

고속 교통표지판 인식 알고리즘

Fast algorithm for Traffic Sign Recognition

Ding Dajun* , 이찬호**

Ding Dajun* , Chanho Lee**

Abstract

Information technology improves convenience, safety, and performance of automobiles. Recently, a lot of algorithms are studied to provide safety and environment information for driving, and traffic sign recognition is one of them. It can provide important information for safety driving. In this paper, we propose a method for traffic sign detection and identification concentrating on reducing the computation time. First, potential traffic signs are segmented by color threshold, and a polygon approximation algorithm is used to detect appropriate polygons. The potential signs are compared with the template signs in the database using SURF and ORB feature matching method.

요 약

자동차가 IT 기술과 융합되면서 편의성과 안전성 그리고 성능이 좋아지고 있다. 이와 관련하여 최근 자동차의 주행시 안전 및 주변 환경과 관련된 정보를 제공하기 위한 많은 알고리즘이 연구되고 있으며 교통표지판 인식 또한 그 중 하나이다. 교통표지판 인식은 안전 운전에 필요한 중요한 정보를 제공해 준다. 본 논문에서는 연산 시간 감소에 중점을 두어 교통표지판을 탐지하고 판별하는 인식 알고리즘을 제안한다. 제안한 알고리즘에서는 색상 임계값을 이용하여 교통표지판 후보를 분할하고 다각형 근사법을 이용하여 적절한 다각형을 찾는다. 이렇게 찾은 패턴에 대해 SURF와 ORB 알고리즘을 이용하여 데이터베이스와 비교하여 교통표지판을 식별한다.

Key words : Pattern Recognition, Traffic Sign, SIFT, SURF, ORB, OpenCV

1. 서론

자동차에 IT 기술이 접목되어 지능형 자동차 개발이 활성화되고 있고 이에 따라 자동차의 편의성과 안전성 그리고 성능이 크게 개선되고 있고 이를 위한 연구 또한 활발히 진행되고 있다. 이러한 연구는 자동차 주행과 관련하여 성능 및 안전성을 개선하는 연

구와 차량의 상태 및 주변 환경 등의 정보를 제공하여 편의성과 안전성을 높이는 분야로 나누어지는데 후자의 경우 카메라를 통한 영상을 분석하여 주변 상황을 인식하는 연구가 중요한 부분을 차지한다.

교통표지판은 보행자와 운전자가 이를 신속하고 정확히 인식하면 모두에게 안전하고 원활한 이동에 도움을 준다. 운전자의 경우에는 달리는 차에서 교통표지판을 인식해야 하므로 도로가 복잡하거나 교통표지판이 한 번에 너무 많이 나타거나 날씨가 안 좋은 경우 등 운전 환경이 좋지 않으면 인식에 어려움을 겪는다. 교통표지판 인식 시스템이 있는 경우 그러한 상황에서도 교통표지판을 신속 정확하게 인식하여 운전자에게 정보를 제공함으로써 운전자의 운전 행위에 도움을 준다[1]. 또한 자동 운전 시스템에서는 차량 스스로 교통 표지판을 인식하여 그에 적합한 주행을 하는데 필요한 정보를 제공한다. 네비게이션 시스템에 교통표지판 정보가 포함되기도 하지만 GPS 신호

* ding.dajun@gmail.com, 02-825-8108, Dept. of Electronic Engineering, Soongsil University

★ chlee@soongsil.ac.kr, 02-820-0710, School of Electronic Engineering, Soongsil University

※ This work was sponsored by Industrial Strategic Technology Development Program funded by the Ministry of Knowledge Economy (MKE, Korea) (10039188, Development of multimedia convergence programmable platform for smart vehicles). The EDA tools are supported by IDEC.

Manuscript received Oct. 3, 2012; revised Nov. 29, 2012 ; accepted Dec. 3 2012

에 의한 오차와 DB에 포함되지 않은 도로나 표지판이 있을 경우 안전에 문제가 발생할 수 있다. 따라서 영상을 통한 교통표지판 인식은 주행 중 주변 상황을 가장 확실하게 파악할 수 있고 네비게이션은 보조적으로 인식을 증가와 인식 시간 감소에 도움을 줄 수 있다.

교통표지판은 일반적으로 일정한 형태와 색상을 가지고 있다. 예를 들어, 대한민국의 도로 상태에 대한 36개의 주의 표지판은 그림 1에 나타난 것처럼 모두 노란 배경에 빨간색 테두리를 갖는 삼각형 모양이다. 그리고 내부에는 검은색으로 도로 상태를 나타내는 그림문자(pictogram)가 포함된다. 문자가 포함될 경우 한글과 영어로 병기된다. 따라서 교통표지판 인식을 위해서는 특정 형태를 인식하는 방법과 특정 색을 인식하는 방법이 있는데 형태를 인식하는 방법은 배경에 비슷한 형태가 상당히 많이 발견될 가능성이 커서 인식 확률이 떨어지고 시간도 오래 걸린다. 따라서 특정 색을 인식하는 방법이 많이 사용된다. 이때 RGB 색 공간은 빛의 변화에 민감하여 임계값을 이용한 색 분할이 어렵다. 한편 HSV(Hue Saturation Value) 색 공간에서는 비슷한 색을 갖는 관심 물체의 hue와 saturation 값에 대한 변화가 작다[1,2]. 따라서 HSV 색 공간으로 변환하여 색 분할을 하는 방식이 주로 이용된다. 한편, RGB 색 채널과 픽셀 벡터값을 결합하여 색 분할을 하는 방식이 이용되기도 하였다[3].

색 분할이후에는 대상의 형태를 파악해야 하는데 3가지 방법이 알려져 있다. 첫 번째는 Hough 변환을 이용하여 직선을 찾아내고 이를 연결하여 다각형과 원을 탐지하는 것이다[1]. 두 번째는 DtBs(Distance to Borders) 방법을 적용하여 형태 정보를 얻는 것이다[2]. 네 방향의 DtBs 벡터를 계산하여 기하학적 정보를 얻는데 이용한다. 세 번째는 웨이블릿 변환에 근거한 CPT(Central Projection Transformation) 알고리즘을 이용하는 것이다[3]. 즉, 파형 응답을 분석하여 원과 삼각형, 사각형을 구분한다.

기하학적 도형의 형태를 얻은 뒤에는 교통표지판 후보를 찾고 이를 식별한다. 이를 위해 템플릿 매칭[4], 부스팅(boosting) 알고리즘[5]이나 신경 네트워크(neural network)[6]과 같은 기계 학습법(machine learning), 그리고 특징 매칭(feature matching)[1,3] 등이 이용되었다. 기계 학습은 가장 널리 사용되는 방식중 하나로 학습 단계에 따라 데이터베이스를 훈련시키고 탐지된 물체에 대해 훈련된 데이터베이스를 이용하는 것이다. 이 방식은 학습에 너무 많은 시간이 소요되고 학습되지 않은 정보에 대해서는 식별이 어려운 단점이 있다. 기계 학습을 완료하기 위해

SVM(Support Vector Machine)을 적용하기도 하였다[2]. 특징 매칭 방법으로는 SIFT가 사용되었는데 가장 고전적인 방법이지만 계산 시간이 오래 걸리는 단점이 있다[1]. 한편 탐지된 교통표지판 후보 도형을 CPT(Central Projected Transformation) 기반의 형태(shape) 특징을 추출하여 식별한 방식도 있으며 Hu[7], Tchebichef[8], Zernike[9] 등의 불변 모멘트(invariant moments) 방식에 비해 좋은 인식을 보여주었다[3]. 그러나 여전히 계산 시간이 길다는 단점을 가지고 있다.

본 논문에서는 차량에 적용 가능한 고속의 교통표지판 인식 알고리즘을 제안한다. 탐지 단계에서는 색 분할과 다각형 근사법, 그리고 아핀 변환을 이용한 형태 추출이 이루어지고 식별 단계에서는 SURF(Speed Up Feature Transform)[10]와 ORB(Oriented FAST and Rotated BRIEF)[11]를 이용한 특징점 생성과 데이터베이스의 템플릿(template)와 매칭을 이용한다. 제안한 알고리즘에 따라 OpenCV 라이브러리를 이용하여 교통표지판 인식 소프트웨어를 구현하였으며 다양한 샘플에 대해 각 단계별 연산 시간과 인식을 비교하였다.

II. 교통표지판 탐지 과정

제안한 교통표지판 인식 알고리즘은 교통표지판 탐지(detection)와 식별(identification)의 두 단계로 구성된다. 탐지의 첫 번째 단계는 교통표지판 후보를 찾기 위해 주어진 색 공간(color space)에서 색 분할(color segmentation)을 하는 것이다. 주의와 규제 표지판은 삼각형, 오각형, 팔각형, 그리고 원의 형태로 구성된다. 이중 팔각형이 하나, 오각형이 두 개, 그리고 역삼각형이 두 개이므로 이들은 식별 과정이 매우 간단하다. 주의와 규제 표지판의 특징은 가장자리 또는 내부가 붉은색으로 되어 있다는 점이다. 이를 이용하면 교통표지판 후보 영역을 상당히 좁힐 수 있다. 색 분할을 위해서는 서론에서 언급한 바와 같이 HSV 색 공간으로 변환하여 색 분할을 하는 방식을 이용한다. 빛에 의한 영향 외에 날씨에 의해 이미지가 선명하지 않은 경우에는 별도의 알고리즘을 통해 복구해야하는데 본 논문의 범위를 벗어나는 작업이므로 다루지 않았다[12].

그림 1(a)에서 교통표지판 탐지 과정중 색 분할 결과를 보여준다. 입력 영상에는 삼각형의 위험 표지판과 붉은색 블롭(blob) 잡음이 포함되어 있다. 블롭 잡음에 대해서는 뒤에서 자세히 설명하겠지만 교통표지판 인식에 방해가 되는 물체가 주위에 있음을 가정하

고 이러한 영향을 배제하고 교통표지판을 인식할 수 있음을 보이기 위한 것이다. HSV 색 공간에서 붉은 색은 상위와 하위의 두 개의 hue 영역을 가지고 있다 [2]. 그림 1(b)와 (c)는 각각 상위 영역과 하위 영역에서 임계값을 이용한 이진화 영상이다. 붉은 색이 두 영역에 걸쳐 분포하므로 두 영상 모두 붉은색을 완벽히 표현하지 못하고 있다. 두 영상을 하나로 합치면 그림 1(d)와 같이 붉은색에 대한 완성된 분할 영역을 얻을 수 있다. 다음 동작은 잡음을 제거하고 경계선을 복구하는 선처리(preprocessing) 과정이다. 제안한 알고리즘에서는 메디안(Median) 필터와 모폴로지(Morphology) 방식을 이용하여 처리하였고 그 결과가 그림 1(e)에 나타나 있다.

색 분할 이후에는 이진화된 영상에서 원하는 다각형 패턴을 찾는다. 먼저 모든 물체에 대해 윤곽선 데이터를 찾기 위한 작업을 한다. 경계점들의 위치를 모으는 과정을 통해 윤곽선을 얻는다. 이 윤곽선은 두 개의 변수를 입력으로 하는 함수의 값이 상수인 곡선을 따라 구한다[13]. 윤곽선들은 외부 윤곽선과 내부 홀(hole)로 나누어진다. 그림 1(f)에서 외부 윤곽선은 붉은색으로, 내부 홀은 파란색으로 구분된다. 교통표지판 탐지 과정에서 관심 부분은 외부 윤곽선에 가장 가까운 1차 내부 홀이다. 이는 주의 및 규제 교통표지판이 대부분 붉은 테두리를 가지고 있고 그 내부에 노란 바탕과 그림문자가 존재하기 때문이다. 1차 내부 홀 윤곽선을 찾으면 다각형 근사법을 적용한다[14]. 이 알고리즘은 윤곽선에서 두 개의 극점(extremal point)을 찾아 그들을 잇는 선분을 만든다. 그리고 윤곽선의 다른 극점중 선분에서 가장 먼 것이 이 선분에 추가한다. 이 과정을 반복하여 그 다음 먼 극점을 계속 추가하다가 모든 극점이 정확도 파라미터에 의해 정해지는 거리보다 작아지면 멈춘다. 이러한 반복 과정의 결과가 삼각형, 오각형, 팔각형이 되면 교통표지판 후보가 탐지된 것으로 간주한다. 그림 1(g)는 파란색 삼각형을 찾은 결과를 보여준다. 원의 경우에는 Hough 변환을 이용한다. 먼저 XY 좌표를 극좌표(polar coordinate)로 바꾸고 데이터 수집 알고리즘을 통해 Hough 공간에서 가장 유력한 점을 모아 가능한 원을 구성한 뒤 다시 XY 좌표로 변환하여 원의 중심과 반지름을 얻는다.

1차 내부 홀 윤곽선을 이용하여 원하는 다각형을 찾는 방식은 두 가지 이점이 있다. 첫째로 일반적인 물체는 대부분 내부에 홀을 가지고 있지 않아 많은 물체를 제거할 수 있다. 둘째로 외부 윤곽선을 이용할 경우 탐지 과정에서 해당 물체의 배경이 매우 민감하게 영향을 미친다. 예를 들어 주위에 붉은색 자동차가 있거나 붉은색 옷을 입은 행인이 있을 경우

색 분리 이후 얻는 결과는 원하는 다각형이 아닌 복잡하고 불규칙한 모양의 도형이 될 것이다. 그림 1(f)와 (g)를 보면 붉은 블롭 모양의 물체는 인위적으로 넣은 것이기는 하나 탐지가 된 것을 확인할 수 있다. 반면에 제안하는 알고리즘의 1차 내부 윤곽선을 이용하는 방식은 일반적으로 교통표지판을 잘 보이는 곳에 다른 물체가 잘 가리지 않도록 설치하므로 그러한 잡음에 훨씬 덜 민감하다.

위에서 언급한 과정이 끝나면 교통표지판 후보는 그림 1(h)에 나타난 것처럼 아핀 변환(Affine Transformation)[15] 과정을 거쳐 200 x 200 픽셀 크기의 표준 영상으로 바뀐다. 이는 카메라를 통해 입력되는 영상이 일반적으로 차량의 정면을 향하므로 교통표지판은 기울어진 형태가 된다. 그런데 그 기울어진 정도에 따라 영상 속의 교통표지판은 다르게 보여 후반부의 식별이 어려워진다. 따라서 아핀 변환을 통해 선택된 부분을 데이터베이스와 비슷한 크기로 표준화하면 식별 과정에서 인식율이 좋아진다. 마지막으로 배경을 제거하여 식별 준비를 하면 그림 1(i)와 같은 결과를 얻는다.



(g) Triangle image (h) Affine transformation (i) Final image
Fig. 1. Sequence of a traffic sign detection

그림 1. 교통표지판 탐지 과정의 예

III. 교통표지판 식별 과정

제안한 알고리즘에서는 교통표지판 후보 탐지후 식별과정을 거치므로 매우 효과적으로 키포인트 생성과 매칭 과정을 진행한다. 교통표지판 탐지 과정을 통해 후보 영상을 얻으면 데이터베이스와 비교하여 식별하는 과정을 거친다. 본 논문에서는 식별을 위한 특징 추출을 위해 SURF[10]와 ORB[11]를 사용하였다. 제안하는 알고리즘에 따라 교통표지판을 인식하는 과정의 예가 그림 2와 3에 나타나 있다.

SURF는 SIFT(Scale Invariant Feature Transform)보다[16,17] 계산 속도는 빠르면서 정확도는 비슷하게 유지하는 방식이다. 2D Haar 웨이브릿(wavelet) 응답의 합을 계산하는 방식이다. 기본적인

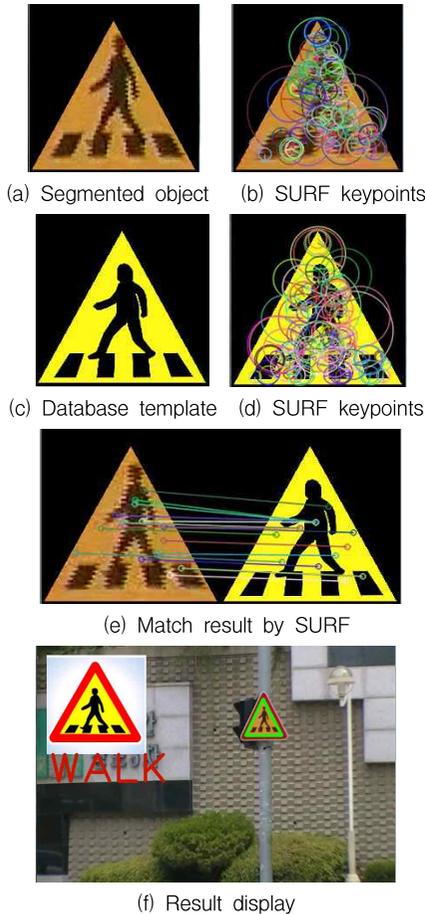


Fig. 2. Traffic Sign Identification using SURF
 그림 2. SURF를 이용한 교통표지판 식별

영상 특징을 얻기 위해 SURF는 Hessian 블록(blob) 탐지기의 determinant의 Haar 웨이브릿 근사법을 이용한다. 두 영상간의 매치 포인트를 찾는 작업은 각 영상에 대한 키포인트 리스트를 찾고 내적(dot product)이나 SSD(Sum of Square Difference) 비교를 통해 서술자(descriptor) 벡터를 매칭하여 이루어진다.

그림 2(a)와 (b)는 입력된 영상에서 탐지된 교통표지판 후보 영상과 SURF 알고리즘에 의해 생성된 키포인트를 나타낸다. 그림 2에서 입력 영상은 720 x 480 크기의 동영상이고 탐지 과정에서 추출된 도형은 아핀 변환에 의해 200 x 200의 표준 크기로 바뀐다. 이 때 추출된 키포인트는 155개이다. 그림 2(c)와 (d)는 데이터베이스의 템플레이트와 그에 대해 SURF 알고리즘을 이용해 키포인트를 찾은 결과이다. 그림 2(d)에서 추출된 키포인트의 수는 88개이다. 입력 영상에서 얻은 도형은 확대된 형태여서 가장다리가 매끄럽지 않아 더 많은 키포인트가 추출되었다. 두 키포인트 리스트에 대해 서술자를 비교하여 매칭 포인트를 찾은 결과가 그림 2(e)에 나타나 있다. 매칭 포인트의 수는 15개이다.

ORB는 FAST 키포인트 탐지와 BRIEF 키포인트 서술(description) 알고리즘의 조합이다. FAST와 그 변형 방식들은 시각적 특징과 매치되는 실시간 시스템

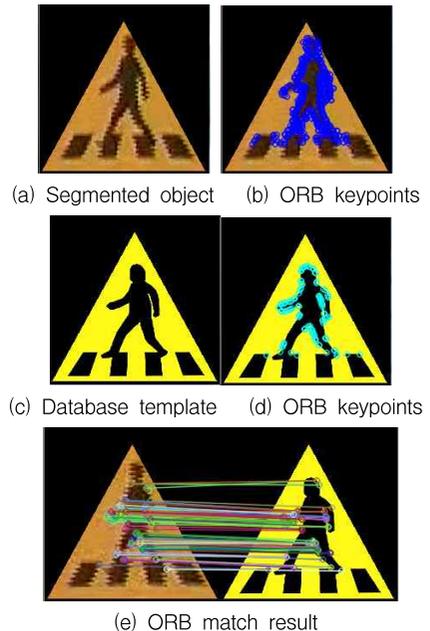
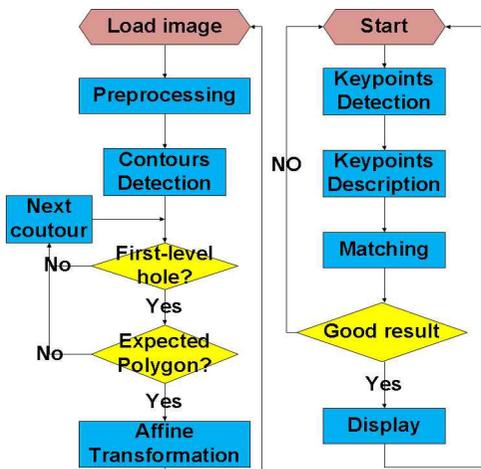


Fig. 3. Traffic Sign Identification using ORB
 그림 3. ORB를 이용한 교통표지판 식별

의 키포인트를 찾기 위해 선택하는 방식이다. BRIEF는 스무드(Smoothed) 효과를 적용한 이미지에서 픽셀 사이의 단순한 이진(binary) 테스트를 사용하는 최근에 발표된 특징 서술자이다. 그 성능은 조명효과, 블러(blur), 그리고 원근 왜곡(perspective distortion) 등에 강한 특성을 포함하여 여러 가지 면에서 SIFT와 비슷하다. 그림 3(b)에서 입력 도형으로부터 추출된 키포인트는 293개이고 그림 3(d)의 데이터베이스 템플레이트에서 추출된 키포인트는 183개이다. 마찬가지로 입력 도형의 키포인트 수가 더 많다. 그림 3(e)에서 매칭 포인트의 수는 82개로 SURF보다 훨씬 많음을 알 수 있다.

그림 4에 제안한 알고리즘을 탐지 단계와 식별 단계로 나누어 순서도로 나타내었다. 그림 4(b)에서는 SURF와 ORB가 각각 적용된다.



(a) Sign Detection Step (b) Sign Identification Step
Fig. 4. Flow chart for traffic sign recognition algorithm

그림 4. 교통표지판 인식 알고리즘의 순서도

IV. 성능 분석 결과

제안한 알고리즘을 OpenCV 라이브러리를 이용하여 소프트웨어로 구현하였다. 실행 환경은 Intel Core2 2.83GHz CPU와 4GB의 메모리이고 단일 코어만을 사용하였다. 성능 분석에는 5개의 동영상과 정지영상을 합쳐 1280개의 영상이 사용되었고 영상 크기는 640 x 480에서 1510 x 950까지이다.

그림 5에 나타난 5 종의 교통표지판에 대해 표1에 탐지시간이 나타나 있다. 그림 문자에 따른 차이를 비교하기 위해 삼각형의 표지판만을 이용하였다. 표

1에 나타난 값은 각 프레임별로 얻은 시간의 평균값이며 동일한 교통 표지판을 포함하는 경우 프레임간 탐지 시간의 편차는 크지 않다. 탐지 시간 24 - 31ms 중에서 색 분할 시간이 11 - 15ms로 절반 정도를 차지하고 아핀 변환이 6 - 9 ms로 30%, 다각형 근사법이 나머지 20%의 시간을 차지한다. 색 분할은 전체 영상에 대해 진행되므로 가장 많은 시간이 소요되고 영상 크기가 클수록 많은 시간이 요구된다. 다각형 근사법은 색 분할에서 얻은 후보군에 대해 진행되므로 후보가 많을수록 많은 시간이 필요하다. 이렇게 얻은 다각형에 대해 교통표지판과 일치하는 도형을 찾는다. 아핀 변환은 이 조건에 맞는 도형에 대해서만 실시한다. 아핀 변환 후에는 데이터베이스의 템플레이트와 같은 크기의 영상이 만들어진다.

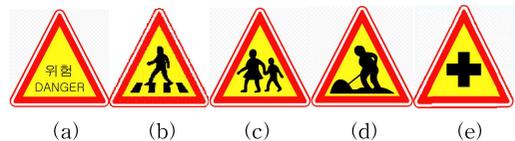


Fig. 5 Examples of traffic signs for the performance analysis (a) Danger (b) Pedestrian (c) Children (d) Road works (e) Crossroad
그림 5. 성능 분석을 위해 이용된 교통표지판의 예 (a) Danger (b) Pedestrian (c) Children (d) Road works (e) Crossroad

Table 1. Detection time for traffic signs in Fig. 5 [ms]
표 1. 그림 5의 교통표지판 탐지시간 [ms]

Database Template	Color Threshold	Polygon Approximation	Affine Transformation	Total
Danger	14.54	6.43	8.41	29.38
Pedestrian	11.26	5.36	7.28	23.9
Children	14.41	6.88	8.48	29.77
Road works	14.51	6.51	5.8	26.82
Crossroad	14.89	7.06	8.94	30.89

교통표지판 식별은 SURF와 ORB의 두 가지 방식으로 진행되는데 키포인트 생성/서술과 매칭 과정이 완전히 다르므로 실행시간에도 큰 차이를 보인다. 표 2는 두 가지 방식의 실행 시간 비교 결과가 나타나 있다. 키포인트 생성 및 서술은 ORB가 4-7배 정도 빠르고 키포인트 매칭은 SURF가 2-3배 정도 빠르다. 3은 교통표지판 식별 및 전체 인식 시간을 나타내고 있다. 전체적인 식별 시간은 ORB 방식이 패턴표에 따라 1.3 - 5배 정도 빠른 것을 알 수 있다.

Table 2. Identification time for traffic signs in Fig. 5 [ms]

표 2. 그림 5의 교통표지판 식별 시간 [ms]

Database Template	SURF		ORB	
	Detection Description	Matching	Detection Description	Matching
Danger	35.90	8.60	6.05	27.77
Pedestrian	25.97	5.43	4.61	14.77
Children	35.61	7.81	4.86	22.73
Road works	26.25	4.96	5.87	15.60
Crossroad	23.65	1.20	3.44	1.91

가장 차이가 많이 나는 교차로(crossroad)의 경우 십자 형태의 단순한 패턴이고 가장 시간차가 적은 위험 표지의 경우 영문과 국문이 글자가 포함되어 패턴으로는 가장 복잡하다. 따라서 패턴이 단순할수록 ORB 알고리즘이 큰 효과를 발휘하는 것을 알 수 있다. 표4에 실험에 사용된 교통표지판의 입력 영상과 데이터베이스 영상에 대해 생성된 키포인트와 매칭 결과가 나타나 있다. 프레임별로 키포인트 수가 완전히 동일하지는 않으나 그 차이가 2-3개 이내이고 키포인트 수와 인식 시간의 차이를 보는 것이 목적이므로 평균적인 키포인트 수를 갖는 특정 프레임에 대한 결과를 이용하였다. 키포인트 수가 많을수록 계산 시간이 오래 소요되고 매칭 포인트 수도 많음을 알 수 있다. 탐지와 식별을 모두 포함하여 인식 시간을 보면 SURF 알고리즘을 이용한 경우 56 - 74ms, ORB 알고리즘을 이용한 경우 36 - 63ms로 ORB 알고리즘을 이용하면 15 - 40% 정도의 성능 향상을 알 수 있다.

표 5에는 여러 가지 교통표지판의 인식률이 나타나 있다. 인식률은 87.5 - 99.7%로 평균 96.1%이다. 표 3 - 5를 비교해 보면 SURF의 경우에는 일정한 수 이상의 매칭이 이루어지면 일정 수준의 인식률을 얻을 수 있음을 알 수 있다. ORB의 경우에는 매칭 포인트의 수와 인식률이 어느 정도 상관관계를 갖고

Table 3. Recognition time for traffic signs in Fig. 5 [ms]

표 3. 그림 5의 교통표지판 인식 시간 [ms]

Database Template	Detection	Identification		Total	
		SURF	ORB	SURF	ORB
Danger	29.38	44.50	33.82	73.88	63.2
Pedestrian	23.9	31.4	19.38	55.3	43.28
Children	29.77	43.42	27.59	73.19	57.36
Road works	26.82	31.21	21.47	58.03	48.29
Crossroad	30.89	24.85	5.35	55.74	36.24

Table 4. Numbers of generated and matched keypoints

표 4. 생성된 키포인트 수와 매칭 결과

Road sign	SURF			ORB		
	DB	Object	Match	DB	Object	Match
Danger	163	138	20	361	374	143
Pedestrian	88	81	17	183	222	87
Children	115	99	51	265	316	220
Road works	99	70	38	244	191	126
Crossroad	30	23	6	46	58	32

증가하다가 어느 수준 이상이 되면 인식률이 포화됨을 알 수 있다. 위험 표지처럼 패턴이 복잡하여 연산 시간이 오래 걸리면 인식률이 높고 교차로처럼 패턴이 단순하여 키포인트와 매칭 포인트 수가 작아 연산 시간이 짧은 경우에는 인식률도 낮은 것을 알 수 있다. 이를 통해 명확한 비례 관계가 성립하지는 않지만 대체적으로 패턴이 복잡할수록 키포인트 생성과 서술, 매칭에 많은 시간이 필요한 대신 매칭되는 키포인트가 많으면 인식이 잘 되고 패턴이 단순하면 키포인트 생성 및 서술, 그리고 매칭 시간이 짧고 매칭되는 키포인트가 작아 인식이 잘 안 됨을 알 수 있다.

Table 5. Recognition ratios of traffic signs

표 5. 교통표지판 인식률

Database	SURF	ORB
Danger	99.7 %	99.6 %
Pedestrian crossing	98.8 %	92.7 %
Watch out for children	98.3 %	98.8 %
Road works	96.2 %	98.4 %
Crossroad	87.5 %	88.6%

제안한 알고리즘의 동작 검증의 위해 비오는 야간에 다양한 형태의 교통표지판을 인식한 실험 결과가 그림 6에 나타나 있다. 가랑비가 내리는 야간에도 조명에 의해 교통표지판이 보이는 경우에는 인식이 잘 되는 것을 확인할 수 있고 오각형이나 원의 형태도 인식함을 보이고 있다. 원형 표지판의 경우 Hough 변환에 필요한 시간이 다각형 근사법보다 시간이 오래 걸려 표지판 탐지시간이 20% 정도 증가한다. 그러나 식별 시간까지 포함한 인식 시간은 8-10% 정도 증가한다. 현재 원형 인식에 이용하는 Hough 변환은 일반적인 원형을 찾도록 하고 있어 교통표지판의 특성을 활용하여 이에 최적화하면 연산 시간을 감소시킬 수 있다.



Fig. 6. Experimental results of recognition for various shapes of traffic signs in a rainy night.

그림 6. 비오는 저녁에 다양한 형태의 교통표지판을 인식한 실험 결과

교통표지판 인식 관련한 기존 연구에서는 인식물에 중점을 두어 계산 시간에 대해서는 크게 신경을 쓰지 않았다. 따라서 계산 시간과 관련된 자료를 찾기가 어렵다. CPT를 이용한 특징 추출의 경우 0.7초에서 6.3초의 결과를 보여주고 있으나 비디오 입력 영상에서 교통표지판을 인식하는 방식이 아니어서 직접적인 비교는 어렵다[3]. 또한 GPGPU를 이용하여 60ms의 인식 시간을 보여준 결과도 있으나 CPU만의 결과와는 공정한 비교가 안 된다[18]. 표 6에 본 논문과 가장 유사한 방식으로 SIFT를 이용한 식별 단계의 성능 비교 결과가 나타나 있다. 비교 영상과 조건이 달라 정확한 비교는 어렵지만 비슷한 인식물을 보이면서 인식 시간은 제안한 알고리즘이 훨씬 우수한 것을 알 수 있다. 또한 전체 연산 시간이 60ms 이내로 15fps의 HD 수준의 동영상에 대해 실시간 동작이 가능하다.

Table 6. Comparison with other work
표 6. 기존 결과의 비교

Algorithm	[1]	Proposed	
	SIFT	SURF	ORB
Image size	1024 x 768	1510 x 950	
Platform	Intel Core2 2.4GHz	Intel Core2 2.86GHz	
Detection time	80 ms*	28 ms	
Identification time	100 ms	35 ms	21.5 ms
Recognition rate	95%	96%	

* Excluding image loading time

V. 결론

본 논문에서는 지능형 자동차를 위한 교통표지판 인식 알고리즘을 제안한다. 제안한 알고리즘은 교통

표지판 후보 영역을 찾는 탐지 단계와 데이터베이스의 자료와 비교하여 교통표지판을 식별하는 단계로 나누어진다. 탐지 단계에서는 교통표지판의 형태와 색 정보를 이용하여 효과적으로 후보 영역을 찾고 식별 단계에서는 이를 SURF 또는 ORB 알고리즘을 이용하여 데이터베이스의 자료와 비교하여 일치 여부를 확인한다. 제안한 알고리즘은 연산 시간 감소에 중점을 두어 실시간 동작에 근접함을 목표로 한다. OpenCV 라이브러리를 이용하여 구현한 결과 패턴의 복잡도에 따라 36 - 74ms 정도의 인식 시간이 소요되었으며 ORB 알고리즘이 SURF 알고리즘에 비해 전체 인식 과정은 15 - 40% 정도, 식별 단계는 1.3 - 5배 정도 성능이 우수함을 확인하였다. 본 연구결과는 지능형 자동차에서 운전자와 자동 운전 시스템에게 주행환경 정보를 제공하여 안전 운전에 도움을 줄 것으로 기대된다. 또한 국내처럼 네비게이션 지도와 주변 정보가 상세 제공되는 경우에는 이와 결합하여 인식 속도 및 인식률을 더욱 개선시킬 수 있을 것으로 기대된다.

References

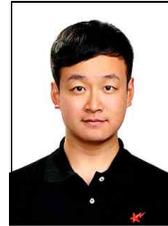
- [1] FeiXiang Ren, Jinsheng Huang, "General Traffic Sign Recognition by Feature Matching", 24th International Conference Image and Vision Computing New Zealand, pp.409 -414, 2009
- [2] S. Maldonado-Bascon, S. Lafuente Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, F Lopez-Ferrerias, "Road-Sign Detection and Recognition Based on Support Vector Machines", Intelligent Transportation Systems, Vol.8, pp. 264-278, 2008
- [3] K. Zhang, "Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature", Intelligent Transport Systems, IET, Vol. 6, pp. 282-291, Sep. 2012
- [4] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani, "Robust road sign detection and recognition from image sequences", in IEEE Intelligent Vehicles Symposium, pp. 278-283, 1994.
- [5] C. Bahlmann, Y. Zhu, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information", IEEE Intelligent Vehicles Symp. pp. 255-260, 2005
- [6] A. Lorsakul, J. Suthakorn, "Traffic sign recognition using neural network on OpenCV:

Toward intelligent vehicle driver assistance system", J. Intelligent Service Robotics, 2009

- [7] M. K. Hu, "Visual pattern recognition by moment invariant", IEEE Trans. Inf. Theory, Vol. 28(8), pp. 179-187, 1962
- [8] R. Mukundan, S. H. Ong, P. A. Lee, "Image analysis by Tchebichef moments", IEEE Trans. Image Process., Vol. 10(9), pp. 1357 - 1364, 2001
- [9] H. Fleyeh, M. Dougherty, D. Aenugula, S. Baddam "Invariant road sign recognition with fuzzy ARTMAP and Zernike moment", Proc. 2007 IEEE Intelligent Vehicles Symp., pp. 31 - 36, June 2007, Istanbul, Turkey
- [10] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded up robust features(SURF)", Computer Vision Image Understanding, pp.346-359, 2008
- [11] E. Rublee, V. Rabaud, "ORB: An efficient alternative to SIFT or SURF", Computer Vision (ICCV), pp. 2564-2571, Nov. 2011
- [12] D. Park, H. KO, "Fog-degraded Image Restoration Using Characteristics of RGB Channel in Single Monocular Image", International Conference on Consumer Electronics (ICCE), pp. 141-142, Jan. 13-16, 2012, Las Vegas, USA.
- [13] R. Courant, H. Robbins, and I. Stewart. "What Is Mathematics An Elementary Approach to Ideas and Methods", Oxford University Press, New York, 1996
- [14] D. Douglas, T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer 10(2), pp. 112-122, 1973
- [15] D. Zuo, "Application of affine transformation in traffic sign detection", Intelligent Control and Information Processing (ICICIP), pp. 277-280, 2010
- [16] T. Lindeberg, "Scale-space for discrete signals", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 12(3), pp. 234-254, 1990
- [17] D. Lowe. "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision IJCV, Vol. 60(2), pp. 91-110, January 2004
- [18] V. Glavtchev, P. M. Ozelik, "Feature-Based Speed Limit Sign Detection Using a Graphics Processing Unit", 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 195 - 200, 2011

BIOGRAPHY

Ding Dajun (Student member)



2010 : BS degree in Electronic and Information Engineering, Shandong University of Science and Technology.

2011~present : Graduate student in MS degree in Dept. of Electronic Engineering,

Soongsil University

<Research Area> Image Pattern Recognition

Lee Chanho (Member)



1987 : BS degree in Electronic Engineering, Seoul National University.

1989 : MS degree in Electronic Engineering, Seoul National University.

1994 : Ph.D degree in Electrical Engineering, University of California Los Angeles

1994~1995 : Senior Research Engineer, Samsung Electronics.

1995~present : Professor, School of Electronic Engineering Soongsil University

<Research Area> SoC on-chip-network, Memory controller, Image Pattern recognition, SoC design methodology, Implementation of video codec