무인운전차량의 자율주행을 위한 경로점 기반 경로계획

Path Planning for Autonomous Navigation of a Driverless Ground Vehicle Based on Waypoints

송 광 열, 이 준 웅*
(Gwang-Yul Song¹ and Joon-Woong Lee^{1,*})

¹Department of Industrial Engineering, Chonnam National University

Abstract: This paper addresses an algorithm of path planning for autonomous driving of a ground vehicle in waypoint navigation. The proposed algorithm is flexible in utilization under a large GPS positioning error and generates collision-free multiple paths while pursuing minimum traveling time. An optimal path reduces inefficient steering by minimizing lateral changes in generated waypoints along a path. Simulation results compare the proposed algorithm with the A* algorithm by manipulation of the steering wheel and traveling time, and show that the proposed algorithm realizes real-time obstacle avoidance by quick processing of path generation, and minimum time traveling by producing paths with small lateral changes while overcoming the very irregular positioning error from the GPS.

Keywords: path planning, autonomous vehicle, obstacle avoidance, collision-free waypoint navigation, GPS positioning error

I. 서론

미국에서는 DARPA가 주관하는 그랜드챌린지 및 어반챌린지(Grand Challenge/Urban Challenge)와 같은 자율주행자동차 경진대회가 개최되고 있고[1], 인터넷 기업인 구글은 스마트 카로 알려진 자율주행자동차를 개발하고 있다. 최근 국내에서도 완성차 업체와 정부의 지원 하에 자율주행자동차 경진대회가 매년 개최되고 있다[2].

자동차의 자율주행은 차량이 주행할 경로를 계획하는 것이 중요 이슈다. 본 연구도 자율주행을 위한 경로계획이 핵심이다. 연구에 사용된 차량은 차량 전방의 장애물 검출용 레이저 스캐너와 스테레오 카메라를 장착했고, 차량의 위치와 주행방향(heading) 식별을 위해 시중에서 시판되는 저가의 GPS수신기를 탑재하였다. 연구의 주 논점인 경로계획에서 보았을 때 문제는 GPS 수신데이터의 위치오차를 극복하는 것이다. 그 이유는 DGPS (Differential Global Positioning System) 수신기와 달리 연구에 사용된 GPS 수신기는 위치 오차가 크기 때문이다. 특히, 차량이 주행할 경로를 기준으로 횡방향으로 위치오차가 크면 그 만큼 차량이 주행경로를 이탈할 가능성이 크므로 이 문제의 해결이 중요하다.

본 연구에서 추구하는 자율주행은 주행해야 할 도로의 폭과 도로 중앙 위치가 경로점(waypoint)으로 주어진 상태에서 이루어지므로 경로점 추종(waypoint navigation)으로 볼 수 있다. 따라서 경로계획에서는 전역경로(global path)는 주어졌다고 간주한다. 주행중인 차량 전방의 장애물은 차량에 장착된 장애물 검출센서에 의해 식별되어, 그 결과는 차량 주행경로

가 설정될 맵상에 미리 표시된다. 이 때 장애물 검지영역 내에서 경로점을 추종하되 장애물을 회피할 수 있는 경로를 생성하는 것이 지역경로(local path) 생성이다[3, 12]. 본 연구에서는 지역경로 생성 및 갱신 속도를 높여 GPS의 위치오차가 크게 발생하거나 장애물이 나타나더라도 차량이 도로를 이탈하지 않고 주행할 수 있는 최적경로 생성을 목표로 한다. 단, 차량의 진행방향과 관련된 GPS관측 오차는 무시한다.

지역경로생성 알고리즘으로는 A*, D*[3-5], 비지어곡선 활용법[6] 등이 있다. A*알고리즘은 연산속도도 빠르고 주어진 전역경로를 최대한 유연하게 추종할 수 있는 경로를 생성하므로 자동차의 자동주차 등에 활용되고 있다[4]. 그러나 A*나 D*알고리즘은 저자들의 이전 연구[7]에서 보인 바와 같이 주어진 전역경로를 추종하려는 경향이 크기 때문에 GPS 수신데이터의 위치오차가 클 경우 횡방향으로 변동이 심한 S자 형태의 경로가 빈번하게 생성되는 문제가 발생한다. 따라서 현재 주어진 상황에 A*알고리즘을 적용하기는 곤란하다.

본 연구의 경로생성은 기존연구[3-7]와 달리 A*와 같은 특정한 기존 알고리즘에 의존하지 않고, 조향휠 조작과 주행속력 변화를 최소화하는 데에 초점을 맞춘다. 이를 위해 주어진 도로폭과 사용하는 GPS수신기의 공칭 위치오차를 합한 영역 내에서 경로점들을 연결한 경로와 평행한 다수의 경로를 생성하고[8], 이 가운데서 목적지에 가장 빨리 도달할 수 있는 경로를 선택한다. 따라서 GPS의 위치오차가 크더라도 S자 형태의 경로생성이 최소화되므로 부드럽고 빠른 주행을 꾀할 수 있다. 경로생성의 목적이 장애물 회피와 경로점 평행 추종이지만, 차선추종이 요구될 경우 차선관련 정보만 주어지면 본 경로생성법을 활용할 수 있다. 특히 주어진 도로폭 내에서 여러 개의 후보경로들이 0.012초 마다 생성되므로 돌발 장애물이 나타나더라도 이 장애물과의 간섭이 없는 경로가 존재하면 이 경로를 선택한다. 만약 모든 경로가 간섭이 될 경우 차량은 정지한다. 제한된 경로생성은 후진(backward

* Corresponding Author

Manuscript received August 21, 2013 / revised December 18, 2013 / accepted December 26, 2013

송광열, 이준웅: 전남대학교 산업공학파
(skyclass@nate.com/joonlee@chonnam.ac.kr)

※ 이 논문은 2011년도 교육과학기술부의 재원으로 한국연구재단지원을 받아 수행된 기초연구사업임 (No: 2011-0012323).

movement) 없는 전진만을 가정한다.

논문의 구성은 다음과 같다. II 장에서는 자동차의 운동모델과 경로점 개요를 설명한다. III 장에서는 논문에서 제안한 경로계획에 대해 기술한다. IV 장에서는 실험결과, 그리고 V 장에서 결론을 맺는다.

II. 자동차의 운동모델과 경로점

1. 자동차의 운동모델

본 연구의 경로계획에서는 S. Joachim 등[3]과 R. M. Murray [9]가 사용했던 그림 1에 보인 바와 같은 앞뒤 타이어를 가진 자동차에 대해 간단하게 묘사된 운동모델(kinematic model)을 이용한다. 이 모델은 뒤 타이어가 자동차의 중심축에 일치하고 앞 타이어는 수직축을 중심으로 회전을 허용한다. 또한 앞뒤 각각 한 쌍의 휠을 액슬의 중심에 있는 단순 휠로 모델링 했다. 차량의 상태(state)는 차량의 위치 (x, y) , 차량의 몸체가 수평축과 이루는 각 δ (차량의 heading) 그리고 차량의 몸체와 앞 바퀴가 이루는 각 ψ (조향각)로 구성된 (x, y, δ, ψ) 로 표기한다. 그림 1에서 y 축은 진북(true north), v 는 차량의 속도, L 은 차축거리, κ 는 차량의 궤적이 이루는 곡률을 나타낸다.

시점 t 에서 차량의 상태벡터 각 요소는 다음 식으로 표현된다[3].

$$\begin{pmatrix} x_t \\ y_t \\ \delta_t \\ \psi_t \end{pmatrix} = \begin{pmatrix} x_{t-1} + \varepsilon \cos(\delta_{t-1}) \\ y_{t-1} + \varepsilon \sin(\delta_{t-1}) \\ \delta_{t-1} + \frac{\varepsilon}{L} \tan(\psi_{t-1}) \\ \psi_{t-1} \end{pmatrix} \quad (1)$$

여기에서 ε 는 실험적으로 선택할 파라미터로 Δt 동안의 증분 값(step width)이다. 차량의 조향각 ψ 는 최대 조향 한계가 있으므로 다음과 같은 제약을 부과한다.

$$|\psi_t| < \psi_{\max}, \quad |\psi_t - \psi_{t-1}| < \Delta\psi \quad (2)$$

여기에서 $\psi_{\max}$ 는 최대 조향각이고, $\Delta\psi$ 는 순간 조향변화 가능량이다.

그림 1. 차량 운동모델.

Fig. 1. Kinematic model of an automobile.

표 1. 경로점 정보.

Table 1. Waypoint information.

기호	정보
p	경로점의 순서
$w_p(x, y)$	p^{th} 경로점의 위치
$Q_p(x, y)$	p^{th} 경로점에서의 도로 폭(m)
-	기타(제한속도, 도로형태 등)

2. 경로점

경로점(waypoint)은 차량이 주행해야 할 위치에 대한 위경도 좌표이다. 본 연구에서 경로점 정보는 표 1에 보인 바와 같이 위치뿐만 아니라 그 위치에서의 도로의 폭도 함께 제공된다. 경로점은 20m~30m간격으로 주어졌다. 앞으로 이 경로점들을 연결한 경로를 참경로(true path)로 호칭한다.

III. 경로계획

1. 기본 개요

상존하는 GPS위치오차로 인해 참경로와 일치된 위치로 차량을 주행시킬 수 없고, 차량 전방에 장애물이 있을 경우 이 장애물을 회피해야 하며, 곡로나 각이 진 코너에서 주행시간 단축을 위해 도로를 벗어나지 않고 장애물과 충돌이 없다면 참경로를 벗어날 수 있다.

이런 상황에 대응하려면 차량이 추종할 수 있는 경로를 생성해야 한다. 특히 장애물 회피를 위해 경로를 그림 2에 보인 바와 같이 변경구간과 주행구간으로 나눈다. 변경구간은 전방에 장애물이 있을 때 장애물을 피하기 위해 참경로와 간격을 조절하는 구간 또는 곡로에서 주행시간을 단축하기 위해 주행경로 변경을 위해 참경로와 간격을 조절하는 구간을 일컫는다. 주행구간은 참경로와 평행하게 일정한 간격을 유지하는 구간을 일컫는다.

2. 경로점 보충

경로점들은 이들을 통해 앞으로 차량이 주행할 도로형태를 예측할 수 있으므로 차량의 조향이나 속도제어에 중요한 역할을 한다. 그러나 제공된 경로점들은 너무 듬성듬성해서 단순히 이 경로점들을 선형으로 연결한 경로를 추종할 경우 급하게 차량을 회전시켜야 할 경우가 자주 발생한다. 따라서 부드러운 조향을 실현하기 위해 그림 3에 보인 바와 같이 제공된 경로점들 사이에 보조 경로점들을 생성시켜 참경로를 완만한 곡선의 형태로 만든다.

주어진 두 경로점 $w_p(x, y)$ 와 $w_{p+1}(x, y)$ 사이에 채울 보조

(a) Existence of obstacles.

(b) Partition of a path.

그림 2. 장애물 시나리오와 경로분할.

Fig. 2. Obstacles and partition of a path.

그림 3. 보조 경로점 및 경로점 평활화.

Fig. 3. Supplementary waypoints and waypoints smoothing.

경로점들은 간격이 0.5m 이하가 되게 했다. 이를 위해 각 보조점들의 증분치 $i(x, y)$ 를 식 (3)을 통해 결정한다.

$$i(x, y) = 0.5 \frac{w_{p+1}(x, y) - w_p(x, y)}{\|w_{p+1}(x, y) - w_p(x, y)\|} \quad (3)$$

여기에서 0.5m는 실험적으로 결정한 값이고, $\|\cdot\|$ 은 두 점 사이의 유클리디안 거리이다. 증분치를 이용하여 q 번째 보조 경로점 $P_q(x, y)$ 를 다음과 같이 생성한다.

$$P_{j+k}(x, y) = P_{j+k-1}(x, y) + i(x, y), \quad k = 1, 2, \dots \quad (4)$$

여기에서 $P_j(x, y) = w_p(x, y)$ 이다. p 번째 경로점 $P_p(x, y)$ 의 평활화는 이 경로점 전·후로 r 개의 경로점들을 택하여 다음과 같이 수행한다.

$$s_p(x, y) = \frac{\sum_{k=p-r}^{p+r} P_k(x, y)}{2r+1} \quad (5)$$

여기서 $s_p(x, y)$ 는 p 번째 평활화된 경로점이며, r 은 주어진 경로점들의 거리에 따라 달라질 수 있는데, 원하는 평활화 정도에 맞게 실험을 통해 결정한다.

3. 기준 경로의 생성

식 (5)에 의해 평활화된 경로점들을 연결한 경로를 수정된 참경로라고 하자. 그림 4에 보인 바와 같이 이 참경로에 평행한 경로를 현재 차량의 위치를 이용하여 다음 절차에 의해 설정한다. 1) 현재 차량의 위치와 가장 가까운 경로점을 찾고 이 점을 $s_c(x, y)$ 라 한다. 2) $s_c(x, y)$ 와 $s_{c+1}(x, y)$ 을 연결하는 선과 현재 차량의 위치 사이의 거리를 구하고 이 거리를 d 라 한다(그림 4(a)). 3) 차량에 장착된 레이저 스캐너의 검지거리 R 을 이용하여 $s_c(x, y)$ 로부터 R 만큼 떨어진 경로점 $s_{c+m}(x, y)$ 을 찾는다. 일례로 R 이 50m라면 평활화된 경로점 사이의 거리가 0.5m이므로 m 은 100이 된다. 4) $s_c(x, y)$ 를 포함하여 $(m+1)$ 개의 평활화된 경로점들 각각에 대해 자신과 자신 다음에 오는 경로점을 연결하는 선을 만들고 이 선과 상기 2)에서 얻은 d 를 이용하여 그림 4(b)에 보인 바와 같이 새로

그림 4. 기준경로의 생성.

Fig. 4. Creation of a reference path.

운 기준점들 $b_k(x, y)$, $k = 0, \dots, m$ 을 생성한다. 이와 같이 만든 기준점들을 연결한 경로를 기준경로(reference path)라 한다.

4. 후보경로 생성

차량이 주행 중에 장애물을 만날 경우, 도색 표식으로 구획된 차선을 추종해야 할 경우, GPS 위치오차가 클 경우 주어진 경로점 추종이 어려울 수 있다. 이 때 차량이 주행 가능한 경로를 제공해야 한다. 현재 우리의 연구환경에서는 GPS 위치오차가 상존하므로 차량이 주행할 수 있는 경로생성은 필수불가결하다. 이를 위해 참경로를 기준으로 좌·우로 주어진 도로폭과 사용중인 GPS 수신장치의 사양으로 제공된 위치오차를 합한 범위 내에서 여러 개의 경로를 생성한다. 그러나 도로폭은 제한된 경로계획 알고리즘에 큰 변수는 아니다. 도로폭이 좁으면 후보경로의 수량이 적고, 넓으면 많을 뿐이다. 생성할 후보경로들의 간격은 실험을 통해 0.5m로 정했다. 각 후보경로는 그림 2에서 설명한 것처럼 변경구간과 주행구간으로 나뉜다. 변경구간은 차속의 함수로 정하는 것이 보통이지만, 본 연구에서는 10m~20m 범위에서 실험을 통해 적정거리를 설정하였다. 각 후보경로는 기준경로의 기준점들과 동일한 수량의 가상 경로점들을 생성하며, 모든 후보경로들의 시작점은 현재 차량의 위치인 $b_0(x, y)$ 로 정한다.

후보경로 l 의 변경구간은 다음과 같이 형성된다. 1) 설정된 변경구간의 끝에 위치한 기준경로의 점을 $b_n(x, y)$ 라 한다. 2) $b_n(x, y)$ 과 이에 대응하는 경로 l 의 가상 경로점 $c'_n(x, y)$ 사이에 위치할 가상 경로점들을 정하기 위한 증분치를 식 (6)에 의해 계산한다.

$$\Delta'(k) = \frac{\|b_n(x, y) - c'_n(x, y)\|}{n+1} \times k, \quad k = 0, 1, 2, \dots, n \quad (6)$$

여기에서 위 첨자 l 은 후보경로의 인덱스, $n+1$ 은 변경구간내의 기준점들의 수량, $\|\cdot\|$ 은 기준경로와 후보경로 l 사이의

그림 5. 후보경로의 생성.

Fig. 5. Creation of candidate paths.

거리를 나타낸다. 3) 그림 5에 보인 바와 같이 기준점들의 법선방향으로 $\Delta'(k)$ 떨어진 곳에 가상 경로점을 정한다. 후보 경로 1의 주행구간의 가상 경로점들은 그림 5에 보인 바와 같이 기준점들의 법선방향으로 0.5m의 배수만큼 떨어진 곳에 정한다.

5. 후보경로 수정

앞 절에서 생성한 후보경로들은 조향각, 헤딩, 장애물 유무 등의 고려 없이 만들었기 때문에 차량이 추종할 수 없을 수 있다. 따라서 장애물 유무 검사와 함께 차량이 추종할 수 있도록 경로를 수정한다. 수정과정에서는 II 장에서 설명한 차량운동모델과 pure pursuit에서 사용된 목표점(goal point)법 [10]을 이용하여 차량이 통과할 수 있는 점들을 계산해 낸다. 이 때 장애물과의 충돌 가능성 판단은 장애물의 위치가 표기된 위험지도(risk map) [4]상에서 이루어진다. 또한 기준경로도 후보경로의 하나로 간주한다. 현재 차량의 위치 $b_0(x,y)$ 는 모든 후보경로들의 시작점 (x_i, y_i) , 즉 첫 통과점이 된다.

단계 1: 경로수정이 이루어지지 않은 경로 1이 남아 있다면 단계 2로 가고, 그렇지 않다면 경로수정을 끝낸다.

단계 2: 차량을 둘러싸는 최소사각형(bounding box)과 위험지도상의 장애물과의 간섭여부를 검사한다. 간섭이 있다면 현 경로의 수정을 종료하고 단계 1로 가고, 간섭이 없다면 단계 3으로 간다.

단계 3: 차량 규격(specification)으로 주어진 최소 회전반경 밖에서 이 회전원과 거리가 0.5m 이내에 있는 가상 경로점들 가운데 인덱스가 가장 큰 $c'_k(x,y)$ 을 찾는다. 그 이유는 차량이 주행할 수 있는 자세를 취할 수 있도록 하고, 각이 진 코너에서 경로가 꼬이는 것을 방지하기 위해서다. 이 $c'_k(x,y)$ 를 목표점이라 하고(그림 6 참조), 변수 D_i 에 인덱스 k 를 저장한다.

단계 4: 목표점이 존재하지 않을 경우 현 경로의 수정을 종료하고 단계 1로 간다.

단계 5: 목표점까지의 주행방향(heading)을 다음과 같이 계산한다.

- (x_i, y_i) 로부터 목표점까지의 벡터와 현재 헤딩의 벡터를 각각 $\mathbf{f}$ 와 $\mathbf{g}$ 로 놓는다.

- $\mathbf{f}$ 와 $\mathbf{g}$ 사이의 각을 계산한다. $\Delta\delta_i = \cos^{-1}\left(\frac{\mathbf{f} \cdot \mathbf{g}}{\|\mathbf{f}\| \|\mathbf{g}\|}\right)$. 식

그림 6. 목표점 선정.

Fig. 6. Searching for a goal point.

(2)에 의해 조향각을 계산한다. $\psi = \left| \tan^{-1}\left(\frac{\Delta\delta_i \cdot L}{\varepsilon}\right) \right|$. 그러

나 조향각은 식 (3)에 기술한 것처럼 한계가 있기 때문에 다음과 같이 결정한다.

$$\psi_i = \begin{cases} \min(\psi, \psi_{\max}, \psi_{i-1} + \Delta\psi), & \text{if } g_x f_y - g_y f_x > 0 \\ \max(-\psi, -\psi_{\max}, \psi_{i-1} - \Delta\psi), & \text{if } g_x f_y - g_y f_x < 0 \end{cases}$$

- ψ_i 를 이용하여 헤딩을 계산한다. 즉 $\delta_i = \delta_{i-1} + \Delta\delta$, 여기

에서 $\Delta\delta = \frac{\varepsilon}{L} \tan \psi_i$.

단계 6: 식 (2)에 의해 차량의 상태벡터를 갱신한다. 갱신된 상태벡터의 (x_{i+1}, y_{i+1}) 가 다음 통과점이 된다. (그림 7참조) 따라서 경로상의 통과점 수량은 식 (2)의 ε 의 영향을 받는다. 본 연구에서 ε 은 0.7m를 택하였는데 통과점 수량을 늘리려면 작은 값을 취하고, 수량을 줄이려면 큰 값을 취한다.

단계 7: 새로운 통과점에서의 차량의 속력을 다음 식에 의

해 계산한다[11]. $v = \min\left(\alpha \sqrt{\frac{1}{\kappa}}, 100\right)$, 여기에서 $\frac{1}{\kappa} = \frac{L}{\tan(\psi)}$

이며 α 는 속력 안정계수로 곡률이 κ 인 도로를 안정적으로 주행할 수 있도록 실험적으로 선택한다. 그리고 새로운 통과점까지의 경과 시간(lap time)은 $T_{i,i} = T_{i,i-1} + \frac{\varepsilon}{v}$ 이다.

단계 8: 통과점 (x_{i+1}, y_{i+1}) 을 저장하고 단계 2로 간다.

6. 최적경로 선정

경로수정 알고리즘을 통해 수정된 경로들 가운데 D_i 에 저장된 목표점의 인덱스와 경과시간 T_i 에 의해 하나의 경로가 최종 경로로 선정된다. 경로 선정과정에 D_i 이 T_i 보다 우선한다. 그 이유는 D_i 의 값이 크다는 것은 차량이 장애물과의

그림 7. 새로운 통과점 결정.

Fig. 7. Determination of a new passage point.

충돌 없이 목적지까지 경로를 잘 추종한다는 의미이기 때문이다. D_i 값이 같은 경로가 여럿 있을 때 그 가운데 T_i 가 가장 작은 경로가 선정되면 이 경로상의 통과점들이 차량 제어기에 입력된다.

IV. 실험결과

1. 실험차량

본 연구에 사용된 차량은 그림 8에 보인 바와 같다. 장애물 검출을 위해 레이저스캐너와 스테레오카메라를 장착하였고, 차량의 heading과 자기위치 인식을 위해 GPS 수신기를 탑재했으며, 차량의 자율조향을 위해 전동기를 장착했다.

2. GPS 수신데이터 필터링

연구에 사용한 GPS 수신 데이터는 차량이 일직선으로 주행을 해도 계속 심하게 흔들렸고, 특히 저속이거나 정지상태에서 오차는 더욱 컸다. 따라서 가공 없이 원시 데이터를 바로 사용할 수 없어서 그림 9에 보인 바와 같이 빨간색 관측값을 칼만 필터를 사용하여 파란색으로 표현된 것처럼 데이터의 변동폭을 줄였고, 경로추종에서는 이 필터링된 데이터를 사용하였다.

3. 장애물검출

그림 8에 보인 바와 같이 노면위의 장애물을 검출하기 위해 세 대의 레이저 스캐너를 차량 범퍼 부근의 중앙과 좌우에 설치하였다. 그림 10은 이 스캐너들로 검출한 장애물을 위험지도에 표시한 것이다.

그림 8. 실험차량과 센서 및 조향장치.

Fig. 8. Experimental vehicle, sensors and steering actuator.

그림 9. GPS관측 데이터에 의한 차량위치.

Fig. 9. Vehicle position by GPS observation data.

4. 경로계획에 대한 시뮬레이션 결과

제안된 경로계획 알고리즘의 성능확인을 위해 그림 10에 보인 바와 같이 차량에 장착된 스캐너에 의해 장애물 검출을 하고 이 검출결과가 표시된 위험지도상에서 경로생성 시뮬레이션을 수행하였다. 연구에 사용된 GPS 수신기는 항상 위치오차가 큰 데이터를 제공했지만, 시뮬레이션은 GPS 위치오차가 있는 경우와 없는 경우로 나누어 수행하였다.

표 2는 단순 사각형(경로-1), 8자형경로(경로-2), S자형 경로(경로-3), 실제 지도로부터 얻은 경로(경로-4) 등의 형태로 주어진 참경로에 대해 장애물이 없고, GPS 위치오차가 없다고 가정된 상황에서 제안된 경로계획 알고리즘이 선택한 경로의 주행거리를 나타낸 것이다. 이 시뮬레이션에서는 도로폭 10m를 가정하였는데, 그 이유는 자율주행자동차 경진대회가 열렸던 곳의 도로폭이 최대 10m였기 때문이다. 도로폭에 따라 생성된 후보경로의 수량과 최종적으로 선택된 경로의 주행거리는 달라질 수 있다. 표 2에 보인 바와 같이 선택된 경

그림 10. 위험지도상에 표시된 장애물 검출 예: (a) 통합 위험지도, (b) 스캐너①에 의한 위험지도, (c) 스캐너②에 의한 위험지도, (d) 스캐너③에 의한 위험지도, (e) 스테레오시스템의 왼쪽 카메라로 촬영된 영상과 장애물 표시.

Fig. 10. Obstacles displayed on a risk map: (a) Combined risk map, (b) Risk map by scanner①, (c) Risk map by scanner②, (d) Risk map by scanner③, and (e) Image from left camera of stereo cameras.

표 2. 선택된 경로의 주행거리 계산.

Table 2. Driving distance of a selected path.

구분	경로-1	경로-2	경로-3	경로-4
참경로	1083.6m	821.1m	286.3m	3656.1m
선택 경로	1032.6m	751.8m	222.6m	3574.2m

그림 11. GPS수신 위치오차가 없는 경우의 경로.

Fig. 11. Path generation under no error from GPS.

로의 거리는 참경로보다 짧다.

그림 11은 GPS 위치오차가 없다는 가정하에 수행한 시물레이션 결과이다. 따라서 도로중앙부에서 미리 취득한 참경로는 위험지도의 중앙부에 나타나 있다. 그리고 도로 가장자리와 중앙부에 장애물들이 존재한다. 그림 11(a)에 보였듯이 A*알고리즘은 참경로에 회귀하려는 경향이 매우 강하다. 따라서 장애물들이 있는 경우 핸들 조작을 빈번하게 요하는 경로를 생성할 수 밖에 없고, 장애물을 피하는 경로가 생성되더라도 이 경로가 장애물에 지나치게 근접되므로 충돌가능성도 크다. 반면에 제안된 알고리즘이 생성한 후보경로들은 모두 참경로와 평행하므로 들쭉날쭉한 경로가 없고, 이들 가운데 장애물과 간섭 없이 목표지점에 가장 빨리 도달할 수 있는 경로를 택한다. 그림에서 빨간색으로 표시된 경로이다.

그림 12는 실제 상황에 근사하게 GPS 수신 데이터에 위치오차가 크게 존재하고 차량이 주행하는 도로의 가장자리에 장애물들이 존재할 경우에 행해진 시물레이션 결과이다. 이 위치오차 때문에 도로중앙부에서 얻은 참경로가 도로를 벗어난 경우가 많기 때문에 이 예는 그런 경우를 다룬 것이다.

그림 12. GPS 수신 위치오차 및 장애물이 있는 경우의 경로.

Fig. 12. Path generation under the error from GPS, and obstacles.

그림 13. 급 코너부근에서 경로생성.

Fig. 13. Path generation at a nose corner.

이미 그림 11에서 보였듯이 A*알고리즘이 생성한 경로는 장애물을 회피하지만, 참경로를 추종하려는 경향이 커서 들쭉날쭉하고 장애물에 접근되므로 빈번한 핸들조작을 요하며 장애물과의 충돌 가능성도 크다. 반면에 제안된 알고리즘은 참경로와 평행한 경로들을 생성하고 이 가운데서 장애물과의 충돌 없이 목적지에 가장 빨리 도달할 수 있는 경로를 선택하였다. 그림 11(b)와 그림 12(b)에 제시된 일부 후보경로들이 장애물 앞에서 끊겨 있는 것을 볼 수 있는데, 이것은 후보경로 수정과정의 장애물과의 충돌 가능성 검사에서 충돌이 가능한 것으로 인식된 결과이다.

그림 13은 노변에 장애물이 많고 급격한 코너 부근에서의 경로생성을 보인 예다. 그림 13(a)는 전역경로를 보인 것이고 그림 13(b)는 현 차량의 위치에서 제안된 알고리즘에 의해 지역경로를 생성한 것이다. 장애물과의 간섭이 없고 차량이 주행 가능하다면 최단 경로를 선택하고 있음을 알 수 있다.

그림 14는 제안된 경로계획 알고리즘이 직각에 가까운 코너부근에서 생성한 경로를 보인 것이다. 노변에 있는 장애물들은 보도와 차도를 구획하는 연석(curb)이다. 그림 13의 왼쪽은 차량이 지나왔고 또 지나갈 경로를 연속적으로 보인 것이고 오른쪽은 현재 차량의 위치에서 생성된 후보경로들과 선택된 경로를 확대하여 보인 것이다.

그림 14. 직각에 가까운 코너부근에서 경로생성.

Fig. 14. Path generation at the corner with right angle.

그림 15. 다양한 시나리오에서의 경로생성: (a) 막힌 곳, (b) 단일 장애물, (c) 두 장애물이 엇갈려 있을 때, (d) 두 장애물이 나란히 있을 때.

Fig. 15. Path generation under various scenarios: (a) dead end, (b) single obstacle, (c) cross-located double obstacles, and (d) evenly aligned double obstacles.

그림 15의 시뮬레이션 결과는 도로상에 막혀 있는 곳 (dead-end)이나 다양한 위치에 장애물들이 있을 때 제한한 알고리즘이 어떻게 경로를 생성하는지 보인 것이다.

V. 결론

주어진 경로점을 정확히 추종하려면 차량의 자기위치 인식이 정확해야 한다. 그러나 현재 시중에 시판되는 저가의 GPS 수신기가 제공하는 데이터는 5m 정도의 위치오차가 상시 내포되어 있어서 주어진 경로점을 곧바로 추종할 수 없다. 제안된 알고리즘은 이렇게 큰 오차가 수반된 차량의 자기위치 데이터로 다양한 형태의 장애물이 존재하는 도로에서 자율주행을 수행할 수 있는 경로생성을 목표로 했다.

이 목표 실현을 위해 A*알고리즘을 적용하여 경로를 생성하였는데, 4장의 시뮬레이션 결과에서 보였듯이 생성된 경로는 주어진 경로점을 연결한 참경로를 추종하려는 경향이 매우 강했다. 그러나 차량의 자기위치 인식 데이터의 오차가 커져 주어진 경로점을 연결한 참경로는 도로영역밖에 위치하는 경우가 많다. 이러한 상황에서 A*알고리즘이 생성한 경로는 매우 들쭉날쭉하여 빈번한 핸들조작을 요하고, 차량의 속도도 높일 수 없었다.

따라서 본 연구에서는 새로운 경로생성 알고리즘을 제안하였다. 생성된 경로는 차량의 자기위치 인식 데이터의 큰 오차를 극복하기 위해 주어진 경로점을 연결한 참경로에 평행하도록 했다. 이렇게 함으로써 A*알고리즘이 생성한 경로가 갖는 문제점을 극복할 수 있었다. 특히 제안된 알고리즘은 Intel Core2 Duo E8400 CPU에서 0.012초만에 경로를 갱신하므로 장애물이 돌발적으로 나타난 경우도 대응할 수 있다.

REFERENCE

- [1] http://en.wikipedia.org/wiki/DARPA_Grand_Challenge.
- [2] J. H. lee, E. H. Jung, and B. C. Ko, "Status and prospects of intelligent autonomous vehicle contest," *Korea Multimedia Society*, vol. 14, no. 1, pp. 31-41, 2010.
- [3] S. Joachim, T. Gindele, D. Jagszent, and R. Dillmann, "Path planning for cognitive vehicles using risk maps," *IEEE*

Intelligent Vehicles Symposium, pp. 1119-1124, 2008.

- [4] A. Stentz, "Optimal and efficient path planning for partially known environments," *Proc. IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3310-3317, 1994.
- [5] J. H. Im, S. H. You, G. I. Jee, and D. H. Lee, "A path generation algorithm for obstacle avoidance in waypoint navigation of unmanned ground vehicle," *Journal Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 8, pp. 843-850, 2011.
- [6] J. Choi, "Path planning based on Bezier curve for autonomous ground vehicles," *Proc. of 2008 World Congress on Engineering and Computer Science*, pp. 158-166, 2008.
- [7] G. Y. Song and J. W. Lee, "Path generation for autonomous vehicle using A* algorithm," *Proc. of 2012 27th ICROS Annual Conference (ICROS 2012)*, pp. 125-126, 2012.
- [8] A. Lacaze, Y. Moscovitz, N. DeClariss, and K. Murphy, "Path planning for autonomous vehicles driving over rough terrain," *Proc. of the IEEE/ISIC/CIRA/ISAS Joint Conference*, pp. 50-55, 1998.
- [9] R. M. Murray, "Nonholonomic motion planning: steering using sinusoids," *IEEE Transactions on Automatic Control*, vol. 38, pp. 700-716, 1993.
- [10] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1992.
- [11] Y. H. Lee, "Automatic speed control system for vehicle approaching and driving on a curve," *ASME 2008 International Mechanical Engineering Congress and Exposition*, pp. 345-353, 2008.
- [12] W. Hopkins, T. J. Song, and A. D. Long, "Saturn launch vehicle's navigation," *Automatica*, vol. 7, no. 9, pp. 26-35, 2004.

송 광 열

제어·로봇·시스템학회 논문지 제17권 제4호 참조.

이 준 응

제어·로봇·시스템학회 논문지 제19권 제6호 참조.