

협업 환경을 위한 LED 기반 사용자 위치 추적 및 인터랙션 시스템

User Location Determination and Interaction for Collaborative Environments using LED Tracking

김 인태, In-Tae Kim*, 박준, Jun Park**

*서울대학교 컴퓨터공학부, **홍익대학교 컴퓨터공학과

요약 다수의 사용자가 동일한 디스플레이 화면을 보면서 의사를 교환하거나 공동 작업을 수행하는 협업환경에서의 인터랙션 기술은 네트워크 기술, 디스플레이 기술만큼이나 효율적인 협업을 위해 중요한 요소 기술이다. 이러한 인터랙션의 성공적인 수행을 위해서 중요한 기능 중의 하나는 사용자의 상황인식을 들 수 있는데 이는 사용자의 돌발 행동, 사용자들간의 의사교환 정도 등을 측정하고 예측하는데 유용하게 사용될 수 있다. 협업환경 내에서의 사용자의 위치 파악은 사용자의 상황인식뿐 아니라 다수의 사용자가 동시에 레이저 포인터를 이용하여 디스플레이 장치를 가리키고 있는 경우, 어느 사용자가 디스플레이 화면 중 어느 위치를 포인팅 하고 있는지를 계산하는데 도움을 줄 수 있다. 본 연구에서는 제한된 환경 내에서 사용자의 3 차원 위치를 추적할 수 있는 시스템을 제안한다. 개발된 시스템은 사용자의 위치 계산뿐 아니라 사용자의 제스처도 함께 인식하여 사용자가 협업환경 시스템을 제어할 수 있고 공중에서 글씨 쓰는 등의 인터랙션도 수행할 수는 기능을 지원한다.

핵심어: 협업환경, 3차원 위치 계산, 사용자 상황인식, LED

1. 서론

동일한 공간 혹은 원거리 공간에 위치한 사용자들간의 효율적인 의견 교환이나 공동 작업을 위해서 다수의 컴퓨팅 장치, 디스플레이 장치, 인터랙션 장치들로 구성된 협업 환경이 자주 사용된다. 다수의 사용자가 참여하는, 특히 여러 명의 원격 사용자들이 참여하는, 협업환경에서는 사용자의 상황인식은 유비쿼터스 기반의 스마트 협업 환경에서 중요한 문제 중의 하나이다. 인식된 사용자의 상황은 사용자의 참여도, 사용자의 돌발 행동, 사용자들간의 의사교환 정도 등을 측정하고 예측하는데 유용하게 사용될 수 있다. 특히, 각각의 사용자의 위치 파악은 사용자가 현재 어떠한 양태(예: 발표 중 인지, 타 사용자와 논쟁 중 인지, 협업과 무관한 작업을 수행 중 인지 등)로 참여하고 있는지 등을 인식하는데 도움을 줄 수 있다. 뿐 아니라, 사용자의 위치는 다수의 사용자가 동시에 레이저 포인터를 이용하여 디스플레이 장치를 가리키고 인터랙션을 수행하고 있는 경우, 어느 사용자가 디스플레이 화면 중 어느 위치를 포인팅 하고 있는지를 계산하는 데에도 도움을 줄 수 있다.

본 연구에서는 사용자의 프레젠테이션 용 마우스에 LED를 장착하고 그 LED의 위치 추적을 통해 사용자의 위치를 계산하는 시스템을 소개하고자 한다. 개발된 시스템을 사용하면 사용자의 위치 계산 뿐 아니라 사용자의 제스처도 함께

인식하여 사용자가 협업환경 시스템을 제어할 수 있고 공중에서 글씨 쓰는 등의 인터랙션도 수행할 수 있다.

2. 사용자 위치 추적 시스템

2.1 기존의 사용자의 위치 추적 시스템

협업 환경에서 사용자의 위치를 추적하기 위한 방법은 여러 가지가 있으나 그 중에서 보편적으로 수용 가능한 방법은 RFID(혹은 RF)를 이용한 방법이다. 이는 사용자가 RFID tag를 가지고 있고 환경 곳곳에 RFID reader기를 설치하여 근접 정도를 센싱 한 결과를 사용하여 2차원 평면 상에서의 사용자의 위치를 측정하는 방법이다. 이 방법을 사용하는 경우, 사용자는 크기도 작고 저렴한 RFID tag를 휴대하기만 하면 되지만 RFID reader기가 고가이고 그 정밀도 또한 (설정에 따라 차이가 있으나) 0.5m정도 밖에 되지 않아 활용성이 제한적이다. 이 밖에도 3차원 트래킹에 사용되는 장비들(초음파 트래커, 자기장 트래커 등)을 사용할 수도 있으나 이 또한 고가이고 거추장스러운 케이블이 연결되어 있는 것이 대부분이다.

이에 반해 컴퓨터 비전 기반의 위치 추적 시스템은 정확도가 상대적으로 높고 저렴하다는 장점이 있다. 그러나 대체

로 어둡고 조명 환경이 수시로 변하는 디스플레이 장치 앞에서의 위치 추적은 견고성이 많이 낮아져 실용적이지 못한 경우가 많다.

2.2 LED 기반 사용자 위치 추적 시스템

본 연구에서는 저렴하고 정밀도도 높은 비전 기반의 접근 방법을 사용하면서 조명환경에 영향을 거의 받지 않는 LED를 매체로 사용한다. 제안하는 시스템에서는 협업 환경에서 주요 사용자들에게 일종의 포인팅 장비(uT-pointer)를 휴대하게 하고 이 장비를 사용하여 디스플레이 장치를 포인팅하거나 관련된 인터랙션을 수행하게 하는 환경을 목표로 한다. 본 연구와 관련된 스마트 미팅 스페이스는 그림 1에 도시된 바와 같다.

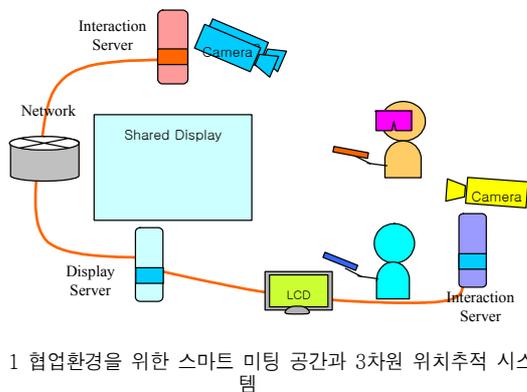


그림 1 협업환경을 위한 스마트 미팅 공간과 3차원 위치추적 시스템

본 논문에서 제안하는 사용자의 위치 추적 시스템은 사용자가 휴대하는 uT-pointer에 장착된 LED를 기반으로 한다. 협업환경 내에 설치된 calibration된 두 대 이상의 카메라가 LED를 인식하고 추적하게 하게 되면 삼각 측량법에 의해서 uT-pointer의 3차원 위치를 계산할 수 있게 된다. 조명 환경이 일정하지 않은 협업 환경 내에서의 LED 위치 인식을 위해서 본 연구에서는 가시광선 대신 적외선을 사용하며 카메라에 적외선 투과 필터를 장착하여 노이즈 및 아웃라이어 발생의 빈도를 줄였다.

추적되는 사용자의 uT-pointer의 위치는 사용자가 인터랙션을 수행하거나 허공에 글씨를 쓸 수 있는 기능을 지원하여 desktop-free 인터페이스를 제공한다. 이러한 인터페이스는 마우스의 대부분의 기능과 키보드의 일부 기능을 제공하여 사용자들의 일반 업무에 필요한 대부분의 기능을 사용할 수 있게 해 준다. 사용자 간의 구분은 LED의 on/off 패턴 변화를 측정하여 수행할 수 있다. 또한, 계산된 사용자의 위치 정확도는 폭이 10m인 협업환경에서 10cm이내가 되어 RFID등의 기술보다 높은 정확도를 가진다.

더 나아가서는 하나의 LED 대신에 두 개의 LED를 사용하는 경우 사용자의 위치뿐 아니라 uT-pointer가 가리키는 방향 까지도 측정이 가능하다. 이러한 방향 데이터는 그 정밀도와 정확도가 낮은 편이지만, 사용자의 레이저 포인팅 위치 추적을 함께 사용할 경우, 두 명 이상의 사용자가 동시에 디스플레이를 가리키고 있는 경우에도 어느 사용자가 어느 위치를 가리키고 있는지 구분도 어느 정도 가능하다. 이 정보는 여러 명의 사용자가 동시에 디스플레이 상에서 작업하

는 경우 레이저포인터를 통한 정보의 수정, 정보의 흐름 제어, 데이터 및 태스크 마이그레이션 등에 유용하게 사용될 수 있다.

3. 배경 기술

3.1. Tsai's Camera Calibration Method

3차원 공간 속에서의 적외선 LED의 위치를 추적하기 위해서 두 대의 카메라의 상대적 위치를 미리 계산하는 calibration 작업을 수행하게 된다. 오프라인으로 카메라의 파라미터를 구하는 (카메라의 위치와 방향 계산(pose estimation) 을 포함) 방법 중 가장 널리 사용되는 것이 Tsai의 'Camera Calibration Method'이다.

본 연구에서는 두 대의 카메라 사이의 상대적 위치 관계를 구하기 위해서, Marker를 이용하여 Marker와 카메라1 사이의 위치 관계와, Marker와 카메라2 사이의 위치관계를 각각 구한 후 이들 결과 값을 사용하였다.

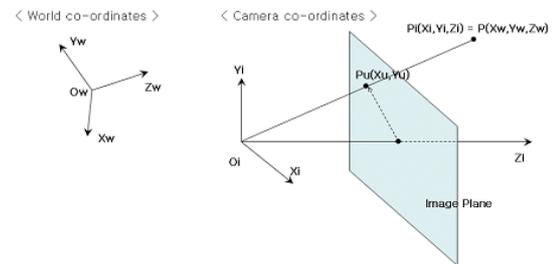


그림 2 Calibration을 위한 좌표계와

마커가 위치한 곳을 중심 좌표계라고 가정하면, 마커 (World coordinate) 와 카메라(Camera coordinates) 사이의 상대적 위치는 Euler Angles (θ, ϕ, ψ) 과 Translation Parameters (T_x, T_y, T_z) 로 표시될 수 있다. 어떤 한 점(P) 에 대한 Real World에서의 좌표를 (X_w, Y_w, Z_w), Camera co-ordinates에서의 좌표를 (X_i, Y_i, Z_i) 라고 했을 때, 이 두 좌표 사이에는 다음의 수식과 같은 관계가 성립한다.

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad \text{with}$$

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\cos\phi - \sin\psi\cos\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

따라서 Calibration을 두대의 카메라에 대해 수행하게 되면 마커와 카메라1 사이의 파라미터와 마커와 카메라2 사이의 파라미터를 구할 수 있고 이들을 이용하여 카메라1과 카메라 2 사이의 위치 및 방향 정보도 구할 수 있다.

3.2. 3차원 공간에서의 위치

두 카메라 사이의 위치 관계와 두 대의 카메라의 image

plane에서 각각의 좌표, 그리고 카메라의 기본적인 parameter를 이용하여 3D 공간에서의 위치 $P_w(X_w, Y_w, Z_w)$ 를 구할 수 있다. 단순히 생각하기 위해 두 카메라 사이의 Euler Angles (θ, ϕ, ψ)를 $(0, 0, 0)$ 이라고 하고, 두 카메라 사이의 Translation Parameters (T_x, T_y, T_z)를 $(D_x, 0, 0)$ 이라고 가정한다. 마지막으로, 카메라의 focal length를 f_l 이라고 할 때, 다음과 같은 식이 성립한다.

$$X_w = \frac{D_x(k-j)}{2(D_x - (k+j))}, Z_w = \frac{f_l(j+k)}{D_x - (j+k)}, Y_w = \frac{Z_w(m - T_y) + T_y}{f_l}$$

여기에서 j, k, m 값은 카메라의 image plane에서의 점의 좌표와 두 카메라 사이의 상대적 위치로부터 구해지며, D_x 는 두 카메라의 x축 상에서의 중심좌표 간의 거리이다. 이 과정을 그림으로 도시하면 그림 3에 나타난 바와 같다.

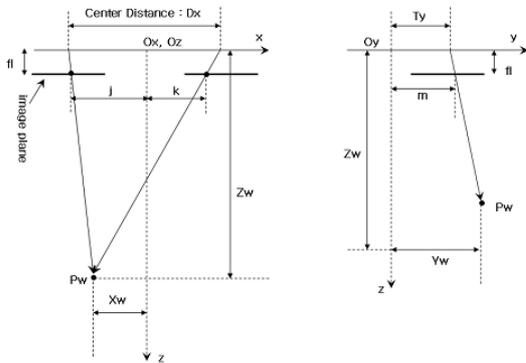


그림 3 Simplified Plot for P_w

4. 구현 및 실험

4.1. 시스템 구성

본 논문에서 구현한 두 대의 카메라를 이용하여 3차원 공간 속에서 LED의 위치를 구하는 전체 System은 크게 세 개의 부분으로 구성되어 있는데, 이를 살펴보면 그림 4와 같다. ①, ②, ③으로 묶어 놓은 것이 각각의 모듈이며, 대표적인 역할과 구현에 쓰인 Library 등에 대해 간략히 나열하였다. 화살표는 data의 흐름을 나타낸다.

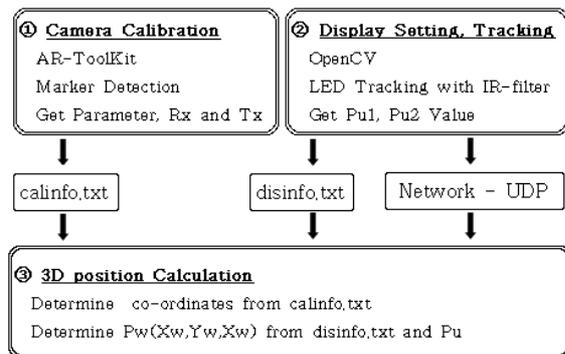


그림 4 System의 전체적인 구성

첫 번째 모듈은 Marker Detecting 방식을 이용하여 카메라

1과 Marker, 카메라2와 Marker 사이의 위치 관계를 구하는 역할을 한다. 두 대의 카메라와 Marker 사이의 위치 관계를 구하는 동안 Marker는 고정되어 있으며, Marker를 인식하면 그 위치 관계 data가 calinfo.txt라는 파일에 저장되게 된다. 구현에는 AR-ToolKit Library를 사용하였다.

첫 번째 단계가 끝난 후, 두 번째 모듈에서는 적외선 필터를 켜고 카메라를 이용하여 적외선 LED의 위치를 tracking하게 된다. LED의 위치를 tracking한 결과 값은 카메라의 image plane 상에서 x, y의 pixel 값으로 표시되는데, 이 pixel 값을 구하기 위해서 모니터의 해상도와 LED의 위치를 추적할 모니터 상에서의 영역을 입력 받아서 사용하게 된다. 또한 나중에 pixel 값을 거리 값(millimeter 단위)으로 변환하기 위해서 모니터의 크기와 가로 세로 비율 또한 입력 받게 된다. 모니터의 해상도와 크기, 그리고 가로와 세로 비율 값들은 disinfo.txt라는 파일에 저장되게 된다.

두 번째 모듈은 또한 모니터의 해상도와 추적 영역의 정보를 이용하여 카메라 image plane에서의 LED의 x, y pixel 값을 구하고 이 값을 네트워크를 통하여 세 번째 모듈로 보내주는 역할도 하게 된다. 구현에는 OpenCV Library를 사용하였다.

세 번째 모듈은 첫 번째 모듈로부터 카메라와 real world의 위치 관계(calinfo.txt)를, 두 번째 모듈로부터 모니터의 정보(disinfo.txt)와 LED의 위치값($P_w(X_w, Y_w, Z_w)$)을 넘겨받아서 Real World에서 LED의 위치값($P_w(X_w, Y_w, Z_w)$)을 구하는 역할을 한다. 또한 선택적으로 마우스 인터페이스를 선택한 경우에는 마우스의 좌표를 옮기는 기능을 제공한다.

본 연구와 관련한 스마트 미팅 공간에서는 다양한 종류의 인터랙션 도구들이 Shared Display를 이용하는데, 이 디스플레이를 제어하는 컴퓨터와 각각의 인터랙션 도구들을 제어하는 컴퓨터가 따로따로 존재하기 때문에, 필수적으로 network를 통해 pixel 값을 전달해야 한다.

4.2. Display Setting & Tracking

LED Tracking System은 Calibration System에 Infra-Red Filter를 추가하고, marker를 제외하는 대신 적외선 LED 장치를 추가하여 구성된다. 적외선 LED 장치는, 일반적으로 쓰이는 마우스 인터페이스를 가진 presentation용 pointer에서 붉은 레이저 대신 그 위치에 적외선 LED를 장착하여 만들었다. Tracking System의 전체적인 구성은 그림 9와 같다.

그림 5 Tracking System 적외선 LED를 tracking하고 카메라 image plane에서의 x, y 좌표를 얻어서 모니터의 pixel 값을 구하는 프로그램은 openCV Library를 이용하여 구현하였다. Tracking System은 3장에서 살펴본듯이 Display 관련 값들을 설정하는 부분과 적외선 LED를 Tracking하는 부분으로 나누어진다.

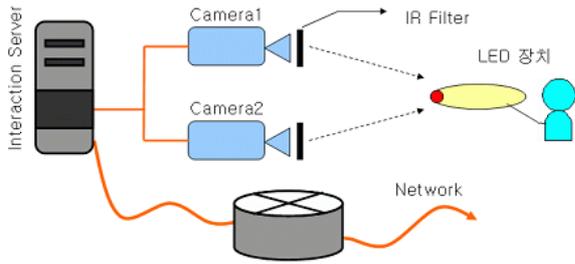


그림 5 Tracking System

우선 Display 설정부터 살펴보면 Shared Display에서의 마우스의 좌표를 결정하기 위해 가로와 세로의 pixel 수를 입력받게 된다. 본 연구에 사용된 Shared Display는 1600x1200의 해상도를 갖는 모니터 6x4개를 이용하여 9600x4800의 해상도를 갖는 tiled display를 목표로 구성되었다.

Display 설정에서는 또한 마우스 좌표의 영역을 입력 받는다. 공간에서의 위치 계산에는 카메라의 모든 영역이 사용되지만, 마우스 인터페이스를 이용할 때에는 사용자의 편의를 위해 이렇게 입력 받은 값을 마우스의 좌표를 계산할 때 사용하게 된다. Display 설정이 끝나면(혹은 이미 설정되어 있는 값을 변경할 필요가 없을 경우에는) LED를 Tracking 하게 된다.

WM_FRAME_MSG 메시지가 발생하면 callback 함수인 LEDTracking_Callback() 함수가 실행된다. 이 함수는 카메라 image에서 가장 밝은 부분의 중심 좌표를 찾는 역할을 하는데, 이 좌표가 LED의 위치이다. 구하고 나면, Xform2Scr() 함수를 호출한다. 이 함수는 카메라 image 상의 좌표 값을 모니터의 pixel좌표로 변환하고, 또 다른 함수인 action()함수를 호출한다. 호출된 action() 함수는 UDP Protocol을 이용하여 모니터의 pixel 값을 다음 모듈로 보내주는 역할을 한다. 실행되는 모습은 아래 그림 6, 그림 7과 같다.



그림 6 LED Tracking

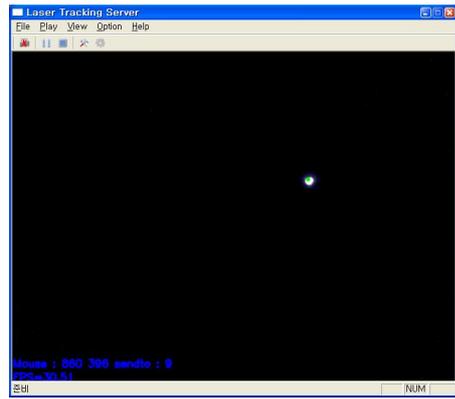


그림 7 해상도 1280x1024로 세팅된 상태에서의 monitor pixel 값 (860, 396)

4.3. Calculating the Position PW

카메라 사이의 상대적 위치가 결정되고, 두 카메라의 image plane에서의 LED의 위치가 결정되면 공간상에서 LED의 3차원 좌표값을 계산할 수 있다.

세 번째 모듈은 3장에서 살펴본 바와 같이 앞서 두 모듈로부터 얻은 정보를 이용하여 계산을 수행하여 LED의 공간 좌표를 결정하고, 아래 그림과 같은 shared display장치에서 LED를 통한 마우스 인터페이스를 사용할 경우에는 LED의 pixel 좌표를 추가로 계산하게 된다.

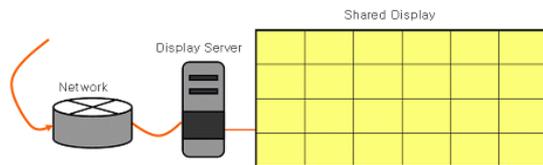


그림 8 Shared Display

세 번째 모듈은 MFC Dialog 기반으로 구현 되었으며, 프로그램은 UDP 통신을 위한 초기화를 한 후, 두 카메라에서의 위치를 통해 값을 계산한다. 이때에 두 대의 카메라의 중심점을 원점으로 하기 위해 카메라의 평행이동 값을 통해 보정을 하게 된다.

5. 결론

본 논문에서는 조명환경이 일정하지 않은 협업 환경을 위한 LED를 기반 사용자 위치 추적 시스템을 제안하였다. 본 시스템에서 제안하는 방법은 다수의 calibration된 카메라를 사용하여 사용자에게 부착된 (어깨, 혹은 사용하는 포인터) 적외선 LED의 3차원 위치를 계산하고 추적하는 방법이다.

추적된 위치는 사용자의 상황인식에 유용하게 사용될 뿐 아니라, 제스처 기반의 커맨드 실행, 공간 상에서의 글씨쓰기 등의 기능을 지원하는데 사용된다. 이는 마우스나 키보드 같은 데스크탑 용 인터페이스의 사용이 어려운 협업환경에서 유용하게 사용될 것으로 기대된다.

또한, 두 개 이상의 LED를 사용하는 경우 사용자가 가리키는 포인팅 방향의 측정도 가능하다. 이러한 방향 데이터는 그 정밀도와 정확도가 낮은 편이지만, 사용자의 레이저 포인팅 위치 추적을 함께 사용할 경우, 두 명 이상의 사용자가 동시에 디스플레이를 가리키고 있는 경우에도 어느 사용자가 어느 위치를 가리키고 있는지 구분도 어느 정도 가능하다.

Acknowledgements

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발사업의 지원에 의한 것임

참고문헌

- [1] Grant Wallace, Otto J. Anshus, Peng Bi, Han Chen, Yuqun Chen, Douglas Clark, Perry Cook, Adam Finkelstein, Thomas Funkhouser, Anoop Gupta, Matthew Hibbs, Kai Li, Zhiyan Liu, Rudrajit Samanta, Rahul Sukthankar, and Olga Troyanskaya. "Tools and Applications for Large-Scale Display Walls," IEEE Computer Graphics and Applications, July 2005.
- [2] Benko, H., Ishak, E.W., Feiner, S. "Collaborative Mixed Reality Visualization of an Archaeological Excavation." The International Symposium on Mixed and Augmented Reality (ISMAR 2004), November 2004.
- [3] J. Rekimoto and M. Saitoh, "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments," Proc. ACM CHI '99, Pittsburgh, PA, May 15-20, 1999, pp. 378-85.
- [4] D.H. Wilson and A. Wilson, "Gesture Recognition Using the XWand", Technical Report CMU-RI-TR-04-57, Robotics Institute, Carnegie Mellon University, April 2004
- [5] Daw-Tung Lin, "Spatio-temporal hand gesture recognition using neural networks," Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence.
- [6] T. Darrell and A. Pentland, "Space-Time Gestures. Proceedings, "IEEE Conference on Computer Vision and Pattern Recognition., 1993, pp. 335-340.
- [7] X.D. Huang, Y. Ariki, and M.A. Jack, "Hidden Markov Models for Speech Recognition." Edinburgh: Edinburgh Univ. Press, 1990.
- [8] Roger Y. Tsai, "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pp. 323-344
- [9] R. Azuma, "A Survey of Augmented Reality", Presence: Teleoperators and Virtual Environments 6, 4, pp.355~385, 1997
- [10] H. Ishii, B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", in Proceedings of ACM CHI '97, pp. 234-241, Mar. 1997
- [12] I. Poupyrev, "Augmented Groove : Collaborative Jamming in Augmented Reality", ACM SIGGRAPH 2000 Conference Abstracts and Applications, pp. 77, 2000
- [14] Y.Lee, Sj.Oh, Y.Suh and W.Woo, "Developing vr-UCAM for Interaction on Distributed Virtual Environments", KHCI2005, pp. 507-512, 2005
- [15] G. Welch and G. Bishop, "An introduction to the kalman filter". CS TR 95-041, UNC, 1995
- [16] ARToolKit http://www.hitl.washington.edu/research/shared_space/download