

## 객체 추출과 Non-Local 필터를 이용한 2D 영상의 3D 변환

\*강근호 \*\*이왕로 \*\*\*유지상

광운대학교 전자공학과

\*keunhodivine@kw.ac.kr \*\*wangcarrot@kw.ac.kr \*\*\*jsyoo@kw.ac.kr

### Conversion of 2D to 3D image using Object extraction and Non-local filter

\*Kang, Keun-ho \*\*Lee, Wang-ro \*\*\*Yoo, Jisang

Dept. Electronic Engineering Kwangwoon University

#### 요약

본 논문에서는 움직임 추정(Motion Estimation, ME), 색상 라벨링(Labeling) 그리고 Non-Local means 필터 등을 이용하여 2D 영상을 3D 입체 영상으로 변환하는 기법을 제안한다. 제안하는 기법에서는 프레임 간의 움직임 추정 방법을 사용하여 물체의 움직임 벡터를 추출하며 색상 라벨링 작업을 통해 세밀한 객체를 추출한다. 객체를 추출한 후 영상을 이동시켜서 우 영상을 생성한다. 우 영상을 생성하는 과정에서 채워지지 않은 화소들이 발생하는데 전체 화소의 상관도를 고려하는 Non-local means 필터를 사용하여 이 부분을 처리한다. 생성된 우 영상과 원본 영상인 좌 영상으로 비월주사(interlace)하여 최종 3D 입체 영상을 생성한다.

#### I. 서론

3D 입체 영상이 소개된 것은 오래되었지만 일종의 체험 대상일 뿐 실생활과 관련되어 활성화 된 것은 최근일이다. 할리우드를 중심으로 3D 영화들이 개봉되어 큰 인기를 끌면서 3D 미디어에 관한 관심과 이에 대한 수요도 증가하고 있다. 이러한 추세에 맞추어 3DTV, 3D 카메라 등 개인용 기기도 상용화 제품이 출시되고 있으며 전 세계적으로 3D 입체 방송에도 큰 관심을 갖게 되었다.

스테레오 카메라를 사용하여 3D 입체 영상을 제작하는 것은 비용과 시간이 많이 소요되며 이직도 2D 영상에 비해 촬영하기가 어려운 문제점이 있다. 또한 2D로 제작된 영상물이 많이 존재하는데 이것을 활용하는 것이 스테레오 카메라로 제작하는 것보다 비교적 용이하다. 2D 영상을 3D 입체 영상으로 변환하는 연구는 3D에 관심커지면서 더욱 활성화되고 있으면 다양한 기법들이 제안되었다. 향후 3D 입체 영상물을 제작할 수 있는 환경이 더욱 용이해지고, 제작상의 문제들이 해결될 때까지 이러한 변환 기법을 이용한 3D 영상물 생성 노력은 계속 될 것으로 예상된다.

본 논문에서도 2D 영상을 3D 입체 영상으로 변환하는 기법을 제안하고 있다. 사람의 시선이 집중되는 부분인 객체를 중심으로 깊이감을 부여하게 된다. 객체를 찾는 과정에서 앞 뒤 9개 프레임의 움직임 정보로부터 효과적인 움직임 정보를 구하여 객체 검출에 활용한다. 또한 non-local 필터를 이용해 우 영상 생성 시 발생하는 가려짐 영역을 효과적으로 처리한다. 최종적으로 객체에 깊이감을 부여하여 생성된 우 영상과 원본 영상인 좌 영상을 비월주사하여 3D 영상을 재현한다. 본 연구에서 제안하는 기법이 기존의 3D 입체 영상 변환 기법들의 문제점들을 보완하여 높은 성능을 가진다는 것을 재현된 영상으로부터 알 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 전체적인 구조도와 세 부적인 동작을 설명하고, 3장에서는 제안한 기법에 대한 실험을 수행하여 그 결과를 보였다.

#### II. 제안하는 변환 기법

그림 1은 변환 시스템의 전체적인 구조도이다. 먼저 입력 영상의 앞 뒤 프레임 차를 이용하는 움직임 추정을 통하여 움직임 벡터를 계산하고 영상에 대해서 색상 라벨링 작업을 수행하여 영상을 나눈다 [1,2]. 움직임 추정 결과와 색상 라벨링 결과를 이용하여 객체를 추출하고 배경과 객체에 다른 깊이 값을 부여한다. 부여된 깊이 값에 따라 영상을 이동하여 우 영상을 생성한다. 우 영상 생성 과정에서 발생하는 가려짐 영역은 Non-local means 필터를 이용하여 처리하게 되며 최종적으로 원영상인 좌 영상과 생성된 우 영상의 행을 교차시켜 3D 입체 영상을 재현하게 된다[3].

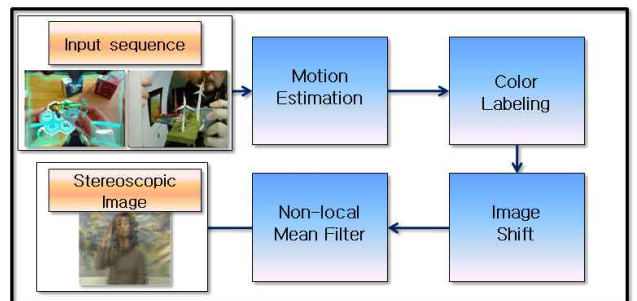


그림 1. 시스템 구조

## 2.1. 영상의 움직임 정보 추출

블록 기반의 움직임 추정 방법을 이용하고 유사성을 평가하는 기준은 식 (1)의 SAD(Sum of Absolute Difference)를 이용해서 최소의 SAD를 갖는 움직임 벡터를 구하고, 움직임 벡터를 영상으로 표현하기 위해서 벡터의 크기를 화소의 값으로 표시한다[1].

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |f_N(i, j) - f_{N-1}(i+x, j+y)| \quad (1)$$

본 논문에서는 2x2 크기의 블록으로 움직임을 추정을 수행한다. 2x2 크기의 블록으로 움직임을 추정을 하면 그림 2와 같이 움직임 정보에 잡음이 많아지지만 더 세밀한 움직임 정보를 추출할 수 있다는 장점이 있다. 또한 본 논문에서는 움직임 추정을 시 앞 뒤 9개 프레임까지 참조하여 움직임 정보를 획득한다[1].

앞 뒤 9개 프레임에 대하여 움직임을 추정을 하면 잡음의 분포는 분산적이기 때문에 비교적 객체부분을 잘 추출할 수 있다. 즉 그림 3에서 보면 움직임 예측에 사용된 프레임의 수가 많아질수록 객체 부분이 강조되는 것을 알 수 있다. 또한 그림 3의 세 번째와 네 번째 사진처럼 객체입도 불구하고 움직임이 적은 부분은 더 많은 프레임을 참조하여 움직임을 추정함으로써 객체로 판별하는데 도움이 된다. 또한 너무 과도한 움직임을 갖는 객체를 판별하기 위해서 현재 프레임에 가까운 참조 프레임의 움직임 추정치에 더 큰 가중치를 두고 움직임 정보를 구하게 된다. 그림 3의 마지막 영상은 가중치를 적용하여 구한 움직임 벡터를 모두 더하여 평균을 구한 결과 영상이다.

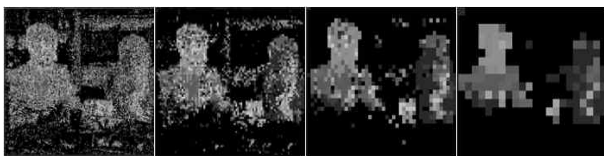


그림 2. 블록 크기에 따른 움직임 추정치 (왼쪽부터 2x2, 4x4, 8x8, 16x16)

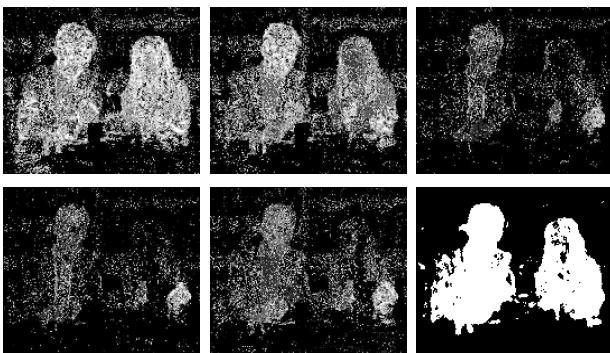


그림 3. 앞 뒤 프레임 참조에 따른 확률기반 움직임 추정 (왼쪽 위부터 현재와 과거 9프레임, 현재와 과거 4프레임, 현재와 과거 1프레임, 현재와 미래 2프레임, 현재와 미래 4프레임, 최종 결과)

앞 뒤 프레임 참조로 평균을 구한 최종 움직임 추정 결과에서 남

아 있는 잡음과 객체 안에 채워지지 않은 홀(hole) 들은 형태학적(morphological) 연산인 팽창(dilation)과 침식(erosion) 연산을 이용한다[4]. 그림 4는 원 영상과 원 영상을 침식 연산 1회 수행하고 내부팽창 연산 3회를 거친 최종 움직임 추정 결과 영상이다.

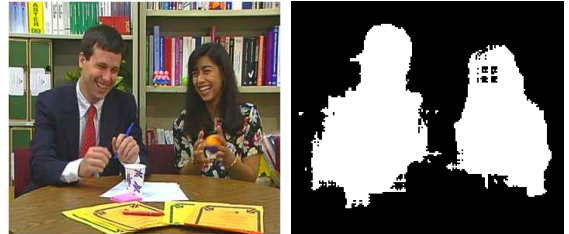


그림 4. 원본 영상과 침식 팽창 결과

## 2.2. 색상 라벨링 작업

본 논문에서는 색상 라벨링의 방법으로는 널리 알려진 Rosenfeld의 방법을 사용한다[2]. 또한 객체 추출을 위한 효율적인 색상 라벨링 전처리 과정으로 식 (2)를 이용하여 RGB 색상계를 HSI 색상계로 변환 후 Mean-Shift 필터를 적용한 후 색상 라벨링 작업을 수행한다[2,3]. 휘도값의 범위는 0부터 255이고, 그림 5와 같이 HSI 색상의 Hue값의 범위가 0부터 360° 이기 때문에 색상 라벨링 작업을 통해 휘도가 같지만 색상이 다른 부분들을 다른 라벨로 표시할 수 있다[2,4].

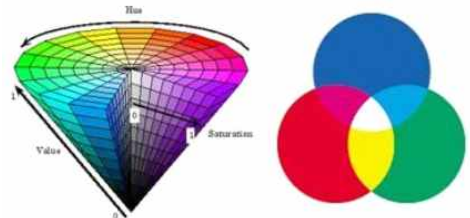


그림 5. HSI, RGB 색상 모델

$$I = \frac{1}{3}(R + G + B)$$

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]$$

$$H = \cos^{-1} \left[ \frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right] \quad (2)$$

if  $B > G$ ,  $H = 360^\circ - H$

또한 Mean-Shift 필터는 사용자가 지정한 범위의 중심점을 기준으로 탐색을 한다. 필터의 적용 범위는 가능한 한 비슷한 색상을 가지는 영역 안에서만 이루어지고 다른 색상을 가지는 영역의 경계를 넘지 않는다. 따라서 탐색 범위 안에 같은 색상의 화소를 많이 포함하는 방향으로 탐색 범위를 이동하여 필터를 적용하게 된다. 이와 같은 방법으로 탐색 범위가 정해지면 그 범위에서의 화소들의 평균값을 구하여 초기 탐색창의 중심점의 값이 되도록 한다[3]. 이렇게 함으로써 단순히

blurring되는 문제를 Mean-Shift 필터로 경계를 보존할 수 있었다. 처리에 적합한 조건들로는 탐색 영역을 원의 형태를 띤 범위를 설정한다 [3]. 탐색창의 반지름을 크게 할수록 참조하는 화소 개수가 많아져 연산 속도가 느려지기 때문에 본 논문에서는 적절한 값을 선택한다. 식 (3)은 색상이 다른 영역의 경계 부근에서 탐색 영역 안에 포함되는 화소를 결정하는 기준이다.

$$dR^2 + dG^2 + dB^2 < threshold \quad (3)$$

여기서  $dR, dG, dB$ 는 탐색창의 중심화소와 현재 화소와의 R, G, B 각 성분의 차이이다. 문턱치 값을 작게 할수록 영상의 경계는 유지되어 Mean-Shift 필터의 결과로 생기는 blurring 효과를 줄일 수 있다. 본 논문에서는  $threshold$  의 적절한 값을 선택하여 색상 라벨링의 전체 과정으로 사용한다. 그림 6에 Mean-Shift 필터링 결과와 색상 라벨링 결과를 보여주고 있다.



그림 6. Mean-Shift 처리 결과와 색상 라벨링 결과

### 2.3. 객체 추출 기법

앞 절에서 구한 움직임 추정 결과와 색상 라벨링 결과를 비교 분석하여 객체를 추출한다. 색상 라벨링 결과 각 라벨의 영역안의 화소들이 의 일정 비율 이상 움직임 정보를 갖지 않으면 비 관심 객체로 판단하여 제거한다. 최종적으로 남은 영역만 모으면 움직임 정보가 있는 관심 객체를 추출할 수 있다. 그림7에 추출된 객체를 보여주고 있다.



그림 7. 움직임 추정 결과 영상, 라벨링 결과 영상

추출된 객체에 깊이 값을 부여하고 이 깊이 값에 따라 객체를 오른쪽으로 이동시켜 우 영상을 생성한다. 또한 시각적으로 객체만 튀어나와 보이는 부자연스러운 부분을 없애기 위해서 배경도 같은 방법으로 이동시킨다. 그림 8는 생성된 우 영상이다.



그림 8. 우 영상 결과

### 2.4 Non-local means 필터링

우 영상 생성 시 값이 채워지지 않은 화소들이 생긴다. 이것은 해당 화소가 값을 가지지 않기 때문에 검은색으로 표현되어 잡음처럼 보이기도 한다. 따라서 우 영상을 생성하고 이러한 채워지지 않은 점을 채우기 위한 후 처리 과정은 꼭 필요하다. Non-local means 필터는 잡음 제거에 많이 사용되는 기법 중 하나이다[5]. 본 논문에서는 채워지지 않은 화소들을 잡음으로 가정하고 이 기법을 적용한다.

$$v(i) = u(i) + n(i) \quad (4)$$

$$v = \{v(i) | i \in I\}$$

여기서  $v(i)$ 는 관측된 영상,  $u(i)$ 는 원 영상,  $n(i)$ 는 잡음성분이다. 이 때  $i$ 는 화소의 위치이다. 이 경우 잡음이 포함된 영상  $v(i)$ 는 가려짐 영역을 처리 하지 않은 잡음 영상에 해당한다. 잡음이 있는 우 영상  $v(x,y)$ 에서 잡음이 없는 원래의 값  $u(i)$ 를 복원한다는 의미이다. Non-local means 필터가 대부분의 다른 잡음 제거 필터들과 다른 점은 잡음의 값을 복원하기 위해 주위(local)의 값들만 참조하지 않고 영상의 전체 화소들을 참조한다는 점이다. 식 (4)는 Non-local means 필터를 보여주고 있다.

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j), \quad (5)$$

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}}, \quad 0 \leq w(i,j) \leq 1,$$

$$\sum_j w(i,j) = 1$$

$$Z(i) = \sum_j e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}}$$

여기서  $NL[v](i)$ 는 화소  $i$  위치에서의 추정된 값이며,  $w(i,j)$ 는  $(i,j)$  위치에서의 화소 값과 유사한 정도에 따라 가중치를 준 것이다.  $h$ 는 필터의 차수를 나타내며,  $\|v(N_i) - v(N_j)\|_{2,\alpha}^2$ 은 유클리디안 거리를,  $N_k$ 은 화소  $k$  위치를 중심으로 이웃 블록을 나타낸다. 그림 9는 Non local means 필터 성능을 보여주기 Lena 영상에 적용한 결과이다.



그림 9. Non-local means를 이용한 잡음 제거 결과

Non-local means 필터는 블록간 화소 값을 비교하여 제일 유사한 블록을 찾는 필터이므로 연산량이 많다. 본 논문에서는 우 영상 생성 시 생기는 채워지지 않은 화소에서만 필터링을 수행하여 시간을 단축시킬 수 있다[5]. 그림 10은 그림 8의 영상에 적용한 결과 영상이다.



그림 10. Non-Local means 처리 결과

### III. 실험 결과

본 논문에서 제안한 기법을 사용하여 2D 영상의 3D 변환 성능을 검증해 보았다. 그림 11은 Paris(352\*288) YUV420 형식의 영상으로 정지된 카메라 환경의 2D 영상을 사용하여 3D 입체 영상을 만들어낸 결과이다.



그림 11. 원 영상과 3D 입체 결과 영상

원 영상과 우 영상의 비월주사결과 객체 중심적으로 입체감이 느껴지고 가려짐 영역처리 결과도 배경과 자연스럽게 연결이 되어 높은 품질의 결과를 도출하였다.

### IV. 결론

본 논문에서 제안한 기법은 움직임 정보와 색상 라벨링 작업을 이용하여 객체를 추출하고 영상의 객체를 이동시켜서 3D 영상을 만드는

객체 중심의 3D 변환 기법이다. 먼저 앞 뒤 9 프레임까지의 움직임 추정을 통하여 움직임 정보를 추정하고 객체를 추출하였다. 앞 뒤 하나의 프레임으로만 사용했을 때보다 정확한 객체 판별 및 잡음 제거에서 좋은 성능을 보여주었다.

또한 색상 라벨링 기법의 전처리 과정으로 Mean-Shift 필터를 사용한 후 영역을 더하여 객체 추출 결과를 더 정확하게 하였고 Hue 값으로 색상 라벨링 기법을 사용하여 휘도 값보다 더 넓은 영역을 사용해 세밀한 객체 판별에 좋은 성능을 보여주었다. 또한 우 영상 생성 시 발생하는 가려짐 영역은 Non-local means 필터를 적용하여 채움으로써 높은 품질의 우 영상을 생성할 수 있었다.

### < 감사의 글 >

이 논문 또는 저서는 2010년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임. (한국연구재단 - 2010 - 0026245)

### < 참고 문헌 >

- [1] Philip H.S. Torr and Andrew Zisserman: Feature Based Methods for Structure and Motion Estimation, ICCV Workshop on Vision Algorithms, pages 278-294, 1999.
- [2] Rosenfeld, J., Pfaltz, J.L., "Sequential Operations in digital Processing," JACM, 13, 471-494, 1966.
- [3] D.Comaniciu and P.Meer "means Shift Analysis and Applications," Proc.Seventh Int'l Conf. Computer Vision, pp. 1197-1203, Sept. 1999.
- [4] I.Pitas, "Digital Image Processing Algorithms And Applications". WILEY, 2000.
- [5] A. Buades, B. Coll and J.M. Morel. "A non-local algorithm for image denoising". IEEE Int. Conf. on Computer Vision and Pattern Recognition. 2005