

Optical Flow와 Normalized Cut을 이용한 2차원 동영상의 3차원 동영상 변환

정재현 · 박길배 · 김주환 · 강진모 · 이병호[†]

서울대학교 전기공학부

☎ 151-744 서울특별시 관악구 관악로 599

(2008년 12월 29일 받음, 2009년 2월 13일 수정본 받음, 2009년 2월 17일 게재 확정)

본 논문에서는 2차원 동영상을 normalized cut과 optical flow를 이용하여 3차원 동영상으로 변환하는 방법을 제안하였다. 이를 통해 특정 디스플레이 장치와 특정 동영상 포맷에 국한되지 않는 2차원 동영상의 3차원 동영상 변환 방법을 제안하였다. 본 연구에서는 2차원 동영상의 3차원 변환을 위하여 먼저 영상을 객체로 분할하고, 분할된 객체의 깊이를 추정하는 방법을 사용하였다. Normalized cut은 영상분할의 한 방법으로, 본 연구에서는 연산속도 향상을 위하여 기존 방법에 watershed 알고리즘을 적용하였고, 정확도 향상을 위하여 가중치에 optical flow를 추가하였다. Normalized cut을 이용하여 분할된 영상의 깊이 정보를 추정하기 위하여 optical flow를 이용하였다. Optical flow의 차이를 통해 정의할 수 있는 가려진 영역의 분할 영상 변화를 통해 순서적 깊이 정보를 추정한다. 추정된 순서적 깊이를 보정하기 위해 optical flow의 절대적 크기를 이용해 운동시차로 상대적 깊이를 추정하였다. 최종적으로 추정된 깊이 정보는 순서적 깊이와 상대적 깊이의 곱을 평균 optical flow로 나누어, 순서적 깊이의 차이를 보정하였다. 제안한 방법의 검증을 위하여 2차원 동영상을 3차원 동영상으로 변환하여 깊이 정보가 추정됨을 확인하였다.

주제어: Three-dimensional display, 2D/3D conversion, Optical flow, Normalized cut

I. 서 론

최근 2차원 디스플레이 기술 개발이 포화 상태에 다다름에 따라, 3차원 디스플레이 기술이 차세대 디스플레이로서 큰 주목을 받고 있다.^[1] 현재 많은 연구그룹으로부터 다양한 형태의 3차원 디스플레이가 발표되고 있지만 상용화를 위해서는 아직 해상도, 시야각, 관찰자의 피로 등 해결해야 할 많은 문제점을 안고 있다. 그 중에서도 가장 큰 문제점은 3차원 디스플레이용 콘텐츠의 부족이다. 현재 방송이나 영화와 같은 영상매체 시장의 경우 대부분이 2차원 콘텐츠로, 이와 같은 시장환경하에서 3차원 디스플레이를 상용화하는 경우 콘텐츠의 부족으로 사용자의 요구를 충분히 만족시킬 수 없고, 3차원 디스플레이에 대한 부정적인 인식을 키울 수 있다. 따라서 본격적인 3차원 디스플레이 시장이 활성화 되기 전까지 이러한 문제점을 극복하기 위한 2차원/3차원 동시 재생이 가능한 디스플레이, 2차원 콘텐츠의 3차원 변환과 같은 중간 자적 기술이 크게 주목 받고 있다.^[2-7]

현재까지 진행된 2차원 콘텐츠의 3차원 변환에 관한 연구는 주로 동영상 위주로 발전하였으며, 2차원 동영상의 연속된 2개의 프레임 전체를 비교하여 좌우안에 시간지연을 준 영상을 보여주어 입체로 느끼게 하는 방법을 주로 사용하였다.^[4-7] 이는 Ross 현상으로, 사람의 두 눈이 시간적으로 지연된 2차원 영상을 볼 경우, 3차원으로 인식한다는 정신물리학 이론에 기초하고 있다.^[8] 이를 응용한 2차원 콘텐츠의 3차원 변환에 관한 연구는 Okino 그룹에서 제안한 MTD(Modified Time Difference)를 이용한 방법,^[4] Garcia의 공간 시간 보간을 이용한 방법,^[5] Matsumoto의 운동시차를 이용한 방법^[6]이

있다. 하지만 기존에 제안된 방법들은 블록 매칭을 통해 운동정보를 얻음으로써 정확성이 떨어지고, 콘텐츠의 변형에 의해 화질이 저하되는 문제점이 있다. 또한 시간지연 영상을 보여주는 방법을 이용하므로, 패럴랙스 배리어나 렌티큘러와 같이 수평시차만 존재하는 3차원 디스플레이에 특화되어 집적영상과 같이 수직시차도 존재하는 3차원 디스플레이에 적용하기 어려운 문제점이 있다.

본 연구에서는 이러한 문제점들을 극복하고자 기존의 기준영상과 시간지연을 준 영상을 좌우안에 보여주어 입체로 보이게 하는 방법 대신, 기준영상과 시간지연영상의 optical flow를 이용하여 정확한 움직임 정보를 얻고, 이를 이용하여 영상을 객체별로 분할하여 각 객체의 깊이 정보를 추출하는 방법을 통해 깊이 정보를 완성하도록 한다. 제안한 방법을 이용할 경우, 깊이 정보를 객체별로 알 수 있어 정확성 높은 3차원 콘텐츠로 변환이 가능하며, 수평 시차뿐 아니라 수직 시차도 표현 할 수 있어 다양한 형태의 3차원 디스플레이에 적용이 가능하다. 또한 특정 동영상 포맷에 국한되지 않고, 연속적인 영상 프레임을 가지는 모든 동영상 포맷에 적용할 수 있어 활용의 폭이 넓다. 이를 위하여, 제안된 방법에서는 optical flow와 normalized cut을 이용하여 영상을 객체별로 분할하고, 분할된 객체간의 깊이 정보를 추정한다. 깊이 정보를 추정하기 위해 optical flow를 이용하여 상대적인 깊이 정보를 추정하고, optical flow의 차이 값을 이용하여 가려진 영역을 계산, 객체 사이의 선후 관계를 추정하여 최종 깊이 정보를 완성한다.

본 논문의 구성은 다음과 같다. 우선 논문의 II장에서는 2차원 동영상을 3차원 동영상으로 변환하기 위하여 먼저 수행한 normalized cut을 이용한 객체 분할, 정확성 향상을 위

[†]E-mail: byoungcho@snu.ac.kr

한 optical flow의 사용, 속도 향상을 위한 watershed 알고리즘의 사용에 대해 다룬다. III장에서는 분할된 객체 영상을 바탕으로 optical flow를 이용하여 가려진 영역을 정의하고 순서적 깊이와 상대적 깊이를 추정하여 깊이 정보를 완성하는 과정을 구체적으로 논한다. IV장에서는 실험을 통하여 2차원 동영상의 3차원 변환 결과를 제시한다.

II. Normalized cut과 optical flow를 이용한 객체 분할

본 연구에서 제안한 optical flow와 normalized cut을 이용한 2차원 동영상의 3차원 변환 과정은 그림 1과 같다. 먼저 2차원 동영상에서 시간적으로 연속인 2장의 프레임 영상을 추출하여 시간적으로 앞서는 영상을 기준영상 F_{ref} 로 정의하고, 시간적으로 연속인 영상을 F_d 로 정의한다. 본 연구에서는 전체 프레임 정보를 바탕으로 변환하는 기존의 2차원 콘텐츠의 3차원 변환 과정과 달리, 프레임을 배경과 각 객체들로 분할하고, 분할된 객체간의 상대적 깊이와 순서적 깊이를 이용하여 3차원 동영상으로 변환한다. 따라서 먼저 normalized cut과 optical flow를 이용하여 2차원 콘텐츠를 객체들로 분할하는 과정이 필요하다.

Normalized cut은 그래프 분할 알고리즘의 하나로 전체 영상 분할을 그래프 분할에 대응시켜 고유벡터를 구하는 문제로 일반화하는 알고리즘이다.^[9] 전체 영상은 식 (1)과 같이 각각의 픽셀을 노드 i, j 픽셀 사이의 유사도 w_{ij} 를 에지로 가지는 그래프 G 로 표현할 수 있다.

$$G = \{(i, j), w_{ij}\} \quad (1)$$

이 때 이 그래프를 영상에 반영하게 되면, 영상의 각 픽셀의 밝기, 색상 등과 같은 픽셀의 특징을 나타내는 값이 노드가 되고, 각각의 픽셀 사이의 유사도가 에지가 되어 그래프의 분할을 이용하여 영상을 분할할 수 있다. 그래프 내에서의 분할 $cut(A, B)$ 는, 식 (2)와 같이 각 노드 i, j 간의 가중치

$w(i, j)$ 의 합으로 나타낼 수 있으며, 가장 간단한 그래프 분할법은 이 분할 값이 가장 작은 영역별로 분할하는 것이다.

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j) \quad (2)$$

하지만 이 경우 외따로 떨어져 있는 노드의 경우 가중치가 가장 낮아 외따로 떨어진 노드만으로 분할되어 정확한 분할이 이루어지지 않는 단점이 있다. 이 점을 보완한 방법이 normalized cut이다. Normalized cut을 정의하기 위해서는 먼저 식 (3)과 같이 집합 A 에 속하는 모든 정점에서 그래프에 존재하는 모든 정점의 집합 V 까지 가중치의 합 $assoc(A, V)$ 를 구할 필요가 있다. 최종적으로 normalized cut은 식 (4)와 같은 형태를 가지게 된다. 앞선 분할의 최소값으로 영상을 분할할 때와 달리, 영상을 A, B 로 분할함에 있어 $cut(A, B)$ 를 각각 $assoc(A, V), assoc(B, V)$ 로 나누어 정규화 함으로써, 고립된 노드의 $cut(A, B)$ 는 작은 값을 가지지만, 전체 $Ncut(A, B)$ 값은 보다 큰 값을 가지게 되어 잘못된 분할을 방지해 준다. 따라서 $Ncut(A, B)$ 가 가장 작은 값을 가지게 되는 A, B 를 찾아 분할하여 영상 내에서의 객체를 분할해 낼 수 있다.

$$assoc(A, V) = \sum_{i \in A, k \in V} w(i, k) \quad (3)$$

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (4)$$

하지만 normalized cut이 정확한 분할이 이루어지기 위해서는 노드간의 유사도를 잘 표현할 수 있는 가중치의 선정이 중요하다. 기존의 normalized cut을 이용한 객체분할에서 주로 사용한 가중치는 픽셀간의 밝기 차이, 픽셀간의 거리 차이이다. 이를 식으로 나타내면 식 (5), (6)과 같다. 이때 F_i 는 i 번째 픽셀에서의 밝기를 나타내고, σ 는 전체 영상의 밝기에 따른 표준편차를 의미한다. 그러나 픽셀간의 밝기 차이만으로 가중치를 설정하여 분할하는 경우, 동일한 물체 내에서도 조명에 따라 밝기가 변하는 경우가 있으므로 절대적이라 할 수 없다. 따라서 특정 범위 내에서의 픽셀간의 인접 정도를 식 (6)과 같이 가중치로 설정한다. r 범위 내의 픽셀 중에서, X_i 는 i 번째 픽셀의 위치를 나타내고, σ_p 는 전체 위치의 표준편차를 의미한다. 거리상의 인접도를 이용한 가중치는, 밝기가 차이가 나는 영역이라 하더라도 거리상으로 인접한 경우에 대해 보완해주는 역할을 한다.

$$w_{ij_brightness} = \exp\left(\frac{-|F_i - F_j|^2}{\sigma_i^2}\right) \quad (5)$$

$$w_{ij_position} = \begin{cases} \exp\left(\frac{-|X_i - X_j|^2}{\sigma_p^2}\right) & , |X_i - X_j| < r \\ 0 & , otherwise \end{cases} \quad (6)$$

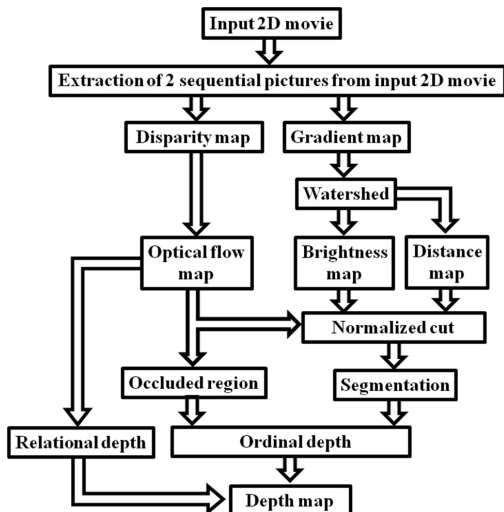


그림 1. 제안된 2차원 동영상의 3차원 동영상 변환과정.

그러나 픽셀간의 밝기 차이와 인접한 정도만으로 normalized cut을 수행하는 경우, 영상 내의 물체의 형태가 복잡한 경우에는 원활한 결과를 보여주지 못한다. 이를 보정하고자 본 연구에서 새롭게 적용한 가중치가 optical flow이다.

Optical flow는 동영상과 같은 시간적으로 연속적인 영상에서 동일한 객체의 밝기 값이 일정하게 유지된다는 가정 하에 시공간 미분방정식을 풀어 객체의 움직임을 찾아내는 알고리즘이다.^[10] Optical flow는 움직임 정보를 추정할 수 있는 블록 매칭과 같은 알고리즘에 비해 정확도가 높다는 장점이 있다. 본 연구에서는 시간적으로 연속적인 두 영상 사이의 optical flow를 구하여 동일한 객체에 속하는 픽셀의 경우 optical flow의 크기도 유사하다는 점을 이용하여 식 (7)과 같이 가중치로 활용한다. 단, optical flow는 카메라 시점이 정지해 있거나 일정하게 이동하는 경우에만 물체의 움직임이 일정한 값으로 측정되고, 영상 분할의 기준으로 사용할 수 있으므로 본 연구에서는 카메라 시점이 일정하다고 가정하고 분할의 가중치로 사용하였다. 이 때, $d(i, j)$ 는 i 번째 픽셀과 j 번째 픽셀의 움직임 유사도, σ_0 는 전체 optical flow의 표준편차를 의미한다.

$$w_{ij_opticalflow} = \exp\left(\frac{-d(i, j)}{\sigma_0^2}\right) \quad (7)$$

Optical flow를 가중치로 활용하기 위해서는 먼저 i 번째 픽셀과 j 번째 픽셀간의 움직임 유사도 $d(i, j)$ 를 정의해야 한다. 움직임 유사도는 motion vector를 이용한 움직이는 객체 추출 시 사용되는 가중치 식을 인용하였다.^[11] 픽셀간의 optical flow를 \vec{v}_i, \vec{v}_j 라 할 때, 움직임 유사도는 두 벡터가 같은 크기의 벡터가 동일한 방향으로 진행할 때 가장 작은 값이 되고,

반대의 경우 최대값이 되도록 식 (8)과 같이 정의된다.

$$d(i, j) = \begin{cases} 1 & , |\vec{v}_i| \text{와 } |\vec{v}_j| \text{ 둘 다 } 0 \text{인 경우} \\ |\vec{v}_i| + |\vec{v}_j| & , |\vec{v}_i| \text{와 } |\vec{v}_j| \text{ 둘 중 하나만 } 0 \text{인 경우} \\ \left((|\vec{v}_i|+1)(|\vec{v}_j|+1) \left(\frac{1-\cos\theta}{2} \right) \right) & , |\vec{v}_i| \text{와 } |\vec{v}_j| \text{ 둘 다 } 0 \text{이 아닌 경우} \end{cases} \quad (8)$$

이들 가중치를 활용하여 normalized cut을 수행하기에 앞서, 입력 동영상의 전체 픽셀을 그래프의 노드로 설정하는 경우 연산량이 해상도에 따라 반대해지므로, 영상의 밝기, 색상을 고려하여 watershed 알고리즘으로 초기 영역 분할을 통해 연산량을 줄이도록 한다.^[12] Watershed 알고리즘은 영상의 gradient 값을 이용하는 알고리즘으로 영상의 에지를 구분하고, 특히 밝기 정보의 변화가 큰 부분에서 영상을 분할해 내는 특징을 가진다. 따라서 추출해 낸 2장의 영상을 바탕으로 gradient map을 구하고 watershed 알고리즘을 적용하면 그림 2와 같다. 픽셀을 하나의 노드로 가정하던 기존의 방법에 비해, Watershed로 분할된 영역을 노드로 정의하고 각 영역의 밝기의 평균, 픽셀 위치의 평균을 할당해 주면 normalized cut의 연산량을 크게 줄여 줄 수 있다. 연속된 입력 영상 F_{ref} 와 F_d 로부터 optical flow를 구하고 밝기, 위치와 함께 가중치로 적용하여 watershed로 분할된 영상에 normalized cut을 수행하면 그림 3과 같이 객체가 분할된 영상을 얻을 수 있다.

III. Optical flow를 이용한 객체의 깊이 추정

앞 장에서 normalized cut을 이용하여 분할한 객체에 적합

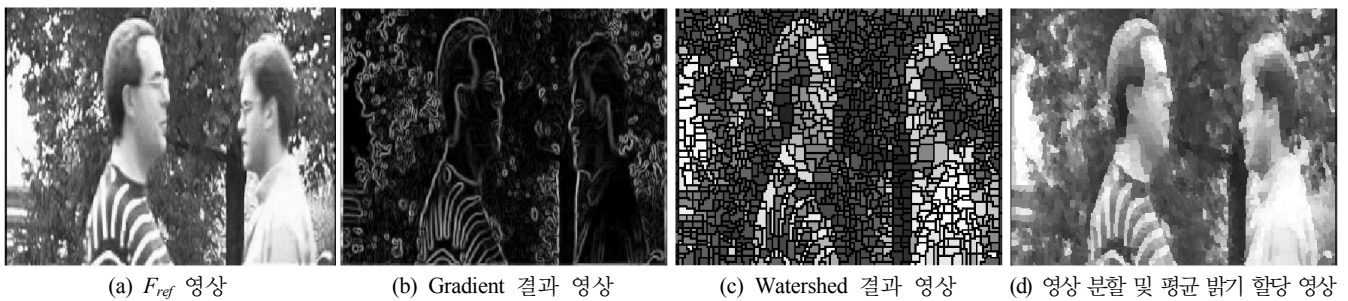


그림 2. Watershed 알고리즘을 이용한 영상 분할 결과.

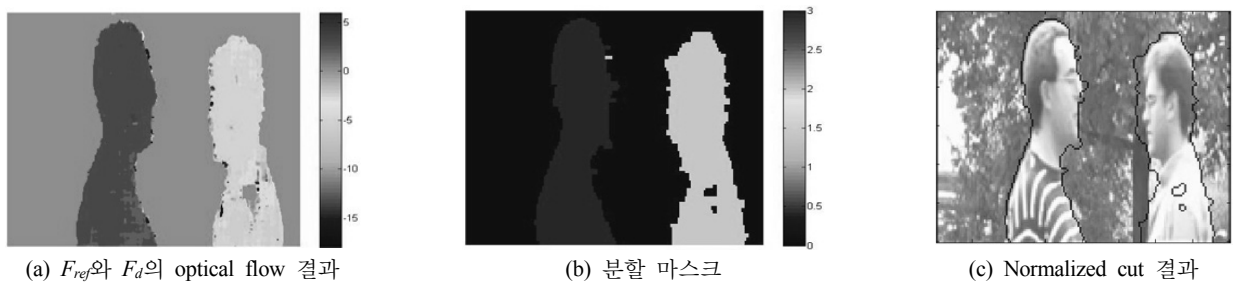


그림 3. Normalized cut을 이용한 영상 분할 결과.

한 깊이 정보를 추정하기 위해 optical flow 알고리즘을 사용한다. 먼저 각 객체의 순서적인 깊이를 추론하기 위해서는 optical flow를 이용한 가려진 영역의 정의가 필요하다. 가려진 영역은 optical flow가 연속되는 두 영상 사이의 동일한 밝기를 가지는 픽셀의 이동을 나타내는 것에 착안한 것으로, 첫 프레임 영상 F_{ref} 에서 보이는 픽셀이 두 번째 프레임 영상 F_d 에서는 보이지 않게 되는 영역을 나타낸 것이다.^[13] 이를 구해내는 방법으로 optical flow의 차이를 이용한다.

순서적인 깊이를 추정하기 위해서 먼저 배경 영역을 찾아낸다. 본 연구에서는 카메라 시점이 일정한 영상의 경우로 한정하였으므로, 배경 영역은 optical flow가 0에 가까운 영역으로 한정할 수 있다. 반면 움직이는 객체의 경우 짧은 시간 내에서도 optical flow의 크기를 통해 이동하는 정도와 그 객체의 영역을 찾아낼 수 있다. F_{ref} 와 F_d 두 영상 사이에 구할 수 있는 optical flow는 두 가지로, 실제 영상의 시간 흐름 순서인 F_{ref} 에서 F_d 로 진행해가는 optical flow U_{12} 와, 반대로 F_d 에서 F_{ref} 로 진행해가는 optical flow U_{21} 이다. 따라서 이 optical flow들의 절대값의 차이를 구해보면 그림 4와 같다. F_{ref} 에서는 보이다 F_d 에서는 보이지 않는 영역을 O_{12} , 그 반대로 F_{ref} 에서는 보이지 않다가 F_d 에서는 보이는 영역을 O_{21} 이라 할 때, O_{12} 는 $|U_{21}| - |U_{12}|$ 가 0보다 큰 영역이고, O_{21} 은 $|U_{21}| - |U_{12}|$ 가 0보다 작은 영역이다. 따라서 그림 5와 같이 가려지는 영역을 분할정보와 함께 이용하면 기본적인 객체 전후 관계를 예측할 수 있다.

그림 5의 경우와 같이 한 프레임 내에 좌측으로 이동하는 객체와 우측으로 이동하는 객체가 존재할 때, 가려진 영역내의 객체가 F_{ref} 에서는 좌측으로 이동하는 객체였고, F_d 에서는 우측으로 이동하는 객체라면, 좌측으로 이동하는 객체보다

우측으로 이동하는 객체가 더 앞쪽에 존재함을 추정할 수 있다. 따라서 optical flow의 차이를 통해 정의된 가려진 영역의 객체변화를 통해 그림 5(d)와 같이 순서적인 깊이를 추정할 수 있다.

순서적 깊이의 추론을 보다 일반화 시키면 그림 6과 같다. 그림 6에서와 같이 각각 원과 사각형의 객체가 화살표 방향으로 이동하는 경우, 눈금으로 표시된 가려진 영역내의 객체

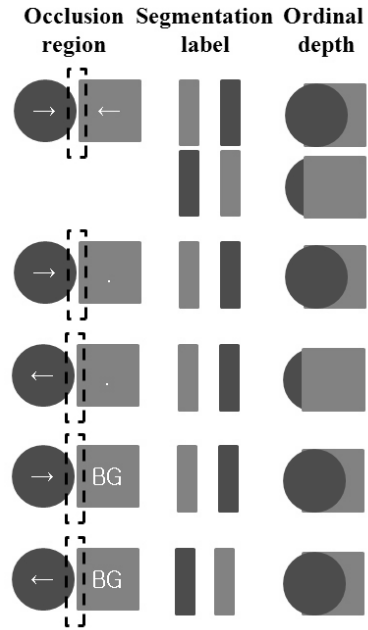


그림 6. 가려진 영역 내에서 객체 변화에 따른 상대적 깊이의 전후 관계.

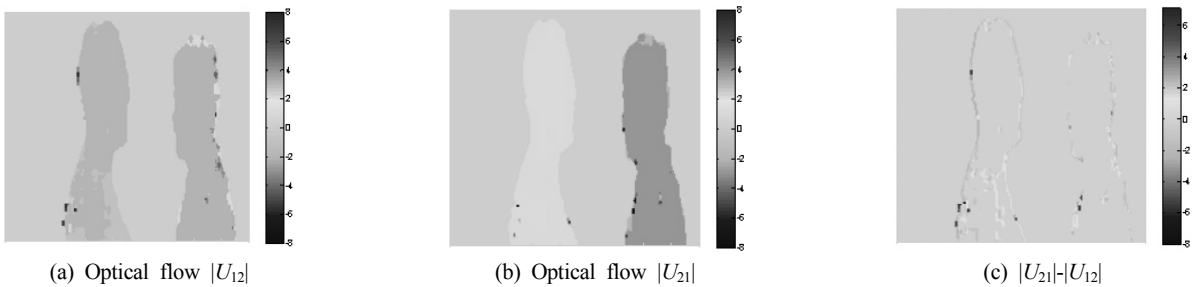


그림 4. Optical flow를 이용한 가려진 영역.

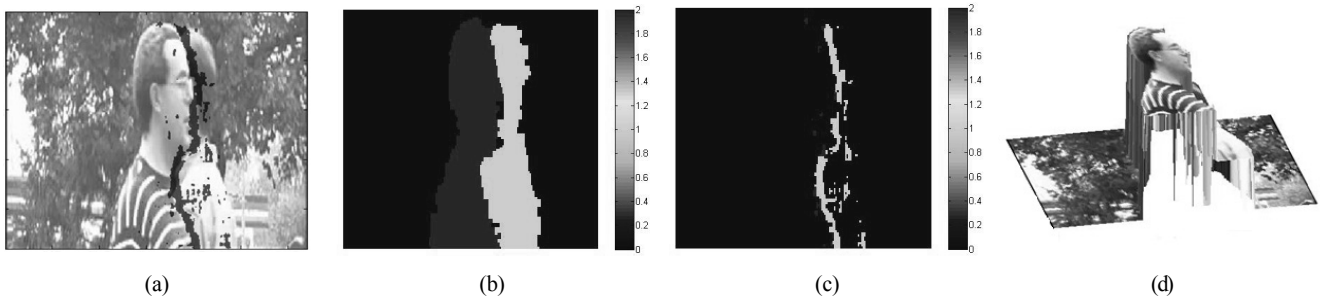


그림 5. 순서적 깊이 추정 (a) Optical flow를 이용한 가려진 영역 (b)객체 분할 정보 (c)가려진 영역 내의 객체 정보 변화 (d) 추정된 순서적 깊이.

변화를 통해 원과 사각형의 상대적인 깊이를 추정할 수 있다. 예를 들어 첫 번째 예와 같이 원과 사각형이 좌우에서 다가오는 경우, 가려진 영역의 객체가 사각형에서 원으로 바뀌는 경우에는 원이 더 앞쪽에 존재하고, 그 반대의 경우는 사각형이 더 앞쪽에 존재하는 것으로 추론 할 수 있다. 따라서 가려진 영역내의 객체 변화가 존재한다면 그림 6과 같이 모든 경우에 대한 순서적 깊이가 추정 가능하다. 식 (9)와 같이 A객체의 순서적 깊이가 정보는, 배경 객체를 0으로 할 때 객체가 깊이 방향으로 배열되는 순서.

$$D_{ordinal}(A) = Segment\ order(A) \quad (9)$$

그러나 순서적 깊이만으로는 객체의 전후 관계만을 파악할 수 있고, 얼마나 앞쪽에 존재하고 뒤쪽에 존재하는지 깊이 정보를 정확하게 표현할 수 없다. 따라서 객체 사이의 전후 정도를 보정해 줄 수 있는 상대적 깊이가 필요하게 된다. 상대적 깊이는 기본적으로 운동시차에 착안하여 계산한다. 운동시차는 관찰자 가까이 존재하는 객체가 먼 곳에 존재하는 객체보다 더 빨리 이동하는 것으로, 객체가 실제로는 동일한 속도로 움직이고 있어야 한다는 제약 조건이 있다. 그러나 주된 객체간의 깊이 정보는 순서적 깊이를 통해 얻고, 객체 사이의 상대적인 전후의 정도를 구해내는 값으로 운동시차를 사용하기에 적절하므로 optical flow를 이용하여 상대적 깊이를 얻는다. 상대적 깊이는 객체의 이동속도가 기준이므로, 식 (10)과 같이 객체 분할 과정을 통해 얻은 각 객체의 평균 optical flow로 정의하였다. 이 때, $n(A)$ 는 객체 A의 총 픽셀 수이며, U_i 는 픽셀 i 에서의 optical flow이다.

$$D_{relational}(A) = \frac{1}{n(A)} \sum_{i \in A} |U_i| \quad (10)$$

식 (10)을 이용하여, 그림 5의 평균 optical flow를 계산한 후 상대적 깊이를 나타내면 그림 7과 같다. 그림 7에서 보는 바와 같이 상대적 깊이는 그림 5의 순서적 깊이와 달리 우측으로 이동하는 객체가 좌측으로 이동하는 객체보다 더 뒤쪽에 존재하는 것으로 나타남을 알 수 있다. 즉 객체의 이동속도는 우측으로 이동하는 객체가 더 빠르다는 것을 의미한다. 이러한 오차를 수정하기 위해 상대적 깊이는 순서적 깊이가 존재할 때, 객체 사이의 상대적 깊이 보정에 활용하도록 한

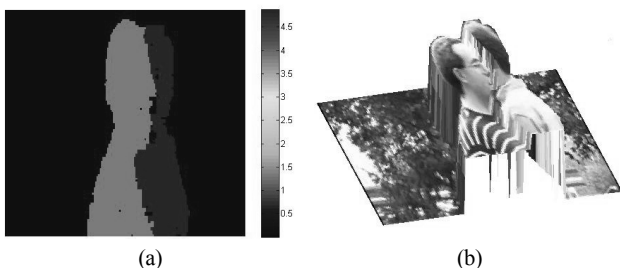


그림 7. 상대적 깊이 추정 (a) 객체의 평균 optical flow (b) 추정된 상대적 깊이.

다. 즉, 깊이 정보 $D(A)$ 는 식 (11)과 같이 순서적 깊이에 상대적 깊이를 전체 optical flow 크기의 평균으로 나눈 값을 곱하여 표현한다. 영상 전체가 n 개의 픽셀로 구성된 경우, optical flow의 평균 크기를 $U_{average}$ 라고 정의할 때, 깊이 정보는 객체 A의 순서적 깊이 $D_{ordinal}(A)$ 와 상대적 깊이 $D_{relational}(A)$ 의 곱을 $U_{average}$ 로 나눈 것으로 정의한다. 따라서 객체가 전체 평균 optical flow 크기보다 더 빠르게 움직이는 경우 순서적 깊이에 1보다 큰 값의 상대적 깊이를 곱하게 되고, 반대의 경우는 1보다 작은 상대적 깊이를 곱하게 된다. 따라서 순서적 깊이의 정도를 조절해 주어 깊이 정보를 보정해 줄 수 있게 된다. 그러나 이때 구해진 깊이 정보 $D(A)$ 에 의한 객체의 순서가, 앞서 계산한 순서적 깊이의 순서와 달라지게 되는 경우, 최종 깊이 정보 $Depth(A)$ 는 순서적 깊이 정보 $D_{ordinal}(A)$ 가 되고, 그렇지 않은 경우 $D(A)$ 가 된다.

$$U_{average} = \frac{1}{n} \sum_i |U_i|$$

$$D(A) = D_{ordinal}(A) \times \frac{D_{relational}(A)}{U_{average}} \quad (11)$$

$$Depth(A) = \begin{cases} D(A) & , D(A), D(A) \text{ 배열 순서가 } D_{ordinal}(A) \text{와 같은 경우} \\ D_{ordinal}(A) & , D(A) \text{ 배열 순서가 } D_{ordinal}(A) \text{와 다른 경우} \end{cases}$$

앞선 과정을 통해 입력된 2차원 동영상의 연속된 프레임 영상들을 2프레임 단위로 처리하면 각각 상대적 깊이와 순서적 깊이를 얻을 수 있고 최종 깊이 정보를 추정할 수 있다. 이 경우 순서적 깊이를 알아낼 수 없는 구간의 깊이 정보를 추정 하는 것이 문제가 될 수 있다. 본 연구에서는 이 문제를 해결하기 위해 다음과 같은 과정을 사용한다. 우선 순서적 깊이를 구할 수 있는 동영상 구간으로부터 1차적으로 순서적 깊이를 얻고, 그 구간 전체에 대하여 순서적 깊이를 평균한 값을 바탕으로 순서적 깊이를 얻을 수 없는 동영상 구간에 대해 임의로 순서적 깊이를 추정한다. 이러한 과정을 통하여 전체 동영상에 대해 2차원의 객체 분할 및 깊이 추정 과정을 마치면 초기 단계의 3차원 동영상상이 얻어진다. 이렇게 얻어진 동영상은 프레임별 객체의 깊이 정보의 편차가 커서 매끄럽지 못한 영상을 보여주므로, 다시 한 번 전체 동영상 구간에 대해 객체 간 3차원 깊이 정보를 평균해주는 과정을 필요로 한다. 그림 8은 초기 단계에 하나의 프레임에 대하여 3차원 깊이 정보를 구하는 과정을 보여준다. 동일한 방법으로 전체 영상에 대하여 깊이 추정을 진행하고, 평균적인 각 객체의 깊이 정보를 추정하여 2차원 동영상 전체에 깊이 정보를 부여하게 되면 3차원 동영상으로의 변환이 완료된다.

IV. 실험

제안한 방법을 검증하기 위해 실제 동영상에 대하여 객체 분할 및 깊이 추정을 실시하였다. 각 실험에서 영역분할은 밝기, 상대적인 위치, optical flow를 가중치로 사용한 normalized cut을 이용하였고, 객체가 하나뿐인 경우는 상대적 깊이의 의미가 없으므로 무시하도록 하였다. 실험은 Intel Core 2 Quad

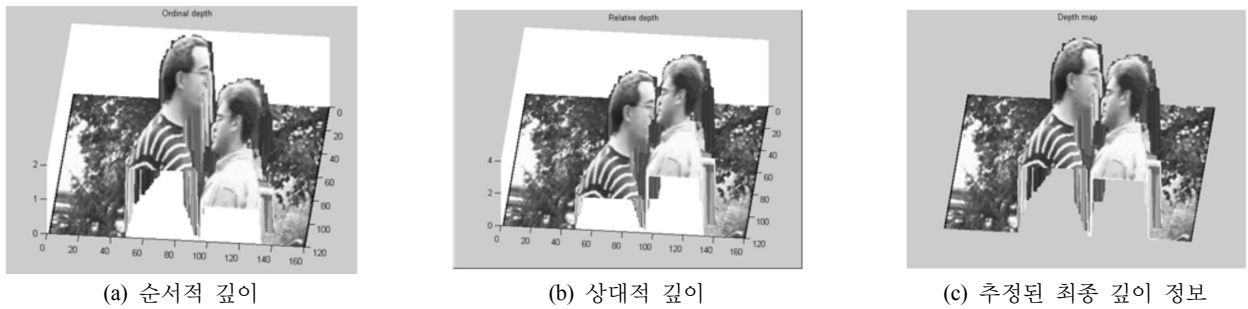


그림 8. 최종 깊이 추정 과정.



그림 9. 2차원 동영상의 3차원 변환 결과.

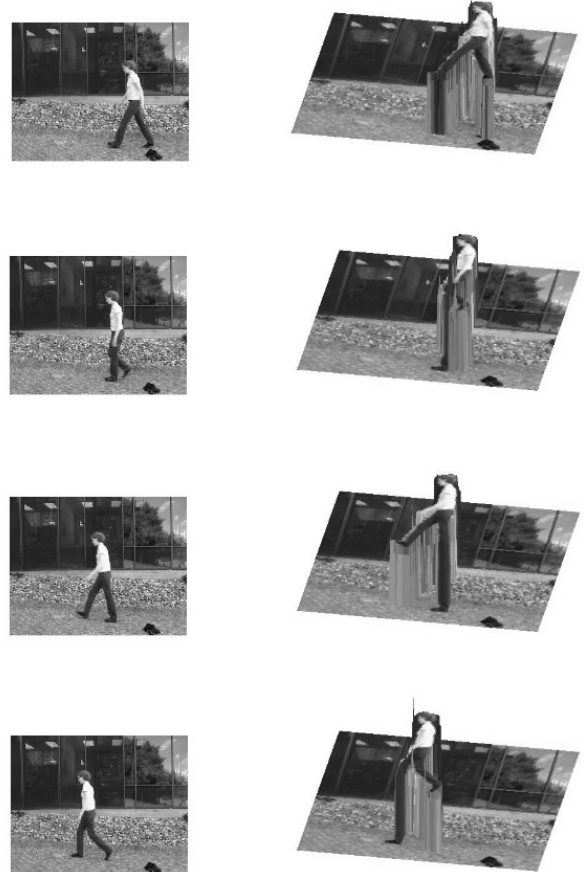


그림 10. 단일 객체 2차원 동영상의 3차원 변환 결과.

Q6600 CPU와 4GB RAM으로 구성된 시스템에서 수행하였으며, OS는 Window XP, 개발 툴은 Matlab 을 사용하였다.

그림 9는 객체 둘, 배경 하나로 이루어진 2차원 동영상에서 영상을 분할하고 깊이를 추출한 결과를 보여주고 있다. 이 동영상은 객체가 둘이므로 객체 사이의 순서적 깊이를 구할 수 있으며, 상대적 깊이도 고려해 주어야 한다. Normalized cut을 이용한 분할 과정에서 몇몇 영상에 잡음이 있지만, 전체적인 객체의 깊이 정보가 추정이 가능함을 알 수 있다.

그림 10은 객체 하나, 배경 하나로 이루어진 동영상을 분할하고 깊이를 추정한 결과를 보여준다. 이 경우는 객체와 배경 사이의 순서적 깊이만 의미 있는 정보가 된다. 그림 9, 그림 10을 통해 2차원 영상으로부터 3차원 영상의 깊이 정

보를 추정해 낼 수 있음을 알 수 있다.

V. 결 론

본 논문에서는 normalized cut과 optical flow를 이용하여 2차원 동영상을 3차원 동영상으로 변환하는 방법을 제안하였다. 먼저 2차원 동영상으로부터 optical flow를 구하여 가중치로 추가하였고, watershed 알고리즘을 통해 개선한 normalized cut을 이용하여 객체를 분할하였다. 또한 분할된 객체의 순서적 깊이는 optical flow를 이용한 가려진 영역을 이용하여 추정하였고, 상대적 깊이는 optical flow의 절대값을 이용하여 추정하였다. 최종적인 3차원 동영상은 순서적 깊이에 상

대적 깊이를 곱하고 전체 평균 optical flow를 나누어 보정하여 표현하였다. 본 연구를 통해 생성된 3차원 동영상은 기존의 2차원 동영상에 깊이 정보를 가진 형태로, 특정 3차원 디스플레이 방식에 의존하지 않고 각 디스플레이 방식에 맞게 변환하여 사용할 수 있다. 또한 제안한 방법은 특정 2차원 동영상 포맷에 국한되지 않으므로, 현재 존재하는 모든 2차원 동영상 콘텐츠에 활용할 수 있어, 2차원 디스플레이 시장과 3차원 디스플레이 시장을 이어주는 중간기술로 활용이 가능할 것으로 예상된다.

감사의 글

본 연구는 BK21 프로그램(Information Technology)의 지원을 받았습니다.

참고문헌

- [1] T. Okoshi, "Three-dimensional displays," *Proc. IEEE*, vol. 68, no. 5, pp. 548-564, 1980.
- [2] J.-H. Park, H.-R. Kim, Y. Kim, J. Kim, J. Hong, S.-D. Lee, and B. Lee, "Depth-enhanced three-dimensional-two-dimensional convertible display based on modified integral imaging," *Opt. Lett.*, vol. 29, no. 23, pp. 2734-2736, 2004.
- [3] H. Choi, J. Kim, S.-W. Cho, Y. Kim, J. B. Park, and B. Lee, "Three-dimensional-two-dimensional mixed display system using integral imaging with an active pinhole array on a liquid crystal panel," *Appl. Opt.*, vol. 47, no. 13, pp. 2207-2214, 2008.
- [4] T. Okino, G. Murata, K. Taima, T. Inuma, and K. Oketani, "New television with 2D/3D image conversion technologies," *Proc. SPIE*, vol. 2653, pp. 96-103, 1996.
- [5] B. J. Garcia, "Approaches to stereoscopic video based on spatiotemporal interpolation," *Proc. SPIE*, vol. 2653, pp. 85-95, 1996.
- [6] Y. Matsumoto, H. Terasaki, K. Sugumoto, and T. Arakawa, "Conversion system of monocular image sequence to stereo using motion parallax," *Proc. SPIE*, vol. 3012, pp. 108-115, 1997.
- [7] M. B. Kim, M. S. Song, D. K. Kim, and K. C. Choi, "Stereoscopic conversion of monoscopic video by the transformation of vertical-to-horizontal disparity," *Proc. SPIE*, vol. 3295, pp. 65-75, 1998.
- [8] J. Ross, "Stereopsis by binocular delay," *Nature*, vol. 248, pp. 363-364, 1974.
- [9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.
- [10] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185-204, 1981.
- [11] 장세일, 황선규, 김희율, "Markov Random Fields를 이용한 움직이는 객체 추출 및 추적," 대한 전자공학회 하계 학술대회, vol. 26, no. 1, pp. 2100-2103, 2003.
- [12] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583-598, 1991.
- [13] A. S. Ogale, C. Fermüller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 988-992, 2005.

Three-Dimensional Conversion of Two-Dimensional Movie Using Optical Flow and Normalized Cut

Jae-Hyun Jung, Gilbae Park, Joohwan Kim, Jinmo Kang, and ByoungHo Lee[†]

School of Electrical Engineering, Seoul National University, Gwanak-Gu Gwanakro 599, Seoul 151-744, Korea

[†]*E-mail: byoungHo@snu.ac.kr*

(Received December 29, 2008; Revised manuscript February 13, 2009; Accepted February 17, 2009)

We propose a method to convert a two-dimensional movie to a three-dimensional movie using normalized cut and optical flow. In this paper, we segment an image of a two-dimensional movie to objects first, and then estimate the depth of each object. Normalized cut is one of the image segmentation algorithms. For improving speed and accuracy of normalized cut, we used a watershed algorithm and a weight function using optical flow. We estimate the depth of objects which are segmented by improved normalized cut using optical flow. Ordinal depth is estimated by the change of the segmented object label in an occluded region which is the difference of absolute values of optical flow. For compensating ordinal depth, we generate the relational depth which is the absolute value of optical flow as motion parallax. A final depth map is determined by multiplying ordinal depth by relational depth, then dividing by average optical flow. In this research, we propose the two-dimensional/three-dimensional movie conversion method which is applicable to all three-dimensional display devices and all two-dimensional movie formats. We present experimental results using sample two-dimensional movies.

OCIS codes: (100.0100) Image processing; (100.6890) Three-dimensional image processing; (100.5010) Pattern recognition and feature extraction