

PC카메라의 얼굴인식을 이용한 게임컨트롤러의 설계 및 구현

유용빈, 이의석, 권동섭
명지대학교 컴퓨터소프트웨어 학과
e-mail: supermania@gmail.com
zenos2408@gmail.com
dongseop@gmail.com

Design and Implementation of Game Controller by Face detection using a pc camera

Yong-Bin Yu, Eu-Seok Lee, Dong-Seop Kwon
Department of Computer Software Engineering, Myoung Ji University

요 약

최근 비약적으로 향상되고 있는 개인용 컴퓨터의 성능과 새로운 사용자 경험을 원하는 요구와 맞물려 실시간 영상처리분야의 활발한 연구와 발전이 거듭되고 있다. 본 논문은 개인용 컴퓨터의 주변기기로 시중에 널리 보급되어있는 PC 카메라를 이용해 실시간으로 화면 속에서 얼굴을 인지하고 그 이동을 추적함으로써 키보드와 마우스와 같은 보편적인 입력장치 사용하는 경우와 다른 새로운 사용자 경험을 구현할 수 있는 응용 프로그램의 설계 및 구현방법을 제시한다. 본 논문에서는 Java 개발 환경에서 JMF를 이용해 영상장치로 부터 정보를 수집하고 수집한 정보를 기본적인 영상처리 기법과 Integral Image, SSR 필터를 이용한 패턴 분류, 기계 학습 모델인 SVM을 이용해 안면을 인식한다. 이러한 방법들을 통해 인식한 정보를 추적하고 결과를 바탕으로 게임과 같은 다른 응용프로그램으로 키 이벤트를 전달하여 구현하였다.

키워드 : 얼굴, 눈, 인식, SSR, SVM

I. 서론

인체에 대한 다양한 인식 대상 중 안면의 인식은 사용자의 제어가 용이하다는 측면과 가장 직관적으로 활용할 수 있다는 점 그리고 다양한 응용이 가능하다는 측면에서 다양한 생체인식의 분야 중 가장 활발하게 연구되고 있는 분야라고 볼 수 있다.

본 논문은 상대적으로 가격이 저렴해 주변에서 손쉽게 접할 수 있는 개인용 PC 카메라를 활용해 실시간 영상처리 정보를 수신하고 기본적인 영상처리 기법과 SVM 기계학습 모델[1]을 이용해 위치를 인지하고 이동을 추적해 일반적인 1인칭 시점을 구현한 게임 환경에서 사용자가 좀 더 현실감 있는 사용자 경험을 체험할 수 있도록 할 수 있는 응용 프로그램 개발에 초점을 맞추고 진행되었다.

본 연구는 일반적으로 많이 사용하는 C++와 OpenCV[2]

라이브러리를 활용하지 않고 Java 플랫폼 상에서 개발되었으며 부분적으로 JNI(Java Native Interface) 와 Win32 API가 활용되었다. 개발환경은 Eclipse 에서 진행되었으며 영상장치 인식을 위해 JMF(Java Media Framework) 을 사용했고 테스트 코드의 3D오브젝트를 위해 Java 3D 가 사용되었다. 마지막으로 인지 처리에 핵심적인 역할을 하는 SVM 모델의 학습 및 사용을 위해 Support Vector Machines의 자바 구현인 LIBSVM[3]을 활용해 구현하였다.

II. 관련연구

2.1. 얼굴 인식

영상처리에 있어 얼굴의 검출은 패턴 인식 분야중 하나로

주로 입력으로 주어지는 비트맵 영상의 패턴을 일정크기의 윈도우를 결정하고 모든 영역을 비교해 주어진 프레임 안에 얼굴로 추정되는 영역이 있는지를 확인 방식으로 진행된다. 이 작업은 크게 두 가지의 작업으로 나누어지는데 첫 번째로는 주어진 프레임에서 얼굴이 존재할 수 있는 영역을 먼저 찾아낸다. 이후 상대적으로 연산비용이 높은 얼굴검출 프로세스의 대상 군을 좁히는 작업을 먼저 수행하게 된다.

주어진 프레임의 필터를 적용하고 색상, 명암, 체도 등의 정보를 이용해 얼굴이 존재할 수 있는 피부의 값 범위의 임계치를 기준으로 판단하는 방식이 일반적이다 본 연구에서는 필터링을 통해 RGB, HSB, YCbCr 3가지 컬러모델에서 피부색으로 추정할 수 있는 영역을 분류했으며 참고한 활용한 연구는 다음과 같다. RGB [Brand and Mason 2000], [Jones and Rehg 1999][4]., HSB [Fleck et al. 1996][5], YCbCr [Phung et al. 2002][4]

2.2. 패턴 인식

실질적인 얼굴 영역의 검출은 해당 프레임 영역이 얼굴에 해당하는 패턴과 같거나 혹은 유사한지를 판별해야 하는데 이 문제는 분류기(Classifier)를 학습시켜 구분하는 방식으로 진행된다.[6]

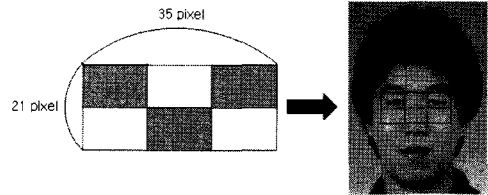
분류기의 학습알고리즘으로는 CBCH(Cascade of Boosted Classifiers working with Haar-like features) , LDA(Linear Discriminant Analysis) , 뉴런 네트워크, SVM(Support Vector Mechine) 등 많은 알고리즘이 있고 일반적으로 OpenCV 기반의 구현에서 사용된 CBCH 방식을 많이 사용한다.

본 연구에서는 SVM을 활용해 최근에 패턴분류에 있어서 각광을 받고 있는 SVM 모델은 1995년 Vladimir Naumovich Vapnik 에 의해 개발된 통계적 학습이론으로서 학습데이터와 범주 정보의 학습진단을 대상으로 학습과정에서 얻어진 확률분포를 이용하여 의사결정함수를 추정 한 후 이 함수에 따라 새로운 데이터를 이원 분류하는 것으로 VC (Vapnik-Chervonenkis) 이론이라고도 한다.[7] 특히 SVM 은 분류 문제에 있어서 일반화 능력이 높기 때문에 많은 분야에서 응용되고 있다.

일반적으로 얼굴검출의 SVM을 활용한 경우 우선적으로 수행하는 알고리즘이 있다. 우리가 얼굴을 분류하기 위해 사용하는 영상은 회색조로 변환된 영상의 35 x 21 (735 픽셀) 영역의 패턴이다. 이 영역을 가로가 3 세로 2 형태로 구획을 나누어 볼 때 정상적으로 얼굴로 분류되는 최적의 영역은 눈으로 추정되는 상단의 좌우 영역과 코로 추정되는 하단의 가운데 영역이 다른 분위에 비해 어두운 특성을 가진다.

SVM은 이 영역의 각 비트맵 정보를 학습하고 분류한다. 이러한 특성을 기반으로 얼굴을 인식할 때 해당 35 x 21 영역을 BTE(Between The Eyes : 눈 사이 영역) 이라고 부르고 이 영역을 3 x 2 형태의 나누고 패턴을 처리하는 방식을 SSR(Six Segmented Rectangular filters) 이라고 한다.(그림 1) SSR을 구성하는 각각의 영역은 그 영역에 포함

된 모든 픽셀의 그레이 스케일 비트맵 값을 합산한 값으로 패턴을 인식하는데 이때 각 픽셀의 값을 합산하는 해 처리하는 적분이미지(Integral Image) 방식 이라고 지칭한다.[8]



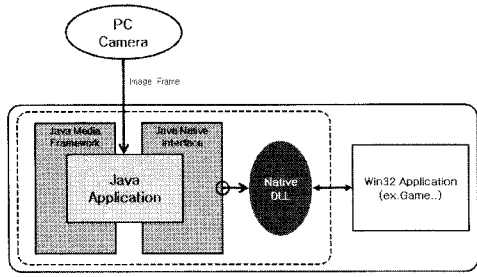
(그림 1) SSR 필터의 적용 방법

2.3. 개발환경 관련

본 연구에서는 영상처리의 세부과정에 대한 처리를 최대한 직접 구현 하고자 C++와 OpenCV 라이브러리를 사용하지 않고 연구자들에게 익숙한 Java 플랫폼 상에서 진행되었다. Java 플랫폼은 운영체제에서 구동되는 가상기계 (Virtual Machine)을 통해 응용프로그램이 구동되는 형태의 실행환경을 가지는데 이 환경은 응용프로그램의 플랫폼 종속성을 피할 수 있게 해 준다는 장점이 있는 반면 호스트 운영체제의 자원인 장치를 활용하거나 VM이외의 프로세스와 통신을 하는데 있어서 많은 제약사항이 따른다. 이번 연구에서는 USB로 연결하는 PC 카메라의 영상인식을 응용프로그램의 데이터 인풋으로 사용해야하는데 이 장치의 인식 문제가 있고 또 최종적으로 검출한 얼굴을 추적한 결과를 별도의 1인칭 FPS 게임 프로그램의 키 이벤트로 생성해야하는데 두 가지 모두 Java의 실행환경 제약사항에 해당하는 문제로 이 문제를 보안하기 위해 별도의 보안 프레임워크 라이브러리와 API 인터페이스를 활용했다.

먼저 장치의 인식과 사용부분은 JMF(Java Media Framework) 을 사용했다. JMF는 Java에서 가장 취약한 멀티미디어 제어 관련 API 및 SDK를 보강한 것으로 음성 및 영상의 캡처 및 저장 전송 등에 사용되는 프레임워크이며 부수적으로 호스트 운영체제의 멀티미디어 장비를 VM 상에서 사용할 수 있도록 중재해주는 JMF Registry 를 제공한다.

또한 일반 응용프로그램으로 키 이벤트를 전달하기 위해서는 JNI(Java Native Interface)를 활용했다. JNI는 앞서 언급한 자바 실행환경의 한계를 보완하기 위해 고안된 API로 자바 응용프로그램이 운영되는 호스트 운영체제의 시스템 호출을 동적으로 호출해 사용할 수 있는 방법을 제공한다. JNI를 활용하면 작성한 응용프로그램의 Class로부터 호스트 C/C++ 형태의 헤더파일이 자동으로 생성되고 해당 헤더파일의 규칙에 맞게 동적연결 라이브러리 형태로 시스템 호출부분을 처리해 생성하면 해당 Class의 로딩시점에 시스템 호출을 포함한 동적 라이브러리를 호출할 수 있게 된다. 위의 구조를 (그림 2)에서 나타내었다.



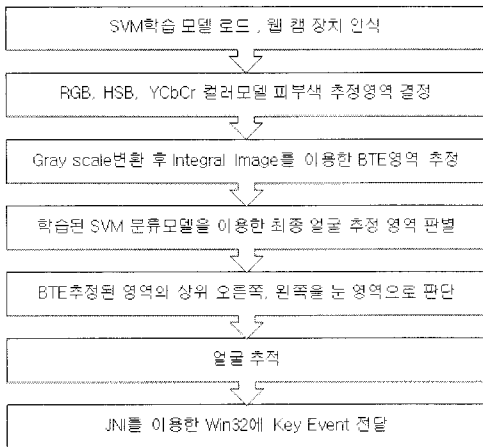
<그림 2> 프로그램 구조

III. 컨트롤러 설계 및 구현

3.1 세부구현

얼굴의 인식처리 기능은 동일하고 인식결과를 활용하는 방법에 있어서 Java 내부 응용프로그램에 활용하는 응용프로그램과 JNI를 이용해 외부 응용프로그램으로 전달하는 두 가지 구현을 제작했다. 진행과정에서 아래와 같은 이슈들이 발생한다. 응용프로그램의 세부 구현은 <그림3>과 같다.

<그림 3> 얼굴 인식 과정 흐름도

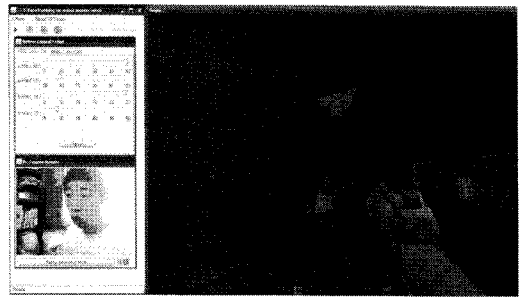


- ① 응용프로그램이 구동되면 text 형태로 저장된 SVM 학습 자료를 읽고 학습 자료를 바탕으로 한 분류 모델을 메모리상에 상주시킨다.
- ② JMFRegistry 로 등록된 USB PC 카메라 장치는 320x240 크기의 정지영상 프레임을 초당 30회 device 객체로 전달하고 이 전달 받은 이미지 프레임을 대상으로 인식과 추적을 수행한다.
- ③ 비트맵 이미지 형태로 전달받은 각각의 프레임은 얼굴의 위치를 검출하기에 앞서 얼굴이 존재할 수 있는 피부색 추정영역을 검색대상으로 제한하기 위해 RGB, HSB, YCbCr 컬러모델로 각각의 프레임을 변환하면서 피부색 추정영역을 먼저 결정한다. 이때 각각의 컬러 모델에서 설정하는 임계 치에 따라 추정영역의 범위가

가 달라지고 각각의 추정영역을 AND 연산으로 합산해 나가는 순서에 따라라도 최종 피부색 추정영역을 결정 하는데 차이가 발생 한다.

- ④ 주어진 프레임 상에서 BTE 영역을 결정하는 최종 목표를 달성하기 위해 피부색 추정 영역을 그레이스케일로 변환한 후 작분 이미지를 합산한 형태로 순회하고 그렇게 생성된 BTE 추정 영역을 ①을 통해 학습된 SVM 분류모델을 통해 최종 얼굴 추정 영역으로 판별을 한다.
- ⑤ BTE 로 얼굴 추정 영역이 결정되면 상단 좌우측 영역 (S1, S3)를 눈 영역으로 가정하고 해당 영역의 중점을 양 눈의 위치로 생각한다. 이 눈 위치의 이동을 통해 얼굴의 좌우 이동을 추적한다. 얼굴의 z축 이동 (원근 이동) 의 경우 시연자가 카메라에서 더 멀어질 경우 원근에 의해서 눈과 눈 사이의 거리가 초기보다 더 작아 지므로 이를 비율로 활용해 원근이동을 추정한다.
- ⑥ 이렇게 결정된 좌우이동, 원근이동에 대한 정보를 별도의 응용프로그램의 지정된 Key Event 로 발생하기 위해 JNI로 작업된 DLL의 시스템 함수를 호출한다.

완성된 프로그램의 동작화면은 <그림 4>와 같다. <그림 5>는 얼굴 인식 부분만 확대한 그림이다.



<그림 4> 동작 화면



<그림 5> 얼굴 인식 부분 확대

IV. 결론

본 논문에서는 다양한 컬러모델을 통한 피부톤 검출과 기계 학습 분류기를 통한 패턴 인식기법을 활용해 사용자의 얼굴의 위치를 인지하고 그 위치를 추적해 이를 사용자의 입력 정보로 활용해 3D FPS 게임을 제어하는 응용프로그램으로 구현하였다.

설계 및 구현에 있어 가장 큰 특징은 첫째 Java 플랫폼 기반의 구현물로서 이후 이기종간의 포팅이 용이하도록 설계 하였고 둘째 인지 응용프로그램이 제어하는 응용프로그램을 내부적으로 동작하는 응용프로그램이 아닌 외부의 프로그램을 제어할 수 있도록 설계해 응용프로그램의 범용성을 높이는 구조에 집중했다.

이러한 연구결과를 바탕으로 기본적인 인지성능 향상과 함께 처리속도를 최적화 하는 작업을 지속적으로 진행해 나가면서 현재 특정 OS 에 종속적으로 구현된 부분들을 모듈화해 레이어로 구성해 Java VM이 동작하는 이기종 플랫폼으로의 이전을 시도하고자 한다.

[8] Paul Viola Michael J. Jones, "Robust Real-time Object Detection" International Journal of Computer Vision 57(2), 137-154, 2004

참고문헌

[1] Shinjiro KAWATO1, Nobuji TETSUTANI1,2 and Kenichi HOSAKA1 "Scale-Adaptive Face Detection and Tracking in Real Time with SSR Filters and Support Vector Machine", IEICE TRANS. INF. & SYST., L.E88-D, NO.12 DECEMBER 2005

[2] OpenCV Library, Intel, <http://sourceforge.net/projects/opencvlibrary/>

[3] LIBSVM, Chih-Chung Chang and Chih-Jen Lin, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

[4] Vladimir Vezhnevets, & Vassili Sazonov, Alla Andreeva. "A Survey on Pixel-Based Skin Color Detection Techniques."

[5] Shinjiro Kawato, Jun Ohya, "Automatic skin-color distribution extraction for face detection and tracking". WCCC-ICSP 2000. 5th International Conference on Volume 2, Issue , 2000 Page(s):1415 - 1418 vol.2

[6] 복합 알고리즘을 이용한 실시간 얼굴검출 및 SVM 인식 기술 : 박정선, 이상웅, 유명현, 정여아, 양희덕, 한국정보보호학회, 2002

[7] Vladimir Naumovich Vapnik, The Nature of Statistical Learning Theory, 2nd ed., Springer-Verlag, New York, NY, 2000.