

# 신뢰성있는 웨이블릿 비디오 전송을 위한 패킷화 기법

이 주 경<sup>†</sup>·강 진 미<sup>†</sup>·김 충 길<sup>††</sup>·정 기 동<sup>†††</sup>

## 요 약

웨이블릿 변환(Wavelet Transform)된 비디오는 주파수와 해상도가 다른 부대역으로 분해되므로 전송 오류가 발생한 패킷의 위치에 따라 복원된 프레임 간 화질 편차가 크게 된다. 복원된 프레임의 화질 변화가 클수록 사용자가 느끼는 비디오의 화질은 떨어진다. 특히, 움직임 예측을 이용한 웨이블릿 비디오의 경우, 특정 부대역에서 발생한 오류는 같은 프레임의 다른 부대역 뿐 아니라 이후 프레임의 화질에도 지속적인 영향을 미치게 된다. 본 논문에서는 웨이블릿 기반 비디오를 네트워크로 전송하기 위해 패킷화를 수행할 때, 오류발생 패킷의 위치에 관계없이 일정한 화질을 유지하며 오류 은닉이 쉬운 블록기반 패킷화 기법인 BDP(Block based Dispersive Packetization)를 제안한다. 본 논문은 MRME(Multi-Resolution Motion Estimation)를 적용하여 압축된 비디오와 무선 네트워크에서의 오류 발생 모델을 이용하여 성능평가를 수행하였다. 실험결과 제안된 기법은 프레임을 일정한 블록으로 분할하여 순차적으로 패킷화하는 BP나 픽셀단위로 분산하는 DP기법에 비해 주·객관적인 성능 모두 뛰어난 결과를 알 수 있었다.

## Packetizing Scheme for Reliable Transmission of Wavelet Video Stream

Joo-Kyong Lee<sup>†</sup> · Jin-Mi Kang<sup>†</sup> · Chung-Kil Kim<sup>††</sup> · Ki-Dong Chung<sup>†††</sup>

## ABSTRACT

Since Wavelet Transform decomposes a video frame into subbands with various frequencies and resolutions, the reconstructed video quality at a receiver fluctuates according to the location of transmission errors within frames. This deteriorates the whole visual quality of the video. Specifically, for a wavelet based video which exploits the motion estimation prediction scheme, the transmission errors of a subband not only have a bad effect on other subbands within a same frame but also propagates to the subsequent frames. In this paper, we propose BDP(Block Based Dispersive Packetization) scheme, for a wavelet based video stream, which maintains constant video quality despite packet location that a transmission error occurs. To evaluate the performance of the proposed scheme, we use MRME(Multi-Resolution Motion Estimation) scheme to compress a video in Inter coding mode and Gilbert's error model to generate the error patterns in wireless network environment. The simulation results show that BDP is more efficient than BP (Block based Packetization) or DP (Dispersive Packetization) in both PSNR and visual quality.

**키워드 :** 웨이블릿 변환(Wavelet Transform), 비디오 압축(Video Coding), 패킷화(Packetization), 오류전파(Error Propagation)

### 1. 서 론

개인 휴대용 단말기가 널리 보급되면서 멀티미디어 데이터 전송을 위한 네트워크 환경은 기존의 유선 환경에서 무선 환경으로 급속히 변하고 있다. 이에 따라 전송 대역폭이 낮은 비디오 압축 표준인 H.263[1], MPEG-4[2] 등이 관심을 끌고 있으며 이미지 압축 표준으로 JPEG-2000[19]이 최근 발표되었다. 또한 낮은 비트율에서 효율적인 웨이블릿 기반 비디오 코덱 구현에 많은 연구가 진행되어 왔다[3-5].

일반적으로 비디오 코덱은 압축효율을 높이기 위해 프레임간의 차이값에 VLC(Variable Length Coding)를 수행하

는데 압축된 데이터는 네트워크 오류에 쉽게 손상되는 단점이 있다. 비디오의 네트워크 오류를 제어하는 기법은 수행 주체에 따라 부호기에서의 소스코딩 기법[6-8], 복호기에서의 오류은닉 기법[9, 10], 부호기와 복호기가 상호 작용을 수행하는 피드백 채널 기법[11, 12] 등으로 나눌 수 있으며 이 기법들은 대부분 DCT 기반 코덱을 가정하고 있다. 웨이블릿 기반 이미지 및 비디오의 오류제어 기법으로는 SPIHT(Set Partitioning In Hierarchical Trees)에 적용한 소스 코딩 기법[13, 14], 프레임 내 압축(Intra coding)된 웨이블릿 비디오를 패킷화하는 기법[15], 이미지 패킷 기법[16]이 있으며 프레임간 참조(Inter coding)를 이용하는 비디오의 오류제어에 대한 연구는 전무하다.

본 논문에서는 프레임간 참조를 이용하는 웨이블릿 비디오의 네트워크 오류를 제어하기 위한 패킷화 기법을 제안하

<sup>†</sup> 준 회 원 : 부산대학교 대학원 컴퓨터공학과

<sup>††</sup> 준 회 원 : 부산대학교 대학원 멀티미디어 협동전공

<sup>†††</sup> 총 산 회 원 : 부산대학교 전자계산학과 교수

논문접수 : 2003년 5월 19일, 심사완료 : 2003년 8월 19일

고자 한다. 본 논문은 프레임 전체에 대하여 웨이블릿 변환을 수행하는 옥타브 웨이블릿 변환(Octave Wavelet Transform)<sup>1)</sup>을 가정하고 있으며 옥타브 웨이블릿 변환된 비디오는 분할된 부대역마다 그 중요도가 다른 특징이 있다. 이 부대역을 프레임 내의 위치에 따라 차례로 패킷화한다면 네트워크 오류가 발생한 패킷의 위치에 따라 비디오의 화질 편차가 매우 크게 되며, 이는 수신측 사용자의 느끼는 화질을 감소시키는 요인이 된다. 본 논문에서는 오류가 발생한 패킷의 위치에 관계없이 일정한 화질을 유지하고 오류 은닉이 쉬운 BDP(Block Based Dispersive Packetization) 기법을 제안한다. BDP 기법은 전체 부대역을 동일한 수의 블록으로 나누고, 공간-주파수상으로 이웃한 블록이 같은 패킷에 할당되지 않도록 인터리빙을 수행하는 기법이다. 제안하는 기법은 블록 단위의 인터리빙을 수행함으로써 DP(Dispersive Packetization)에 비해 오류 발생 영향의 범위를 줄일 수 있으며, 사용자의 주관적인 화질도 높일 수 있다.

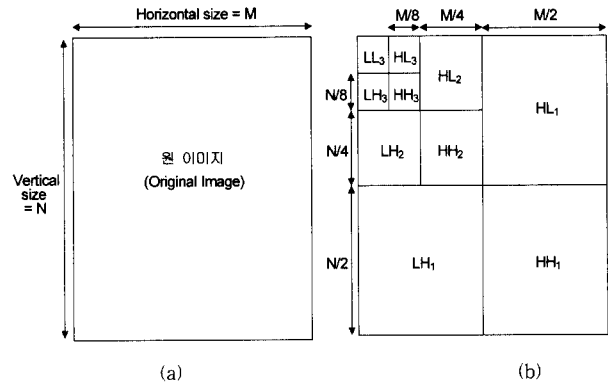
본 논문의 구성은 다음과 같다. 2장에서는 웨이블릿 변환의 특성을 알아보고 3장에서는 웨이블릿 비디오의 패킷화 기법인 PZW, DP 기법을 알아본다. 4장에서는 본 논문에서 제안하는 BDP 기법을 설명하고, 5장에서 무선 환경에서의 오류 발생을 모의 실험하여 제안된 기법의 성능을 비교 평가한다. 6장에서는 결론과 향후 연구 과제에 대하여 밝힌다.

## 2. 웨이블릿 변환

2차원 웨이블릿 변환된 프레임은 주파수 및 해상도가 각각 다른 여러 부대역으로 분해되므로 다양한 크기의 비디오 전송이 가능하다는 장점이 있다. 이 부대역들은 하나의 저주파수 부대역(LL)과 수평, 수직, 대각선 방향 성분을 가지는 고주파수 부대역으로 구성된다. 저주파수의 부대역은 프레임 전체의 대략적인 정보를 유지하기 때문에 전체 부대역 중에서 가장 중요하며 그 외 부대역은 각각 수평, 수직, 대각선 방향의 에지 정보를 포함한다.

(그림 1)(b)는 3단계 웨이블릿 변환된 결과로 총 10개의 부대역이 생성된 모습을 보여주고 있다. 이때, 웨이블릿 변환 단계가 높을수록 픽셀의 중요도는 커지게 되고, 원이미지를 복원할 때 상위 단계의 부대역은 하위 단계의 복원에 영향을 미치게 된다. 예를 들면, 2단계 변환된 부대역들에 대하여 IDWT(Inverse Discrete Wavelet Transform)를 수행하기 위해서는 LL<sub>2</sub>, LH<sub>2</sub>, HL<sub>2</sub>, HH<sub>2</sub>가 필요하며 이 중 LL<sub>2</sub> 대역은 3단계 부대역(LL<sub>3</sub>, LH<sub>3</sub>, HL<sub>3</sub>, HH<sub>3</sub>)을 IDWT 한 것이다. 이러한 웨이블릿 변환의 특성으로 프레임을 단순히

적당한 크기로 분할하여 순차적으로 패킷화한다면 특정 패킷에 중요한 부대역이 집중되는 현상이 발생하게 된다.



(그림 1) 웨이블릿 변환 전 이미지 (a)와 3단계 웨이블릿 변환 후 이미지 (b)의 예

## 3. 관련 연구

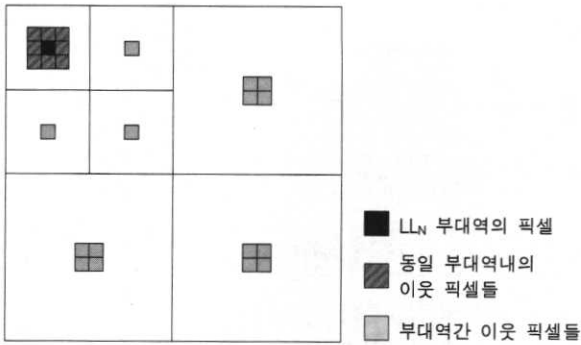
본 장에서는 웨이블릿 비디오의 패킷화 기법에 대한 기존 연구와 프레임간 참조 기법인 MRME 기법에 대하여 살펴보기로 한다. 본 논문은 MRME에 기반한 동영상의 패킷화 기법을 제안한다.

### 3.1 DP(Dispersive packetization)[15]

DP 기법은 각 패킷에 할당되는 계수들의 공간-주파수상의 상호연관성이 중복되지 않도록 분산시켜 패킷화하는 기법이다. (그림 2)는 2단계 변환된 이미지에서 특정 계수가 공간-주파수상으로 이웃한 계수를 나타낸다. DP 기법의 핵심은 LL<sub>M</sub>(M은 변환 단계) 부대역의 계수와 공간-주파수상의 이웃한 계수를 다른 패킷에 분산함으로써 LL<sub>M</sub> 부대역에 오류가 발생하더라도 공간-주파수 상의 이웃한 계수를 이용하여 오류 은닉을 수행할 수 있다는 것이다. 그러나 공간-주파수 상으로 이웃한 모든 계수가 완전히 분산되려면 필요한 총 패킷수가 크게 증가하므로 LL<sub>M</sub> 부대역의 이웃한 계수만 분산하고 나머지 부대역의 이웃한 픽셀은 중복 가능하도록 한다.

(그림 3)은 어떤 프레임의 LL<sub>4</sub> 부대역의 계수를 분산하기 위한 마스크의 예를 보여준다. 이 프레임은 9개의 패킷에 분산되어 저장되며 그림에서 각 계수는 이웃한 계수와 같은 패킷에 할당되지 않는다. 이 기법의 장점은 LL<sub>M</sub> 부대역의 모든 이웃한 계수를 서로 다른 패킷에 할당함으로써 이웃한 계수를 이용한 오류은닉이 쉽다는 것과 모든 패킷의 중요도가 동일하다는 것이다. 그러나 (그림 3)에서 알 수 있듯이 5번 패킷에 오류가 발생하면 LL<sub>M</sub> 부대역 전체에 오류 발생의 영향이 미치게 되고, 이는 프레임 전체의 복원에 영향을 미치게 되는 단점이 있다.

1) 2차원 웨이블릿 분해 방식은 이미지를 등간격으로 분해하여 부대역을 만드는 등분해 방식, 저주파 대역을 계속해서 나누어가는 옥타브(octave) 분해 방식, 특정부위를 나누어 가는 웨이블릿 패킷 방식이 있다.



(그림 2) 공간-주파수 상의 이웃한 픽셀들

1	2	3	7	8	9	1	2	3	7	8	9	1	2	3
4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
7	8	9	4	5	6	7	8	9	4	5	6	7	8	9
3	1	2	8	9	4	3	1	2	8	9	4	3	1	2
6	4	5	3	1	7	6	4	5	3	1	7	6	4	5
9	7	8	6	2	5	9	7	8	6	2	5	9	7	8
1	2	3	7	8	9	1	2	3	7	8	9	1	2	3
4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
7	8	9	4	5	6	7	8	9	4	5	6	7	8	9
3	1	2	8	9	4	3	1	2	8	9	4	3	1	2
6	4	5	3	1	7	6	4	5	3	1	7	6	4	5
9	7	8	6	2	5	9	7	8	6	2	5	9	7	8
1	2	3	7	8	9	1	2	3	7	8	9	1	2	3
4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
7	8	9	4	5	6	7	8	9	4	5	6	7	8	9
3	1	2	8	9	4	3	1	2	8	9	4	3	1	2
6	4	5	3	1	7	6	4	5	3	1	7	6	4	5
9	7	8	6	2	5	9	7	8	6	2	5	9	7	8
1	2	3	7	8	9	1	2	3	7	8	9	1	2	3
4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
7	8	9	4	5	6	7	8	9	4	5	6	7	8	9

(그림 3) 21×15 크기의 LL4 부대역 마스크의 예

3.2 PZW(Paketized Zerotree Wavelet)[16]

PZW 기법은 ATM 셀을 위하여 제안한 패킷화 기법이다. 이 기법은 ATM 셀과 동일한 크기의 패킷에 제로트리의 우선순위를 고려하여 트리 단위로 비트열을 할당한다. (그림 4)에서는 PZW 기법의 예를 나타낸다. 예에서 원래의 비트열은 여러 개의 제로트리가 중요도에 따라 인터리빙되어 제로트리의 루트가 포함된 패킷에 오류가 발생하면 트

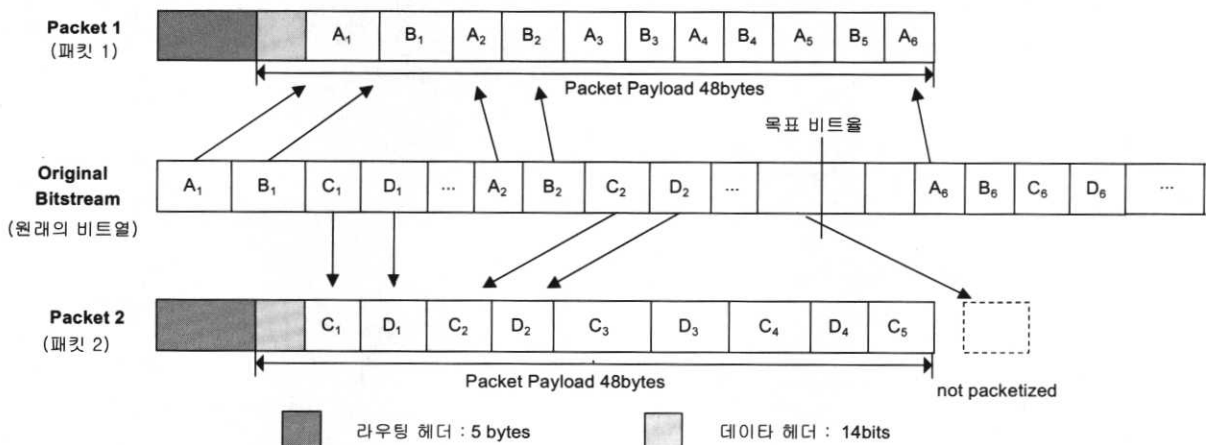
리의 나머지를 복원할 수 없는 단점이 있었다. PZW에서는 같은 제로트리에 속하는 비트열을 같은 패킷에 할당하여 특정 패킷에 오류가 발생하더라도 일부 제로트리에 오류가 국한되도록 한다. 예에서는 각 패킷에 2개의 제로트리를 할당하고 있으며 패킷 크기를 초과하는 비트열에 대해서는 패킷화를 수행하지 않으며(D5), 목표하는 비트열을 모두 채우고도 남으면 이후의 비트열도 포함한다(A6). 이 기법은 트리를 분산하여 패킷화함으로써 패킷이 손실되더라도 해당 패킷내의 트리만 손실되고 나머지 패킷의 트리는 보호된다는 장점이 있다.

3.3 MRME 기법[3]

MRME 기법은 웨이블릿 변환된 프레임간 움직임 예측을 수행하는 기법으로 블록매칭 기법이나 부대역간(subband-to-subband) 예측기법에 비하여 계산 복잡도와 성능에서 뛰어나다. MRME는 움직임 예측을 위해 최상위 단계 부대역의 모션벡터(Motion Vector)를 이용하여 하위 부대역의 움직임 예측에 활용한다. 최상위 단계의 어느 부대역을 이용하는냐에 따라 4가지의 기법이 있으며 본 논문에서는 LLM 부대역의 모션벡터를 이용하는 세 번째 기법을 이용하기로 한다. 이 기법의 구체적인 수행 과정을 살펴보면, 먼저 각 부대역을 동일한 수의 블록으로 분할하고, LLM 부대역의 각 블록에 대하여 모션벡터를 구한다. 이 모션벡터를 이용하여 나머지 부대역에서 모션벡터의 크기와 정확성을 조정하게 된다. 식 (3.1)은 i 단계의 j 부대역 (x, y)블록의 모션벡터를 구하는 식을 보여주고 있다. 이 값은 LLM 부대역의 모션벡터에 i 단계 부대역의 크기 비율 (2<sup>N-i</sup>)를 곱하여 기본적인 모션벡터를 생성하고, 좀 더 정확한 결과를 구하기 위해 움직임 예측 (δ<sub>i,j</sub>(x, y))를 수행하여 구할 수 있다.

$$V_{i,j}(x, y) = 2^{M-i} V_{M,0}(x, y) + \delta_{i,j}(x, y) \quad (3.1)$$

for  $i=0, 1, 2, \dots, M$  and  $j=0, 1, 2$



(그림 4) PAW 기법의 패킷화 예

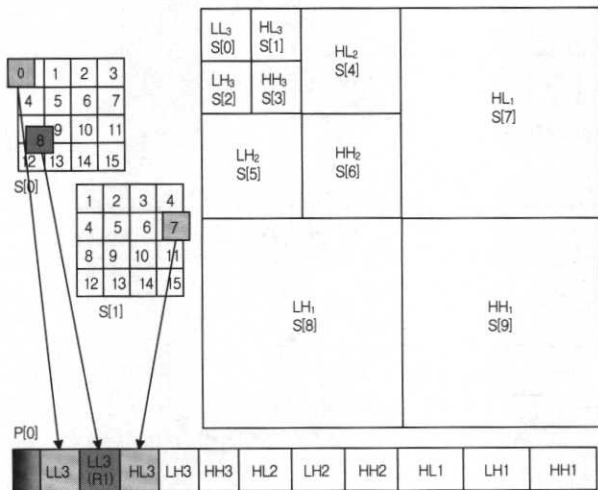
### 4. BDP(Block based Dispersive Packetization)

본 논문에서 제안하는 BDP 기법은 웨이블릿 동영상을 무선 네트워크로 전송할 때 전송 오류로 인한 품질저하를 개선하는 것에 목적을 두고 있다. 즉, 웨이블릿 변환된 동영상의 특징을 이용하여 추가의 소스코딩 기법을 적용하지 않고 오류은닉과 오류전파를 제어하자는 것이다. 웨이블릿 변환을 거친 프레임은 다해상도의 부대역으로 분할되며 각 부대역은 부대역의 위치에 따라 가로, 세로, 대각선 방향으로 연관성을 가진다. BDP 기법은 공간-주파수상의 상호 연관성을 이용하여 부대역을 분산함으로써  $LL_M$  부대역의 손실을 최대한 방지할 수 있다. 이를 위해 각 부대역을 동일한 수의 블록으로 분할하고 상호 연관성이 있는 블록이 같은 패킷에 할당되지 않도록 분산하여 저장한다. 이때,  $LL_M$  부대역을 보호하기 위해  $LL_M$  부대역의 각 블록을 중복하여 패킷에 할당하며 같은 패킷에 할당된 2개의  $LL_M$  블록은 다른 블록이어야 한다. 블록을 중복하여 전송하는 것은 패킷 사이즈를 증가시키는 단점이 있으므로 본 논문에서는  $LL_M$  부대역을 중복하고  $HH_1$  부대역을 삽입하지 않도록 한다.

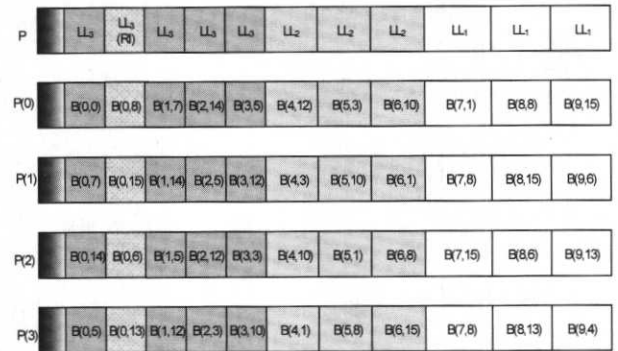
BDP기법은 블록단위로 패킷에 할당되므로 DP에 비해 오류가 미치는 범위를 한정시킬 수 있는 장점이 있다. DP의 경우는 (그림 3)에서와 같이 특정 패킷의 오류가 프레임 전체에 고루 전파된다.

#### 4.1 BDP 기법의 예

(그림 5)는 3단계 변환된 프레임의 패킷화 예를 보이고 있다. 이때, 부대역의 블록 수와 패킷의 수를 16개로 정하여  $LL_3$  부대역의 모든 블록들을 서로 다른 패킷에 분산하고 있다. 패킷에 할당된 각 블록은 그 블록의 모션벡터와 웨이블릿 계수를 함께 저장한다. 그림에서  $LL_3$  부대역의 0번 블록이 0번 패킷에 할당되고, 이 블록과 이웃하지 않는 8번



(그림 5) 3단계 변환된 프레임의 패킷화 예



(그림 6) 패킷화된 패킷들의 블록 구성

블록이 중복되어 같은 패킷에 할당된다.  $HL_3$  부대역에서는  $LL_3$  부대역에서 할당된 블록(0, 8)과 이웃하지 않는 7번 블록이 삽입된다.

(그림 6)은 프레임의 블록을 패킷화한 후 0번~3번 패킷에 할당된 부대역과 그 블록번호를 보여준다.  $B(i, j)$ 는 (그림 6)의  $S[i]$ 번 부대역의  $j$ 번 블록을 의미한다.

#### 4.2 패킷화 과정

BDP 기법은 세 단계의 처리과정으로 이루어진다. 첫째, 부대역의 각 블록들이 서로 다른 패킷에 삽입되도록 패킷 수와 블록 수를 결정하는 과정이다. 이때, 블록의 총 수는 전송에 적합한 패킷의 크기, 네트워크 전송의 효율을 저해하지 않는 패킷 수 등을 고려하여 결정해야 한다. 공간-주파수 상의 이웃한 블록들이 서로 다른 패킷에 할당되도록 하기 위해 생성해야 할 패킷의 수는 웨이블릿 변환 단계  $M$ 에 영향을 받는다. 그러나  $M$ 이 너무 크면 패킷 사이즈가 줄어들어 네트워크 전송이 비효율적일 수 있으므로 본 논문에서는 식 (4.1)과 같이 패킷의 수를 정한다. 즉, 패킷 수와 부대역 당 블록수를 동일하게 하며 그 범위는  $M$ 에 의해 생성된 부대역의 수 ( $S_M = 3M + 1$ )보다는 작거나 같으며  $M-1$ 에 의해 생성된 부대역의 수 ( $S_{M-1} = 3M - 2$ )보다는 크거나 같다. 그러나 어떤 프레임을  $M$  단계 웨이블릿 분할하고  $S_{M-1}$ 개의 패킷을 생성한다면 같은 부대역에 속하면서 같은 패킷에 할당되는 블록이 발생하게 된다. 이 경우 분해 단계가 낮은 부대역의 블록( $HH_1, HL_1, LH_1, \dots$ )부터 분산에서 제외되도록 하여 품질저하의 영향을 최소화한다.

$$S_{M-1} \leq N_{packets} = N_{blocks} \leq S_M \tag{4.1}$$

$$S_M = 3M + 1, 3 \leq M \leq 5$$

둘째, 공간-주파수상의 이웃한 블록이 서로 다른 패킷에 할당되도록 분산하여 저장하는 과정이다. 패킷에 할당되는 부대역의 순서는 (그림 5)의 부대역 번호 순이며  $LL_M$  부대역을 먼저 각 패킷에 할당한다. 이때, 공간-주파수상의 이웃한 블록은 다른 패킷에 할당되도록 분산해야 한다. 부대

역 내에서  $i$ 번 블록이 할당되는 패킷 번호는  $(i-1)$ 번 블록에 의해서 결정되므로 부대역 간의 인터리빙을 위해 이웃한 부대역의 0번 블록은 서로 다른 패킷에 할당되어야 한다. 식 (4.2)는 각 부대역의 0번 블록이 어느 패킷에 할당되는지 결정하는 수식이다. 이때,  $B_{j,i}$ 는  $S[j]$  부대역의  $i$ 블록이 할당되는 패킷의 번호를 의미한다.  $M$ 은 웨이블릿 변환 단계를 의미하며,  $N$ 은 패킷의 수 및 블록의 수를 의미한다. 식 (4.2)에 의하면  $S[j]$ 번 부대역의 0번 블록은  $S[j-1]$ 번 부대역의 0번 블록이 할당된 패킷으로부터  $\lfloor N/2 \rfloor - 1$ 만큼 떨어진 곳의 패킷에 할당된다.

$$B_{j,0} = \left( \left( \left\lceil \frac{N}{2} \right\rceil - 1 \right) + B_{j-1,0} \right) \bmod N \quad (4.2)$$

$$1 \leq j \leq 3M, B_{0,0} = 0$$

식 (4.2)에 의해 각 부대역의 첫 블록이 할당될 패킷이 정해지면 부대역의 나머지 블록은 식 (4.3)에 의해 분산된다. 식 (4.2)와 마찬가지로  $i$ 번 블록은  $i-1$ 번 블록에서  $\lfloor N/2 \rfloor - 1$ 만큼 떨어진 곳의 패킷에 할당된다.

$$B_{j,i} = \left( \left( \left\lceil \frac{N}{2} \right\rceil - 1 \right) + B_{j-1,i} \right) \bmod N \quad (4.3)$$

$$0 \leq j \leq 3M, 1 \leq i \leq N-1, B_{0,0} = 0$$

마지막으로, 프레임 전체에서 가장 중요한 정보인  $LL_M$  부대역 보호를 위해  $LL_M$  부대역의 블록을 중복하는 단계이다. 중복되는 블록 ( $DB_{0,i}$ )는 식 (4.3)에서 할당되는 블록 ( $B_{0,i}$ )으로부터  $\lfloor N/2 \rfloor$ 만큼 떨어진 곳에 할당한다. 데이터 보호를 위해 중복되는 블록은 (그림 6)의 예처럼 첫  $LL_M$  블록 다음에 삽입되도록 한다.

$$DB_{0,i} = \left( \left( \left\lceil \frac{N}{2} \right\rceil + B_{0,i} \right) \bmod N \right) \quad (4.4)$$

$$0 \leq i \leq N-1, B_{0,0} = 0$$

위에서 설명한 세 단계의 처리 과정을 정리하면 (그림 7)과 같다.

```

Packetizer(Input : Frame, Block_numbers,
           Transform_level, Output : Packets)
{
    SN = 3 * Transform_level + 1;
    /* 부대역 수 */
    PN = BN = block_numbers;
    /* 부대역의 총 블록 수 = 패킷 수 */
    B = SubbandToBlocks(BN, Frame);
    /* 부대역을 블록으로 분할 */
    P = S[0] = LL[0] = 0;
    /* 부대역이 할당되는 패킷 번호 P */
}
    
```

```

Insert_block_to_packet(B[0][0], P);
/* 0번 부대역의 0번 블록을 P 패킷에 삽입 */
for (i = 1; i < BN; i++) {
    /* LL 부대역의 나머지 블록 할당 */
    LL[i] = P = (P + [PN/2] - 1) % PN;
    /* LL[i]는 블록 중복을 위해 기록 */
    Insert_block_to_packet(B[0][i], P);
}
for (j = 0; j < SN; j++)
    /* 각 부대역에 대하여 */
    if (j == 0) {
        /* LL 부대역의 중복 블록이면 */
        for (i = 0; i < BN; i++) {
            P = (LL[i] + [PN/2]) % PN;
            Insert_block_to_packet(B[0][i], P);
        }
    }
    else {
        /* 나머지 부대역에 대하여 */
        P = S[j] = (S[j-1] + [PN/2]) % PN;
        /* 첫 블록의 패킷 번호 */
        Insert_block_to_packet(B[j][0], P);
        for (i = 1; i < BN; i++) {
            P = (P + [PN/2] - 1) % PN;
            Insert_block_to_packet(B[j][i], P);
            /* 패킷에 삽입 */
        }
    }
}
    
```

(그림 7) 패킷화 알고리즘

### 5. 성능 평가

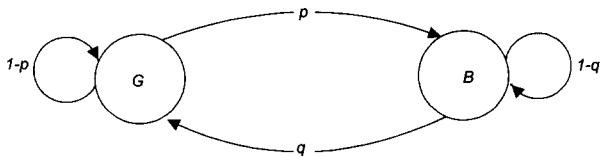
제안된 기법의 성능을 평가하기 위해 Geoff Davis[17]의 웨이블릿 이미지 압축 코덱을 프레임간 참조를 수행하는 비디오 코덱으로 수정하였다. 수정된 비디오 코덱의 움직임 예측기법으로 MRME 기법을 적용하였다. 압축된 동영상의 첫 프레임은 I 프레임으로 압축하고, 이후 프레임은 P 프레임으로 생성된다. 실험에서는 3단계 웨이블릿 변환을 수행하였으며 사용된 웨이블릿 필터는 Antonini 7/9이다. 실험에 사용되는 비디오는 QCIF(Quarter Common Intermediate Format) 포맷의 Akiyo와 Foreman 파일이며 Akiyo는 화면의 움직임 정도가 낮으며 Foreman은 움직임 정도가 높은 파일이다.

<표 4> 실험에 이용된 비디오 소스

파일명	Akiyo.qcif	Foreman.qcif
동영상 포맷	QCIF(176 × 144)	QCIF(176 × 144)
재생율	원영상	30fps
	압축영상	10fps
움직임 정도	낮음	높음
평균 PSNR	38dB	33dB

본 논문에서는 패킷이 전송되는 네트워크 환경을 무선 네

트위크로 가정하였으며 무선 네트워크에서의 오류 패턴을 생성하기 위해 길버트 오류 모델[18]을 이용하였다. (그림 8)의 길버트 모델은 “Good” 상태와 “Bad”만을 가진 2-상태 마코프 체인 모델로서 무선 네트워크에서 발생하는 비트 오류의 특징을 나타내기 위해 제안되었다. 각 상태는 일정한 BER(Bit Error Rate)를 가지며 “Bad” 상태의 BER  $e_B$ 가 “Good” 상태의 BER  $e_G$ 보다 훨씬 크다. 본 논문에서는 “Good” 상태의 에러율은 매우 낮으며 ( $e_G = 0.0001972644427729$ ), “Bad” 상태에서 에러율  $e_B = 0.1$ 이라고 가정한다. (그림 8)에서  $q$ 는 “Bad” 상태에서 “Good” 상태로 상태가 변할 확률이며,  $p$ 는 “Good” 상태에서 “Bad” 상태로 바뀔 확률이다. 본 논문에서는  $q = 0.00078125$ 로 고정하고,  $p = 10^{-4} \sim 10^{-1}$ 까지 조정하여 실험하였다.



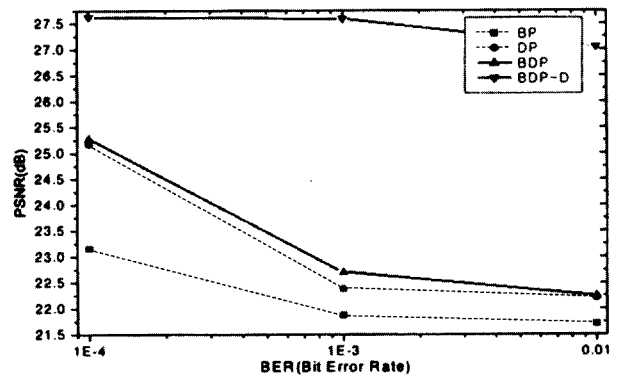
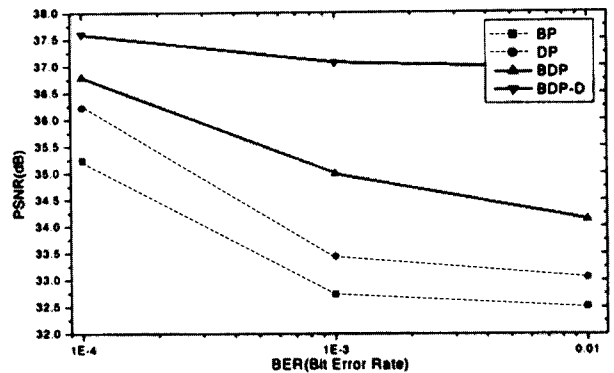
(그림 8) 길버트 오류발생 모델

BDP 기법의 성능 평가 단위로 PSNR(Peak-signal-to-noise ratio)을 사용하였다. 실험에 사용된 QCIF 파일은 4 : 2 : 0 YUV 성분으로 구성되는데, 실험에서는 휘도 성분의 평균 PSNR을 성능평가 단위로 사용한다. PSNR은 다음과 같이 정의된다.

$$PSNR[dB] = 10 \cdot \log_{10} \frac{255^2}{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (5.1)$$

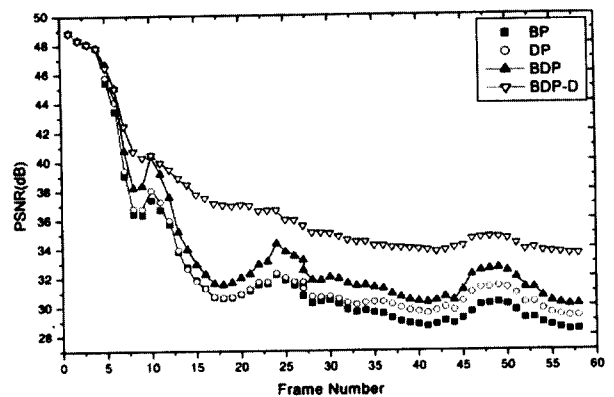
식 (5.1)에서  $x_i$ 는 압축 전 프레임의 픽셀 값이며,  $\hat{x}_i$ 는 복원된 값이다. 전체적으로 소스의 픽셀 값과 복원된 픽셀 값의 차이가 크면 PSNR은 감소한다. 이 PSNR은 복호화된 원래 이미지의 객관적인 화질 측정 단위이다.

(그림 9)는 비디오의 압축율이 2배인 경우, 비트 에러율 변화에 따른 Akiyo와 Foreman의 화질을 비교한 것이다. 안정적인 실험 결과를 얻기 위해 초기 4개 프레임에는 오류가 발생하지 않는다고 가정하였으며 프레임 당 패킷 수는 16개로 고정하였다. BDP의 경우, 패킷 내의 특정 블록에 비트 에러가 발생하면 그 블록을 모두 버렸으며 DP도 BDP의 블록에 해당하는 크기마다 동기화정보가 들어있다고 가정하였다. 실험에서 BP(Block based Packetization)는 프레임을 지정한 수의 블록으로 분할하여 순차적으로 패킷화하는 기법을 의미한다. BDP-D는  $LL_M$  부대역을 중복하고 패킷 사이즈를 줄이기 위해  $HH_1$  부대역을 패킷에 삽입하지 않는 BDP 기법을 의미한다. 움직임 보상을 수행한 BDP의  $LL_M$  부대역과  $HH_1$  부대역의 크기는 거의 동일하므로 BDP-D 기법을 이용하면 화질을 크게 향상시킬 수 있다.



(그림 9) 비트 에러율에 따른 PSNR 비교 : Akiyo(위), Foreman(아래)

(그림 9)에서 BDP는 DP에 비해 화면간 변화가 작은 Akiyo에서 성능이 뛰어나며 비트 에러율이 높아도 안정적인 성능을 보여준다. 화면간 변화가 큰 Foreman에서는 비트 에러율이 높으면 BDP의 효율이 DP와 유사하게 나타난다. 그러나, BDP-D 기법을 적용한 경우는 비트 에러율에 관계없이 2~4dB 정도 높은 PSNR값을 보여주고 있다.



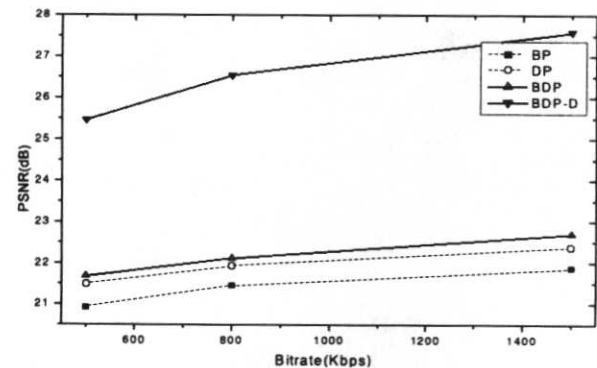
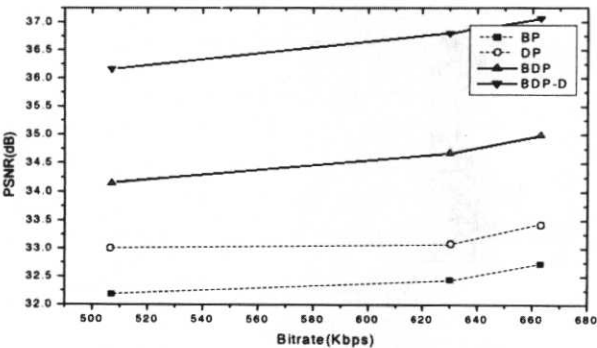
(그림 12) Akiyo 파일의 프레임 변화에 따른 PSNR 비교

(그림 10)은 비트 에러율이  $10^{-3}$ 이고 압축율이 2배인 경우. 그림에서 첫 4프레임에는 오류가 발생하지 않는다고 가의 Akiyo 비디오 내 프레임의 PSNR 변화를 나타내고 있다 정했으므로 5번째 프레임에서 급격히 화질이 감소한다.

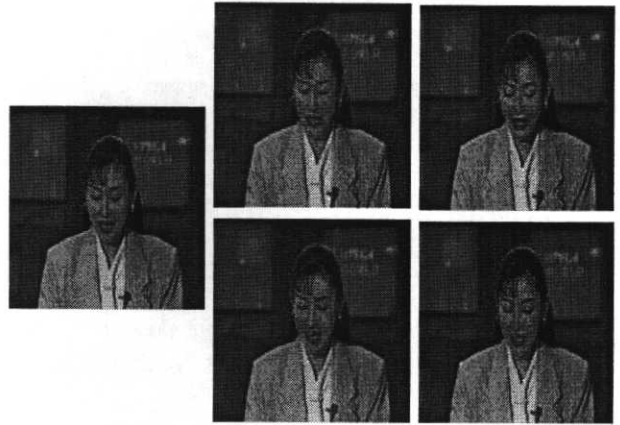
이후 프레임에서는 BDP-D의 성능이 가장 뛰어나며 BDP는 DP에 비하여 0.5dB~1dB 정도가 높음을 알 수 있다.

(그림 11)은 비트율 변화에 따른 성능을 비교한 것으로 비트 에러율이  $10^{-3}$ 일때, 압축율을 2배~6배까지 조정하여 PSNR 변화를 살펴보았다. Akiyo와 같이 화면의 변화가 작은 비디오에서는 BDP가 DP에 비하여 1dB~1.5dB 정도 높았으며, Foreman과 같이 화면의 변화가 큰 경우에는 약간의 성능 향상을 보였다. 그러나  $LL_M$  부대역을 추가하고,  $HH_1$  부대역을 제거한 BDP-D의 경우 오히려 Foreman과 같은 파일에서 더 효율적임을 알 수 있었다. 이것은  $LL_M$  부대역이 프레임 내의 다른 부대역과 이후 프레임에 미치는 영향이 매우 크다는 것을 의미한다.

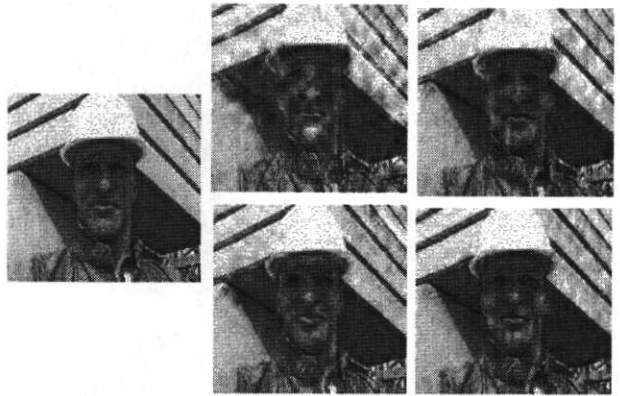
(그림 12)와 (그림 13)은 압축율이 2배, 비트 에러율  $10^{-4}$ 일 때, Akiyo와 Foreman 비디오의 시각적인 화질을 비교한 것이다. Akiyo의 경우 BP는 화면의 곳곳에 얼룩이 퍼진듯한 자국이 있으며 DP는 얼굴 주위의 얼룩과 선명하지 않은 얼굴 윤곽선을 볼 수 있다. BDP와 BDP-D는 얼굴 부분에서 약간 거친 부분을 제외하고 전체적으로 선명한 모습을 보이고 있다. Foreman의 경우 BDP-D가 가장 뛰어난 화질을 보여주고 있으며, BP와 DP는 오류 발생으로 인한 얼룩이 화면 전체에 퍼져 화면전체의 선명도를 떨어뜨리고 있다. 이 실험으로 DP는 중요부대역의 오류가 화면 전체에 퍼지고, BDP는 일부 영역에 한해 전파되므로 BDP의 성능이 우수함을 알 수 있다.



(그림 12) Bitrate 변화에 따른 성능비교 : Akiyo(위), Foreman(아래)



(그림 14) Akiyo의 20번째 프레임의 화질 비교 : 원래 이미지(맨 왼쪽), 가운데 윗줄에서 시계방향으로 BP, DP, BDP-D, BDP 순



(그림 15) Foreman의 20번째 프레임의 화질 비교 : 원래 이미지(맨 왼쪽), 가운데 윗줄에서 시계방향으로 BP, DP, BDP-D, BDP 순

## 6. 결 론

본 논문은 무선 환경에서 신뢰성있는 웨이블릿 비디오 전송을 위한 BDP(Block based Dispersive Packetization) 기법을 제안하였다. BDP 기법은 웨이블릿 변환된 프레임을 블록단위로 분할하여 패킷에 할당하도록 한다. 이때, 블록의 공간-주파수 상의 이웃한 블록은 서로 다른 블록에 할당되도록 인터리빙을 수행한다. 블록의 분산은 패킷 손실이 전체 프레임의 품질 저하에 미치는 영향을 최소화 할 수 있고, 각 패킷의 중요도를 균등하게 하여 안정적인 화상 전송을 가능하게 한다. 제안된 기법을 픽셀단위로 분산하는 DP 기법, 순차적으로 패킷화하는 BP 기법과 비교한 결과 화면의 변화 정도에 관계없이 성능이 높았으며 특히, BDP와 같은 패킷크기를 유지하며  $LL$  부대역을 중복한 BDP-D의 경우 가장 뛰어난 성능을 보였다. 본 논문에서는 오류 은닉을 적용하지 않고 실험을 진행하였는데, 향후 연구 과제로 효율적인 블록단위 오류은닉 기법을 제안하고 실험하도록 한다.



참 고 문 헌

[1] "Video coding for low bitrate communication," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.263, 1998.

[2] R. Koenen, "Overview of MPEG-4 standard," in ISO/IEC JTC1/SC29/WG11 N4668, <http://mpeg.telecomitalia.com/standards/mpeg-4/mpeg-4.htm>.

[3] Y. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," IEEE Trans. Circuits Syst. Video Technol., Vol.2, pp.285-296, Sept., 1992.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Transactions on Signal Processing, Vol.41, No.12, pp.3445-3462, Dec., 1993.

[5] Amir Said, William A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. CSVT, Vol.6, No.3, pp. 243-250, June, 1996.

[6] Y. Takashima, M. Wada, and H. Murakami, "Reversible variable length codes," IEEE Trans. Communications., Vol. 43, pp.158-162, Feb./Mar./Apr., 1995.

[7] D. W. Redmill and N. G. Kingsbury, "The EREC : An error resilient technique for coding variable-length blocks of data," IEEE Trans. Image Processing, Vol.5, pp.565-574, Apr., 1996.

[8] S. Wenger, G. Knorr, J. Ott, F. Kossentini : "Error resilience support in H.263+," IEEE Trans. on circuit and System for Video Technology, Vol.8, No.6, pp.867-877, Nov., 1998.

[9] S. Aign, "Error concealment for MPEG-2 video," Signal Recovery Techniques for Image and Video Compression and Transmission, A. K. Katsaggelos and N. P. Galatsanos, editors, ch.8, Kluwer Academic Publishers, pp.235-268, 1998.

[10] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," IEEE Trans. Image Proc., Vol.4, No.4, pp.470-477, Apr., 1995.

[11] T. Nakai and Y. Tomita : "Core Experiments on Feedback channel Operation for H.263+," ITU-T SG15 contribution LBC 96-308, Nov., 1996.

[12] Wada M. "Selective Recovery of Video Packet Loss Using Error Concealment," IEEE J. Select. Areas.

[13] A. A. Alatan, and M. Zhao, "Unequal error protection of SPIHT encoded image bitstreams," IEEE JASC, Vol.18, No. 6, pp.814-818, June, 2000.

[14] S. Cho and W. A. Pearlman, "Error Resilient Video Coding with Improved 3-D SPIHT and Error Concealment," SPIE /IS&T Electronic Imaging 2003, Proceedings SPIE, Vol. 5022, Jan., 2003.

[15] I. V. Bajic, J. W. Woods and A. M. Chaudry, "Robust transmission of packet video through dispersive packetization and error concealment," in Proc. Packet Video Workshop

(PV2000), Cagliari, Sardinia, Italy, May, 2000.

[16] Rogers and P. Cosman. "Robust wavelet zerotree image compression with fixed-length packetization," In Proc. DCC '98, pp.421-428, 1998.

[17] <http://www.geoffdavis.net>.

[18] J. Ebert, A. Willig. "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation," Technical Report, TKN-99-002, Technical University of Berlin, March, 1999.

[19] <http://www.jpeg.org/JPEG2000.html>.



이 주 경

e-mail : jklee@melon.cs.pusan.ac.kr  
 1996년 부산대학교 전자계산학과(학사)  
 1998년 부산대학교 전자계산학과(석사)  
 1998년~2001년 한국전력공사 근무  
 2001년~현재 부산대학교 대학원 컴퓨터 공학과 박사과정

관심분야 : 멀티미디어 데이터 압축, 오류제어



강 진 미

e-mail : wolf@melon.cs.pusan.ac.kr  
 2003년 부산대학교 컴퓨터공학과  
 2003년~현재 부산대학교 대학원 컴퓨터 공학과 석사과정

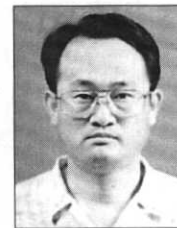
관심분야 : 멀티미디어, JPEG2000, QoS



김 충 길

e-mail : chgkim@melon.cs.pusan.ac.kr  
 1994년 부산대학교 전자계산학과(이학사)  
 1996년 부산대학교 전자계산학과(석사)  
 1998년~2000년 창신대학 전임강사  
 2001년 부산대학교 멀티미디어 협동과정 전공(박사수료)

관심분야 : 이미지 처리



정 기 동

e-mail : kdchung@melon.cs.pusan.ac.kr  
 1973년 서울대학교(학사)  
 1975년 서울대학교 대학원(석사)  
 1986년 서울대학교 대학원 계산통계학과 (이학박사)  
 1990년~1991년 MIT, South Carolina 대학 교환 교수

1995년~1997년 부산대학교 전자계산소 소장  
 1978년~현재 부산대학교 전자계산학과 교수  
 1997년~현재 부산대학교 대학원 멀티미디어 협동과정학과 교수  
 관심분야 : 병렬처리, 멀티미디어