

혼합 영상의 3D 변환을 위한 이미지 영역 검출

이준석 위진영 길종인 김만배

강원대학교 IT대학 컴퓨터정보통신공학과

{leejunseok, jywee, jgil, manbae}@kangwon.ac.kr

Detecting Image Regions in Mixed Images for 3D Conversion

Jun-Seok Lee, Jin-Young Wee Jong-In Gil and Manbae Kim

Dept. of Computer & Communications, Kangwon National University

요약

최근 3D산업의 급성장으로 인해 3D콘텐츠 제작에 대한 관심이 확산되고 있다. 지금까지는 3D콘텐츠 제작을 영화나 애니메이션 같은 자연영상을 3D변환하는 것에 초점이 맞추어져 있었다. 자동 3D변환에서는 window 영상처럼 텍스트, 이미지, 로고 등의 혼합되어 있는 영상을 처리하는데 어려움이 있다. 특히 텍스트는 동일한 깊이맵을 얻지 못하면, 깨짐, 흔들림 등의 문제점이 발생한다. 해결방법으로 혼합 영상을 구분하고 영상 영역만을 탐색해서, 3D변환을 하고, 다른 영역은 2D로 처리함으로써 상기 문제점을 극복할 수 있다.

1. 서론

입체영상기술은 시청자에게 장면의 깊이감을 제공함으로써 실제 환경에서 보는 것과 같은 입체감을 느끼게 하는 기술이다. 입체영상 생성 기법 중에 3D변환은 자동으로 2D영상에서 깊이맵(depth map)을 추출하여, 2D+Depth로 좌우 입체영상을 생성하는 기술이다 [1-3]. 이 기술은 영화나 애니메이션과 같은 자연영상의 3D콘텐츠에는 우수한 화질 및 3D품질을 얻을 수 있으나, 그림 1의 webpage와 같이 텍스트, 이미지, 로고 등이 혼합되어 있는 영상물에서는 화질 및 품질의 저하가 발생한다. 단순한 방법은 webpage 등의 윈도우 영상을 3D변환을 하지 않고, 2D영상을 그대로 보여주는 것이다. 이 방법은 3D감을 지속적으로 시청자에게 전달하지 않아, 시청자가 불편함을 느끼게 된다.



그림 1. 윈도우 영상의 예

본 논문에서는 상기 문제점을 극복하기 위해서 입력영상을 window image와 full-screen image로 구분하고, window image를 입력영상으로 하여 이미지가 존재하는 RIR (Rectangular Image Region)을 찾는 방법을 제안한다. 응용에서는 이 RIR만 입체변환을 하고, 나머지 영역은 텍스트 영역으로 가정하여, 입체변환을 수행하지 않고 2D로 보여준다.

2. 제안방법

그림 2는 입력받은 영상에서 window image를 구분하고, RIR을 찾기

위한 전체적인 순서도를 나타내고 있다. 먼저 입력받은 영상을 grayscale image로 변환 후 histogram을 구하여 제일 많이 포함된 화소의 비율에 따라 일차적으로 영상을 구분한다. 구분된 window image의 line을 검출하고, 라인 수 정보를 바탕으로 window image와 full-screen image를 구분한다. 그 후 레이블링 맵을 생성하고, 각각의 레이블 영역을 포함하는 bounding box를 구한다. bounding box의 라인 수, 면적 등의 정보를 이용하여 RIR로 부적합하다고 판단되는 bounding box를 제거한다.

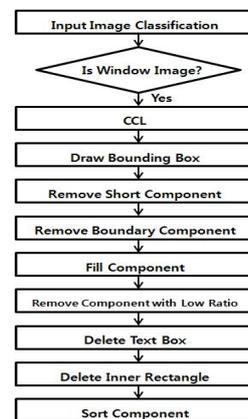


그림 2. Rectangle Search 순서도

3. Full-screen/Window Image 구분 방법

영상의 Histogram을 분석하고, 최대 빈도수를 가지는 레벨의 비율로 일차적으로 영상을 구분한다. 구분된 영상을 다시 line finding으로 라

인을 검출하고, short line을 제거한다. 그 후 임계치 이상의 길이를 가지는 수직, 수평 line의 수를 이용해서 full-screen image와 window image를 구분한다.

A) *Classify With Histogram*: 웹페이지의 특징 중 하나는 색이 고르게 분포하는 full-screen 영상과 달리 배경의 영향으로 한 가지 특정 색상의 분포가 크다는 것이다. 따라서 색의 분포를 알려주는 히스토그램을 이용하여 영상을 분류한다. 먼저 영상을 grayscale로 변환한 후, 히스토그램을 구하여 최대 빈도수(frequency)를 가지는 레벨을 선택한다. 이 레벨에 해당하는 픽셀의 비율 R을 계산하고 임계값과 비교하여 작을 경우 full-screen image로, 아니면 window image로 판단한다.

B) *Line Detection*: window image와 full-screen image의 차이 중 다른 하나는 웹페이지는 full-screen image에 비해 직선이 많이 존재한다는 것이다. 이는 입력 데이터의 line pixel의 비율을 측정하여 판단한다. 0°, ±45°, 90°의 라인을 검출하기 위해 다음과 같은 3×3 콘볼루션 마스크를 사용한다[4].

$$\begin{aligned}
 0^\circ &= \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} & 90^\circ &= \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \\
 +45^\circ &= \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} & -45^\circ &= \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}
 \end{aligned}$$

R을 각각의 마스크의 응답이라고 할 때, 영상의 어떤 주어진 점에서 가장 가능성이 높은 방향의 마스크의 응답을 max(R)로 구할 수 있다. max(R)이 임계치보다 크면 출력 영상의 화소에 255값을 할당한다.

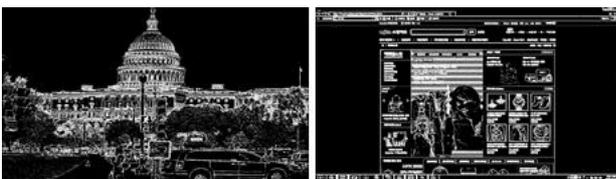


그림 3. Line finding 결과영상.

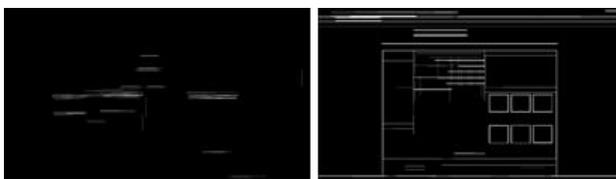


그림 4. Line finding 후 수직/수평 line만을 검출한 영상

Line 추출 방법으로 얻어진 라인 픽셀은 그림 3에서 보여진다. 알고리즘을 통해 구한 line pixel의 비율로 입력 데이터가 window image인지 full-screen image인지 판단할 수 있다.

단순히 line pixel의 비율로 영상을 분류하는 것은 임계값에 민감하기 때문에, 보다 정확한 구분을 하기 위해 수평/수직 라인이 일정 길이 이상으로 이어진 line 만을 검출한다. 검출된 line의 개수로 입력 영상이

window image인지 full-screen image인지 판단할 수 있다 (그림 4).

4. RIR 탐색방법

window image 구분 과정에서 생성된 line 영상을 가지고 다음과 같은 작업을 수행한다.

A) *CCL*: 8-방향으로 탐색하여 연속적으로 붙어있는 픽셀들을 개별적인 영역으로 레이블링한다. 레이블링을 수행하면서 각 영역의 픽셀 수를 구하고 최종적으로 레이블맵을 생성한다.

B) *Bounding Box*: 이렇게 만들어진 라인 정보를 토대로 레이블링 작업을 수행한다. 레이블링된 각각의 분리된 영역의 수평 최소, 최대값 및 수직 최소, 최대값을 구하면 bounding box의 좌상단 좌표, 우하단 좌표, 너비, 높이를 구할 수 있다.

C) *Feasible Rectangle 검사*: bounding box를 그린 후 너비와 높이를 조사하여 둘 중 하나라도 50 Pixel을 넘지 못하면 box에서 제거하였다. 그 후 각각의 레이블링된 영역을 조사하면서 pixel수가 300 이하가 되는 레이블 영역을 비활성화 시킨다. pixel수가 작은 레이블 영역들은 대체로 불필요한 라인들로 이루어져 있다. 또한 사각 형태를 띠고 있더라도 너무 작기 때문에 RIR으로써 부적합하다.

D) *Component Filling*: 지금까지의 과정에서 구한 box들은 image region일 수도 있고 아닐 수도 있다. 예를 들어 웹페이지의 블로그 배경의 경우 다양한 무늬의 디자인을 사용할 수 있다. 이러한 무늬들은 상대적으로 명확한 라인을 검출하게 만들며 지금까지의 과정에서는 이를 image region으로 충분히 오판 할 수 있다. 이렇게 잘못 검출된 box들을 제거해야 하는 과정이 필요하며, 이는 입력 영상이 window image가 아니라 full-screen image로 잘못 들어왔을 경우에도 잘못된 box를 검출하지 않게 해준다.

5. 실험결과

full-screen image, window image 영상의 구별과 RIR 검출확률에 대해 각각 실험을 시행하였다. 영상 구별의 정확도 실험을 위해 포털, 동영상, 영화, 그 외 full-screen image 및 webpage image(window image)를 입력으로 사용하였으며 총 11,283장의 입력영상을 테스트하였다.

표 1. 전체 입력 데이터 판별 결과. DR=detection ratio

Test Image	총 개수	실패 개수	DR
Portal	725	7	99
Other Webpage	7,499	253	96
Full-Screen	3,063	64	98
Total	11,287	324	97

표 1은 전체 입력 데이터의 결과를 나타낸다. Portal은 포털 사이트를 테스트 한 결과이고 Other Webpage는 카테고리 별 분류에 따른 것이 아닌 일반적인 여러 형태의 webpage 입력 데이터를 테스트한 결과이고 Full-Screen은 풀 스크린 영상을 테스트 한 결과이다. 검출율 DR은 portal image가 99%, 다른 웹페이지가 96%, 자연영상 등의 full-screen image가 98%이다. 전체적으로 97%의 검출율을 얻었다.

RIR 검출 정확도를 구하기 위해 실험 데이터로 10장의 포털사이트 (예: Nate, Naver, Daum, Google)의 캡처 영상을 사용되었다. 검출된

rectangle의 RIR여부는 눈으로 확인한 ground truth data를 기준으로 판정하였다. Misdetection은 RIR이지만 검출하지 못한 개수를 의미한다. False alarm은 RIR이 아님에도 불구하고 잘못 검출한 rectangle의 개수를 의미한다. Hit는 적중률로 RIR을 제대로 검출한 개수이다. 표 1에서 보는 것처럼, 총 146개의 ground-truth RIR에서 16개의 misdetection, 2개의 false alarm, 128개의 Hit를 얻었다. 그림 5는 윈도우영상에서 검출된 RIR을 보여준다.

표 2. RIR 검출 정확도

Test image	No. of Images	Misdetection	False alarm	Hit
1	9	1	0	8
2	16	3	0	13
3	67	6	1	60
4	11	3	0	8
5	10	1	0	9
6	2	0	0	2
7	14	1	1	12
8	10	0	0	10
9	2	1	0	1
10	5	0	0	5
합	146	16	2	128



그림 5. 검출된 RIR의 예

6. 결론

본 논문에서는 window image를 구분하고 이를 입력영상으로 하여 rectangular image region(RIR)을 찾는 방법을 제안하였다. 상기 연구는 3D변환에서 텍스트등의 문제점을 극복하고, 시청자에게 편한한 3D를 전달할 수 있는 기술이다. 영상 구분 과정에서 입력 데이터 중 지도, 패턴, 단일화 된 영상은 낮은 검출율을 보였다. 이를 위해 추가적인 알고리즘에 대한 연구가 진행되어야 할 것이다. 또한 어디까지를 window image로 분류하고 어디까지를 full-screen image로 분류할 지에 대한 정확한 기준이 필요하다. RIR 검출 알고리즘에서는 라인정보에 의존성을 좀 더 낮추고 더욱 다양한 정보를 활용하여 형과 배경을 구분할 수 있도록 연구가 진행되어야 할 것이다.

감사의 글

본 논문은 중소기업청의 기술혁신개발사업 (S2056930) 및 지식경제부 및 지식경제부 및 산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 지원을 받았음 [KI002058, 대화형 디지털 홀로그램 통합서비스 시스템의 구현을 위한 신호처리 요소기술 및 하드웨어 IP 개발]

참고문헌

[1] J. Kim, A. Baik, Y. Jung and D. Park, "2D-to-3D image/video conversion by using visual attention analysis," ICIP, 2009.

[2] S. Battiato, A. Carpa, S. Curti and M. La Cascia, "3D Stereoscopic Image Pairs by Depth-Map Generation," Proceedings of 3DPVT, 2004.

[3] W. Tam and L. Zhang, "3D-TV Content Generation: 2D-To-3D Conversion," Proceedings of IEEE ICME, 2006.

[4] M. Sonka and V. Hlavac and R. Boyle, *Image processing, Analysis, and Machine Vision*, 3rd Ed., Thomson Press, 2008.