2차원 동영상으로부터 객체 기반의 3차원 입체 변환 기법

한효정[O] 변혜란
연세대학교 컴퓨터과학과
{sylvanus[O], hrbyun}@cs.yonsei.ac.kr

Object-based Stereoscopic Conversion From a Monoscopic Video

Hyojung Han[0], Hyeran Byun
Yonsei University

요 약

객체 기반의 3차원 입체 변환 기법은 연속적으로 입력되는 2D 동영상에서 객체를 추출하여 입체 영상으로 변환하는 기법을 말한다. 두 눈에 투시되는 각 객체마다 서로 다른 시차를 가져야 입체감을 느낄 수 있다. 따라서 2D 영상에서 정확한 객체를 추출하는 것이 중요하다. 본 논문에서는 프레임간의 차이를 이용하여 대략의 움직이는 객체 영역을 얻고, 그래프 컷 알고리즘을 사용하여 정확하고 안정적인 객체를 자동으로 추출한다. 스크린과 양안 사이의 거리를 고려하여 입체 영상을 만들어 낸다. 후처리 단계에서는 입체 영상을 만들어 내면서 생긴 빈 공간을 채운다. 실험에서는 2D 영상으로부터 입체 영상을 생성한 것을 보여 준다.

1. 서 론

우리가 물체를 볼 때 왼쪽 눈과 오른쪽 눈에 조금의 차이가 있는 영상이 맺히게 된다. 이것을 양안 시차(binocular disparity)라 한다. 뇌에서 양안 시차가 있는 두 영상을 종합하여 한 영상으로 보여주면서 우리는 입체감을 느끼게 된다.

기존의 입체 영상은 스테레오 카메라나 이미지 편집 툴을 이용하여 획득하였다. 하지만 스테레오 영상 입력기기를 이용한 제작에는 한계가 있고, 이미지 편집 툴에 의한 편집은 시간이 많이 걸린다. 따라서 기존의 2D 동영상을 입체 영상으로 변환하여 볼 수 있다면 보다 다양한 컨텐츠 제작에 도움이 될 수 있을 것이다.

스테레오스코픽 변환과 관련된 지난 연구를 살펴보면 Okino 그룹[1]의 MTD(Modified Time Difference)를 이용한 방법, Garcia[2]의 인간의 시각 특성인 공간 시간 보간(spatial-temporal interpolation)을 이용한 방법, Matsumoto[3] 영상의 깊이 정보를 이용한 방법 등이 있었다. 하지만 기존의 방법들은 이미지 변형에 의해 영상의 화질이 떨어졌다. 또한 움직이는 물체의 속도와 방향을 결정해야 했고, 수직 운동인지 수평 운동인지 구분하여야 했다. 하지만 수직 운동의 영상이나 정지한 영상에서도 양안 시차가 존재할 경우 입체적으로 볼 수 있다. 따라서 본 논문은 객체의 움직임에 상관없이 객체를 입체적으로 보는 방법을 제안하였다.

본 논문의 구성은 다음과 같다. 2절에서 중요 객체를 자동적으로 추출하는 방법에 대하여 소개한다. 3절에서는 추출한 객체를 이용하여 입체 영상을 생성하는 방법에 대하여 설명한다. 4절에서는 기술한 방법의 결과 영상을 보여준다. 5절에서는 결론을 맺는다.

2. 중요 객체 자동 추출

본 논문에서는 중요 객체를 움직이는 객체로 본다. 먼저 프레임 차이를 이용하여 대강의 객체 영역을 획득한 후 Graph cuts 알고리즘을 이용하여 정교한 객체를 자동으로 추출한다.

2.1 프레임 차이를 이용한 객체 분포 획득

비디오 데이터의 연속적인 두 개의 입력 영상을 임계값(Threshold value)에 의해 구분하는 것은 변환 검출(change detection)의 기본 개념이다. 그러나 움직이는 객체의 행동, 배경 잡음, 객체와 배경과의 색상 대비와 같은 문제가 있기 때문에 프레임 차이(Frame Difference)만을 가지고 정확한 객체를 얻어 낼 수 없다.

본 논문에서는 다른 방법을 쓰지 않고 프레임 차이만을 가지고 대강의 객체 분포를 획득하였다.

임계값을 얻기 위해서 유의성 검정 기법(Significance test technique)을 사용한다.[4] 검정할 확률은 프레임 차이의 절대값이 된다. 영상의 픽셀은 변화가 없고 프레임 차이는 평균이 0인 정규 분포를 따른다고 가정한다. 확률 분포 함수는 아래 식과 같다.

$$p(FD \mid H_0) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{FD^2}{2\sigma^2}\right)$$

위의 식에서 FD는 프레임 차이이고, σ^2는 프레임 차이의 분산을 의미 한다. H_0는 영가설을 나타내고, 영상의 픽셀은 변화가 없다는 것을 의미한다.

$$\alpha = Prob(|FD| > TH \mid H_0)$$

임계값은 유의 수준(significance level)에 의하여 결정된다. 두 개의 관계식은 수식 (2)와 같다. 여기서 α는 유의 수준이고 TH는 임계값이다. 그림 1은 프레임 차이에 의해 나온 결과의 예이다.

(a) 원영상 (b) FD 영상 (c) 원영상 (d) FD 영상
그림 1. "akiyo"와 "claire" FD 실험 결과 영상

2.2. 형태학적 알고리즘에 의한 영역 구분

프레임 차이에 의해 나온 결과는 픽셀 단위의 계산이기 때문에 그림 1에서 보는 바와 같이 영역으로 구분할 수 없다. 의미 있는 영역을 만들어 내기 위해서 기본적 형태학적 알고리즘인 모폴로지 연산과 영역 채움 알고리즘을 이용하였다. 본 논문에서는 확실한 전경 영역, 확실한 배경 영역, 불확실 영역 3영역으로 구분하였다.

그림 2의 (b)와 (d)에서 흰색은 전경, 검은색은 배경, 회색은 불확실 영역을 나타낸다.

(a) 원영상 (b) 결과 영상 (c) 원영상 (d) 결과 영상
그림 2. "akiyo"와 "claire" 결과 영상

2.3. 그래프 컷을 이용한 정교한 객체 추출

그래프 컷 방법론은 Boykov[6] 등에 의해 제안되었다. 사용자가 직접 배경 부분과 전경 부분을 지정하여서 그 색상 분포를 이용하여 배경과 전경을 분리하였다.

본 논문에서는 사용자 인터랙션 없이 연속적인 비디오 데이터 영상에서 움직이는 객체의 영역을 자동으로 추출하여 객체와 배경을 분리하도록 하였다. 우선 이미지 내의 화소의 집합을 P라 하고, 두 인접 화소 $\{p,q\}$의 전형적인 8-이웃 시스템(standard neighborhood system)을 N이라고 한다. 영상을 n으로 인덱스하고 그레이(grey) 값을 $z=(z_1,\cdots,z_{|M|})$로 나타낸다. 집합 P의 픽셀 p에 대한 0과 1의 이진 레이블을 θ_p라 한다. 여기서 1은 객체, 0은 배경을 의미한다. 따라서 $\theta=(\theta_1,\theta_2,\ldots,\theta_{|R|})$ 은 분할된 객체를 의미한다.

에너지 함수(energy function)는 다음과 같다.

$$E(\theta)=\lambda\cdot D(\theta)+B(\theta)$$

여기서 계수 λ은 데이터 항목(data term) $D(\theta)$의 중요도를 나타낸다. $D(\theta)$와 경계선 항목(Boundary term) $B(\theta)$은 수식 (4)와 같이 정의 한다.

$$D(\theta)=\sum_{p\in P}D_p(\theta_p)$$

$$B(\theta)=\sum_{p,q\in N}\delta(\theta_p,\theta_q)\cdot B_{p,q}(\theta_p,\theta_q)$$

$$D_p(\theta_p)=\begin{cases}-\ln\ ob(z_p|\theta_p) & \text{if } p\in U\\ (K-c)\theta_p+c & \text{if } p\in O\\ (c-K)\theta_p+c & \text{if } p\in B\end{cases}$$

$$B_{p,q}(\theta_p,\theta_q)=dist(p,q)^{-1}\cdot\exp\left(-\frac{(I_p-I_q)^2}{2\sigma^2}\right)$$

수식 (4)에서 O B U는 형태학적 알고리즘 이후에 나온 전경 영역 O 배경 영역 B 불확실 영역 U를 의미한다. 델타 함수 $\delta(\theta_p,\theta_q)$는 $\theta_p\neq\theta_q$일 경우에 1을 할당하고, 나머지 경우에는 0을 할당한다. 여기서 K는 아래와 같다.

$$K=1+\max_{\{p,q\}\in N}\sum B_{p,q}$$

상수 c는 보통의 경우 1로 놓는다. 입력된 이미지는 대부분 객체와 배경 사이에는 높은 대비(contrast)가 나타나는 특징을 가진다. 따라서 식은 대비를 줄 수 있는 수식 (4)의 $B_{p,q}(\theta)$와 같아야 한다.

본 논문에서는 맥스-플로우(max-flow) 알고리즘[5]을 이용하여 Graph cuts을 구현 하였다. 그림 3을 보면 원영상에서 객체가 추출된 모습을 볼 수 있다.

(a) (b) (c) (d)
그림 3. (a)와 (c)는 원영상, (b)와 (d)는 객체 추출 영상

3. 입체 영상 생성

2차원 동영상에 3차원 입체 영상을 보기 위해서는 동일점이 좌영상과 우영상에 투시될 때 투시점들 간의 시차(parallax)가 있어야 한다. 양안으로 장면을 볼 때 두 눈에 투시되는 영상에 미묘한 차이가 있어 깊이감을 느낄 수 있는 것과 같은 원리이다.

본 논문에서는 스크린 보다 물체가 더 튀어 나와 보이는 문제(the screen surround problem)를 해결하기 위해서 양의 시차만을 고려하였다.[7] 따라서 객체는 스크린 뒤에 있고, 배경은 객체 뒤에 위치하게 된다.

3.1. 시차 계산

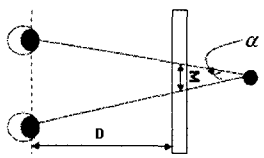


그림 4. 스크린과 시차와의 관계

그림 4에서 M은 좌안과 우안이 스크린에 투영되었을 때의 점 사이의 거리, D는 사용자와 스크린 사이의 거리, α는 양의 시차 각을 나타낸다.

$$\tan\left(\frac{\alpha}{2}\right)=\frac{\left(\frac{M}{2}\right)}{D}$$

수식 (6)은 스크린과 시차와의 관계식을 나타낸다. 시차각의 값이 1.5도를 초과 하면 입체 영상이 불안정하게 보인다. 시차 각 α 값이 1.0도에서 1.4도 사이로 보고, 스크린과 사용자의 거리 D 값은 0.8m에서 1m사이라 생각하여 최대 시차 값이 8픽셀을 초과하지 않도록 하였다.

3.2. 빈 홀 채우기

객체를 추출한 이후에 영상을 쉬프트(shift) 시키거나 변형을 주면 영상에 빈 홀이 생긴다.

본 논문에서는 빈 홀을 채우기 위해서 Criminisi의 Region Filling 방법론[8]을 사용하였다. 그림 5에서 녹색의 빈 홀이 비슷한 배경에 의해서 채워짐을 볼 수 있다.

그림 5. 단계별로 빈 홀이 채워지는 모습

4. 실험 및 결과

본 논문에서 제안한 방법은 Windows XP, Visual C++ 6.0을 이용하여 Pentium IV 2.4GHz 1GB RAM 상에서 구현되었다. 실험에서는 352×288 크기의 "akiyo" 동영상과 176×144 크기의 "claire" 동영상을 사용하였다.

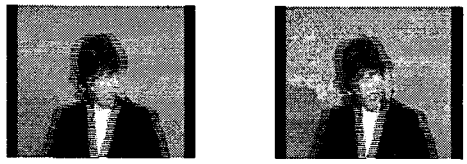
본 논문에서는 300 프레임의 "akiyo" 동영상과 494 프레임의 "claire" 동영상에서 자동으로 객체를 추출하여 입체 영상을 생성하는 것을 구현하였다.

그림 6의 (a)는 "akiyo" 영상을 객체를 추출한 후 8픽셀 쉬프트 시켜 입체 영상을 만들어 낸 것이고, (b)는 객체를 원점을 기준으로 Y축 방향으로 10도 회전시켜 Interlaced 방법으로 입체 영상을 만들어 낸 것이다.

그림 7은 "claire" 동영상 데이터를 8픽셀 쉬프트 시켜서 Interlaced 방법으로 입체 영상을 만든 것이다. (a)는 100번째 영상이고, (b)는 400번째 영상을 보여주고 있다.

(a) 쉬프트 (b) Y축 회전

그림 6. "akiyo"의 Interlaced 영상



(a) 100번째 프레임 (b) 400번째 프레임

그림 7. "claire"의 Interlaced 영상

5. 결론

본 논문은 2차원 동영상에서 객체를 추출하여 3차원 입체로 변환하는 기법에 대하여 제안하였다. 정교하게 객체를 추출하였기 때문에 사용자가 원하는 대로 객체를 조정할 수 있어 객체의 방향이나 속도에 상관없이 입체 영상을 볼 수 있다. 미래에 해야 할 일로는 다수의 객체 또는 객체의 특정 부분만을 추출하여 적당한 시차를 주어 입체 영상을 만드는 것이다.

감사의 글

본 연구는 광주과학기술원(GIST) 실감방송연구센터(RBRC)를 통한 대학IT연구센터(ITRC)의 지원을 받아 수행하였습니다.

참고 문헌

[1] T. Okino and et. al, "New television with 2D/3D image conversion techniques," SPIE, Vol. 2653, Photonic West, 1990.

[2] B. J. Garcia. "Approaches to stereoscopic video based on spatial-temporal interpolation,"SPIE, Vol. 2635, Photonic West, 1990.

[3] Y. Matsumoto, et al, "Conversion system of Monocular Image Sequence to Stereo using Motion Parallax," SPIE Photonic West, vol. 3012, pp 108-115, 1997

[4] T. Aach, A. Kaup, and R. Mester, "Statistical model based change detection in moving video," Signal Processing, vol. 66, pp. 203-217, Apr. 1998.

[5] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," In 3rd. Intnl. EMMCVPR. Springer-Verlag, September 2001, to appear.

[6] Y. Boykov and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images" In ICCV, vol. 1, pp. 105-112, 2001.

[7] C. H. Choi, B. H. Kwon, M.-B. Choi, "A real time field-sequential stereoscopic image converter" Consumer Electronics, IEEE Transactions on, vol. 50, pp. 903-910, Aug. 2004.

[8] A. Criminisi, P. Perez, K. Toyama "Region Filling and Object Removal by Exemplar-Based Image Inpainting" IEEE Transactions on Image Processing, vol. 13, no. 9, Sep, 2004.