적응 알고리즘을 이용한 자동차 번호판 인식 시스템 개발에 대한 연구

장 승 주†

요 약

자동차 번호판 인식 시스템에서 가장 중요한 요소는 자동차 이미지 영역에서 번호판 영역을 추출, 추출된 영역에서 문자 추출, 추출된 문자의 인식등의 과정이다. 본 논문은 자동차 번호판 인식 과정에서 적응 알고리즘을 이용하여 보다 정확한 인식이 될 수 있도록 한다. 본 논문에서 사용하는 적응 알고리즘은 기존의 방식과는 달리 특징한 알고리즘을 이용한 인식을 하지 않고 다양한 알고리즘을 이용한 인식 결과의 조합으로 최적의 해법을 찾는다 번호판 인식을 위한 적응 알고리즘은 원형 정합 알고리즘, 벡터 알고리즘, 세선화 알고리즘 등이다. 적응 알고리즘을 이용한 실험 결과 자동차 이미지에 대해서 90% 이상 인식이 가능함을 확인할 수 있었다.

A Study of Character Recognition using Adaptive Algorithm at the Car License Plate

Seung-Ju Jang†

ABSTRACT

In the recognition system of car license plate, it is very important to extract the character from the license plate and recognize the extracted character In this paper, I use the adaptive algorithm to recognize the character of license plate image. The adaptive algorithm is compounded of thinning algorithm, template matching algorithm, vector algorithm, and so on. The adaptive algorithm was used to recognize the character from license image. In the result of experiment, character recognition is about up to 90% with the adaptive algorithm for the character region.

1. 서 론

최근들어 정보 통신 분야의 급속한 기술 확산으로 인하여 자동 인식 시스템 분야의 기술이 중요한 부분으로 대두되기 시작하고 있다. 특히 자동차 번호판 자동 인식 시스템은 우리의 일상 생활에 활용 가치가 높다. 번호판 자동 인식 시스템은 주차장 관제 시스템, 통행 요금 징수, 교통량 조사, 버스 전용 차로 위반 등과 같은 시설의 자동화에 크게 기여할 수 있는 분야이다[1-3]. 현재 전세계적으로 영상 처리에 대한 다양한 연구가 이루어지고 있다. 영상 처리 분야는 무한한 연구 주제를 가지고 있는 분야이다. 그 중에서 업무 자동화로 쉽게 접근할 수 있는 분야가 자동차 번호판에 대한 영상 처리이다

많은 국가에서 자동차 번호판을 자동으로 인식하기 위한 작업에 박차를 가하고 있다. 그러나 많은 연구에도 불구하고 인식율이 그리 뛰어난 연구 결과들이 발표되고 있지 못하다. 대부분의 연구 접근 방식이 획일적인 방식을 채택하고 있다는 것과 특정한 환경에 유

※ 이 논문은 1999년 중기청 산학연 공동기술개발 지역컨소시엄 과제의 지원을 받아 수행하였음
† 정 회 원 동의대학교 컴퓨터공학과 교수
논문접수 : 2000년 6월 19일, 심사완료 : 2000년 9월 27일

리한 방법들을 적용하는 것으로 그치고 있다.

실용적인 번호판 인식 시스템 개발에 있어서 중요한 부분은 두가지로 요약할 수 있다. 첫 번째는 번호판 영역을 정확히 추출하는 것이고 둘째는 추출된 번호판 영역을 가지고 문자 및 숫자를 인식하는 것이다.

국내에서 자동차 번호판에 대한 연구는 1980년대 말부터 본격적으로 이루어졌다. 이치화를 통해 윤곽선을 검출한 이진 영상에서 휴변환(Hough transform)으로 번호판의 수직·수평 경계선을 검출하는 방법[6,12-14], 명암 값의 변화에 의한 숫자 영역을 검출한 후 번호판 영역으로 확대해 나가는 방법[1, 2]. 명암 값의 편차가 많은 부분을 대상으로 휴변환을 이용하는 방법[2, 3], 히스토그램 정규화 과정을 거쳐 이치화하고 숫자 영역을 검출하는 방법[4, 5]등이 있다. 또한 투영(Projection)에 의한 방법, 원형 정합(template matching)에 의한 방법, 영역 분할에 의한 방법, 명암도 변화를 이용한 방법 등이 있다[2, 7-11]. 휴 변환으로 번호판의 수직, 수평 성분을 추출하는 방법은 메모리의 양과 휴변환에 처리시간이 많이 소요된다는 단점이 있다.

대부분 자동차 번호판 인식 시스템은 크게 자동차 번호판 영역 추출과 인식과정으로 나눈다. 정확한 자동차 인식을 위하여 가장 먼저 선행되어야 하는 것이 자동차 번호판 영역의 정확한 추출과 추출된 번호판 영역에 대한 문자 인식 과정이다. 본 논문은 자동차 번호판 인식 과정에서 문자 인식 부분에 초점을 맞춘다. 기존 논문의 경우 문자 인식 방법은 특정한 환경에 유리한 경우에 한정해서 이루어지고 있다. 그리고 대부분의 경우가 원형 정합 방법, 신경망을 이용한 방법 등 특정한 알고리즘에 부분적인 결함들의 해결에 주력하고 있다[1, 3, 4].

본 논문은 기존 연구의 이러한 문제점을 해결하기 위하여 적응 알고리즘을 이용하여 자동차 번호판 문자 인식을 한다. 적응 알고리즘을 이용한 자동차 번호판 문자 인식 방법은 문자 인식의 정확도를 높이기 위하여 원형정합 알고리즘, 세선화 알고리즘, 벡터 알고리즘 등 여러가지 알고리즘을 병행, 적용하여 모든 가능한 조합 형태의 결과를 가지고 정확한 판단을 하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 적응 알고리즘을 이용한 자동차 번호판 인식 알고리즘, 3장에서는 구현 및 실험, 4장 결론의 순으로 언급한다.

2. 적응 알고리즘을 이용한 자동차 번호판 인식 알고리즘

적응 알고리즘을 이용한 자동차 번호판 인식 알고리즘의 전체적인 흐름은 다음 (그림 1)과 같다. 전체 흐름도는 아래 그림과 같이 진행되며 각 항목은 하나의 Class로 이루어 지며 계층적으로 생성한 데이터를 하위 계층으로 전달한다.

(그림 1) 자동차 번호판 인식 흐름도

적응 알고리즘을 이용하여 자동차 번호판 인식을 위하여 다음과 같은 자동차 번호판에 대한 기본적인 조건들이 수반되어야 한다. 자동차 이미지를 Load하여 원 이미지와 Gray 이미지, Binary 이미지로 변환하여 각 Buffer에 저장한다. 원 이미지는 256 Color의 BMP 파일이다.

•Gray 이미지 변환

Gray 이미지는 원 이미지의 정보를 R, G, B의 값으로 각각 추출하여 Gray 값으로 변환한다. Gray 변환식은 다음과 같다.

$$Gray_Value = (R*30 + G*59 + B*11)/100$$

Code

```
R = (OrgCar[i + j*width]>>16)&0x000000ff;
G = (OrgCar[i + j*width]>>8)&0x000000ff;
B = OrgCar[i + j*width]&0x000000ff;
GrayCar[i + j*width] = (R*30+G*59+B*11)/100;
```

OrgCar

8bit	R(8bit)	G(8bit)	B(8bit)

원 이미지의 데이터는 RGB 값이 모두 붙어 있는

32Bit 값이므로 RGB 값은 각각 8bit 식 Shift 연산을 통해 얻을 수 있다.

● **Binary 이미지 변화**

Gray 이미지의 각 픽셀의 값은 0~255까지의 값을 갖는다. 0~255까지의 값 중에서 180을 기준으로 180 이하인 값은 0으로 180 이상인 값은 255로 변환하여 Binary 이미지로 변환한다.

```
Code    if(GrayCar[i+j*width]>180)
        BinaryCar[i + j*width] = 255;
        else BinaryCar[i + j*width] = 0.
```

2.1 자동차 번호판 영역 추출

자동차 번호판의 영역을 추출하는 방법으로 명암 벡터 값을 이용하였다 명암 벡터란 Gray 이미지의 각 픽셀의 인접한 두 픽셀의 차이를 그레프 형태로 나타낸 것으로 번호판의 특성상 전체 자동차의 이미지 중에서 번호판의 영역의 명암 벡터 값이 높은 수치를 나타낸다.

번호판의 영역은 전체 크기는 가로 세로 비율이 2 : 1 크기이며 번호판의 외각은 흰색(자가용) 이나 청색(영업용)의 테두리가 있으며 글자의 색상은 흰색(자가용)이나 청색(영업용)으로 되어있다. 번호판의 색상은 자가용의 경우 녹색이며 영업용의 경우 노란색으로 되어있다 이와같이 번호판 영역과 글자 영역의 색상 대비가 명확함으로 Gray 이미지나 Binary 이미지에서의 색상 대비 수치는 높다. 이런 특성으로 번호판 영역의 명암 벡터 값은 번호판 영역 내에서 글자와 번호판 색상의 대비가 큼으로 수치가 높게 나타난다. 아래의 (그림 2)를 보면 명암 벡터가 어떻게 나타나는지 알 수 있다. 군용, 외교관용 차량등 특수한 차량에 대한 번호판은 본 논문에서는 일단 고려하지 않는다 특수한 차량은 차량 숫자가 적고 본 논문에서 제안하는 적응 알고리즘을 이용할 경우 쉽게 접근이 가능하기 때문에 향후 연구 분야로 남겨둔다.

(그림 2)에서 처럼 번호판 영역의 명암 벡터 값은 높게 나타난다. 그러므로 명암 벡터 값이 높게 나타나는 영역이 번호판의 영역이 된다. 그러나 명암 벡터 값이 평균이상 높게 나타나는 영역이 하나 이상 존재할 수 있다. 자동차의 앞쪽은 번호판 이외의 여러 가지 장식에 의해 번호판 영역의 명암 벡터만큼 수치가 높게 나타나는 부분이 다수 존재하기 때문이다

(a) 원 이미지

(b) gray 이미지의 명암벡터

(c) binay 이미지와 명암 벡터

(그림 2) Binary 이미지와 명암 벡터 그래프

위 그림처럼 번호판 영역보다 위쪽의 차량 장식 부분이 명암 벡터 값이 더 높게 나타난다 이런 차량 이미지와 같은 경우가 있을 수 있으므로 명암 벡터 값만으로는 번호판 영역을 정확히 추출 할 수 없다. 그래서 명암 벡터 값에서 찾은 영역을 모두 저장하고 그 값 D들 중에서 번호판 색상의 값을 참가하여 번호판 후보들 중에서 번호판 색상의 값은 많이 포함하고 있는 영역을 추출한다. 그리고 명암 벡터의 후보 중 전체 녹색 값이 더 많은지 비교하고 그 영역의 가로 세로 비율이 2 : 1이 되는지 판단하여

최종 번호판 영역의 좌표를 찾아낸다.

2.2 문자 영역 추출

추출된 번호판 영역에서 문자 영역을 추출하는 과정이다. 번호판의 문자는 2줄로 이루어지며 윗줄에는 지역문자와 숫자, 아래 줄에는 문자 1자와 숫자 4자로 이루어진다. 아래 줄의 문자의 경우 문자 1자와 숫자 4자 총 5자의 문자로 고정되나, 윗줄의 문자는 지역문자 2자 와 숫자 1자가 올 때가 있고 숫자 2자가 올 때도 있다.

그래서 윗줄의 문자가 총 3자 인지 4자 인지 판별하여야 하는데 매우 까다로운 작업이다 또 문자 영역의 좌표를 추출하는데는 잡음에 매우 민감하므로 전 단계에서 2진화 과정이 매우 중요하다.

먼저 아래의 그림과 같이 문자의 영역은 번호판의 상하를 기준으로 중심에서 3등분으로 나누어진다 그래서 번호판 좌표의 1/2를 기준으로 좌우로 픽셀을 Scan하면서 중심이 되는 수평축을 찾는다. 상위 글자는 위에서 1/3 정도 차지하고 아래 글자는 2/3 정도 차지하므로 상하 글자의 경계선을 찾을 수 있다.

상하 경계선에서 위의 문자와 아래 문자의 Y좌표를 추출한다. 전체 문자는 번호판의 중심에서 균일하게 분포하므로 중심에서 좌우로 Scan하면서 각 문자의 좌표를 추출할 수 있다. 위쪽 문자의 경우 문자와 함께 문자 좌우로 번호판 나사가 이진 과정에서 남아 있을 수 있다. 그래서 위쪽 문자의 경우 나사가 문자로 잘못 인식될 수 있으므로 고려사항으로 참고해야 한다

그래서 위쪽 문자는 중심에서 지역문자 2자를 먼저 추출하고 우측으로 Scan하면서 숫자 영역을 찾는데 한 문자일수도 있고 두 문자일수도 있으므로 지역문자와 문자의 크기를 비교하면서 문자인지 나사인지 분별할 수 있다.

(그림 3) 번호판에서 문자 영역 추출

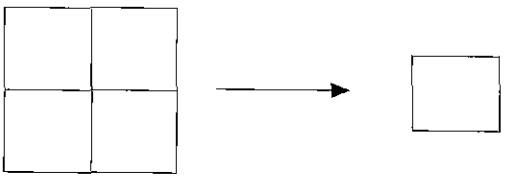
2.3 문자 인식

● 숫자 인식

숫자의 인식은 원형정합을 이용하여 저장된 숫자의 패턴과 일치하는 픽셀의 수치를 합산하여 가장 높은 수치를 나타내는 숫자를 선택하는 방법이다. 먼저 32 x 32 크기의 문자 데이터를 16 x 16 크기로 축소한다 이 과정에서 문자 경계선에 존재하는 잡음을 축소시켜 인식율을 높인다.

크기 축소 Code

```
for(int i = 0;i<32;i = i+2)
        for(int j = 0,j<32,j = j+2)
        {
                if((Chx32[ch][i][j] + Chx32[ch][i+1][j] +
                Chx32[ch][i][j+1] - Chx32[ch][i+1][j+1])
                > value)
                {
                        Numx16[ch][i/2][j/2] = 1,
                }
                        else Numx16[ch][i/2][j/2] = 0;
        }
```



위 그림처럼 4개의 픽셀에서 1개의 픽셀로 축소하는데 4개의 픽셀 중 하나라도 픽셀의 값이 1인 픽셀이 존재하면 축소될 픽셀의 값은 1로 찍는다.

int UpNum[2][16][16] : 위쪽 숫자의 이미지 데이터 저장
int DnNum[4][16][16] : 아래쪽 숫자의 이미지 데이터 저장
int UpNumRe[2] : 인식된 위쪽 숫자의 코드 저장
int DnNumRe[4] : 인식된 아래쪽 숫자의 코드 저장
int NumData[10][16][16] : 파일의 숫자 폰트 저장

아래의 Code는 숫자 폰트의 데이터(NumData[])와 입력된 숫자 이미지의 데이터 (UpNum[])을 픽셀마다 비교하면서 일치하는 수치(hit_count)를 저장한다.

```
for(sy=0,sy<=15,sy++)
{
        for(sx=0,sx<=15,sx++)
        {
                if(UpNum[i][sx][sy] == 1)
                {
                        one_count++,
                        if(NumData[index][sx][sy] ==
                          UpNum[i][sx][sy])
                        hit_count[i][index]++;
```

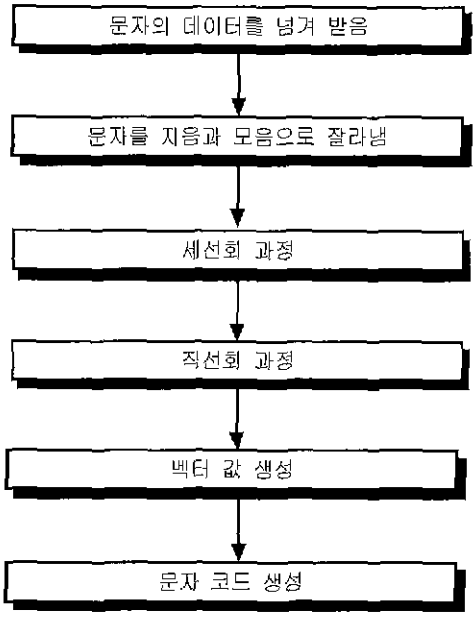
```
            }
        }
}
```

입력된 숫자 이미지는 숫자 폰트와 모두 비교되어 가장 수치가 높은 숫자가 선택된다.

```
for(i=0; i<2;i++) {
        for(index=0, index<10; index++) {
                if(hit_rate[i][index] > max) {
                        max = hit_rate[i][index];
                        mindex=index;
                }
        }
        max = 0;
        CarNumber[i] = mindex;
        mindex = 0;
}
```

● **문자 인식**

자동차 번호판의 문자의 경우 위쪽의 자음 문자는 광역시와 각 지방의 도의 이름으로 구성되고 아래쪽의 문자는 'ㄱ' 부터 'ㅎ' 까지의 자음과 모음 'ㅏ', 'ㅓ', 'ㅗ', 'ㅜ', 'ㅣ', 'ㅡ' 6가지로 구성되므로 단순한 형태이다. 그래서 문자의 정확한 특징 추출이나 인공지능 기법 없이 인식이 될 수 있다. 문자의 인식은 문자의 벡터 값을 이용하는데 문자의 벡터 값은 'ㄱ'의 경우 좌에서 우로 위에서 아래로 보통 문자를 쓸 때 손으로 쓰는 방향을 데이터로 변환하여 사용한다 위의 (그림 4)는 문자 인식을 위한 과정을 나타낸다.



(그림 4) 문자 인식을 위한 과정

● **모음의 인식**

문자의 데이터에서 자음을 잘라내고 나머지 부분에는 모음이 저장되어 있다. 이 데이터를 가지고 모음의 패턴을 분석하여 인식한다. 아래쪽 문자에만 적용하기 때문에 모음의 종류가 단순한 6가지 종류이므로 패턴 분석으로 가능하다.

● **지역문자 인식**

지역 문자의 경우 자음 부분만 인식하여 두 글자의 자음 조합에 따라 지역문자를 인식한다. "서울"의 경우 앞 글자의 'ㅅ'과 뒷글자의 'ㅇ'으로 "서울"이라고 판단할 수 있다 이것은 지역 문자의 종류가 한정되어 있어 가능하다. 이런 방법으로 "부산"의 경우 'ㅂ'과 'ㅅ', "경북"의 경우는 'ㄱ'과 'ㅂ'과 같이 나타나므로 모든 지역 문자의 조합은 유일하다. 따라서 지역문자의 앞 글자와 뒷글자의 조합으로 인식 가능하다. 모든 지역에 대한 문자 조합을 <표 1>로 나타내면 다음과 같다.

● **아래쪽 문자 인식**

아래쪽의 문자는 자음 11자와 모음 6자로 구분된다 위에서 기술한 방법으로 자음과 모음을 분리하여 인식하고 인식된 Code값의 조합으로 문자를 인식할 수 있다.

3. 구현 및 실험

3.1 번호판 인식

번호판 인식 알고리즘 구현을 위한 주요 멤버 변수와 주요 멤버 함수를 다음 <표 2>, <표 3>와 같다. <표 2>은 자동차 이미지 영역에서 자동차 번호판을

〈표 1〉 지역문자의 첫글자

지역	서울	부산	대구	인천	광주	대전	울산	경남	경북	전남	전북	강원	경기	제주
문자조합	ㅅ,ㅇ	ㅂ,ㅅ	ㄷ,ㄱ	ㅇ,ㅊ	ㄱ,ㅈ	ㄷ,ㅈ	ㅇ,ㅅ	ㄱ,ㄴ	ㄱ,ㅂ	ㅈ,ㄴ	ㅈ,ㅂ	ㄱ,ㅇ	ㄱ,ㄱ	ㅈ,ㅈ

〈표 2〉 번호판 인식을 위한 주요 멤버 변수

주요 멤버 변수	설 명
int X1,Y1,X2,Y2	최종 번호판 영역의 좌표
int *grpx,*grpy	명암 맥터 그레프 Buffer
int Green_sum	번호판 영역의 녹색 값의 수치
int rfx[20],rex[20]	명암 맥터의 후보 영역을 찾기 위해 사용하는 좌표 기억 Buffer
unsigned char *GreenP	전체 이미지의 녹색 값만 저장한 Buffer
GrayRect[200][100], BinaryRect[200][100], ThinRect[200][100]	각 번호판 영역의 데이터 저장

인식하기 위한 주요 변수들을 설명하고 있다. <표 2>에 설명하고 있는 변수들에 자동차 번호판 추출을 위한 정보들을 관리한다. <표 3>는 자동차 이미지에서 자동차 번호판 추출을 위한 구현 함수들을 보여준다. <표 3>에 제시된 함수들은 <표 2>에 제시된 변수들의 값을 이용하여 정확한 자동차 번호판 영역을 추출한다

3.2 문자 추출

자동차 번호판 영역에서 문자를 추출하기 위한 주요 멤버 변수 및 함수는 다음 <표 4>, <표 5>와 같다. <표 4>은 추출된 자동차 번호판 이미지에서 각 문자를 추출하기 위해서 필요한 변수들을 설명하고 있다. <표 5>는 자동차 번호판 이미지에서 각 문자를 추출하기 위한 함수들을 설명하고 있다.

3.3 문자 인식

추출된 문자에서 문자 인식을 위한 주요 멤버 변수

〈표 3〉 자동차 번호판 인식을 위한 주요 멤버 함수

주요 멤버 함수	설 명
void Init(HWND hWnd,int xsize,int ysize, unsigned long *OImg,unsigned char *GImg)	윈도우 핸들과 CScrLib에서 생성한 자동차 이미지의 정보를 넘겨받아 저장하고 메모리 DC를 생성한다. 그리고 Y축을 기준으로 X축의 명암 벡터를 생성한다.
BOOL GetGreen(int level)	차량 이미지의 녹색 값만 추출하여 Green_P에 저장하고 전체 이미지 중에서 녹색이 차지하는 빈도를 %로 환산하여 40%이상이면 TRUE를 리턴 한다
GetRect_Green(void)	우선 녹색 값만으로 번호판 영역을 찾아낸다 자가용인 경우 Green_P에 저장된 녹색 값 만으로도 번호판 영역을 대략적으로는 찾을 수 있다.
float IsItFlat(int XX1, int YY1, int XX2, int YY2)	번호판 영역 좌표의 2 1 비율을 산출하여 2 : 1에 가까운 비율을 리턴하고 2 : 1 비율과 다를경우 FALSE를 리턴
BOOL ScanWidth(int y)	y의 값에 해당하는 Y축 한 줄의 전체 녹색 값을 산출하여 30픽셀 이상이면 TRUE를, 이하 이면 FALSE를 리턴 한다
void RunRect(void)	번호판 영역을 잘라내는 전체 과정을 수행
void GetCarRect(void)	번호판 영역을 추출하여 Gray, Binary, 세선화 이미지를 생성하고 각 Buffer에 저장
void ThinImage(int xsize, int ysize, int imagesize,unsigned char pm1[200][100])	세선화 이미지를 만드는 함수이다 첫 번째 두 번째 인수는 세선화 이미지를 만들 크기이고 세번째는 이미지의 전체 크기(xsize*ysize)이며 네 번째는 이미지의 buffer이다.
void IsContrast()	번호판 영역의 2진화 과정에 히스토그램 분석에 의해 2진화시킨다
void PutCarRect(int x,int y,int Img)	번호판 이미지를 출력하는 함수로 출력할 X, Y좌표와 번호판 이미지의 종류 Img를 입력하면 번호판 이미지를 메모리 DC에 출력한다 Img의 입력은 0일 때 번호판의 원 이미지, 1인 때 Gray, 2일 때 Binary, 3일 때 세선화된 이미지를 출력한다.

〈표 4〉 문자 추출을 위한 주요 멤버 함수

주요 멤버 변수	설 명
unsigned char RectImg[200][100]	번호판 영역을 저장하는 임시 Buffer
int StrSize	문자의 개수 저장
RECT chpos[9]	문자 영역의 좌표 저장 Buffer
unsigned long Chx32[9][32][32],Chx64[9][64][64]	문자를 저장하는 Buffer, Chx32는 가로 세로 크기가 32 by 32. Chx64는 64 by 61 크기의 Buffer

〈표 5〉 문자 추출을 위한 주요 멤버 함수

주요 멤버 함수	설 명
void Init(HWND hWnd,unsigned char Img[200][100])	윈도우 핸들과 CCarRect에서 번호판 영역의 정보를 넘겨받고 메모리 DC 생성
BOOL Xscan(int size,int x, int y,BOOL op)	Y축을 기준으로 수평으로 x, y에서 size만큼 픽셀이 존재하면 op값에 따라 TRUE나 FALSE를 리턴
BOOL Yscan(int size,int x, int y,BOOL op)	X축을 기준으로 수직으로 x, y에서 size만큼 픽셀이 존재하면 op값에 따라 TRUE나 FALSE를 리턴
void GetCharRect()	문자 영역의 좌표를 추출하는 함수, 최종 좌표는 chpos[]에 저장
void CCharRect . SaveChar(int size)	추출된 문자의 좌표로 문자를 buffer에 저장, size가 0이면 32x32 buffer인 Chx32[]에 저장되고 1이면 64x64 크기의 buffer인 Chx64[]에 저장

〈표 6〉 문자인식을 위한 주요 멤버 변수

멤 버 변 수	설 명
int UpNum[2][16][16]	위쪽 숫자의 이미지 데이터 저장
int DnNum[4][16][16]	아래쪽 숫자의 이미지 데이터 저장
int UpNumRe[2]	인식된 위쪽 숫자의 코드 저장
int DnNumRe[4]	인식된 아래쪽 숫자의 코드 저장
int NumData[10][16][16]	파일의 숫자 폰트 저장

〈표 7〉 문자 인식을 위한 주요 멤버 함수

주요 멤버 함수	설 명
void GetNumData()	파일에서 숫자 폰트를 읽어 와서 NumData[]에 10개의 숫자를 저장
void SaveFont(int NumData[9][16][16])	디버깅을 위해 처리된 숫자를 다시 파일에 출력한다. 숫자 폰트의 표본을 만들 때 사용
void DivNum()	전체 문자 중에서 숫자만 잘라서 저장한다. 지역문자 다음의 숫자는 UpNum[]에 아래쪽의 숫자는 DnNum에 저장하고 문자의 개수(StrSize에 저장되어 있음)에 따라 UpNum에는 한 개 혹은 두 개의 숫자가 입력
void RecUpNum()	위쪽 글자의 인식 부분으로 최종 UpNumRe[]에 이미지 형태가 아닌 숫자의 형식으로 저장된다. 인식은 원형정합 방법을 사용
void RecDnNum()	RecUpNum() 같은 방법으로 아래쪽 숫자 인식
void Recognize_Number()	최종적으로 위의 멤버 함수들을 사용하여 숫자 인식

및 함수는 다음 <표 6>, <표 7>과 같다. <표 6>는 추출된 문자에서 문자인식을 하기 위한 주요 변수들을 설명하고 있다. <표 7>은 추출된 문자들에서 문자 인식을 위한 구현 함수들을 설명하고 있다.

3.4 실 험

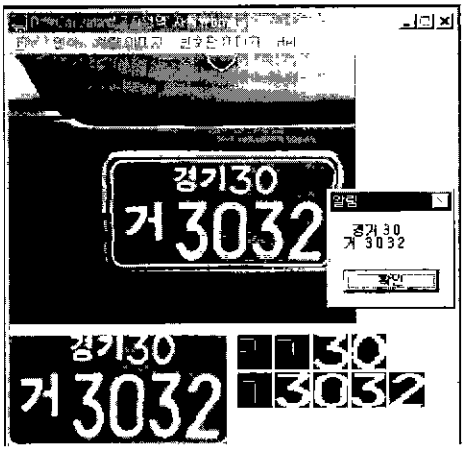
본 논문에서 제안한 적응 알고리즘을 이용한 자동차 번호판 인식 실험을 수행하였다. 실험 데이타는 디지털 카메라로 자동차 번호판 영상 정보를 만들고 이 사진을 가지고 자동차 번호판 인식을 한다. 자동차 번호판 인식은 본 논문에서 제안한 적응 알고리즘을 이용한 문자 인식에 초점을 맞추어 자동차 번호판 이미지에서 자동차 번호판 추출에 대한 실험은 배제한다. 정상적인 자동차 번호판 영역이 추출된 상태에서 자동차 문자 인식에 대한 실험을 통해서 정확한 실험의 결과를 얻을 수 있다.

입력 영상의 해상도는 640 x 480 화소의 크기와 256 칼라 색상을 가진다. 그리고 실험을 위해 사용된 컴퓨터는 펜티엄 III이다. 실험의 편의를 위하여 모든 차량에 대해서 하지 않고 개인용 차량을 중심으로 수행하였다

다음 (그림 5)는 자동차 번호판 인식을 위한 프로그램을 수행한 화면이다. 이 화면에서 다이얼로그 박스는 최종적으로 자동차 번호판 인식 결과의 문자열을 출력하여 보여 주고 있다

자동차 번호판 영역에 대한 문자 인식 실험 결과 전체 자동차 번호판 추출 결과에서 90%정도의 인식이 됨을 확인하였다. 이 수치는 현재 계속적으로 실험을 진행중에 있고 실험 결과의 신빙성을 얻기 위하여 다

각적인 방향에서 진행을 하고 있다 보다 정확한 인식을 위하여 많은 실험을 거쳐서 안정된 번호판 인식을 위한 적응 알고리즘을 구축할 예정이다



(그림 5) 자동차 번호판 인식 프로그램 수행 화면

4. 결 론

본 논문은 자동차 번호판 인식을 위한 알고리즘으로 적응 알고리즘을 제안했다. 적응 알고리즘은 번호판 문자 인식을 위하여 특정한 알고리즘을 사용하지 않고 원형 정합 알고리즘, 벡터 알고리즘, 세선화 알고리즘 등을 이용한 가능한 많은 정보를 수집하여 보다 정확한 판단이 될 수 있도록 한다. 기존의 자동차 이미지에서 자동차 번호판 영역 추출을 위한 방법으로 여러가지 기법들을 이용하는데 이러한 방법들의 근본적인 문제는 특수한 환경에 유리하게 실험이 이루어지고 있다는 사실이다. 그리고 인식률도 특수한 데이타에 대한 특수한 환경 중심으로 이루어지고 있다는 것이다

본 논문에서 제안한 적응 알고리즘은 자동차 이미지에서 문자 인식을 위한 알고리즘이다.자동차 이미지에서 자동차 번호판 영역 추출을 위하여 색상 정보, 자동차 이미지에서 명암 변화 정보, 자동차 번호판의 크기 정보, 번호판의 년적 정보등을 이용하여 자동차 번호판을 추출한다. 본 논문에서 제안한 적응 알고리즘은 정확한 자동차 번호판 문자 인식을 위하여 여러가지 정보를 종합하여 정확한 판단 자료로 사용한다 본 논문에서 제안하는 적응 알고리즘을 사용함으로써 기존의 알고리즘들을 실제 시스템에 접목하여 사용할 수 없는 문제점을 해결할 수 있다 또한 실제 자동차 번호판 인식을 위한 알고리즘으로 사용 가능하다.

그리고 실험 결과 소량의 데이타를 가지고 실험을 하였지만 90% 이상의 인식율을 보이고 있다. 앞으로 많은 데이타를 가지고 실험을 하여 보다 안정된 알고리즘임을 증명할 것이고 현장에 설치 작업을 통한 조율을 할 것이다 따라서 본 논문에서 제안한 알고리즘을 이용한 자동차 번호판 알고리즘을 실제 프로그램에 접목하여 이용 가능하도록 할 것이다.

참 고 문 헌

[1] 김숙, 조형기, 민준형, 최종욱, "명암벡터를 이용한 차량번호판 추출 알고리즘", 한국정보과학회논문지(B), 제25권 제4호, 1998.4.

[2] 김병기, "명암변화와 칼라 정보를 이용한 차량 번호판 인식", 한국정보처리학회, 제6권 제12호, 1999.12.

[3] 이진수, 권오준, 방승양, "개선된 자소 인식 방법을 통한 고인식률 인쇄체 한글 인식", 한국정보과학회 논문지(B), 제23권 제8호, 1996.8.

[4] 김의정, 김태균, "오프라인 문서에서 개별 문자 추출과 한자 인식에 관한 연구", 한국정보처리학회 논문지, 제4권 제5호, 1997.5

[5] 김의정, 김태균, "최대 블록화 방법을 이용한 문자획 특징 추출에 관한 연구", 한국정보처리학회 논문지, 제4권 제4호, 1997.4.

[6] 조등욱, "화상 구조 파악에 의한 화상의 잡음 제거 및 경계선 추출", 한국정보처리학회 논문지, 제4권 제7호, 1997.7.

[7] 곽후근, 김성호, 정규식, "한글 문자의 구조 정보에 기반한 동적 정보 복원", 한국정보과학회 논문지, 제25권 제3호, 1998.3

[8] 박희선, 이성환, '오프라인 글씨 인식을 위한 은닉 마르코프 메쉬 랜덤 필드 모델", 한국정보과학회 논문지, 제22권 제12호, 1995.12

[9] N. A. Khan et. al., "A License Plate Recognition System," Proc. Intl. Conf. on Applications of Digital Image Processing XXI, pp.14-24, 1998

[10] J. R. Cowell, "Syntatic Pattern Recognizer for Vehicle License Plates," IEEE Transactions on Vehicular Technology, Vol.4, No.4, pp.790-799. 1995.

[11] H. A. Hrgt, et al, "A High Performance License Plate Recognition System," Proc. IEEE Intl. Conf on Systems, Man and Cybernetics, Vol.5, pp.4357-4362, 1998.

[12] L. Fausett, Fundamentals of Neural Networks, Addison Wesley Publishing COmpany Inc., 1989.

[13] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, Machine Vision, McGraw-Hill, 1995.

[14] Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis, and Machine Vision, Brooks/Cole Publishing Company, 1999

장 승 주

e-mail : sjjang@hyomin.dongeui.ac.kr

1985년 부산대학교 계산통계학과 졸업(계산학)

1991년 부산대학교 계산통계학과 (계산학 이학석사)

1996년 부산대학교 컴퓨터공학과 (공학박사)

1987년~1996년 한국전자통신연구원 시스템S/W 연구실

1996년~현재 동의대학교 컴퓨터공학과 조교수

관심분야 : 운영체제, 보안, 분산처리 시스템