카메라 컨트롤러를 이용한 포인터 추적 장치 개발

이용환* · 주현웅** · 송성해***

Development of Camera Controller with Pointer Tracking Unit

Yong-hwan, Lee* · Hyun-woong, Ju** · Sung-hae Song***

요 약

시청각 수업이나, 세미나 등에서 프로젝터와 함께 레이저 포인터와 마우스, 키보드를 같이 사용하고 있으나 레이저 포인터와 함께 마우스나 키보드를 사용 시 불편을 초래한다. 따라서 본 논문에서는 레이저 포인터의 활용도를 높이기 위해 레이저 포인터를 이동하면 그 위치를 정확히 추적하고 그 정보를 표시함으로서 마우스 기능까지 수행할 수 있는 하드웨어를 구현하였다. 시스템은 FPGA로 카메라 인터페이스를 구현하고 ARM을 이용한 영상처리 소프트웨어와 패턴 추적을 위한 소프트웨어를 제작하였다. 레이저 포인터의 정확한 위치 파악과 처리속도를 높이기 위해 하드웨어로 잡음 처리기를 구현하여 사용하였고 소프트웨어로 스펙트럼을 분석하였다. 또한 자주 나타나는 레이저 포인터의 패턴을 저장하여 정확한 추적이 가능하도록 하였다. 구현결과 20M거리 이내에서 98%의 인식률을 확인하였다.

ABSTRACT

Presentation with a projector and a laser pointer is widely used in seminar or conference. The function of a laser pointer in the presentation is just indicating a certain object. In this paper, to give a mouse-like function to a laser pointer, we implement a system that locates the track of a laser pointer. The system contains a FPGA that implements camera interface and noise filter. A software for ARM processor is programmed to analyze the spectrum of the captured image and track the pattern of a laser pointer with previously stored image. As a result, the tracking system could locate the position correctly most of time within 20m with 98% accuracy.

키워드

Laser Pointer, ARM, Camera Interface, Image Processing

Ⅰ. 서 론

대학이나 중고등학교에서 실시하는 시청각 수업이나 회사에서의 세미나 또는 워크숍에는 프레젠테이션을 많이 사용한다. 프레젠테이션을 이용하면 짧은 시간에 필요한 정보를 전달할 수 있다. 강사는 프레젠테이션을 수행하면서 키보드와 마우스 및 레이저 포인터를 이용하여 중요한 부분을 강조하거나 표시하면서 프레젠테이션을 수행한다. 레이저 포인터는 마우스에 비해 상대적으로 프레젠테이션에서 필요로 하는 부분을 쉽게 강조 할 수 있기 때문에 많이 사용된다. 하지만 페이지의 이동이나 특정 부분에 강조를 위하여 펜 기능을 이용하려고 할 경우에는 마우스의 역할이 필요하다. 이 경우 강사는 레이저 포인터와 마우스를 같이 제어해야 하기 때문에 번거롭고 발표 보조자가 도움을 주는 경우에도 서로 의사소통을 하여야 부드러운 진행이 된다.

최근 시판되는 프레젠테이션 보조 장치들은 페이지 이동이나 간단한 마우스 포인터 이동까

* 금오공과대학교 전자공학과(yhlee@kumoh.ac.kr)
** 금오공과대학교 전자공학과 교수

지는 지원하고 있으나 마우스와 같은 세밀한 제어는 불가능하다. 이러한 문제점을 해결하기 위하여 PC를 이용한 방법이 고안되었다. 일반적으로 많이 보급되어 있는 웹캠을 사용하여 PC에 입력된 영상의 레이저포인터 자취를 분석하고 추적하여 마우스 대신 사용가능하도록 하였다. 하지만 이 방법은 웹캠 2개를 사용해야 한다는 부담과 함께 PC에서 영상처리와 추적 함수를 포함한 프로그램을 동작시켜야 하기 때문에 CPU 부하에 많은 부담을 주고 메모리 등의 많은 자원을 사용하여 시스템의 성능을 약화시킨다[1]. 이러한 PC 자체의 자원을 사용하는 방법보다 외부에 전용 하드웨어를 설치한다면 사용 시 PC에 부하를 주지 않아 쾌적한 프레젠테이션 환경을 제공할 수 있기 때문에 이러한 장치의 개발이 필요하게 되었다.

본 논문은 영상추적 장치의 하드웨어 및 소프트웨어 개발을 목표로 하였다. 영상 처리용 카메라 인터페이스는 Altera사의 Excalibur[2]를 사용하여 구현하고 이를 ARM922T 프로그래밍을 이용하여 레이저 포인터의 패턴을 분석한다. 메모리에 저장된 패턴을 이용하여 입력된 레이저 포인터와 패턴을 비교하여 레이저 포인터를 정확하게 추적하는 시스템을 설계한다.

II. 설계과정

2.1 레이저 포인터 추적 방식

영상은 히스토그램의 형태로 표현되며 이 그래프는 영상에서 각 색상이 존재하는 수를 표현한다. 균일하게 밝은 영상에서의 히스토그램은 화소들의 수치가 높은 영역에 모여 분포하는 현상을 그림 1과 같이 볼 수 있다.

그 중에서 레이저 포인터의 패턴은 특정한 RGB의 히스토그램으로 나타나 사람의 지문과 같은 고유한 패턴으로 표현될 수 있다[3]. 각 패턴은 레이저 포인터를 사용하는 환경에 따라 큰 변화가 없으며 다만 약간의 히스토그램의 변화가 생기는 것을 그림 2를 통하여 알 수 있다.

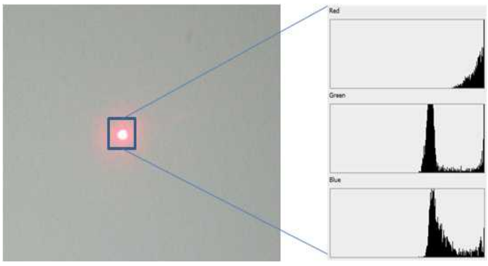


그림 1. 레이저포인터의 히스토그램
Fig. 1 Histogram of image with a laser pointer

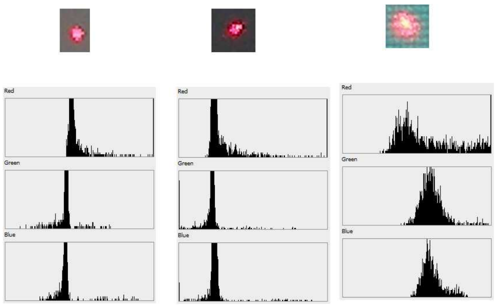


그림 2. 사용 환경에 따른 패턴의 변화
Fig. 2 Pattern change with surroundings

이러한 문제의 해결을 위하여 각 경우마다 패턴을 만들어 색상정보 체계를 구축해야 한다. 프레젠테이션을 하는 장소에 따라 밝기가 다르고 사용하는 시간대가 다르다. 여러 상황에서 미리 입력해 놓은 패턴을 사용하면 어떠한 상황에서도 레이저 포인터 패턴을 구분할 수 있다. 얻어진 모든 픽셀을 스캔하기 이전에 영상의 이진화를 거쳐 레이저 포인터의 유효영역을 검출한다. 영상과 상관없는 부분의 필터링 과정을 거치면 패턴과 상관없는 부분의 패턴 비교 부분의 수행이 줄어들게 되어 빠른 결과를 얻을 수 있는 이점이 있다. 이진화 이후 검출된 유효영역과 레이저 포인터의 패턴을 비교하여 레이저 포인터의 위치를 확인한다.

2.2 카메라 인터페이스 설계

카메라 인터페이스는 하드웨어로 설계되며 ARM 칩 외부의 FPGA로 구현된다. 영상 이진처리 및 CIS(CMOS image sensor) 카메라[4]의 내부 레지스터 설정과 카메라 영상정보를

SRAM에 저장하고 관리하는 역할을 한다. 전체적인 카메라 인터페이스의 구성은 그림 3과 같다.

카메라의 상태설정을 위하여 I^2C[5] 인터페이스를 사용한다. I^2C는 그림 4와 같은 방식으로 이루어지며 FPGA로부터 패킷을 보내고 카메라에서 받아 레지스터를 갱신한다. 카메라 해상도 설정을 위하여 I^2C인터페이스를 사용하여 입력 영상 값을 YCbCr422 8bit으로 설정한다. 신호의 입력은 그림 5와 같은 형식으로 카메라로부터 입력된다. 카메라로부터 들어오는 Y값은 두 픽셀의 값을 공유하기 때문에 출력된 YCbCr422를 RGB로 복원하기 위하여 Y0, Cb, Y1, Cr을 묶음으로 생성한다.

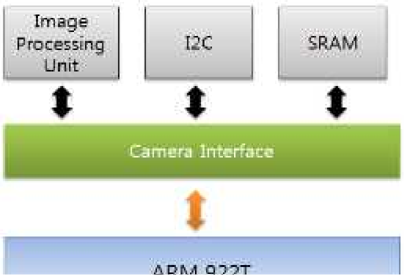


그림 3. 카메라 인터페이스의 전체 구성
Fig. 3 Block diagram of camera interface

그림 4. I^2C의 읽기/쓰기 순서
Fig. 4 Write and read sequence in I^2C interface

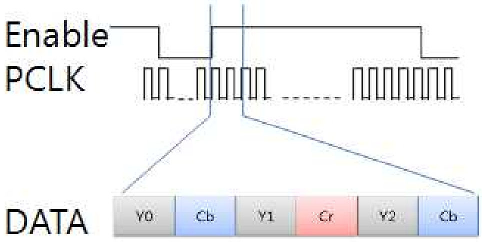


그림 5. YCbCr 422 영상 Data 출력 순서
Fig. 5 Data sequence of YCbCr 422 image

YCbCr422를 RGB로 변환하기 위해 ITU의 국제 표준 디지털 부호화 규격인 CCIR-601[6]을 사용한다. 이 표준은 디지털 TV의 부호화를 위하여 작성되었고 대부분 카메라 센서에서 사용된다. 구현에 사용된 카메라는 Reverse CCIR-601의 RGB 변환을 사용하고 있으며 계산식은 다음과 같다.

$$R = Y + 1.371(Cr - 128)$$
$$G = Y - 0.698 \times (Cr - 128) - 0.336(Cb - 128)$$
$$B = Y + 1.732x(Cb - 128)$$

그림 6은 RGB 변환 장치의 블록 다이어그램이다. CCIR-601 계산식을 FPGA에 구현하기 위하여 Y, Cb, Cr을 각각 모듈에서 계산하여 도출된 값을 최종 연산하여 RGB 값을 구한다. 각 RGB 값 연산 시 소수점 부분은 고정 소수점 방식을 이용하여 계산함으로써 변환된 값이 최대한 정확하도록 한다. 생성된 RGB 데이터는 이미지 처리 장치로 전달된다. 이미지 처리 장치는 ARM으로부터 전달된 값을 바탕으로 이진화 영상을 생성한다. 저장된 패턴을 바탕으로 히스토그램의 최대 및 최소 기준을 설정한다. YCbCr에서 추출된 영상의 값이 유효범위일 경우 인식이 가능하도록 이진화를 수행한다. 생성된 데이터들은 최종적으로 SRAM에 원본 영상과 이미지 처리 장치에서 생성된 이진 영상이 저장된다.

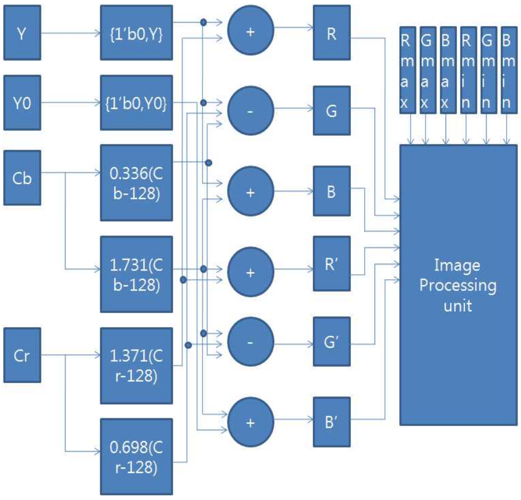


그림 6. YCbCr의 RGB 이미지 변환 장치
Fig. 6 YCbCr to RGB image converter

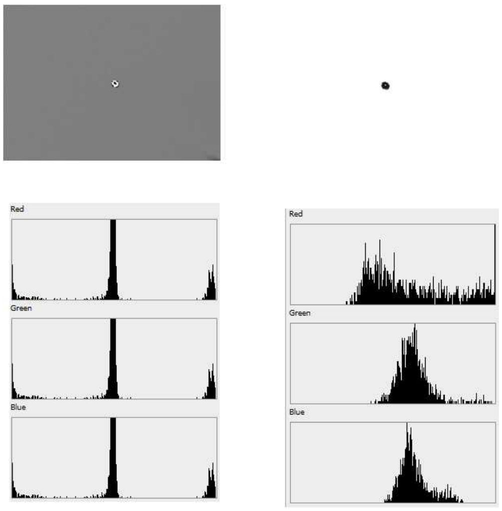
2.3 레이블링과 패턴 분석

이진화 영상 정보에는 레이저 포인터 자취가 작은 셀의 집합으로 되어 있다. 이러한 픽셀들을 하나로 묶어 개체로 인식하도록 하기 위하여 그림 7과 같이 레이블링 기법을 사용하였다. 레이블링 기법은 연결된 셀들에 동일 번호를 붙이고 하나의 개체로 묶어준다. 기준 픽셀 S를 중심으로 상하좌우 4 픽셀을 비교하여 주변 픽셀을 묶어준다. 인식된 개체는 번호가 부여된 후 메모리에 저장되어 패턴 인식 시 빠른 비교가 가능하도록 한다.

레이블링 된 개체는 레이저 포인터 패턴의 히스토그램과 비교하는 과정을 거친다. 정확한 패턴 추적결과를 얻기 위해서는 HSI 좌표계를 사용하여 Hue를 비교하는 방법이 높은 판독성을 보이지만 DSP가 내장되지 않은 마이크로컨트롤러에서는 COS 연산과 제곱근 연산때문에 실행속도가 매우 느리다. 따라서 히스토그램으로 비교하는 방식을 사용하며 판독성은 Hue를 비교하는 방법과 비슷하다. 히스토그램의 Red, Green, Blue 모든 경우의 히스토그램에 대해 판별한다.

그림 8은 HSI와 히스토그램 방식을 이용하여 영상을 처리한 결과이다. 레이저 포인터의 히스토그램은 HSI가 선명하나 결과는 히스토그램방식이 조금 더 패턴 분석에 응용하기 좋은 것을 알 수 있다. 이렇게 내장된 히스토그램 패턴과 대조하여 60% 이상 정확할 경우 레이저 포인터로 인식하며 이때의 좌표가 그림 9와 같이 추출된다. 레이저 포인터의 패턴은 주변 환경의 밝기나 포인터를 비추는 표면의 재질에 따라 변화가 생길 수 있다. 이를 위하여 레이저 포인터의 패턴을 사용자가 사용 전 미리 입력하여 패턴 비교 시 활용하도록 함으로써 정확도를 높일 수 있게 하였다.

그림 7. 레이블링 기법
Fig. 7 Labeling method



(a) HSI (b)히스토그램방식

그림 8. HSI 방식과 히스토그램 방식의 비교
Fig. 8 Comparison between HSI and histogram

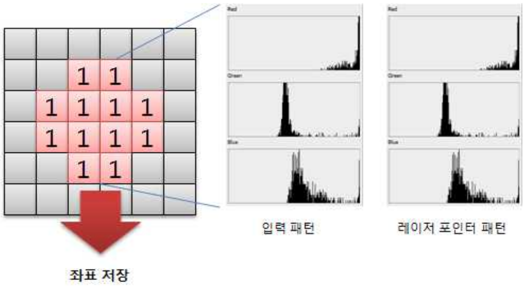


그림 9. 입력영상과 레이저 포인터 패턴
Fig. 9 Input image and pointer patter

2.4 패턴의 데이터 분석 및 저장

레이저포인터 추적 장치를 사용하는 환경과 시간은 자주 변하기 때문에 레이저 포인터의 패턴이 일정하게 한 가지 스펙트럼만을 형성하지 않는다. 따라서 여러 환경에서 레이저 포인터의 사진을 찍어 그 패턴을 입력하고 레이저포인터 추적 장치를 사용하는 환경에서 레이저 포인터 스펙트럼을 직접 채취하여 저장하여 데이터베이스를 구성한다. 구성한 데이터는 히스토그램을 이용한 패턴을 분석 시 데이터로 제공하여 정확한 추적결과가 나오도록 한다. 그림 10은 패턴 분석 장치와 패턴의 데이터베이스 장치의 관계를 나타낸다.

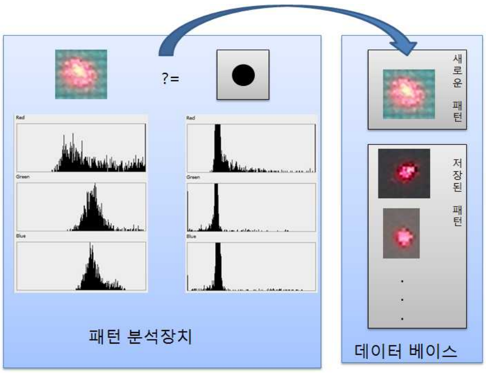


그림 10. 패턴 분석 장치와 데이터 베이스 구성
Fig. 10 Pattern analyzer and database

III. 실험결과

전체적인 시스템의 구성은 FPGA 칩과 ARM을 이용한 시스템으로 그림 11과 같이 구성하였다. FPGA에는 CIS를 위한 카메라 인터페이스와 영상처리장치를 구현한다. 영상처리 장치는 입력된 영상을 RGB 대역으로 변환하여 메모리에 저장 한다. 저장한 이미지는 ARM에서 좌표추적 프로그램을 통하여 처리하여 최종적인 결과를 얻는다. 본 논문에서 구현한 하드웨어는 Excalibur 칩을 사용하여 구현하였으며 사용자가 손쉽게 사용 가능하도록 인터페이스를 구현하였다.

그림 12는 저장된 패턴을 바탕으로 하여 포인터의 추적 결과를 나타낸다. 레이저 포인터가 있는 부분의 대각선 위로 작은 붉은 사각형을 만들어 레이저 포인터 추적 결과를 나타내었다. 그림 13은 프로젝터에서 나온 영상의 2진 추적 결과를 실제 알아볼 수 있는 RGB 영상 위에 나타낸 것으로 정확하게 표시되는 것을 확인 할 수 있다.

최종으로 결과를 확인하기 위하여 스크린과 추적 장치를 1M 간격으로 늘리면서 레이저 포인터를 10번씩 임의의 위치를 지정하도록 하여 인식률과 오차 검사를 수행하였다. 정확도를 비교하기 위하여 패턴 데이터베이스가 없는 추정 장치의 오차율 그래프와 함께 비교한 그래프를 그림 14에 나타내었다. 만일 높은 해상도의 CIS를 사용 시 거리는 배로 늘어날 수 있다.

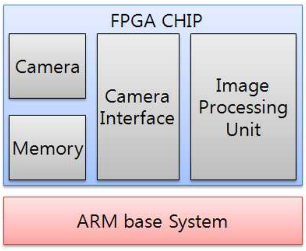


그림 11. 시스템 구성
Fig. 11 System architecture

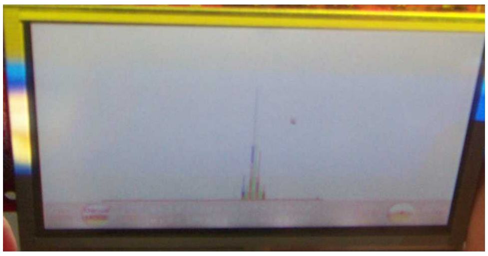


그림 12. 패턴 추적 결과를 이진 영상으로 표현
Fig. 12 Image as a result of pattern tracking

그림 13. RGB 영상위에 추적결과를 표현
Fig. 13 Pattern tracking result on RGB image

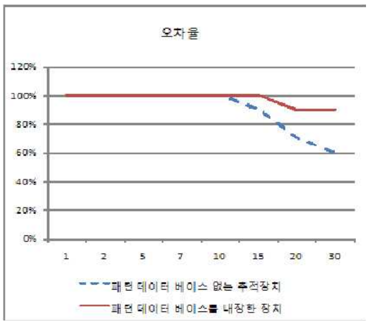


그림 14. 측정 결과 비교
Fig. 14 Comparison of test results

Ⅳ. 결 론

본 논문에서는 Excalibur를 이용하여 레이저 포인터의 스펙트럼을 분석하고 추적하는 시스템을 구성하였다. 구성한 시스템은 붉은색 레이저 포인터를 기본으로 인식 가능하도록 구현하였으며, 초록색 빛의 고휘도를 가지는 포인터 또한 사용 가능하다. PC를 사용한 방식은 CPU에 부하를 주기 때문에 영상의 이진화 부분을 하드웨어로 구현함으로써 성능을 높일 수 있도록 하였다. 레이저 포인터의 패턴을 데이터베이스화하여 인식률을 높였다. 실험 결과 약 20M 이내에 시스템을 설치할 경우 98%이상의 인식률을 보여 충분히 사용 가능함을 확인하였다.

※ 본 논문은 2007년도 PoP-iT 인력양성 사업단에 의하여 지원되었습니다.
※ 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

참고문헌

[1] 김경민, 박귀태, “라인스캔 카메라 인터페이스를 갖는 실시간 영상 전처리 시스템의 설계”, 제어 자동화 시스템 공학 논문지, Vol. 3, No. 6, 1997, pp. 626-631

[2] "Excalibur Sharpened by Nios 2.0", Microprocessor report, Vol. 15, No. 12, 2001, pp. 13-15

[3] 박지영, 이준호, “휴먼마우스 구현을 위한 효율적인 손끝좌표 추적 및 마우스 포인트 제어기법”, 정보과학회논문지, Vol. 29, No. 11, 2002, pp. 851-859

[4] Micron, “MT9M111 1/3-Inch SOC Megapixel CMOS Digital Image Sensor Datasheet”, Micron

[5] Bacciarelli, L., et. al, “Design, testing and prototyping of a software programmable I2C/SPI IP on AMBA bus”, Research in Microelectronics and Electronics 2006, pp. 373-376

[6] Croll, M. G, Devereux V. G, Weston M., "Using the 8 bit CCIR Recommendation 601 digital interface", Broadcasting convention, 1998, pp. 268-271

저자약력

이용환(Yong-hwan Lee)

1993년 연세대학교 전자공학과 (공학사)
1999년 연세대학교 전자공학과 (Ph.D.)
1999~2002 하이닉스반도체
2003~2004 삼성전자
2004~ 금오공과대학교 전자공학부

<관심분야> SoC, 임베디드 시스템 및 소프트웨어, 마이크로프로세서 구조

주현웅(Hyun-woong, Ju)

2007년 대전대학교 IT전자공학과 (공학사)
2007.3~ 금오공과대학교 대학원 전자공학과

<관심분야> SoC설계, 임베디드 시스템

송성해(Sung-hae, Song)

1975년 광운대학교 응용전자공학과
1978년 경응의숙대학교 전기공학과 (공학석사)
1981년 경응의숙대학교 전기공학과 (공학박사)
1981~1984 ETRI 반도체본부
1984~1991 삼성전자
1991~ 금오공과대학교 전자공학부 부교수

<관심분야> 반도체