

Optical flow의 레벨 간소화 및 노이즈 제거와 에지 정보를 이용한 2D/3D 변환 기법

한현호¹, 이강성², 이상훈^{2*}
¹광운대학교 대학원, ²광운대학교 교양학부

2D/3D image Conversion Method using Simplification of Level and Reduction of Noise for Optical Flow and Information of Edge

Hyeon-Ho Han¹, Gang-Seong Lee² and Sang-Hun Lee^{2*}

¹Kwangwoon University Graduate School of Information Contents

²Kwangwoon University General Education

요 약 본 논문은 2D/3D 변환에서 깊이정보 생성을 위해 연산량을 감소시키는 레벨 간소화 기법을 적용하고 객체의 고유벡터를 이용하여 노이즈를 제거한 Optical flow를 이용하는 방법을 제안한다. Optical flow는 깊이정보를 생성하기 위한 방법 중 하나로 두 프레임간의 픽셀의 변화 벡터 값을 나타내어 움직임 정보를 나타내며 픽셀 단위로 처리하므로 정확도가 높다. 그러나 픽셀 단위 연산으로 긴 연산 시간이 소요되며 모든 픽셀을 연산하는 특성상 노이즈가 생길 수 있는 문제점이 있다. 본 논문에서는 이를 해결하기 위해 레벨 간소화 과정을 거쳐 연산 시간을 단축하였고 Optical flow를 영상에서 고유벡터를 갖는 영역에만 적용하여 노이즈를 제거한 뒤 배경 영역에 대한 깊이 정보를 에지 영상을 이용하여 생성하는 방법을 제안하였다. 제안한 방법으로 깊이정보를 생성한 뒤 DIBR(Depth Image Based Rendering)으로 2차원 영상을 3차원 입체 영상으로 변환하였고 SSIM(Structural SIMilarity index)으로 최종 생성된 영상의 오차율을 분석하였다.

Abstract In this paper, we propose an improved optical flow algorithm which reduces computational complexity as well as noise level. This algorithm reduces computational time by applying level simplification technique and removes noise by using eigenvectors of objects. Optical flow is one of the accurate algorithms used to generate depth information from two image frames using the vectors which track the motions of pixels. This technique, however, has disadvantage of taking very long computational time because of the pixel-based calculation and can cause some noise problems. The level simplifying technique is applied to reduce the computational time, and the noise is removed by applying optical flow only to the area of having eigenvector, then using the edge image to generate the depth information of background area. Three-dimensional images were created from two-dimensional images using the proposed method which generates the depth information first and then converts into three-dimensional image using the depth information and DIBR(Depth Image Based Rendering) technique. The error rate was obtained using the SSIM(Structural SIMilarity index).

Key Words : 2D/3D Conversion, Optical Flow, Object Segmentation, Depth map, SSIM

1. 서론

3차원 영상이 IMAX 영화관 등을 통해 성공적으로 상

용화됨으로서 3차원 영상의 인기가 증가하였다. 이러한 영향으로 인해 가정에서도 3차원 영상을 시청하려하게 되었고 이는 3DTV 기술의 발전으로 이어졌다. 그러나

본 논문은 2010년도 광운대학교 교내 학술 연구비 지원에 의해 연구되었음.

본 논문은 중소기업청에서 지원하는 2011년도 산학연공동기술개발사업(No.00046375)의 연구수행으로 인한 결과물임을 밝힙니다.

*교신저자 : 이상훈(leesh58@kw.ac.kr)

접수일 11년 12월 12일

수정일 (1차 12년 01월 05일, 2차 12년 01월 13일)

게재확정일 12년 02월 10일

기존의 3차원 콘텐츠 생성 방법인 스테레오 카메라로 촬영하려면 많은 비용이 필요하고 이미지 편집 툴을 이용할 경우 많은 시간이 소요되어 콘텐츠의 생성이 더디게 되고 수가 부족하여 3DTV 사용자를 만족시킬 수 없으며 3D 산업의 병목현상을 야기할 수 있으므로 이를 해결할 3차원 콘텐츠의 생산이 중요하다. 따라서 3차원 콘텐츠의 생성을 위해 2D 영상을 이용하여 3D 영상을 생성하는 방법의 많은 연구가 진행되고 있다[1].

인간이 입체 영상을 인식하는 원리는 양안의 시차로 인해 물체가 좌, 우안에 각각 다른 시점의 영상으로 투영되는데 인간의 뇌에서는 이 좌, 우안에 들어온 영상을 뇌에서 합성하여 입체 영상으로 인식하게 된다.

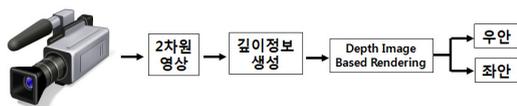
이 원리로 인간이 3차원 입체 영상을 보기위해 좌안과 우안에 해당하는 영상을 생성한 뒤 그 영상들을 각각 좌, 우안에 보여주는 방법이 고안되었다. 이 방법을 기반으로 기존 영상에서 좌, 우안에 해당하는 시점의 영상을 생성하기 위해 다양한 방법들이 제안되었다.

입체 영상을 생성하는데 있어 가장 중요한 요소는 깊이 정보 생성이다. 깊이 정보 생성을 위해 기존의 방법들은 모션을 통한 깊이 추출법, 역 필터링 기법, 전후관계를 연산하여 기하학적인 관점에서의 깊이 추출법, 음영을 이용한 깊이 추출법 등을 이용한다. 그러나 이러한 기존의 방법들은 알고리즘 자체의 약점이 존재하고 정확도가 떨어지는 문제점이 있다[2].

본 논문에서는 기존 Optical flow의 약점으로 알려진 긴 영상 처리 시간과 카메라의 이동, 조명의 변화 등에 취약한 문제점을 해결하기 위해 레벨 간소화 과정과 노이즈 제거 과정을 추가하고 배경 영역에 대한 깊이를 보정함으로써 기존 Optical flow의 약점을 개선하고 보다 정확한 입체 영상을 생성할 수 있도록 하였다.

2. 입체 영상 변환 기법

2D/3D 영상 변환 기술은 기존의 단안 카메라로 촬영된 영상 즉, TV나 CD, DVD에서 얻어지는 영상을 영상 처리 기술을 통해 좌영상과 우영상으로 구성된 입체 영상을 생성하는 것으로 복잡한 원리 및 영상 처리 기술을 필요로 한다.



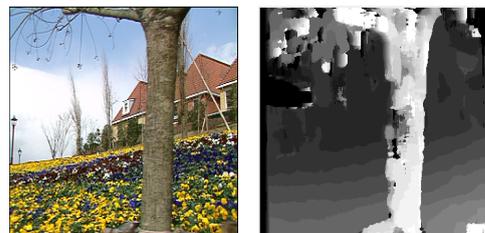
[그림 1] 2D/3D 영상 변환의 원리
[Fig. 1] 2D/3D conversion principle

2D/3D 영상 변환은 현재 영상과 이전 영상을 이용하여 깊이정보를 생성하고 생성된 깊이정보를 이용하여 새로운 시점의 영상을 생성한다[3].

기존의 영상 생성을 위한 깊이정보 생성 방법은 영상에서의 움직임이 추정 가능한 블록매칭기반 알고리즘과 Optical flow를 이용하여 입체 영상 변환을 수행하였다.

2.1 블록 매칭 기반 깊이정보 생성

블록 매칭은 현재 영상과 이전 영상을 블록단위로 분할한 뒤 다른 영상의 블록에 한 블록씩 대응시켜서 블록 내의 픽셀 변화가 가장 적은 블록을 찾아 동일 지점으로 판단하여 영상의 움직임 정도를 연산한 뒤 깊이정보를 생성한다.



(a) (b)
[그림 2] 원본 이미지(a)와 블록매칭의 결과(b)
[Fig. 2] Input image(a) and block matching result(b)

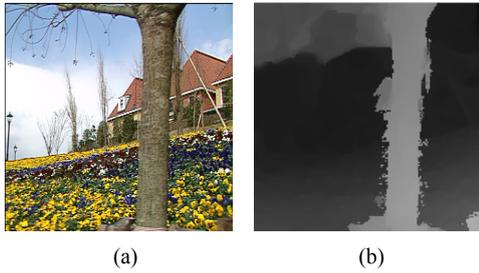
이러한 블록 매칭은 가장 기본적인 방법이며 처리 속도가 빠르다. 그러나 처리 단위의 문제로 인해 정확도가 떨어지며 객체의 이동방향에 민감하기 때문에 한 영상에서의 움직임 방향이 여럿일 경우 정확한 깊이정보를 생성할 수 없는 단점을 가지고 있다[4].

2.2 Optical flow를 이용한 깊이정보 생성

Optical flow는 두 프레임간의 픽셀의 변화 벡터 값을 나타내는 특징을 갖고 있어 연속적인 영상에서의 동일점에 대한 위치차이 값을 나타낸다[5]. 이 위치차이 값을 기반으로 객체의 움직임을 추정하여 깊이정보를 생성한다.

다음 식 (1)은 기존에 제안된 Optical flow의 전체 수식으로, $o(u, v)$ 는 폐색 변수이며 $P(u, v)$ 는 CIE LAB 색공간의 벡터를 나타낸다[6].

$$O_{flow} = \exp \left(- \frac{|u - u'|^2 + |v - v'|^2}{2\sigma_1^2} - \frac{|P(u, v) - P(u', v')|^2}{2\sigma_2^2} \right) \frac{o(u', v')}{o(u, v)} \quad (1)$$



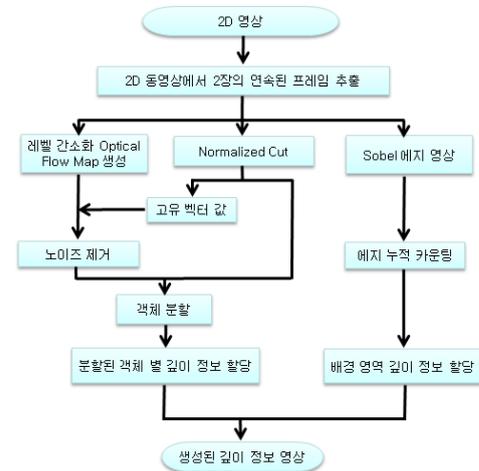
[그림 3] 원본 이미지(a)와 Optical flow의 결과(b)
[Fig. 3] Input image(a) and Optical flow result(b)

그러나 기존의 Optical flow는 연속되는 영상 전체에서 모든 픽셀에 대해 변화 벡터를 각각 구하므로 연산량이 매우 많아 연산 시간이 오래 걸리는 문제점이 있으며 두 프레임간의 픽셀의 변화 벡터를 구하므로 카메라의 이동이나 조명 변화에 취약해 불필요한 정보가 생길 수 있어 이러한 경우에는 정확한 깊이정보 생성이 불가능하다[7].

이러한 단점을 보완하기 위해 기존 Optical flow의 연산 시간을 레벨 간소화 방법을 이용하여 줄이도록 하였고 영상의 고유벡터를 갖는 영역에 한해 Optical flow값을 적용하고 배경 영역에 대한 깊이를 보정함으로써 카메라의 이동이나 조명의 변화에 취약하여 불필요한 정보가 생기는 문제점을 해결하였다.

3. 제안하는 깊이정보 생성 방법

제안하는 방법은 2D 영상에서 2장의 연속된 프레임을 추출한 뒤 추출된 프레임들을 연산시간을 단축하기 위한 레벨 간소화 Optical flow를 적용하여 움직임을 추정하고, Normalized cut 과정을 수행하면서 얻어진 고유 벡터값을 이용해 Optical flow의 취약점인 카메라의 이동과 조명 변화로 인한 노이즈가 생성된 영역을 제거하여 보다 정확한 움직임을 추정한 뒤, 추정된 움직임 정보를 Normalized cut을 이용한 객체 분할의 가중치로 이용하여 분할하고 분할된 객체 별 깊이를 할당하였다. 노이즈 제거 과정으로 인해 객체를 제외한 배경 영역의 깊이 정보가 제거되는 문제를 보완하기 위해 영상의 에지 정보를 기반으로 배경 영역의 깊이 정보를 생성하여 할당함으로써 최종 깊이 정보 영상을 생성하였다.



[그림 4] 제안하는 깊이정보 생성 방법의 흐름도
[Fig. 4] Proposed flow chart of create depth information

3.1 레벨 간소화 과정

Optical flow는 두 프레임간의 픽셀의 변화 벡터를 구하는 알고리즘으로 전체 영상에서 프레임들의 모든 픽셀에 대한 변화 벡터를 구하는 것은 많은 연산 시간이 필요하기 때문에 이를 줄이고자 Optical flow의 연산을 영상의 복잡도를 기준으로 연산 단계를 간소화한 뒤 변화가 큰 특징점을 구하였다.

영상의 복잡도를 연산하기 위해 인근 픽셀간 차의 절대값을 이용한다. 인근 픽셀간 차의 절대값이 큰 경우 해당 픽셀들은 서로 다른 색상정보를 갖는 것으로 볼 수 있고 이러한 경우가 많을수록 많은 정보를 갖는 것으로 간주되어 복잡한 영상이라 판단할 수 있게 된다.

식 (2)는 영상의 복잡도를 연산하기 위해 영상을 7×7 크기의 블록으로 나누고 각 블록 내 픽셀 변화량이 임계값 T_{Thre} 이상일 경우 해당 블록은 복잡도 값 T 를 갖도록 하였다.

$$Comp(M, N) = \sum_{m=0}^6 \sum_{n=0}^6 \begin{cases} T, & \text{if } (|I_{x,y} - I_{x+1,y}| + |I_{x,y} - I_{x,y+1}|) \geq T_{Thre} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

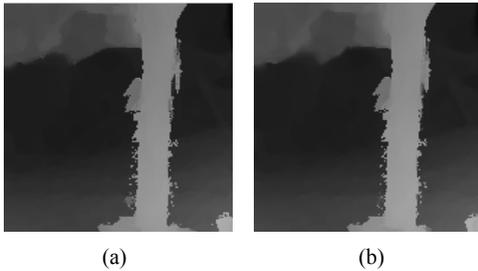
여기서 M, N 은 7×7 크기를 갖는 블록의 x축 위치와 y축 위치이며 m, n 은 7×7 크기의 블록 내 x좌표와 y좌표를 뜻하고 $I_{x,y}$ 는 블록 내 (x,y)좌표에서의 픽셀 값을 나타낸다.

연산된 블록 별 복잡도 값을 누적시켜 영상의 복잡도 I_{comp} 를 구한다. 구해진 I_{comp} 를 영상의 폭 U 와 높이 V , 간

소화 계수 S 의 제공으로 나누어 해당 영상에서의 Optical flow 연산 레벨을 결정하였다.

$$I_{comp} = \sum_{M=0}^{U/7-1} \sum_{N=0}^{V/7-1} Comp(M,N) \quad (3)$$

$$L_{Iter} = I_{comp} / UVS^2 \quad (4)$$



[그림 5] Optical flow(a,c)와 레벨 간소화 결과(b,d)
[Fig. 5] Optical flow(a,c) and level simplified result(b,d)

[표 1] 연산시간 비교 표
[Table 1] Comparison computational time table

영상	연산 시간	
	기존 Optical flow	Proposed
Tree (352×240)	0.34	0.24
Golf (384×288)	0.42	0.29

간소화 결과 기존 Optical flow 결과와의 오차가 크지 않고 연산시간이 평균 30%정도 감소하였다.

3.2 Optical flow의 노이즈 제거 방법

기존 Optical flow에서 카메라의 이동이나 조명의 변화에 취약한 점을 보완하기 위해 노이즈를 제거하는 방법을 제안하였다.

기존 노이즈 제거 방법에서는 노이즈를 제거하기 위해 순차적 영상 별 Optical flow 값이 일정 오차 범위 내에 있는 영역에 한해 적용하였으나 객체 이외의 영역이 인식되어 적용되었고 객체 내 깊이 변화가 큰 문제로 인해 객체의 깊이 값이 매끄럽지 않은 문제가 있었다.

이를 보완하기 위해 객체 분할 과정에서 이용되는 Normalized Cut의 과정에서 생성되는 고유 벡터값을 이용하였다.

Normalized Cut은 전체 영상의 픽셀을 노드 u, v 로 정의하고 픽셀 간의 유사도 $w(u,v)$ 를 예지로 가지는 그래프 $H = \{(i, j), w(u,v)\}$ 로 표현한다. 그래프 분할은 노

드 u, v 간의 가중치 값 $w(u, v)$ 의 합인 $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ 을 통해 이루어진다[8].

Normalized Cut은 식 (5)와 같이 특정 집합 A 에 속하는 점에서 그래프에 존재하는 모든 점의 집합 C 까지의 가중치 합 $assoc(A, C)$ 를 구한다.

$$assoc(A, C) = \sum_{u \in A, t \in C} w(u, t) \quad (5)$$

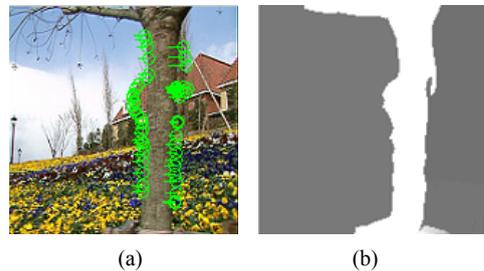
특정 집합 B 에 적용될 $assoc(B, C)$ 도 동일하게 정의하면 Normalized Cut의 전체 수식은 식 (6)과 같다.

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, C)} + \frac{cut(A, B)}{assoc(B, C)} \quad (6)$$

이러한 Normalized Cut의 고유벡터 값은 Normalized Cut의 최적 분할 과정에서 정의된 $d(u, v) = \sum_v w(u, v)$ 로 구성된 대각행렬 D 와 픽셀간의 유사도 값으로 구성된 대칭배열 W 를 이용하여 구할 수 있다.

$$\lambda Dc = (D - W)c \quad (7)$$

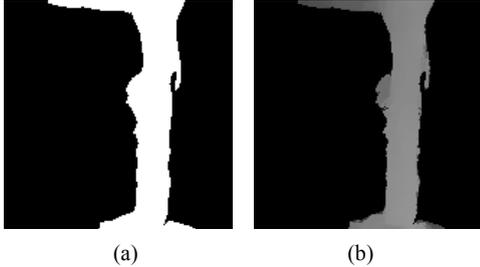
다음 그림 6은 추출된 두 프레임 간 객체의 고유 벡터 방향을 표시한 결과(a)와 그 값을 해당 좌표에 할당할 결과 값(b)이다.



[그림 6] 고유 벡터의 방향(a)와 결과 값(b)
[Fig. 6] Eigenvector(a) and eigenvalue(b)

생성된 고유 벡터 값 그림 6의 (b)는 영상에서 움직임을 가지고 있는 객체를 추출해 낼 수 있도록 한다. 그러나 단순히 고유벡터 값을 Optical flow에 적용할 경우 객체에 적용될 Optical flow값이 일정하지 않기 때문에 객체의 깊이 정보가 정확하게 반영될 수 없는 문제점이 있다. 이를 보정하기 위해 뚜렷한 고유 벡터값을 갖는 영역을 선별하기 위해 고유 벡터값의 이진화 과정을 거치고

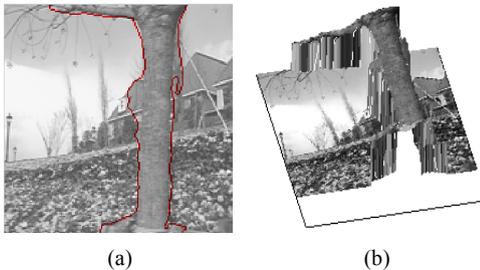
해당 영역에 Optical flow 값을 적용함으로써 카메라의 이동이나 조명의 변화로 인해 생길 수 있는 노이즈가 제거된 깊이 정보가 생성된다.



[그림 7] 고유 벡터값의 이진화 결과(a)와 Optical flow 적용 결과(b)
 [Fig. 7] Binarization of eigenvector(a) and applying optical flow(b)

노이즈가 제거된 Optical flow를 Normalized Cut의 가중치로 이용하여 객체를 분할(8)하고 분할 결과에 Optical flow의 값을 대입함으로써 객체의 깊이정보를 할당하게 된다.

$$OFNcut(A, B) = \frac{\sum_{u \in A, v \in B} w(u, v)}{\sum_{u \in A, t \in C} w(u, t)} + \frac{\sum_{u \in A, v \in B} w(u, v)}{\sum_{v \in B, t \in C} w(v, t)} \quad (8)$$

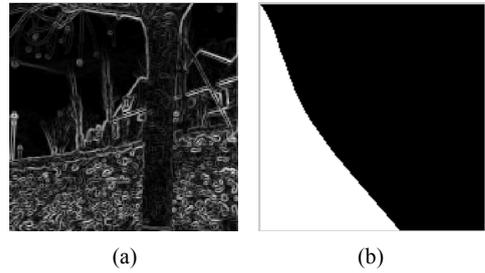


[그림 8] 최종 객체 분할 결과(a)와 깊이정보 할당 결과(b)
 [Fig. 8] Object segmentation(a) and depth information applying result(b)

3.3 배경 영역의 깊이정보 생성

Optical flow의 노이즈 제거로 객체를 제외한 영역의 깊이정보를 제거하였기 때문에 이를 보완하기 위한 배경 영역의 깊이정보를 영상의 에지 수를 기반으로 깊이정보를 생성하여 배경 영역에 대한 깊이정보를 할당할 수 있도록 하였다.

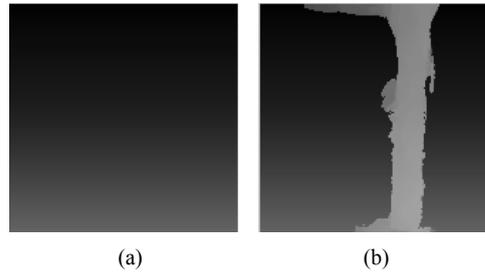
먼저 영상의 에지를 추출하고 영상의 첫 행부터 행 별 에지의 수를 누적시킨다.



[그림 9] 에지 영상(a)와 에지 누적값(b)
 [Fig. 9] edge image(a) and cumulative value of edge(b)

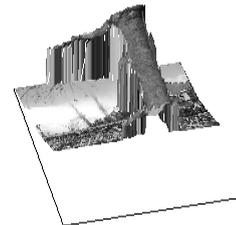
누적된 에지 값을 이용하여 영상의 하단부터 에지 누적값의 변화 폭을 기준으로 배경 영역의 깊이 정보를 생성한다.

생성된 배경 영역의 깊이 정보와 객체의 깊이 정보를 합성하여 최종 깊이 정보를 생성한다.



[그림 10] 배경 깊이 정보(a)와 최종 깊이 정보(b)
 [Fig. 10] background depth(a) and final depth map(b)

그림 11은 생성된 최종 깊이 정보를 영상에 할당하여 나타낸 결과이다.



[그림 11] 최종 깊이 정보 할당 결과
 [Fig. 11] final depth applying result

4. 실험 및 고찰

2D/3D 변환 방법의 최종 입체 영상을 생성하는 방법은 현재이미지와 이전이미지의 합성 방법과 현재 이미지와 깊이 정보를 이용하여 새 시점의 영상을 만든 뒤 현재

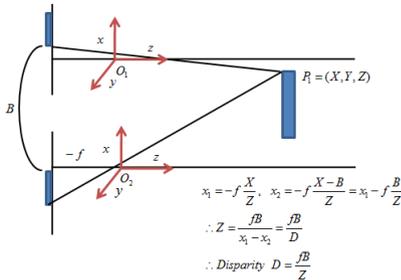
이미지와 합성하는 방법인 DIBR(Depth Image Based Rendering) 방법이 있다. 본 논문에서는 DIBR을 이용하여 최종 입체 영상을 생성하도록 하였다.

4.1 Depth Image Based Rendering

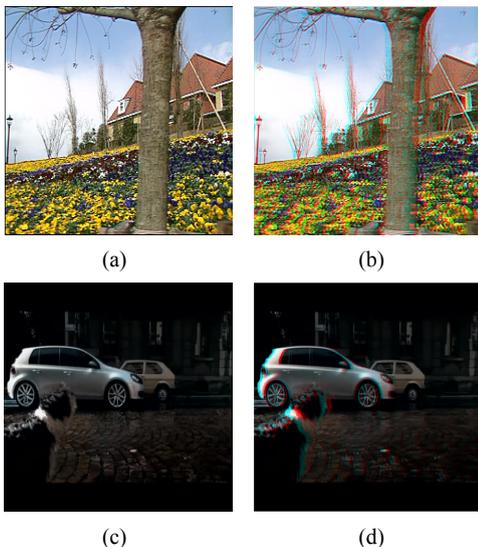
DIBR은 현재시점에 해당하는 컬러 영상과 영상의 깊이 정보를 이용하여 새로운 시점에 해당하는 입체 영상을 재생하는 방식이다[9].

DIBR은 단안 컬러 영상을 깊이정보를 기반으로 워핑시켜 영상의 입체감을 표현하는데 기존 영상에서의 가려진 영역에 대한 정보가 없어 픽셀값이 채워지지 못한 홀영역이 생기고 생성된 홀 영역을 주변 픽셀정보를 이용하여 채우는 홀 채움 과정을 거쳐 새로운 시점의 영상을 생성하게 된다.

다음 그림 12는 DIBR을 이용하여 차이영상을 생성하는 방법에 관한 그림이다.



[그림 12] DIBR의 깊이를 이용한 차이영상 생성
[Fig. 12] Create disparity image using depth map



[그림 13] 원본 이미지(a,c)와 최종 입체영상 생성 결과(b,d)
[Fig. 13] Input image(a,c) and Final output image(b,d)

4.2 SSIM(Structural SIMilarity index) 비교

생성된 입체영상의 오차율을 분석하기 위해 SSIM 방법을 이용하였다.

SSIM은 원 영상과 비교영상의 식별 수준 차이에 따라 원영상을 1로 기준삼아 오차가 심하면 심할수록 0에 접근하는 형태로 평가하는 방법이다.

[표 2] 생성 영상 오차율 비교 표

[Table 2] Comparison error rate table

영상	SSIM		
	BMA	Optical flow	Proposed
Tree	0.669	0.724	0.887
Golf	0.617	0.782	0.843

기존 알고리즘에 비해 제안 알고리즘의 SSIM값이 높아 보다 오차가 없는 영상임을 알 수 있다.

5. 결론

본 논문에서는 2D 영상을 3D 영상으로 변환하기 위해 레벨 간소화 과정을 거치고 객체의 고유벡터 값을 이용하여 노이즈를 제거한 Optical Flow를 이용하여 객체의 깊이 정보를 생성하고 영상의 에지 정보를 이용해 배경 영역의 깊이 정보를 생성하는 방법을 제안하였다. 제안 방법을 이용하여 Optical flow의 연산시간을 평균 약 30% 정도 줄였으며, 기존방법에 비해 SSIM값이 높아 보다 식별 수준이 향상된 결과를 얻을 수 있었다.

향후 연구에서는 실시간 처리를 위한 3차원 영상 생성 방법과 입력 영상에 따라 자체적인 채보정 과정을 포함한 기법의 연구가 필요하다.

References

- [1] Xun Cao, Zheng Li, and Qionghai Dai, "Semi-Automatic 2D-to-3D Conversion Using Disparity Propagation", IEEE TRANSACTIONS ON BROADCASTING, VOL. 57, NO. 2, pp. 491-499, JUNE 2011.
- [2] Xiaojun Huang, Lianghao Wang, Junjun Huang, Dongxiao Li, Ming Zhang, "A Depth Extraction Method Based On Motion and Geometry for 2D to 3D Conversion", 2009 Third International Symposium on Intelligent Information Technology Application, pp. 294-298 Nov 2009.

- [3] Guo-Shiang Lin Cheng-Ying Yeh Wei-Chih Chen Wen-Nung Lie, "A 2D to 3D conversion scheme based on depth cues analysis for MPEG videos", IEEE International Conference on Multimedia and Expo ICME, pp. 1141-1145, 2010.
- [4] 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.
- [5] Hyeon-Ho Han, Yeong-Pyo Hong, Jin-Su Kim, Sang-Hun Lee, "2D/3D image Conversion Method using Object Segmentation for Decrease Processing and Create Depth Map", Proceedings of the KAIS Fall conference, Vol.11 No. 2 pp. 92-95, 2010
- [6] Deqing Sun, Roth, S., Black, M. J., "Secret of Optical Flow Estimation and Their Principles", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432-2439, 2010.
- [7] Hyeon-Ho Han, Gang-Seong Lee, Sang-Hun Lee, "A Study on 2D/3D image Conversion Method using Create Depth Map", Journal of The Korea Academia-Industrial cooperation Society, Vol. 12, No. 4 pp. 1897-1903, 2011
- [8] Jianbo Shi and Jitendra Malik, "Normalized Cuts and Image Segmentation", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 8, 2000.
- [9] W.-Y. Chen and Y.-L. Chang and S.-F. Lin and L.-F. Ding and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *Proc. ICME*, pp. 1314-1317, 2005.

한 현 호(Hyeon-Ho Han)

[준회원]



- 2009년 2월 : 광운대학교 컴퓨터 공학과 졸업(공학사)
- 2011년 2월 : 광운대학교 정보콘텐츠 대학원 유비쿼터스 컴퓨팅학과 졸업(공학석사)
- 2011년 3월 ~ 현재 : 광운대학교 대학원 재학(박사과정)

<관심분야>

무선 네트워크, 3D영상처리

이 강 성(Gang-Seong Lee)

[정회원]



- 1986년 2월 : 광운대학교 컴퓨터 공학과 졸업(공학사)
- 1988년 2월 : 광운대학교 대학원 컴퓨터공학과 졸업(공학석사)
- 1993년 2월 : 광운대학교 대학원 컴퓨터공학과 졸업(공학박사)
- 2009년 2월 : 서울대학교 음악대학원 작곡과 졸업(석사)

- CMU(Carnegie Mellon University), U.S.A. 교환교수
- 1990년 9월 ~ 현재 : 광운대학교 정교수

<관심분야>

음성공학, 음악공학, 미디어아트, 작곡

이 상 훈(Sang-Hun Lee)

[종신회원]



- 1983년 2월 : 광운대학교 응용전자 공학과 졸업(공학사)
- 1987년 8월 : 광운대학교 대학원 전자공학과 졸업(공학석사)
- 1992년 2월 : 광운대학교 대학원 전자공학과 졸업(공학박사)
- 1990년 9월 ~ 현재 : 광운대학교 정교수

- 2010년 3월 ~ 2012년 1월 : 광운대학교 교양학부장
- 2012년 2월 ~ 현재 : 광운대학교 정보통신처장

<관심분야>

무선인터넷, 무선네트워크, USN, 영상인식, 3D영상처리