

리바이저: 프로젝션 화면에서 문서교정을 위한 실시간 제스처 인터페이스

문채현*, 강 현*, 정기철**, 김항준*

*경북대학교 컴퓨터공학과, **송실대학교 미디어학부

e-mail : chmoon@ailab.knu.ac.kr

TheReviser: A Real-time Gesture Interface for a Document Polishing Application on the Projection Display

Chae Hyun Moon*, Hyun Kang*, Keechul Jung**, Hang Joon Kim*

*Dept. of Computer Engineering, Kyungpook National University

**School of Media, College of Information Science, Soongsil University

요 약

본 논문은 프로젝션 화면에서 사용자와 컴퓨터간의 상호작용을 위한 인터페이스인 리바이저를 제안한다. 리바이저는 프로젝션 화면에서 사용자의 손 제스처로 문서교정 부호를 인식하여 전자문서를 교정하는 인터페이스 시스템이다. 프로젝션 화면에서 이러한 제스처를 인식하려면 두 가지 문제가 있다. 하나는 다양한 빛의 변화와 복잡한 배경을 가지는 프로젝션 화면에서 전방물체를 검출하는 것이고, 다른 하나는 사용자의 연속적인 움직임에서 사용자의 의도를 알아내는 것이다. 전방물체를 검출하기위해 프로젝터로 입력되는 영상과 카메라에의해 캡처된 영상 사이의 색상정보를 사용한다. 이 색상정보는 다양한 빛의 변화와 복잡한 배경에서 전방물체를 구별할 수 있게한다. 연속된 제스처를 인식하기위해 적출과 인식이 포함된 새로운 제스처 적출방법을 제안한다. 실험결과 제안된 시스템에서 의미없는 움직임이 포함된 제스처 시퀀스에서 사용자 제스처의 인식률은 평균 93.22%로 만족할만한 결과를 보였다.

1. 서론

프로젝션 화면(Projection Display)는 전기적인 신호로부터 큰 영상을 생성하는 디스플레이의 일종이다[1]. 최근들어 프로젝션 화면에서의 상호작용 시스템들은 유비쿼터스 컴퓨팅(Ubiquitous Computing)과 증강 현실(Augmented Reality)기술을 위하여 많은 가능성을 제공한다. 사용자는 프로젝션 화면상에서 가상 물체를 마치 실제 물체처럼 직접 만지고, 쥐고, 명령을 내릴 수가 있다. 프로젝션 화면에서의 상호작용 시스템들은 사용자에게 큰 작업공간과 디스플레이 화면을 제공하고, 키보드나 마우스를 대신하여 직접적인 상호작용을 가능하게 한다. 이는 디지털데스크(DigitalDesk), 브라이트 보드(Bright Board), 인텔리전트 룸(Intelligent Room)등의 응용제품들에서 보여져 왔다[2-4].

본 논문에서는 프로젝션 화면상의 상호작용 시스템인 리바이저(TheReviser)를 제안한다. 리바이저는 사용자에게 프로젝션 화면에서 워드 프로세서의 문서를 보여주고, 펜으로 실제 문서를 교정하듯이 사용자의 손 제스처(Gesture)로 전자문서 교정을 할 수 있게하는 인터페이스 시스템이다. 이러한 시스템을 구현하려면, 두 가지 문제가 있다. 하나는 빛의 변화가 다양한 프로젝션 화면에서에서 전방물체를 검출하는 것이고[5], 다른 하나는 사용자의 의미없는 움직임에서 제스처 명령을 적출하는 것이다[6].

리바이저는 새로운 색상 기반의 전방물체 검출(Foreground Objects Detection)방법과 인식 기반의 제스처 적출(Gesture

Spotting)방법이 적용된다. 적용된 전방물체 검출방법은 프로젝트로 입력되는 프레임 버퍼 영상과 카메라에 의해 캡처된 카메라 영상간의 차이 값을 기반으로 한 방법이고, 제스처 적출방법은 사용자의 연속적인 움직임에서 제스처 부분을 적출하고 인식하는 인식 기반의 제스처 적출방법이다.

본 논문의 구성은 다음과 같다. 2 장에서는 제안된 시스템인 리바이저와 전방물체 검출방법과 그리고 제스처 적출방법을 설명하고, 3 장에서는 제안된 시스템의 실험 결과를 보여준다. 끝으로 4 장에서 결론을 기술한다.

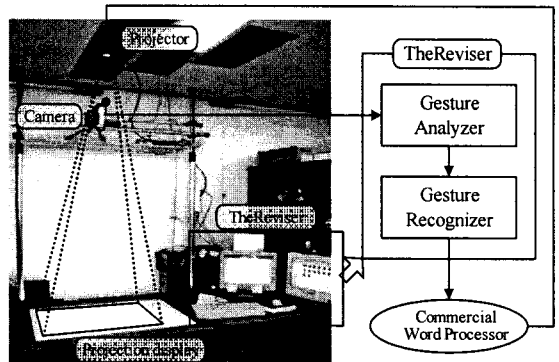


그림 1. 제안된 시스템의 구성도.

2. 리바이저

리바이저는 프로젝션 화면에서 컴퓨터와 사용자 사이에 상호작용을 위한 인터페이스이다. 리바이저는 프로젝션 화면에서 마우스를 사용하는 것처럼 사용자가 손으로 가상의 물체를 조작할 수 있게 하고, 사용자는 실제 문서를 빨간색 펜으로 교정하는 것과 같이 손으로 가상의 문서를 교정할 수 있다. 그림 1은 프로젝션 화면에서 워드 프로세서의 전자문서를 교정하는 리바이저의 구성도를 보여준다. 프로세서의 실제 물리적인 책상위에서 컴퓨터의 화면을 투영하고, 카메라로 투영된 화면과 사용자의 손 움직임이 포함된 영상을 캡처한다. 리바이저는 사용자의 손 움직임이 포함된 영상에서 사용자의 제스처를 인식하여 전자문서를 교정한다.

리바이저는 총 12개의 문서교정 제스처들을 처리한다. 그림 2는 6개의 상태와 6개의 상태변환으로 이루어진 리바이저의 상태 다이어그램을 보여준다. 리바이저는 이 상태 다이어그램에 의해 제어되는데, 여기에서 상태는 인식된 제스처에 따라 수행되는 교정 명령이고, 상태변환은 인식된 제스처에 따라 상태를 변화시키는 것이다. 리바이저의 초기 상태는 'IDEL' 상태로, 이 상태는 아무 교정 명령도 수행하지 않는 상태이다. 각 상태에서 상태변환은 제스처가 인식될 때 발생한다. 'unconditional' 상태변환은 어떤 상태에서 교정 명령이 수행된 후에 발생한다. 예를 들면, 'IDEL' 상태에서 'REVISE a' 상태로의 상태 변환은 인식된 제스처가 'revision gesture group #1' 일 때 발생하고, 상태가 변환되면 'REVISE a' 라는 교정 명령이 수행된 후에 다시 'IDEL' 상태로 상태를 변환한다.

리바이저가 처리하는 제스처는 선택 제스처와 교정 제스처로 나누어진다. 선택 제스처는 워드 프로세서에서 한 문단이나 단어를 선택하는 제스처로 문단이나 단어의 시작과 끝을 나타내는 두개의 위치 인자를 필요로 한다. 교정 제스처는 실제 문서를 교정하는 제스처로 요구되는 위치 인자의 수에 따라서 4 종류의 교정 제스처가 있다. 위치 인자들은 인식한 제스처에 따라 각각 다른 방법으로 계산한다.

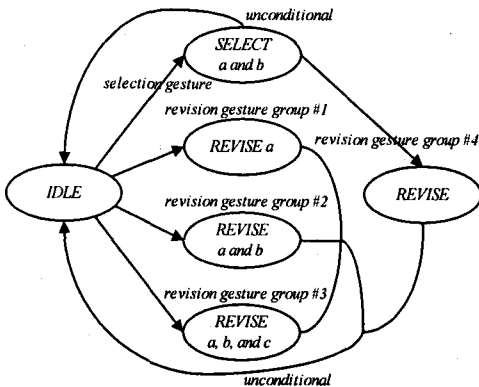
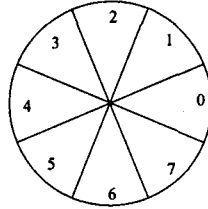


그림 2. 리바이저의 상태 다이어그램.

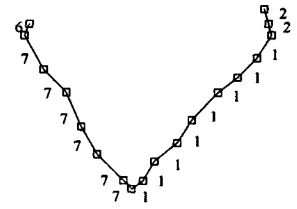
제한된 제스처 인식방법에서는 fingertip의 이동경로로 사용자의 손 제스처를 모델링한다. 이 fingertip 이동경로는 8방향으로 양자화된 심볼열로 표현된다. 이 심볼열의 계산은 연속적인 두 프레임에서 fingertip의 위치와 수평방향과의 각으로 계산된다. 그림 3은 제한된 '사이띄우기'라는 제스처 모델과 8방향의 심볼을 보여준다. 이 제스처는 '67777777111111122'의 심볼열로 표현된다.

문서교정을 위한 제스처들은 표 1과 같다. 이 제스처들은

실제 문서교정에 사용되는 교정부호로 정의한다. 사용자의 fingertip 움직임에서 어떤 문서교정 제스처인지를 인식하기 위해 두 종류의 영상을 처리한다. 첫번째 영상은 투영된 화면위의 카메라에서 얻어진 카메라 영상이고, 다른 하나는 프로젝터에 입력되는 프레임 버퍼 영상이다.



(a) 8 방향 심볼



(b) '사이띄우기' 제스처

그림 3. 8 방향 심볼과 샘플 제스처.

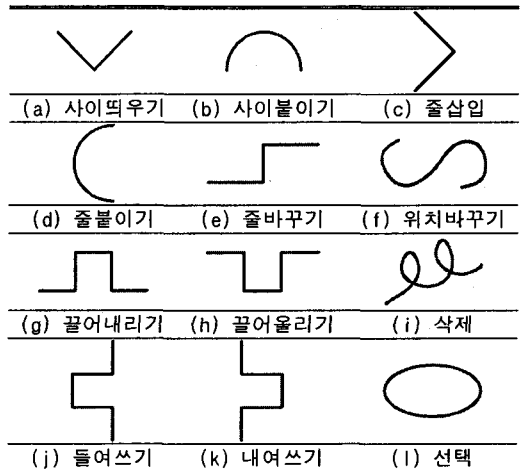
리바이저는 그림 1과 같이 제스처 분석 모듈과 제스처 인식 모듈로 이루어진다. 제스처 분석 모듈은 입력되는 여러 영상에서 전방물체를 추출한 후 사용자의 fingertip 위치를 찾고 이동경로를 추출한다. 여기서 전방물체는 사용자 손 영역이고, 추출된 손 영역에서 fingertip 위치를 찾고, 이동경로를 추출한다. 사용자의 손 영역에서 fingertip의 위치를 찾고 이동경로를 추출하는 방법은 Oka의 방법을 사용한다[3].

Fingertip 이동경로를 추출한 후, 제스처 인식모듈이 수행되는데, 제스처 인식 모듈은 사용자의 연속적인 fingertip 이동경로에서 제스처 부분을 적출하고 심볼릭 제스처로 인식한다. 연속적인 fingertip 이동경로는 여러 후보 제스처들로 분할되고, 분할된 후보 제스처들은 제스처 인식기에 의해 문서교정 기호로 인식된다.

2.1 전방물체 검출

이 절에서는 프로젝션 화면에서 프로젝터로 투영할 프레임 버퍼 영상과 카메라 영상에서 얻은 색상 정보를 이용한 전방물체 검출 방법을 설명한다. 프로젝터로 투영된 영상을 다시 카메라로 입력받게 되면 영상은 왜곡되게 된다.

표 1. 문서교정을 위한 심볼릭 제스처들.



제한된 방법에서는 카메라 영상이 기하왜곡과 색상왜곡이 존재한다고 가정한다. 이 두 왜곡을 보정하기 위해, 먼저, 기하왜곡은 프레임 버퍼 영상과 카메라 영상간의 일치하는

픽셀들을 찾음으로써 보정되고, 색상왜곡은 두 영상간의 샘플 색상들의 변화를 측정함으로써, 투영된 화면의 색상 변화 모델을 구성하여 보정된다. 두 보정단계를 거친 영상과 프레임 버퍼 영상과의 픽셀 밝기 차이를 계산하여 전방물체를 검출하게된다.

2.1.1 기하보정

프레임 버퍼 영상과 카메라 영상 간의 픽셀들을 일치시키기 위해 기하보정 방법이 필요하다. Weller는 디지털 데스크에서 4-6 개의 점을 와핑하는 방법을 소개하였고[7], Ashdown은 점 와핑 방법의 한계를 넘는 투영 변환 방법을 제안하고 평가하였다[8]. 제안된 방법에서는 투영 변환방법을 사용한다. 그림 4 와 같이 프레임 버퍼영상과 카메라 영상간의 관계를 투영 변환라고 가정한다. 원점 'o'는 카메라의 위치와 프로젝터의 위치이고, 점 P 와 P'는 각각 프레임 버퍼 영상과 일치하는 카메라 영상의 한 점이다.

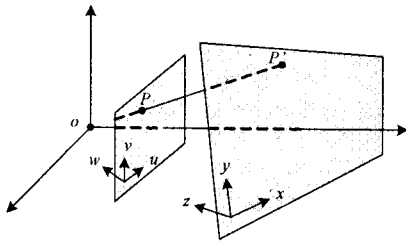


그림 4. 프레임 버퍼영상과 카메라영상간의 투영변환관계

이 투영 변환 관계는 식 1의 행렬(H)에 의해 표현된다.

$$(x, y, w) = H \begin{pmatrix} u \\ v \\ z \end{pmatrix}, H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (1)$$

a 부터 i 까지 9 개의 상수는 4 개 이상의 프레임 버퍼 영상의 점(u, v)과 프로젝션 화면 상의 점(x, y)의 매칭되는 점들에 의해서 계산된다. 매칭되는 점을 이용한 9 개의 상수의 계산은 Hartley 와 Zisserman 의 방법을 사용하여 구한다[9].

2.1.2 색상보정

프로젝션 화면에서 카메라 영상의 색상왜곡을 보정하는 색상 보정 방법을 설명한다. 색상왜곡은 프레임 버퍼 영상과 카메라 영상 간에서 일치하는 픽셀들의 색상차이로 정의된다. 프레임 버퍼 영상과 카메라 영상간의 색상 차이를 고려하여 현재 픽셀 또는 블록이 전방물체에 속하는지 안 하는지를 결정한다. 시간에 따른 변화를 측정하기 위하여 프로젝션 화면의 일부에 밝기 바(Bar)를 만든다. 밝기 바는 k 개의 블록으로 이루어져 있고, 각각의 블록은 여러 단계의 밝기 값을 가진다. 이는 투영되는 화면의 상단에 항상 보이도록 설치되어 프레임 버퍼 영상과 카메라 영상 간의 밝기 왜곡 정보를 계산한다. 이 왜곡정보를 이용하여 어떤 픽셀 또는 블록이 전방물체에 속하는지 안 하는지를 아래 식들에 의해 결정된다.

$$F_k = f(x_k, y_k) \quad (2)$$

$$\hat{C}_k = \frac{B_{k+1} - F_k}{B_{k+1} - B_k} + B_k, B_k \leq F_k \leq B_{k+1}, 0 \leq k \leq 9 \quad (3)$$

$$O_k = \begin{cases} \text{background,} & \|\hat{C}_k - C_k\| < \alpha \\ \text{foreground object,} & \text{otherwise} \end{cases} \quad (4)$$

여기에서 어떤 픽셀(s)의 프레임 버퍼 영상 내의 밝기 값(F_k)과 카메라 영상 내의 밝기 값(C_k) 사이의 차이에 따라 픽셀의 전방 물체 여부(O_k)가 결정된다. \hat{C}_k 는 밝기 값(F_k)를 가진 픽셀이 왜곡되어졌을 때 추정 값으로 밝기의 측정 값(B_k), 0 ≤ k ≤ 9 와의 선형 보간에 의해서 결정된다.

2.2 제스처 적출

이 절에서는 적출과 인식을 동시에 수행하는 제스처 적출 방법을 설명한다. 제스처 적출방법은 후보 컷 생성단계와 제스처 인식단계로 이루어진다. 후보 컷 생성단계는 연속되는 사용자의 움직임에서 후보 컷을 검출한 후, 심볼 열과 특징 벡터 열로부터 임시 제스처를 생성하고, 제스처 인식 단계는 임시 제스처의 인식 값을 사용하여 정의된 제스처를 인식하는 단계이다.

2.2.1 후보 컷 생성

후보 컷 생성 단계에서는 fingertip의 속력과 fingertip 이동 경로의 곡률 정보를 사용하여 후보 컷을 검출한다. 따라서, 시간도메인에서 속력이 '0'이 되는 프레임, 공간도메인에서 몇 개의 곡률을 가진 프레임일 때 후보 컷이 생성된다. 임시 제스처들은 후보 컷들이 검출된 후에 생성된다. 임시 제스처는 생성된 후보 컷들로부터 제스처의 시작점과 끝점이 될 수 있는 제스처다. N×N 배열로 구성된 시간공간 안에서 특정 범위를 가르키는 슬라이딩 윈도우를 사용하여 임시 제스처를 검출한다. 윈도우 안에서 모든 임시 제스처들에 대한 인식값이 계산한 후, 인식기는 가장 높은 인식 값을 가지고 임계값보다 높은 임시 제스처를 인식한다. 제스처가 슬라이딩 윈도우 안에서 인식된 후, 제한된 크기의 윈도우는 다음 제스처를 적출하기 위해 움직인다. 이전 제스처의 끝점이 다음 슬라이딩 윈도우의 시작점이 된다. 효율성 증대를위해, 임시 제스처들은 두 개의 인접하는 슬라이딩 윈도우를 포함하고 있는 겹쳐진 블록들을 제외한 슬라이딩 윈도우 안에서 인식된다.

2.2.2 제스처 인식

제스처 인식단계는 적출된 임시 제스처의 심볼 열이 주어지면, 제스처 인식기는 적출된 심볼 열이 제스처인지를 결정한다. 제안된 인식방법은 제스처 템플릿을 이용한 다이나믹 프로그래밍을 사용하여 제스처를 인식한다. 이 방법은 열 삽입들과 열 삭제들과 열 대체를 포함한 2 개의 문자열의 값들을 매칭하는 것을 기본으로 한다. 이 때, 하나의 문자열은 하나의 임시 제스처며 또 다른 하나는 제스처 템플릿이다. 제안된 매칭 알고리즘안에서는 식 5 와 같은 재귀적 특성을 정의한다.

$$dist_{i,j} = cost(i, j) + \min\{\alpha \times dist_{i-1,j}, \beta \times dist_{i,j-1}, \gamma \times dist_{i,j-1}\} \quad (5)$$

이 식에서 dist_{i,j}는 i 번째 심볼과 j 번째 심볼을 읽을 때의 최소 매칭 값이고, α, β 그리고 γ는 제약조건이다. cost(i, j)는 i 번째 심볼과 j 번째 심볼 간의 거리를 반환하는 함수이다. cost(i, j)를 계산하기 위해, i 번째 클러스터의 중심과 j 번째 클러스터의 중심간의 유클리디안 거리가 사용된다. 이처럼, 제스처는 입력 심볼열과 각 제스처 템플릿간의 거리에서 얻는 재귀적 특성을 사용한다. 인식기는 가장 짧은 거리를 가진 제스처에 대해 가장 높은 값을 반환하고, 이 제스처를 인식한다.

3. 실험 결과

제안된 시스템의 수행을 위한 실험 환경은 카메라, 프로젝터, 책상 그리고 표준 컴퓨터로 이루어진 Wellner의 디지털 데스크와 유사한 환경이다[2]. 프로젝션 화면은 필립스 ProScreen 4750 LCD 프로젝터에 의해 투영된 1024×768 크기의 32 비트 트루칼라 영상이고, 카메라 영상은 소니의 DCR-TRV 530 캠코더를 사용하여 캡처된 영상이다. 시스템의 구현은 Visual C++ 6.0 과 DirectShow 9.0 SDK 를 사용하여 720×480 크기의 24 비트 컬러 영상을 초당 15 프레임을 획득하는 시스템을 구축하였다.

그림 5 는 전방물체 검출 결과를 보여준다. (d)는 (c)의 기하보정 영상을 색상보정을 거친 후, (a)의 프레임 버퍼 영상과 차이로 전방물체를 검출한 결과이다. 사실 아래 결과는 차이값으로 이진화한 후, 모폴로지 연산과 연결성분 분석으로 노이즈를 제거한 영상이다.

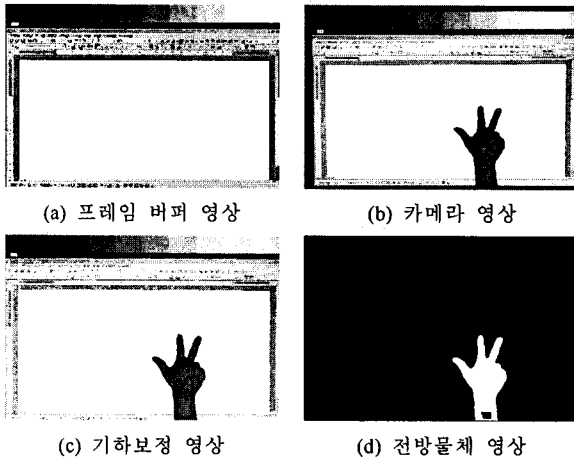


그림 5. 전방물체 검출의 결과.

그림 6 은 제스처 적응의 결과들을 보여준다. 그림에서 오른쪽위의 기호가 제스처를 인식한 결과를 보여준다. 가운데의 점들은 각 프레임에서 fingertip의 위치를 나타내고, 왼쪽 아래의 숫자열은 fingertip의 방향을 나타내는 심볼열이다.

제안된 시스템의 수행결과, 제스처가 아닌 패턴을 제스처라고 하는 삽입에러, 제스처를 찾지 못하는 삭제에러 그리고 다른 제스처와 혼동하는 대체에러 같은 형태의 에러들이 나타났다.

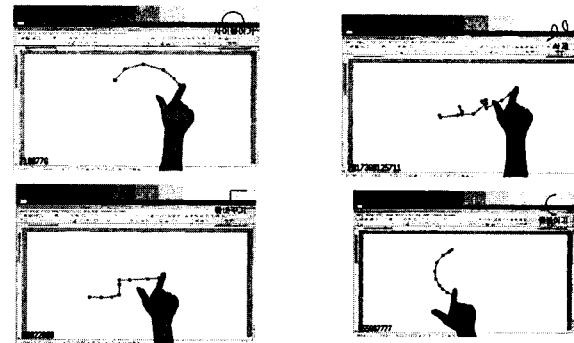


그림 6. 제스처 적응의 결과.

표 2. 제스처 적응의 수행결과.

제스처	#	e	에러 형태			검출비	신뢰도
			삽입	삭제	대체		
사이드무기	185	175	3	4	6	94.59	93.09
사이불이기	195	186	3	4	5	96.38	93.94
선택	200	193	5	4	3	96.50	94.15
삭제	200	195	5	2	3	97.50	95.12
위치바꾸기	200	194	3	3	3	97.00	95.57
중삽입	198	191	2	2	5	96.46	95.50
중불이기	166	157	3	4	5	94.58	92.90
끌어내리기	163	149	6	7	7	91.41	88.17
끌어올리기	160	150	2	5	5	93.75	92.59
중바꾸기	200	194	3	4	2	97.00	95.57
돌려쓰기	197	183	5	10	4	92.89	90.59
내어쓰기	195	182	4	9	4	93.33	91.46
합계	2259	2149	44	58	52	95.03	93.22

제안된 방법의 신뢰도를 측정하기 위해 Lee and kim[6]의 측정법을 사용하였는데, 표 2 와 같은 결과를 얻었다. 표에서 '#'는 제스처의 수행횟수이고, '@'는 바르게 인식한 제스처의 횟수이다. 제스처 적응의 신뢰도와 속도는 슬라이딩 윈도우의 크기에 따라 결정되었는데, 크기가 6 일 때 가장 높은 신뢰도와 가장 빠른 수행속도를 가졌다.

4. 결론

본 논문에서는 프로젝션 화면에서 사용자의 손으로 전자 문서를 교정하는 시스템인 리마이저를 제안하였다. 제안된 시스템은 새로운 색상 기반의 전방물체 검출방법과 인식 기반의 제스처 적응방법이 적용된 상호작용 시스템이다. 다양한 빛의 변화를 가지는 프로젝션 화면에서 전방물체를 검출하기 위해 프로젝터로 입력되는 영상과 카메라에 의해 캡처된 영상 사이의 색상정보를 사용한다. 인식 기반의 제스처 적응방법은 연속적인 사용자의 움직임이 포함된 제스처 시퀀스에서 제스처 부분을 적응하고 인식하는 방법이다. 실험결과, 제스처의 인식률은 평균 93.22%로 만족할만한 결과를 보였다. 프로젝션 화면에서 실시간으로 사용자의 손 제스처를 인식하여 문서를 교정하는데 효율적인 성능을 보였다.

참고문헌

- [1] E.H. Stupp, M.S. Brennesholtz, Projection Display, John Wiley & Son, 1999.
- [2] P. Wellner, "The DigitalDesk Calculator: Tactile Manipulation on a Desk Top Display", Proceedings of UIST 1991, ACM, pp. 27-33, 1991.
- [3] J.Q. Stafford-Fraser, P. Robinson, "BrightBoard: A Video-Augmented Environment", Proceedings of CHI 1996, ACM, pp. 134-141, 1996.
- [4] M.H. Coen, "Design Principles for Intelligent Environments", Proceedings of the AAAI'98, Madison, WI, 1998.
- [5] K. Oka, Y. Sato, "Real-time Fingertip Tracking and Gesture Recognition", IEEE Computer Graphics and Applications, pp. 64-71, 2002.
- [6] H.K. Lee, J.H. Kim, "An HMM-based Threshold Model Approach for Gesture Recognition", IEEE Transaction on PA MI, 21 (10), pp. 961-973, 1999.
- [7] P. Wellner, "Self Calibration for the DigitalDesk", EuroPARC Technical Report EPC-93-109, 1993.
- [8] M. Ashdown, P. Robinson, "The Escritoire: A Personal Projected Display", Journal of WSCG, 11(1), pp. 33-40, 2003.
- [9] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2001.