

TheReviser : 가상 데스크 상의 제스처 기반 문서 교정 시스템

정 기 철* · 강 현**

요 약

리바이저 시스템은 프로젝션 화면 상에서 종이 문서의 수정시 사용되는 교정 제스처와 동일한 제스처를 이용한 온라인 문서 교정 시스템이다. 이를 위해, 프로젝션 화면 상에서 손이나 문서와 같은 전경물체추출 기술과 연속 동작으로 부터의 제스처 인식 기술이 필요하다. 많은 조명 변화와 복잡한 배경 상에서 전경물체를 검출하기 위해서 기하보정과 색상보정을 수행하고, HMM 기반 제스처 인식기를 구현하였다. 실험 결과로부터 연속 제스처에서 93.22%의 인식률을 나타남을 볼 수 있다.

TheReviser : A Gesture-based Editing System on a Digital Desk

Keechul Jung* · Hyun Kang**

ABSTRACT

TheReviser is a digital document revision application on a projection display, which allows us to interact a digital document with the same gestures used for paper documents revision. To enable these interactions, TheReviser should detect foreground objects such as hands or pens on a projection display, and should spot and recognize gesture commands from continuous movements of a user. To detect foreground objects from a complex background in various lighting conditions, we perform geometry and color calibration between a captured image and a frame buffer image. TheReviser uses an HMM-based gesture recognition method. Experimental results show that the proposed application recognizes user's gestures on average 93.22% in test gesture sequences.

키워드 : 디지털데스크(Digital Desk), 제스처 인식(Gesture Recognition), 전경물체 검출(Foreground Detection), 색상 보정(Color Calibration), 기하 보정(Geometry Calibration)

1. 서 론

프로젝션 화면(projection display)에서의 상호 작용(interaction) 시스템은 증강현실(augmented reality)과 유비쿼터스 컴퓨팅(ubiquitous computing)등의 연구분야에 많은 활용도를 지니는 분야이다. 프로젝션 화면에서 사용자는 가상의 물체를 실제 물체처럼 만지거나 명령을 내릴 수 있게 하고, 문서 등의 실제 물체에 가상의 정보를 오버레이(overlay) 할 수 있다. 프로젝션 화면은 사용자에게 큰 작업환경을 제공하고, 키보드나 마우스를 대신하여 손을 이용한 직접적인 상호작용을 할 수 있는 장점이 있다. 이와 관련된 연구들로 디지털 데스크(DigitalDesk)[24]와 브라이트보드(Bright board)[20], 인텔리전트 룸(Intelligent room)[3], 인터랙티브 워크스페이스(Interactive workspace)[10] 등이 있다.

본 논문은 프로젝션 화면 상의 디지털 문서 교정 시스템인

리바이저(TheReviser) 시스템을 제안한다(그림 1). 리바이저는 프로젝션 화면에서 사용자에게 워드프로세서의 디지털 문서를 보여주고, 사용자의 손 움직임으로부터 문서 교정 제스처(gesture)를 인식함으로써 디지털 문서를 교정할 수 있는 환경이다. 이를 위해 본 연구에서는 다양한 밝기 변화를 가지는 프로젝션 화면 상의 전경물체(foreground objects) 검출과, 손을 이용한 교정용 제스처 인식 방법을 제안한다[12, 15].

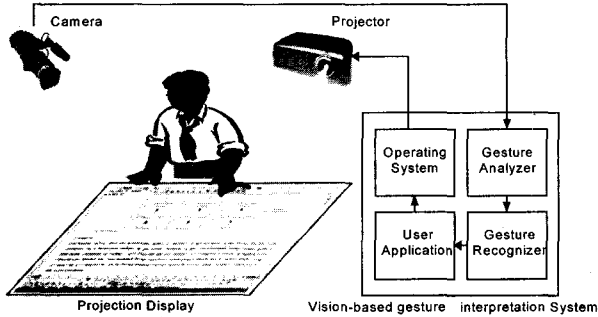
프로젝션 화면 상의 전경물체를 검출하기 위해, 일반적으로 추가적인 센서나 시간에 따른 영상의 변화정보를 이용한다. Oka는 EnhancedDesk 시스템에서 적외선 카메라를 사용하여 사람의 체온을 감지함으로써 사용자의 손(전경물체)을 검출하였고[15], Rekimoto는 Smartskin 시스템에서 압력 센서를 사용하여 전경물체를 검출하였다[18]. Kjeldsen와 Kender의 BlueSpace와 Stafford-Fraser의 BrightBoard는 전경물체가 움직인다고 가정하고, 연속된 두 프레임 사이의 변화정보를 이용하여 전경물체를 검출하였다[12, 20]. 위와 같은 전경물체 검출 방법은 많은 제약이 있다. 적외선 카메라는 무생물인 물체는 검출이 어렵고, 압력센서의 경우는 사용자의 손이 센서에 닿지않으면 검출이 불가능하며, 연

* 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2003-042-D00166).

† 종신회원 : 숭실대학교 정보과학대학 미디어학부 교수

** 정 회 원 : 한국전자통신연구원 디지털콘텐츠단 연구원
논문접수 : 2003년 12월 16일, 심사완료 : 2004년 6월 11일

속된 두 프레임의 변화정보를 이용한 방법은 움직이지 않은 물체는 검출할 수 없다는 단점이 있다.



(그림 1) 프로젝션 화면에서 비전기반의 제스처 인식 시스템의 구성도

또한 제스처 기반 문서 교정 시스템을 위해 의미있는 제스처의 검출(detection)과 인식(recognize) 과정이 필요하다. 제스처 검출을 위해, 이전 연구들은 연속된 제스처의 검출과 인식에서, 의미없는 움직임을 처리하기 위해 분리된 제스처 인식기들을 확장·결합하여 사용하였다[9, 26]. 그리고 또 다른 연구에서는 3개의 제스처 움직임 구분을 정의하고 이 구분들의 시간 변화를 검출하여 제스처를 검출하였다[27, 28].

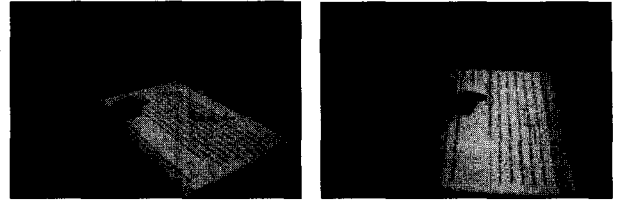
본 논문에서는 프로젝터로 입력되는 영상과 카메라로 촬영되는 영상 사이의 차이를 이용한 새로운 전경물체 검출방법을 제안하며, HMM을 이용하여 교정 제스처를 인식한다. 제안된 전경물체 검출 방법은 프로젝션 화면을 캡처한 카메라 영상에 기하 왜곡(geometry distortion)과 색상 왜곡(color distortion)이 존재한다고 가정하고, 기하 보정과 색상 보정을 거친 후, 카메라 영상과 프로젝터로 입력되는 영상과의 차이를 계산하여 전경물체를 검출한다. 사용자의 의미없는 움직임이 포함된 영상에서 교정 제스처를 검출하기 위해 사용자 손의 특정한 모양을 이용하여 제스처를 검출하며, 검출된 제스처는 각각의 인식 결과를 기반으로 하여 문서교정 제스처로 인식한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안된 리바이저 시스템과 문서교정을 위한 제스처에 관하여 설명하고, 3장에서는 전경물체 검출 방법을, 4장에서는 제스처 인식방법을, 5장에서는 실험 결과를 보여준다. 끝으로 6장에서 결론 및 향후과제를 기술한다.

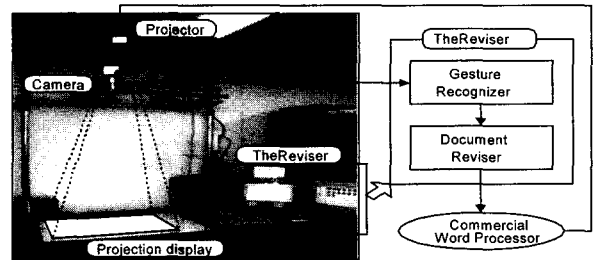
2. 리바이저

리바이저는 (그림 2)와 같이 사용자에게 표준 모니터처럼 디지털 문서를 보여주고 문서교정 제스처를 이용하여 문서의 내용을 교정할 수 있게한다. 이는 프로젝션 화면 상에서 사용자와 컴퓨터 간의 상호작용 인터페이스를 지원하며, 사용자의 손 움직임을 인식하여 디지털 문서를 교정하는 시스템으로써, 표준 모니터보다 큰 작업공간을 지원하고, 여러 사용자가 프로젝션 화면에서 작업을 할 수 있게하며, 사

용자가 손을 이용하여 실제 문서를 펜으로 교정하듯이 디지털 문서를 교정할 수 있게 한다.



(그림 2) 프로젝션 화면에서의 디지털문서 교정의 예



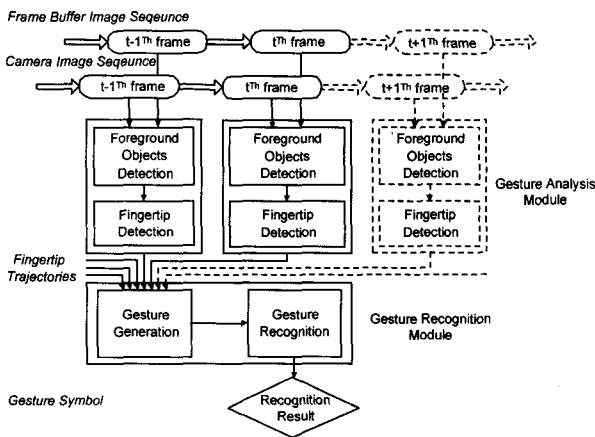
(그림 3) 리바이저의 구성도

(그림 3)은 리바이저의 전체적인 구성도를 보여준다. 프로젝션 화면은 실제 물리적인 책상 위에 위치한 프로젝터에 의해 생성되고, 카메라는 투영된 화면과 사용자의 손 움직임이 포함된 영상을 캡처한다. 리바이저는 크게 제스처 인식기와 문서 교정기로 구성된다. 제스처 인식기는 프로젝션 화면에서 사용자의 손 제스처가 어떤 문서교정 제스처인지를 인식하고, 문서 교정기는 인식된 문서교정 제스처에 따라 워드프로세서의 디지털 문서를 교정한다. 리바이저에서 문서교정을 위해 사용된 사용자의 손 제스처는 (그림 4)와 같이, 한글 문서교정에 사용되는 일반적인 문서교정 부호 15개 중 12개를 사용하며[7], 제스처 모양에서 기호 '●'는 제스처의 시작 부분이고, '▶'는 제스처의 끝 부분이다.

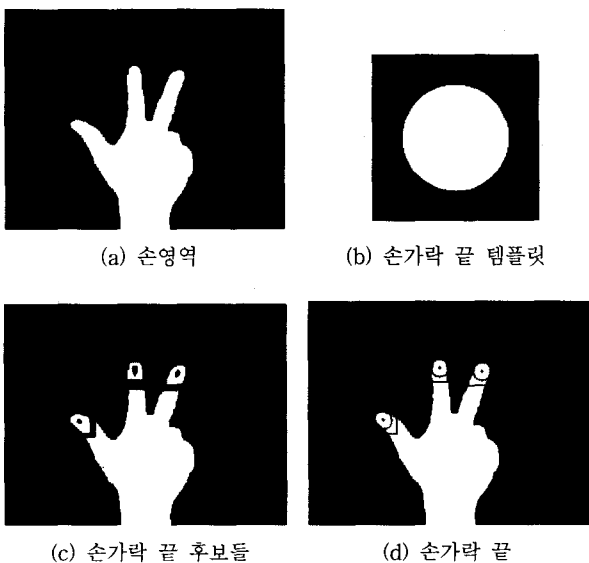
종류	그룹	제스처	제스처 모양	인자수	위치 인자들	
선택제스처		선택		2	시작, 끝	
교정 제스처	그룹 1	삭제		0		
		사이띄우기		1	중간	
	그룹 2	줄삼입		1	중간	
		줄바꾸기		1	중간	
	그룹 3	사이붙이기		2	시작, 끝	
		줄잇기		2	시작, 끝	
	그룹 4	위치바꾸기		3	시작, 중간, 끝	
		들어쓰기		3	시작, 중간, 끝	
		내어쓰기		3	시작, 중간, 끝	
		끝어올리기		3	시작, 중간, 끝	
			끝어내리기		3	시작, 중간, 끝

(그림 4) 문서교정을 위한 심볼릭 제스처들

(그림 5)는 리바이저에서 제스처 인식기의 블록 다이어그램을 보여준다. 제스처 인식기는 크게 제스처 분석모듈(gesture analysis module)과 제스처 인식모듈(gesture recognition module)로 이루어진다[16]. 제스처 인식기는 사용자의 제스처를 인식하기 위해, 두 종류의 영상 시퀀스를 처리한다. 첫번째 영상 시퀀스는 프로젝션 화면 위의 카메라로 캡처된 카메라 영상(camera image)이고, 다른 하나는 프로젝터에 입력되는 프레임버퍼 영상(frame buffer image)이다. 제스처 분석모듈은 입력되는 두 종류의 영상 시퀀스에서 전경물체(손)를 추출한 후, 사용자의 손가락 끝위치를 찾고 이동경로를 추출한다. 사용자의 손 영역에서 손가락 끝의 위치를 찾고 이동경로를 추출하는 방법은 Oka의 방법을 사용한다[15]. 제스처 인식모듈은 손가락 끝의 이동경로들을 입력받아 제스처 부분을 추출하여 제스처를 생성하고, HMM (Hidden Markov Model)을 이용하여 어떤 문서교정 제스처 인지를 인식한다[14, 21].



(그림 5) 제스처 인식기의 블록다이어그램



(그림 6) 손가락 끝 검출의 예

Oka가 제안한 손가락 끝의 위치를 찾는 방법은, 사용자의 손 영역에서 원 모양의 템플릿을 매칭하여 가장 잘 매칭되는 여러개를 손가락 끝 후보로 찾은 후, 그 후보들 중에서 가장 잘 매칭되는 후보와 일정거리에 있는 후보들은 제거하고, 남은 후보 중 템플릿의 세 면의 픽셀이 손 영역이 아닌 것만을 남겨 손가락 끝 위치를 찾는다. (그림 6)은 사용자의 손 영역에서 손가락 끝 위치를 찾는 예를 보여준다. (그림 6)(a)는 사용자의 손 영역을 나타내고, (그림 6)(b)는 손가락 끝의 템플릿이다. (그림 6)(c)는 찾은 손가락 끝 후보들을 나타내고, (그림 6)(d)는 최종적으로 찾은 손가락 끝을 나타낸다.

3. 전경물체 검출

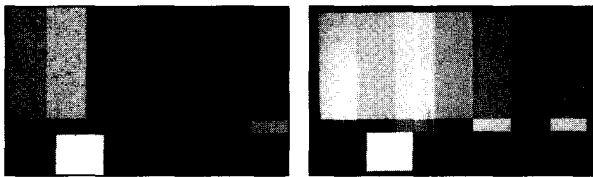
본 3장에서는 심한 조명변화와 복잡한 배경을 가지는 프로젝션 화면에서 프로젝터로 투영할 프레임버퍼 영상과 카메라로 캡처된 카메라 영상의 색상 정보를 이용한 전경물체 검출 방법을 기술한다. 만약 프로젝션 화면을 캡처한 카메라 영상에 왜곡이 존재하지 않는다면, 두 영상의 차이를 이용하여 간단하게 전경물체를 찾을 수 있을 것이다. 하지만, 프로젝터로 투영된 영상을 다시 카메라로 입력받은 영상에는 다양한 왜곡이 존재한다[5, 19]. 본 연구에서는 전경물체 검출 시, 카메라 영상에 기하 왜곡과 색상 왜곡이 존재한다고 가정하며, 기하 왜곡은 프레임버퍼 영상과 카메라 영상 사이의 일치하는(corresponding) 픽셀들을 찾음으로써 보정하고, 색상 왜곡은 두 영상 사이의 샘플 색상들의 변화를 측정하여 프로젝션 화면의 색상 변화 모델을 구성하여 보정한다. 이러한 두 보정단계를 거친 카메라 영상과 프레임버퍼 영상 사이의 픽셀값의 차이를 계산하여 전경물체를 검출한다. (그림 7)은 왼쪽의 프레임버퍼 영상의 하안점들의 위치가 오른쪽의 카메라 영상에서 변하는 것을 보여준다.



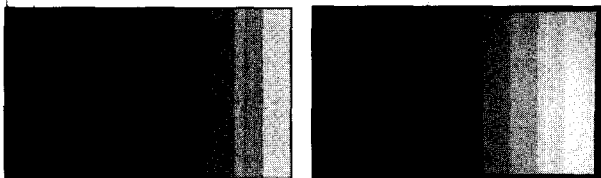
(그림 7) 프레임버퍼 영상(왼쪽)과 카메라 영상(오른쪽)간의 기하왜곡

또 다른 왜곡으로, 프로젝터 및 카메라의 특성에 의해 일반 모니터 영상과 프로젝션 화면을 캡처한 카메라 영상의 색의 차이에 따른 색상 왜곡이 있다. 프레임버퍼 영상 내의 픽셀들의 색상은 프로젝터 및 카메라의 특성에 의해 왜곡된다. LCD 프로젝터의 경우 입력 신호의 세기에 대해서 출력 신호의 세기는 특정곡선 형태를 가진다[1, 29]. (그림 8)과 (그림 9)는 각각 SMPTE 영상과 명암 영상을 이용한 색상왜곡의 예를 보인다. (그림 8)에서는 대부분의 색상 값에서 두

영상이 차이가 많은 나는 것을 알 수 있다. (그림 9)는 10개의 단계별 명암 영상으로 각 단계가 일정한 단위로 밝기값이 설정되어있어서, 프레임버퍼 영상과 카메라 영상으로부터 프로젝터의 특성을 확인할 수 있다. 10단계 명암 영상의 가장 밝거나 가장 어두운 부분들에서는 서로간의 색의 구분이 어렵지만, 중간 밝기의 색상들에서는 서로가 뚜렷이 구별됨을 알 수 있다.



(그림 8) SMPTE¹⁾ 컬러 영상에서의 색상왜곡(프레임버퍼 영상(왼쪽), 카메라 영상(오른쪽))



(그림 9) 10단계 명암 영상에서의 색상왜곡(프레임버퍼 영상(왼쪽), 카메라 영상(오른쪽))

3.1 기하 보정

프레임버퍼 영상과 카메라 영상 사이의 대응하는 픽셀들의 좌표를 일치시키기 위한 기하보정이 필요하다. 프로젝션 화면에서 사용자의 손을 이용한 가상물체와의 상호작용을 위한 전처리 단계로, Weller는 가상 물체와 사용자의 손 위치 사이의 좌표변환을 위해 4~6개의 점을 와핑(warping)하는 방법을 사용하였고[25], Ashdown[1]와 Sukthakar[23]는 투영 변환 방법을 사용하였다[23].

제안하는 전경물체 검출 방법은 투영 변환을 사용하여 기하 왜곡을 보정한다. 기하보정을 위한 투영변환 방법은 프로젝션 화면과 카메라 영상이 평면(planar plane)이라고 가정하고, 프레임버퍼 영상과 카메라 영상 간의 관계를 투영 변환이라고 가정한다. 두 평면 사이의 좌표 변환은 식 (1)의 행렬에 의해 표현된다.

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

a 부터 i 까지 9개의 상수는 4개 이상의 프레임버퍼 영상의 점(u_i, v_i)과 매칭되는 프로젝션 화면 상의 점(x_i, y_i)들에 의해서 계산된다(크기 변화는 무시한다). 이 변환 행렬의 상수들은 식 (2)를 사용하여 구해지며[6], 식 (2)의 좌측 행렬

을 A 라 했을때 이는 프레임버퍼 영상(u_i, v_i)과 카메라 영상(x_i, y_i)사이의 4개 이상의 일치하는 점들에 의해 생성된다. a 부터 i 까지의 상수들은 $A^T A$ 의 가장 작은 고유값들(eigenvalue)과 일치하는 고유벡터(eigenvector)로 얻어진다.

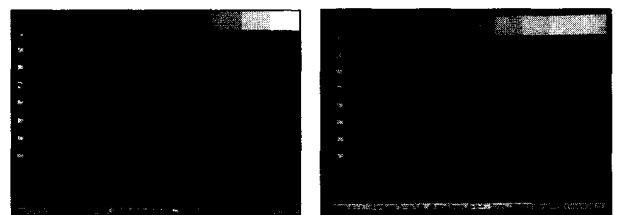
$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0 x_0 - v_0 y_0 & -x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 x_1 - v_1 y_1 & -x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2 x_2 - v_2 y_2 & -x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3 x_3 - v_3 y_3 & -x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0 y_0 - v_0 x_0 & -y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 y_1 - v_1 x_1 & -y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2 y_2 - v_2 x_2 & -y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3 y_3 - v_3 x_3 & -y_3 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{bmatrix} = 0 \quad (2)$$

3.2 색상 보정

색상 보정 단계는 프로젝션 화면에서 프레임버퍼 영상과 카메라 영상 사이의 위치가 일치하는 픽셀들의 색상 차이를 보정하는 단계이다[19]. 카메라 영상과 프레임버퍼 영상 사이의 일치하는 픽셀들의 색상 차이를 계산하기위해, 프레임버퍼 영상과 카메라 영상의 RGB 색상정보로부터 밝기정보를 사용하며, 이는 식 (3)에 의해 YCrCb 색상공간에서 계산된다.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (3)$$

프레임버퍼 영상과 카메라 영상사이의 색상 관계를 추정하기위해, 프레임버퍼 영상은 모든 범위의 밝기 값을 가지고 있어야하지만, 실제 프레임버퍼 영상은 항상 모든 범위의 밝기값을 가지고 있지는 않다. 그러므로 시간적인 색상 변화를 구하기 위해 프로젝션 화면의 상단에 모든 범위의 밝기를 가지는 '밝기 마커(brightness marker)'를 배치한다(그림 10).



(그림 10) 프레임버퍼 영상(왼쪽)과 카메라 영상(오른쪽)의 밝기 마커들

밝기 마커로부터 얻은 k 개의 색상변화값들은 다항식 곡선적합(polynomial curve fitting) 방법을 사용하여 모든 범위의 색상왜곡을 추정할 수 있다. 다항식 곡선적합 방법은 주어진 데이터들을 표현할 수 있는 이상적인 곡선을 얻어내는 방법으로써, 프레임버퍼 영상과 카메라 영상사이의 전체적인 색상변화를 표현하기위해 사용된다. 여기서, 다항식 곡선은 식 (4)와 같이 정의된다. 제안된 방법은 $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_n)^T$ 의 상수들을 가지는 n 차수의 다항식을 사용한다.

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_n x^n \quad (4)$$

1) The SMPTE(Society of Motion Picture and Television Engineers) color bar is the most common pattern for testing encoders, decoders, and other video devices Whitaker and Benson, 2000.

밝기 마커로부터 얻은 k개의 색상변화값들은 $x = (x_1, x_2, \dots, x_k)^T$ 이고, 원래 색상값들은 $y = (y_1, y_2, \dots, y_k)^T$ 이다. 이 값들은 식 (5)와 같은 행렬식으로 정의된다. 식에서 $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_k)^T$ 는 잔여 백터값(residuals)들로 에러를 나타낸다. 이 행렬식은 최소 제곱법(least square method)을 사용하여 식 (6)과 같이 최적의 상수들을 추정한다[13]. 일치하는 픽셀의 카메라 영상에서의 밝기 값과 왜곡 곡선이 주어지면, 카메라 영상의 왜곡된 밝기 값의 추정값을 찾아 실제 밝기 값과 비교하여 전경물체 여부를 결정한다.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \dots & x_k^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_k \end{bmatrix} \quad (5)$$

$$\mathbf{y} = \mathbf{X} \mathbf{a} + \boldsymbol{\epsilon} \quad (6)$$

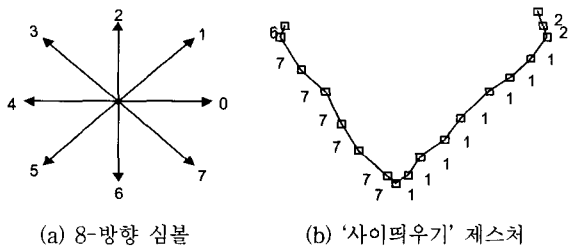
$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

4. 제스처 인식

제안된 제스처 인식방법은 제스처 생성단계와 제스처 인식단계가 수행된다. 제스처 생성단계는 사용자의 손가락 끝 이동경로 중 제스처인 부분을 추출하여 심볼열로 표현하고, 제스처 인식단계는 표현된 심볼열을 은닉 마르코프 모델(Hidden Markov Model : HMM)을 이용하여 어떤 문서교정 제스처인지를 인식한다.

4.1 제스처 생성 단계

제스처 생성 단계는 사용자의 손가락 끝 이동경로를 입력받아 손가락의 이동방향을 나타내는 심볼열로 표현하는 단계이다. 사용자의 손가락 끝 이동경로 중 제스처 추출 단계의 연산 시간을 단축하기위해 제스처인 부분은 손가락 끝의 개수가 하나인 부분이라고 가정한다. 따라서 의미있는 제스처는 손가락 끝의 개수가 한 개로 바뀌는 부분을 시작으로 한 개에서 다른 개수로 바뀌는 부분까지의 이동경로를 의미한다. 이와 유사하게 Oka는 사용자의 손 제스처를 엄지손가락 존재 유무로 판별한다[15].



(그림 11) 8-방향 심볼과 '사이띄우기' 제스처

검출된 손가락 끝 이동경로를 심볼열로 표현하는데, 연속된 두 프레임에서 손가락 끝이 이동한 방향을 8-방향으로 양자화 된 방향 심볼로 표현한다. (그림 11)은 제안된 '사이

띄우기' 라는 문서교정 제스처와 8-방향의 심볼을 보여준다. 이 제스처는 '6777777111111122'의 심볼열로 표현된다.

4.2 제스처 인식 단계

일반적으로 제스처 인식을 위해 DTW(Dynamic Time Warping)[2], FSM(Finite State Machines)[4], HMM[14, 21] 등을 사용한다. 본 시스템에서는 시간간격적인 변이를 가진 음성인식과 온라인 필기인식 분야 등에서 성공적으로 널리 사용되어온 HMM을 사용한다[8, 26]. HMM이란 전이(transition)에 의해 연결된 상태(state)들의 모임으로 각 전이는 두 가지 확률의 집합을 표현한다. 하나는 전이를 하기 위해 필요한 전이확률(transition probability)이고, 다른 하나는 전이가 발생할 때마다 유한한 알파벳에 속하는 각 출력심볼(output symbol)을 발생시키는 조건부 확률을 나타내는 출력확률(output probability)이다[8]. HMM의 구조를 선택하는 것은 훈련 데이터의 양이나 표현하려고 하는 모델에 따라 다르다. 본 논문에서는 시간적 순서에 대한 고려를 위해 좌-우 모델(left-right model)을 사용하며, 각 제스처 모델의 상태의 수는 문서교정 제스처의 복잡성에 따라 3~10개로 설정한다(<표 1>).

<표 1> 문서교정 제스처에 따른 제스처 모델의 상태 수

문서교정 제스처	상태 수	문서교정 제스처	상태 수
선택	8	줄잇기	7
삭제	10	위치바꾸기	7
사이띄우기	3	들여쓰기	6
줄삼입	3	내어쓰기	6
줄바꾸기	4	끌어올리기	6
사이붙이기	5	끌어내리기	6

각 제스처 모델의 학습은 Baum-Welch 알고리즘을 사용한다. Baum-Welch 알고리즘은 모델 파라미터 재추정(model parameter reestimation)문제에 대한 해결방법으로 많이 사용되고 있으며, 현재의 모델 파라미터를 이용하여 훈련 데이터에 대한 모델 정보를 계산하고, 이를 이용하여 훈련 데이터의 정보가 반영되도록 파라미터를 수정하는 과정을 반복적으로 적용하여 모델과 훈련 데이터 사이에 유사도(likelihood)값을 높이는 방식이다. 위와 같은 훈련과정을 모든 훈련 데이터에 대하여 실행하면, 최종적으로 원하는 제스처 모델을 얻을 수 있다. 입력되는 심볼열과 각 제스처 모델과의 유사도를 나타내는 확률값은 forward 알고리즘을 이용하여 계산한다. 제스처 인식은 입력되는 심볼열과 각 모델과의 확률값들 중 가장 높은 값을 갖는 제스처로 인식한다.

5. 실험 결과

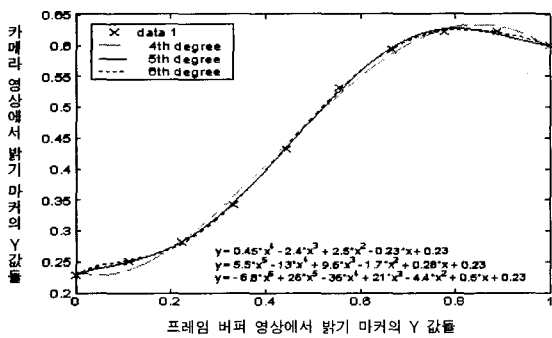
5.1 실험 환경

제안된 시스템의 수행을 위한 실험 환경은 카메라, 프로

젝터, 책상 그리고 표준 컴퓨터로 구성된다. 프로젝션 화면은 필립스 ProScreen 4750 LCD 프로젝터에 의해 프로젝션된 1024×768 크기의 32비트 트루컬라 영상이고, 카메라 영상은 소니의 DCR-TRV 530 캠코더를 사용하여 캡처된 영상이다. 시스템의 구현은 Visual C++ 6.0과 OpenCV beta3.1 [28], 그리고 DirectShow 9.0 SDK를 사용하여, 720×480 크기의 24비트 컬러 영상을 프레임 그래버(frame grabber) 등의 추가적인 장비없이 초당 15프레임을 획득하는 시스템을 구축하였다.

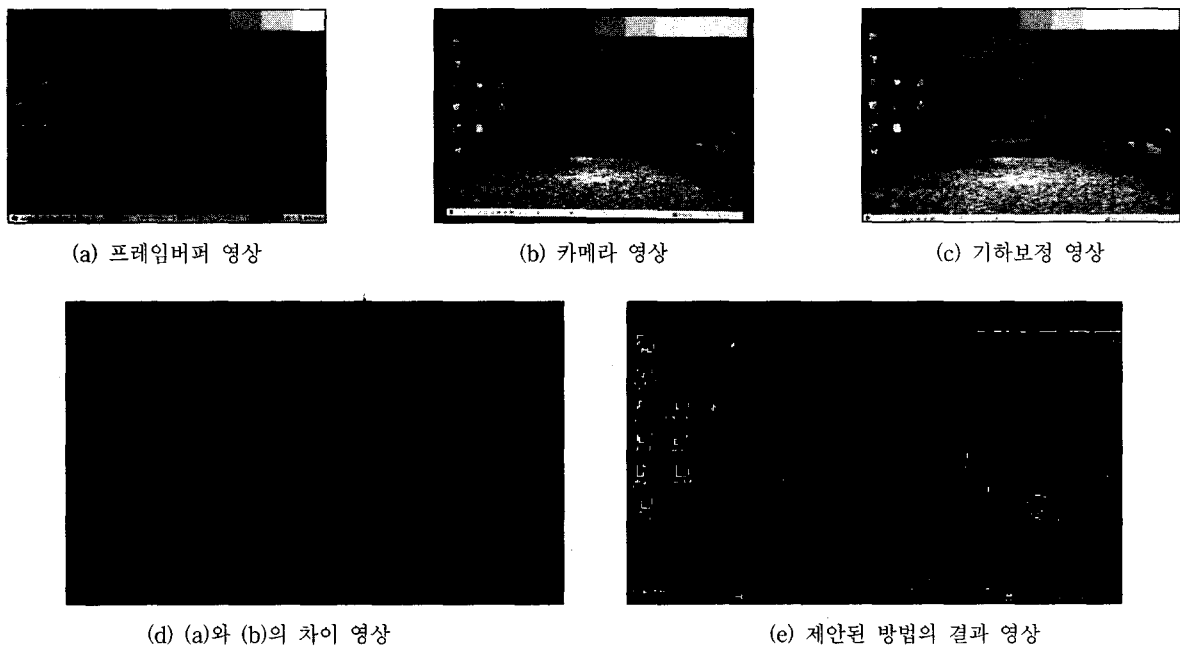
5.2 전경물체 검출의 수행 결과

제안된 전경물체 검출 방법을 평가하기 위해 다양한 환경에서 여러 종류의 실험을 수행하였다. (그림 12)는 색상 왜곡을 보정하기 위해, 프레임버퍼 영상과 카메라 영상의 밝기 마커에서 얻은 픽셀의 밝기 값으로부터 추정한 다양한 다항식의 차수에 따른 색상 왜곡 곡선들을 보여준다.



(그림 12) 다항식의 차수에 따른 왜곡 곡선들

(그림 13)은 복잡한 배경에서 전경물체를 검출한 결과를,



(그림 13) 복잡한 배경에서 전경물체 검출의 수행결과

(그림 14)는 어두운 배경에서의 수행 결과를 보여준다. 각 그림에서 (그림 14)(a)는 프로젝터로 입력된 프레임버퍼 영상, (그림 14)(b)는 프로젝션 화면을 캡처한 카메라 영상, (그림 14)(c)는 카메라 영상을 기하보정한 영상, (그림 14)(d)는 프레임버퍼 영상과 카메라 영상 사이의 차이 영상, (그림 14)(e)는 제안된 방법의 전경물체 검출 영상이다. 각 그림의 (그림 14)(e) 영상에서 흰색 픽셀이 전경물체를 나타낸다. 잡음에 의해서 오-추출(false alarm)된 전경물체는 모폴로지 연산들로 제거할 수 있다.

(그림 15)는 프레임버퍼 영상, 기하보정된 카메라 영상, 전경물체가 검출된 영상들의 확대 영상이다. 확대된 영상들은 원래 영상의 일부를 표현한 사각형으로 영상의 밝기 값만 포함된 영상이다. 이 사각형에서 전경물체는 책의 일부뿐이고, 배경 물체는 인터넷 익스플로어 아이콘과 윈도우 XP의 배경 화면이다. 프레임버퍼 영상을 확대한 영상에서 글자들은 변형을 보이는데, 이것은 1024×768 크기의 프로젝션 화면의 720×480 크기의 카메라 영상으로의 변형과 카메라의 낮은 해상도 때문이다. 전경물체인 책에서도 배경 영역이 나타나게 되는데(그림 15)(c), 이것은 책의 색상과 프로젝터 화면의 색상이 다를 때 발생하는 현상이다. 노이즈를 제거하기 위해 모폴로지 연산은 클로징, 오픈링연산을 3×3 사각형 마스크를 이용하여 수행하였다. 이후 연결 성분 분석 시에 다음의 3개의 규칙을 사용하여 잡음을 걸러 내었다. 연결 성분들은 다음 조건중에 하나라도 만족하면 전경물체가 된다.

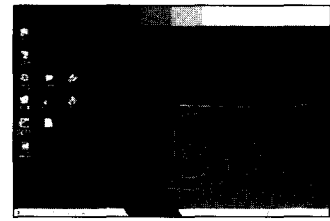
- ① 300 픽셀 크기 이상의 연결요소
- ② 종횡비가 30%에서 300%인 연결요소
- ③ 40% 이상의 픽셀 밀도를 가진 연결요소



(a) 프레임버퍼 영상



(b) 카메라 영상



(c) 기하보정 영상

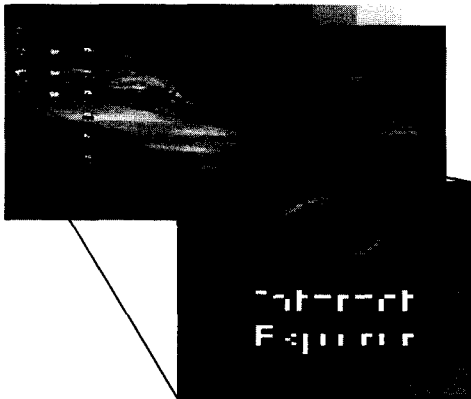


(d) (a)와 (b)의 차이 영상

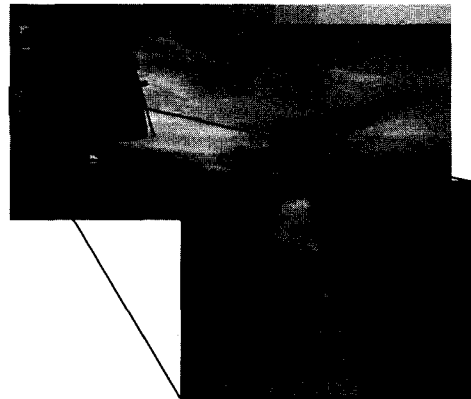


(e) 제안된 방법의 결과 영상

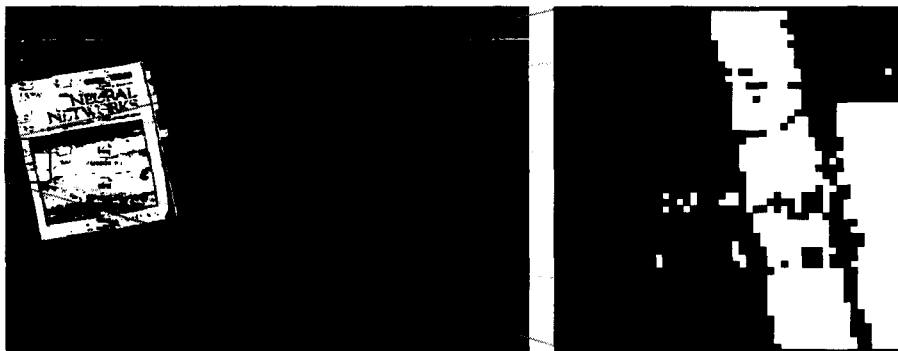
(그림 14) 어두운 배경에서 전경물체 검출의 수행결과



(a) 프레임 버퍼영상을 확대한 영상



(b) 카메라 영상을 확대한 영상

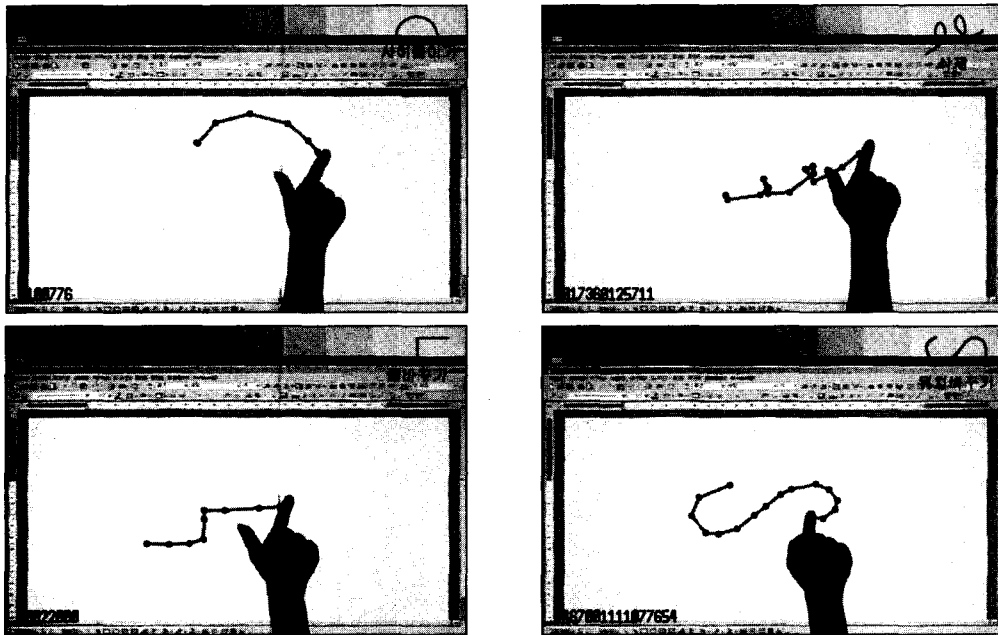


(c) 전경물체 검출 영상의 확대한 영상(전경물체는 흰색)

(그림 15) 입력영상과 전경물체 검출영상의 일부분을 확대한 영상

5.3 제스처 인식의 수행 결과
 각 제스처의 모델을 만들기 위해 프로젝션 화면에서 20명 각

각의 12개의 제스처에 대해 20번씩 입력받은 제스처 데이터베이스를 사용하였다. 각 제스처의 테스트는 10명의 사람이 12개의



(그림 16) 제스처 인식의 수행결과의 예

제스처를 다른 크기와 길이로 여러 번 수행하여 테스트하였다. (그림 16)은 제스처 인식 단계의 수행 결과의 예로써, 손가락 끝 이동경로를 보여주고 이동경로에 대한 심볼열은 영상의 좌측 하단에, 인식된 문서교정 제스처는 우측 상단에 출력하였다.

전체 시스템의 수행시간은 평균적으로 프레임 당 기하보정은 5ms, 색상보정은 20ms, 그리고 후처리를 포함한 전경물체 검출은 15ms이다. 또한, 검출한 전경 물체에서 손가락 끝의 이동경로의 추출과 인식에 걸리는 시간은 30ms이다. 제스처 인식기의 성능을 평가하기 위해 검출비(detection ratio)와 신뢰도(reliability)를 성능평가의 척도로 사용한다. 사용자의 연속 제스처를 인식할 때, 제스처가 아닌 패턴을 제스처라고 하는 삽입에러(insertion error), 제스처를 찾지 못하는 삭제에러(deletion error), 다른 제스처와 혼동하는 대치에러(substitution error)와 같은 세 가지 형태의 에러가 나타난다[14]. 삽입에러나 삭제에러는 사용자의 손가락 개수정보를 이용하는 제스처 생성단계의 오류로 나타난 것이고, 대치에러는 제스처 인식단계의 오류로 나타난 것이다. 검출비는 식 (7)과 같이 측정한다[14].

$$\text{Detection ratio} = \frac{\text{correctly recognized gestures}}{\text{input gestures}} \quad (7)$$

여기서 삽입에러는 검출비를 측정하는데 고려되지 않는다. 그러나, 삽입에러는 삭제 또는 대치에러를 야기시키는 원인이 될 수 있다. 따라서, 정확한 제스처 인식의 성능을 평가하기 위해서는 식 (8)과 같은 삽입에러를 고려한 신뢰도를 측정하였다[14].

$$\text{Reliability} = \frac{\text{correctly recognized gestures}}{\text{input gestures} + \text{insertion errors}} \quad (8)$$

결론적으로, 제안된 방법은 <표 2>와 같은 결과를 보여주었다. 실험 결과 제스처 인식에서는 95.03%의 높은 검출비와 93.22%의 높은 신뢰도를 보였다.

<표 2> 제스처의 인식 결과

제스처	입력 제스처 수	인식 제스처 수	에러 타입			검출비	신뢰도
			삽입	삭제	대체		
사이띄우기	185	175	3	4	6	94.59	93.09
사이붙이기	195	186	3	4	5	95.38	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
줄바꾸기	198	191	2	2	5	96.46	95.50
줄잇기	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
줄삽입	200	194	3	4	2	97.00	95.57
들여쓰기	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

6. 결 론

본 논문에서는 프로젝션 화면 상의 상호작용 시스템인 '리바이저'를 제안하였다. 리바이저는 프로젝션 화면에서 사용자에게 워드프로세서의 문서를 보여주고 실제 문서를 교정하는 것처럼 사용자의 손 제스처로 디지털 문서를 교정하는 시스템이다.

제안된 전경물체 검출방법은 이전 연구들과 달리 색상정

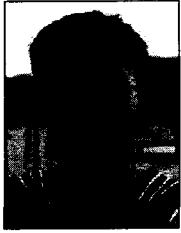
보를 사용하는 방법으로, 기하 보정과 색상 보정을 수행함으로써 추정된 색상변화 정보를 사용하여, 프로젝션 화면에서 전경물체를 검출하고, 추출된 사용자의 손가락 끝의 이동경로들을 입력받아 HMM을 사용하여 문서교정 제스처를 인식하였다.

실험 결과, 제안된 전경물체 검출방법은 복잡한 배경이나 다양한 빛의 변화를 가지는 프로젝션 화면에서 사용자의 손이나 실제 문서 같은 전경물체를 검출하는데 효율적인 결과를 보였다. 그러나, 카메라의 제한된 해상도와 프로젝터의 왜곡 때문에 에러가 발생하였다. 제스처 인식 단계의 결과는 테스트 영상 시퀀스에서 93.22%의 신뢰도를 보였다.

그러나 아직 제안된 시스템에는 처리해야 할 문제점이 있다. 카메라 해상도 때문에 워드프로세서의 글자들을 읽을 수가 없었고, 그 때문에, 전경물체 검출에서 에러가 발생하였다. 또한, 제스처 인식에서의 많은 계산량으로 인해 빠른 응답 속도를 가지는 입력장치로 만들지 못하였다. 향후 과제는 전경물체 검출에서 밝기 마커를 사용하지 않고 전경물체를 검출하는 방법과 사용자의 움직임으로부터 더욱 정확한 제스처 추출에 관한 연구를 하는 것이다.

참 고 문 헌

- [1] M. Ashdown, P. Robinson, "The Escritoire : A Personal Projected Display," *Journal of WSCG*, Vol.11, No.1, pp. 33-40, 2003.
- [2] M. Black, A. Jepson, "Recognition Temporal Trajectories using the Condensation Algorithm," *IEEE International Conference on Automatic Face and Gesture Recognition*, Japan, pp.16-21, 1998.
- [3] M. H. Coen, "Design Principles for Intelligent Environments," *Fifteenth National Conference on Artificial Intelligence*, (AAAI'98), Madison, WI, 1998.
- [4] J. Davis, M. Shah, "Visual gesture recognition," *Vision Image and Signal Processing*, Vol.141, No.2, pp.101-106, 1994.
- [5] Elliot, M. A. Hearst, "A Comparison of the Affordances of a Digital Desk and Tablet for Architectural Image Tasks," *International Journal Human-Computer Studies*, Vol.56, pp.173-197, 2002.
- [6] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2001.
- [7] <http://my.netian.com/~kimbdo/cos/munsil/munsil-41.htm>.
- [8] X. D. Huang, Y. Ariki, M. A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh Univ. Press, 1990.
- [9] S. Iba, J. M. V. Weghe, C. J. J. Paredis, P. K. Khosla, "An Architecture for Gesture-based Control of Mobile Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2, pp.851-857, 1999.
- [10] B. Johanson, T. Winograd, A. Fox, "Interactive Workspaces," *IEEE Computer*, Vol.36, pp.99-101, 2003.
- [11] R. Kjeldsen, J. Kender, "Visual Hand Gesture Recognition for Window System Control," *Proceedings on International Workshop on Automatic Face - and Gesture - Recognition (IWAAGR)*, pp.184-188, 1995.
- [12] R. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, T. Levas, M. Podlaseck, "Interacting with Steerable Projected Displays," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp.20-21, May, 2002.
- [13] D. C. Lay, *Linear Algebra and Its Applications*, Addison Wesley, 1994.
- [14] H. K. Lee, J. H. Kim, "An HMM-based Threshold Model Approach for Gesture Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.10, pp.961-973, 1999.
- [15] K. Oka, Y. Sato, H. Koike, "Real-Time Fingertip Tracking and Gesture Recognition," *IEEE Computer Graphics and Applications*, pp.64-71, 2002.
- [16] V. I. Pavlovic, R. Sharma, T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction : a Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.677-695, 1997.
- [17] F. Quek, "Toward a Vision-based Human Gesture Interface," *International Conference on Virtual Reality Software and Technology*, pp.17-31, 1994.
- [18] J. Rekimoto, "SmartSkin : An Infrastructure for Freehand Manipulation on Interactive Surfaces," *CHI 2002*, April, 2002.
- [19] G. Sharma, M. J. Vrhel, H. J. Trussell, "Color Imaging for Multimedia," *Proceedings of the IEEE*, Vol.86, No.6, June, 1998.
- [20] J. Q. Stafford-Fraser, P. Robinson, "BrightBoard : A Video-Augmented Environment," *CHI 1996*, ACM, pp.134-141, 1996.
- [21] T. Starner, A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models," *Technical Report TR-375*, Media Lab, MIT, 1995.
- [22] E. H. Stupp, M. S. Brennessoltz, *Projection Display*, John Wiley & Son, 1999.
- [23] R. Sukthankar, R. G. Stockton, M. D. Mullin, "Smarter Presentations : Exploiting Homography in Camera-Projector System," *International Conference on Computer Vision*, pp.247-253, 2001.
- [24] P. Wellner, "The DigitalDesk Calculator : Tactile Manipulation on a Desk Top Display," *ACM Symposium on User Interface Software and Technology (UIST '91)*, pp.27-33, 1991.
- [25] P. Wellner, "Self Calibration for the DigitalDesk," *EuroPARC Technical Report EPC-93-109*, 1993.
- [26] H. Kang, H. J. Kim, "Design of an Interface on PDA for Korean," *IEEE Transactions on Consumer Electronics*, Vol. 46, No.3, pp.834-838, Aug., 2000.
- [27] Whitaker, J., Benson, B., *Standard Handbook of Video and Television Engineering*, McGraw-Hill, 2000.
- [28] <http://www.intel.com/research/mrl/research/opencv/>.



정 기 철

e-mail : kcjung@ssu.ac.kr

1994년 경북대학교 컴퓨터공학과(공학사)

1996년 경북대학교 대학원 컴퓨터공학과
(공학석사)

1999년 Intelligent User Interfaces group
at DFKI(The German Research
Center for Artificial Intelligence,
GmbH), Germany, 방문연구원

2000년 Machine Understanding Division, ElectroTechnical
Laboratory in Japan, 방문연구원

2000년 경북대학교 대학원 컴퓨터공학과(공학박사)

2000년~2001년 성균관대학교 박사후연구원

2001년~2002년 PRIP Lab, Michigan State University Postdoc

2003년~현재 숭실대학교 정보과학대학 미디어학부 교수

관심분야 : 컴퓨터비전, HCI, 영상처리, 증강현실, 인공지능



강 현

e-mail : hkang@etri.re.kr

1999년 경북대학교 대학원 컴퓨터공학과
(공학석사)

2004년 경북대학교 대학원 컴퓨터공학과
(공학박사)

2000년~2002년 (주)네비웍스 인공지능
연구소 책임연구원

2004년~현재 한국전자통신연구원 디지털콘텐츠단 연구원

관심분야 : 컴퓨터 비전, 증강 현실, HCI, 게임