

## 지능형 자동 주차 지원 시스템의 구현

### Implementation of an Intelligent Automatic Parking Assist System

박 정 술\*, 한 민 홍\*

Cheong-Sool Park\*, Min-Hong Han\*

#### 요 약

본 논문에서는 차량을 자동으로 주차시키기 위한 시스템을 구현하고 이 시스템을 구동시키기 위한 방법론을 제시한다. 차량이 자동으로 주차하기 위해서는 첫째, 차량의 위치와 주차위치, 장애물의 위치를 파악하고 둘째, 장애물을 피하여 정확한 주차 위치에 도달할 수 있는 차량의 이동 경로 산출하고 셋째, 생성된 경로를 따라 차량이 이동할 수 있도록 핸들을 제어해야 한다. 차량의 위치와 주차위치, 장애물의 위치를 판단하기 위하여 카메라로부터 입력 받은 영상의 변환을 통해 평면 지도를 생성하는 기법과 차량의 이동 모델을 통해 차량의 속도와 조향각도를 이용하여 이동한 차량의 위치를 판단하는 관성 항법 기법을 이용하였다. 차량의 이동 경로 산출에 있어서는 차량의 회전 반경을 고려한 Simple path method와 Bezier spline을 이용한 경로 수정 방법을 이용하였다. 또한, Divided arc method를 이용하여 장애물을 피하는 다양한 이동 경로를 생성하였다. 생성된 경로 중 다양한 목적함수를 만족시키는 제일 좋은 경로를 선택하기 위한 방법을 적용하여 하나의 경로를 선택하였다. 차량의 이동 경로 상에 장애물을 피해 정확한 위치에 차량을 위치시킬 수 있는 방법을 테스트 하였다. 생성된 경로를 따라 차량이 움직이기 위한 제어 기법으로는 Virtual road method를 이용하여 기계적인 시간 지연 등의 문제점을 해결하였다.

#### Abstract

In the paper, we propose an intelligent automatic parking assist system. To realize an automatic parking, first, the prospective parking position and the location of a vehicle should be recognized. Second, the system should compute a path which introduces the parking position precisely with avoiding any obstacles. Third, the handle should be controlled so that the vehicle moves through the path. To calculate the location of the vehicle and its surroundings, the system applies the camera image method to transforming input images to the plane map. It also uses the inertial navigation method which recognizes the position and the direction of a moving vehicle by using a kinematic model of the vehicle. To generate a path of the vehicle, the simple path method and the Bezier spline method are tested. The divided arc method which generates multiple paths is also tested. We apply a method which makes the system choose the best path with multiple objective functions. We introduce the virtual road method, as a solution for the problem of mechanical time delay, to have the vehicle followed the designated path.

**Key words** : autonomous parking, vehicle, trajectory, image processing

#### I. 서 론

초보 운전자뿐만 아니라 일반 운전자에게도 주차는 많은 주의를 요한다. 차량의 위치와 근처 지형지물까지의 거리, 조향 각도와 차량의 예상 진로의 관계 등을 운전자의 감각

에 의존하여 판단해야 하기 때문이다. 좁은 공간에서 주차의 경우 운전자의 순간적인 위치 판단 실수나 핸들 조작의 실수로 인해 다른 차량이나 장애물과 충돌을 일으킬 수 있다. 이로 인한 경제적 손실이나 운전자들의 주차에 대한 중압감을 해소하기 위해서는 주차 시에 정확한 차량의 위치와 근처 지형지물까지의 거리의 판단을 통해 주차 위치까지의 가능 경로를 판단하고, 핸들의 정확한 제어를 도울 수 있는 시스템이 필요하다.

차량을 주차시키는 것은 보통 도로에서 운전하는 것과는

\*고려대학교 산업공학과

접수 일자 : 2005. 7. 14

수정 완료: 2005. 9. 28

논문 번호 : 2005-3-1

다르게 핸들 조작을 통해 차량의 위치를 변화 시키면서 동시에 차량이 향하는 방향을 변화시켜 원하는 위치에 원하는 방향으로 차량을 위치시키는 일련의 작업이다. 이와 같은 일련의 작업을 운전자 대신 수행하기 위해서는 첫째, 차량의 위치와 주차 위치 그리고, 주변의 장애물의 위치를 판단해야 한다. 둘째, 일정 위치까지의 가능 경로(feasible path)를 산출하고 그 경로를 순차적으로 추적(tracking)해야 한다. 셋째, 차량의 핸들을 정확히 제어하여 계산된 경로대로 진행할 수 있어야 한다.

기존의 연구들은 다양한 방법의 경로 생성에 관한 연구가 추가 되어왔다. 평행주차를 위한 2개의 원호를 이용한 주차 방법[1], 1개의 원호와 직선으로 이루어진 경로를 퍼지론을 이용해 개선해나가는 방법[2], Wavelet과 RBF neural network를 이용하여 차량의 주차 경로를 생성한 방법[4]등 다양한 방법들이 연구되어 왔다. 또한, 주차를 위한 공간을 인식하는 분야에서는 I. E. Paromtchik등[8]은 레이저 스캐너를 이용하여 공간을 인식하였고, K. Jiang등[7]은 초음파 센서를 이용하기도 했다. 또한, J. Xu등[3]은 영상을 이용하여 공간을 인식하였다. 다양한 센서를 종합적으로 활용한 M. Wada등[6]의 연구도 있었다. 생성된 경로를 따라가는 제어 분야의 연구도 있었다. I. Rivals등[10]은 neural network를 이용하여 차량이 생성된 경로를 따라가며 경로에서 벗어났을 경우 다시 경로로 복귀하는 제어에 관한 프레임 워크를 제공하였다.

본 연구에서는 차량의 위치와 주차위치, 장애물 등의 위치를 파악하기 위하여 카메라로부터 입력 받은 영상을 변환을 통해 평면 지도(plane map)로 생성하는 기법과 차량의 속도와 조향각도를 이용하여 이동한 차량의 위치를 판단하는 관성 항법기법을 이용하였다. 차량의 이동 경로 산출에 있어서 차량의 회전 반경을 고려한 Simple path method과 Bezier spline을 이용한 경로 수정 방법과 이를 확장시킨 다양한 이동 경로를 생성할 수 있는 Divided arc method을 이용해 차량의 이동 경로 상에 장애물을 피해 정확한 위치에 차량을 위치시킬 수 있는 방법을 테스트 하였다. 차량의 조향각과 움직임에 대한 모델을 이용하여 이동 가능한 경로를 생성하고 장애물과 핸들 제어의 제약점 등을 고려하여 최선의 경로를 선택하는 기법을 이용하였다. 생성된 경로를 따라 차량이 움직이기 위한 제어 기법으로는 Virtual road method을 이용하여 기계적인 시간 지연 등의 문제점을 해결하였다.

## II. 위치 판단

### 2.1. 평면 지도(Plane Map) 생성 기법

주차를 위해서는 우선 주차 환경에 대한 정보를 수집해야 한다. 주차 환경에 대한 정보를 수집하는 방법으로는 초음파를 이용한 방법[7], 레이저 스캐너를 이용한 방법[8], 카메라를 이용한 방법[3] 등이 있는데 본 연구에서는 운전자가 눈을 통해 주차 환경의 정보를 수집하는 점에 착안하여 카메라를 이용한 방법을 이용하였다. 카메라를 통해 일차적으로

입력되는 영상은 변환을 통해서만 거리 정보를 추출 할 수 있다. J. Xu등 [3]이 제안한 3\*3 Matrix를 이용한 방법은 측정치에 의해서 Matrix를 생성하고 이를 이용하여 평면 좌표계로 변환하였으나 본 연구에서는 카메라의 초점거리, 지면에 대한 각도, 카메라의 높이를 이용하여 공식화하여 평면의 지도로 변환하는데 사용하였다. 이와 같은 방법은 카메라의 설치 위치, 각도를 변경하였을 때도 영상을 변환할 수 있는 장점이 있다.

그림 1의 삼각형  $Ocb$ 에서  $\overline{Oc} = f$ ,  $\overline{cb} = y$ ,  $\angle Ocb = \frac{\pi}{2}$  이므로  $\theta$  는 다음 식으로 표현된다.

$$\theta = \tan^{-1}\left(\frac{y}{f}\right)$$

또한, 삼각형  $O\hat{O}B$ 에서  $\angle BO\hat{O} = \Phi - \theta$  이므로 평면 좌표상에서의  $Y$ 는

$$Y = H \times \tan(\Phi - \tan^{-1}\left(\frac{y}{f}\right)) \quad \text{--- (1)}$$

으로 표현된다.

또한, 삼각형  $OBA$ 와  $Oba$ 는 닮음이므로 다음과 같은 비례 관계가 성립하며

$$\overline{Ob} : \overline{ab} = \overline{OB} : \overline{AB} \quad \text{--- (2)}$$

선분  $\overline{Ob}$ 는 삼각형  $Ocb$ 에 의해  $\overline{Ob} = \frac{f}{\cos\theta}$  으로 표현된다.

또한, 삼각형  $OBA$ 에서  $\overline{OB} = \frac{H}{\cos(\Phi - \theta)}$  이므로 비례식 (2)에 의해

$$\frac{f}{\cos\theta} : x = \frac{H}{\cos(\Phi - \theta)} : X$$

으로 표현될 수 있다. 위의 비례식에 의해 평면 좌표상에서의  $X$ 는 다음과 같이 표현된다.

$$X = x \times \frac{H}{\cos(\Phi - \tan^{-1}\left(\frac{y}{f}\right))} \times \frac{\cos\theta}{f} \quad \text{--- (3)}$$

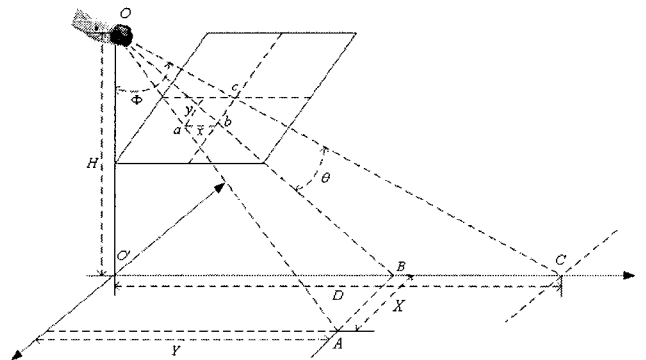


그림 1. 입력 영상 좌표계와 평면 좌표계  
Fig. 1. Coordination of input image and plane map

식(1)와 식(3) 를 이용하여 일차적 입력 영상에서 주차선 정보를 추출한 후 변환하면 다음 그림 2와 같은 결과를 얻을 수 있다.

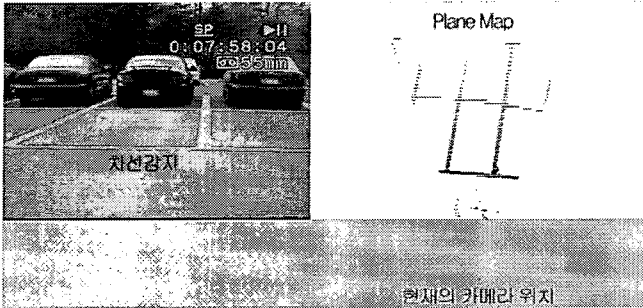


그림 2. 영상의 평면 변환

Fig. 2. Transformation of input image to plane map

현재 카메라의 위치를 기준으로 한 평면 좌표계 상에 주차선의 위치를 표현할 수 있다. 따라서 차량의 정확한 주차 위치와 현재 차량의 위치간의 관계를 쉽게 표현할 수 있다. 이와 같은 표현기법은 순차적인 영상의 입력에 있어서도 큰 강점을 가지고 있다. 차량이 이동할 때 카메라를 통해 입력되는 1차적인 영상은 계속해서 변화한다. 그러나 3차원의 세계를 2차원적인 영상으로만 받아들이게 됨으로 이를 통해 현재 차량의 위치를 계산하거나 이전 영상 정보와 현재의 영상 정보를 비교하는데 많은 계산이 필요하게 된다. 그러나 평면 좌표계로 변환한 후에는 이전 영상 정보의 업데이트는 Correlation 비교를 통해 변환된 영상을 이동된 위치에 덧씌우는 것만으로도 가능하게 된다.

2.2. 영상을 이용한 장애물 감지

영상을 이용하여 장애물을 감지 하기 위해서는 장애물의 종류를 우선 정의해야 한다. 주차시 주요한 장애물 요소는 다른 주차 공간에 주차되어 있는 차량이 될 것이다. 차량 외에도 사람 및 기타 물건들이 존재할 수 있으나 본 연구에서는 차량을 주요한 장애물로 보고 차량을 탐지하는 기법을 소개하도록 한다. 노광현[9]에 의하면 차량의 밑부분에는 그림자가 발생하여 도로면과는 다른 성질을 가지는 점을 이용하여 Edge를 추출하고 Edge에 의해 도로위의 차량을 탐지하였다. 이와 같은 점에 착안하여 본 연구에서는 주차선 내에 존재하는 차량의 유무를 판단하여 한 주차선 내부 전체를 장애물로 인식하는 다음과 같은 기법을 활용하였다.

**Step 1.** 주차선을 감지하기 위하여 원 영상  $f(x,y)$ 로부터 주차선의 후보를 나타내는 이진 영상  $f_w(x,y)$ 을 색상의 임계값(color threshold)인  $T_w$ 에 의해 만들어 낸다.  $f_w(x,y)$ 의 산발적인 잡음을 효과적으로 제거하기 위해

Morphology 기법인 Erosion(침식연산)과 Dilation(팽창연산)을 연속적으로 수행하여 잡음이 없는 깨끗한 영상을 구한다.

**Step 2.**  $f_w(x,y)$ 를 2.1절의 영상 변환 기법을 활용하여  $F_w(X,Y)$ 로 변환한다.  $F_w(X,Y)$ 에서 직선을 찾아 내기 위해 chain code를 이용해 직선성분을 찾아 내고 직선성분을 이용해 사각형의 영역인 주차 구획을 이진 영상인  $P_k(X,Y)$ 를 만든다.

**Step 3.** 장애물인 차량을 감지하기 위해 각 주차 구획  $P_k(X,Y)$ 내의 영역 대해 Edge를 추출한다. Edge추출은 원 영상인  $f(x,y)$ 로부터 추출해야 하므로 각 주차 구획  $P_k(X,Y)$ 를  $p_k(x,y)$ 로 역변환한다. 각 Edge추출에는 기본적인 Mask의 하나인 Sobel Mask를 x축과 y축으로 2회 적용한다. 추출된 Edge를 임계값을 적용하여 영역내의 Edge의 개수를 구하고 Edge의 개수가 일정 개수 이상인 주차 구획을 장애물이 존재하는 주차 구획으로 판단한다.

위의 기법을 활용하여 장애물을 감지한 결과는 그림 3과 같다.

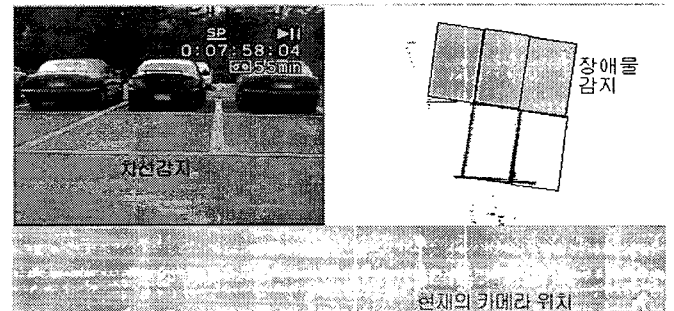


그림 3. 장애물 감지

Fig. 3. Obstacle detection

2.3. 차량의 모델링

평면지도 상에서 차량의 위치를 정확히 파악하기 위해서는 차량의 조향각과 속도를 통해 차량의 위치를 계산할 수 있는 모델이 필요하다. [1,2,3,4,5,6,7,8]의 모든 연구에서 차량의 움직임 모델을 활용하였다. 각각의 표현은 조금씩 차이가 있으나 그 결론은 '차량의 회전 반경의 중심은 두 뒷바퀴의 중심이다.'는 점에서 일치한다. 이는 기본적으로 앞바퀴의 조향각과 차량의 길이만 고려한 2륜 차량의 모델을 이용하여 4륜 차량의 회전계적을 구하는 것과 동일한 방법이다. 그림 4와 같이 4륜 차량의 앞바퀴와 뒷바퀴의 중심에 바퀴를 가지고 있는 2륜 차량으로 가정하면 앞바퀴와 뒷바퀴 사이의 길이가  $l$ 인 차량의 뒷바퀴는 원점에 위치하고 앞바퀴는  $P_1$ 에 위치할 때, 차량이 그리는 궤적은 중심이  $P_3$ 인 두 개의 원으로 나타낼 수 있다. 이 때 앞바퀴는 바깥쪽 원주를 뒷바퀴는 안쪽 원주 상으로 움직인다.

현재 앞바퀴가  $P_1$ 에 위치하고 있고  $P_2$ 지점으로 이동하고자 한다면 앞바퀴의 조향각( $\Phi$ )은 다음과 같이 구할 수 있다.

$$\sqrt{r^2 + l^2} = \sqrt{y^2 + (x-r)^2}$$

$$r = \frac{x^2 + y^2 - l^2}{2x} \quad \text{--- (4)}$$

따라서 조향각은

$$\Phi = \tan^{-1} \frac{l}{r} \quad \text{--- (5)}$$

로 구할 수 있다.

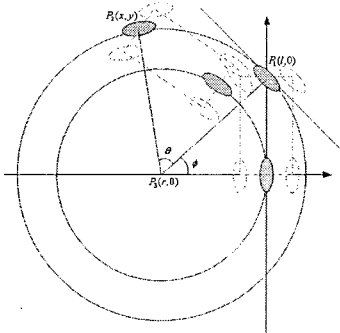


그림 4. 차량의 움직임 모델  
Fig. 4. Kinematic model of vehicle

위의 차량의 움직임 모델을 이용하여 차량을 모델링하고 속도와 조향각을 입력하면 차량의 이동 경로를 구할 수 있다. 그림 5는 차량의 속도와 조향각을 입력하여 차량의 이동 경로를 표시한 그림이다.

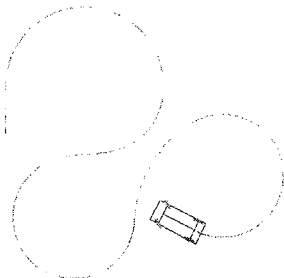


그림 5. 차량의 궤적 표시  
Fig. 5. Draw of vehicle movement

### III. 경로 생성

#### 3.1. Simple path method

차량 모델에서 보는 바와 같이 차량은 일정한 회전 중심을 기준으로 선회한다. 회전 중심을 기준으로 선회하는 점을 이용하여 Y. K. Lo등[1]은 평행 주차시 2개의 원호를 이용한 주차 경로를 생성하였고, T. S. Li등[2]은 전방으로 주차하기 위한 경우에 1개의 원호와 1개의 직선으로 이루어진 경로를 생성하였다. 이들의 연구는 특수한 경우(평행 주차, 수직 주차)에 차량을 원하는 위치에 위치시키기 위

한 방법론으로써 활용된 것이다. 본 연구에서는 이를 더 일반화하여 차량을 원하는 위치에 원하는 방향으로 위치시킬 수 있는 방법론인 Simple path method(SPM)를 제안한다.

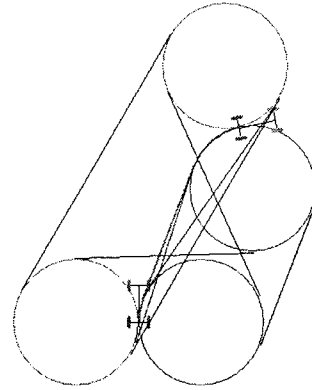


그림 6. Simple path method의 원리  
Fig. 6. Principal of Simple path method

SPM은 그림 6과 같이 2개의 원호와 1개의 직선으로 이루어진 경로를 산출한다. 2개의 원호는 시작위치의 회전반경의 일부와 최종위치의 회전반경의 일부이다. 1개의 직선은 이 두 원호의 공통 법선이 된다. SPM에 의해 경로를 생성하는 방법을 살펴보면 다음과 같다.

**Step 1.** 시작위치에서의 수직으로 회전 반경  $r_1$ 만큼 떨어진 두 회전반경의 중심인  $O_{s1}$ 과  $O_{s2}$ 를 구한다. 또한, 최종위치에서 수직으로  $r_2$ 만큼 떨어진 두 회전반경의 중심인  $O_{e1}$ 과  $O_{e2}$ 를 구한다.

**Step 2.** 시작위치의 회전 반경의 중심인  $O_s(x_s, y_s)$ 를  $O_{s1}$ 과  $O_{s2}$  중 하나를 선택하고  $O_e(x_e, y_e)$ 를  $O_{e1}$ 과  $O_{e2}$  중 하나를 선택한다. 선택한 두 중심으로부터 거리가 각각  $r_1$ 과  $r_2$ 만큼 떨어진 직선의 방정식  $y = mx + d$  을 구하기 위한 조건은

$$\frac{mx_s - y_s + d}{\sqrt{m^2 + 1}} = r_1$$

$$\frac{mx_e - y_e + d}{\sqrt{m^2 + 1}} = r_2$$

이 된다. 구해진 직선은 두 원의 위치 관계에 따라 최대 2개의 내접선, 2개의 외접선으로 이루어진다.

**Step 3.** 하나의 직선은 하나의 경로를 이루게 되는데 차량의 방향을 유지하기 위한 몇 가지 제약을 가지게 된다. 차량의 회전 방향과 내접선인가 외접선인가의 여부에 따라서 이와 같은 제약이 발생하게 되는데 각각의 제약은 다음과 같이 된다.

차량의 왼편의 최소 회전 반경 원에서는 CW(Clock Wise)방향이 후진방향이고 CCW(Counter Clock Wise)방향이 전진

방향이다. 차량의 오른쪽 회전 반경 원에서는 CW가 전진방향이고 CCW가 후진 방향이 된다. 시작 위치에서 전진 방향으로 시작하게 되면 외접선을 따라 이동하게 되면 이미 기술한 방향이 차량의 전,후진 방향이 되고 내접선을 따라 이동하게 되면 기술한 방향의 반대 방향이 전,후진 방향이 된다. 이와 같은 성질을 이용하여 그림 7과 같이 전진 주차 방향뿐만 아니라 후진 주차 방향까지도 생성할 수 있다.

**Step 4.** Step 2로 돌아가 다른 회전반경의 원을 선택하여 경로를 산출한다. 4가지 조합의 경로를 산출할 수 있다. 또한,  $r_1$ 과  $r_2$ 를 일정 범위내에서 변화시키면서 경로를 산출하면 다양한 경로를 구할 수 있다. 다양한 경로중 장애물을 지나는지 여부를 판단하여 한 경로를 선택할 수 있다. 그림 7에서와 같이 회전 반경을 변화시켜 장애물을 피할 수 있는 경로를 산출하였다.

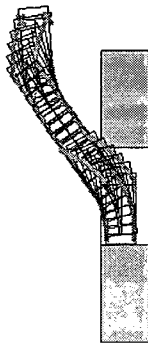


그림 7. SPM에 의한 후진 평행 주차 경로

Fig. 7. Backward parallel parking path using SPM

SPM에 의해 생성된 해를 살펴보면 초기에 핸들을 일정 각도로 한번 틀었다가 다시 전부 풀어서 중앙으로 만든다. 그리고 마지막에 다시 한번 핸들을 일정 각도로 틀게 된다. 이와 같은 동작은 모터로 구동되는 핸들의 경우에는 불가능한 동작이다. 일정 각도까지 돌아가는 모터의 속도가 있기 때문에 그러하다. 따라서, 점진적으로 변화하는 조향각을 가지는 경로로 수정해줄 필요가 있다.

### 3.2. Bezier spline method

경로를 수정하는 방법으로는 T. S. Li등[2]과 K. Y. Lian등[5]의 퍼지를 이용한 방법, L. Li[4]등의 wavelet, neural network를 이용한 방법이 연구되었다. 본 연구에서는 W. L. Nelson[11]의 Bezier spline을 이용한 경로 생성 연구에서 착안하여 Bezier spline을 이용한 주차 경로 수정 방법을 제안한다.

Bezier spline은 다음과 같이 정의된다. [12]

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t), \quad 0 \leq t \leq 1$$

그리고, 베이지어 베이스(Bazier basis), 또는 블렌딩 함수(blending function)인  $J_{n,i}(t)$ 는 다음과 같다.

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Bezier spline의 가장 큰 특징은 시작점에서의 미분치는 첫 점에서 두 번째 점으로 향하는 벡터와 같은 방향을 가진다는 것이다. 이 성질을 이용하여  $n=3$ 인 Bezier spline을 이용해 차량의 경로를 생성할 수 있다.  $i=0$ 일 때, 출발 위치,  $i=1$ 일 때, 출발 위치에서 SPM에 의해 생성된 원호의 방향으로  $l_1$ 만큼 떨어진 위치,  $i=2$ 일 때, 최종위치에서 SPM에 의해 생성된 원호의 방향으로  $l_2$ 만큼 떨어진 위치,  $i=3$ 일 때, 최종위치에 각각  $B_i$ 를 설정한다. 이렇게 생성된 곡선은 블렌딩 함수에 의해 생성된 임의의 곡선이므로 원호의 일부분이라 할 수 없다. 따라서 이 곡선을 따라 차량을 이동 시켰을 때 차량의 방향이 정확히 접선 위를 따라가서 최종 위치에 정확히 도착하는지 여부를 알아 볼 필요가 있다. 이를 위해서 2.3.의 차량의 회전 궤적 모델을 이용하여 다음과 같은 시뮬레이션 실험을 했다.

먼저 차량의 경로가 되는 Bezier spline을 생성한다. 차량의 초기 위치와 끝 위치를 정점으로 하고 0에서 1까지의 구간을 20개로 나누어 곡선을 생성하였다. 시뮬레이션상의 차량이 생성된 곡선의 각 포인트들을 2.3.의 (4)(5) 공식을 이용하여 따라가게 만들었다. 현재 자기의 위치에서 각 포인트들까지 가기 위한 핸들의 각도를 계산하고 차량이 이동하도록 만든 후 최종 위치에서 차량의 방향의 오차를 계산하였다. 그 결과는 그림 8과 같다.

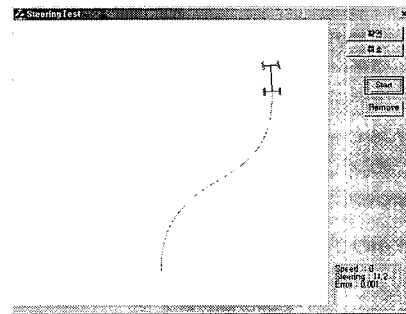


그림 8. 베지어곡선 차량 시뮬레이션 결과

Fig. 8. Result of bezier spline path simulation

Bezier spline을 따라갔을 때 차량의 방향은 정확히 원하는 방향이 됨을 알 수 있었다. 여기서 한가지 문제점은  $l_1$ 과  $l_2$ 를 결정하는 것이다. 이 두 값을 결정하는 문제는 차량의 최소 회전 반경의 제약 조건과 장애물을 지나는 조건을 이용하여 제약 조건을 만족하지 못하는 포인트의 개수를 최소화하는 문제가 된다.

### 3.3. Divided arc method

SPM과 Bezier spline을 이용한 경로 생성 방법을 확장하여 본 연구에서는 Divided arc method(DAM)을 제안한다.

DAM은 차량이 이동하는 경로가 수많은 원호로 이루어져 있는 점을 이용하여 만들어졌다. DAM에 의해 경로를 생성하는 방법을 살펴보면 다음과 같다.

**Step 1.** 초기의 차량의 위치와 방향을  $X_0(x_0, y_0, \theta_0)$ 라 하고, 도착시 차량의 위치와 방향을  $E(x_e, y_e, \theta_e)$ 라 한다. 탐색의 깊이를  $n$ 으로 설정하고 Step 2로 간다.

**Step 2.**  $X_k$ 의 위치에서 조향각  $\Phi$ 를 최소 조향각 $\Phi_{min}$ 부터 최대 조향각 $\Phi_{max}$ 까지  $\Delta\Phi$ 만큼씩 증가시키며  $\Delta d$ 만큼 이동시킨 위치 새로운 위치  $X_{k+1}(x_{k+1}, y_{k+1}, \theta_{k+1})$ 을 구한다.

$X_{k+1}$ 의 개수는  $\frac{(\Phi_{max} - \Phi_{min})}{\Delta\Phi}$  개 만큼 발생한다.

**Step 3.**  $k+1=n$ 이면 각  $X_{k+1}$ 의 위치에서  $E$ 까지 가는 조향각을 2.3의 식(3)(4)를 이용하여 구하고 구해진 조향각에 따라 이동한 위치  $\theta_{move}$ 와  $\theta_e$ 를 비교하여 같은지 확인하고 같으면 이전에 경로를 추가한다.

**Step 4.**  $X_k$ 와  $X_{k+1}$  사이에 장애물이 존재하는지 확인한다. 장애물이 존재하지 않을때는 새로운 위치인  $k=k+1$ 로 하고 Step 2로 돌아간다.

DAM은 차량의 위치를 트리구조를 탐색해가는 것과 같이 초기 위치로부터 탐색해 나가며 최종 위치에 도달하는 경로와 장애물을 피하는 경로를 찾아가는 재귀호출함수의 형태를 띄고 있다. 차량의 위치를 탐색해 나가는 트리 구조를 실제 차량이 이동하는 경로 상에서 살펴보면 그림 9와 같다.

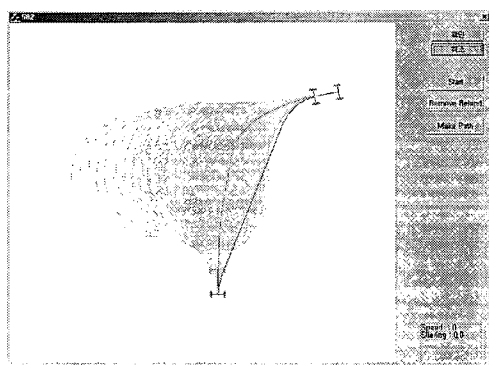


그림 9. 차량 경로 탐색 트리 구조의 실제 형태

Fig. 9. Actual figure of searching tree structure in case of vehicle path

DAM에 의하면 그림 8과 같이 시작 위치에서 최종 위치까지 갈 수 있는 다양한 경로를 산출할 수 있게 된다. 이와 같은 방법은 장애물이 존재할 때도 장애물을 피해서 최종 위치에 차량을 정확히 위치 시킬 수 있는 경로를 선택할 수 있게 한다.

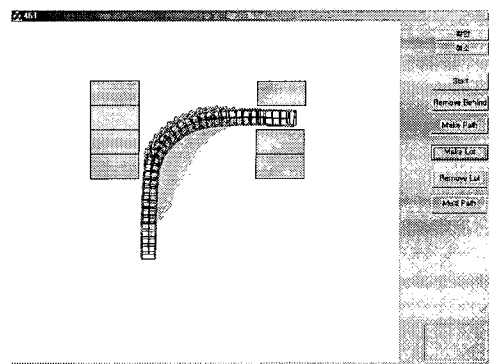


그림 10. 장애물이 있을 때의 경로

Fig. 10. Vehicle path with obstacle

2.2에서 정의한 바와 같이 주차 공간의 차량의 유무에 따라 장애물을 판단하기로 했으므로 그림 10과 같이 직사각형의 형태로 된 장애물이 있을 때의 장애물을 피해서 갈 수 있는 다양한 경로가 생성된 것을 알 수 있다.

### 3.4. 경로 선택 방법

본 연구에서는 다른 연구들과 다르게 다중의 경로를 생성하였기 때문에 이 경로들 중에서 실제 주행을 위한 한 가지 경로를 선택해야 한다. 그림 10에서 박스의 형태로 표시된 경로와 같이 한 가지 경로를 선택하는 기준으로는 장애물과 거리주행시 조향각의 변화 그리고 거리이다. 각 경로상에 장애물과의 거리는 멀어야 하고, 경로를 따라 주행시에 조향각의 변화는 덜 있어야 하며, 최종 위치까지의 거리는 가까워야 한다. 이 세 가지의 기준을 동시에 만족시킬 수 있는 경로를 선택하기 위해 각 기준에 따라 경로들에 순위를 부여하였다. 부여된 순위를 경로별로 합산하여 합산 순위값을 계산하고, 계산된 순위값이 가장 작은 경로를 선택하는 방법을 이용하였다.

이와 같이 순위를 이용하는 방법을 사용한 이유는 각각의 기준값이 다른 단위를 사용하고 있기 때문에 선형 조합에 의한 식의 최대화 또는 최소화 하는 방식을 사용할 수 없기 때문이다.

## IV. 조향 제어

### 4.1. Virtual road method

생성된 경로를 따라서 차량이 이동하기 위해서는 차량 현 위치를 파악하고 그 위치와 생성된 경로상의 위치를 비교하여 차량의 조향을 제어해주는 방법이 필요하다.

차량이 도로상을 주행하는 경우의 조향 제어는 화면상의 도로의 영상에 의해 도로 영상의 중심을 향해 차량이 이동하도록 하는 조향 제어 방법을 사용하고 있다. 도로상을 주행할 때는 차량의 움직임에 따라 이에 대한 피드백으로 영상이 변화하기 때문에 일정 수준 이하의 조향각을 입력해 주기만 하면 차량이 도로 밖으로 주행하지 않는다. 본 연구에서는 이와 같은 점에 착안하여 주차 시에 차량의 생성된 경

로를 가상의 길로 보고 가상의 길의 가운데로 차량이 이동할 수 있는 조향각을 입력해주는 방법을 활용하였다.

생성된 경로를 평면지도 상에 그려준다. 그림 11과 같이 차량의 뒷바퀴의 중심점  $P_1$ 을 기준으로 동심원을 2개 그린다. 동심원 2개와 가상의 길이 만나는 점의 중심을 각각  $P_2$ ,  $P_3$ 라 할때  $P_1, P_2, P_3$ 를 지나는 원을 그리고 그 원의 중심점이 회전 반경이 된다. 회전 반경을 기준으로 2.3.의 식(4)를 이용하여 차량의 조향각을 구할 수 있다. 조향각을 산출하여 입력해주면 경로를 벗어나 있는 경우에도 경로를 찾아서 최종 주차 위치에 도달하는 것이 가능하게 된다.

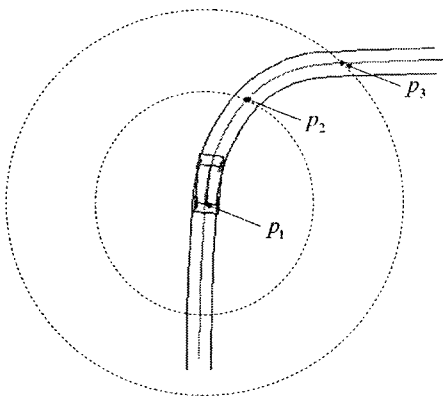


그림 11. Virtual road method의 원리  
Fig. 11. Principal of Virtual road method

## V. 차량 시스템

### 5.1. 주차를 위한 차량 시스템 개요

차량 시스템의 구성은 그림 12와 같다. 컴퓨터는 영상처리, 경로생성, 제어 신호 발생, 차량의 위치 판단 등의 동작을 수행하고 두 개의 마이크로프로세서가 시리얼 인터페이스를 통해 컴퓨터에 연결되어 제어신호를 처리한다.

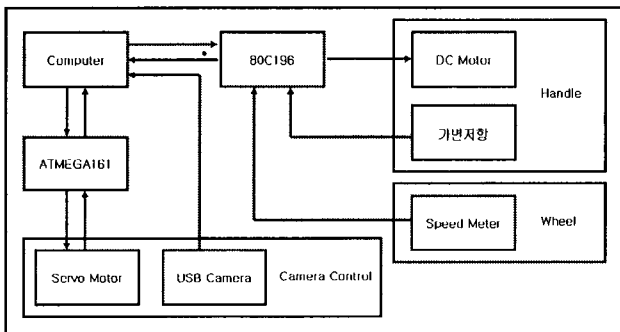


그림 12. 무인 주차 시스템의 블록 다이어그램  
Fig. 12. Block diagram of automatic parking system

### 5.2. 핸들 제어 시스템

주차 시 자동으로 핸들을 제어하기 위하여 DC모터를 이용

한 핸들 제어 시스템을 제작하였다. 그림 13과 같이 DC모터를 구동하기 위해 16 bit microcontroller 80C196을 이용하여 모터를 구동시킬 수 있는 회로를 구성하였고, 시리얼 인터페이스를 통해 컴퓨터와 연결하여 핸들을 구동할 수 있도록 하였다. 핸들의 각도를 인식하기 위해서 가변 저항을 이용하여 80C196에서 AD 컨버팅을 통해 핸들의 각도를 인식하고 모터에 흐르는 전류의 방향을 제어함으로써 핸들을 원하는 각도에 위치시킬 수 있도록 하였다. 또한 모터가 목적 각도에 도달했을 때 정확한 제어를 위해서 모터에 흐르는 전압을 PWM(Pulse Width Modulation)에 의해 조절하여 부드럽게 목적 각도에 도달하도록 하였다.

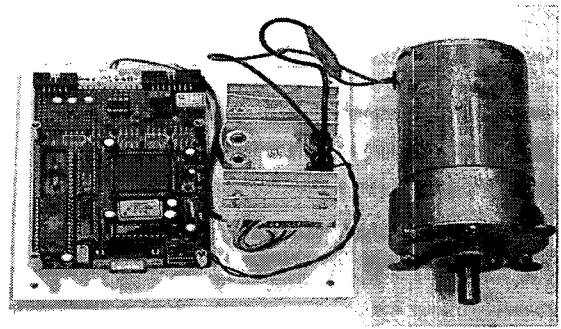


그림 13. 자율 조향 제어부  
Fig. 13. Control component of automatic steering

### 5.3. 영상 입력 시스템

차량의 전방의 영상을 획득하기 위하여 그림 14와 같이 USB 카메라를 백미러 위치에 설치하였다. USB 카메라를 회전시키기 위해서 서보모터를 장치하여 카메라를 일정 각도씩 회전시킬 수 있도록 하였다. 서보모터의 구동을 위해서 ATMEGA161 microcontroller를 이용하였으며 시리얼 인터페이스를 이용해 컴퓨터로부터 카메라의 각도를 제어할 수 있도록 구성하였다.

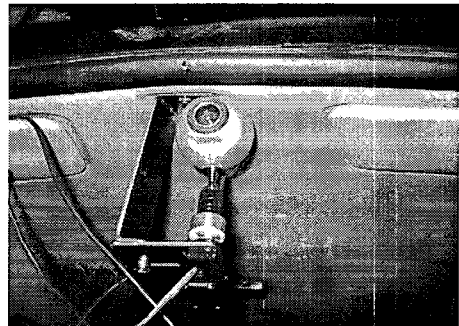


그림 14. 카메라 구동부  
Fig. 14. Camera and servo motor

## VI. 시스템 구현 결과 및 고찰

제작된 시스템을 이용하여 그림 15와 같이 주차장에서 무인 주차를 실험하였다. 다양한 시작 위치, 다양한 주변 차량의 위치 등의 설정하고 차량의 최종 주차 위치를 측정하였다.

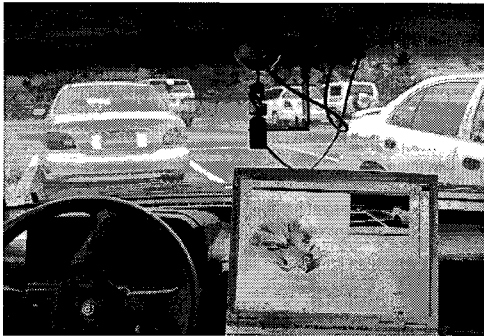


그림 15. 자동 주차 실험

Fig. 15. Experiment of automatic parking

실험 초기에는 하드웨어적인 문제점(모터의 반응 속도 지연, 조향각 AD 컨버팅의 정확도 문제, 속도계로부터 나오는 펄스 간격, 카메라의 셋팅 오차 등)에 의해 생성된 경로와 실제 이동한 경로의 차이가 많이 발생했지만 문제점들을 어느 정도 해결함으로써 생성된 경로와 실제 경로의 차이가 20cm 이내에 들어감을 알 수 있었다.

오차가 발생하는 주요 원인은 카메라의 셋팅 오차에 의한 거리 계산 착오이다. 카메라의 각도가 1도만 변해도 멀리 떨어져있는 영상에서의 거리는 1m정도 변하기 때문에 카메라의 각도를 미세하게 설정해줄 필요가 있다. 이와 같은 문제점을 해결하기 위해서는 카메라 각도의 변화를 측정할 수 있는 자이로 센서나 틸트 센서를 카메라에 부착하여 영상이 들어올 때마다 카메라의 현재 위치와 각도를 확인하는 방법을 활용하여 보완할 필요가 있다.

※ 본 연구에 대한 결과 동영상은 [http://ivhs.korea.ac.kr/parking\\_demo](http://ivhs.korea.ac.kr/parking_demo)에서 확인할 수 있다.

### VII. 결론

본 논문에서는 영상을 이용하여 주변 장애물을 인식하는 방법론과 영상의 변환을 통해 평면지도를 생성하는 기법을 설명하였고 장애물을 피해 원하는 위치와 방향에 차량을 위치시키는 방법에 대해 설명하였다.

표 1. 기존 연구들과 본 연구의 특징 비교

Table 1. Characteristic comparison between existing papers and this paper

논문 번호	센서	알고리즘 특징	생성 경로수	주차 유형	사용시스템
1	초음파	2개의 원호를 이용	1개	평행주차	모형자동차
2	외부의 CCD	퍼지	1개	평행주차, 수직주차	모형자동차
3	CCD	2 Step	1개	평행주차	CyCab <sup>3)</sup>
4	none	Wavelet, RBF neuralnet	1개	수직주차	시뮬레이션
5	none	퍼지	1개	평행주차	시뮬레이션
6	GPS, Gyro	PPR생성과 PPR를 이용한 경로 생성	1개	평행주차, 수직주차	실물자동차
7	초음파	2개의 원호를 이용	1개	평행주차	모형자동차
8	Laser 스캐너	속도의 한도를 일시 제어	1개	수직주차	CyCab
도요다의 프리우스	CCD	발표하지 않았음	1개	후방평행주차	실물자동차
본연구	CCD	SPM과 Bezier 곡선을 이용한 경로 수정, DAM	다양함	평행주차, 수직주차	실물자동차, 시뮬레이션

3) CyCab이란 무인자동차 연구팀 위한 전용차량

표 1에서 보는 바와 같이 본 연구에서는 기존의 연구들과는 다르게 다중의 경로를 생성하는 방법론을 개발하였다. 다중 경로 생성의 장점은 다양한 주차 유형(전후진 평행 주차, 전후진 수직 주차)에 대처할 수 있고, 돌발 상황이 발생해도 다른 경로를 선택할 수 있다는 것이다.

추후 영상 처리와 하드웨어에 대한 보완 연구를 통해 시스템의 정밀도를 높이고, 돌발 상황에 대비한 다중 경로 선택 방법론을 개발하여 시스템의 신뢰성을 높이는 연구가 필요하고 생각된다.

또한, 기존의 연구들이 특징적인 주차 유형에만 초점을 맞춰 연구해온 반면 본 연구는 일반적으로 장애물을 피하는 다중 경로 생성에 중점을 두었으므로 향후 무인 자동차 분야에서 다양한 장애물과 좁은 공간에서의 차량의 움직임을 제어하는 야지 주행, 골목길 주행 등의 분야에서 활용이 가능하다고 생각된다.

### 참고 문헌

- [1] Y. K. Lo, A. B. Rad, C. W. Wong and M. L. Ho, "Automatic parallel parking," *Intelligent Transportation Systems*, Vol. 2, pp.1190 - 1193, 2003.
- [2] T. S. Li and S. J. Chang, "Autonomous fuzzy parking control of a car-like mobile robot," *Systems, Man and Cybernetics*, Vol. 33, pp.451 - 465, 2003.
- [3] J. Xu, G. Chen and M. Xie, "Vision-guided automatic parking for smart car," *Intelligent Vehicles Symposium*, pp.725 - 730, 2000.
- [4] L. Li and F. Y. Wang, "Parking guidance system for front wheel steering vehicles using trajectory generation," *Intelligent Transportation Systems*, Vol. 2, pp.1770 - 1775, 2003.
- [5] K. Y. Lian, C. S. Chin and T. S. Chiang, "Parallel parking a car-like robot using fuzzy gain scheduling," *Control Applications*, Vol. 2, pp.1686 - 1691, 1999.
- [6] M. Wada, K. Yoon, H. Hashimoto, S. Matsuda and H. Masuda, "Development of advanced parking assistance system," *Intelligent Transportation Systems*, pp. 716 - 721, 1999.
- [7] K. Jiang and L. D. Seneviratne, "A sensor guided autonomous parking system for nonholonomic mobile robots," *Robotics and Automation*, Vol. 1, pp. 311 - 316, 1999.
- [8] I. E. Paromtchik and H. Asama, "Driver Assistance for Autonomous Parking Maneuvers," *Cybernetics and Informatics*, Vol. 7, pp. 138-143, 2003.
- [9] 노광현, "안전 주행을 위한 비전 기반 차량 탐지 방법," *고려대학교 박사학위논문*, 2001.
- [10] I. Rivals, L. Personnaz, G. Dreyfus and D. Canas, "Real-time control of an autonomous vehicle : a neural



network approach to the path following problem," *5th International Conference on Neural Networks and their Applications*, pp. 219-229, 1993.

[11] W. L. Nelson, "Continuous steering-function control of robot carts," *Industrial Electronics*, Vol. 36, pp.330-337 1989.

[12] D. F. Rogers and J. A. Adams, "Mathematical Elements for Computer Graphics," *McGraw-Hill*, 1990.



**박 정 술(Cheong-Sool Park)**

1979년 3월 5일생. 2003년 고려대학교 산업시스템정보공학과(공학사).

2003년~현재 고려대학교 대학원 산업시스템정보공학과 석사과정.

관심분야 : 영상처리, 지능형 자동차, ITS



**한 민 홍(Min-Hong Han)**

1964년 서울대학교 기계공학과(공학사).

1973년 University of Minnesota 산업공학과(공학석사).

1981년~1982년 숭실대학교 산업공학과 조교수.

1985년 Georgia Institute of Technology 산업공학과(공학박사).

1985년~1988년 Texas A&M University 산업공학과 조교수.

1988년~1990년 포항공과대학 산업공학과 부교수.

1991년~현재 고려대학교 산업공학과 정교수.

관심분야 : 지능형 자동차, 지능 로봇, ITS, 컴퓨터비전

---