

다중 블롭 마커를 이용한 스테레오 비전 혼합현실 시스템의 구현

양기선, 김한성, 손광훈
연세대학교 전기전자공학과

전화 : 02-2123-2879 / 핸드폰 : 016-260-0367

Stereo vision mixed reality system using the multi-blob marker

Abstract

This paper describes a method of stereo image composition for mixed reality without camera calibration or complicate tracking algorithm. The proposed system tracks the panel which has blob makers, and composes virtual objects naturally using the method of texture mapping which is often used in geological computer graphics mapping when we do mapping 2D computer graphic data or man-made 2D images.

The proposed algorithm makes it possible for us to compose virtual data even in the case that the panel is bent. For composing 3D object, the system uses depth information obtained from stereo image so that we do not need cumbersome procedure of camera calibration.

I. 서론

혼합현실(Mixed reality)은 가상현실(Virtual reality)의 확장으로, 가상현실은 완전한 가상환경에 사용자가 몰입해 있는 동안은 사용자는 그 주위에 실세계를 볼 수 없는 반면, 혼합현실은 실세계와 합성되어 사용자로 하여금 가상 물체와 실세계를 모두 볼 수 있게 한다. 그러므로 혼합현실은 실세계에 가상현실을

더함으로써 현실감을 더한다고 말할 수 있다[1].

이런 혼합현실은 1960년대에 시작하여 지난 10여 년간에 걸쳐서 특히 많은 어플리케이션으로 개발되었다. 예를 들어, 입체 영상을 볼 수 있는 장비인 HMD(Head Mounted Display)를 이용한 의학용 가상화 장치, 복잡한 장비의 수리와 유지를 위한 가상 설명서, 게임등과 같은 응용분야에서 이미 많은 혼합현실의 예를 보여준 바 있다[2]. 그러나 이러한 방식들은 일반적으로 3-D 추적 장치(Electro magnetic, Ultrasonic, Mechanical tracker 등)의 부가장치나 복잡한 트래킹 알고리즘이 필요하다. 이 외에도 혼합현실은 특히 지난 수년간 영화나 TV등의 디지털 영상 분야에서 많이 적용되어 왔다. 기존에 TV방송용으로 사용된 가상 스크린(Virtual screen), 가상 윈도우(Virtual window)등으로 불리는 것들은 모두 하나의 카메라로부터 획득된 영상에서 색상 패턴 블롭(Blob)을 추적하여 동영상등의 가상 영상을 합성하는 혼합현실 시스템들이다[3]. 이러한 시스템은 보드의 모양이 휘면 투영변환을 시킬 때 개변수 값이 달라지므로 합성에 실패하거나 자연스러운 합성을 할 수 없다. 또, 3차원 물체를 시점에 맞게 정합하기 위해서는 복잡한 카메라 보정(Camera calibration)을 위한 과정이 필요하였다.

본 논문에서는 그러한 문제점들을 해결하기 위해서, 텍스처(Texture)를 분할 매핑(Mapping)함으로써 패턴이 있는 보드가 휘어도 이차원 가상 영상과 보드와의 자연스러운 합성이 가능하게 하였다. 또, 스테레오 영상으로부터 깊이정보를 추정하여, 복원된 3차원 좌표계에서 세 점 이상을 이용한 평면의 법선벡터를 이용하여 3차원 물체를 정합하였다. 그리고 그 결과를 좌우 두 영상에 정합하여 HMD를 이용하여 봄으로써 입

체감을 느낄 수 있는 실감형 혼합 현실 시스템을 구현하였다.

II. 시스템 구성

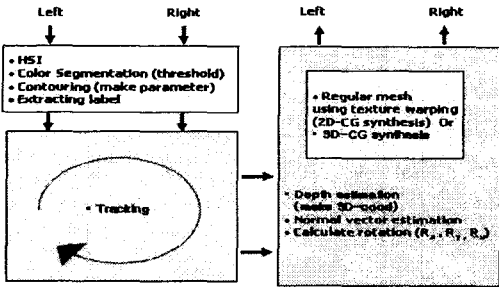


그림 1 시스템 블록도

그림 1의 블록도를 보면 시스템은 크게 블럼 추출, 블럼 추적, 가상 데이터 합성의 세단계로 이루어진다. 먼저, 블럼 추출 단계는 영상을 이진화하여 몇 가지 영상처리 단계를 거쳐 특성 파라미터(둘레길이, 원형도, 사이즈)를 추출한다. 그 다음 이전 프레임에서 추출된 블럼의 중심좌표 주위에 적응형 탐색윈도우를 씌워, 블럼을 추적한다. 마지막으로, 합성 단계에서는 2차원 영상은 텍스처 분할 매핑을 이용하여 합성하고, 3차원 물체는 좌우영상의 특징점으로부터 깊이정보를 추출한 뒤, 세 점의 평면의 법선 벡터를 찾아 그 법선벡터로부터 회전변환을 위한 모델 뷰(Model view) 행렬을 구하여 3차원 물체와 마커 보드와의 자연스러운 합성을 구현한다.

III. 블럼 추출

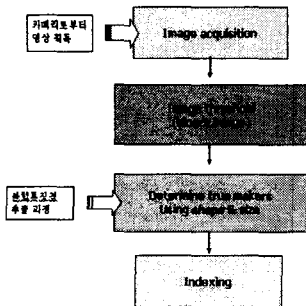


그림 2. 레이블링 및 블럼 추출 과정

그림 2의 블록도는 블럼을 추출하기 위한 과정을 보여준다. 카메라로부터 획득된 영상은 먼저 HSI 영상으로 바꾸어 이진영상으로 변환된다. HSI 영상으로 바꾸는 이유는 색상이 조명의 영향을 받기 때문에 채도(Saturation)대비를 높여 녹색과 파란색을 잘 추출하기

위함이다. 이렇게 추출된 영상은 중간 값(Median) 필터로 노이즈를 제거하고, 각 블럼에 번호가 부여된다. 이런 과정을 레이블링(Labeling)[4]이라고 하는데, 레이블링은 영상을 스캔하면서 영역 확장(Region growing) 기법[5]을 이용하여 같은 영역에 같은 번호를 부여함으로써, 블럼을 구분 할 수 있게 한다. 그 다음, 체인코드(Chain code)알고리즘[5]에 따라 같은 레이블을 가지고 있는 블럼의 둘레를 따라가면서, 다음 수식에 따라 특성 파라미터(면적, 원형도, 중심점)를 계산 한다.

$$\text{면적} = \text{같은 번호가 부여된 픽셀의 수} \quad (1)$$

$$\text{원형도} = e = 4\pi(\text{면적})/(\text{둘레길이})^2 \quad (2)$$

$$\text{중심점} : (u_x, u_y) = \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i, \frac{1}{n} \sum_{i=0}^{n-1} y_i \right) \quad (3)$$

이상과 같은 과정을 통해 입력된 영상으로부터 마커의 위치와 각 독립된 마커의 레이블 번호를 얻을 수 있다. 이러한 정보 즉, 사이즈, 원형도, 둘레길이, 블럼의 중심좌표들이 찾고 싶은 마커들을 구분하기 위해서 사용된다. 그러나, 만약 보드가 카메라로부터 멀어지거나 가까워질 때 사이즈가 달라질 때도 원하는 블럼들만을 선택할 수 있어야 하므로, 영상에서 보드가 차지하는 비율에 따라 다음과 같은 식으로 블럼 사이즈의 문턱치를 결정하였다.

$$S_{\max} = \frac{xy}{M} s_{\max} \quad (4)$$

$$S_{\min} = \frac{xy}{M} s_{\min}$$

$$M = \text{width} \times \text{height}$$

여기서, x는 영상에서 보드의 x 축 길이, y는 보드의 y축 길이를 나타내고, s_{max}는 영상에서 보드가 차지하는 비율이 100% 일 때 블럼의 크기, s_{min}는 영상에서 보드가 차지하는 비율이 1/4 일 때 블럼의 크기를 나타내며, width는 영상의 x 축 길이, height는 영상의 y 축 길이를 나타내므로 고정된 상수 값을 갖는다. 최종적으로, 원형도(0.5<e<0.9) 와 S_{min}< blobs < S_{max} 인 영역만을 원래 찾고자하는 블럼으로 선택하게 된다.

IV. 블럼 추적 알고리즘

블럼을 추적(Tracking)하기 위해서 레이블을 결정할 때 매 프레임마다 위치 좌표를 새롭게 결정한다면

마커 보드가 회전이나 이동되었을 때, 레이블의 번호가 계속 변하게 된다. 이것은 매 프레임마다 처음부터 동일한 작업을 매번 수행하게 하므로 합성하기 까지 시간이 많이 소요되는 원인이 된다. 따라서, 이전 프레임에서 결정된 중심 좌표를 이용하여 다음 프레임에서의 블럽의 중심 좌표를 예측하는 방법을 제안한다. 또한, 블럽이 가려짐으로 인해 소실되었을 때도 이전 프레임과 현재 프레임의 블럽 중심좌표의 변위를 이용하여 가려진 블럽의 위치를 할 수 있다.

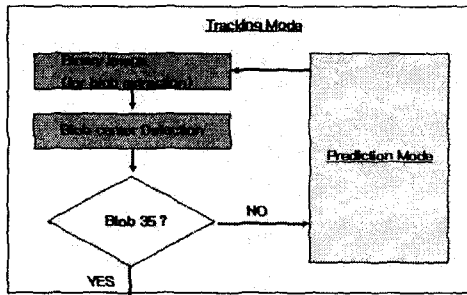


그림 3. 블럽 추적 블록도

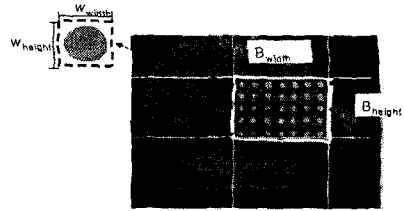
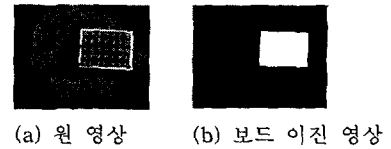
그림 3은 블럽 추적을 위한 블록도이다. 일단 첫 프레임에서는 모든 블럽(35개)이 추출 되어야 한다. 만일 첫 프레임에서 모든 블럽을 찾지 못하면 다음 프레임에서 35개의 블럽들을 찾는다. 이후에 모든 블럽이 발견되어 지면, 그 다음 프레임부터는 추적 모드에 들어가게 된다.

- (1) 보드 이진 영상을 추출, 영상에서 보드의 비율에 따라, 적응형 탐색 윈도우 설정.
- (2) 탐색 윈도우안에 블럽이 만약 존재한다면 이전 프레임의 레이블값으로 영역확장 수행하고 이때 블럽의 파라미터(크기, 원형도, 중심좌표)를 추출함, 만약 존재하지 않으면, 그 블럽을 소실된 블럽(혹은 가려진 블럽)으로 간주.
- (3) (1)~(2) 과정 완료 후 사이즈와 원형도에 따라 찾고자 하는 블럽을 선별하고 만일 제외되었거나 가려진 블럽은 예측모드를 수행.
- (4) 예측모드에서 추적된 블럽의 수가 35개이면 종료하고 (1)번을 반복하고, 그렇지 않으면 (3)번을 반복하여 사라진 점을 예측.

이때, 영상에서 보드의 비율에 따라 탐색 윈도우를 적응형으로 바꾸는 이유는, 마커 보드와 카메라간의 거리에 따라 블럽의 크기가 달라 질 수 있기 때문이다. 그림 5, 식(5)와 같은 방법으로 간단한 동적 탐색 윈도우(Dynamic search window)알고리즘에 따라 탐색 윈도우를 결정함으로써 블럽 추적을 용이하게 할 수 있다.

$$W_{width} = \frac{20}{160} B_{width} \tag{5}$$

$$W_{height} = \frac{20}{160} B_{height}$$



(c) 탐색 영역과 적응형 탐색 윈도우 크기 결정

그림 4. 적응형 탐색 윈도우 크기 결정

또, 가려졌거나 원형도와 사이즈에 의해서 원래 있어야 할 블럽이 제외된 블럽은 예측을 통하여 현재 블럽의 위치를 결정하게 된다. 블럽이 장애물에 가려졌을 경우에 수평 방향으로 가려진 블럽의 위치를 탐색 하면서, 가려진 점이 발견되면 예측할 블럽 주변에 존재하는 블럽들의 중심점의 변위를 이용하여 가려진 마커의 위치를 예측하게 된다. 이때, 이전 프레임과 현재 프레임간의 블럽중심의 방향성 벡터를 고려하여 그 주변의 모든 방향성 벡터의 변화량의 평균값을 적용하여, 가려진 점의 위치를 예측하게 된다. 다음과 같은 식을 적용하였다.

$$\vec{m} = \frac{\vec{a}_1 + \vec{a}_2 + \vec{a}_3 + \vec{a}_4 + \vec{a}_5 + \vec{a}_6 + \vec{a}_7 + \vec{a}_8}{N} \tag{6}$$

가려진 점을 중심으로 주변의 8개점에 대한 변환 벡터를 고려하되, 만일에 주변의 점들 중에서 가려진 점이 존재한다면 예측할 때 제외하고, 나머지 존재하는 변환 벡터들만을 가지고 예측하게 한다.

V. 가상 데이터 합성

- (1) 2차원 가상 영상 과 실영상의 합성

2차원 가상영상을 합성할 때는 가상영상을 분할하여 매핑하였다. 이렇게 하는 이유는 보드가 휘었을 때에도 자연스러운 합성을 가능하게 하기 위해서이다. 만일에 장애물에 보드가 가려졌을 경우, 장애물은 보드와 같이 배경의 일부로 보고, 실제로 보이는 면, 즉 보드에서 파란색과 녹색 부분만을 렌더링(Rendering) 한다.

(2) 3차원 가상 물체와 실영상의 합성

3차원 가상 영상을 합성 할 때는 좌우 영상에서 추출된 특징 점의 깊이 정보를 이용하여 특징 점을 3차원 좌표로 추정하여 세 점 이상으로부터 평면의 법선벡터를 구한다. 그 법선 벡터를 정규화하여 식(7)~(9) 회전 변환을 포함한 모델 뷰(Model view)행렬을 유도하여 합성에 적용하였다.

$$Z = f - \frac{fB}{x_2 - x_1} \quad (7)$$

$$a \times b = [n_x, n_y, n_z] = \begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \quad (8)$$

$$\begin{aligned} R_x &= a \cos(n_z / \sqrt{(n_x^2 + n_z^2)}) \\ R_y &= a \cos(n_x / \sqrt{(n_x^2 + n_z^2)}) \\ R_z &= a \cos(\sqrt{(y_2^2 - y_1^2)} / \sqrt{(x_2^2 - x_1^2)}) \end{aligned} \quad (9)$$

여기서 R_x, R_y 는 법선 벡터를 정규화하여, x축과 y축에 대한 회전각을 계산한 것이다. R_z 는 자신과 인접해 있는 블럭의 중심점 간에 직선의 기울기로부터 계산되었다. 이것으로부터 $R = (R_x, R_y, R_z)$ 과 블럭의 위치 T를 이용하여 모델 뷰 행렬을 찾는다.

$$M = RT \quad (9)$$

VI. 모의실험 결과

모의실험은 OpenGL그래픽 라이브러리, P4 CPU 1.6GHz 컴퓨터, CCD (카메라간격 6cm)카메라 두 대로부터 획득된 스테레오 정지 영상을 이용하여 수행되었다. 실험 결과, 그림 5처럼 합성하는데 걸리는 시간은 한 프레임 당 가리지 않았을 때 약 0.07초, 가렸을 때 약 0.06 초가 소요되었다.



(a-1) 2차원 가상 영상 합성 결과



(a-2) 2차원 가상 영상 합성 결과



(b) 3차원 가상 영상 합성 결과(좌우 영상)

그림 5. 실험 결과

VII. 결론

제안된 시스템은 모의실험을 통해 2차원 가상 영상의 경우 마커가 가렸을 때도 자연스러운 합성이 가능하고, 보드의 휘어짐으로 인한 문제 역시 해결됨을 확인하였다. 또한 스테레오 영상으로부터 마커의 깊이를 추정함으로써, 마커의 포즈(pose)에 따라 실사와 3차원 가상 물체와의 자연스러운 합성이 가능하였다. 본 논문은 스테레오 혼합현실을 위한 방송 시스템에서 콘텐츠의 한 가지 방법으로 활용될 수 있을 것으로 사료된다.

참고문헌

- [1] Ronald Azuma. "A Survey of Augmented Reality.", Presence : Teleoperators and Virtual Environments. Vol. 6. No.4. Aug.1997
- [2] Ronald Azuma, Yohan Baillet, Reinhold Behringer, Steven Feiner, Simon Julier, Blair MacIntyre. "Recent Advances in Augmented Reality.", IEEE. Nov. 2001.
- [3] 박성춘, 남승진, 오주현, 박창섭, "색상패턴 추적을 이용한 실시간 증강영상 시스템", 한국방송공학회 논문지 2002년 제 7권 제 1호, p2~9.
- [4] R. C. Gonzalez, R. E. Wood, Digital Image Processing, second edition, Addison-Wesley publishing company, 1993.
- [5] George C. Stockman, Linda G. Shapiro, Computer Vision, prentice Hall publishing company, 2001