

## 객체 기반 MPEG-4 동영상의 입체 변환

### Stereoscopic Conversion of Object-based MPEG-4 Video

박상훈\*, 김만배\*\*, 손현식\*\*

\*\*강원대학교 컴퓨터정보통신공학과

\*광주과학기술원 정보통신공학과

Sang Hoon Park, Man Bae Kim, Hyun Sik Sohn

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

Dept. of Info. and Telecom., KJIST

E-mail: [manbae@kangwon.ac.kr](mailto:manbae@kangwon.ac.kr)

#### Abstract

In this paper, we propose a new stereoscopic video conversion methodology that converts two-dimensional (2-D) MPEG-4 video to stereoscopic video. In MPEG-4, each image is composed of background object and primary object. In the first step of the conversion methodology, the camera motion type is determined for stereo image generation. In the second step, the object-based stereo image generation is carried out. The background object makes use of a current image and a delayed image for its stereo image generation. On the other hand, the primary object uses a current image and its horizontally-shifted version to avoid the possible vertical parallax that could happen. Furthermore, URFA(Uncovered Region Filling Algorithm) is applied in the uncovered region which might be created after the stereo image generation of a primary object. In our experiment, we show MPEG-4 test video and its stereoscopic video based upon our proposed methodology and analyze its results.

#### 1. 서론

입체 동영상(stereoscopic video)은 각 영상에서 좌영상(left image)과 우영상(right image)을 통한 양안시차(binocular disparity)를 제공함으로써 시청자에게 깊이감을 인식(perception)할 수 있게 한다. 일반적으로 입체 동영상을 제작하기 위해서는 입체 카메라를 이용해야 한다. 그러나 입체 동영상 제작은 일반 동영상 제작에 비해 엄청난 비용과 시간이 드는 작업으로서 영화나 방송과 같은 분야에 적용하기가 어렵다.

이러한 이유로 MPEG(Moving Picture Experts Group)이나 NTSC(National Television System Committee)와 같이 기존에 존재하는 2차원 동영상을 입체 동영상으로 변환하는 연구가 많이 있었다 [2,3,4,5,6]. 그러나 MPEG-1, MPEG-2, NTSC와 같은 프레임 기반(frame-based) 동영상을 이용한 입체 동영상 변환에는 몇 가지 문제가 있다. 주객체가 비강체(non-rigid)인 경우에는 이 부분에서 인접 영상간에 변화가 심할 수 있기 때문에 입체 영상 생성시 주객체 영역에서 입체감을 떨어뜨리는 수직시차(vertical

parallax)가 발생할 수 있다. 이것은 영상내에 주객체에 대한 별도의 운동 및 영역 정보가 없기 때문이다.

이와 달리 객체 기반(object-based)의 부호화(encoding)를 지원하는 MPEG-4 동영상을 변환할 경우에는 이러한 문제점을 해결할 수 있다. MPEG-4에서는 이러한 객체를 비디오 객체 평면(video object plane: VOP)이라고 하며 VOP 단위의 부호화 및 복호화(decoding)를 수행한다. 이를 위해 MPEG-4 부호기(encoder)는 각 객체의 휘도 및 색차와 같은 텍스처 정보(texture information)외에 형상 정보(shape information)를 입력으로 받아 부호화를 수행한다[7,8].

본 논문에서는 MPEG-4의 객체 기반 부호화를 이용한 입체 동영상 변환 방법에 대해 제안한다. 변환할 동영상의 각 영상은 주객체(primary object)와 배경객체(background object)로 구성된다. 제안 방법은 크게 카메라 운동 유형 결정(camera motion analysis) 과정과 입체 영상 생성(stereo image generation) 과정으로 구성된다. 배경 객체의 운동 벡터(motion vector: MV)는 카메라 운동 유형 결정에 사용된다.

본 논문의 구성은 다음과 같다. 1 절의 서론에 이어 2 절에서는 제안하는 입체 동영상 변환 방법에 대해 설명한 후 3 절에서는 연구 결과를 설명한다. 마지막 4 절에서는 결론 및 향후 연구방향에 대해 설명한다.

#### 2. 제안하는 MPEG-4 동영상의 입체 변환

그림 1은 제안 방법의 전체 블록 다이어그램이다. 입력으로 들어오는 MPEG-4 비디오의 각 영상은 배경객체(background object: VOP<sub>BG</sub>)와 주객체(primary object: VOP<sub>PO</sub>)로 구성된다. MPEG-4 비디오 데이터는 역다중화기(demultiplexer)를 거쳐 부호화된 VOP<sub>PO</sub>와 VOP<sub>BG</sub>로 분리된다. 각각은 VOP 복호기를 통해 복원된다. VOP<sub>BG</sub>의 운동 벡터 필드(motion vector field: MVF)는 카메라 운동 분석(camera motion analysis)에 사용된다. 여기서 MVF는 영상의 매크로블록(macroblock:MB)당 존재하는 운동벡터(motion vector: MV)의 집합으로 정의한다. 카메라 운동 분석 후 객체 라 운동 유형(camera motion type)이 결정되면 최종적으로 객체 기반 입체 영상 생성(object-based stereo image generation) 과정을 거쳐 좌영상(left image)과

우영상(right image)을 생성한다.

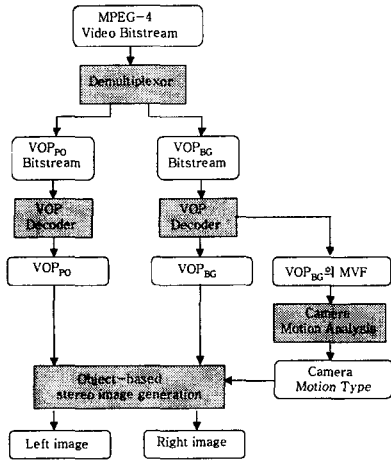


그림 1. 제안하는 입체 동영상 변환의 블록 다이어그램

VOP<sub>BG</sub>의 MVF는 카메라 운동 유형 결정에 사용된다. 카메라 좌측(camera left) 운동에서는 대부분의 운동 벡터는 우측 방향을 향할 것이고 카메라 우측(camera right) 운동인 경우에는 좌측 방향을 향할 것이다. 카메라 정지(camera static)인 경우에는 대부분의 운동 벡터의 크기는 0일 것이다.

카메라 운동 유형의 결정을 위해 먼저 VOP<sub>BG</sub>의 MVF에 대한 평균(average)을 취하여 카메라 운동 변위값(dx<sub>cam</sub>, dy<sub>cam</sub>)을 구한다. 카메라 운동 유형 결정은 dx<sub>cam</sub>의 부호를 가지고 판단한다. dy<sub>cam</sub>은 입체 영상 생성 시 좌우 영상간에 수평 조절을 위해 사용된다. dx<sub>cam</sub> 값이 0보다 작으면 camera-right, 0이면 static, 0보다 크면 camera-left이다.

입체 영상 생성은 주객체 영역에 대해서는 현재 영상의 주객체와 이를 수평 이동한 것을 사용하고 배경 객체 영역에 대해서는 현재 영상과 지연 영상의 배경 객체를 사용한다. 이것은 앞에서 지적한 주객체 영역에서 발생할 수 있는 수직 시차를 없애기 위한 것이다. 먼저 주객체 영역의 좌우 영상 부분을 생성하기 위해서는 주객체의 수평 이동값을 결정해야 한다. 주객체의 수평 이동값의 범위는 식 (1)과 같다.

$$0 < S_{PO} < T_{FM} \quad (1)$$

수식 (1)에서 S<sub>PO</sub>는 주객체의 수평 이동 값을 의미하고 T<sub>FM</sub>은 고속 운동 임계값을 의미한다[6]. 주객체의 수평 이동 값은 주객체에 대한 음의 시차 값을 결정하며 T<sub>FM</sub>을 넘어가지 않는 범위 내에서 조절이 가능하다. S<sub>PO</sub>에 대한 이동 방향은 카메라 운동 유형에 따라 좌우된다. 그림 2는 카메라 운동 유형에 따른 주객체 영역의 입체 영상 생성 방법을 보여준다. (a)와 같이 카메라 우측 운동과 카메라 정지인 경우에는 현재 영상의 주 객체를 좌영상의 주객체로 사용하고 이것을 S<sub>PO</sub> 만큼 좌측으로 이동한 것을 우영상의 주객체로 사용한다. 이와 반대로 (b)와 같이 카메라 좌측 운동인 경우에는 현재 영상의 주 객체를 우영상의 주

체로 사용하고 이것을 S<sub>PO</sub> 만큼 우측으로 이동한 것을 좌영상의 주객체로 사용한다.

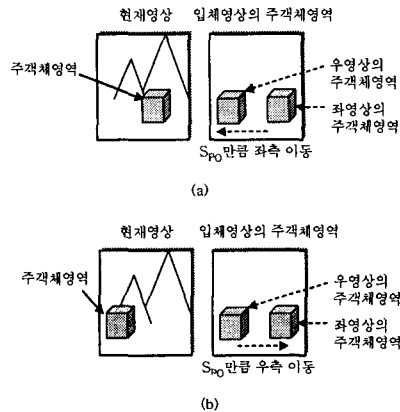


그림 2. 주 객체 영역에 대한 입체 변환 방법

주객체 영역과 달리 배경객체 영역에 대해서는 현재 영상과 지연영상의 배경객체를 사용한다. 이를 위해서는 먼저 현재 영상으로부터 몇 번째 지연영상을 선택할 것인지를 결정하는 delay factor f<sub>d</sub> 값을 구해야 한다. 식 (2)는 f<sub>d</sub>를 구하는 방법이다.

$$f_d = ROUND \left[ \frac{T_{FM}}{dx_{cam}} \right] \quad (2)$$

여기서 dx<sub>cam</sub>은 카메라 운동 유형 결정시 구한 카메라 운동 수직 변위값으로서 카메라 운동량에 비례한다. 즉 카메라 운동량이 크면 현재 영상으로부터 시간적으로 많이 떨어지지 않은 지연 영상을 사용하고 카메라 운동량이 작으면 시간적으로 많이 떨어진 지연 영상을 사용함으로써 배경 객체 부분의 수평시차를 조절한다.

f<sub>d</sub> 값이 결정되면 카메라 운동 유형에 따라 배경객체 영역의 좌우 영상을 결정해야 한다. 표 1은 카메라 운동 유형에 따른 배경 객체 영역의 좌우영상 결정 방법이다. 카메라 정지에서는 지연 영상을 사용하더라도 음의 시차를 얻을 수 없기 때문에 현재 영상의 배경객체만을 이용한다.

표 1. 좌우 배경 객체 선택 방법

Motion Type	Camera Motion	left VOP <sub>BG</sub>	right VOP <sub>BG</sub>
A	Static	current VOP <sub>BG</sub>	current VOP <sub>BG</sub>
B	Left	current VOP <sub>BG</sub>	delayed VOP <sub>BG</sub>
C	Right	delayed VOP <sub>BG</sub>	current VOP <sub>BG</sub>

이 때 카메라가 비수평으로 운동함으로써 입체영상의 배경 영역에서 비수평 시차가 발생할 수 있기 때문에 이것을 없애기 위해 수평 조절을 해주어야 한다. 수평 조절은 입체 영상의 배경 영역을 위해 선택된 지연 영상의 배경 객체에 대해 수직 이동을 함으로써 수행된다. 식 (3)은 수직 이동값, S<sub>BG</sub>를 구하는 방법이다.

$$S_{BG} = dy_{cam} \times f_d \quad (3)$$

여기서  $dy_{cam}$ 은 카메라 운동 유형 결정시 구한 카메라 운동 수직 변위값이다.  $f_d$ 값을 곱하는 이유는  $dy_{cam}$ 을 현재 영상과 지연 영상과의 간격에 따라 스케일링(scaling)하기 위한 것이다. 그림 3은 카메라 운동이 있는 경우의 현재 영상과 지연 영상을 이용한 배경 영역의 전체적인 입체 영상 생성 방법을 보여준다.

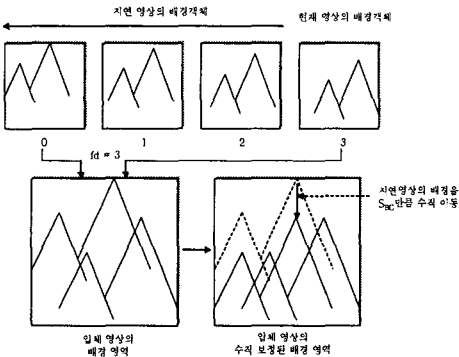


그림 3. 배경 영역에 대한 입체 변환 방법

주객체 영역의 입체 영상 생성시 주객체의 수평 이동한 것을 사용함에 따라 원래 주객체가 있던 영역에는 채워지지 않는 영역(uncovered region: UR)이 발생 할 수 있다. 이 영역은 해당 배경 정보로 채워야 한다. 그러나 UR에 해당하는 배경 정보는 인접한 영상으로부터 정보를 얻어 와야 한다. 이를 위해서는 인접한 영상 중에 UR에 해당하는 배경 정보를 가장 많이 포함하는 영상,  $F_T$ 를 찾아야 한다.

이를 위해 카메라 운동이 있는 경우와 없는 경우 두 가지로 나누어 URFA(UR Filling Algorithm)를 적용 하였다. 그림 4는 카메라 운동의 유무에 따른 URFA를 보여준다. (a)는 카메라 운동이 없는 경우로서 인접한 영상들에서 UR이 발생한 영역과 동일한 위치에서 배경 객체의 화소를 가장 많이 갖는 영상을  $F_T$ 로 선택하여 해당 영역을 UR에 복사한다. (b)는 카메라 운동이 있는 영상에서의 URFA로서 두 단계의 과정을 거쳐야 한다. 먼저 UR의 외곽 배경 영역인 UR'을 설정한 후 인접한 영상과의 UR' 영역을 비교하여 차이가 가장 많이 나는 영상을  $F_T$ 로 결정한다. 인접한 영상간에 UR'을 비교하기 위한 방법으로는 식 (4)와 같이 MAD(mean absolute difference)를 적용하였다.

$$MAD = \frac{1}{N} \sum_{(n \in UR)} |F_{UR}(n) - F_{K-L}(n)| \quad (4)$$

여기서  $N$ 은 UR'이 포함하는 화소(pixel)의 개수를 의미한다. 또한  $F_{UR}(n)$ 은 UR이 발생한 영상의 UR'내에 속하는 화소를 의미하여  $F_{K-L}(n)$ 은 인접 영상의 UR'영역에 포함되는 화소를 의미한다.

$F_T$ 를 찾은 후에는  $F_T$ 내에서 UR'과 가장 비슷한 영역을 찾은 후 내부 영역을 UR에 복사해야 한다. UR'과  $F_T$ 내에서의 UR'간 비교 방법으로서 역시 식 (4)와 동일한 MAD를 적용하였다. (4)와 다른 점은 비교할

영역들이 모두  $F_T$  내에 있다는 점이다.

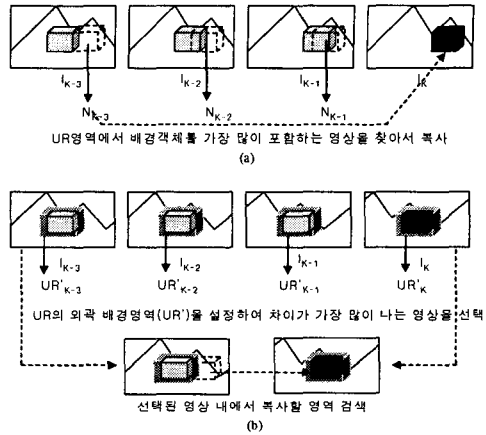


그림 4. URFA: (a) 카메라 정지, (b) 카메라 운동

### 3. 실험

실험 동영상으로는 MPEG-4 테스트 비디오 동영상에서 "stefan"과 "coastguard"를 사용하였다. 표 2는 실험 동영상의 부가 정보를 보여준다. 각 동영상의 형식은 CIF로서 영상의 크기는 352x288이고 VOP 개수는 주객체와 배경 객체의 두개로 구성된다.

표 2. 실험에 사용된 동영상

Sequence Name	Sequence Format	The number of VOP	Total number of frames
Stefan	CIF	2	280
Coastguard	CIF	2	280

그러나 실험 동영상은 부호화되지 않은 데이터이기 때문에 실험을 위해서는 먼저 MPEG-4 비디오 형식으로 부호화해야 한다. 이를 위해 실험 동영상 파일을 입력받아 VOP 부호기를 통해 각 객체의 비트스트림을 생성하였다. 이 후 VOP 복호기를 통해 각 객체의 MVF, 화소 정보, 영상 정보 파일을 추출하여 디스크에 저장한 후 구현한 실험용 프로그램을 이용해 입체 변환하였다.

먼저 그림 5와 6은 stefan 동영상에 대한 입체 변환 결과이다. stefan은 주객체가 비강체로서 이 영역에서 인접한 영상간에 변화가 심한 동영상이다. 그림 5는 7번째 영상에 대한 입체 변환 결과로서 추출된 배경 객체의 MVF, 입체 파라미터, 좌우영상, 비월영상을 순서대로 보여준다. MVF 분석 결과 카메라 운동 수평 변위값은 -12가 나왔기 때문에 카메라 우측 운동으로 결정되었다. 제안 방법에 따라 우영상은 현재영상이 그대로 사용되었고 좌영상의 배경객체 영역과 주객체 영역은 두 번째 지연영상의 배경객체와 현재영상의 주객체가 우측으로 이동된 것이 각각 사용되었다.

그림 6은 42번째 영상에 대한 결과로서 MVF 분석 결과 카메라 운동 수평 변위값이 0이 나왔기 때문에

카메라 정지 영상으로 결정되었다. 좌우영상의 배경객체 영역은 현재영상의 배경객체가 동일하게 사용되었다. 그러므로 주객체 영역만이 음의 시차를 갖는다.

MPEG-4 데이터 입체 변환의 연산 오버헤드(overhead)를 측정하기 위해 표 3과 같이 MPEG-4 디코딩에 걸리는 시간과의 상대적인 비교 실험을 하였다. 실험에서는 순수한 MPEG-4 디코딩 속도와의 비교를 위해 렌더링 과정은 포함시키지 않았다. 표 3에서  $N_F$ 는 전체 프레임 수,  $T_d$ 는 MPEG-4 동영상의 디코딩 시간(sec),  $T_s$ 는 입체 변환 시간,  $R$ 은  $T_s/T_d \times 100$ 이다. 두 개의 실험 동영상에 대해 실험을 수행하고 평균을 구한 결과 입체 변환을 수행시 디코딩 시간에 약 28%의 연산량이 필요한 결과를 보여준다. 현재 실험은 입체 변환 모듈의 최적화를 하지 않은 상태에서 이루어졌기 때문에 최적화 작업에 따라 연산 오버헤드는 많이 줄어들 수 있다.

표 3. 입체 변환 연산량 실험 결과

실험영상	$N_F$	$T_d$ (sec)	$T_s$	R
stefan	270	35	10	29%
coastguard	270	37	10	27%
total	540	72	20	28%

제안 방법은 객체에 기반한 입체 변환을 수행함으로써 기존의 영상 기반 동영상의 입체 변환시 생기는 문제점을 해결한다. 이를 위해 배경 객체와 주객체에 대해 독립적인 입체 변환 방법을 적용하였다. 먼저 배경 객체에 대해서는 현재 영상과 지연 영상의 배경 객체를 이용한 적절한 좌우 영상 결정을 적용하였다. 좌우 영상 결정은 배경 객체의 MVF를 분석함으로써 결정되는 카메라 운동 유형에 좌우된다. 주객체에 대해서는 현재 영상의 주객체만을 사용함으로써 수직시차가 발생하지 않고 항상 음의 시차가 생기게 하였다. 이때 발생하는 주객체 영역의 빈 공간을 배경으로 채우기 위해 URFA를 적용하였다.

향후 계획으로는 영상 내에 다수의 객체가 존재하는 MPEG-4 동영상을 입체 변환하는 방법에 대한 연구가 필요하다.

참고 문헌

- [1] D. F. McAllister (editor), Stereo computer graphics and other true 3D technologies, Princeton, NJ: Princeton University Press, 1993.
- [2] B. J. Garcia, "Approaches to stereoscopic video based on spatial-temporal interpolation," SPIE Vol. 2635, Photonic West, 1990.
- [3] T. Okino and et. al, "New television with 2D/3D image conversion technologies," SPIE Vol, 2653, Photonic West, 1995.
- [4] M. Kim and et al., "Stereoscopic conversion of monoscopic video by the transformation of vertical-to-horizontal disparity", SPIE Vol. 3295, Photonic West, Jan, 1998.
- [5] Man Bae Kim and Sang Hun Lee, "A new method for the conversion of MPEG encoded data into stereoscopic video", J. of the Society for 3D Broadcasting and Imaging, Vol 1. No. 1, pp. 48-59, June 2000.
- [6] 김란배, 박상훈, "키 프레임을 이용하는 2차원 동영상의 3차원 입체 변환", HCI 학회, 2002.2.
- [7] 미키 스케이치, 고성제(역), "MPEG-4의 세계", 영풍문고, 1999.
- [8] ISO/IEC JTC1/SC29/WG11, "Coding of Moving Pictures and Audio", 2000.

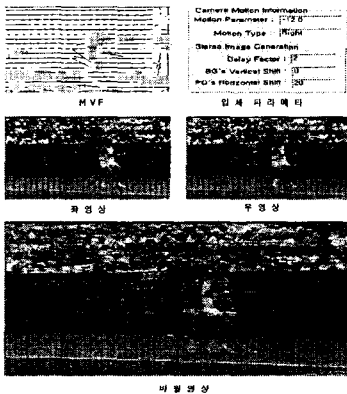


그림 5. stefan 동영상의 7번째 영상

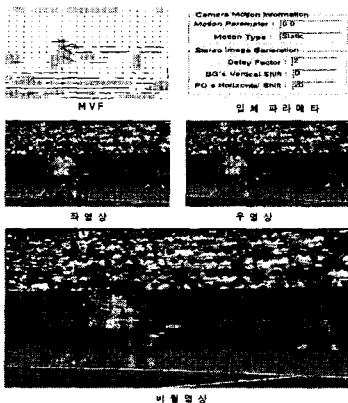


그림 6. stefan 동영상의 42번째 영상

4. 결론

본 논문에서는 배경객체와 주객체로 부호화된 MPEG-4 동영상을 입체 동영상으로 변환하는 방법에 대해 제안하였다.