에 좁아진 범위만큼 연산량은 상당히 줄어 들 것이다.

결론적으로 만약 움직임 후보 블록의 $NNMP_{EX}$ 가 조건 (21)을 만족하지 못한다면 해당 움직임 블록은 조기 중단 되고 다음 움직임 후보 블록으로 이동하게 된다. 만족한다면 최적의 움직임 벡터가 될 가능성이 있다고 판단되어 기존의 전역 탐색 방법을 적용하여 실제 $NNMP$ 를 구하게 된다.

III. 제안하는 알고리즘

이번 장에서는 CIBT 움직임 예측에 적용되는 새로운 조기 중단 알고리즘이 기술되며, 그 핵심 아이디어는 이미 II장에서 설명하였다. 상기 알고리즘을 CIBT에 적용하기 위

해 CIBT에 맞는 새로운 조기 중단 지표인 $CNNMP_{EX}$ 와 그에 따른 새로운 판별 조건인 μ_c, σ_c 를 고려해야 한다. 하지만 $CNNMP$ 의 정의 (6)에서 볼 수 있듯이 추가된 CM 연산으로 인해 $NNMP_{EX}, \mu, \sigma$ 와 비교할 때 $CNNMP_{EX}$ 와 μ_c, σ_c 를 구하기 위해서는 추가로 별도의 연산이 필요할 수 밖에 없다. 이제 II장에서와 동일한 방법으로 $CNNMP_{EX}$ 와 μ_c, σ_c 를 유도하여 추가되는 연산량에 대해서 확인하도록 한다.

1. $CNNMP_{EX}(m, n)$

$CNNMP_{EX}$ 를 유도하기 위해 우선 \underline{x}_c 와 \underline{y}_c 라는 두 개의 벡터를 정의한다. \underline{x}_c 와 \underline{y}_c 를 각각 현재 (CM^t)와 이전 (CM^{t-1}) 프레임의 제한된 움직임 블록의 이전 벡터라 가정한다면, $CNNMP$ 의 정의 (6)에 따라 새로운 이전 벡터 \underline{z}_c 와 확률 변수 \mathbb{Z} 은 다음과 같이 정의 할 수 있다.

$$\begin{aligned} \underline{z}_c &= (\underline{x}_c \parallel \underline{y}_c) \cdot (\underline{x} \oplus \underline{y}) \\ \mathbb{Z}' &\sim \text{binomial}(256, C) \end{aligned} \quad (22)$$

마찬가지로 1이 될 확률 C 는 $(\underline{x}_c \parallel \underline{y}_c)$ 와의 AND 조건을 고려하여 $NNMP_{EX}$ 에서의 확률 A 와 동일한 방법으로 다음 식 (23)과 같이 계산되어지며, 따라서 $CNNMP_{EX}$ 는 다음과 같다.

$$CNNMP_{EX} = n \times C = 256 \times C \quad (24)$$

2. $\mu_c(i, j)$ and $\sigma_c(i, j)$

CIBT에 맞는 새로운 판별 조건인 μ_c 와 σ_c 를 유도하기 위해 이전 벡터 \underline{w}_c 와 확률 변수 \mathbb{W}' 를 다음과 같이 정의하도록 한다.

$$C = \left[1 - \left(\frac{256 - w_H(\underline{x}_c)}{256} \right) \times \left(\frac{256 - w_H(\underline{y}_c)}{256} \right) \right] \times \left[\left(\frac{256 - w_H(\underline{x})}{256} \right) \times \left(\frac{w_H(\underline{y})}{256} \right) + \left(\frac{w_H(\underline{x})}{256} \right) \times \left(\frac{256 - w_H(\underline{y})}{256} \right) \right] \quad (23)$$

$$\begin{aligned} \underline{w}_c &= (\underline{x}_c \parallel \underline{x}'_c) \cdot (\underline{x} \oplus \underline{x}') \\ \mathbb{W}' &\sim \text{binomial}(256, D) \end{aligned} \quad (25)$$

같은 방법으로 확률 D 는 다음과 같이 구하여 진다.

$$D = \left[1 - \left(\frac{256 - w_H(\underline{x}_c)}{256} \right)^2 \right] \times \left[\frac{512 \times w_H(\underline{x}) - 2w_H(\underline{x})^2}{256^2} \right] \quad (26)$$

마찬가지로 D 를 구한 이후에 μ_c 와 σ_c 는 다음 식으로 계산되어 진다.

$$\begin{aligned} \mu_c &= n \times D = 256 \times D \\ \sigma_c &= \sqrt{(256 \times D \times (1 - D))} \end{aligned} \quad (27)$$

이제 제안하는 알고리즘을 CIBT에 적용하기 위한 모든 핵심 항목들 $CNNMP_{EX}, \mu_c$ 와 σ_c 를 전부 유도 하였다. 하지만 상기 항목들로 결과를 도출한 이후, 안타깝지만 위 항목들은 CIBT의 조기 중단을 위한 판별 지표와 조건으로 선택되지 못한다. 원래 CIBT에 본 논문의 조기 중단 기법을 적용하기 위한 기댓값을 구하기 위해서는 $NNMP_{EX}$ 가 아닌 $CNNMP_{EX}$ 를 사용하는 것이 타당하다. 하지만 본 논문에서는 다음 두 가지 이유로 인해 $CNNMP_{EX}$ 대신에 $NNMP_{EX}$ 를 CIBT의 조기 중단을 위한 판별 지표로 사용 하였다. 첫째 $NNMP_{EX}$ 대비 $CNNMP_{EX}$ 를 구하기 위해서는 추가적으로 과도한 연산이 필요하게 된다. 식 (23)과 (26)에서 알 수 있듯이 $CNNMP_{EX}$ 를 구하기 위해서는 $NNMP_{EX}$ 의 1이 나올 확률 A 를 구한 뒤 추가로 또 하나의 화소 문턱 조건인 ($C^t \parallel C^{t-1}$)이 1이 될 수 있는 세 가지 경우의 수를 함께 고려해야 하므로 A 보다는 훨씬 복잡한 연산을 필요로 하게 되어 조기 중단을 위한 연산의 효율성 측면에서 제안하는 알고리즘의 전체적인 성능을 떨어뜨리게 된다. 두 번째 이유는 조기 중단을 위한 지표로서 $NNMP_{EX}$ 가 $CNNMP_{EX}$ 와 거의 동일한 성능을 보여 준다는 점이다. 왜

나하면 판별지표인 $NNMP_{EX}$ 를 검증하기 위한 판별 조건인 μ 와 σ 가 $NNMP_{EX}$ 를 구하기 위한 기저인 B^t 에서 동일하게 얻어졌기 때문이다. 즉 $NNMP_{EX}$ 의 판별 조건으로서 μ_c 와 σ_c 가 아닌 μ 와 σ 를 사용하였기 때문에 조기 중단을 위한 조건식으로서 충분한 역할을 해낼 수 있게 된다.

표 3은 $NNMP_{EX}$ 가 화질 및 연산량 측면에서 별다른 열화 없이도 $CNNMP_{EX}$ 를 훌륭히 대신할 수 있음을 보여준다. 하지만 상기 이유를 떠나서 중요하게 짚고 넘어가야 할 점은 $NNMP_{EX}$, μ 와 σ 모두 단지 조기 중단을 위한 단순한 인자 일 뿐이라는 점이다. 즉, 조기 중단 기법을 통해 불필요한 움직임 후보 블록을 필터링한 후에는 그 다음 움직임 예측 방법이 1BT 이던 CIBT이냐는 중요한 문제가 되지 않는다.

이제 이전 논문에서 사용된 판별 조건 식 (21)의 연산량을 추가적으로 개선할 수 있는 방법을 알아보도록 하자. 만약 식 (21)의 모든 항을 256으로 나누어 주면 다음 식 (28)로 다시 써어 질 수 있으며 이는 판별 조건으로 식 (21)과 동일한 결과를 가지게 된다.

$$(\mu - k\sigma)/256 \leq A(m, n) \leq (\mu + k\sigma)/256 \quad (28)$$

앞에 언급하였듯이 (m, n) 번째 움직임 후보 블록의 각 인자가 1이 될 확률 $A(m, n)$ 는 단지 두 번의 곱셈과 한번의 덧셈만으로 쉽게 구할 수 있으며, 판별 조건인 $(\mu - k\sigma)/256$ 와 $(\mu + k\sigma)/256$ 은 k 의 값만 정해져 있다면 두식은 단지 $w_H(x)$ 만으로 이루어진 식이기 때문에 $w_H(x)$ 가 나올 수 있는 0부터 255까지 모든 경우의 수에 따라 값이 정해지게 된다. 그래서 해당하는 상 하한치 각 256개의 값을 미리 계산하여 LUT에 저장한 뒤 $w_H(x)$ 의 값에 따라 불러 쓸 수 있다면 어떤 연산도 필요치 않게 된다. 추가되는 것은 단지 $w_H(x)$ 의 값을 구한 후 그 값에 따라 LUT에 저장된 결과 값을 불러오기 위한 한번의 메모리 접근과 LUT를 위한 그저 작은 양의 메모리일 뿐이다.

표 2는 각 항목들의 움직임 블록 당 연산량을 실제 $CNNMP$ 를 구하기 위해 소요되는 연산량과 비교하여 제안하는 알고리즘의 연산량 측면에서의 효율성을 보여준다.

이때 $A(m, n)$ 의 총 덧셈 연산량에는 $w_H(y)$ 를 구하기 위한 255번의 덧셈 연산이 포함 되어있다. 하지만 $w_H(x)$ 는 현재 움직임 블록 당 단 한번만 구하면 되기 때문에, 최적의 움직임 벡터를 찾기 위한 블록 당 전체 움직임 예측에 소요되는 연산량과 비교하여 무시할 정도이기 때문에 $w_H(x)$ 를 구하기 위한 255번의 덧셈 연산량은 표 2에서는 고려 하지 않도록 한다.

표 2. 16x16 크기의 움직임 블록 당 연산량 비교
Table 2. Number of operations per 16x16 Motion Block

	CNNMP (m,n)	A (m,n)	$(\mu \pm k\sigma)/256$
+	255	256	0
x	0	2	0
XOR	256	0	0
OR	256	0	0
AND	256	0	0
Memory Access in LUT	0	2	1

표 2를 통해 알 수 있듯이 만약 식 (28)에 의해 조기 중단된 경우 움직임 블록 당 단 두 번의 곱셈과 추가되는 한번의 덧셈 연산만으로 각 256번의 XOR, OR, AND 연산을 줄일 수 있게 된다. 더군다나 경계 조건을 고려한다 하더라도 16X16 크기의 움직임 블록 당 하나의 최적의 움직임 벡터를 찾기 위해 총 984번의 움직임 후보 블록의 $CNNMP$ 계산이 필요하다고 했을 때, 더 나아가 전체 모든 움직임 블록을 고려했을 때 제안하는 조기 중단 알고리즘을 통해 줄일 수 있는 전체 연산량은 엄청나게 된다.

최종적으로 최소의 연산량으로 최적의 움직임 벡터를 찾기 위해 제안하는 알고리즘의 정규 절차는 다음과 같다. 움직임 후보 블록의 실제 $CNNMP(m, n)$ 를 계산하기 전에 각 움직임 후보 블록의 $A(m, n)$ 를 μ 와 σ 로 이루어진 판별식과 비교하게 된다. 이때 μ 와 σ 는 현재 움직임 블록 (i, j) 당 오직 한 번만 구하면 된다. 만약 $A(m, n)$ 이 판별 조건 식 (28)을 만족하지 못한다면 $CNNMP(m, n)$ 는 움직임 벡터가 될 가능성이 없다고 판단하여 실제 $CNNMP(m, n)$ 계산을 생략하고 다음 움직임 후보 블록으로 이동하게 된다. 만족한다면 기존의 전역 움직임 탐색 기

법을 적용하게 된다. 그림 1은 본 논문에서 제안하는 조기 중단 알고리즘의 전체적인 순서도를 보여준다.

IV. 실험 결과 및 분석

모의실험은 다양한 움직임 특성을 보이는 ‘Akiyo’, ‘Football’, ‘Foreman’, ‘Hall’, 및 ‘Mobile’ 이렇게 다섯 가지 CIF (352×288, 100 frames) 시퀀스를 사용하였으며 움직임 예측을 위한 것이므로 휘도 성분만이 고려되었다. 실험 조건으로 움직임 블록 크기와 탐색 범위는 각각 16×16과 ±16이 사용 되었으며 탐색 순서는 나선형 (spiral) 기법이 적용 되었다. 또한 제안하는 알고리즘의 객관적인 성능

을 측정하기 위해 다음 두 가지 측정법이 사용 되었다. 첫 번째 측정 방법은 각 시퀀스의 움직임 블록 당 소요된 평균적인 탐색수인 ANSP (Average number of search positions per block), 즉 그림 1의 식 (28)의 조기 중단 과정을 거친 후 실제 CNNMP 연산을 한 움직임 블록 당 평균 횟수이고 두 번째는 화질 비교에 흔히 쓰이는 복원 영상의 PSNR 성능 측정 방법이다. 참조로 PSNR의 수식은 다음과 같다.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{29}$$

$$MSE = \frac{1}{N} \sum_{i=0}^{n-1} (x_i - y_i)^2$$

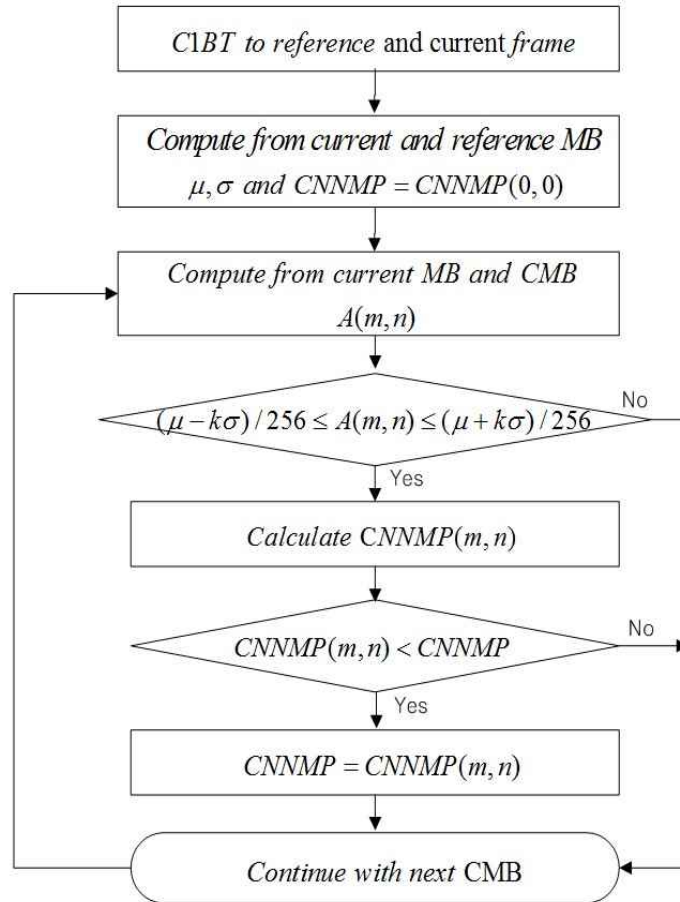


그림 1. 제안하는 알고리즘의 순서도
Fig. 1. Overall flowchart of the proposed algorithm

우선 표 3은 III장에서 언급 하였듯이 $NNMP_{EX}$, μ , σ 와 $CNNMP_{EX}$, μ_c , σ_c 의 서로 다른 두 가지 판별 조건으로 실험하였을 때 각각의 결과이다. 표 3의 결과를 통해 예상대로 $CNNMP_{EX}$ 를 사용한 방법이 $NNMP_{EX}$ 를 사용한 방법 보다 전체적으로 미미하나마 더 나은 특성을 보이고 있지만, 또한 조기 중단을 위한 판별 지표로서 $NNMP_{EX}$ 가 $CNNMP_{EX}$ 를 충분히 대신 할 수 있음을 함께 보여준다. 앞에서 언급하였듯이 $CNNMP_{EX}$ 를 구하기 위해 추가로 필요로 하는 연산량과 결과의 미미한 차이를 고려하여 이후 실험은 $NNMP_{EX}$ 를 통한 방법으로 진행되었고, 표준 편차 (σ)의 계수 k 는 표 3과 4에서는 실험적으로 최적화된 0.25가 적용 되었다.

표 4는 제안하는 알고리즘 성능의 공정한 비교를 위해 이진 변환 하지 않은 전형적인 MAD (Minimum Absolute Difference)와 이진 변환 후 기존의 전역 탐색 기법을 적용한 2BT, 1BT, C1BT, 그리고 제안하는 조기 중단 기법을 적용한 1BT^[7], C1BT의 PSNR과 ANSP 이득을 함께 비교하였다. 여기서 ANSP 이득이란 전역 탐색시의 움직임 후보

블록의 탐색 횟수 대비 제안하는 알고리즘을 적용 했을 때의 개선 이득을 의미한다. 시퀀스별 그리고 k 값에 따른 자세한 ANSP 값은 표 5를 통해 확인 할 수 있다.

표 4의 모의실험 결과를 통해 제안 하는 조기 중단 알고리즘이 PSNR을 통한 화질 판단 기준에서 기존의 전역 탐색 기법을 이용한 C1BT 대비 평균적으로 0.15dB의 미미한 손실, 즉 화면상으로 동일한 화질을 보여 줄을 알 수 있다. 반면에 ANSP 이득은 평균적으로 전역 기법을 사용한 다른 알고리즘 대비 평균 62.3%, 최대 71.66%나 개선됨을 보여 준다. 게다가 제안하는 알고리즘은 연산량이 획기적으로 줄어드는 데도 불구하고 PSNR 측면에서 상용중인 이진 변환 기법인 1BT 대비 모든 시퀀스에서 월등한 성능을 보여주며, 1BT는 물론 C1BT와 비교하여도 게다가 더 복잡한 연산량을 필요로 하는 2BT와 비교하여도 거의 동일하거나 우수한 성능을 보여준다.

이제 표 1에서 언급했듯이 또 다른 측면에서 k 값의 변화에 따른 PSNR과 연산량과의 관계를 확인해 볼 필요가 있다. 표 5는 0.1부터 2까지 여러 k 에 값에 따른 제안된 알고

표 3. 판별 지표에 따른 평균 PSNR (dB)와 ANSP (탐색 횟수)
Table 3. Average PSNR (dB) and ANSP (Searching points) according to the evaluation factors

Method \ Sequences		Akiyo	Football	Foreman	Hall	Mobile	Average
C1BT with NNMP EX	PSNR	42.47	22.75	32.08	33.90	23.50	30.94
	ANSP	295	553	279	333	394	371
C1BT with CNNMP EX	PSNR	42.48	22.91	32.13	33.96	23.58	31.01
	ANSP	212	458	278	392	624	393

표 4. 시퀀스 별 평균 PSNR (dB)와 ANSP 이득
Table 4. Average PSNR (dB) and ANSP gain (%) for various sequences

Method \ Sequences	Akiyo	Football	Foreman	Hall	Mobile	Average
MAD	42.81	24.20	33.45	34.35	24.47	31.86
2BT [4]	42.40	23.03	31.90	33.26	23.57	30.83
1BT [3]	42.03	22.66	31.82	32.51	23.52	30.51
1BT with early termination [7]	42.04	22.38	31.61	32.75	23.41	30.44
C1BT [5]	42.51	23.07	32.27	33.99	23.61	31.09
C1BT with early termination	42.47	22.75	32.08	33.90	23.50	30.94
ANSP gain (%) with early termination	70.00	43.83	71.66	66.14	60.00	62.30

표 5. k 값의 변화에 따른 PSNR (dB)와 ANSP (탐색 횟수)
 Table 5. PSNR (dB) and ANSP (Searching points) with variations of k

Test Sequences		C1BT	C1BT with variations of k				
			2	1	0.5	0.25	0.10
Akiyo	PSNR	42.51	42.51	42.51	42.51	42.47	42.19
	ANSP	984	776	557	449	295	156
Football	PSNR	23.07	23.07	23.07	22.95	22.75	22.22
	ANSP	984	948	861	706	553	343
Foreman	PSNR	32.27	32.27	32.25	32.22	32.08	31.51
	ANSP	984	800	621	443	279	140
Hall	PSNR	33.99	33.99	33.99	33.96	33.90	33.73
	ANSP	984	870	716	507	333	176
Mobile	PSNR	23.61	23.61	23.61	23.60	23.50	22.93
	ANSP	984	943	876	572	394	226
Average	PSNR	31.09	31.09	31.09	31.05	30.94	30.52
	Rate	1.00	1.00	1.00	0.99	0.99	0.98
	ANSP	984	867	726	535	371	208
	Rate	1.000	0.882	0.738	0.544	0.377	0.212
	Gain (%)	0	11.8	26.2	45.6	62.3	78.8

리즘의 PSNR과 ANSP 결과를 보여준다.

표 5의 결과를 통해서 제안하는 알고리즘의 k 의 값의 변화에 따른 PSNR과 연산량의 상관관계를 알 수 있다. 예상과 같이 k 의 값이 작아질수록 PSNR 손실이 미미하게 이루어지는 대신에 연산량은 급격하게 줄어드는 것을 볼 수 있다. 예를 들면 k 값이 1일 때는 전역 탐색 방법과 비교하여 PSNR 손실은 거의 없는 대신에 ANSP는 평균적으로 26% 정도의 소폭 개선되는 효과 밖에 없지만, k 값이 0.1일 때는 시퀀스에 따라 최대 0.85dB의 ('Football') PSNR 손실이 발생 하지만 반면에 ANSP는 평균 78.8% 정도로 획기적으로 좋아 짐을 알 수 있다. 특히 'Akiyo'나 'Hall' 시퀀스처럼 움직임 양이 크지 않은 영상에서는 k 값이 작다 하더라도 PSNR 손실이 거의 발생하지 않는 것을 볼 수 있다. 상기 결과로부터 동영상의 특성에 따라 k 값을 효율적으로 조절 할 수 있다면 k 값이 연산량을 효율적으로 조절 할 수 있는

강력한 도구로 쓰일 수 있음을 확인 할 수 있다.

V. 결 론

본 논문에서는 이전 변환을 통한 C1BT 움직임 예측 기법의 연산량을 줄이기 위한 새로운 조기 중단 알고리즘을 제안하고 모의실험을 통하여 제안하는 알고리즘의 성능을 검증하였다. 제안하는 알고리즘은 현재와 이전 움직임 블록을 이항 분포로 모델링하여 불필요한 탐색 위치를 필터링 하기 위한 효과적인 판별 조건을 만들어 냈고 전역 탐색 기법과 거의 동일한 화질 성능을 유지하면서도 연산량은 획기적으로 줄일 수 있는 새로운 조기 중단 기법을 보여 주었다. 이는 극도의 연산량을 필요로 하는 움직임 예측부의 단점을 개선하여 동영상 부호화기의 실시간화에 큰 도움이 될 것이라 기대 할 수 있다.

참 고 문 헌

[1] Information Technology - Coding of Audio Visual Objects - Part 2 : Visual, JTC1/SC29/WG11, ISO/IEC 14496-2 (MPEG-4 Visual), 2002.

[2] "Advanced Video Coding for Generic Audiovisual Services,"ITU-T Recommendation H.264, May 2005.

[3] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity Block-based Motion Estimation via One-Bit Transforms," IEEE Trans. Circuits Syst. Video Technol., vol.7, no.5, pp.702-706, Aug. 1997

[4] A. Erturk and S. Erturk, "Two-Bit Transform for Binary Block Motion Estimation," IEEE Trans. Circuits Sys. Video Technol., vol.15, no.7, pp.938-946, July2005

[5] O. Urhan and S. Erturk, "Constrained one-bit transform for low complexity block motion estimation," IEEE Tans. Circuits and Syst. Video Technol., vol.17, no.4, pp.478-482, April2007.

[6] O. Urhan "Constrained one-bit transform based fast block motion estimation using adaptive search range," IEEE Tans. Consumer Electron, vol.56, no.3, Augst 2010.

[7] H. Park, C. Choi, J. Jeong, "Early termination algorithm for one-bit transform-based motion estimation using binomial distribution," WIAMIS April 2011.

저 자 소 개



박 한 진

- 2001년 2월 : 홍익대학교 전파공학과 졸업
- 2001년 1월 ~ 현재 : 삼성전자 영상디스플레이 사업부 책임연구원
- 2010년 3월 ~ 현재 : 한양대학교 전자통신컴퓨터공학과 석사과정
- 주관심분야 : 영상처리, 영상압축, 영상화질 개선



최 창 렬

- 1997년 2월 : 한양대학교 전파공학과 졸업
- 2010년 9월 : 한양대학교 전자통신공학과 포닥과정
- 2011년 9월 ~ 현재 : 한양대학교 전자통신공학과 계약교수
- 주관심분야 : Digital Watermarking, 영상압축, 영상처리



정 제 창

- 1980년 2월 : 서울대학교 전자공학과 졸업
- 1982년 2월 : KAIST 전기전자공학과 석사
- 1990년 : 미국 미시간대학 전기공학과 공학박사
- 1980년 ~ 1986년 : KBS 기술연구소 연구원 (디지털 및 뉴미디어 연구)
- 1990년 ~ 1991년 : 미국 미시간대학 전기공학과 연구교수 (영상 및 신호처리 연구)
- 1991년 ~ 1995년 : 삼성전자 멀티미디어 연구소 (MPEG, HDTV, 멀티미디어 연구)
- 1995년 ~ 현재 : 한양대학교 전자통신컴퓨터공학과 교수 (영상통신 및 신호처리 연구실)
- 1998년 11월 : 과학기술자상 수상
- 1990년 12월 : 정보통신부장관상 수상
- 2007년 : IEEE Chester Sall Award 수상
- 2008년 : ETRI Journal Paper Award 수상
- 2011년 5월 : 제46회 발명의 날 녹조근정훈장 수훈
- 주관심분야 : 영상처리, 영상압축, 3DTV