

컬러 인덱스 캐릭터 영상의 무손실 압축

권혁민, 김만배
강원대학교 컴퓨터정보통신공학과
E-mail: kwonhm@kwnu.kangwon.ac.kr

A lossless compression scheme for color-indexed character images

Hyukmin Kwon and Manbae Kim

Department of Computer, Info. and Telecom, Kangwon National University

요약

현재 모바일 환경에서 PDA, 휴대폰 등으로 많은 캐릭터 영상들이 서비스되고 있다. 본 논문에서는 이러한 캐릭터 영상들의 무손실 동영상 압축 방법을 제안한다. 캐릭터 영상은 256 컬러 인덱스 영상이며, 영상간에 유사한 컬러로 제작된다. 제안한 압축 방법은 기존 MPEG, H.264에서 사용하는 동영상 부호화 기법 이외에도 캐릭터 영상의 특성을 반영한다. 각 블록 타입을 DC mode, INTER mode, 및 INTRA mode로 분류하는데, 다른 mode와 유사하게 DC mode는 캐릭터 영상의 특성을 활용한다. INTER mode는 예측 에러의 분포를 분석하여 세가지 예측 방법들 중에서 압축 효율이 우수한 방법으로 처리한다. 또한 INTRA mode는 4개의 서브모드로 나누어 압축 효율성을 제고한다.

12개의 캐릭터 실험 영상들은 각각 12장의 영상으로 구성되는데, 제안 압축 방법을 적용하면 평균 1.44:1의 압축율을 얻는다. 또한 실험 결과에서는 각 블록 타입의 비율 및 INTRA와 INTER 블록의 비율을 조절하여 얻어지는 압축율의 변화를 조사하였다.

1. 서론

현재의 무선 인터넷 기술은 하루가 다르게 빠른 발전을 하고 있다. 모바일 기기의 메모리와 CPU의 성능 향상으로 텍스트 및 단순한 이미지를 제공하는 서비스에서 동영상의 서비스가 가능해 지고 있다. 상대적으로 적은 메모리 크기와 낮은 처리 속도를 가진 모바일 기기는 모바일 환경에서 다양한 동영상 및 멀티미디어 서비스의 구현을 필요로 한다.

본 논문에서는 모바일 분야에서 많은 서비스가 제공되는 캐릭터 영상(character image)들의 무손실 동영상 압축 기법을 제안한다. 일반적으로 사용되는 true color 이미지는 용량이 매우 크기 때문에 화질 열화의 최소화 및 용량의 감소가 필요하다. 따라서, 이미지에서 많이 사용되는 8비트 256개의 색을 선정하여 사용하면 이미지의 용량을 RGB 24 bit에 비하여 1/3로 줄게 된다. 1,600만개의 색 중에서 256개의 색을 팔레트라는 룩업테이블(lookup table)로 만들어 저장된다. 각 화소의 값은 색을 표현하는 값이 아닌 팔레트의 인덱스이다. 팔레트의 크기는 256 이외에도 64, 16을 가질 수도 있다. 화소는 6비트, 4비트의 메모리만 사용하게 되어 메모리양이 작게 되지만, 영상 화질이 상대적으로 열화되는 단점이 있다. 따라서 일반적으로 256 인덱스 컬러를 사용한다.

부호화(encoder)에서는 일정한 컬러수로 영상을 표현하는 인덱스 영상(indexed image)를 입력으로 하며, 무손실 동영상 압축(lossless compression)을 구현한

다. 관련 연구로는 주로 컬러 인덱스 영상(color-indexed image)의 압축으로 PWC[1], EIDAC[2], RAPP[3], 및 Chen이 제안한 방법[4] 등이 있다.

MPEG 및 H.264 등의 동영상 부호화 기법들은 중복성을 줄이기 위하여 영상내의 공간적 상관도(spatial correlation)와 시간적 상관도(temporal correlation) 특성을 이용하는 인트라 코딩(intra coding)과 인터 코딩(inter coding) 기법을 활용한다[5]. 캐릭터 영상에서는 블록의 화소가 동일하거나, 영상간의 화소 값의 변화가 특정 부분에 집중되기도 하다. 따라서 기존 동영상 압축 기법을 기본으로 캐릭터 영상들의 블록 특성에 따라 부호화 방법을 보다 세분화하여 압축 효율의 향상을 얻고자 하는 것이 본 논문의 공헌이다.

논문의 구성은 다음과 같다. 2절에서는 제안 부호화 알고리즘의 개요를 설명한다. 3절에서는 구현 알고리즘을 자세히 소개한다. 4절에서는 제안된 알고리즘을 12개의 실험 영상에 적용한 실험 결과를 검증한다. 마지막 5절에서 결론을 맺는다.

2. 제안 방법

그림 1의 부호화 과정을 보면 입력된 프레임 F_n 은 다음 프레임의 운동 예측(Motion Estimation: ME) 및 운동 보상(Motion Compensation: MC)을 위해 버퍼 F_{n-1} 에 저장되고 현재 프레임은 $N \times N$ 블록 단위로 처리한다. 블록내의 모든 화소 값이 동일하면 DC 모드이고, 아니면 이전 프레임을 참조하여 인트라 모드

(intra mode) 또는 인터 모드(inter mode)로 블록들을 분류한다. 각각의 블록들을 DC, 인트라, 및 인터 모드로 분류하고 서로 다른 방식으로 부호화한다.

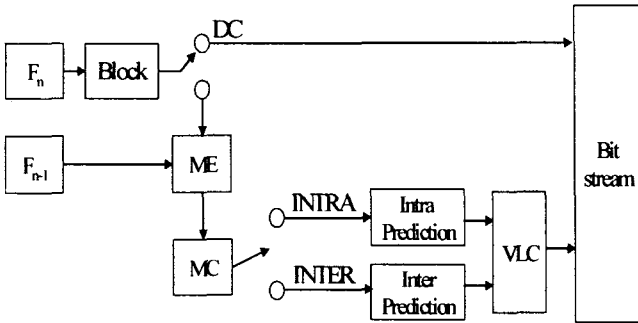


그림 1. 부호화기

복호화(decoder)의 경우는 부호화의 역 과정을 거친 후 원래의 데이터 값으로 복원한다. 그림 2에서처럼 복원된 영상은 다음 영상을 위해 버퍼(PIB)에 저장되며 입력 순서대로 복호화된다.

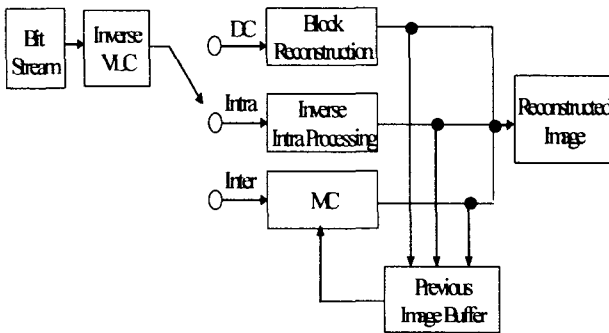


그림 2. 복호화기

3. 캐릭터 영상의 부호화

캐릭터 영상은 인덱스 데이터이므로 적은 데이터 손실에도 영상의 변화가 심하다. 따라서 무손실 압축이 고려되어야 한다. 블록의 화소가 동일하거나, 영상간의 화소 값의 변화가 특정 부분에 집중되기도 한다. 따라서 기존 동영상 압축 기법을 기본으로 캐릭터 영상들의 블록 특성에 따라 부호화 방법을 보다 세분화하여 압축 효율의 향상을 얻고자 한다.

A. 캐릭터 영상의 특성

실험 영상 데이터는 분석 결과 다음과 같은 특성을 가진다.

(1) 입력으로 사용되는 인덱스 영상은 Windows의 system color table을 사용하여 모두 동일한 팔레트인 CLUT (Color LookUp Table)을 갖는다.

- (2) 전체 블록들에서 동일한 화소 값을 가지는 블록들의 비율은 약 20%이다.
- (3) 또한, ME/MC에서 $MV=(0,0)$ 인 블록의 비율은 약 5%이다.
- (4) 부호화하기 위한 화소 데이터는 인덱스이므로, 기존의 자연 영상과 다른 분포를 가진다. 따라서 특성에 맞는 예측 부호화 방법이 필요하다.

그림 3의 실험 영상들은 부분적인 차이를 보이지만, 상대적으로 많은 유사성을 갖는다. 특히 외부 영역에서는 거의 변화가 없음을 확인할 수가 있는데 이러한 부분은 특별한 처리 없이 이전 프레임의 정보를 그대로 사용할 수가 있다. 이처럼 각각의 블록별 특성을 고려함으로써 압축율의 향상이 가능하다.

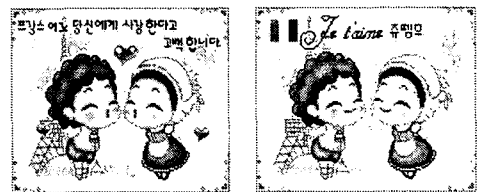


그림 3. 컬러 인덱스 캐릭터 영상들. 160x120 및 256 indexed color

B. 블록 타입 결정

엔코더(Encoder)에서는 모든 프레임을 NxN 블록 단위로 압축을 수행한다. 프레임내의 블록들은 3가지 모드로 나누어져 각각 서로 다른 방식으로 처리된다. 블록 타입(block type)은 DC mode, INTRA mode, INTER mode로 구성되며, 그림 4는 타입 결정 방법을 보여준다.

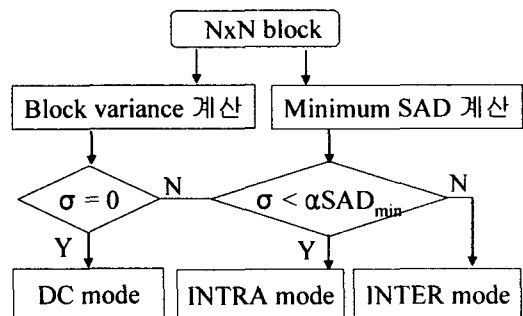


그림 4. 블록 타입 결정 방법

먼저 입력 프레임의 블록 화소 값의 편차, σ 를 식 (1)에서 계산한다.

$$\bar{x} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \quad (1)$$

$$\sigma = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - \bar{x}|$$

여기서 $x(i, j)$ 는 화소의 인덱스 값이다.

그리고 현재 블록의 운동 벡터는 이전 프레임에서 구한다. 운동 예측으로부터 얻어진 이전 프레임내의 가장 유사한 블록과의 에러 값인 SAD_{min} (Sum of Absolute Difference)을 다음 식 (2)에서 구한다.

$$SAD_{min} = \min_{m, n} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - x_{mc}(i + m, j + n)| \quad (2)$$

여기서 mc 는 motion-compensation이다.

σ 가 0이면 블록내의 화소 값이 모두 동일하므로 DC mode가 된다. σ 가 $\alpha \cdot SAD_{min}$ 보다 작은 경우는 블록 화소들의 유사성이 크므로 INTRA mode로 분류되고, 아니면 INTER mode로 타입을 결정한다. α 는 부호화 단계에서 INTER와 INTRA mode 블록 개수의 비율을 제어하기 위해서 사용된다.

C. DC mode

DC mode는 해당 블록의 화속 데이터 값이 모두 같다. 따라서 1개의 화소 값만의 부호화가 가능하므로 상대적으로 우수한 압축 효과가 발생한다.

D. INTRA mode

INTRA mode는 H.264의 INTRA Prediction mode 중 vertical mode와 horizontal mode를 이용하며[6], 캐릭터 영상의 특성을 조사하여 neighbor mode를 추가하였다. 그림 5는 임의로 선택한 4x4 인덱스 데이터를 보여준다. 인접 화소의 데이터와 비교하여 상기 3 종류의 mode를 선택하였다.

184	184	184	184	209	184	209	184	209	209
184	<i>184</i>	<i>184</i>	<i>184</i>	<i>209</i>	209	<i>184</i>	<i>209</i>	<i>209</i>	<i>209</i>
184	<i>184</i>	<i>209</i>	<i>209</i>	<i>209</i>	209	<i>209</i>	<i>209</i>	<i>209</i>	<i>209</i>
184	<i>184</i>	<i>209</i>	<i>209</i>	<i>209</i>	209	<i>255</i>	<i>255</i>	<i>255</i>	<i>255</i>
209	<i>209</i>	<i>209</i>	<i>209</i>	<i>209</i>	255	<i>255</i>	<i>255</i>	<i>255</i>	<i>255</i>
209	209	209	209	209	138	8	8	8	8
209	<i>209</i>	<i>209</i>	<i>209</i>	<i>209</i>	98	8	8	8	8
209	<i>209</i>	<i>209</i>	<i>209</i>	<i>255</i>	98	8	8	8	8
255	<i>255</i>	<i>255</i>	<i>255</i>	<i>255</i>	98	8	8	8	8
255	<i>255</i>	<i>246</i>	<i>246</i>	<i>246</i>	98	<i>138</i>	8	8	8
211	144	144	211	211	161	200	161	200	161
144	<i>211</i>	<i>211</i>	<i>144</i>	<i>211</i>	161	<i>200</i>	<i>161</i>	<i>200</i>	<i>161</i>
144	<i>211</i>	<i>211</i>	<i>144</i>	<i>211</i>	161	<i>200</i>	<i>161</i>	<i>200</i>	<i>161</i>
211	<i>144</i>	<i>144</i>	<i>211</i>	<i>211</i>	161	<i>200</i>	<i>161</i>	<i>200</i>	<i>161</i>
211	<i>211</i>	<i>211</i>	<i>211</i>	<i>144</i>	57	<i>57</i>	<i>57</i>	<i>53</i>	<i>57</i>

그림 5. 인덱스 데이터. 4x4 블록(*italic*) 및 8개의 인접 화소

각 mode는 그림 6처럼 어느 위치의 인접 화소를 예측에 사용하는가에 따라 구분된다.

	A	B	C	D
E	a00	a01	a02	a03
F	a10	a11	a12	a13
G	a20	a21	a22	a23
H	a30	a31	a32	a33

그림 6. 4x4 블록(a00 ~ a33) 및 인접 화소들(A~H)

Intra mode 1 (neighbor mode)은 인접 화소를 고려하지 않고, 현재 화소와 이전 화소의 차이를 구한다. 따라서 그림 6에서 a00, a00-a01, a01-a02, ...의 값을 구한다. Intra mode 2 (vertical mode)는 현재 블록의 가장 근접한 상위 블록의 가장 가까운 화소 4개 (A, B, C, D)를 이용한다. a00, a10, a20, a30은 A-a00, A-a10, A-a20, A-a30로 예측 값이 얻어진다. 다른 화소들도 유사하게 B, C, D를 이용한다. Intra mode 3 (horizontal mode)은 Intra mode 2와 같은 방법을 사용하지만 기준 화소는 E, F, G, H이다. a00, a01, a02, a03는 E-a00, E-a01, E-a02, E-a03로 예측된다. 다른 화소들도 유사하게 F, G, H를 이용한다. 이와 같은 3가지 방법을 적용한 후 에러의 절대값을 최소로 갖는 모드를 찾아낸 후, 해당 모드로 에러 값을 부호화한다. 다음 식 (3)은 세 모드에서의 에러 합을 구하는 것을 정리한 것이다.

$$S_1 = |a_{01} - a_{00}| + |a_{02} - a_{01}| + \dots + |a_{33} - a_{32}|$$

$$S_2 = \sum_{i=0}^3 [|A - a_{0i}| + |B - a_{1i}| + |C - a_{2i}| + |D - a_{3i}|] \quad (3)$$

$$S_3 = \sum_{i=0}^3 [|E - a_{0i}| + |F - a_{1i}| + |G - a_{2i}| + |H - a_{3i}|]$$

영상내의 블록들의 위치에 따라 3가지 mode를 모두 적용시킬 수 없는 경우가 있다. 즉 첫 번째 블록이나 첫 번째 열과 행의 블록들은 1개 혹은 2개 모드만 적용할 수 있다. 따라서 각 블록에 적용시키는 방법은 그림 7과 같다. 16개의 블록으로 구성된 영상을 보여주며, B₁₂₃는 블록 B에 mode 1, 2, 3의 적용이 가능함을 의미한다.

	B ₁₃	B ₁₃	B ₁₃
B ₁₂	B ₁₂₃	B ₁₂₃	B ₁₂₃
B ₁₂	B ₁₂₃	B ₁₂₃	B ₁₂₃
B ₁₂	B ₁₂₃	B ₁₂₃	B ₁₂₃

그림 7. 블록 위치 및 적용 모드

D. INTER mode

INTER mode는 다시 4가지 서브모드(submode)로 구분된다. 여기서는 MV(Motion Vector)와 PE (Prediction Error) 블록의 값을 이용한다. PE는 현재 블록과 MC 블록의 차이이다. 예를 들어 Z_MV는 MV 값이 Zero=(0,0)이고 NZ_PE는 MC 블록과의 현재 블록의 차이인 PE 블록에서 적어도 하나의 값이 0이 아니다. 모든 값이 0이면, Z_PE로 표기한다. 표 1은 INTER mode의 4가지 서브모드와 특성을 설명한다.

표 1. INTER mode의 4가지 서브모드

서브모드	특 성
Z_MV_Z_PE	MV=(0,0), 모든 PE값이 0임
NZ_MV_Z_PE	MV≠(0,0), 모든 PE값이 0임
Z_MV_NZ_PE	MV=(0,0), PE에서 적어도 하나는 0이 아님
NZ_MV_NZ_PE	MV≠(0,0), PE에서 적어도 하나는 0이 아님

서브모드에 따른 부호화 방법은 다음과 같다.

- (1) Z_MV_Z_PE: 예러와 MV가 0이므로 부호화는 서브모드 신택스만 저장한다.
- (2) NZ_MV_Z_PE: 이전 프레임에 동일한 블록이 존재하므로 MV를 저장한다.
- (3) Z_MV_NZ_PE: 예러가 존재하므로 예러 블록을 부호화한다.
- (4) NZ_MV_NZ_PE: 예러 블록 및 MV를 부호화한다

E. 신택스

표 2는 각 모드에 대한 신택스(syntax)와 추가 정보를 보여준다. 신택스에는 1 byte가 할당되어 모드 타입을 알려준다. DC mode는 DC_MODE 및 DC값이 필요하므로 2 byte가 필요하다. INTRA mode는 INTRA_MODE 및 예측 예러의 허프만 부호화로 가변 byte가 할당된다. 4 종류의 INTER mode에서 Z_MV_Z_PE는 신택스 정보만 필요하므로 1 byte가 할당되고, NZ_MV_Z_P는 신택스 정보 및 운동벡터가 필요하므로 3 byte, Z_MV_NZ_PE는 신택스 정보 및 예측 예러의 허프만 부호화로 가변 byte가 할당된다. 마지막으로 NZ_MV_NZ_PE는 운동 벡터 및 허프만 부호화로 마찬가지로 가변 byte가 할당된다.

표 2. syntax 및 추가 정보

	Syntax	value	byte
[block_type]	DC_MODE	DC 값	2
	INTRA_MODE	PE	*
	Z_MV_Z_PE		1
	NZ_MV_Z_PE	(MVx MVy)	3
	Z_MV_NZ_PE	PE	*
	NZ_MV_NZ_PE	(MVx MVy), PE	*

F. Huffman 부호화

마지막으로 INTRA mode와 INTER mode (Z_MV_NZ_PE, NZ_MV_NZ_PE)에서 얻어진 예러 값들은 일반적인 Huffman 부호화를 거쳐 최종 압축 비트 스트림이 만들어진다 [7].

기존 방법에서는 블록 또는 매크로블록 단위로 허프만 부호화를 적용하지만, 본 방법에서는 프레임 단위에 적용하였다. 또한 INTRA mode와 INTER mode 블록은 분리하여 부호화한다. 전자의 경우에는 한 프레임에 있는 해당 블록들의 값들을 허프만 부호화에 사용된다. 신택스 구조는 symbol, 각 symbol에 대한 huffman code와 bit length, 그리고 모든 데이터에 대한 huffman code로 구성된다. 후자도 유사하게 부호화된다.

4. 실험 결과

입력영상은 160x120의 Indexed bmp이다. 12 개의 실험 영상을 사용하며, 각 영상은 12 장으로 구성된다. 그림 8은 사용된 12 개중 4개의 실험 영상인 “5개 국어사랑해”, “월별아이들”, “탄생화” 및 “탄생별자리”의 인덱스 영상들을 보여준다.

블록은 4x4 와 8x8을 이용하여 성능 평가를 한 결과, 4x4가 높은 압축율을 보였다. 따라서 실험에서는 4x4 블록을 이용한다. 블록 기반 운동 예측 BMME를 이용하고, 그림 4의 블록 타입 결정 방법을 사용하여 구한 모드 별 블록의 개수는 표 3에서 보여진다. 각 실험 영상의 총 블록 개수는 158,400 이므로, DC_mode는 22 %, INTRA mode는 19 %, INTER mode는 약 59%를 차지한다. 표 4는 제안 알고리즘을 적용하여 얻어진 압축율이며, 평균 압축율은 1.44:1이다.

INTRA 및 INTER 블록의 비율에 따른 압축율의 변화를 보기 위해 그림 4에서 α 값을 변경하면서 압축율을 조사한 결과는 표 5에서 보여진다. 인트라 블록의 개수가 증가하면 (즉, α 값이 증가할수록) 압축율이 높아진다. 그러나 어느 이상이 되면 오히려 압축율이 떨어지게 된다. 표에서 보는 것과 같이 α 값이 3.0일 경우 가장 높은 압축율을 얻는다. α 값에 따른 압축율의 증가율이나 최대가 되는 경우가 α 값이 영상에 따라 다른 결과를 얻게 되었다. 영상의 블록 타입중 Z_MV_Z_PE와 NZ_MV_Z_PE의 블록 개수가 많은 영상의 경우는 인트라 코딩의 경우가 전체 압축율 향상에 더 좋은 결과를 가져왔고, 그렇지 않은 경우는 인트라 코딩의 경우가 더 적합하다는 것을 보여준다.

5. 결론

본 논문에서는 입력 영상으로 사용되는 인덱스 캐릭터 이미지 특성에 대해 설명한 후, 그 특성에 적합

한 새로운 무손실 압축 알고리즘을 제안하고 부호화 방법을 제안하였다. 캐릭터 영상의 특성에 따라 블록을 DC, INTRA, INTER로 분류하고, INTRA 및 INTER는 여러 서브모드로 나누어 압축 효율을 높이고자 하였다. 실험 결과를 토대로 영상의 특성에 따라 보다 적합한 압축 방법에 대해서도 설명하였다.

향후 연구는 캐릭터 영상의 특성을 상세히 분석하여 보다 효율적인 압축 알고리즘을 연구하고자 한다. 또한 압축 효율을 높이기 위해서는 인덱스 데이터의 재배열(palette reordering) [8] 등의 전처리 과정이 필요하다면 실제 압축율에 영향을 준다.

감사의 글

본 연구는 정보통신부의 대학 IT연구센터 육성지원인 ITRC 프로그램의 지원을 받아 수행되었음.

참고 문헌

1. P. J. Ausbeck Jr, "The piecewise-constant image model," *Proc. IEEE*, vol 88, pp. 1779-1789, Nov. 2000.
2. Y. Yoo, Y.G. Kwon, and A. Ortega, "Embedded image-domain compression using context models," in *Proc. 6th IEEE Int. Conf. Image Processing*, vol. 1, Kobe, Japan, Oct 1999. pp. 477-481.
3. V. Ratnakar, "RAPP: lossless image compression with runs of adaptive pixel patterns," *Proc. 32nd Aslomar Conf. Signals, Systems, and Computers*, vol. 2 1998, pp.1251-1255.
4. X. Chen, S. Kwong, and J. F. Feng, "A new compression scheme for color-quantized images," *IEEE Trans. Circuits Syst. Video Technology*, vol. 12, pp. 904-908, Oct. 2002.
5. K. R. Rao and J. J. Hwang, *Techniques and standards for image, video and audio coding*, Prentice-Hall, 1996.

6. I. Richardson, *H.264 and MPEG-4 video compression*, John Wiley & Sons Ltd., 2003.
7. D. Salomon, *Data Compression*, 2nd Ed., Springer, 2000.
8. A. Pniho and A. Neves, "A survey on palette reordering methods for improving the compression of color-indexed images," *IEEE Trans. Image Processing*, vol. 13, No. 11, Nov. 2004.

표 4. 영상별 압축율 ($\alpha = 0$). 압축 데이터는 byte임.

실험 영상	압축 데이터	압축율
5개국어사랑해	125,664	1.83
12계절이쁘이	201,989	1.14
나라별사랑해	118,586	1.94
서울을대표하는빌딩	216,612	1.06
월별아이들	187,819	1.23
탄생별자리	187,455	1.23
탄생석	157,347	1.46
탄생화	167,968	1.37
부적	148,285	1.55
달력	133,847	1.72
스포츠카	163,277	1.41
깜찍아기	107,693	2.14
평균	159,711	1.44

표 3. 실험 영상의 모드별 해당 블록의 개수. 영상의 전체 블록 개수는 13,200임. 전체 영상의 블록 개수는 158,400임

Test Data	DC_MODE	INTRA_MODE	Z_MV_Z_PE	Z_MV_NZ_PE	NZ_MV_Z_PE	NZ_MV_NZ_PE
5개국어사랑해	2,043	946	1,557	668	1,780	6,206
12계절이쁘이	1,903	3,226	47	99	65	7,860
나라별사랑해	4,313	1,335	639	1,123	390	5,400
서울을대표하는빌딩	1,345	2,420	11	63	96	9,265
월별아이들	2,847	3,740	24	18	70	6,501
탄생별자리	2,118	3,472	18	140	11	7,441
탄생석	3,794	1,201	29	1,311	223	6,642
탄생화	1,822	3,461	10	94	177	7,636
부적	2,350	2,993	11	172	418	7,256
달력	3,695	5,023	0	4	28	4,450
스포츠카	4,799	1,463	3	12	379	6,544
깜찍아기	3,065	898	868	236	2,554	5,579
Total Blocks	34,094	30,178	3,217	3,940	6,191	80,780

5개국어 사랑해



월별 아이들



탄생화



탄생별자리

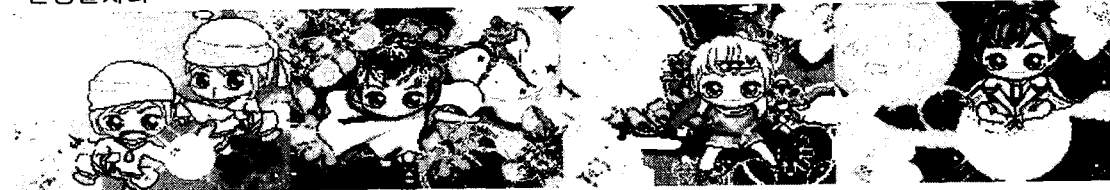


그림 8. 실험 인덱스 영상

표 5. α 값에 따른 영상별 압축. $\alpha = 1.5 \sim 3.0$. 압축 데이터는 byte임.

실험 영상	$\alpha = 1.5$		$\alpha = 2.0$		$\alpha = 2.5$		$\alpha = 3.0$	
	압축데이터	압축율	압축데이터	압축율	압축데이터	압축율	압축데이터	압축율
5개국어사랑해	119,074	1.93	115,742	1.99	114,723	2.00	114,752	2.00
12계절이쁜이	189,209	1.21	179,565	1.28	175,519	1.31	173,595	1.32
나라별사랑해	114,139	2.01	111,011	2.08	109,285	2.11	108,798	2.12
서울을대표하는빌딩	199,422	1.16	186,318	1.24	180,934	1.27	178,388	1.29
월별아이들	172,161	1.34	163,267	1.41	159,864	1.44	159,077	1.45
탄생별자리	175,532	1.31	167,379	1.38	164,019	1.40	162,734	1.41
탄생석	153,198	1.50	147,444	1.56	145,736	1.58	145,058	1.59
탄생화	158,252	1.46	155,098	1.49	154,011	1.50	153,332	1.50
부적	127,323	1.81	117,812	1.96	114,688	2.01	113,192	2.03
달력	121,630	1.89	118,143	1.95	116,794	1.97	116,171	1.98
스포츠카	155,175	1.48	146,480	1.57	141,255	1.63	138,626	1.66
감짜야기	103,534	2.23	102,484	2.25	102,379	2.25	102,717	2.24
평균	149,054	1.55	142,561	1.62	139,933	1.65	138,869	1.66