DBpia



# 단순 컴퓨터 셀 애니메이션 영상에 효율적인 압축 알고리듬

정회원 민 병 석*, 정 제 창**, 최 병 욱**

# An Efficient Compression Algorithm for Simple Computer Cell Animation

Byong-seok Min*, Je-chang Jeong*, Byung-uk Choi* *Regular Members*

요 약

본 논문에서는 비교적 간단한 컴퓨터 셀 애니메이션 동영상을 대상으로 낮은 비트율에서 효율적인 압축 알고리듬을 제안한다. 제안하는 알고리듬은 인트라 프레임 부호화와 인터 프레임 부호화를 기본 구조로 한다. 인트라 프레임 부호화에서는 팔레트를 이용한 칼라 양자화, 색인 재배치, JPEG-LS의 ADPCM 및 매핑, 분류화 및 블록단위의 색인에 대한 엔트로피 부호화로 구성되고, 인터 프레임 부호화에서는 움직임의 특성을 분류하여 블록 단위의 움직임 리플레니쉬먼트 기법을 이용해서 부호화한다. 제안하는 알고리듬은 기존의 방식인 Flash, FLC, Motion-JPEG, MPEG-1 및 MPEG-4 등과 낮은 비트율에서 부호화 성능을 비교해 볼 때 우수한 성능을 나타내었다.

ABSTRACT

In this paper, we propose an efficient algorithm to compress simple computer cell animation at very low bit rate. The structure of proposed algorithm consists of intra frame coding and inter frame coding. In inter frame coding, animation is encoded by color quantization using a palette, rearrangement of index, ADPCM used in JPEG-LS, mapping, classification, and entropy coding. In interframe coding, classifying the characteristics of motion, animation is encoded by block based motion replenishment. Experimental results show that the proposed methods turns out to outperform conventional methods including Flash, FLC, Motion-JPEG, MPEG-1, and MPEG-4.

## Ⅰ. 서 론

최근 애니메이션은 영화, TV 광고, 교육, 게임 및 오락 등 다양한 산업계에서 활발히 사용되고 있어 그 부가가치가 매우 높게 평가되고 있다. 또한, 개인용 컴퓨터와 인터넷의 대중적인 보급으로 애니메이션의 오프라인(off-line) 및 온라인(on-line) 서비스가 확산되고 있다. 애니메이션은 동영상 형태의 정보로서 디지털화해서 저장 및 통신 서비스를 할 때, 정보량이 방대하므로 동영상을 압축하는 것이 필수적이다. 현재, 디지털 영상을 압축하기 위한 다양한 방법이 개발되어 왔으나, 대부분 일반적인 자연 영상을 대상으로 개발된 것들이다. 변환 부호화를 기반으로 하는 방식으로는 산업계에서 널리 사용하는 Motion-JPEG과 국제 표준인 MPEG-1, MPEG-2 및 MPEG-4 등이 있다. Motion-JPEG은 움직임 정보를 처리하지 않고 각 프레임을 JPEG으로 압축 부호화 한 것으로 부호화 및 복호화 속도는 대단히 빠르나 압축률이 높지 않다. MPEG-1은 비디오 시디(Video CD)와 같은 저장 매체용으로 개발되었으나, 개인용 컴퓨터 상에서 소프트웨어적으로 실시간 부호화 및 복호화가 가능해서 인터넷 상에서 널리 이용되고 있으나, 낮은 비트율(low bit

* 한양대학교 전자통신공학과
** 한양대학교 전기전자컴퓨터학부
논문번호 : 010321-1106, 접수일자 : 2001년 11월 6일

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

rates)에서 블록킹(blocking) 현상 및 색상의 변화가 크게 발생한다. MPEG-2는 저장 매체인 디브이디(DVD)와 디지털 방송에 대응하도록 개발되었으나 복잡도가 매우 높아 개인용 컴퓨터 상에서 소프트웨어만으로 구현되고 있지 않다. 최근 완성된 MPEG-4는 다양한 응용 분야를 지원할 수 있도록 개발되어 있으나, 높은 압축률을 얻기 위해서는 복잡도가 매우 높아 개인용 컴퓨터에서 실시간 처리가 어렵다. 이러한 MPEG 계열의 방식은 자연 영상을 효율적으로 압축하기 위해, 이산 여현 변환(discrete cosine transform)을 기반으로 주파수 변환, 스칼라 양자화 및 엔트로피 부호화(entropy coding), 움직임 예측 및 보상을 핵심 알고리듬으로 하고 있어 복잡도가 상대적으로 높다[1,2,4,6,7].

국제 표준은 아니지만 웹 상에서 널리 사용되는 Flash[16]는 간단한 애니메이션을 표현할 수 있는 소프트웨어 저작도구로서 벡터 그래픽(vector graphic) 방식으로 정보를 저장하기 때문에 Flash로 애니메이션을 제작할 때는 압축률이 매우 높다. 그러나, 기존의 애니메이션 영상을 Flash 형식으로 변환하려면 각 프레임을 키 프레임(key frame) 형태로 저장해야 하므로 압축률도 높지 않고 에지(edge) 및 윤곽선의 열화가 크다. 그리고, Autodesk사의 FLC[5]는 애니메이션 저작 도구로서 매 화면 단위로 지역 팔레트(palette)를 수작업으로 지정하지만, 기존의 RGB 24비트 동영상 화일을 FLC로 변환할 때는 일반적으로 자동처리를 통해 256 색상의 전역 팔레트를 설정한다. 따라서, 동영상에서 장면(scene)이 많이 변화하는 경우, 색상의 열화가 크게 발생하고 움직임이 작은 프레임을 생략하기 때문에 움직임이 자연스럽지 못하다는 단점을 갖고 있다.

애니메이션은 제작 방식에 따라 그림 애니메이션, 모델 애니메이션, 컴퓨터 애니메이션 등으로 크게 나눌 수 있다. 셀 애니메이션은 그림 애니메이션의 범주에 포함되며 완전 애니메이션(full animation)과 제한 애니메이션(limited animation)으로 나뉘어 진다. 완전 애니메이션인 경우 초당 24 프레임(frame)이 보여지며 제한 애니메이션인 경우 초당 1~12 프레임이 보여진다. 그리고, 컴퓨터 셀 애니메이션은 그림 애니메이션과 컴퓨터 애니메이션을 혼합해서 제작하는 방식이다. 랜더링(rendering) 또는 빛 추적기법(ray tracing)과 같은 3차원 컴퓨터 그래픽을 이용해서 정교한 질감을 표현하는 애니메이션과 달리 비교적 간단한 기법으로 제작된다. 제작과정은 기획안, 원화, 동화, 컴퓨터 스캔(scan), 컴퓨터 컬러링(coloring), 컴퓨터 리터치(retouch), 컴퓨터 편집, VTR 편집, 네거필름(negative film) 녹화, 더빙(dubbing) 및 소리 효과, 작품 완성의 순서로 진행된다. 컴퓨터 셀 애니메이션에서는 배경과 같이 컴퓨터로 제작하기 곤란한 질감을 필요로 하는 것은 수 작업으로 제작하고, 주인공들과 같은 객체들은 수 작업으로 제작된 윤곽선만을 스캐너로 입력받아 컴퓨터 상에서 채색하고 있다[14].

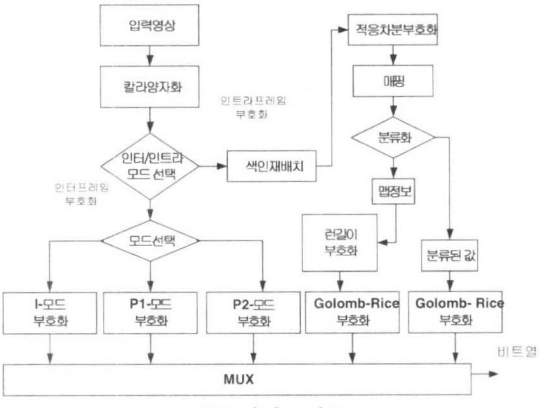
본 논문에서는 제한 애니메이션 형태의 비교적 간단한 내용으로 구성되어 있는 컴퓨터 셀 애니메이션을 부호화 대상으로 하므로, 그 특징을 영상 부호화 관점에서 보면 다음과 같다. 첫째, 인트라 프레임(intra frame)의 부호화 측면에서 고려할 때, 컴퓨터 셀 애니메이션에서 배경은 수 작업으로 제작되기 때문에 사용되는 색깔의 개수는 자연 영상 보다 매우 작아 컴퓨터 상에서 256 색깔 이내의 팔레트를 이용해서 표현하더라도 주관적 화질의 열화가 매우 작다. 주인공과 같은 객체는 윤곽선을 수 작업으로 제작하기 때문에 선명하다. 그리고, 객체들은 몇 개의 영역들로 구성되고 각 영역들은 단일 색깔을 갖는다. 따라서, 인접한 화소의 용장성이 매우 높고, 객체의 윤곽선이 선명해서 열화되지 않아야 주관적 화질이 향상된다. 둘째, 인터 프레임(inter frame) 부호화 측면에서 고려하면, 간단한 애니메이션은 배경 전환이 많지 않고 주인공과 같은 객체에서 지역적으로 움직임이 발생한다. 이러한 지역적인 움직임에 대해 MPEG 계열에서 채택하고 있는 움직임 예측 및 보상 방식으로 처리하는 것보다 블록 단위로 전후 프레임간 차이를 계산함으로써 움직임 영역을 추출하고 처리하는 것이 실시간 처리에 유리하다. 제한 애니메이션은 초당 12 프레임으로 화면에 보여지기 때문에, 움직임이 화면 전체에서 크게 발생하는 경우가 많은데, MPEG 계열에서 채택하고 있는 블록 단위 움직임 예측 및 보상 방식으로 처리하면, 낮은 비트율에서는 객체의 단순한 영역에서 블록킹 현상이 현저하게 발생한다. 이러한 경우, 인터 프레임 부호화보다 인트라 프레임 부호화가 유리하다[17].

본 논문에서 제안하는 알고리듬은 크게 인트라 프레임 부호화와 인터 프레임 부호화를 기본 구조로 한다. 인트라 프레임 부호화에서는 팔레트와 블록단위의 인덱스에 대한 부호화로 구성되고, 인터 프레임 부호화에서는 움직임의 특성을 분류하여 블

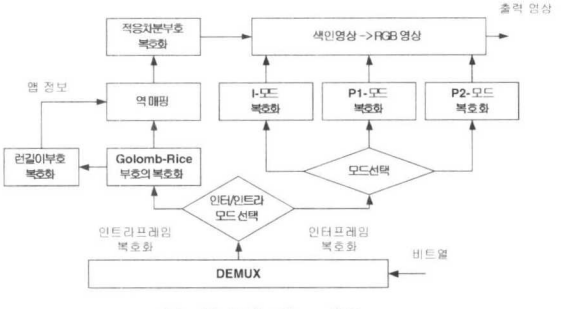
Copyright (C) 2003 NuriMedia Co., Ltd.

www.dbpia.co.kr

록 단위의 움직임 리플레니쉬먼트(replenishment) 기법을 이용해서 부호화한다. 제안하는 압축 알고리듬의 구조를 그림 1에 나타내었다. 본 논문의 구성은 다음과 같다. 2장에서 인트라 프레임 부호화를 제안하고 3장에서 인터 프레임 부호화를 제안하고 4장에서 실험 결과에 대해 기존의 방법과 비교 및 고찰을 하고 5장에서 결론을 맺는다.




(a)부호화알고리듬




(b) 복호화 알고리듬

그림 1. 제안하는 압축 알고리듬

## Ⅱ. 인트라 프레임 부호화

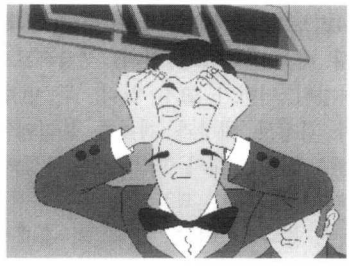
인트라 프레임 부호화는 화면 내에서 인접한 화소간의 용장성을 제거하여 정보량을 줄이는 방법이다[3,4]. 기존의 이산 여현 변환과 웨이블릿(wavelet) 변환을 기반으로 하는 변환 부호화 방법은 주파수 영역에서 양자화 및 엔트로피 부호를 사용함으로써 높은 압축 효율을 얻을 수 있지만 복잡도가 높은 단점이 있다. 본 논문에서는 애니메이션을 부호화하기 위해서 공간 영역에서의 처리를 기반으로 하는 부호화 방식을 제안한다. 화면내 부호화 측면에서 보면 애니메이션의 특징은 다음과 같다. 첫 번째, 화소당 24비트(약 1600만 칼라) 색상을 8비트(256 칼라) 색상 내지 5비트(32 칼라) 색상으로 감축해도 주관적 화질의 저하가 크지 않다. 두 번째, 애니메이션 제작 과정상 배경과 객체가 분리되어 제작되는데, 객체들을 몇 개의 영역들로 구성되며, 특정 영역내의 색상은 단일 색상으로 이루어져 있어 용장성이 매우 높다. 그리고, 객체들의 윤곽선이 뚜렷해서 윤곽선에서 작은 열화가 발생하더라도 전체적인 주관적 화질이 크게 떨어진다.

제안하는 방식은 칼라 양자화, 색인 영상의 색인 재배치, 적응 차분 부호화, 분류화, 엔트로피 부호의 단계를 갖는다. 24비트 입력 영상에 대해 색상을 감소시키기 위해서 8진 트리를 이용한 칼라 양자화를 수행하고[10,11], 색인영상에 대해 색인 재배치를 통해 색인의 변화을 줄이고, 이웃한 화소들간의 용장성을 줄이기 위해 JPEG-LS의 적응 차분 부호화를 수행한다[19]. 이때 발생하는 차분 신호들의 통계적 분포의 특성을 엔트로피 부호화기에 적절한 분포로 변환하는 맵핑을 거친 후, 전방 분류화를 통해 상위 분류(significant class)와 하위 분류(non-significant class), 그리고 분류의 위치를 나타내는 2진 정보인 맵(map) 정보를 생성한다. 이 단계에서 상위 분류와 하위 분류는 Golomb-Rice 부호화[8]를 통해 엔트로피 부호화된다. 그리고, 2진 맵 정보는 런 길이 부호화 후 Golomb-Rice 부호화된다.

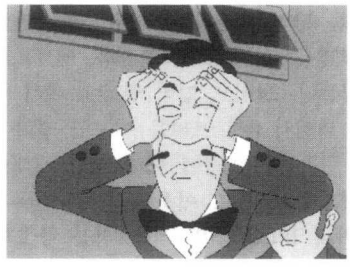
### 1. 8진 트리를 이용한 칼라 양자화

팔레트를 이용한 칼라 양자화는 컴퓨터 그래픽에서 주로 사용되는 방법으로, RGB, YUV와 Lu*v* 등의 색 좌표계 공간에서 색상의 통계적 성질에 따라, 장면 내에서 주요한 색상만으로 영상을 표현하는 방법이다. 이러한 방식으로는 메디안 컷(median cut), 시퀀셜 스칼라 양자화(sequential scalar quantization), LVQ(learning vector quantization) 신경망을 이용하는 방법과 8진 트리(octree)등이 있다. 본 논문에서는 화질이 비교적 우수하며 처리속도가 빠른 8진 트리를 이용하였다. 그림 2에 색상수 별로 칼라 양자화한 화면을 나타내고 있는데 8비트인 경우 주관적 화질면에서 24비트 화면과 거의 차이가 없으며 8비트 이하에서는 화질의 차이가 나타나고 있다. 입력 영상의 24비트 색상을 압축률에 따라 8비트 256 칼라 이하로 칼라 양자화를 수행해서 색인 영상과 칼라 테이블을 부호화한다.

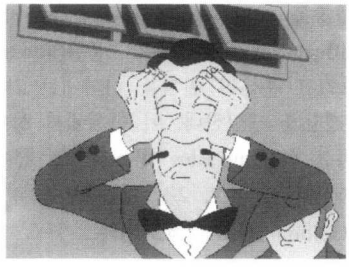
Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr



(a) 입력영상 (24비트 칼라)



(b) 입력영상 (256 칼라)



(c) 입력영상 (64 칼라)

(d) 입력영상 (32 칼라)

그림 2. 색상 수에 따른 영상

## 2. 색인 재배치

색인 재배치는 칼라 양자화를 거쳐서 나온 색인 영상과 팔레트의 압축률을 높이기 위해 색인들을 재배열하는 과정이다. 색인 영상은 원래 영상의 칼라 수보다 훨씬 더 적은 컬러 수로 제한되고 같은 색인 영상을 나타내는 화소들은 집중화되는 특성이 있다. 엔트로피 부호를 적용할 때, 동일한 색인을 갖는 영역은 압축률이 높아지는 반면, 다른 색상의 영역으로 전환할 때 많은 비트가 소모되고, 이웃하는 색인들의 차이 값이 클수록 더 많은 비트를 소모한다. 본 논문에서는 색인들의 재배치를 통해 이웃하는 화소들의 전체적인 색인의 차이를 줄임으로써 압축률을 향상시키는 것을 제안한다.

색인 재배치 알고리듬의 입력은 색인 영상 $I$이고, 출력은 색인 재배치 영상 $I'$가 된다. 이에 따라, 칼라 테이블도 다시 구성되어야 한다. 새로운 색인 영상 $I'$는 원래 색인 영상 $I$보다 더 부드러운 신호가 된다. 색인 재배치는 부호기에서 색인 영상과 칼라 테이블의 정보를 변화시키기 때문에 복호기에서는 부가적인 작업이 필요가 없다.

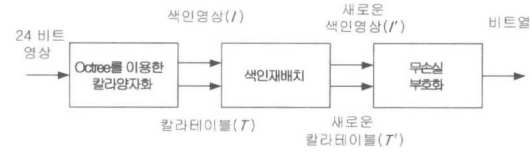



그림 3. 색인 재배치 부호화

복호기에서는 색인 영상 $I'$와 칼라 테이블 $T'$를 압축된 비트열에서 복호하고, 영상 복호에 이것들을 사용한다.

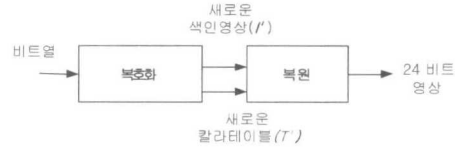



그림 4. 색인 재배치 복호화

원 색인 영상에서, 색인 값 0, 1, ... , M-1이 칼라 심벌 $S_0, S_1, \ldots, S_{M-1}$을 각각 나타낸다. 색인 재배치 알고리듬의 목적은 각각의 심벌 $S_i$을 $[0, M-1]$의 범위를 가지는 새로운 색인 값으로 1:1 매핑을 하는 것이다. 제안하는 방법은 greedy 방법으로 심벌 각각에 대해 다시 색인 재배치를 수행한다. 각각의 심벌에 대한 재 할당은 원 색인 영상의 분포와 이전까지 심벌에 재 할당한 분포에 근거하여 최적화한다. 원 색인 영상에서 얻은 확률분포는 두 개의 다른 심벌 $S_i$와 $S_j$의 cross-count $C(S_{i,S_j})$을 나타내는 테이블이다. Cross-count $C(S_{i,S_j})$는 원 색인

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

영상에서 심벌 $S_i$ 을 가진 화소가 심벌 $S_j$ 을 가진 화소에 공간적으로 인접한 횟수로 정의된다. 자주 인접한 심벌에 가까운 색인 값을 할당함으로써 압축률을 높일 수 있다. 색인 재배치 알고리즘의 단계는 다음과 같다.

단계 1. 원 색인 영상에서 각 심벌 $S_i$ , $S_j$ 쌍에 대해 cross-count $C(S_i,S_j)$를 계산한다. 각 심벌 $S_i$에 대해, 누적 cross-count $C_i = \sum_{j=0, j\neq i}^{M} C(S_i,S_j)$를 계산한다.

단계 2. 가장 큰 누적 cross-count $C_i$를 가지는 심벌 $S_{\max}$를 찾는다. 이 것을 $L_0$라고 한다. $L_0$를 집합 $P = \{ L_0 \}$에 넣는다. P는 공간적으로 정렬된 심벌로 이루어진다. $P$의 크기를 N이라 하고, N = 1로 둔다. $P$에 새로운 값들이 들어갈 때는 반드시 왼쪽 끝이나 오른쪽 끝으로만 들어갈 수 있다. 심벌이 집합 $P$에 일단 들어가게 되면 할당되었다고 지정한다.

단계 3. 할당되지 않은 새로운 심벌이 선택되고, 집합 $P$의 왼쪽 끝이나 오른쪽 끝으로 할당된다. 왼쪽 끝 위치로부터 $D_i = \sum_{j=0}^{N-1} w_{(N,j)} C(S_i,L_j)$의 위치함수를 최대화하는 할당되지 않은 심벌 $S_i$을 선택한다. $w_{(N,j)}$ 는 위치함수 $D_i$ 전체에 $C(S_i,L_j)$의 영향을 조정하는 가중치이다. $w_{(N,j)}$ 는 $L_j$와 집합 $P$의 현재 끝의 위치와의 거리에 따라 결정된다. N은 각 한 번의 반복 후에 $w_{(N,j)}$ 가 변할 수 있다는 것을 나타내기 위해 사용된다. $w_{(N,j)}$ 는 $\log(1+1/d_{(N,j)})$일 때이다. $d_{(N,j)}$는 끝의 위치와 $L_j$와의 거리이다. 여기서 선택된 심벌을 $S_{\max L}$라고 하고 집합 $P$의 오른쪽 끝에 대한 심벌을 $S_{\max R}$라고 한다.

단계 4. $S_{\max L}$ 와 $S_{\max R}$ 중에서 더 큰 $D_i$ 값을 갖는 값을 집합 $P$에서 해당하는 끝에 할당하고 이것을 $L_N$이라 하자. N = N+1로 한다.

단계 5. 만약 N < M 이면, 단계 3부터 다시 수행한다.

단계 6. 정수 0,.....,M-1을 집합 $P$의 심벌에 왼쪽에서 오른쪽 순서로 또는 오른쪽에서 왼쪽 순서로 할당한다. 재배치된 색인은 원래 신호 값 $i$ 을 $S_i$ 로 교체함으로써 얻어지는 것이다.

### 3. 적응 차분 부호화와 색인 차분 신호의 매핑

애니메이션 영상의 특성상 수 개의 색상으로 구성되어 있고 영역들이 단일 색상으로 구성되어 있으며 객체와 영역을 나타내는 윤곽선이 에지 형태로 뚜렷하다. 이러한 특성의 영상에 대해 DCT와 같은 주파수 영역에서 고주파 성분을 크게 양자화하면 윤곽선이 손상되고 윤곽선 주위로 블록킹 현상이 나타나 주관적 화질이 떨어진다. 오히려, 영상 내에서 공간상의 상관관계가 매우 높기 때문에 차분 부호화를 적용하는 것이 부호화의 효율을 높일 수 있다.

본 논문에서는 차분 부호화 방법으로 복잡도가 낮은 JPEG-LS에서의 적응 차분 부호화를 적용하였다. 여기에서 현재 화소의 값에 대한 예측을 수직성분과 수평성분의 신호로부터 현재의 화소값을 예측하므로 수평 또는 수직 화소만을 이용해서 1차원으로 처리하는 것보다는 복잡도는 증가하지만, 방향성분을 고려하고 있기 때문에 동일한 압축율에서는 화질을 향상시킬 수 있다. 그리고, 예측 차수를 높이고 수평과 수직 화소에 가중치를 부가하는 예측 부호화도 가능하지만 애니메이션 특성상 1차 차수의 예측이 적절하였다.

색인 영상을 적응 차분 부호화한 신호는 음과 양의 서포트(support)를 갖는 라플라시안 분포를 나타내므로, Golomb-Rice 부호화기를 적용하기 위해서 양의 서포트를 갖는 지수 분포에 근사하도록 매핑을 수행한다. 변환하는 방법을 식(1)에 나타내었다.

$$\begin{aligned} &\text{if } (index > 0) \\ &\quad new_index = 2 \times index - 1 \\ &else \\ &\quad new_index = -2 \times index \end{aligned} \tag{1}$$

식 (1)에서 색인의 차분 신호인 index가 갖는 범위는 -255 ~255이다. 양의 index를 홀수로 매핑하고 음의 index를 짝수로 매핑하며 0의 index는 0으

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

로 해서, new_index는 0~511의 범위를 갖게 된다. 여기서 index의 엔트로피와 new_index의 엔트로피는 동일하면서 단지 분포의 특성만 변화하므로 정보량은 증가하지 않는다[12].

### 4. 전방 분류와 엔트로피 부호화

압축 부호화에서 효율을 높이기 위한 방법으로, 엔트로피 부호화에 앞서 신호를 통계적 성질에 따라 분류하는 과정을 제안한다. 다음 절에서 설명하는 Golomb-Rice 부호화기의 전처리 과정으로 단일 문턱치를 사용한 2원 분류화를 적용하였다. 그림 5에서 제안하고 있는 전방 2원 분류화 과정을 나타내었다. 입력 신호들에 대한 통계적 특성에 따라 적절한 문턱치를 구한다. 이것을 그림 6에서 나타내었다. 문턱치를 입력 신호에 적용함으로써 입력신호는 상위 분류와 하위 분류, 그리고 이진 맵 정보(binary map image)를 발생한다.

입력 신호가 이산 무기억 표본(discrete memoryless sample)이면 분류화 과정에서 발생하는 맵정보는 신호의 엔트로피에 영향을 주지 않는다. 따라서, 입력 신호들의 정보가 2개의 분류 신호들과 맵 정보의 정보량을 모두 합한 것과 같으므로 전체 정보량의 증가는 발생하지 않는다.

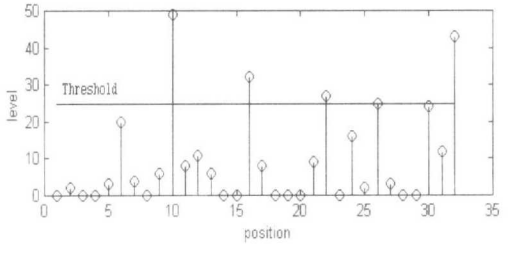



그림 5. 전방 분류화

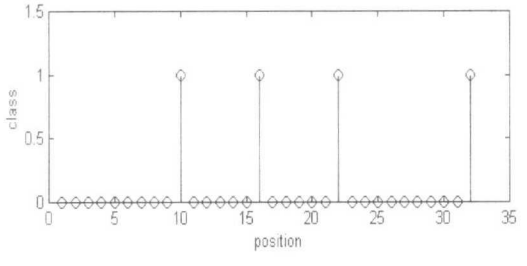



그림 6. 2원 분류 결과

색인 영상에 대해 색인 재배치, 적응 차분 부호화, 매핑과 전방 분류화를 수행한 후, 2원 분류 신호를 부호화하기 위해서 엔트로피 부호화 방법으로 Golomb-Rice 부호화를 적용하였다.

맵 정보는 0과 1이 각각 집중되어 있어 런길이 부호화를 수행하고 Golomb-Rice 부호화로 압축부호화 하였다. Golomb-Rice 부호는 산술 부호에 비해 압축 성능은 다소 낮지만, 부호화와 복호화 과정에서 사용되는 모든 연산이 계수기(counter)와 자리이동 연산에 의해 이루어지기 때문에 매우 낮은 복잡도를 갖는다. 그러므로, 실시간 복호를 필요로 하는 응용 분야에 적합하다.

## Ⅲ. 인터 프레임 부호화

일반적으로 동영상은 이전 장면과 현재 장면간에 많은 용장성을 갖고 있으며, 이러한 정보를 줄이기 위한 방법으로는 MPEG-1과 MPEG-2 등의 표준에서 채택하고 있는 움직임 추정 및 움직임 보상 등이 있다[8,9]. 움직임 추정 및 보상에서는 화소 및 블록 단위로 이전 장면으로부터 현재 화면과 가장 가까운 부분을 탐색해서 차이 성분만 전송하는 방식을 취하고 있다. 이러한 방식은 시스템 구조가 복잡하고, 우수한 화질을 얻기 위해서 탐색시간이 많이 소비되고, 움직임이 크게 발생하는 경우 정보량이 증가해서 부호화 단계에서의 단점이 되고 있다.

본 논문에서는 애니메이션 동영상의 특징을 분석해서 새로운 방식의 화면간 부호화기를 제안하고자 한다. 화면간 부호화 관점에서 일반적인 영상과 다른 애니메이션 동영상의 특징은 다음과 같다. 첫 번째, 움직임이 발생하는 장면간에서는 움직임이 매우 크다. MPEG-1과 MPEG-2에서는 초당 30화면을 재생하므로 움직임이 크게 발생하지 않지만, 애니메이션 동영상은 초당 12~15 화면을 재생하므로 움직임이 매우 크다. 두 번째, 이전 화면과 현재화면이 동일한 경우가 많다. 즉, 움직임 정보가 발생하지 않는 경우가 많다. 세 번째, 움직임이 발생하는 영역이 제한되어 있는 경우가 많다. 특정 주제 장면(scene)에서 배경은 고정되어 있고 인물이라던가 특정 객체만 비교적 움직임이 크게 발생한다.

그림 7에 나타낸 인터프레임 부호화에서는 화면의 특성상 크게 3가지 모드로 분할해서 처리한다. 첫 번째, 이전 화면과 현재 화면이 용장성이 거의 없는 경우는 I-모드로 부호화하고, 두 번째, 이전 화면과 현재 화면이 동일한 경우 P1-모드로 부호화하고, 세 번째, 이전 화면과 현재 화면에서 움직임이

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

발생한 경우 P2-모드로 부호화 함으로써 화면의 특성에 맞는 부호화 알고리듬을 적용한다.

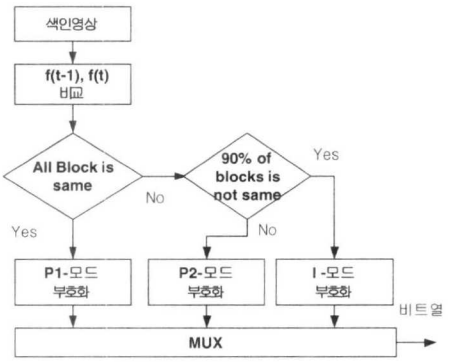



그림 7. 인터 프레임 모드

I-모드 : 이 모드는 이전 화면과 현재 화면간의 시간적 용장성이 거의 없으므로 움직임 정보를 부호화하는 것보다 인트라 모드로 부호화하는 것이 효율적인 경우이다. 일반적으로 화면이 새로운 내용으로 전환되거나 움직임이 전체 화면에서 크게 발생하는 경우에 해당된다. 앞에서 기술한 인트라 모드에서의 부호화 방법을 적용한다.

P1-모드 : 애니메이션 동영상에서는, 현재 화면이 이전 화면과 동일한 경우가 많다. 이러한 경우, 이전 화면의 헤더(header)만을 부호화 함으로써 정보량을 크게 감축시킬 수 있다. FLC나 MPEG-4에서 프레임 스킵(skip)과 유사하다. 본 논문에서 제안한 부호화 방법에서는 이러한 화면이 많을수록 압축률은 향상된다.

P2-모드 : 화면간 부호화 모드에서 움직임이 발생하는 경우에 적용하는 모드로서, 이전 화면과 현재 화면간 차이 화면의 특성이 지역적으로 발생하는 경우에 적용한다. 애니메이션 동영상은 배경의 고정되어 있고 인물과 같은 객체들만 움직이는 경우가 많다. 이러한 경우, 움직임은 전체적으로 발생하는 것이 아니라 지역적으로 발생한다. 따라서, 움직임이 있는 영역의 정보만을 부호화해서 정보량을 감축시킬 수 있다. 움직임 정보의 정확도를 높이기 위해 HSI(Hue Saturation Intensity) 색 좌표계[13]로 변환하고, 움직임이 발생한 영역을 효율적으로 탐색하기 위해서 8x8 화소 크기의 블록 단위로 처리한다. 블록의 크기를 크게 할 경우, 인물의 입 또는 눈 등에서 지역적으로 발생하는 작은 움직임으로 인한 정보량이 증가한다. 이러한 3개 값에 의한 차이가 문턱치 보다 낮으면 움직임이 없음을 0으로 나타내고, 문턱치 보다 높으면 1로 나타내어서 움직임 발생 블록 맵을 구성한다. 움직임이 발생한 블록이 90%이상 되면 I-모드로 부호화를 수행한다. 그렇지 않은 경우, 블록의 색인 정보는 인트라 모드에서와 같이 색인 재배치, 적응 차분 부호화, 매핑 및 분류화를 거쳐, Golomb-Rice 부호화하고, 분류를 나타내는 분류 맵은 런길이 부호화와 Golomb-Rice 부호화를 적용한다. 다만, 움직임 발생 블록의 위치를 나타내는 맵은 정보량이 많지 않으므로 8비트씩 묶어 저장한다.

단계 1 색인 영상을 8 × 8 화소 크기의 블록으로 전체 영상을 분할한다.

단계 2 현재 영상에 대해 블록별로 이전 영상의 같은 위치에 있는 블록과 HSI 신호를 비교해서 이전 영상과 동일하면 0의 값을 지정하고 그렇지 않으면 1의 값을 부여해서 블록 맵을 구성한다. 이것을 식 (2)에 나타내었다.

$$diff = \sum_{i=0}^{i=7}\sum_{j=0}^{j=7}( \Delta H_{m,n}(i,j)^2 + \Delta S_{m,n}(i,j)^2 + \Delta I_{m,n}(i,j)^2)$$

$$\text{if } ( diff \geq Threshold) \quad BlockMap(m,n) = 0$$
$$\text{else} \quad BlockMap(m,n) = 1 \qquad (2)$$

여기에서 각 항은 다음과 같다.

$$\Delta H_{m,n}(i,j) = H^{t}_{m,n}(i,j) - H^{t-1}_{m,n}(i,j)$$
$$\Delta S_{m,n}(i,j) = S^{t}_{m,n}(i,j) - S^{t-1}_{m,n}(i,j)$$
$$\Delta I_{m,n}(i,j) = I^{t}_{m,n}(i,j) - I^{t-1}_{m,n}(i,j)$$

단계 3 이전 화면과 비교하여 현재 화면의 모든 블록이 전부 동일하면, P1-모드로 부호화를 수행한다.

단계 4 현재 화면에서 90%이상의 블록들이 이전 화면의 블록과 같지 않을 때는 I-모드로 부호화한다.

단계 5 현재 화면에서 이전 화면의 모든 블록이 동일하지 않고, 90%이하의 블록이 동일하지 않을 경우는 P2-모드로 부호화한다.

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

## Ⅳ. 실험 결과 및 고찰

제안하는 애니메이션 동영상 부호화 알고리듬의 성능을 평가하기 위해 기존의 부호화 방법인 Flash, FLC, M-JPEG, MPEG-1, MPEG-4 등과 압축률, 평균 PSNR과 주관적 화질을 비교하고 각각의 특성을 고찰한다. 실험 영상은 애니메이션 영화 "매드독"에서 추출한 동영상을 대상으로 하였다. 영상은 320×240 화소 크기, 24비트 칼라, 화면수는 345장이고 초당 15장이다. 각각의 방식을 프레임 단위로 비교하기가 곤란함으로, 전체 비트를 대상으로 비교하였다. 압축률 $R$은 식 (3)와 같이 입력 동영상의 전체 비트로 부호화된 전체 비트를 나눈 것으로 정의한다.

$$R = \frac{total\ number\ of\ encoded\ bits}{total\ number\ of\ image\ sequence\ bits} \quad (3)$$

또한, 객관적 화질을 나타내는 기준으로 평균 $PSNR$은 식 (4)와 같이 정의해서 평가하였다.

$$PSNR = 10\log_{10}\left(\frac{255\times 255\times 3}{\sigma_n^2}\right) \quad (4)$$

$$\sigma_n^2 = \frac{\sum_i\sum_j (R(i,j)-\hat{R}(i,j))^2+(G(i,j)-\hat{G}(i,j))^2+(B(i,j)-\hat{B}(i,j))^2}{number\ of\ pixels}$$

$$PSNR_{ave} = 1/N\sum_{i=1}^{N} PSNR_{i-th\ frame}$$

여기에서 각 항은 다음과 같다.

$R(i,j), G(i,j), B(i,j)$는 입력 영상의 $(i,j)$번째 화소의 $R, G, B$ 성분이다.

$\hat{R}(i,j)\ \hat{G}(i,j), \hat{B}(i,j)$는 복호된 영상의 $(i,j)$번째 화소의 $R, G, B$ 성분이다.

$number\ of\ pixels$은 한 화면의 화소 수이다.

$N$은 입력 영상의 화면수이다.

본 논문에서 제안한 방법으로 "매드독" 동영상을 압축 부호화한 압축률은 표 1과 같다. 표 1에서 나타낸 것과 색인의 재배치가 있는 경우 평균 $PSNR$이 미세하게 차이는 있으나 색상의 수가 작을수록 색인 재배치의 효과가 크게 영향을 주어 전체적으로 볼 때, 압축률이 향상되는 결과를 얻었다.

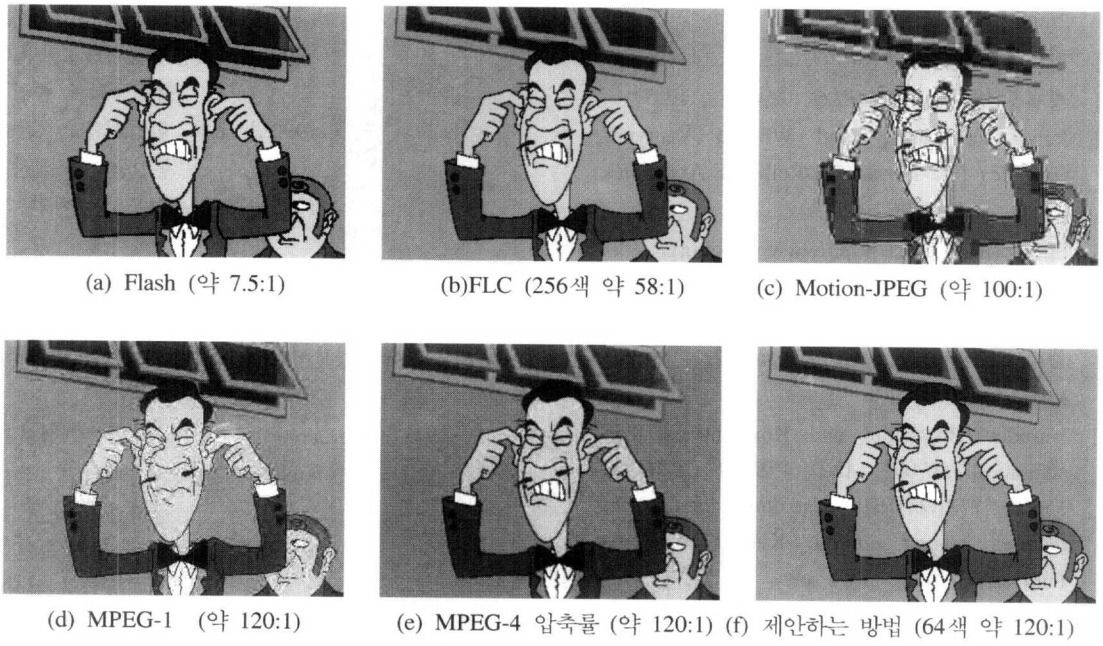
표 1. 압축률과 평균 PSNR

| | | 매드독 영상 | |
|---|---|---|---|
| | | 압축률 | $PSNR_{avg}$ |
| 16색 | 색인 재배치 없음 | 130.6 : 1 | 31.0 |
| | 색인 재배치 | 145.4 : 1 | 31.0 |
| 32색 | 색인 재배치 없음 | 120.7 : 1 | 37.2 |
| | 색인 재배치 | 132.5 : 1 | 37.2 |
| 64색 | 색인 재배치 없음 | 113.1 : 1 | 39.3 |
| | 색인 재배치 | 121.3 : 1 | 39.2 |
| 128색 | 색인 재배치 없음 | 101.6 : 1 | 39.7 |
| | 색인 재배치 | 110.6 : 1 | 39.7 |

제안하는 방법과 다양한 부호화 방식으로 압축한 결과를 그림 8에 나타내었다. 비교하는 부호화 방식은 마이크로소프트사의 윈도우 98 운영체제에 등록되어 있는 비디오 코덱(video codec)과 virtual dub 1.3[15]을 사용하여 압축을 수행하였다.

그림 8 (a)는 웹에서 널리 사용되고 있는 Flash로 부호화 결과를 보이고 있는데, Flash는 기존의 모든 영상을 키 프레임으로 지정해야하므로 압축률이 매우 떨어지고, 비트맵 영상을 벡터 그래픽 방식으로 변환하므로 화질의 열화가 매우 크게 발생한다. FLC 방식은 동영상 전체에 대한 256색상 이하의 전역 팔레트를 사용하고, 인트라 프레임 모드와 스킵 프레임 모드로 구성된다. 인트라 프레임 모드에서 런길이의 변형된 워드라인(word line coding)방식으로 압축부호화를 한다. 스킵 프레임은 전후영상을 비교해서 움직임이 작은 경우 해당 프레임을 삭제하고 이전 화면을 반복 재생하도록 하고 있다. 본 논문에서 제안하고 있는 P1 모드와 유사하다. 그림 8 (b)에 FLC로 부호화한 결과를 나타내고 있다. 동영상 전체에서 256색상의 전역 팔레트만을 사용하고 있기 때문에 장면 전환과 같이 색상이 변화하는 영상인 경우, 원래 영상의 색상과 차이가 발생한다. 본 논문의 방법과 비교해 볼 때, FLC에서는 움직임 정보를 이용한 압축을 수행하지 않고 모든 프레임을 인트라 모드로 부호화하기 때문에 압축률이 크게 저하된다. 그림 8 (c)의 Motion-JPEG은 움직임 정보를 부호화하지 않고 모든 프레임을 JPEG으로 압축하므로 압축효율이 매우 떨어지고 고능률

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr



(a) Flash (약 7.5:1) (b)FLC (256색 약 58:1) (c) Motion-JPEG (약 100:1)

(d) MPEG-1 (약 120:1) (e) MPEG-4 압축률 (약 120:1) (f) 제안하는 방법 (64색 약 120:1)

그림 8. 제안하는 방법과 다양한 방법의 비교

압축시 블록킹 현상이 현저하게 나타난다. 그림 8 (d)의 MPEG-1은 저장매체를 대상으로 수 메가 바이트(Mbyte) 정도의 압축률를 목표로 하고 있어 고능률 압축시 블록킹 현상이 화면 전체에서 발생하고, 특히 애니메이션 동영상에서는 주관적 화질에 크게 영향을 끼치는 에지 부분 및 윤곽선이 크게 열화되고 있다. 또한, 양자화시 YUV신호 중 색상 정보인 UV신호를 간축하므로 고능률 압축 시, 색상이 원 영상과 많은 차이가 발생하게 된다. 그림 8 (e)의 MPEG-4는 제안하는 방법과 비교해 볼 때, 전체적인 성능이 유사했으나 MPEG 계열에서의 양자화 특성상 색상의 변화가 발생하였다. 본 논문에서는 단일 VOP(video object plane)만 지원하는 MPEG-4 코덱을 사용했는데, 실시간 부호화 및 복호화의 제한된 조건이 아니고 애니메이션을 복수개의 VOP로 생성해서 압축한다면 보다 나은 결과를 얻을 수 있을 것이다. 그림 8 (f)는 제안하는 방법으로 비교된 소프트웨어 코덱을 사용한 결과보다 유사한 압축률에서 우수한 결과를 얻었다.

## V. 결 론

본 논문에서는 비교적 단순한 컴퓨터 셀 애니메이션 영상을 대상으로, 애니메이션의 인트라프레임 및 인터프레임의 특징을 반영해서 낮은 복잡도를 갖는 고능률의 새로운 압축방식을 제안하였다. 인트라 부호화에서는 색상수를 줄이기 위해 8진 트리에 기반한 팔레트를 이용한 칼라 양자화를 수행하였고 화면내 용장성을 줄이기 위한 색인 재배치, 적응 차분 부호화, 분류화 및 엔트로피 부호화를 수행하였다. 인터 프레임 부호화에서는 움직임의 특성을 분류하여 블록 단위의 움직임 리플레니쉬먼트 기법을 이용해서 부호화를 수행하였다.

현재 널리 사용되고 있는 소프트웨어 코덱인 Flash, FLC, M-JPEG, MPEG-1 및 MPEG-4과 제안하는 방법을 비교 및 분석함으로써 고능률 압축 부호화시 제안하는 방법이 우수하다는 것을 보였다. 향후, 웹 상에서 실시간 전송 등의 응용을 위한 비트 스트림의 신택스(syntax) 구조 등이 연구되어야 할 것으로 사료된다.

## 참 고 문 헌

[1] Chwan-Hwa John Wu and J. David Irwin, *Emerging Multimedia Computer Communication Technologies*, Pentice Hall, 1998

[2] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video, and Audio Coding,*

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

Prentice Hall, 1996

[3] 정제창 번역, *최신 MPEG*, 교보문고, 1995

[4] 정제창 번역, *응용 MPEG*, 교보문고, 1997

[5] James D. Murray and William Vanryper, *Graphics File Formats*, OReilly & Associates, INC., April, 1996.

[6] File formats from http://www.wotsit.org/

[7] Touradj Ebrahimi, "MPEG-4 Video Verification Model Version 11.0", *ISO/IEC JTC1/SC29WG11 MPEG98/N2172*, Tokyo, March 1998

[8] Vasudev Bhaskaran and Konstantinos Konstatinides, *IMAGE AND VIDEO COMPRESSION STANDARDS Algorithms and Architectures second edition*, Kluwer Academic Publishers, 1997

[9] Arun N. Netravali and Barry G. Haskell, *Digital Pictures 2nd Ed.*, PLENUM, 1995

[10] Dean Clark, "Color octree routines", *Dr. Dobbs Journal*, June, 1995

[11] Ian Ashdown, *A Programmer's Perspective*, John Willey & Sons Inc., 1995

[12] 유훈, 이창호, 민병석, 정제창, "영상 압축을 위한 저 복잡도를 갖는 고효율 엔트로피 부호화기", *제11회 신호처리 합동 학술대회*, 제11권, 제1호, pp. 789-792, 10월 1998년

[13] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989

[14] 경병표, "2D 애니메이션의 동향, *정보과학회지*, 제17권, 제2호, pp. 29-36, 2월 1992년

[15] http://www.virtualdub.org/

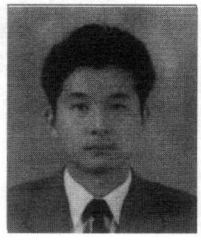
[16] http://www.macromedia.com/software/flash/

[17] B. Min, H. Yoo, J. Jeong, and B. Choi, "A Novel Compression Algorithm for Cell Animation Images", *Proceeing of IEEE ICIP*, Oct., 2001, Vol. 2, pp.459-462

[18] Khalid Sayood, *Introduction to Data Compression*, 2nd Edition, Morgan Kaufmann Publishers, 2000

[11] ISO/IEC JTC1 1/SC 29/WG FCD 14495, Lossless and near-lossless coding of continuous tone still image, 1997

민 병 석(Byong-seok Min) 정회원



1990년 2월 : 한양대학교 전자통신공학과 졸업

1992년 8월 : 한양대학교 전자통신공학과 석사

1993년 3월~현재 : 동대학원 박사과정

1995년 3월~현재 : 충청대학 컴퓨터학부 조교수

<주관심 분야> 영상처리, 영상통신, 멀티미디어통신

정 제 창(Je-chang Jeong) 정회원

1980년 2월 : 서울대학교 전자공학과 졸업

1982년 2월 : 한국과학기술원 전자공학과 석사

1990년 7월:미국 미시간대학교 전기공학과 박사

1981년 9월~현재 : 한양대학교 전자전기컴퓨터공학부 부교수

<주관심 분야> 영상처리, 영상통신, 멀티미디어통신

최 병 욱(Byung-uk Choi) 정회원

1973년 2월 : 한양대학교 전자공학과 졸업

1978년 3월 : 일본게이오대학교 전기공학과 석사

1981년 3월:일본 게이오대학교 전기공학과 박사

1981년 9월~현재 : 한양대학교 전자전기컴퓨터공학부 교수

<주관심 분야> 영상처리, 패턴인식, 멀티미디어통신

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr