

# 압축된 이진 동영상 기반의 GUI에 대한 연구

정재훈, 김대중, 김운경  
 고려대학교 멀티미디어 정보통신 연구실  
 전화 : (02) 3290-3690

## On compressed binary video-based GUI

Jae Hun Jeong, Dae Jung Kim, Woonkyung M. Kim  
 Multimedia Information Communication Lab(MIC Lab). Korea University  
 E-mail : jhjung@korea.ac.kr

### Abstract

멀티미디어의 시대가 도래함에 따라 셀 수도 없이 많은 영상매체들이 범람하고 있다. 과거에는 영상 매체들의 수는 셀 수 있는 대상으로 분류되는 소수의 보조 매체이었던 것에 반해 지금의 영상매체 특히 동영상을 이용한 전달방식은 과거의 인간 인지의 보조수단이 아닌 인지의 주류로 떠오르고 있는 실정이다. 이에 이 수많은 영상매체를 얼마나 빠르게 인지할 수 있는가가 중요한 관점으로 떠오르고 있다.

### I. 서론

새로운 미디어 기술에 기인하여 다양한 동영상 콘텐츠가 생성되고 또한 데이터의 양도 점차 증가함에 따라 효과적인 동영상 관리의 필요성이 요구되고 있다. 방대한 양의 영상이나 비디오 데이터에 대한 효율적인 검색 및 관리를 위해서는 먼저 동영상 데이터를 사용자가 쉽게 처리할 수 있으며, 빠르게 인식되어야 한다. 이를 위해서는 기본적으로 원래의 영상에 있어서 전체적인 미리보기 및 대략적인 모양에 관련된 특징을 제공할 수 있는 이진 영상을 사용한다. 하지만 이러한 이진 영상은 제한된 채널, thin client 환경에 유용하기 위해서는 이진 동영상 압축이 병행되어야 할 것이다. 따라서 본 논문에서는 이러한 이진 영상에 있어서의 압축 기법과 이를 기반으로 기존의 방법과 차별화된 사용자 GUI와 이진 동영상 데이터의 효율적인 파일 관리를 위한 Meta data File Structure에 대해 살펴보고자 한다.

### II. Sliced Binary Video Generating System

이진영상의 간단하고 간결한 특성 때문에, 전체적인 original video contents에 대한 빠른 미리 보기 및 모양에 관련된 정보를 제공할 수 있다. 이러한 이유로 이진 동영상은 여러 application에 활용 가능하며, 다양한 가능성을 제시한다. MPEG-1, 2 파일로부터 휘도(Y) 영상 신호를 추출하여, binary video data 만드는 데에 있어서 아래의 그림과 수식을 통하여 수행된다.

$$S_n(i, j) = T_L [G_n(i, j)], \quad L = 256/2^R$$

$L$ : Level magnitude,  $R$ : bit/pixel

$G_n(i, j)$ :  $ij$ -th gray-scale pixel value of  $n$ -th frame

$S_n(i, j)$ :  $ij$ -th pixel value of BV clip's  $n$ -th frame

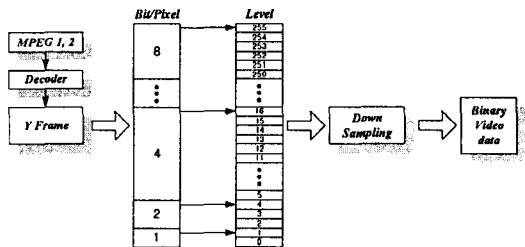


Figure 1: SBV Generating System

256레벨의 영상 신호로부터 적절한 임계값(threshold value)과 pixel당 bit의 수를 결정 한 다음 사용자가 원하는 영상의 사이즈를 결정하여, 이진 영상을 구성할 수 있다. 이때 임계값(threshold value)은 때

프레임 마다 히스토그램의 특성이 다르기 때문에 다른 값을 가지게 된다. 만약 이러한 부분을 고려하지 않고 고정된 임의의 임계값을 사용하게 되면 이진화 하는 과정에서 데이터의 중요한 특징이 사라질 가능성이 있다. 일반적인 비디오 데이터인 경우에는 이러한 방법으로 이진 영상을 구성할 수 있지만, 움직임의 거의 없는 감시 카메라와 같은 경우에는 매 프레임마다 중복되는 부분을 모두 보여주기 보다는 아래의 수식과 같은 방법으로 중복되는 background 부분은 제거하고, 움직임이 발생하는 부분으로만 이진 영상을 구성하였다

$$F_k(i, j) = S_k(i, j) \oplus B(i, j) \\ = [B(i, j) \oplus F_k(i, j)] \oplus B(i, j)$$

$\oplus$  : denotes exclusive-or, XOR, operation

$B(i, j)$  :  $ij$ -th pixel value of background frame

$S_k(i, j)$  :  $ij$ -th pixel value of SBV clip's  $k$ -th frame

### III. 이진 동영상 압축

이진영상을 제한된 채널, thin client 환경에 유용하게 하기 위해서는 효율적으로 압축 되어야 할 것이다. Bi-level Video Codec을 사용하였으며, 무손실 코딩 기법으로써 BV Encoder 와 BV Decoder로 구성되어 있다. 대부분의 이진 동영상 비디오는 두 프레임간의 상관도가 매우 높기 때문에 원본 화면만 압축하는 것 보다는 이진 영상의 특징에 기초한 시간적인 중복되는 부분을 제거하는 전 처리 과정을 통하여 압축을 수행하는 것이 효과적이다.

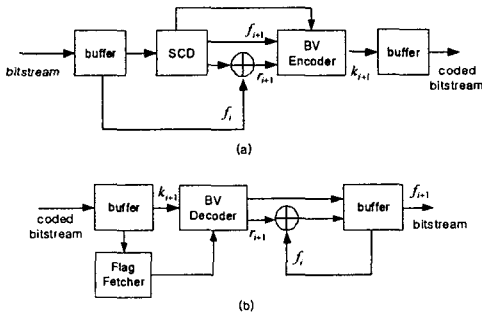


Figure 2: Simplified Block Diagram of BVC

프레임 기반으로 압축을 수행함에 있어서 프레임 종류는 Intra, Inter frame으로 구분 할 수 있다. frame을 구분하는 방법으로는 장면 전환 검출 알고리즘을

용용한 것으로 매 frame마다 아래의 수식과 같이 frame간(현재 frame과 이전 frame) pixel값의 차이를 모두 더한 후 임의의 결정된 threshold 값을 벗어나는 경우 현재 frame을 Intra frame으로 하고 아닌 경우는 Inter frame으로 한다.

$$D_{k,i} = T \left[ \sum_{i,j} |V_k(i, j) - V_{k-1}(i, j)| \right] \quad \text{for } (V_k)_{i=0}^{N-1}$$

$T$ : Threshold,  $D$ : Detection frame

Table 1. 예서는 BVE가 다른 이진 동영상 압축 코덱(JBIGE, MBIE)과의 성능을 비교한 simulation 결과 Table 이다. BVE가 JBIG 보다는 1.36배, MBIE 보다는 1.63배의 압축 효과가 있다는 것을 알 수 있다.

| Filename<br>[#Frames]     | Resolution | Raw<br>[Bytes]       | JBIGE<br>(Raw:<br>JBIGE) | MBIE<br>(Raw:<br>MBIE) | BVE<br>(Raw:<br>BVE) |
|---------------------------|------------|----------------------|--------------------------|------------------------|----------------------|
| Caltrain<br>(33)          | 512x400    | 844,800              | 122,106<br>(6.919:1)     | 157,507<br>(5.364:1)   | 105,404<br>(8.014:1) |
| Carphone<br>(100)         | 176x144    | 316,800              | 55,727<br>(5.685:1)      | 63,579<br>(4.983:1)    | 40,112<br>(7.898:1)  |
| Claire<br>(61)            | 360x288    | 790,560              | 30,068<br>(26.292:1)     | 39,227<br>(20.153:1)   | 18,038<br>(43.827:1) |
| Foreman<br>(200)          | 176x144    | 633,600              | 110,333<br>(5.743:1)     | 125,228<br>(5.060:1)   | 80,932<br>(7.829:1)  |
| Salesman<br>(128)         | 360x288    | 1,382,400            | 372,807<br>(3.708:1)     | 457,132<br>(3.024:1)   | 266,121<br>(5.195:1) |
| Total<br>[Bytes]3,968,160 |            | 691,041<br>(5.742:1) | 842,673<br>(4.709:1)     | 510,607<br>(7.771:1)   |                      |

Table 1. Comparison of Compression Ratios for Various Bi-Level Videos

### IV. GUI Design 및 Software Simulation

수 없이 많은 영상매체를 기존의 순차적인 방식으로 검색하고 인지하는 것은 시간적인 제약이 따르는 것은 당연하다고 볼 수 있다. 기존의 동영상 검색 시스템에서 제안된 monitoring method의 접근 방법은 user가 color/gray scale의 video data를 이용하여 시간상의 browsing 속도의 향상을 추구하는 방법과 machine을 이용하여 user의 요구에 알맞은 algorithm으로 motion을 detection하여 monitoring하는 방법들이 존재한다. 하지만 이러한 방법은 시간에 따른 순차적인 흐름에 따라 frame을 user에게 일방적으로 보여주

는 방식을 택하고 있으므로 user는 수동적인 입장으로 일반적인 display만을 보게 된다. 따라서 기존의 순차적인 방식으로 검색하고 인지하는 것은 시간적인 제약이 따르게 된다. 본 논문에서는 이러한 한계를 극복할 수 있으며, 기존의 time cost를 획기적으로 줄일 수 있는 방법으로 동영상 clip에 대한 3D object를 사용하였다. 이러한 방법을 사용하게 되면 user는 시간과 1차원 공간의 흐름이 아닌 단순한 1차원 상에서 모든 frame의 흐름을 한눈에 파악할 수 있게 된다. 이진 영상의 frame 들을 3차원 공간에 있어서 어떠한 object로 보다 효과적으로 표현하기 위해서는 binary data를 적절한 데이터 타입으로 변형되어야 한다. 만약 이러한 데이터의 조작이 없이 3차원 공간에 각 frame의 pixel 정보를 렌더링 하게 되면 임의의 모양이 형성 되지만, 사용자가 보기에 다소 불편하고 3D object의 특징이 부각 되지 않아 이진 동영상의 대략적인 모습을 파악하기 힘들다. 따라서 각각의 pixel에 데이터가 없는 경우( $S_{i,j}^n = 0$ )는 skip\_pixel로 분류되고, 데이터가 있는 경우( $S_{i,j}^n = 1$ )는 아래의 수식을 통해서 draw\_pixel, hidden\_pixel로 구분( $R_{i,j}^n$ 의 값이 '1' 이면 draw\_pixel, '0' 이면 hidden\_pixel로 구분)된다.

$$R_{i,j}^n = (S_{i,j}^{n-1} \cdot S_{i,j}^{n+1} \cdot S_{i-1,j}^n \cdot S_{i+1,j}^n \cdot S_{i,j-1}^n \cdot S_{i,j+1}^n)$$

(\* denotes AND operation)

이와 같이 구분된 pixel 정보를 통해서 draw\_pixel에 해당하는 부분만 렌더링 하게 되면 전체적인 이진 영상 frame들의 특징을 반영하는 3D object를 구성할 수 있다

Figure 3(a):



Figure 3(b):



Figure 3(c):

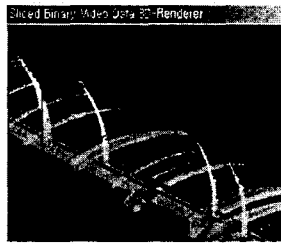
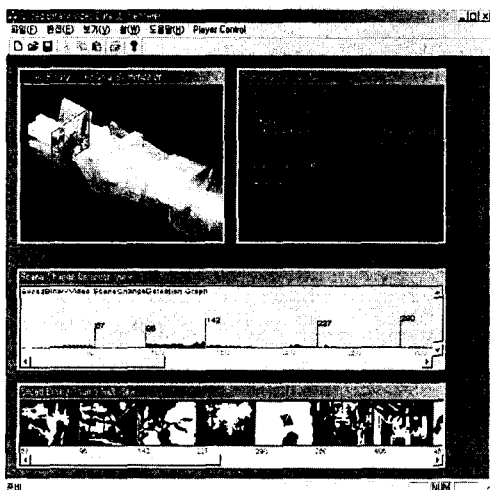


Figure 3(d):



## V. Meta data File Structure

일반적인 작업환경의 저장방식의 경우 한 개의 파일을 처리한 후 그에 대한 workspace file 하나를 생성하는 방식을 사용하고 있다. 이것은 하나의 video content에 대한 처리결과가 한 가지 방식으로만 표현되기 때문이다. 하지만 video content들은 원본이 되는 video에 대해 한 가지 데이터만 존재하는 것이 아니라 수 없이 많은 다양한 형태의 파일로써 임의의 위치에 존재할 수 있다. 예를 들면 원본 비디오 데이터에 대한 동일한 binary data라고 할지라도 frame rate, 이미지 사이즈 및 segmentation 등의 정보가 각기 다를 수 있다. 따라서 meta data에 대한 정보가 제 각각 다르기 때문에 이러한 원본 데이터와 관련 있는 content들을 적절한 틀 속에서 관리할 필요가 있다. 이에 본

논문에서는 원본 데이터와 관련이 있는 각각의 파일들을 사용자의 판단에 따라 저장하는 방식을 제안한다. 원본 비디오에 대한 다양한 파일을 관리하는 툴로서 SMV 파일 format을 사용한다. SMV는 각각의 파일에 대한 데이터 종류(file\_type), 이름(file\_name), 위치정보(data\_LS)의 3가지 정보를 Table 형태로 유지하고 있다. 특히 data\_LS는 여러 개의 데이터의 위치에 대한 정보를 가지고 있는 테이블에 대한 포인터를 가지고 있다. 이러한 이유는 어떠한 위치에 데이터가 존재 했지만, 없는 경우 다른 곳에 데이터를 검색 가능하게 하기 위해서이다. SMV 파일과 관련이 있는 어떠한 파일의 헤더에는 세그먼트별로 digital video의 meta data로 사용되고 있는 영상의 이름, size, frame rate, 시작과 끝 frame number, scene change 처리 결과 값이 하나의 block으로 있다.

## VI. 결 론

본 논문에서는 보다 효율적으로 비디오 데이터를 검색하고, 관리할 수 있는 압축된 이진 동영상 기반의 GUI를 제시하였다. 기존의 동영상 검색 시스템에 비해 보다 사용자에게 편리하고 직관적이며, 다양한 분야(stand-alone PC, PDA, 감시카메라 software, etc)에서의 활용 가능할 것으로 기대된다. 또한 Meta data file structure을 사용함으로써 연관성 있는 data간의 link를 가능하게 함으로써 동일한 정보에 대한 여러 가지의 표현의 결과를 한 파일로 묶어 줄 수 있는 기능을 구현함으로써 기존의 정보에 대한 재처리 결과와 새로이 추가되는 정보의 종합적 분석 및 확인이 가능하게 되었다.

### 감사의 글

본 연구는 (주)소프트멀티미디의 지원 하에 수행되었음을 밝히는 바입니다.

### 참고문헌

- [1] 노정호, 이기훈, 장인석, 김운경 "Video indexing을 위한 Scene Change Detector 구현" 대한전자공학회/추계학술대회(컴퓨터소사이어티), Vol. 24, No. 2, pp107-110
- [2] 김선근, 권병용, 김혁만, 송문호, 김운경

"Morphological Approach to Scene Change Detection" 한국통신학회 추계종합학술대회 Vol. 18, No. 2, pp. 1337-1340

- [3] Minerva M. Yeung, Boon-Lock Yeo "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content" IEEE Transactions on circuits and systems for video technology, Vol.7 No. 5, October 1997, pp. 771-785
- [4] 양수영 "Accelerated video를 이용한 multimedia delivery 설계" 대한전자공학회/하계종합학술대회, Vol. 24, No. 1, p223-226, June 2001