

데스크탑 환경에서의 3차원 상호작용을 위한 비전기반 인터랙션 도구의 설계

최 유 주[†] · 이 선 민^{**} · 유 효 선^{***} · 노 영 섭^{****}

요 약

컴퓨터 그래픽스, 가상현실 및 증강현실 기술이 발전됨에 따라, 이들 기술을 기반으로 하는 다양한 응용 분야에서 3차원 공간에서의 객체 선택 및 조작 등의 3차원 인터랙션 기능들이 요구되고 있다. 본 논문은 고가의 데스크탑용 3차원 마우스 기능을 시뮬레이션 할 수 있는 비전기반의 3차원 인터랙션 프레임워크를 제안한다. 제안 프레임워크는 3색 LED를 이용하여 특수하게 제작된 인터랙션 도구를 포함하며, 비디오 시퀀스로부터 LED의 위치 및 색상을 인식하여 다양한 마우스 이벤트와 6 자유도(Degree Of Freedom)의 인터랙션을 지원한다. 제안 도구는 고가이며 숙련된 조작을 필요로 하는 기존의 3차원 마우스에 비하여 직관적이고 편리하여 별도의 학습이나 훈련 없이 사용될 수 있다. 본 논문에서는 제안 프레임워크를 구성하는 3색 LED를 이용한 포인터 제작 방법, 포인터의 3차원 위치 및 방향 계산법, 비디오 영상에서의 LED 색상 분석 기법에 대하여 설명한다. 또한, 계산된 3차원 위치 및 방향에 대한 오차 측정 결과를 보임으로써 제안 도구의 정확성 및 유용성을 검증한다.

키워드 : 인간 컴퓨터 상호작용, 가상현실, 비전 기반 인터랙션

Design of Vision-based Interaction Tool for 3D Interaction in Desktop Environment

Yoo-Joo Choi[†] · Seon-Min Rhee^{**} · Hyo-Sun You^{***} · Young Sub Roh^{****}

ABSTRACT

As computer graphics, virtual reality and augmented reality technologies have been developed, in many application areas based on those techniques, interaction for 3D space is required such as selection and manipulation of an 3D object. In this paper, we propose a framework for a vision-based 3D interaction which enables to simulate functions of an expensive 3D mouse for a desktop environment. The proposed framework includes a specially manufactured interaction device using three-color LEDs. By recognizing position and color of the LED from video sequences, various events of the mouse and 6 DOF interactions are supported. Since the proposed device is more intuitive and easier than an existing 3D mouse which is expensive and requires skilled manipulation, it can be used without additional learning or training. In this paper, we explain methods for making a pointing device using three-color LEDs which is one of the components of the proposed framework, calculating 3D position and orientation of the pointer and analyzing color of the LED from video sequences. We verify accuracy and usefulness of the proposed device by showing a measurement result of an error of the 3D position and orientation.

Keywords : Human-Computer Interaction, Virtual Reality, Vision-based Interaction

1. 서 론

인터넷 게임 및 인터넷 쇼핑몰 등의 분야에서 게임의 실감성을 높이고, 상품의 내용을 보다 정확하게 전달하기 위한 목적으로 3차원적 그래픽 모델로 구성된 가상공간 구축

사례가 증가되고 있다. 3차원 가상공간은 2차원적 평면의 개념에서 깊이 정보가 추가된 입체 공간이다. 흔히 사용하고 있는 2차원적 마우스는 3 자유도(Degree of Freedom)의 인터랙션에 의한 제한적인 정보만을 제공하기 때문에 이것을 사용하여 6 자유도를 갖는 3차원 가상공간을 네비게이션 한다거나 가상공간상의 입체객체에 대한 조작을 수행하는 데는 많은 어려움이 따르고 있다. 이에 따라 3차원 가상공간에 적합한 편리한 인터랙션 방법에 대한 관심이 높아지고 있다.

3차원 가상공간 및 입체 모델에 대한 조작을 위하여 제작된 3차원 인터랙션 상용 장비들은 사용자 전신 움직임이 추적되는 대형 작업 공간을 기반으로 하는 형태와 데스크 탑 어플리케이션에 적합한 소형 작업 공간을 기반으로 하는 형

* 본 연구는 2006년 교육인적자원부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-331-D00524)

† 종신회원 : 서울벤처정보대학원대학교 컴퓨터응용기술학과 교수

** 준 회원 : Computer Science Department, University of Southern California, Postdoctoral Researcher

*** 준 회원 : 서울벤처정보대학원대학교 컴퓨터응용기술학과 연구원

**** 정 회원 : 서울벤처정보대학원대학교 임베디드시스템학과 교수

논문접수 : 2008년 4월 2일

수정일 : 2008년 5월 2일

심사완료 : 2008년 5월 7일

대로 구분할 수 있다. 데스크 탑 소형 작업 공간에 적합한 3차원 인터랙션 입력 장치는 전자기적 기법을 기반으로 하는 것이 대부분이다. 전자기적 3차원 입력 장치는 응용 어플리케이션이 쉽게 사용할 수 있도록 편리한 개발 라이브러리 등을 함께 제공하고 있으나, 대부분 일반 2차원 입력장치(2차원 마우스, 키보드 등)에 비해 고가의 장비로서 구입 부담이 크다. 또한, 3차원 입력장치에 친숙하지 않은 사용자는 자유롭게 입력 장치를 다루기 위해서 사용법에 대한 학습과 훈련의 시간이 필요하다. 이는 3차원 입력 장치의 사용기법이 직관적이지 않기 때문이다.

이에 본 논문에서는 3차원 가상공간 기반 어플리케이션에 쉽게 적용할 수 있고, 별도의 학습 및 훈련이 요구되지 않는 “저가형 비전 기반 3차원 인터랙션 프레임워크”를 제안한다. 제안 프레임워크는 3색 LED를 사용하여 특수하게 제작된 인터랙션 도구와 두 대의 웹캠, 그리고 카메라를 통해 입력된 영상을 분석하여 이벤트를 인식하는 소프트웨어 엔진으로 구성된다. 본 논문에서는 3색 LED를 기반 한 인터랙션 포인트의 제작 내용과 색상 LED들의 위치와 색상을 추적하여 3차원 마우스에 필요한 기능 이벤트를 추출하기 위한 비전처리 내용을 소개한다. 또한, 처리결과 추출된 LED 포인트의 3차원 좌표 및 회전 각도에 대한 오차율을 측정하여 제안 3차원 인터랙션 프레임워크의 정확도 및 유용성을 검증하고자 한다.

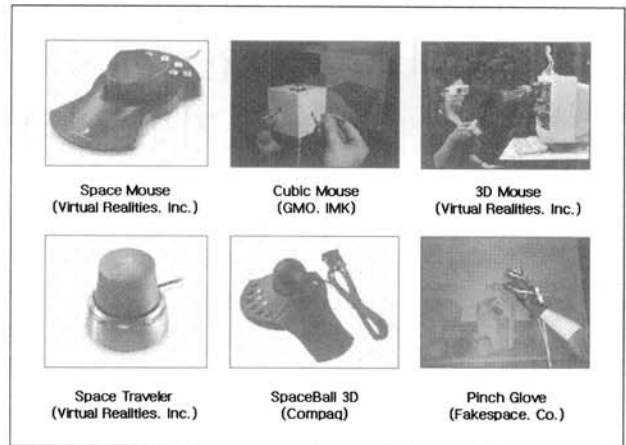
본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 제시하고, 3절에서는 제안 프레임워크의 구성 개요를 살펴본다. 4절에서는 인터랙션 제어를 위한 비전 기반 처리 절차를 제시하고, 5절에서는 실험결과를 보인다. 마지막으로 6절에서 결론을 논한다.

2. 관련연구

데스크 탑 소형 작업 공간에 적합한 3차원 인터랙션 입력 장치는 전자기적 기법을 기반으로 하는 것이 대부분이다. 1990년대말 이후 데스크탑 응용 어플리케이션에 적용하기 위하여 비전 기반 핸드 제스처에 대한 연구가 활발하게 진행되어 오고 있으나, 대부분 프로토타입 시스템의 제시 수준에 머물러 있다[1-6]. 데스크탑 작업 공간에서 널리 사용되고 있는 대표적인 입력 장비로 DataGlove나 3차원 trackball 장비등을 들 수 있다.

가상현실 어플리케이션에 적용되어 널리 사용되고 있는 대표적인 상용 3차원 인터랙션 도구를 정리하면 (그림 1)과 같다. 전자기적 3차원 입력 장치는 응용 어플리케이션이 쉽게 사용할 수 있도록 편리한 개발 라이브러리 등을 함께 제공하고 있으나, 대부분 일반 2차원 입력장비에 비해 고가의 장비로서 구입 부담이 크고, 3차원 입력장치에 친숙하지 않은 사용자는 자유롭게 입력 장치를 다루기 위해서 사용법에 대한 학습과 훈련의 시간이 요구된다.

DataGlove나 비디오 영상을 기반으로 한 핸드 제스처 추적에 의한 인터랙션은 적용 어플리케이션 별로 손에 의해



(그림 1) 전자기적 3차원 상용 장비들

내리는 명령의 형태가 서로 다르기 때문에 사용자는 어플리케이션 별로 제스처 명령을 학습하여야 하고, 특히 Data-Glove의 경우, 착용의 불편함이 있다.

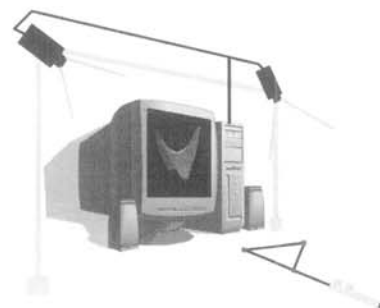
본 논문에서 제안하는 형태와 유사한 비전 기반 인터랙션에 대한 연구로서는 적외선 소스에 반사되는 반사 마커를 사용하여 인터랙션 도구의 위치를 추적하는 접근 방법들이 있다[7,8]. 그러나 기존 접근방법들은 마커의 위치 추적에만 초점을 맞추고 있고, 마우스의 클릭, 드래그와 같은 인터랙션 도구가 가져야 할 다양한 종류의 이벤트들을 지원하지 못하고 있다. 이에 2차원 마우스 사용에 친숙한 일반 사용자가 임의의 응용 어플리케이션에서 쉽게 사용할 수 있는 저가형 범용 3차원 인터랙션 도구의 개발 및 이를 쉽게 일반 응용 어플리케이션에 접목 시킬 수 있는 인터랙션 프레임워크의 구축이 요구되고 있다.

3. 비전기 기반 3차원 인터랙션 프레임워크 구성

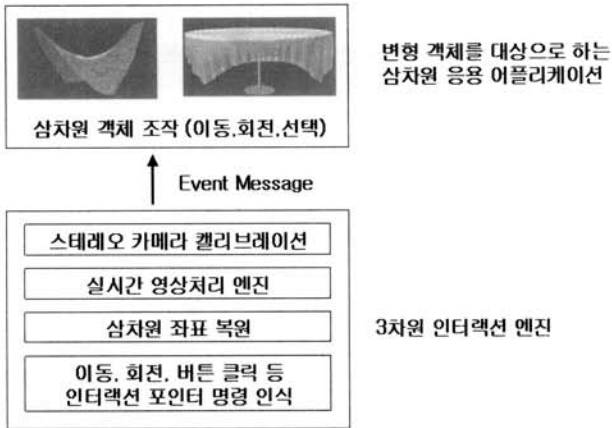
3.1 시스템 개요

본 논문에서 제안하는 3차원 인터랙션 시스템은 데스크탑 환경에서 사용할 수 있도록 고안된 것으로 기존의 핸드 제스처 기반 인터랙션에 비하여 범용적으로 활용할 수 있도록 환경에 제약을 덜 받으면서도 안정적인 인식결과를 도출할 수 있는 LED 마커 기반 방식을 채택하였다.

제안 시스템의 하드웨어 구성도는 (그림 2)와 같다. 두 대의 저가형 웹캠을 모니터 양 옆에 설치하고 3색 LED 기반



(그림 2) 제안 인터랙션 시스템의 하드웨어 구성



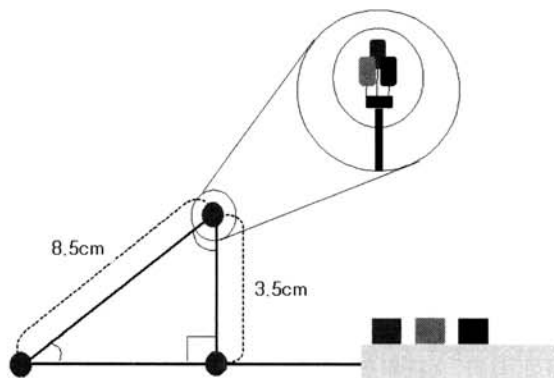
(그림 3) 인터랙션 제어 소프트웨어 엔진 및 응용 어플리케이션 구성도

인터랙션 포인터에 대한 스테레오 영상을 획득한다. 획득된 2차원 영상에서 인터랙션 포인터의 공간적 위치 및 회전 등의 정보를 추출함으로써 가상객체에 대한 3차원 인터랙션을 수행할 수 있도록 한다. (그림 3)은 이를 지원하기 위해 필요한 인터랙션 제어 엔진 및 응용 어플리케이션의 소프트웨어 구성도를 보여 준다. 인터랙션 제어 엔진의 상세한 내용은 4절에서 논하기로 한다.

3.2 3색 LED 기반 인터랙션 포인터

본 연구에서는 2D 마우스의 왼쪽, 오른쪽 마우스 버튼의 기능을 3차원 공간에서 가능하도록 시뮬레이션한 3차원 인터랙션 포인터를 제작하였다. 제작 포인터는 눌러진 버튼에 따라 서로 다른 색 (빨강, 초록, 파랑)을 발산하는 3개의 색상 LED가 정해진 각도와 거리의 삼각형 형태로 연결되어 있어서, 포인터의 이동 뿐 만 아니라 회전의 방향도 추정할 수 있다. 이동 및 회전의 인식 방법은 4절에서 논한다. (그림 4)는 제안 3차원 인터랙션 포인터의 크기와 각 인터랙션 볼의 배치 각도 및 인터랙션 볼 안의 내부 구조를 보여주고 있다.

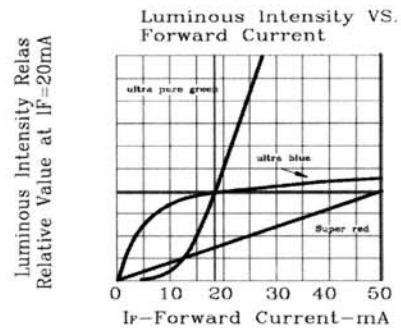
3차원 인터랙션 포인터는 3차원 움직임을 파악할 수 있도록 세 지점에 고휘도의 LED를 배치하고 각각을 사용자의 선택(Red, Green, Blue 버튼)에 따라 세 가지 형태의 점등 (Red-Red-Red, Red-Green-Green, Red-Blue-Blue)이 가능



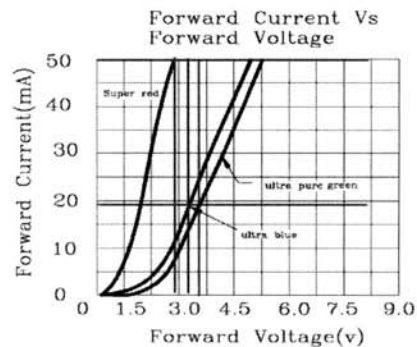
(그림 4) 색상 LED를 포함한 3차원 인터랙션 포인터

하도록 설계하였다. 그런데 RGB의 3가지 색을 표현할 수 있는 LED는 3가지 색상의 LED 각각을 하나로 모아서 만들어 놓은 구조를 가지고 있기 때문에 각각의 LED의 점등에 필요한 구동전류와 구동전류에 따른 밝기가 서로 다르다. 또한 제어 소프트웨어에서도 밝기가 다른 광원을 처리하는 것이 어렵다. 따라서, 본 연구에서는 각각의 광원인 LED의 색상별 밝기(brightness)를 동일하게 설계하였고, LED 인지 프로그램에서 인식하기에 적당한 밝기값을 찾아내어, 안정된 밝기값으로 LED가 구동되도록 하였다.

실제 3차원 인터랙션 포인터에 사용한 LED(599R2GBC-CA)의 경우 구동 전류에 따른 3가지 색상의 밝기는 아래의 (그림 5)와 같으며, 이 경우 녹색과 파랑색은 약 18mA에서, 빨강색은 약 50mA에서 용이하게 비슷한 밝기를 갖게 만들 수 있음을 알 수 있다. 이를 근거로 (그림 6)에서 각각의 순방향 전압(forward voltage)을 결정하면 빨강색은 약 2.9V, 파랑은 약 3.3V, 녹색은 약 3.6V이고, 결정된 순방향 전류(forward current)와 순방향 전압(forward voltage)으로 전류 제한(current limit)용 저항의 저항값을 결정할 수 있다. 그리고 저항은 모든 종류의 값들이 사용되는 것이 아니기 때문에 계산된 값에서 가장 근사한 표준값을 사용하였다. 참고로 전류 제한용 저항값은 아래의 <표 1>과 같이 계산되고 선택되었다.



(그림 5) 구동전류에 따른 세 가지 색상 밝기



(그림 6) 색상별 순방향 전압과 전류

<표 1> 전류제한(current limit)용 저항의 계산, 전원전압 = 6.2V

| | R | G | B |
|----------------------------|-----|-----|-----|
| 순방향전류(I _F , mA) | 50 | 18 | 18 |
| 순방향전압(V _F , V) | 2.9 | 3.6 | 3.3 |
| 계산된 저항값(R, ohm) | 66 | 144 | 161 |
| 사용된 저항값(R, ohm) | 75 | 150 | 150 |

3.3 비전기반 인터랙션 제어 엔진

3차원으로 재구성된 포인터의 위치 및 회전 정보를 통한 가상객체의 인터랙션 제어 엔진은 크게 4가지 기능 모듈로 구성된다. 첫 번째는 스테레오 카메라 캘리브레이션 모듈로서 카메라의 내부 및 외부 파라미터를 추출한다. 카메라 캘리브레이션은 두 대의 카메라를 설치하고, 전처리 작업으로 수행된다.

두 번째는 연속으로 입력되는 카메라 영상에서 각각의 LED 영역 및 색상을 추출하기 위한 영상처리 모듈이다. 영상처리 모듈은 필터를 통하여 실시간으로 입력되는 카메라 영상에서 LED의 영역을 추출하고 추출된 영역의 색상, 즉 LED의 색상을 인식하기 위한 것으로서, 임계값 기반 영역 추출과 connected component labeling을 적용한다.

세 번째는 3차원 좌표 및 회전각도 추정 모듈로서, 우선 추출된 LED 포인트에 대해 포인트간 거리값을 기준으로 하여 각 LED에 대한 ID를 부여하고, 동시에 촬영된 좌우 카메라 영상에서 대응 LED를 정의한다. 대응 LED의 2차원적 좌표값과 카메라 캘리브레이션 모듈을 통해 얻어진 카메라의 내부 및 외부 파라미터 값을 기반으로 삼각화 기법(triangulation)을 이용하여 각 LED 포인트들의 3차원적 좌표를 계산하고, 세 개의 LED 포인트들이 이루는 평면의 변화를 분석하여 회전 각도를 계산한다. 영상에서 인터랙션 장비의 3개의 포인터중 하나 이상의 포인터가 분실될 경우 분실된 포인터에 대한 위치를 추정하기 위해 외삽법(Extrapolation)기법을 이용하여 3차원 좌표를 추정하고 매끄러운 포인터의 좌표이동을 위하여 칼만 필터 기법을 사용하여 3차원 좌표 및 각도를 보정한다.

네 번째는 LED의 위치변화 및 색상 분석을 기반으로 이벤트를 인식하여 응용 어플리케이션으로 이벤트 메시지를 전송하는 모듈이다. LED 색상 조합에 따라 포인터의 이동(navigation), 객체 회전(rotation), 객체 이동(translation) 이벤트로 분류되고, 각 LED 포인트에 대한 3차원 좌표 및 색상, LED 포인트의 회전 각도등이 이벤트 파라미터로 전달된다. 인터랙션 제어 엔진의 각 단계별 처리절차는 4장에서 자세히 다루도록 하겠다.

4. 인터랙션 제어엔진 처리 절차

4.1 스테레오 카메라 캘리브레이션

두 대의 카메라로 획득된 입력영상을 이용하여 목표 물체의 3차원 공간상에서의 위치 및 방향을 추정하기 위해서는 카메라 캘리브레이션이 우선적으로 수행되어야 한다. 카메라 캘리브레이션이란 카메라 자체 특성을 나타내는 내부 정보(intrinsic parameter)와 세계좌표계(world coordinate)와 카메라 좌표계(camera coordinate) 간의 상관관계를 추출하기 위해 필요한 카메라 위치 및 방위로 구성된 외부 정보(extrinsic parameter)를 찾는 과정을 일컫는다[9,10].

본 연구에서는 Tsai[11]가 제안한 방법을 이용하여 스테레오 카메라에 대한 캘리브레이션을 수행하였으며, 이를 위

하여 사전에 정의된 기준점들에 대한 실세계 좌표 $P(X_w, Y_w, Z_w)$ 와 이 점들에 대한 영상에서의 좌표 $pi(x_i, y_i)$ 를 이용하여 실세계 좌표계, 카메라 좌표계, 이미지 좌표계로의 변환 행렬을 계산함으로써 카메라의 내·외부 파라미터를 추출한다.

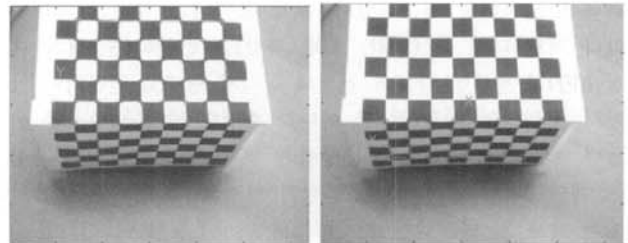
본 연구에서는 아래의 세 단계를 수행하여 카메라의 내부 및 외부 파라미터를 추출하였다.

■ 1단계: 캘리브레이션 패턴 제작 및 영상 획득

본 연구에서는 5cm의 간격으로 그리드가 그려진 평면 캘리브레이션 패턴을 수직으로 연결하여 공간적인 패턴을 만들고, 캘리브레이션 패턴 부착 박스가 두 카메라 영상의 중앙에 오도록 박스의 위치 및 방향을 조정하여 스테레오 영상을 획득한다.

■ 2단계: 캘리브레이션 패턴 내 격자점에 대한 영상좌표 추출

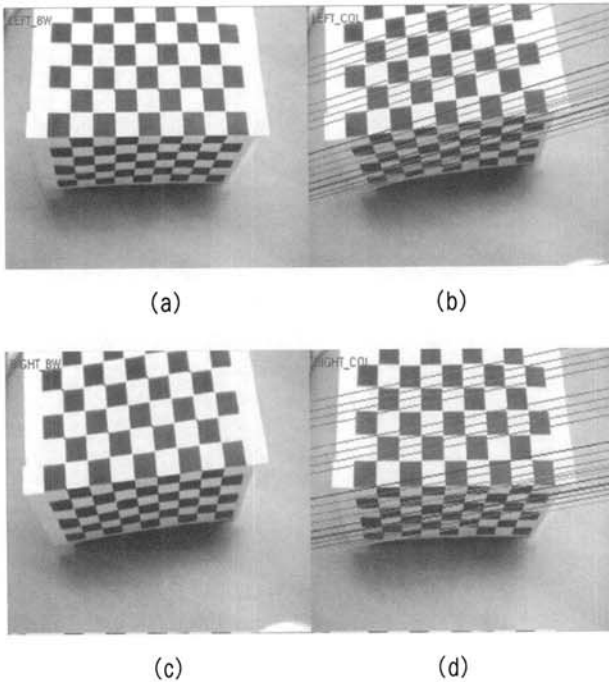
캘리브레이션 패턴 내의 각 격자점에 대한 영상에서의 좌표를 추출한다. 이 때, 격자점 중 하나를 세계좌표계 상에서의 원점으로 지정하고 나머지 격자점에 대하여 위에서 사전에 지정한 간격(50mm)을 이용하여 세계좌표계에서의 좌표 값을 부여한다. 이 때, 오른손 좌표계를 사용하였다. 각 격자점에 대하여 이차원 영상에서의 좌표값을 추출하고 이를 이용하여 입력 파일을 생성한다. (그림 7)은 왼쪽 영상에서 두 개의 평면에 대한 그리드 포인트 추출 영상을 보여주고 있다.



(그림 7) 격자점 추출 결과

■ 3단계: 캘리브레이션 수행

(그림 7)의 왼쪽과 오른쪽 카메라 영상에서 추출된 격자점의 2차원 영상 좌표와 대응되는 세계 좌표계의 좌표를 입력으로 하여 Tsai가 제안한 방법에 의하여 캘리브레이션을 수행함으로써 카메라 내·외부 파라미터를 계산한다. 추출된 카메라 파라미터의 정확성을 검증하기 위하여 한 영상 내 격자점에 대한 에피폴라 라인을 계산하여 다른 영상에 표시한 결과가 (그림 8)에서 보여지고 있다. 에피폴라 라인이 해당 격자점에 대한 다른 영상에서의 대응점을 정확하게 통과하는 경우 카메라 캘리브레이션이 정상적으로 수행되었음을 알 수 있다. (그림 8)(a), (c)는 왼쪽 및 오른쪽 카메라 영상에서의 캘리브레이션 패턴 내에 추출된 대표 격자점들을 각각 보여주고 있으며, (그림 8)(b), (d)는 이에 대한 에피폴라 라인이 각각 오른쪽 및 왼쪽 카메라 영상에서의 대응점을 지나가고 있음을 보여 주고 있다. <표 2>는 캘리브레이션에



(그림 8) 스테레오 영상에서의 캘리브레이션 패턴 대표점과 이에 대한 에피폴라 라인: (a)(c) 왼쪽 및 오른쪽 영상에 대한 대표 격자점 추출 결과, (b)(d) 대표 격자점에 대한 오른쪽 및 왼쪽 영상에서의 에피폴라 라인

<표 2> 좌측 영상에 대한 캘리브레이션 수행 결과 계산된 파라미터 값

| | |
|-----------|------------------------------------|
| 초점거리 | 4.239433 |
| 카메라 화소 크기 | 0.007400 |
| 영상 중심 | 160.125140, 119.841867 |
| 이동 | -105.125130, -2.168138, 660.468155 |
| 회전 | 68.186211, -6.069125, -0.603749 |

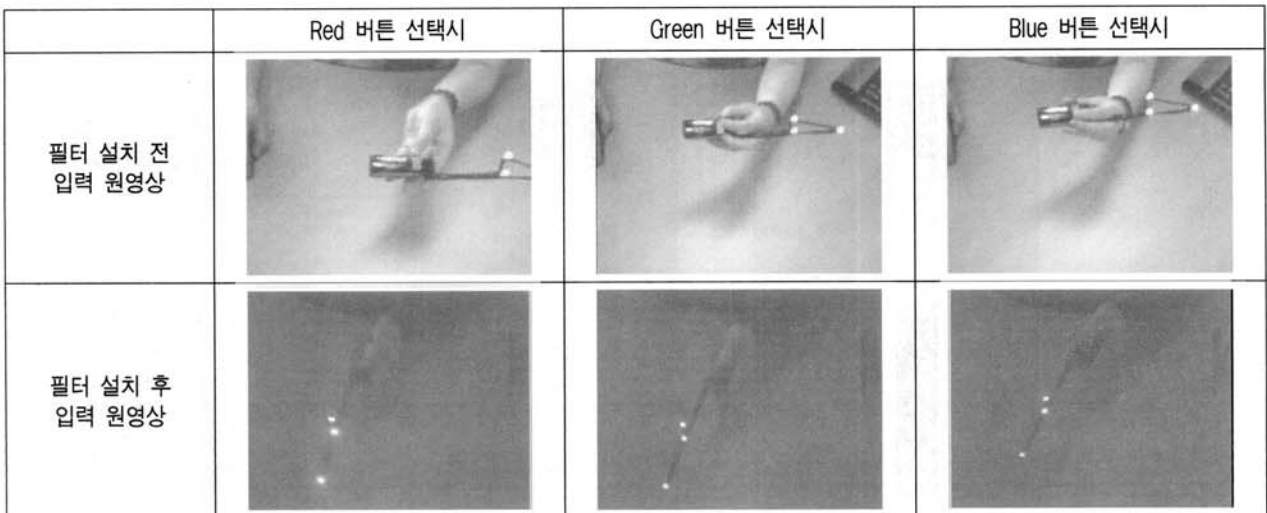
의해 추출된 카메라 내·외부 파라미터 결과 예를 보여주고 있다.

4.2 LED 포인터 3차원 좌표 및 회전 각도 인지

실시간 연속으로 입력되는 카메라 영상을 기반으로 LED 포인터 위치를 추적하기 위하여 우선, LED 포인터를 구성하는 세 개의 LED의 영역을 입력 카메라 영상에서 추출하는 것이 필요하다. 이 때, 외부 조명의 영향을 최소화 하면서 LED의 영역만 인지 할 수 있도록 하기 위하여, 두 대의 웹캠에 높은 밝기의 화소만을 통과 시킬 수 있는 필터를 부착한다. 필터를 부착한 카메라 영상은 LED의 색상은 유지 하면서, 나머지 주변 환경 영상은 낮은 밝기값으로 제거된 형태로 입력된다. (그림 9)는 필터 설치 전·후 영상을 비교하고 있다.

4.2.1 조명 독립적 LED 영역 추출

조명 독립적으로 LED 포인터의 위치를 추적하기 위해 첫 번째 영상처리 단계에서 필터링된 영상을 흑백영상으로 변환하고, 변환된 영상에 대하여 임계값에 의한 이진화를 수행한다. 이진화된 영상을 라벨링하는 방법은 다양하게 있으며 그 대표적인 것으로 Glassfire 알고리즘이 있다. 그러나 이 알고리즘은 재귀적인 연산을 하기 때문에 구동 시에 계산비용이 크다는 단점이 있다. 동영상 기반의 이진화된 영상에서 특정 영역을 추출하기 위해서 연산시간비용이 적은 CCL(Connected Component Labeling)기법을 적용한다. 이 기법은 기존 알고리즘과 달리 이웃한 픽셀을 비교하여 라벨링하기 때문에 연산시간이 기존 알고리즘보다 빠르다는 장점이 있다. 본 연구에서는 CCL알고리즘을 적용하여 이진화 결과영상에 대하여 각각의 LED의 중심점을 추적한다. (그림 10)은 입력 영상에 대한 임계값에 의한 이진화 처리 결과 및 레이블링 알고리즘 수행 후 추적된 LED 포인트 ROI 영역을 보여주고 있다. ROI는 흰색 사각형으로 표시되었다.



(그림 9) 필터 설치 전·후의 입력 원영상

| | Red 버튼 선택시 | Green 버튼 선택시 | Blue 버튼 선택시 |
|-------------------------|------------|--------------|-------------|
| 입력 영상 | | | |
| LED 포인트 관심영역(ROI) 추적 영상 | | | |

(그림 10) LED 포인트 각각에 대한 관심영역(ROI) 추적 결과

4.2.2 LED ID 판별

좌·우 카메라 영상 각각에서 LED의 ID를 판별하기 위하여 각 LED 포인트의 관심영역의 중심 P_1, P_2, P_3 을 구한다. 각각의 영상에서 추적된 세 개의 LED 중심 간 거리를 계산하여 가장 긴 거리를 가지는 LED 중심의 집합(P^{max})과 가장 짧은 거리를 가지는 LED 중심의 집합(P^{min})을 식(1)과 식(2)와 같이 찾아낸다. P^{max} 와 P^{min} 집합 모두에 포함된 LED를 LED 1 (P1)로 정의하고, 짧은 거리를 가지는 LED 집합(P^{min})에 속한 다른 LED를 LED 2 (P2)로, 가장 긴 거리를 가지는 LED 집합(P^{max})에 속하는 다른 하나의 LED를 LED 3 (P3)으로 정의한다. LED ID를 판별하는 방법을 식으로 정리하면 식(3), 식(4), 식(5)와 같다. (그림 11)은 좌·우 카메라 영상에서 LED ID 판별 결과를 보여 주고 있다.

$$P^{min} = \{P_k, P_l\}, \overline{P_k P_l} = \text{MIN}(\overline{P_i P_j}), \quad (1)$$

$$1 \leq i \leq 3, 1 \leq j \leq 3, i \neq j, k = i, l = j,$$

$$P^{max} = \{P_k, P_l\}, \overline{P_k P_l} = \text{MAX}(\overline{P_i P_j}), \quad (2)$$

$$1 \leq i \leq 3, 1 \leq j \leq 3, i \neq j, k = i, l = j,$$

$$P1 = \{P_i | P_i \in (P^{min} \cap P^{max})\}, \quad (3)$$

$$P2 = \{P_i | P_i \in P^{min}, P_i \neq P1\}, \quad (4)$$

$$P3 = \{P_i | P_i \in P^{max}, P_i \neq P1\}. \quad (5)$$

4.2.3 삼각화 기법을 이용한 3차원 좌표 추정

좌·우 영상에서의 대응점 P_1, P_2, P_3 을 이용하면 삼각화 기법(triangulation)[12]에 의하여 각 점에 대한 3차원 공간상의 좌표를 계산할 수 있다. 대응점 중 한 영상좌표를 $p_i(x_i, y_i)$, 캘리브레이션을 통하여 추출한 이차원 영상의 중심을 (c_u, c_v) , 카메라 초점을 f 라 하면, 3차원 공간상에서의 좌표를 $P_w(X_w, Y_w, Z_w)$ 는 식(6) - (10)을 이용하여 근사시킬 수 있다. P_w 는 p_i 를 카메라 좌표계로 변환한 후, 카메라 중심과 연결한 벡터의 연장선 상에 존재하게 된다. 카메라 좌표계에서의 카메라 원점과 대응점을 연결하는 벡터의 연

| | Red 버튼 선택시 | Green 버튼 선택시 | Blue 버튼 선택시 |
|--------------|------------|--------------|-------------|
| 입력 영상 | | | |
| LED ID 판별 결과 | | | |

(그림 11) LED ID 판별 결과

장선은 식 (6)과 같이 표현할 수 있다. 따라서 두 대응점을 식 (6)에 의해 각각 표현한다면, 두 벡터가 교차하는 지점이 두 대응점의 3차원 공간상에서의 좌표가 된다.

식 (7)는 렌즈 왜곡이 없다고 가정할 때, 대응점을 카메라 좌표계에서 세계좌표계로 변환하는 과정이며, 이 때 R은 카메라 외부파라미터인 회전벡터이다. 식 (8)은 카메라 중심과 대응점을 연결한 벡터를 연장한 선을 세계좌표계로 표현한 것이다. 이 때 O는 카메라 중심이며, λ는 임의의 실수이다.

$$p = \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \lambda \begin{pmatrix} (x_i - c_u)/f \\ (y_i - c_v)/f \\ 1 \end{pmatrix} \quad (6)$$

$$w = R^t \begin{pmatrix} (x_i - c_u)/f \\ (y_i - c_v)/f \\ 1 \end{pmatrix} \quad (7)$$

$$l = \{O + \lambda w\} \quad (8)$$

식 (8)에 의하여, 두 대응점에 해당하는 선을 각각 l^L, l^R 라고 표현하면, 두 선분이 만나는 $\lambda = (\lambda^L, \lambda^R)^t$ 를 구하면 3차원 공간상에서의 좌표를 계산할 수 있다. 이 때, 카메라 캘리브레이션 오차 등에 의하여 두 벡터가 정확하게 만나지 않는 경우가 존재하므로, 식 (9)를 이용하여 두 선분의 차이가 가장 적은 λ를 구하여 근사시킨다.

$$|O^L + \lambda^L w^L - O^R + \lambda^R w^R|^2 \quad (9)$$

이 때, $b = O^L - O^R, A = [w^L - w^R]$ 이라 할 때, λ는 식(10)에 의하여 구할 수 있다.

$$\lambda = -(A^t A)^{-1} A^t b \quad (10)$$

(그림 12)는 계산된 3차원 좌표를 인식된 LED ID와 함께 표현한 결과 영상이다.

4.2.4 소실된 LED 포인트에 대한 좌표추정

3차원 인터랙션으로 가상객체의 세밀한 조작용을 위해서는 인터랙션 도구의 3차원 위치 및 회전 각도의 정확성을 요구한다. 그러나 영상처리 단계에서 획득된 LED의 색상과 거리 등을 고려하여 각 LED 포인트를 정의하고, 대응되는 각 LED 포인트의 3차원 좌표를 복원할 때 다른 물체에 의해 LED 포인트가 가려지면, 두 대의 카메라로부터 얻어진 스테레오 영상에서 세 개의 LED가 잡히지 않는 경우가 있다. 이 때, 사라진 LED 포인트의 3차원 궤적을 유지시키기 위한 좌표 추정이 필요하다. 이를 위하여 이전의 여러 프레임에서의 LED의 위치 및 회전 정보를 저장하여 현재 프레임에서 가려진 인터랙션 볼을 추정하는 기술로 보외법(Extrapolation)을 사용한다. 가장 간단한 추정방법으로 식(11)의 일정 외삽법(Constant extrapolation)을 적용할 수 있다. 이 식은 이전의 좌표값으로 현재의 3차원 위치값을 대신한다.

$$P^{est}(t) = P^n(t-1) \quad (11)$$

시간 t에서의 삼차원 위치값 $P^{est}(t)$ 은 이전 시간 t-1의 삼차원 위치값 $P^n(t-1)$ 으로 대체된다. 이 계산법은 추가의 계산과정이 필요하지 않다는 장점이 있다. 그러나 이 식을 이용하여 현재 좌표를 예측할 경우, 이전의 이동과정에 대한 정보를 사용하지 않아 보다 신뢰성 있는 좌표 계산이 어렵다는 한계를 가진다.

이전 프레임들로부터 얻어진 불연속적인 좌표 값으로부터 추정값을 산출하기 위해 추정대상 프레임 이전의 2개 이상의 프레임에서의 LED 포인트의 좌표를 사용해 현재 좌표를 유추한다면 보다 효과적인 추정을 할 수 있다. 본 연구에서는 이전 점들에 대한 가중평균을 이용하여 다음 값을 추정하는 방법인 선형보외법(linear extrapolation)을 사용하며 그 식은 다음과 같다.

$$P_t = \frac{t - t_n}{t_n - t_{n-1}} (P_n - P_{n-1}) + P_n, \quad t_{n-1} \leq t_n < t \quad (12)$$

| | Red 버튼 선택시 | Green 버튼 선택시 | Blue 버튼 선택시 |
|--------------|--|---|--|
| 입력 영상 | | | |
| 3차원 좌표 계산 결과 | $\begin{matrix} 1[198, 1262, -40] \\ 2[198, 1246, -34] \\ 3[182, 945, -96] \end{matrix}$ | $\begin{matrix} 1[265, 2733, 449] \\ 2[213, 1589, 110] \\ 3[185, 1030, -44] \end{matrix}$ | $\begin{matrix} 1[312, 3570, 605] \\ 2[352, 4314, 1030] \\ 3[204, 1395, 111] \end{matrix}$ |

(그림 12) 3차원 좌표 계산 결과

P_t 는 시간 t 에서의 추정 대상 프레임의 3차원 위치값이며, 이전 프레임에서의 값들은 P_n 과 P_{n-1} 이다. P_n 은 이전의 시각 t_n 에서의 위치값이며, P_{n-1} 은 t_n 의 이전 시각 t_{n-1} 에서의 위치값이다. 이 계산은 일정 외삽법 보다 나은 결과를 제공하며 실시간으로 적용하였을 때에도 안정적인 좌표 추정이 가능하다.

4.2.5 회전 각도 인지

인터랙션 포인트의 회전 각도를 인지하기 위하여 회전이 시작되는 초기 시간 t^0 와 t^n 시점에서 세 개의 LED의 중점이 만드는 두 개의 평면을 정의하고, 두 평면간의 각도를 계산한다. 두 평면간의 각도는 각 평면의 법선 사이의 각과 일치하므로 우선 초기 시간 t^0 에 세 개의 LED의 중점이 만드는 평면의 법선 \vec{a} 와 t^n 시점에 세 개의 LED의 중점이 만드는 평면의 법선 \vec{b} 를 정의하였다. 초기시간 t^0 에 세 개의 LED 중점 P_1^0, P_2^0, P_3^0 와 t^n 시점에서 세 개의 LED의 중점 P_1^n, P_2^n, P_3^n 의 좌표가 각각 $P_1^0=(x_1^0, y_1^0, z_1^0), P_2^0=(x_2^0, y_2^0, z_2^0), P_3^0=(x_3^0, y_3^0, z_3^0), P_1^n=(x_1^n, y_1^n, z_1^n), P_2^n=(x_2^n, y_2^n, z_2^n), P_3^n=(x_3^n, y_3^n, z_3^n)$, 라 할 때 법선 \vec{a} 와 \vec{b} 는 각각 식(13)과 식(14)와 같이 정의하였다.

$$\vec{a} = \vec{A}^0 \times \vec{B}^0 = (y_b^0 z_b^0 - z_a^0 y_b^0, z_a^0 x_b^0 - x_a^0 z_b^0, x_a^0 y_b^0 - y_a^0 x_b^0) \quad (13)$$

여기서, $\vec{A}^0 = (x_2^0 - x_1^0, y_2^0 - y_1^0, z_2^0 - z_1^0) = (x_a^0, y_a^0, z_a^0),$
 $\vec{B}^0 = (x_3^0 - x_1^0, y_3^0 - y_1^0, z_3^0 - z_1^0) = (x_b^0, y_b^0, z_b^0)$ 이다.

$$\vec{b} = \vec{A}^n \times \vec{B}^n = (y_b^n z_b^n - z_a^n y_b^n, z_a^n x_b^n - x_a^n z_b^n, x_a^n y_b^n - y_a^n x_b^n) \quad (14)$$

여기서, $\vec{A}^n = (x_2^n - x_1^n, y_2^n - y_1^n, z_2^n - z_1^n) = (x_a^n, y_a^n, z_a^n),$
 $\vec{B}^n = (x_3^n - x_1^n, y_3^n - y_1^n, z_3^n - z_1^n) = (x_b^n, y_b^n, z_b^n)$ 이다.

평면 $\Delta P_1^0 P_2^0 P_3^0$ 와 평면 $\Delta P_1^n P_2^n P_3^n$ 이 이루는 각도 θ 는 식(15)와 같이 두 평면의 법선 벡터에 대한 내적 연산으로 얻어진 값을 적용한다.

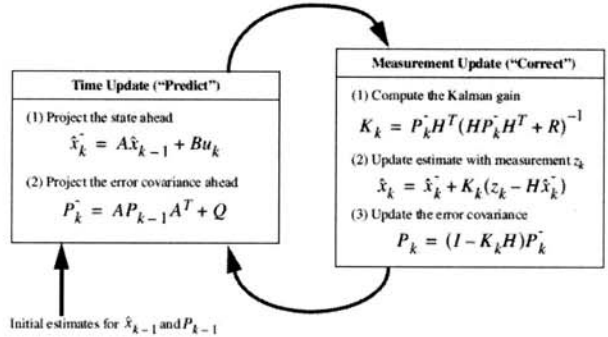
$$\theta = \cos^{-1} \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (15)$$

4.2.6 칼만 필터를 이용한 3차원 좌표 보정

칼만필터[13]는 공학분야에서 널리 쓰이고 있으며, 센서들로부터 들어오는 같은 변수의 값을 연결하거나 또는 불완전하게 측정된 상태값을 통해 시스템을 부정확하게 예측된 값의 결합에 사용된다. 본 논문에서 제안하는 위치 추적 시스템에 칼만필터를 적용함으로써 스테레오 카메라로부터 획득한 LED 포인트의 3차원 좌표 추정시의 잡음을 최소화한다.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

$$z_k = Hx_k + v_k$$



(그림 13) 칼만필터 사이클 프로세스

칼만필터의 기본식은 여러가지 형태로 표현되곤 하지만 가장 일반적으로는, 다음과 같은 시스템 방정식과 관측방정식을 갖고 있는 시스템에 대한 칼만필터를 다음과 같이 표현할 수 있다.

알고리즘에 사용되는 방정식은 예측을 위한 시간 갱신 방정식(Time Update Equation)과 측정 갱신 방정식(Measurement Update Equations)으로 구성된다. 칼만 필터의 'Time Update' 과정은 시간에 앞서 미리 현재 상태를 예측하는 단계로, 순방향으로 현재 상태 추정결과를 전달한다. 'Measurement Update' 과정은 해당 시간에 실제 측정에 의해 전달된 추정 상태 값들을 조정한다. (그림 13)의 시간 갱신 방정식에서 'x'는 최적화를 하고자 하는 상태변수를 의미하고 계수 'A'는 한 단계에서의 상태변수와 다음 단계의 상태변수를 연결하는 변환계수를 표현한다. 'B'와 'u'는 한 덩어리로 인식할 수 있으며 이들은 시스템에 무관한 추가 입력값이다. 마지막으로 'w'는 'k' 단계에서 상태변수 'x'의 참값과의 차이값 또는 "시스템 오차(system error or system noise)"이다. 'w'는 개별적으로 값을 구하거나 지정할 수 없으며, 단지 오랜 관측 및 시스템의 제작시부터 알고있는 참값에 대한 표준편차로써 'Q'라는 변수로 적용된다. 관측방정식에서 'z'는 관측값이고 이는 상태변수 'x'와 변환계수 'H'에 의해 표현되며 'v'는 관측값 'z'와 관측참값과의 오차 (measurement error or measurement noise)이다. 'v'는 'w'와 마찬가지로 개개의 값을 알 수는 없고 관측 참값에 대한 분산인 'R'라는 변수로써 칼만필터 안에서 사용된다.

본 연구에서는 세 개의 LED 포인트 각각의 x, y, z축에 대한 좌표값과, 인터랙션 도구의 회전 값등 10개의 값에 대하여 각각 칼만필터를 적용한다. 이를 통해 좌표값의 현재 상태의 근사치를 유지함으로써 잡음이 최소화되도록 좌표값을 보정한다.

4.3 LED 색상 변화와 마우스 이벤트 매핑

4.3.1 LED 색상 판별

입력 원영상에서 각각의 LED 관심영역에 해당하는 픽셀들의 R, G, B 각각의 값에 대한 합산값 SRed, SGreen,

| | Red 버튼 선택시 | Green 버튼 선택시 | Blue 버튼 선택시 |
|--------------|------------|--------------|-------------|
| 입력 영상 | | | |
| LED 색상 판별 결과 | | | |

(그림 14) LED 색상 판별 결과

SBlue를 식(16), 식(17), 식(18)과 같이 구하고 이들을 비교하여 최대값을 갖는 색상이 LED의 색상으로 판별된다. (그림 14)는 LED 색상 판별결과를 Red, Green, Blue 사각형으로 표시하여 보여주고 있다.

$$SRed = \sum_{P_i \in ROI(LED)} Red(P_i) \quad (16)$$

$$SGreen = \sum_{P_i \in ROI(LED)} Green(P_i) \quad (17)$$

$$SBlue = \sum_{P_i \in ROI(LED)} Blue(P_i) \quad (18)$$

여기서, $Red(P_i), Green(P_i), Blue(P_i)$ 은 픽셀 P_i 에 대한 RGB 컬러 요소에서 각각 R, G, B 요소에 대한 값을 의미하고, $ROI(LED)$ 는 대상 LED의 ROI 영역을 의미한다.

4.3.2 이벤트 매핑

연속 영상에서 LED의 3차원 좌표값과 회전 정보가 응용 어플리케이션에게 전달된다. 이때, 3개의 LED의 색깔 조합으로 응용 어플리케이션에 이벤트를 정의하여 전송한다. 제안한 인터랙션 포인터는 세 개의 버튼을 포함하고 있다. 빨간 버튼은 세 개의 LED 모두에 빨간색의 LED를 점등시키며 3차원 마우스 포인터의 이동(Navigation) 이벤트로 연결된다. 노란 버튼은 한 개의 빨간색 LED와 두 개의 녹색 LED에 전류를 흘려보내어 빨간색과 녹색의 LED조합을 만들며 회전(Rotation) 이벤트로 연결되고, 응용 어플리케이션에게 회전 방향과 회전 각도의 정보를 전달한다. 한 개의 빨간색 LED와 두 개의 파란색 LED의 조합은 파란색 버튼을 누름으로써 만들 수 있는데, 이 버튼을 통해 응용 어플리케이션에 선택된 객체의 이동(Translation) 이벤트를 전달한다.

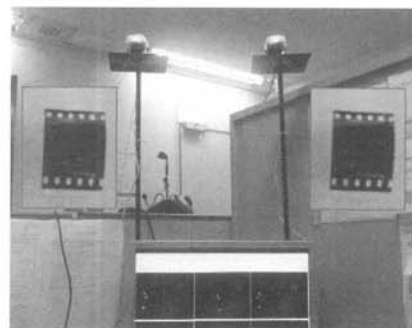
포인터 이동(Navigation) 이벤트와 객체 이동(Translation) 이벤트의 경우에는 이벤트의 종류를 나타내는 이벤트 ID와 세 개의 LED의 3차원 좌표값이 이벤트 파라미터로 전달되고,

회전(Rotation) 이벤트의 경우에는 이벤트 ID, 세 개의 LED의 3차원 좌표값, 회전 각도값이 이벤트 파라미터로 결정되어 전송된다. 응용 어플리케이션은 서로 다른 3개의 이벤트에 대해 응용 어플리케이션에 맞는 기능과 연결지어 사용할 수 있다. 3차원 공간상에서의 물체를 회전하기 위하여 일반 사용자가 복잡한 마우스의 여러 함수키를 익혀야 한다거나 회전을 위하여 3차원 스페이스 마우스의 볼의 움직임을 연습해 보는 과정 없이 제안 3차원 인터랙션 도구를 사용하는 경우, 노란색 버튼을 누른 후 인터랙션 포인터 자체를 회전함으로써 3차원 가상객체를 직관적으로 회전시킬 수 있다.


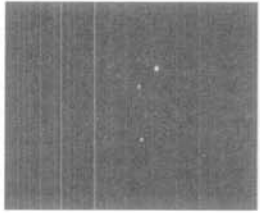


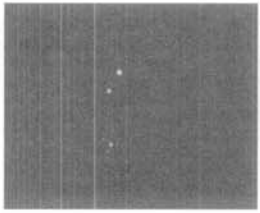

5. 실험 및 결과

5.1 카메라 설치 환경과 구현 LED 포인터

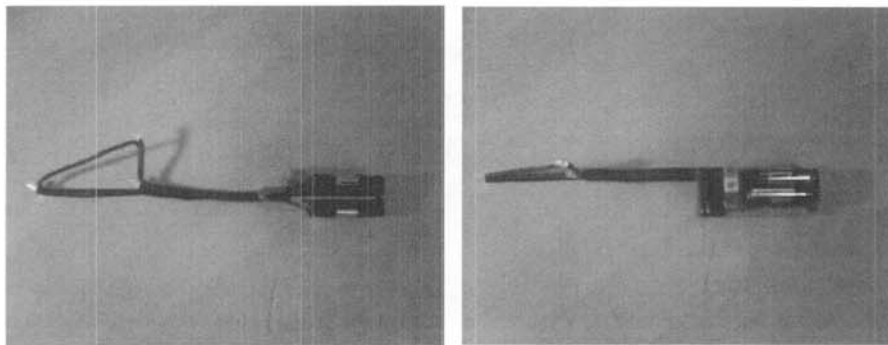
제안 시스템의 구현을 위해 USB 웹캠(Lebeca Plus) 두 대를 Pentium IV 시스템에 연결하였고, 소프트웨어 엔진은 MSVC 6.0 MFC 환경에서 개발하였다. 웹캠을 통하여 입력되는 영상에서 외부 조명의 영향을 최소화 하면서 발광하는 LED의 위치 및 색상만을 정확히 인지 할 수 있도록 하기 위하여 두 대의 웹캠에 인화된 카메라 필름을 이용한 물리적 필터를 부착하였다. 두 대의 카메라 설치 상황과 필터 부착 상황은 (그림 15)와 같다. 제안방법을 통하여 일반 조



(그림 15) 두 대의 웹캠과 웹캠 앞에 부착된 필터

| | 필터 설치전 입력 영상 | 필터 설치후 입력 영상 | LED 추적결과영상 |
|-----------------|---|--|---|
| 어두운 조명 환경 |  |  |  |
| 밝은 조명 환경 |  |  |  |

(그림 16) 서로 다른 조명 환경에서의 LED 추적 결과 비교



(a) 측면에서 본 모습 (b) 상단에서 본 모습
(그림 17) 3색 LED를 이용한 인터랙션 포인터

명 환경에서 촬영된 영상에서 LED 영역을 제외한 배경 영역은 낮은 밝기값으로 제거할 수 있었다. 이는 특수하게 제작된 적외선 소스와 적외선 카메라의 사용 없이 일반 조명 환경에서 범용 웹캠을 이용하여 LED의 위치를 안정적으로 추적할 수 있도록 한 것이다. (그림 16)은 카메라의 입력 밝기 파라미터 값을 고정시킨 상황에서 보조 조명을 이용하여 실내 조명 조건을 다르게 한 경우에도 LED의 위치를 안정적으로 추적한 결과를 보여주고 있다.

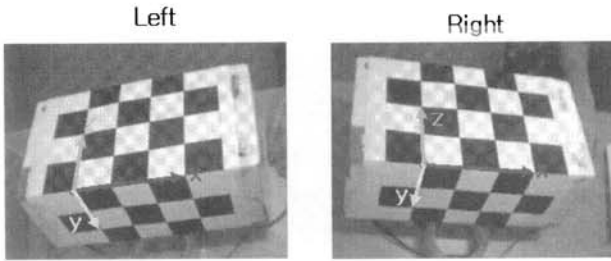
(그림 17)은 3.2절에서 설명한 바와 같이 세 개의 3색 LED를 이용하여 제작된 LED 기반 인터랙션 포인터를 보여주고 있다.

5.2 3차원 좌표와 LED 정보 추출 오차

캘리브레이션 박스에 5X5(cm)의 그리드로 좌·우영상을 입력받아 캘리브레이션을 수행한다. (그림 18)은 필름이 부착되지 않은 상태의 좌·우 카메라로부터 받아들인 스테레오 영상에서 캘리브레이션 박스의 원점과 x, y, z 각 축에 해당하는 3차원 좌표와 그에 대응하는 이차원 프레임을 보여주고 있다.

캘리브레이션 후 얻어진 3차원 좌표 변환행렬의 정확성 및 정밀도를 나타내기 위한 방법으로 이차원 좌·우영상에서의 그리드 캘리브레이션 상자위의 그리드에서의 격자점 값을 행렬에 적용하여 얻은 3차원 좌표값과, 실제 공간에서 미리 정의한 3차원 좌표값의 차이의 평균 및 표준편차를 구하여 캘리브레이션 오차를 계산하면 <표 3>과 같다. <표 3>에 보면, x, y, z 축 방향 모두 평균 1mm 오차 이내로 3차원 좌표값이 계산됨을 알 수 있다.

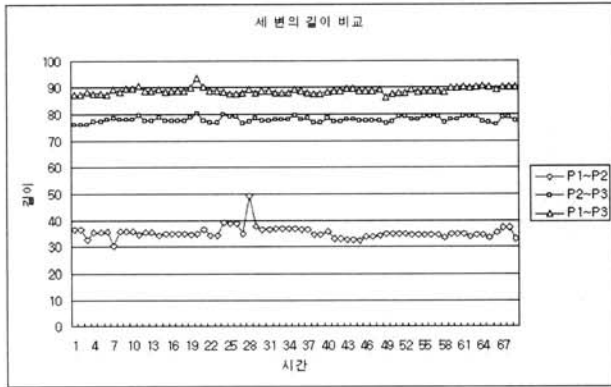
LED ID 및 색상 판정에 대한 오류율을 측정하기 위하여, 2인의 실험자에게 사용 방법에 대한 간단한 설명 후 제작된 포인터를 사용하도록 하였다. 세 개의 LED ID 판정의 정확성을 검증하기 위한 방법으로 물리적 인터랙션 도구의 인터랙션 볼을 꼭지점으로 하는 직각 삼각형을 만들었을 때 길이의 비를 이용하는 방법을 사용하였다. (그림 19)의 그래프에서 볼 수 있듯이, 세 번의 길이는 약 3.5cm, 7.5cm, 8.5cm 정도의 비율을 가지고 있으므로 세 점의 ID 판정은 정확한 것으로 판정되었다. 또한 각각의 버튼 이벤트를 검증하기 위해서 버튼을 눌렀을 때의 색상인식의 정확성을 판별할 수



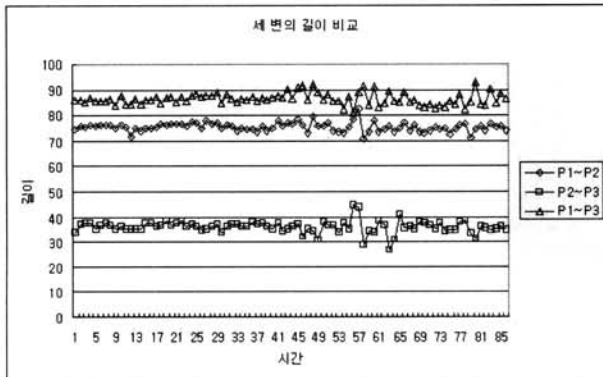
(그림 18) 캘리브레이션을 위한 좌·우 영상

<표 3> 3차원 좌표 오차

| (mm) | x | y | z |
|------|---------|--------|-------------|
| 평균 | 0.4583 | 0.9583 | 1.1667 |
| 분산 | 0.3316 | 0.8733 | 0.8889 |
| 표준편차 | 0.57584 | 0.9345 | 0.942809042 |



(a) 실험자 A의 포인터 거리 결과에 따른 포인터 인식 결과



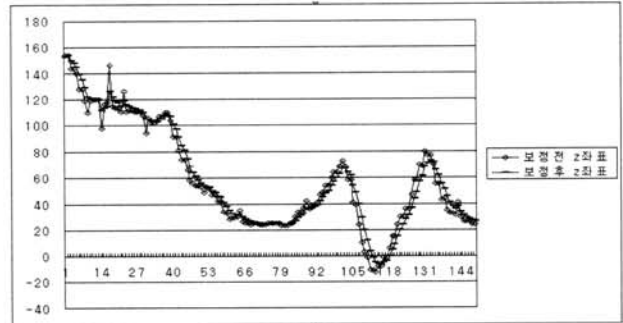
(b) 실험자 B의 포인터 거리 결과에 따른 포인터 인식 결과
(그림 19) 서로 다른 실험자에 따른 포인터 인식 결과

있는데, <표 4>에서 보는 바와 같이 빨간 버튼의 경우 총 1194개의 획득영상으로부터 색상이 잘못 인식된 경우가 없는 0.0%의 오류율을 나타내었고, 노란버튼은 2304개의 획득영상으로부터 15개의 오류로 0.65%의 오류율을 나타내었다. 파란버튼은 1866개의 획득영상 모두에서 이벤트인식이 정확하게 되어서 오류율이 0%로 나타났다.

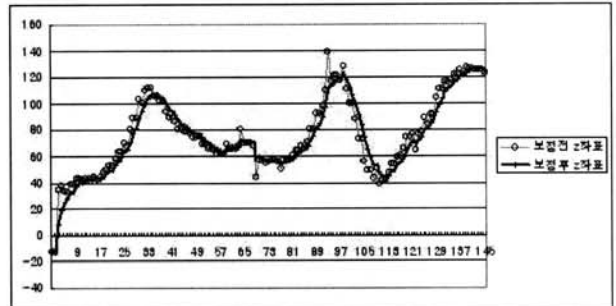
(그림 20)과 (그림 21)은 좌우의 스테레오 영상으로부터

<표 4> 버튼의 색상 인식 결과

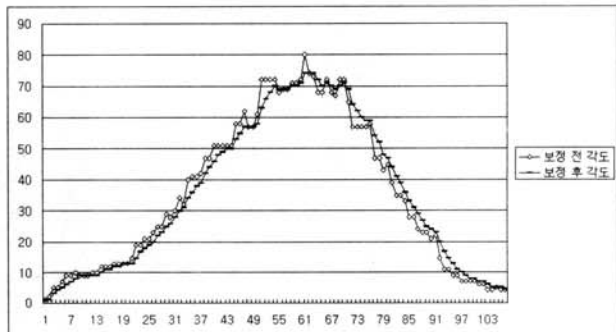
| | 버튼1(빨강) | 버튼2(노랑) | 버튼3(파랑) |
|-------------|---------|---------|---------|
| 총 색상 인식 | 1194 | 2304 | 1866 |
| 잘못 인식된 색상 수 | 0 | 15 | 0 |
| 오류비율 | 0 | 0.00651 | 0 |



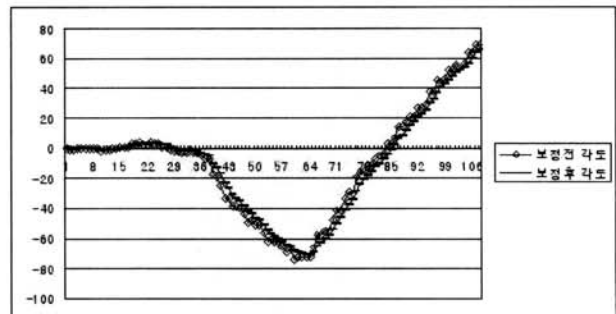
(a) 실험자 A의 칼만 필터 적용 전 후



(b) 실험자 B의 칼만 필터 적용 전 후
(그림 20) 칼만 필터 적용 전·후의 z축 3차원 좌표 비교



(a) 실험자 A의 칼만 필터 적용 전 후



(b) 실험자 B의 칼만 필터 적용 전 후
(그림 21) 칼만 필터 적용 전·후의 회전각 비교

추정된 3차원 좌표값 및 회전각을 칼만 필터로 보정한 후의 좌표값과 회전각과 비교한 것이다. 추정된 원 자료에서의 잡음이 칼만필터를 적용함으로써 매끄럽게 보정된 것을 확인할 수 있다.

제안 방법의 수행 효율성을 검증하기 위하여 한 시점에서 좌우카메라 입력 영상에 대한 주요 처리 모듈의 수행 시간을 측정하였다. 정확한 측정을 위하여 50 프레임에 해당하는 영상에 대한 수행시간을 측정한 후 이에 대한 각 모듈별 평균시간을 계산하였다. <표 5>에서 보는 바와 같이 카메라 영상으로부터 입력받은 정보로 포인트를 추적하는 시간은 0.00457초였으며 추적된 포인트로 3차원 좌표를 복원하는 시간은 0.0000039초의 시간이 소요되었다. 칼만필터와 외삽법등을 이용한 포

<표 5> 실험 수행 시간 (초)

| | 포인트 추적 | 3차원 좌표 계산 | 이벤트정의 시간 | 총 합 |
|----|-------------|---------------|------------|----------|
| 평균 | 0.004571044 | 0.00000389788 | 0.00094585 | 0.005521 |

인터 이동등의 이벤트 정의시간은 0.000946초의 시간이 소요되어 한 프레임당 총 0.0055초 정도의 시간이 소요되었다. 이는 초당 180프레임 이상의 처리가 가능한 성능이다.

(그림 22)은 좌우 카메라 영상 이미지에서 각 LED의 3차원적 좌표를 추출한 그림과 3차원적 정보를 응용 어플리케이션에 전송하여 3차원 그래픽 공간에 인터랙션 도구의 LED가 이루는 삼각형을 표시한 결과 영상을 보여주고 있다. 사용자는 인터랙션 도구의 위치를 데스크탑의 모니터

| | | 원영상 | 색상/좌표인식 | 3차원 공간에 표현된 인터랙션 도구 삼각형 |
|--------|---|-----|---------|-------------------------|
| 버튼 I | 좌 | | | |
| | 우 | | | |
| 버튼 II | 좌 | | | |
| | 우 | | | |
| 버튼 III | 좌 | | | |
| | 우 | | | |

(그림 22) 2차원 이미지와 3차원 그래픽 공간에 표시된 인터랙션 도구

앞 공간에서 자유롭게 이동 회전시킴으로써 3차원 가상공간상의 객체를 쉽게 조작할 수 있다.

6. 결 론

본 논문에서는 고정된 카메라에서 얻어진 영상을 사용하여 데스크탑 기반 환경에서의 3차원 상호작용을 위한 저가형 인터랙션 도구를 제안하였다. 본 논문에서는 3색 LED를 기반한 인터랙션 포인터의 제작 내용과 LED들의 위치와 색상을 추적하여 3차원 마우스에 필요한 기능 이벤트를 추출하기 위한 기술을 논하고 제안 기술의 오차 관찰에 대한 실험결과를 보임으로써 제안 도구의 정확성을 입증하였다. 제안 방법에서는 일정 밝기를 유지하며 발광할 수 있도록 고안된 3색 LED를 이용하여 인터랙션 포인터를 제작하고, LED 외의 외부 빛을 차단하기 위한 필터를 부착한 웹캠을 사용하여 LED의 움직임 영상을 입력받음으로써, 외부 조명에 영향을 받지 않고 안정적으로 LED의 위치를 추적할 수 있도록 하였다. 또한, 인터랙션 포인터를 서로 길이가 다른 변을 가지는 삼각형의 형태로 제작함으로써, LED의 개별 ID 관별이 용이하도록 하였다.

기존 3차원 스페이스 마우스는 3차원 공간상의 객체를 회전이나 이동하기 위하여 3차원 마우스 볼을 이용하는 방법에 대한 학습이 요구되었으나, 제안 도구는 객체를 회전하거나 이동하기 위하여 도구 자체를 회전, 이동하는 직관적인 방법을 사용하고 있기 때문에 일반 사용자가 보다 쉽게 도구를 이용할 수 있도록 하였다. 또한, 3차원 가상공간에서 가상 객체의 조작을 위한 LED 색상을 이용한 서로 다른 이벤트를 지원하여 응용 어플리케이션이 다양한 기능과 연결지어 사용할 수 있도록 하였다. 본 논문에서 제안하는 가상 객체에 대한 실시간 조작기술은 게임, 교육 콘텐츠, 시뮬레이션, 디지털 팩토리 등 폭넓은 응용분야에 적용가능할 것으로 기대하고 있다.

참 고 문 헌

- [1] C. Graetzel, T. Fong, S. Grange and C. Baur, "A non-contact mouse for surgeon-computer interaction," *Technology and Health Care*, Vol.12, No.3, pp.245-257, 2004.
- [2] F.S. Chen, C.M. Fu and C.L. Huang, "Hand gesture recognition using a real-time tracking method and Hidden Markov Models," *Image and Vision Computing*, Vol.21, No.8, pp.745-758, 1 August 2003.
- [3] H.S. Park, E.Y.Kim, S.S. Jang and H.J. Kim, "An HMM Based Gesture Recognition for Perceptual User Interface," *Pacific-Rim Conference on Multimedia (PCM2004)*, LNCS. Vol.3332, pp.1027-1034, 2004.
- [4] S. Li, W. Hsu and H. Pung, "A Real-time Monocular Vision-based 3D Mouse System," *International Conference on Computer Design*, LNCS. Vol.1296, pp.448-455, 1997.
- [5] 한영환, "움직임 정보를 이용한 제스처 인식 시스템", *한국정보처리학회 논문지 B*, Vol.10, No.4, pp.473-478, June 2003.
- [6] Y. Hamada, N. Shimada and Y. Shirai. "Hand shape estimation using sequence of multi-ocular images based on transition network", *In 15th International Conference on Vision Interface*, pp.362-368, May 2002.
- [7] Y.J. Choi, S.M. Choi, S.M. Rhee and M.H. Kim, "Collaborative and Immersive Medical Education in a Virtual Workbench Environment," *Ninth International Conference on Knowledge-based Intelligent Information & Engineering Systems*, LNAI 3683, pp.1210-1217, 2005.
- [8] F. Madritsche and M. Gervautz, "CCD-Camera Based Optical Beacon Tracking for Virtual and Augmented Reality", *Eurographics '96*, Vol.15, No.3, pp.207-215, 1996.
- [9] 김효선, "프로젝션 월 환경에서의 카메라를 이용한 실시간 헤드 트래킹", *이화여자대학교 과학기술대학원 석사학위 청구논문*, 2002.
- [10] 이선민, "스테레오 비디오 기반 상호작용형 사용자 미러링", *이화여자대학교 일반대학원 박사학위 청구논문*, 2007
- [11] Roger Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, pp.364-374, 1986.
- [12] G. Taubin. "Camera model and triangulation", *Note for EE-148 : 3D Photography*, CalTech, 2001.
- [13] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *Notes of ACM SIGGRAPH tutorial on the Kalman filter*, 2001.



최 유 주

e-mail : yjchoi@suv.ac.kr

1989년 이화여자대학교 전자계산학과(이학사)

1991년 이화여자대학교 전자계산학과(이학석사)

2005년 이화여자대학교 컴퓨터학과(공학박사)

1991년~1993년 한국컴퓨터주식회사 기술연구소 주임연구원

1994년~1999년 포스데이타주식회사 기술연구소 주임연구원

2005년~현재 서울벤처정보대학원대학교 컴퓨터응용기술학과 전임강사

관심분야 : 컴퓨터 그래픽스, 가상현실, HCI, 컴퓨터비전, 의료영상처리 등



이 선 민

e-mail : seonminrhee@hotmail.com

1999년 이화여자대학교 컴퓨터학과 졸업
(공학학사)

2001년 이화여자대학교 컴퓨터학과 졸업
(공학석사)

2007년 이화여자대학교 컴퓨터학과 졸업
(공학박사)

2007년 이화여자대학교 컴퓨터그래픽스/가상현실 연구센터, 박사후연구원

2008년~현 재 University of Southern California, Postdoctoral Researcher

관심분야 : 가상현실, 컴퓨터 그래픽스, 컴퓨터 비전, HCI 등



노 영 섭

e-mail : ysroh@suv.ac.kr

1988년 인하대학교 전자공학과(공학사)

1996년 한국과학기술원 정보및통신공학과
(공학석사)

2005년 고려대학교 전기.전자.전파공학과(공학박사)

1987년~1998년 LG전자 미디어통신연구소 선임연구원

1998년~2001년 청강문화산업대학교 이동통신과 교수

2001년~2005년 주식회사 싸이버뱅크 연구개발부문 상무이사

2005년~현 재 서울벤처정보대학원대학교 임베디드시스템학과 교수

관심분야 : 임베디드시스템, 모바일 컴퓨팅, 이동통신, 유비쿼터스 네트워크



유 효 선

e-mail : babydigital@hanmail.net

2002년 이화여자대학교 컴퓨터학과(학사)

2004년 이화여자대학교 컴퓨터학과(석사)

2007년~현 재 서울벤처정보대학원대학교
VCAR 연구실 연구원

관심분야 : 컴퓨터비전, 가상현실 등