

DirectShow를 이용한 MPEG2 비트 스트림의 비디오 효과 구현

유 원 영[†] · 김 지 향^{††} · 이 준 환^{†††}

요 약

현재 동영상 데이터를 이용한 압축영역에서의 특수효과 구현에 많은 관심이 모여지고 있다. 따라서 본 논문에서는 DCT (Discrete Cosine Transform) 압축영역에서 다양한 효과를 위해 최소한의 디코더를 개발하였으며, 이를 이용하여 Wipe, Dissolve, Zoom등의 비디오 효과등을 DCT영역에서 제안하였다. 또한 구현한 디코더와 효과들의 확장성과 이식성이 용이하도록 COM기반의 DirectShow 형식으로 각 모듈을 필터화 하였다.

Video Effect by using DirectShow in MPEG2 bit Stream

Won-Young Yoo[†] · Ji-Hyang Kim^{††} · Joon-Whoan Lee^{†††}

ABSTRACT

The special effects on the compressed MPEG domain become one of the interesting problems. In this paper, we developed minimal decoder to effect in DCT compressed domain, proposed the video effects including wipe, dissolve, and zooming. To increase the expandability and portability, the minimal decoder and the effects are implemented to filters of COM based DirectShow.

1. 서 론

현재 디지털 시대로 접어들면서 동영상 데이터를 이용한 특수효과 구현에 많은 관심이 모여지고 있다. 동영상 데이터는 저장할 데이터 베이스의 한계와 통신을 통해 제공되는 대역폭의 한계로 인해 MPEG(Moving Picture Experts Group)이나 Motion JPEG(Joint Photographic Coding Experts Group)과 같은 압축된 상태로 저장되거나 전송되어진다. 따라서 동영상 데이터를 이용한 특수효과도 MPEG이나 Motion JPEG을 이용하여 구현되고 있고 특히 MPEG을 이용한 특수효과 구현이

많이 이루어지고 있다. MPEG을 이용해서 특수효과를 구현하는 방식을 살펴보면 크게 비압축영역에서 효과를 주는 방식과 압축영역에서 효과를 주는 방식으로 나뉘어진다.

비압축영역에서 효과를 주는 방식은 기존에 많이 이용한 방식으로 화소 단위의 구현이 가능하기 때문에 많은 효과가 구현이 되어 있다[1]. 하지만 이 방식은 화소단위의 접근을 위하여 압축을 푸는 디코딩과정과 원하는 효과후 저장을 위한 재 인코딩과정이 필요하기 때문에 두 번의 DCT 과정에서 생기는 화질열화를 피할 수 없다. 이러한 단점을 보완하고자 디코딩과 재인코딩 과정을 생략하고 압축영역에서 바로 효과를 주는 방식에 더 관심이 모아지고 있다[1-4].

기본적으로 압축영역에서 효과를 주는 방식은 DCT 영역에서 효과를 주는 방식을 말하고 이 방식은 움직임 성분이 있는 프레임과 움직임 성분이 없는 프레임

* 본 논문은 부분적으로 전북대학교부설 영상정보신기술연구소로부터 게재료 지원을 받았다.
† 준 회 원 : 전북대학교 대학원 전자공학과
†† 준 회 원 : 전북대학교 대학원 정보통신학과
††† 정 회 원 : 전북대학교 전자공학과 교수
논문접수 : 2000년 3월 6일, 심사완료 : 2000년 8월 5일

을 고려하여 효과를 취야하기 때문에 구현 과정이 복잡하고 또 구현할 수 있는 효과도 다양하지 못하다. 구현과정의 복잡성을 제거하기 위해 압축영역인 VLD (Variable Length Decoding) 영역에서 효과를 구현하는 방식도 있지만 이 방식도 비디오가 가지고 있는 충분한 정보의 접근이 용이하지 못하기 때문에 구현할 수 있는 효과가 다양하지 못하다는 단점이 있다[5].

따라서 본 논문에서는 압축영역인 DCT영역에서 효과를 주되 위에서 제안된 단점들을 제거하기 위해 (그림 1)과 같은 방식을 제안하였다. 제안한 방식을 살펴보면 먼저 다양한 효과 구현을 위해 B, P 프레임을 I 프레임으로 바꾸어 모든 프레임에서 효과를 구현할 수 있는 최소한의 디코더를 개발하였고 Wipe, Dissolve, Zoom을 응용하여 몇가지 효과를 주었다. 구현한 디코더와 효과들은 확장성과 이식성을 용이하게 하기 위해 DirectShow 기반으로 모듈화시켰다[7-9]. 본 논문에서는 특수효과 후 압축효율을 위해 DCT 압축영역에서 움직임을 예측하는 최소한의 인코딩과정은 생략하였다.



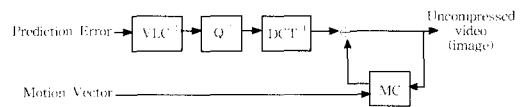
(그림 1) 전환효과를 위해 제안된 모듈

본 논문 2절에서는 제안된 특수효과 구현을 위한 최소한의 디코더 개발에 관해 기술하였고 3절에서는 압축영역에서 구현한 특수효과를 기술하였다. 4절에서는 DirectShow를 이용한 모듈화 방법을 기술하였고 5절에서는 실험한 결과 및 검토를 그리고 마지막 6절에서는 결론을 기술하였다.

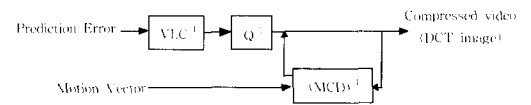
2. 제안된 특수효과 구현을 위한 최소한의 디코더 개발

(그림 2)에서와 같이 기존의 디코더는 MC(Motion Compensation)과정이 비 압축영역에서 이루어진다[6]. 따라서 기존의 디코더를 이용한 특수효과 구현방식은 움직임 보상과정이 가장 뒤에 있기 때문에 움직임 보상전의 압축영역에서 효과들을 적용하기에는 많은 문제점들이 있다. 즉 움직임 성분이 있는 차분 블록을 가진 프레임과 움직임 성분이 없는 인트라 블록을 가진 프레임을 고려하여 효과를 주어야 하기 때문에 다양한

효과를 줄 수 없다는 단점이 있다. 따라서 본 논문에서는 (그림 3)과 같이 [1]에서 제안한 방법에 따라 MC 과정을 DCT 영역에서 구현하는 MC-DCT 모듈을 이용하여 차분이 없는 DCT 압축 이미지를 복호화하는 최소한의 디코더를 개발하였다. 또한 움직임 성분이 있는 B, P 프레임은 움직임 성분이 없는 I 프레임으로 바꾸어 다양한 효과구현을 할 수 있도록 하였다.



(그림 2) 기존의 디코더

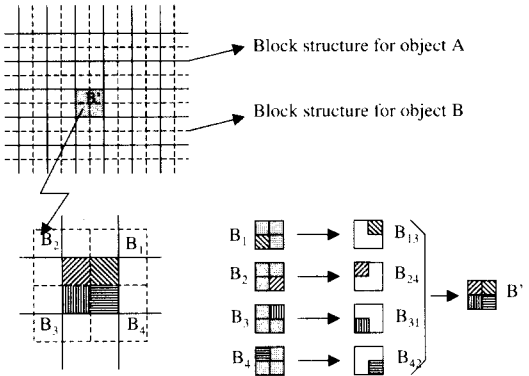


(그림 3) 제안된 최소한의 디코더

B, P 프레임을 I 프레임으로 바꾸는 과정을 살펴보면 B, P 프레임은 전 프레임과의 차분값을 가지고 있는 프레임과 Motion Vector를 적용하여 새로운 위치로 재구성한 프레임을 합하여 새로운 I 프레임으로 만들 수 있다. 이때 Motion Vector를 이용하여 새로운 블록을 구성할 때 새로운 블록이 참조한 블록과 불일치되는 문제가 발생한다. 또한 전 프레임과의 차분값만을 가지고 있는 프레임은 두가지 종류의 DCT type이 존재하게 되어 영상 복원시 문제가 생기게 된다. 따라서 본 논문에서는 첫 번째 문제는 DCT 성질을 이용한 Matrix를 이용하여 해결하였고 두 번째 문제는 필드 DCT type을 프레임 DCT type으로 바꾸므로써 해결하였다.

2.1 블록의 불일치 문제 해결 방법

(그림 4)에서 보면 점선은 참조한 전 프레임의 블록 구조이고 실선은 Motion Vector를 이용해 새롭게 구성할 블록 구조이다. 새롭게 구성할 블록 B'는 참조한 블록 B1의 왼쪽 아랫부분, B2의 오른쪽 아랫부분, B3의 오른쪽 윗부분 그리고 B4의 왼쪽 윗부분으로 이루어진다. B4 하나의 블록을 살펴보면 참조한 위치(왼쪽 윗부분)에서 Motion Vector에 의해 재구성될 위치(오른쪽 아랫부분)로 이동하기 위해서 H_1, H_2 Matrix를 이용하여 식 (1)과 같이 구성하였다.



(그림 4) 전 프레임 4개의 블록(\$B_1, B_2, B_3, B_4\$)를 이용한 새로운 블록(\$B'\$)의 구성

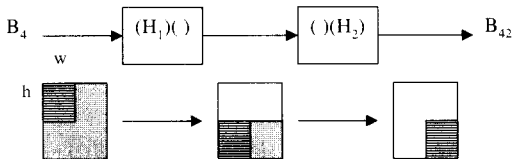
$$B_{12} = H_1 B_1 H_2, \text{ 여기서 } H_1 = \begin{bmatrix} 0 & 0 \\ I_h & 0 \end{bmatrix} H_2 = \begin{bmatrix} 0 & I_w \\ 0 & 0 \end{bmatrix} \quad (1)$$

여기서 \$I_h, I_w\$는 항등행렬이고 크기는 \$h \times h, w \times w\$이다. 위와 같은 행렬식은 DCT 성질에 의해 식 (2)와 같이 이루어지고,

$$DCT(B_{12}) = DCT(H_1) DCT(B_1) DCT(H_2) \quad (2)$$

이와 같은 방식으로 4개의 블록에 대해 적용하여 새로운 블록 \$B'\$를 구할 수 있다.

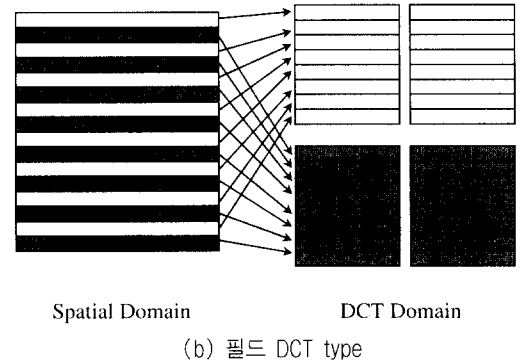
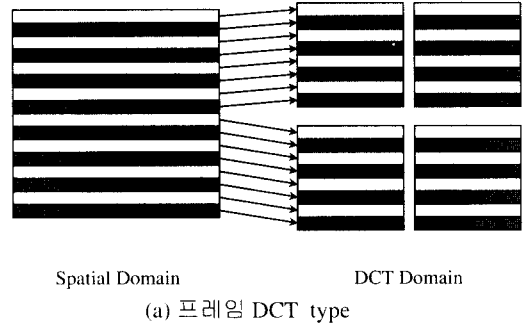
$$DCT(B') = DCT(B_{13}) + DCT(B_{24}) + DCT(B_{31}) + DCT(B_{42}) \quad (3)$$



(그림 5) Matrix를 사용하여 새로운 Subblock을 얻어내는 방법

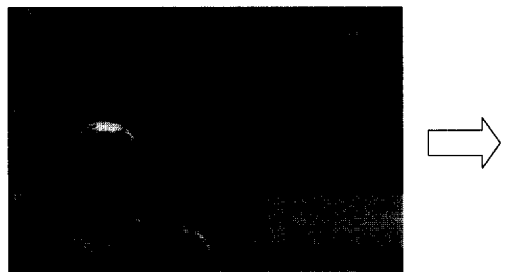
2.2 두가지 DCT type에 의한 영상복원시 발생하는 문제 해결 방법

MPEG-2에서는 필드구조에서는 DCT type이 필드 DCT 한 종류뿐이지만 프레임구조에서는 DCT type이 Macro Block 단위로 프레임 DCT와 필드 DCT 두 종류로 이루어져 있다.



(그림 6) MPEG-2에 있어서의 두 가지 DCT 모드

프레임구조에서는 Macro Block 단위로 2가지의 DCT type이 존재하기 때문에 한 프레임에서 두 가지의 DCT type이 나타날 수 있다. 한 프레임에서 두 가지의 DCT type이 존재할 경우 영상을 제안된 (그림 3)의 최소한의 디코더로 복원해보면 아래 (그림 7)의 (a)와 같이 필드 형태의 블록이 보상되지 않고 나타나므로 (그림 8)과 같은 방식으로 필드 DCT type을 프레임 DCT type으로 바꾸어 제안된 최소한의 디코더 과정을 수행하여 (그림 7)의 (b)와 같이 원래 영상의 결과를 얻을 수 있었다.

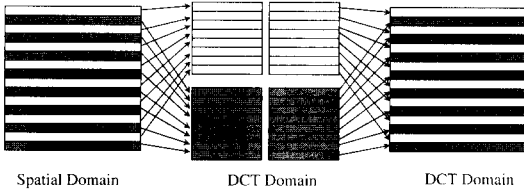


(a) 프레임 DCT와 필드 DCT 둘 다 존재



(b) 프레임 DCT 만 존재

(그림 7) 제안된 최소한의 디코더에서 블록 형태에 따른 복호 영상



(그림 8) 필드 DCT 타입에서 프레임 DCT 타입으로 변환

3. 특수효과 개발

기존에 나와있는 비압축영역에서의 효과를 살펴보면 대부분의 효과들이 Wipe, Zoom, Dissolve의 응용이다. 따라서 이러한 효과들을 압축영역에서 효과적으로 구현하기 위해서는 DCT 성질인 선형성과 직교성을 이용한 Matrix를 이용해야한다. 왜냐하면 DCT 영역은 블록단위로 이루어졌기 때문이다. 이러한 DCT 성질을 이용한 Matrix를 살펴보면 다음과 같다.

$$\text{Cut Matrix} : h_1 = \begin{bmatrix} I_h & 0 \\ 0 & 0 \end{bmatrix}, h_4 = \begin{bmatrix} 0 & 0 \\ 0 & I_h \end{bmatrix} \quad (4)$$

$$\text{Shift Matrix} : h_2 = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}, h_3 = \begin{bmatrix} 0 & 0 \\ I_h & 0 \end{bmatrix} \quad (5)$$

여기서 h_1 과 h_4 Matrix는 블록을 자르는 효과를, 그리고 h_2 와 h_3 는 블록을 쉬프트시키는 효과를 준다. 예를들어 가로방향 2 pixel Down Cut과 가로방향 2 pixel Up Shift를 살펴보면 식 (6), (7)과 같다.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

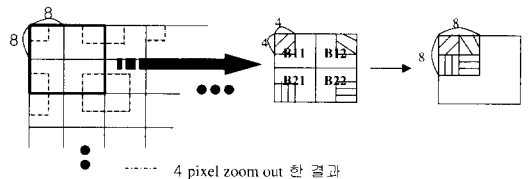
$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

이러한 Matrix들은 알고리즘의 속도를 향상시키기 위해 미리 DCT를 취하여 테이블화 하였다.

3.1 구현한 특수효과

3.1.1 Zoom과 Wipe를 응용한 특수효과

먼저 Zoom과 Wipe를 응용한 효과로 MPEG-2 비디오와 JPEG 이미지를 이용하여 효과를 구현하였다. MPEG-2 비디오의 DCT 계수를 Half Zoom Out하여 JPEG 영상에서 임의의 방향으로 이동하도록 하거나 반대로 JPEG 영상을 Half Zoom Out하여 MPEG-2 비디오에서 임의의 방향으로 이동하도록 하였다. 이때 Half Zoom Out은 (그림 9)와 같이 4개의 블록을 Zoom Matrix를 이용하여 계산하였다.



(그림 9) Half Zoom Out 구성

$$Z_1 = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

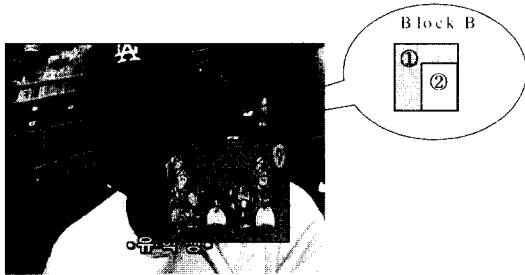
$$Z_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix} \quad (8)$$

$$\begin{aligned} DCT(Z) &= DCT(Z_1)DCT(B_{11})DCT(Z_1') \\ &+ DCT(Z_1)DCT(B_{12})DCT(Z_2') \\ &+ DCT(Z_2)DCT(B_{21})DCT(Z_1') \\ &+ DCT(Z_2)DCT(B_{22})DCT(Z_2') \end{aligned} \quad (9)$$

참고로 Zoom Out은 다른 비율로도 Zoom Out이 가능한데 예를들어 $\frac{3}{4}$ Zoom Out Matrix를 살펴보면 식 (10)과 같다.

$$Z = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3}{4} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

또한 두 영상이 겹치는 부분은 앞절에서 언급한 Cut Matrix와 Shift Matrix를 이용하여 계산하였다.



(그림 10) DCT 영역에서 두 영상이 겹치는 부분의 예

예를 들면 (그림 10)에서와 같이 두 이미지가 겹치는 부분을 확대해 보면 ①과 ②의 두 부분으로 나뉘어진다. 여기서 ①부분은 Cut Matrix를 이용하여 계산하고 ②부분은 Shift Matrix를 이용하여 계산한다. 여기서 ②부분은 디코더 개발 시 사용하였던 Matrix와 같은 방식이다.

$$\textcircled{1} : DCT(h_1) \times DCT(B) + DCT(h_1) \times DCT(B) \times DCT(h_1)$$

$$\textcircled{2} : DCT(h_3) \times DCT(B) \times DCT(h_2)$$

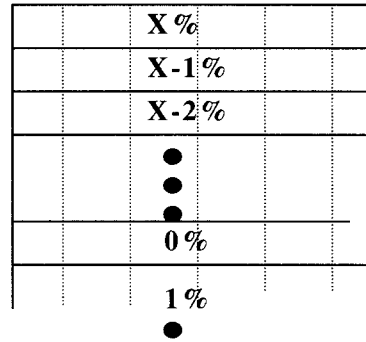
따라서 새로운 DCT 블록 B는 식 (11)과 같다.

$$DCT(B) = DCT(h_1) \times DCT(B) + DCT(h_1) \times DCT(B) \times DCT(h_1) + DCT(h_3) \times DCT(B) \times DCT(h_2) \quad (11)$$

이와 같은 방식으로 전체 Block을 계산하여 더해진 두 이미지를 얻는다.

3.1.2 Dissolve를 응용한 효과

다음에는 Dissolve를 응용한 효과로 Smog Effect라는 효과를 주었다. 이 효과는 MPEG-2 비디오에 사용자가 0~100 사이의 임의의 값 X를 입력하면 (그림 11)과 같이 계산하여 전체 영상이 안개가 낀 것 같은 효과를 주었다. 효과를 주는 방향은 가로, 세로, 대각선 방향으로 다양하게 주었고 각 DCT 블록은 화소 단위의 계산이 가능하다. 예를들어 가로 방향의 한 블록에서 두 가지의 비율을 줄 경우의 계산방식을 살펴보면 다음과 같다.



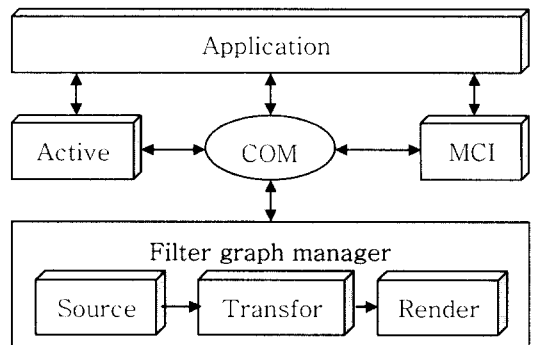
(그림 11) Smog Effect의 구성 방법

$$DCT(B) = X \times DCT(h_1) \times DCT(B) + (X-1) \times DCT(h_1) \times DCT(B) \quad (12)$$

여기서 X는 0%~100%이다.

4. DirectShow를 이용한 모듈화

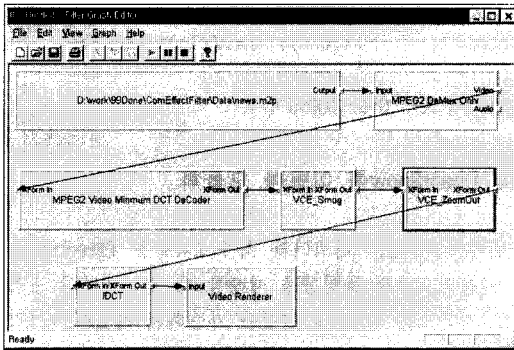
위에서 구현한 디코더와 특수효과들은 DirectShow를 이용함으로써 동영상상을 재생할 때 필터개념의 도입으로 자유로운 확장 및 연결을 가능하게 하였다. (그림 12)에서와 같이 MicroSoft사에서 제공하는 DirectShow 필터들은 Source 필터, Transform 필터, Renderer 필터로 분류되어진다. Source 필터는 파일, 네트워크 또는 다른 미디어에서 실제 데이터를 읽는 모듈이고, Transform 필터는 입력 핀에서 데이터를 받아 변형하여 다시 출력핀에 전달하는 역할을 한다. 그리고 마지막으로 Renderer 필터는 입력핀으로부터 데이터를 받아 화면에 출력하거나 저장매체에 저장하는 역할을 한다[7, 8].



(그림 12) DirectShow 구조

본 논문에서 구현한 최소한의 디코더와 특수효과들은 Transform 필터의 형식으로 개발하였다. 특수효과가 첨가된 DCT 압축 이미지를 출력하기 위하여 IDCT (Inverse Discrete Cosine Transform) 필터를 개발하여 결과를 화면에 출력하였다.

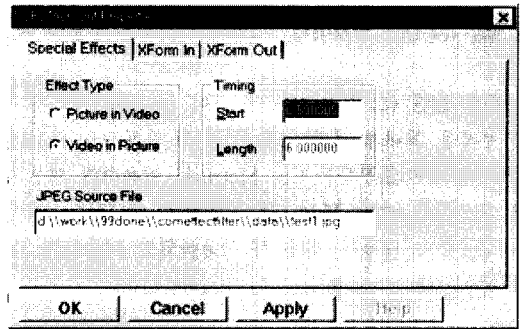
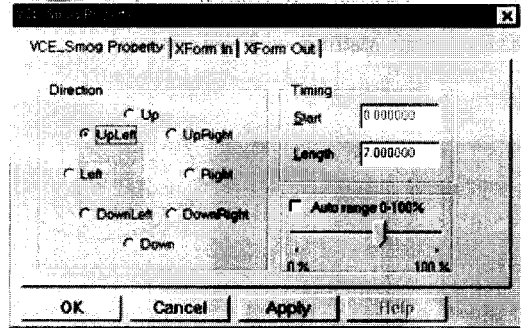
(그림 13)은 Graph Editor tool을 이용하여 개발한 필터들을 연결하여 테스트한 것이고 (그림 14)는 연결된 효과 필터들의 결과 영상을 보여주고 있다. 또한 (그림 15)에서는 구현한 특수효과 필터들의 Property Pages의 적용값들을 조절함으로써 다양한 효과 및 효과 시간을 조절하도록 하였다.



(그림 13) Graph Editor를 이용한 Filter들의 연결



(그림 14) JPEG 영상안에 Smog Effect와 Zoom Out 효과를 준 비디오 영상

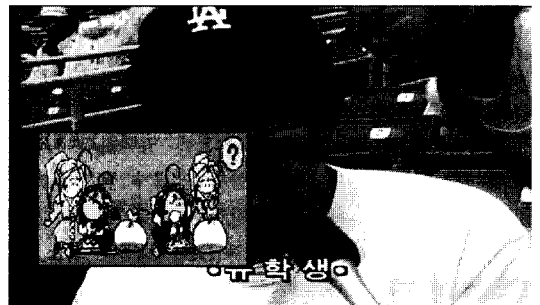


(그림 15) 특수효과 필터들에 대한 Property Page

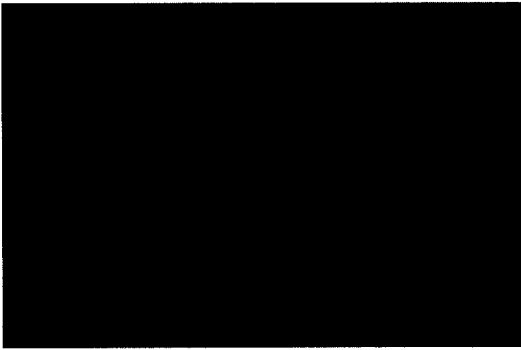
5. 결과 및 검토

특수효과를 적용하기 위해 사용된 비디오 데이터는 MPEG-2로 부호화된 704×480 해상도를 가진 KBS 스포츠 뉴스를 대상으로 하였고 Zoom과 Wipe를 응용한 효과를 위해 JPEG 영상을 이용하였다.

(그림 16)은 Wipe, Zoom을 응용한 효과이고 (그림 17)은 Dissolve를 응용한 효과이다.



(그림 16) MPEG-2 비디오와 Zoom Out 효과를 준 JPEG 영상을 합한 비디오 영상



(그림 17) Smog 효과를 준 비디오 영상

본 논문에서는 제안한 효과들 외에도 앞에서 제안한 Matrix를 이용하여 다른 효과들도 간단하게 구현할 수 있다. 예를 들어 Wipe는 두 영상이 겹치는 부분을 Cut Matrix를 사용하여 구현할 수 있고 Dissolve는 두 영상의 비율을 달리 하여 구현할 수 있다. 또한 Adobe사의 Premiere등에서 보여지고 있는 Centerpeel(중이가 중앙에서 네 방향으로 벗겨지는 효과)과 같은 효과도 중이가 벗겨지는 모양의 영상을 이용하여 구현할 수 있다. 이렇듯 본 논문에서 제안한 최소한의 디코더와 효과 구현을 위한 Matrix를 이용하면 다양한 효과를 구현할 수 있다.

6. 결 론

본 논문에서는 최소한의 디코더를 개발하여 몇 가지 특수효과를 구현하였다. 최소한의 디코더 개발시 문제가 되었던 블록의 불일치문제와 DCT type에 따라 발생하는 복호시 문제점을 제안한 방식들에 따라 해결하였고 효과 구현시 효과의 짐을 높이기 위해 여러 가지 Matrix를 제안하여 효과를 구현하였다. 또한 이 방식을 사용하여 기존에 제안되었던 압축영역에서의 특수 효과 구현방식이 가지고 있었던 문제점인 다양한 효과 구현이 어렵다는 제약점을 해결하였다.

본 논문에서는 Motion Vector와 컬러를 배제한 효과를 구현하였으나 앞으로는 Motion Vector와 컬러를 이용한 효과를 연구할 것이다. 또한 압축의 효율을 높이기 위해서 DCT 압축이미지를 MPEG 비트 스트림으로 변환하는 최소한의 부호화기도 개발할 예정이다.

참 고 문 헌

- [1] S.-F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," IEEE Journal of Selected Areas in Communications, Special Issue on Intelligent Signal Processing, pp.1-11, Jan. 1995.
- [2] Bo Shen and Ishwar K. Sethi, "Block-Based Manipulations on Transform - Compressed Images and Videos," accepted by Multimedia Systems Journal.
- [3] Jianhao Meng and Shih-Fu Chang "CVEPS-A Compressed Video Editing and Parsing System," ACM Multimedia 96 Conference Boston, 1996.
- [4] Bo Shen and Ishwar K. Sethi, "Inter-Block Operations on Compressed Images," Proc. ACM Intl. Conf. Multimedia95. pp.490-499, Nov. 1995.
- [5] 박성일, 박동권, 원치선, "MPEG 압축 영역에서의 시각 특수 효과 구현", 제12회 신호처리 합동학술대회, 제12권 제1호, pp.753-756, 1999.
- [6] ISO-IEC13812-1/ISO-IEC13812-2 International Standards.
- [7] DirectX Media 6.0 at <http://www.microsoft.com/directx/homeuser>.
- [8] D. Rogerson, Inside COM, MicroSoft Press, 1997.
- [9] D. Chappell, Understanding ActiveX and OLE, MicroSoft Press, 1996.
- [10] Tim Wittenburg, Visual Special Effects Toolket in C++, Wiley Computer Publishing, 1997.



유 원 영

e-mail : zero2@iceng.chonbuk.ac.kr

1996년 전북대학교 전자공학과 졸업(학사)

1998년 전북대학교 대학원 영상정보공학과 졸업(공학석사)

1998년~현재 전북대학교 대학원 전자공학과 박사과정
관심분야 : 비디오 분석 및 편집, 영상처리



김 지 향

e-mail : hyang11@hanmail.net

1999년 전북대학교 전자공학과
졸업(학사)

1999년~현재 전북대학교 대학원
정보통신학과 석사과정

관심분야 : 비디오 분석 및 편집, 영상처리



이 준 환

e-mail : chlee@moak.chonbuk.ac.kr

1980년 한양대학교 전자공학과
졸업(학사)

1982년 한국과학기술원 전자
공학과 졸업(공학석사)

1982년~1985년 전북대학교 전자
공학과 조교

1985년~1987년 전북대학교 전자공학과 전임강사

1990년 미주리대 전산학과 졸업(공학박사)

1990년~1994년 전북대학교 전자공학과 조교수

1994년~1999년 전북대학교 전자공학과 부교수

1999년~현재 전북대학교 전자공학과 정교수

관심분야 : 영상처리 및 분석, 인공지능