

OpenCV를 활용한 컬러추적 문자 인식기의 구현

Implementation of the character recognizer using color tracking with OpenCV

김대기

Daegi Kim

요약 최근 지능 기술이 발달하면서 문자 인식(character recognizer)에 관련된 기술도 나날이 발전되어 가고 있다. 이런 환경에서 문자 인식 기술을 구현하는 방법으로는 신경망(Neural Network) 알고리즘을 통한 방법, 템플릿 매칭을 통한 방법, 문자의 특징을 비교하는 방법, 체인코드를 이용하는 방법 등이 있다. 이 레포트에서는 OpenCV와 체인코드 알고리즘을 통해 컬러추적 문자 인식기를 구현하는 방법을 제안한다.

1. 서론

문자인식(Character Recognition)이란 시각(Vision) 정보를 통하여 문자를 인식하고 의미를 이해하는 인간의 능력을 컴퓨터로 실현하려는 패턴인식(Pattern Recognition)의 한 분야로서, 광학 문자 인식, 우편물 자동 분류, 문서인식, 도면인식 등의 분야에서 부분적으로 실용화가 이루어지게 되었으며, 요즘에는 인공지능의 최신 기법인 신경망, 퍼지, 유전알고리즘 등의 응용과 자연어처리, 심리학, 생리학, 인지과학 등 관련 학문과의 접목에 의해 문자인식 기술은 새로운 단계에 접어들게 되었다. 일반적으로 문자인식은 공간 정보만을 이용하는 오프라인 문자 인식(광학문자인식(OCR))과 공간 정보 외에 시간 정보도 함께 이용할 수 있는 온라인 문자 인식(필기체인식(Handwriting Recognition))인식으로 구분된다. [1]

2012년 2학기 "오토마타 및 지능 컴퓨팅" 프로젝트의 일환으로 문자 인식기를 구현하게 되었다. 학기 초에 윤창민 조교님이 프로젝트에 대해 설명해 주시면서 카메라 장치(Camera Device)를 통해 영상을 입력 받고 사람의 손가락(Finger)의 위치를 추적하여 문자를 인식하는 방법을 제안하셨다. 그 말을 듣고 "한번 구현해 보면 재미있겠다"는 생각이 들었고, 팀원들에게 제안하여 프로젝트를 시작하게 되었다. 이 강의를 수강하기 이전부터 OpenCV 라이브러리에 대한 공부를 조금씩 하고 있었던 나는, 컬러 검출(Color Detection)을 통해 사람의 손의 위치를 찾아낼 수 있다고 생각하였다. 하지만 구현 과정에서 사람이 직접 손을 움직여 문자를 입력하기에는 여러 가지 오차가 많이 발생할 수 있다는 문제를 인식하게 되었다. 그래서 결국 빨간색 물체를 추적하여 문자 인식의 입력 수단으로 이용하기로 결정했다.

2. 기반 기술

2.1 Chain Code

문자 인식기를 구현하는 방법에는 여러 가지가 있지만 우리는 체인코드를 이용해 구현하기로 했다. 처음에 조교님께서 샘플로 주신 프로그램 역시 체인코드에 기반해 구현된 프로그램 이었고,

신경망, 퍼지, 템플릿 매칭 등에 비해 비교적 쉽게 구현할 수 있다는 장점이 있다.

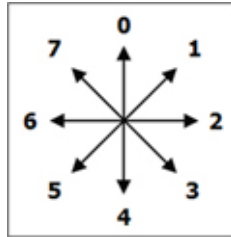


그림1. 8방향 체인코드

프로그램은 사용자 입력의 진행 방향을 분석해 적절한 8방향 체인코드를 출력하게 된다. 예를 들어, 사용자가 'ㄱ'이라는 문자를 입력한다면 체인코드 '24'를 생성하게 된다. 이렇게 생성된 체인코드를 미리 학습된 데이터와 비교해 적절한 자음 또는 모음으로 반환한다면 문자를 인식할 수 있게 된다.

2.2 Color Tracking

우리는 사용자 입력 수단으로 빨간색 물체를 이용하기로 했다. 즉, 카메라 장치를 통해 받은 영상에서 빨간색 컬러 영역을 검출한 후에 중심 좌표를 찾도록 하였다. 컬러 검출과 중심 좌표 계산은 각각 OpenCV 함수 InRangeS(), cvMoments()를 이용해 구현할 수 있었다.

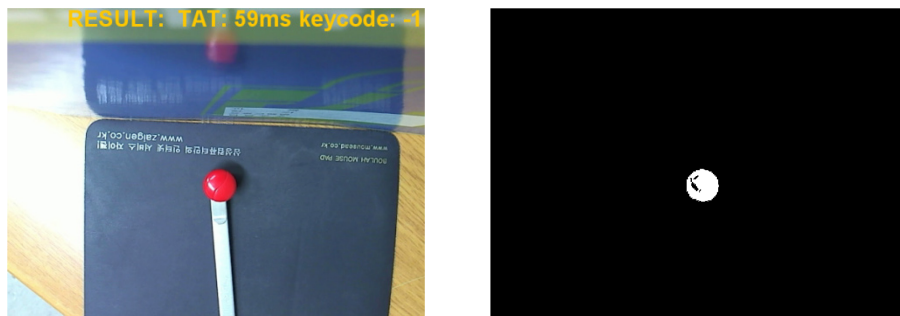


그림2. OpenCV 함수를 이용한 빨간색 컬러 영역 검출

컬러 검출을 위해 빨간색의 HSV 컬러 영역을 찾아내는 과정이 필요했다. InRangeS() 함수는 컬러 검출을 위해 컬러 범위를 파라미터로 받는데, 여기에 적절한 범위를 지정해야 한다. 우리는 많은 실험 끝에 (140,135,135,0) ~ (179,255,255,0)까지의 범위를 빨간색의 HSV 컬러 영역으로 지정하기로 했다.

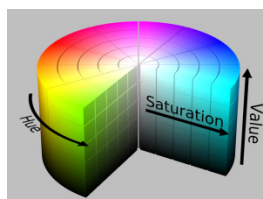


그림3. HSV color solid cylinder alpha low gamma

출처: Wikipedia, http://en.wikipedia.org/wiki/File:HSV_color_solid_cylinder_alpha_lowgamma.png

2.3 Apache Lucene

우리의 프로젝트는 실시간(Real-Time)으로 카메라 영상을 처리해 문자를 인식해야 하기 때문에 빠른 유니코드 검색이 필요했다. Apache Lucene은 확장 가능한 고성능 정보 검색 라이브러리로써, 수년간 가장 유명한 오픈 소스 자바 정보 검색 라이브러리로 잘 알려져 있다. 조교님께서 샘플로 주신 프로그램에서는 일반 텍스트 파일에서 유니코드를 검색하는 것을 확인할 수 있었다. 해당 방법을 우리 프로젝트에 적용해 보았는데, 실시간 검색 속도에 영향을 미치는 것을 발견했다. 때문에 유니코드 검색을 루씬이 처리하도록 변경했다. 일반 파일에서 선형 검색(Linear-Search) 시에 10밀리 초가 소요되었지만, 루씬을 이용한 후에는 200마이크로 초로 단축되었다. 루씬이 이토록 빠른 검색 시간을 보이는 이유는 자체적으로 고안된 알고리즘에 있다. 루씬을 통해 검색을 하기 위해서는 미리 파일 전체를 인덱싱 하는 과정이 필요한데, 이 과정에서 루씬은 빠른 검색을 하도록 역 인덱스(Inverted-index) 형태로 index 파일을 구성해 놓는다고 알려져 있다. [2]

3. 프로그램의 구현

3.1 프로그램 구현 환경

프로그램은 Java 언어로 작성되었으며, 자세한 개발 및 테스트 환경은 아래와 같다.

운영체제	윈도우 7 64비트
Java 버전	1.7
통합 개발 환경	이클립스 주노
OpenCV 버전	2.4.2
JavaCV 버전	0.2
Lucene 버전	4.0.0
카메라 장치	로지텍 Webcam C910

표1. 프로그램 개발 및 테스트 환경

팀원들과의 상의를 통해 모든 팀원에게 익숙한 Java를 Main-Language로 선택하였다. 그렇기 때문에 OpenCV 만으로는 개발을 하기 어려웠고, OpenCV를 Java 언어에서 사용할 수 있도록 지원해주는 JavaCV 라이브러리를 추가적으로 이용해야 했다. 카메라 장치는 로지텍 Webcam C910을 이용했다.

3.2 팀원 역할 분담

나는 주로 OpenCV(JavaCV)를 이용한 영상처리 부분과 Apache Lucene을 이용해 유니코드 인덱싱, 검색을 구현하는 부분을 맡았다.

TrackingColoredObjects.java	컬러 객체를 검출하고 중심점을 찾는다. (메인 클래스)
Indexer.java	루씬 인덱스를 생성한다.
Searcher.java	루씬 인덱스에서 유니코드를 검색한다.

표2. 내가 구현한 주요 클래스

3.3 프로그램 테스트

프로그램 테스트에 대한 부분은 동영상 링크로 대체한다.

링크: <https://vimeo.com/53994347>

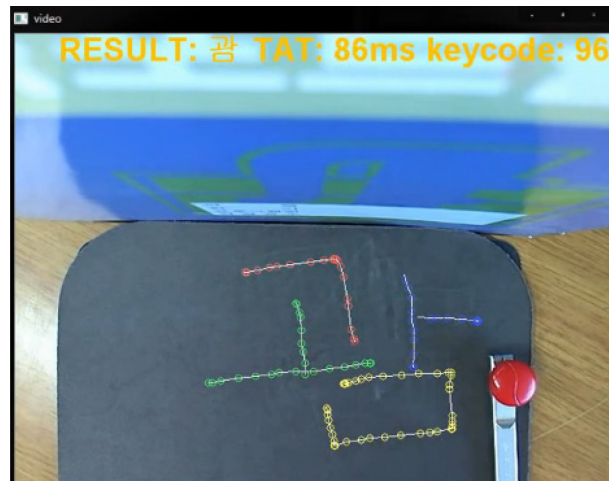


그림3. 프로그램 테스트 결과

4. 결론 및 향후 방향

이번 프로젝트를 통하여 체인코드와 컬러추적을 통해 문자 인식 프로그램을 구현했다. 구현 과정에서 많은 시행 착오가 있었다. 실시간으로 초당 24 프레임을 처리하다 보니 프로그램에 많은 부하가 발생했고 이는 처리 속도의 저하로 나타났다. 아파치 루션을 적용하고 수많은 코드 최적화를 통해 이를 개선했고 2.14Ghz Dual-Core CPU에서 프레임당 60ms~100ms 사이의 처리 속도를 확보하게 되었다.

하지만 사용자 인터페이스 측면에서 우리 프로그램은 불편한 점이 많다. 유저 입력의 시작과 끝을 프로그램에게 알리기 위해 키보드를 사용했다는 점이 가장 치명적이다. 사용자가 원치 않는 입력을 막기 위해 일종의 '스위치' 역할을 하도록 한 것인데, 처음 사용하는 유저에게는 불편할 수 밖에 없다. 이를 개선하기 위해 여러 가지 방법을 생각해 보았는데, 그 중에 가장 고려해 볼만한 것은 Microsoft Kinect를 이용하는 것이다. Kinect는 Xbox 360 비디오 게임 콘솔을 위한 모션 센서 장치이지만, 영상처리 분야에서도 이용되고 있다. Kinect에는 깊이(Depth)를 알아낼 수 있는 센서가 있기 때문에 빨간 물체를 바닥에 붙였는지 혹은 떼었는지를 알 수 있을 것이다. 이를 이용한다면 실제 연필로 종이에 글씨를 쓰듯이 빨간색 물체를 통해 자연스럽게 문자를 입력할 수 있으리라 기대한다.

Reference

- [1] http://www.aistudy.com/pattern/character_recognition.htm
- [2] http://en.wikipedia.org/wiki/Inverted_index