

인간-컴퓨터 상호작용을 위한 신경망 알고리즘기반 레이저포인터 검출[†]

(Laser pointer detection using neural network for
human computer interaction)

정 찬 웅*, 정 성 문**, 이 민 호***

(Chanwoong Jung, Sungmoon Jeong, and Minho Lee)

요 약 본 논문에서는 인간-컴퓨터 상호작용 시스템 구현을 위해 신경망 알고리즘을 적용하여 스크린 상의 레이저포인터를 효과적으로 검출하는 방법을 제안하였다. 제안된 신경망 알고리즘은 먼저 레이저포인터가 없는 입력 카메라 영상의 패치들을 학습시킨다. 학습된 신경망은 카메라 영상으로부터 얻은 입력 패치에 대해 출력 값을 발생시킨다. 만약 미세한 레이저빔의 변화가 입력영상에 감지되면 이를 증폭시켜 레이저 빔을 검출하는 역할을 한다. 제안된 시스템은 레이저포인터,싼 가격의 웹캠 그리고 영상처리 프로그램으로 구현할 수 있다. 그리고 레이저포인터와 컴퓨터의 배경화면 색상이 유사한 경우에도 레이저포인터를 검출할 수 있는 장점이 있으므로 인간-컴퓨터 상호작용 시스템의 성능개선에 기여할 것이다.

핵심주제어 : Laser pointer 인터페이스, 신경망, 인간-컴퓨터 상호작용

Abstract In this paper, an effective method to detect the laser pointer on the screen using the neural network algorithm for implementing the human-computer interaction system. The proposed neural network algorithm is used to train the patches without a laser pointer from the input camera images, the trained neural network then generates output values for an input patch from a camera image. If a small variation is perceived in the input camera image, amplify the small variations and detect the laser pointer spot in the camera image. The proposed system consists of a laser pointer, low-price web-camera and image processing program and has a detection capability of laser spot even if the background of computer monitor has a similar color with the laser pointer spot. Therefore, the proposed technique will be contributed to improve the performance of human-computer interaction system

Key Words : Laser pointer, Neural network, Interface, HCI(Human-computer interaction)

1. 서 론

최근 마우스, 키보드, 터치스크린 그리고 게임 콘솔 등의 기존 인터페이스 시스템을 대신하여 다양한 센서를 이용한 새로운 인터페이스 시스템이 개발되고 있다[1~5]. 닌텐도사의 Wii Remote는 게임 컨트롤러로써 개발된 인터페이스 장치로써 모션 감지기능으로 사용자가 화면상의 출력되는 객체와 상호 작용하며

[†] 이 논문은 2009년 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임. (No. 20090304104140)

* 경북대학교 센서및디스플레이공학과, 제1저자

** 경북대학교 전자전기컴퓨터공학부, 제2저자

*** 경북대학교 전자공학부, 교신저자

객체를 이용할 수 있게 하며 광학 센서와 가속도계를 사용하여 포인트 할 수 있게 되어있다. 마이크로소프트社は 프로젝트 Natal을 통해 가정용 콘솔 게임기 Xbox360에 사용자의 동작 및 음성을 인식하여 특별한 게임 컨트롤러를 사용하지 않고, 순수하게 몸짓과 음성으로 게임 플레이와 다양한 기능 활용이 가능한 인터페이스 장치로써 여러 센서를 사용하여 개발되고 있다. 특히 사용자의 동작을 인식하려고 Natal은 심도 (Depth) 센서를 채택하여 센서로부터 적외선 빛의 펄스를 반사해 카메라에서 전방 물체의 형상을 감지하며 빛이 돌아올 때까지의 시간을 측정하고 카메라와 물체와의 위치 및 거리를 계산하여 동작하는 방식으로 개발되고 있다. 게임 인터페이스뿐만 아니라 컴퓨터 인터페이스 분야에서도 이러한 연구가 계속되고 있는데 MIT Media Lab의 sixth sense Technology[1]는 카메라와 빔 프로젝터, 무선 인터넷 그리고 손가락에 끼는 골무를 이용하여 터치스크린, 마우스, 키보드를 대신하여 컴퓨터나 휴대전화를 조작할 수 있는 새로운 인터페이스를 개발하였다. 또한 음성 인식, 눈동자 인식, 타블렛, 터치스크린 등의 인터랙션 시스템의 개발을 통해 사용자가 더욱 편리하게 컴퓨터를 다룰 수 있는 연구가 활발하다.

이런 다양한 인터페이스 장치 개발의 한 분야로써 레이저 포인터를 이용한 인터페이스 시스템의 개발이 이루어지고 있다. 영상처리와 다양한 센서의 특징들을 기반으로 현재 레이저포인터와 웹 카메라를 이용하여 일반적인 컴퓨터 조작 환경이나 프레젠테이션 환경에서 사용가능한 인터페이스 시스템의 개발이 활발하다[2~5]. Dan R. Olsen and Travis Nielsen은 프레젠테이션 환경에서 사용자가 매번 스크린에서 컴퓨터로 이동하여 프레젠테이션 화면을 조작하는 불편함을 없애고자 프레젠테이션 환경에 맞는 레이저포인터 인터랙션 시스템을 제안하였다[2]. 개발된 시스템은 XWeb 시스템을 기반으로 하여 커서 모델에 따른 패턴 분석으로 작동한다. 이는 매번 인터랙션을 위하여 사용자가 반복적으로 UI (User interface)를 발생시켜 원하는 커서모델에 레이저포인터가 위치하도록 조작해야 하는 불편함과 사용자가 원하는 동작을 빨리 수행하지 못하는 문제점이 있다. B. Shizuki, et al.은 스크린상의 레이저 포인터가 스크린 영역의 네 개 변을

기준으로 영역 안과 밖으로 움직이는 동작을 파악하여 인터랙션을 발생시키는 시스템을 개발하였다[3]. 하지만 영역 내, 외부 이동에 따른 동작이기 때문에 다양한 응용이 힘든 단점이 있다. 김남우 등은 앞서 언급한 문제점을 보완하여 레이저포인터 인터랙션 시스템을 개발하였다. 이는 본 논문에서 제안한 시스템과 가장 유사한 형태로써, 프레젠테이션 환경뿐만 아니라 다양한 컴퓨터 동작 상황을 제어할 수 있는 기반을 만들었다[4]. 하지만 레이저포인터 검출 방법에 영상의 명암도 (intensity)를 이용한 이진화, HSI (Hue, Saturation, Intensity)색 채널 분리, RGB (Red, Green, Blue) 색상 채널 분석을 통한 검출 방법 등을 사용하고 있어 주변 조명의 영향에 따라 명암도가 달라져 정확한 검출이 어렵거나, 레이저포인터가 비춰지는 배경영상의 색상에 영향을 받아 레이저포인터의 검출이 어려운 단점을 보이고 있다. J. G. Lim et al.은 레이저포인터 자체를 고려하기보다 레이저포인터에 부가적인 하드웨어(3축 가속도센서)를 장착[5]하여 새로운 인터페이스 장치를 개발하기 위한 연구를 하였다. 이는 이미 사용하고 있는 레이저포인터의 재사용이 어려운 문제점이 있다. 위에 설명한 다양한 연구에서 나타난 공통적인 문제점으로 레이저 포인터의 색상 검출에 대한 문제가 있다. 이는 웹캠에서 영상을 획득할 때 레이저포인터의 점이 흰 점으로 보이는 문제, 레이저포인터와 유사한 색상의 배경이거나 배경에 레이저포인터의 밝기와 유사한 밝기를 가지는 부분이 있을 때 레이저 포인터 검출을 위한 웹캠의 명암도를 사용자가 직접 조절해야 되는 문제 등 환경 영향에 따른 레이저포인터 검출 성능 저하를 해결하기 위해 보다 정확한 레이저포인터 검출에 대한 연구가 필요함을 나타낸다.

따라서 본 논문에서는 주변 환경 영향에 강건한 레이저포인터 검출을 위해 신경망 학습 알고리즘을 적용한 레이저포인터 검출방법을 제안한다. 현재까지 제안된 알고리즘과의 비교를 위해 RGB 색 채널을 이용한 레이저포인터 검출방법과 영상의 시간적 변화 분석을 통한 검출 방법을 적용한 결과를 비교하여 성능을 나타내었다.

본 논문의 구성은 다음과 같다. 2 절에서는 전체 시스템의 구성, 신경망 학습 알고리즘을 적용하기 위한

전처리 방법과 검출 방법을 설명한다. 3 절에서는 제안한 방법과 현재 많이 사용되고 있는 레이저포인터 검출 방식을 비교한 실험 결과를 보이며 끝으로 4 절에서 결론과 향후 연구에 대해 논의할 것이다.

2. 시스템 구성 및 제안 모델

2.1 시스템 구성

제안하는 시스템은 컴퓨터, 카메라, 빔 프로젝터, 스크린 그리고 레이저포인터와 같은 하드웨어 시스템과 입력된 영상을 처리하는 영상처리 소프트웨어 부분으로 구성된다. 전체 시스템의 프로세서를 제어하고 프레젠테이션이나 기타 일반적인 작업에 사용되는 컴퓨터를 중심으로 컴퓨터에 빔 프로젝터를 연결하여 영상을 스크린 상에 투영한다. 카메라 또한 컴퓨터에 연결하고, 빔 프로젝터로부터 스크린에 투영된 영상 전체를 촬영할 수 있는 충분한 거리에 설치한다. (사용하는 카메라 사양에 따라 거리가 달라질 수 있다. 본 논문의 실험에서 사용한 카메라의 경우 2m 거리에서 전체 스크린과 배경영상이 한 화면에 표현되었다.) 시스템배치를 완료한 후 레이저포인터를 이용하여 컴퓨터와 인터랙션을 발생시키기 위해 사용자가 원하는 약속된 제스처를 스크린에 지시한다. 컴퓨터에 연결된 카메라는 스크린을 계속해서 촬영하며 레이저포인터가

포함된 영상이 입력영상으로 들어오면 이를 영상처리 알고리즘을 통해 분석하여 사용자의 의도에 맞는 적절한 인터랙션을 발생시키게 된다. <그림 1>은 전체 시스템 구성을 나타낸다.

실험은 실제 프레젠테이션 환경과 유사하게 일반 데스크톱 컴퓨터가 아닌 랩톱 컴퓨터를 사용하여 구성하였다. 컴퓨터 및 실험에 사용된 하드웨어 장비의 제원은 <표 1>과 같다.

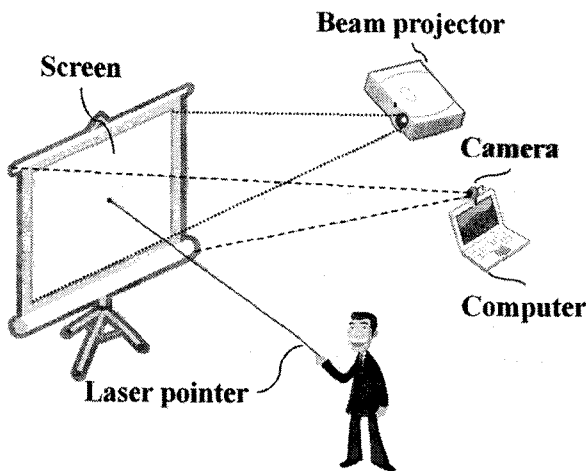
<표 1> 하드웨어 사양

웹캠	컴퓨터	레이저포인터
USB 2.0	Type: Laptop	Red:
30 f/s (최대)	CPU: 2.1GHz (코어2듀오)	630~680nm
해상도: (최대) 1280×1084	GPU: 인텔GMA 해상도: (최대) 1366×768	Green: 532nm

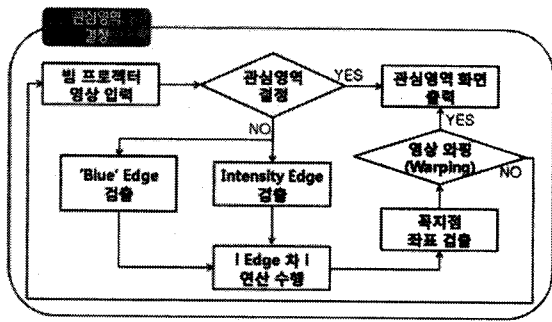
사용된 USB 타입의 웹캠은 해상도에 반비례하여 프레임 속도가 느려지는 특징이 있어 영상 크기는 320×240의 해상도를 기준으로 하였다.

2.2 관심영역(Region of interest) 결정

영상처리 과정을 수행할 때 컴퓨터의 연산량을 줄이기 위해 관심영역을 설정하였다. 이를 위해 빔 프로젝터를 켜 후 최초의 푸른색 바탕화면이 나타나면 카메라로 촬영하고 이를 바탕으로 관심영역을 결정하였다. 관심영역 결정 프로세스는 <그림 2>와 같고, 순서도의 진행에 따른 관심영역 검출 결과는 <그림 3>에 나타내었다.



<그림 1> 시스템 구성



<그림 2> 관심영역 결정 과정

빔 프로젝터가 켜진 후 최초로 나타나는 푸른색 바탕화면을 촬영하여 저장하고, 촬영된 입력영상을 세 가지 색 영상 적색(R), 녹색(G), 푸른색(B)로 분리하고, 분리한 각각의 영상 윤곽선을 Sobel 연산자를 이용하여 구하였다.

Sobel 연산자는 다른 윤곽선검출연산자보다 대각선 성분까지 잘 나타낼 수 있기 때문에 입력영상이 기울어져 있어도 정확한 경계를 검출할 수 있는 장점이 있다. Sobel 연산자는 식 (1), (2)에 나타내었다.

$$G_x = M * A \quad (1)$$

$$G_y = M^T * A \quad (2)$$

$$\text{where, } M = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

식 (1), (2)의 G_x 와 G_y 는 각각 입력영상에 수평성분, 수직성분 Sobel연산자가 컨벌루션된 결과, A 는 입력영상, M 은 Sobel연산자, $*$ 는 컨벌루션 연산자이다.

식 (1), (2)에서 컨벌루션한 최종 결과를 하나의 영상으로 나타내기 위한 연산 과정은 식 (3)과 같다.

$$G_e = \sqrt{G_x^2 + G_y^2} \quad (3)$$

위 과정을 통해 구한 RGB 각각의 경계를 모두 더해 평균을 내면 명암값 윤곽선을 구할 수 있고, 처음 구한 각 색상 채널경계 중 B영역의 경계와 영상의 차이를 통해 사용자에게 필요한 관심영역인 빔 프로젝터

에 의해 투영된 영역의 윤곽선만을 남긴다. 식 (4)에서 명암값의 윤곽선을 식 (5)에서 차 영상을 구하는 식을 나타내었다.

$$I_e = \frac{R_e + G_e + B_e}{3} \quad (4)$$

$$dE = |I_e - B_e| \quad (5)$$

식 (4)의 R_e , G_e , B_e 는 각각 컬러영상에서 분리한 RGB 영상의 윤곽선을 나타내고, 3개의 영상별 윤곽선을 모두 더해 평균을 구한 I_e 는 명암값의 윤곽선을 나타낸다. 식 (5)의 dE 는 차 연산 결과, I_e 는 명암값 윤곽선, B_e 는 B 영상의 윤곽선영상을 나타내고 두 영상의 차이를 구하여 절대값을 취하면 결과영상으로 잡음이 섞인 B 영상이 강조된 윤곽선 영상이 남게 된다.

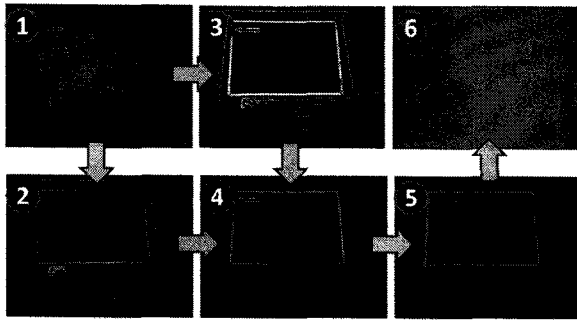
차 연산 결과로 남은 잡음을 제거하기 위해 아래 정의와 같이 임계값(T) 이하의 값을 '0'으로 그 외의 값을 '255'로 정의하여 dE 의 윤곽선을 강조하고 잡음을 제거하였다. 본 시스템에서 사용한 임계값은 레이저포인터가 없는 영상의 히스토그램을 분석하여 히스토그램의 최대값으로 결정하였다[12].

$$\begin{array}{ll} \text{If} & dE < T \\ \text{then} & dE = 0 \\ \text{else} & dE = 255 \end{array}$$

영상의 4개 꼭짓점을 구하기 위해 차 연산 결과 경계영상 전체를 검사한다. 임의의 점에서 가장 먼 점을 첫 번째 꼭짓점으로 설정하고 첫 번째 꼭짓점이 결정되면 이를 기준으로 가장 먼 점을 두 번째 꼭짓점으로 결정한다. 같은 방식으로 두 번째 꼭짓점으로부터 가장 먼 점을 세 번째 꼭짓점으로 결정하여 삼각형을 만들고 네 번째 꼭짓점을 찾기 위해 벡터 내적을 이용하여 좌표평면에서 사각형 넓이가 최대가 되는 꼭짓점을 결정하여 4개의 전체 꼭짓점을 찾아낸다. 찾아낸 꼭짓점에 영상 왜핑 알고리즘을 적용하여 관심영역을 결정 하였다[6] [10].

2.3 레이저포인터 검출

레이저포인터 검출은 색상분석을 통한 검출 방법과



<그림 3> 관심영역 설정 (1) 원 영상, (2) Blue 채널 edge영상, (3) 명암도 edge영상, (4) 명암도 edge 영상과 blue 채널 edge 영상의 차이, (5) 관심영역 외곽선 영상, (6) 외팽된 결과 영상

차 연산을 이용한 검출 결과를 기본으로 사용한다. 본 논문에서 제안하는 신경망을 적용한 레이저포인터 검출 방법은 영상의 배경색과 레이저포인터의 색깔이 유사하거나 배경이 매우 밝아 레이저포인터가 된 점이 육안으로도 구분이 안 될 정도인 부분에 대해서 적용하여 전체적인 레이저포인터의 검출 성능을 높였다.

2.3.1 색상분석방법을 이용한 레이저포인터 검출

앞서 관심영역을 설정하기 위해 입력 영상을 세 가지 색 영상으로 분리하였다. 이를 이용하여 각 색 영상에서 레이저포인터가 어떻게 표시되는지 분석하였다. 사용되는 레이저포인터 Laser Diode (LD)의 색상이 R과 G이기 때문에 색 영상 중 R과 G색상영상을 분석하면 각각 R과 G 색상을 가진 레이저포인터를 찾을 수 있게 된다. 또한 R과 G색상영상을 평활화 하여 Real Red(Real R)과 Real Green(Real G)의 색상정보를 추출하였다. 이를 수식화 하면 식 (6)과 같다 [7][11].

$$\begin{aligned}
 \text{Real R} &= \begin{cases} R - \frac{G+B}{2}, & \text{Real R} > 0 \\ 0, & \text{Real R} \leq 0 \end{cases} \\
 \text{Real G} &= \begin{cases} G - \frac{R+B}{2}, & \text{Real G} > 0 \\ 0, & \text{Real G} \leq 0 \end{cases}
 \end{aligned} \quad (6)$$

각 R, G, Real R, Real G의 색상정보를 구한 후 명암영상의 히스토그램에서 가장 밝은 값을 임계값으로

정해 이진화를 하면 레이저포인터의 위치를 구할 수 있다. 위 이진영상의 픽셀 평균을 구하면 최종적인 레이저포인터의 위치와 화면상의 좌표를 검출할 수 있다.

2.3.2 영상의 차이를 이용한 레이저포인터 검출

색상분석방법을 이용하여 레이저포인터를 검출할 때, 배경색상의 유사도나 스크린에 비춰지는 빛의 영향 그리고 사용된 영상 자체의 밝기가 매우 밝아 레이저포인터가 검출되지 않을 경우가 있어 현재 영상과 이전영상과의 차이를 통해 레이저포인터를 검출하였다. 즉 레이저포인터의 시간적 변화를 봄으로써 레이저 포인터를 찾고 그 위치를 파악할 수 있는 방법이며, 과정은 식 (7)과 같다.

$$\text{Img}_d = \text{Img}_t - \text{Img}_{t-1} \quad (7)$$

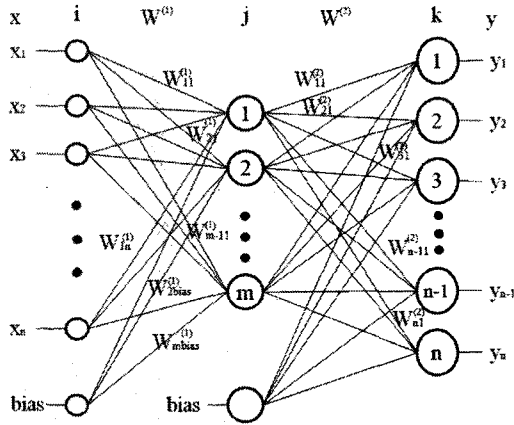
Img_d 는 차 연산 결과를 나타내고 Img_t 는 현재영상, Img_{t-1} 은 이전영상을 나타낸다. 제안한 영상의 차이를 이용한 레이저포인터 검출방법은 색상분석을 통한 검출방법에 비해 더욱 정확하게 레이저포인터의 검출이 가능하지만 여전히 색상분석방법과 같은 문제점을 가지고 있다. 또한 레이저포인터가 한 점을 비춰 움직임이 없을 때 레이저포인터가 검출되지 않는 문제점이 있다.

2.3.3 신경망 알고리즘을 적용한 레이저포인터 검출

제안하는 방법은 신경망을 적용하여 레이저 포인터가 없는 영상을 입력영상으로, 모니터의 배경영상을 목표영상으로 하여 학습하였다. 사용한 학습 알고리즘은 역전파(Error Back propagation) 알고리즘이다 [8~9][13].

역전파 알고리즘은 입력층의 각 마디에 입력패턴 영상을 넣으면 이 신호는 중간층에 전달되고 다시 출력층으로 전달되어 최종 신호를 출력하게 된다. 이 출력값과 목표영상값(기대값)을 비교하여 차이를 줄여나가는 방향으로 연결강도를 조절하고, 상위층에서 역전파하여 하위층에서 이를 근거로 다시 자기층의 연결

강도를 조절하도록 하는 학습 알고리즘이다. <그림 4>는 구성된 신경망을 나타낸다.



<그림 4> 신경망 구조도

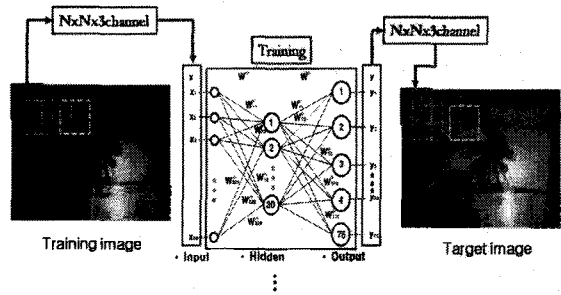
<그림 4>에서 \$x\$는 입력, \$y\$는 출력이고 1개의 은닉층을 가진 2층 구조를 가진 신경망이다. \$m\$과 \$n\$은 각각 은닉층과 출력노드의 개수를 나타내며 입력층과 은닉층은 bias노드를 포함한다. 또한 \$I(x)\$는 입력층, \$j\$는 은닉층, \$k(y)\$는 출력층을 나타내고, \$W^{(1)}\$은 \$W_{ji}\$, \$W^{(2)}\$는 \$W_{kj}\$로써 각층의 연결강도를 나타낸다. 이를 식으로 나타내면 다음과 같다.

$$\Delta_p W_{ji} = \eta(t_{pj} - y_{pj})i_{pi} = \eta \delta_{pj} i_{pi} \quad (8)$$

\$t_{pj}\$는 \$p\$번째 목표출력 패턴의 \$j\$성분, \$y_{pj}\$는 \$p\$번째 입력패턴으로부터 네트워크가 계산한 출력의 \$j\$성분. \$\delta_{pj} = t_{pj} - y_{pj}\$는 목표출력과 실제출력의 차 (오차), \$\Delta_p W_{ji}\$는 입력층 \$i\$로부터 출력층 \$j\$에서 연결강도의 변화량이다.

신경망에 사용한 입력과 실험영상은 \$320 \times 240\$의 영상을 \$5 \times 5\$ 크기의 작은 영상 사이즈로 잘라 사용하였다. \$5 \times 5\$ 크기의 영상을 사용한 것은 영상 전체를 입력으로 사용하게 되면 학습시간이 오래 걸리고, 레이저포인터의 정확한 검출이 어려운 것을 실험을 통해 확인하였기 때문이다. 또한 영상의 손실을 막기 위해 전체영상에서 \$5 \times 5\$ 크기의 영상을 1픽셀씩 이동하며 중복되게 잘라 사용하였다. 목표영상 또한 입력영상의 패턴과 일치시키기 위해 입력영상을 편집한 방법과

같은 방법으로 크기를 맞추어 사용하였다. 위 설명에 따라 각각 학습입력영상 100장, 목표영상 1장, 실험영상은 10장을 1개의 실험데이터집합으로 만들어 사용하였다. 실험영상의 각 데이터집합은 9장의 일반영상 (입력영상과 다른 시간에 수집한 같은 배경의 영상), 1장의 레이저포인터가 있는 영상으로 구성하였다. 목표영상이 1장인 이유는 100장의 학습입력영상이 1장의 목표영상과 유사한 결과를 만들어 낼 수 있는 학습을 통해 실험영상이 입력으로 주어졌을 때 목표영상과 실험영상간의 차이를 알 수 있도록 학습하기 때문이다. <그림 5>는 신경망 학습방식의 전체적인 구성도를 나타낸다.



<그림 5> 신경망 학습 방식

<그림 5>의 '\$N\$'은 마스크의 가로 및 세로의 크기이며, \$N \times N\$ 크기의 영상에서 RGB로 분리한 영상의 픽셀 정보 전체와 바이어스 1개를 포함하여 입력을 결정하였다. 출력은 목표 영상과의 비교를 용이하게 하기 위해 바이어스를 제외한 전체 픽셀정보 개수로 결정하였다.

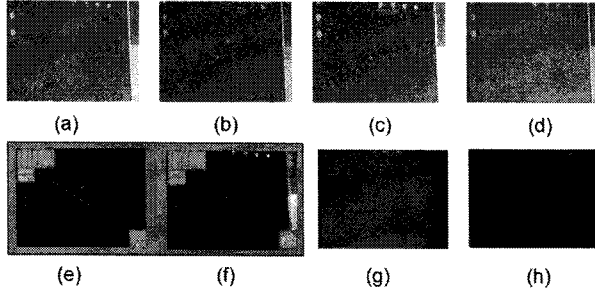
3. 결과 및 고찰

3.1 색상분석방법 레이저포인터 검출 결과

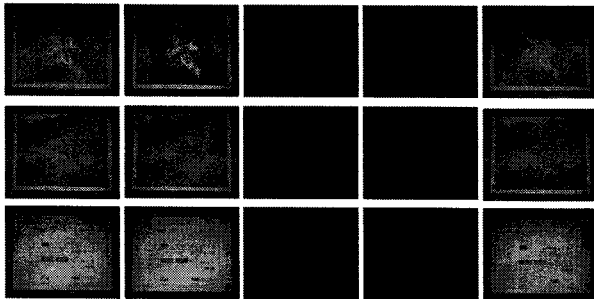
<그림 6>은 색상분석을 통한 레이저포인터 검출 결과를 보이고, <그림 7>은 다양한 배경에서의 레이저포인터 검출 결과를 보인 결과 영상이다.

그러나 배경과 레이저포인터의 색상이 유사하여 배경에서 레이저 포인터가 두드러져 보이지 않는 경우, 카메라가 조명의 영향에 의해 전체 영상을 외곡 되게

받아들이는 경우, 스크린이 레이저 포인터의 빛을 흡수하는 경우 등 다양한 환경 영향에 의해 레이저포인터가 검출되지 않을 수 있다. 따라서 이를 개선하여 환경에 적응적으로 동작할 수 있는 시스템의 개발이 요구된다.



<그림 6> 컬러 영상 분석 (a) 입력 영상, (b) R채널, (c) G채널, (d) B채널, (e) Real R 영상, (f) Real G 영상, (g) Real B 영상, (h) Real Y 영상

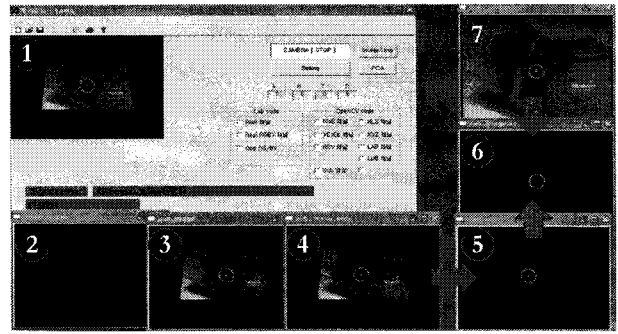


<그림 7> 복잡한 배경영상에서의 레이저포인터 검출 결과 (a) 입력 영상, (b) Gray 영상, (c) Real R 영상, (d) Real G 영상 (e) 출력 영상

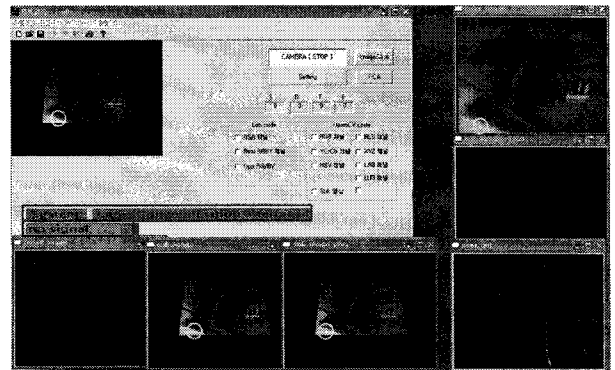
3.2 영상의 차이를 통한 레이저포인터 검출

스크린 화면에서 레이저포인터가 나타날 때, 차 연산을 통한 레이저포인터 검출결과는 <그림 8>과 같다.

그러나 배경의 명도가 높은 경우, 레이저포인터의 검출이 되지 않음을 <그림 9>에서 볼 수 있다. 또한 레이저포인터를 비춘 후 움직임이 없으면 시간에 따른 레이저포인터의 위치 변화를 검출할 수 없음을 실험을 통해 확인하였다.



<그림 8> 차 연산을 통한 레이저포인터 검출 (1) 실행화면, (2) 관심영역 영상(Gray), (3) 현재영상, (4) 이전 영상, (5) (3), (4)번의 차 연산결과, (6) 레이저포인터 위치, (7) 관심영역 영상(Color)



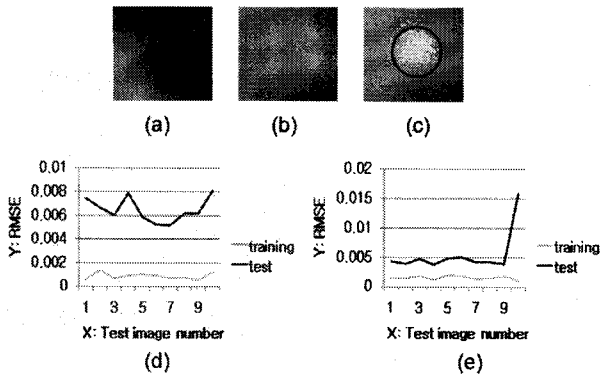
<그림 9> Red 레이저포인터 검출 실패 결과

이와 같은 색상분석방법과 차 연산을 통한 레이저포인터검출방법의 문제점을 해결하기위해 신경망을 적용하여 레이저포인터 검출을 수행하였다.

3.3 신경망 적용 레이저포인터 검출 결과

앞서 두 가지 방법의 결과를 바탕으로 레이저 포인터 검출이 어려운 부분에 대해 실험을 실시하고, <그림 10>은 그 실험 결과를 나타낸다.

신경망의 구성에 따라 결과 값이 다르게 나오는 문제가 있기 때문에 여러 번의 실험을 통해 신경망의 은닉노드의 개수, 학습률, 학습을 위한 반복횟수를 결정하였다. 최종적으로 신경망의 구성은 은닉노드 30개, 반복횟수 30,000번, 학습률 0.003으로 결정하여 실험을 수행 하였다.

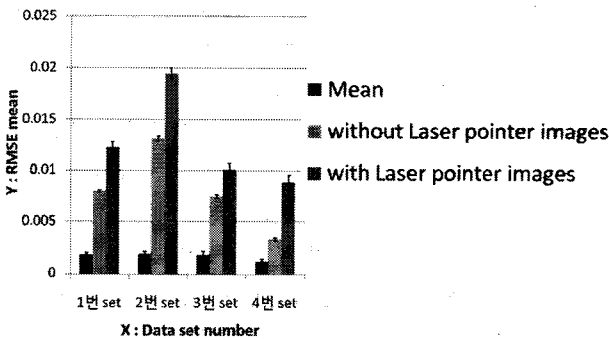


<그림 10> 신경망을 적용한 레이저포인터 검출결과 (a) 목표영상, (b) 학습영상, (c) 실험영상, (d) 10개의 은닉노드를 사용한 결과, (e) 30개의 은닉노드를 사용한 결과

<그림 10>의 실험결과 그래프 (d)와 (e)는 레이저 포인터 검출을 판단하기 위해 실험 영상을 통해 나온 결과와 목표영상과의 차이를 이용하여 오차의 크기를 구한 RMSE (Root Mean Square Error) 이다. 또한 레이저포인터 검출의 정확도를 파악하기 위해 실험 결과를 정리한 <표 2>를 <그림 11>의 오차막대 그래프로 나타내었다.

<표 2> 신경망의 error 값 (unit: 10^{-03})

D	Learning		Test			
	M	S	without Laser pointer images		with Laser pointer images	
			M	S	M	S
1	1.83	0.226	7.97	0.114	12.2	0.555
2	1.90	0.242	1.31	0.248	19.4	0.583
3	1.81	0.305	7.42	0.166	10.1	0.614
4	1.14	0.242	3.31	0.135	8.87	0.665



<그림 11> 오차막대 그래프

<표 2>의 D는 Data set, M은 RMSE의 Mean, S는 RMSE Mean의 Standard deviation을 나타낸다. 데이터 집합은 Learning 각 10개씩, Test 각 5개씩의 데이터를 이용하여 결과를 나타내었다. without Laser pointer images는 레이저포인터가 없는 실험의 평균 RMSE, with Laser pointer images는 레이저포인터가 있는 실험의 평균 RMSE이며 <그림 11>의 오차막대는 표준편차를 이용하여 나타내었다. <그림 11>의 X축은 데이터 집합 번호, Y축은 RMSE 평균을 나타낸다. 앞서 설명한 것과 같이 영상에서 레이저포인터 검출이 제대로 이루어지지 않는 부분을 대상으로 실험을 하였기 때문에 본 실험 결과를 통해 각 데이터 집합에서 신경망을 적용한 본 알고리즘이 제대로 동작하며 그 성능 또한 높음을 확인할 수 있다.

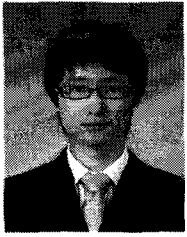
4. 결론

본 논문은 다양한 환경에서 강건한 레이저 포인터 검출을 통해 보다 실용적인 환경에 적응적인 인간-컴퓨터 상호작용 시스템을 제안하였다. 레이저포인터에서 나오는 빔을 검출하기 위해 신경망 알고리즘을 적용하였으며, 실험결과 레이저 포인터와 배경색이 유사한 부분과 조명 영향을 받은 부분에 대해서 제안한 방법이 다른 두 가지 방법에 비해 높은 레이저포인터 검출 성능을 보임을 알 수 있다. 또한 본 연구를 통한 레이저 포인터 검출방법은 환경에 강건하며 다양한 배경에 적용할 수 있는 장점이 있음을 보였다. 이는 기존의 연구에서 실험을 위해 조명을 조절하거나 웹 카메라의 설정을 조절하는 번거로움을 줄일 수 있어 보다 실용적인 시스템을 개발할 수 있는 바탕이 될 수 있다.

앞으로 본 논문에서 제안한 방법을 개선하기 위해 신경망을 통한 검출을 더욱 빠르게 할 수 있는 알고리즘의 개발을 통해 실시간으로 연동할 수 있는 인간-컴퓨터 상호작용 시스템으로의 연구가 계속적으로 수행 되어야 한다.

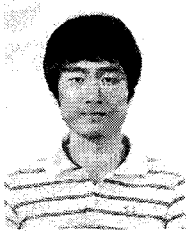
참 고 문 헌

- [1] P. Mistry, P. Maes, L. Chang, "WUW-Wear Ur World-A Wearable Gestural Interface," Proc. of SIGCHI conf. Human factors in computing systems, pp. 4111-4116, 2009.
- [2] D. R. Olsen Jr and T. Nielsen, "Laser pointer interaction," Proc. of SIGCHI conf. Human factors in computing systems, pp. 17-22, 2001.
- [3] B. Shizuki, T. Hisamatsu, S. Takahashi, J. Tanaka, "Laser Pointer Interaction Techniques using Peripheral Areas of Screens," Proc. of the working conference on Advanced visual interfaces, pp. 95-98, Venezia, Italy, 2006.
- [4] 김남우, 이승재, 이준재, 이병국, "영상처리 기반의 레이저 포인터 인터랙션 시스템," 멀티미디어학회지, 제11권, 제3호, pp. 373-385, 2008.
- [5] J. G. Lim, F. Sharifi, D. S. Kwon, "Fast and Reliable Camera-tracked Laser Pointer System Designed Audience," 5th International Ubiquitous Robots and Ambient Intelligence, pp. 529-534, 2008.
- [6] OpenCV
(<http://www.intel.com/technology/computing/opencv/overview.htm>)
- [7] S. B. Choi, B. S. Jung, S. W. Ban, H. Niitsuma, and M Lee, "Biologically Motivated vergence Control System Using Human-like Selective Attention Model," Neurocomputing, vol. 69, pp. 537-558, 2006.
- [8] S. Haykin, Neural Networks: a comprehensive foundation second edition, Englewood Cliffs, NJ: Prentice Hall International, 1998.
- [9] C. Jung, S. W. Ban, S. Jeong, and M Lee, "Input and Output Mapping Sensitive Auto-Associative Multilayer Perceptron for Computer Interface System Based on Image Processing of Laser Pointer Spot," 17th International Conference on Neural Information Processing, pp. 185-192, 2010.
- [10] Bradski, G.R., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library, Sebastopol, CA: O'Reilly Media, 2008.
- [11] 박세현, 김태의, 권경수 "손 제스처 기반의 애완용 로봇 제어," 한국산업정보학회논문지, 제13권, 제4호, pp. 145-154, 2008.
- [12] 김인중, "카메라 기반 문서 인식을 위한 적응적 이진화," 한국산업정보학회논문지, 제12권, 제3호, pp. 132-140, 2007.
- [13] Chanwoong Jung, Minho Lee, "Laser Spot Pattern Recognition Based Computer Interface Using I/O Mapping Sensitive Neural Networks," IEEE International Conference on Consumer Electronics, pp. 874-875, 2011.



정 찬 응 (Chanwoong Jung)

- 2009년 대구가톨릭대학교 전자공학과 공학사
- 2011년 경북대학교 센서및디스플레이공학과 석사수료
- 관심분야 : 인간-컴퓨터 인터페이스, 영상 신호처리, 신경망



정 성 문 (Sungmoon Jeong)

- 2006년 경북대학교 전자전기컴퓨터공학부 공학사
- 2008년 경북대학교 전자전기컴퓨터공학부 공학석사
- 2010년 경북대학교 전자전기컴퓨터공학부 박사수료
- 관심분야 : 생물학기반 영상처리, 신경망, 신호처리



이 민 호 (Minho Lee)

- 정회원
- 1988년 경북대학교 전자공학과 공학사
- 1992년 한국과학기술원 전기및전자공학과 공학석사
- 1995년 한국과학기술원 전기및전자공학과 공학박사
- 1995년~1998년 한국해양대학교 조교수
- 1998년~현재 경북대학교 IT대학 전자공학부 교수
- 관심분야 : 영상 및 음성 신호처리, 생물학기반 선택적 주의집중, 뇌-기계 상호작용, 뇌신경정보

논문 접수일 : 2011년 03월 12일
 1차수정완료일 : 2011년 03월 25일
 게재확정일 : 2011년 03월 28일