

신경 회로망을 이용한 보행자와의 충돌 위험 판단 방법

Collision Risk Assessment for Pedestrians' Safety Using Neural Network

김 범 성, 박 성 근, 최 배 훈, 김 은 태*, 이 희 진, 강 형 진
(Beomseong Kim¹, Seongkeun Park¹, Baehoon Choi¹, Euntai Kim¹, Heejin Lee², and Hyung-Jin Kang³)

¹Yonsei University

²Hankyong National University

³MANDO CORP.

Abstract: This paper proposes a new collision risk assessment system for pedestrians's safety. Monte Carlo Simulation (MCS) method is a one of the most popular method that rely on repeated random sampling to compute their result, and this method is also proper to get the results when it is unfeasible or impossible to compute an exact result. Nevertheless its advantages, it spends much time to calculate the result of some situation, we apply not only MCS but also Neural Networks in this problem. By Monte carlo method, we make some sample data for input of neural networks and by using this data, neural networks can be trained for computing collision probability of whole area where can be measured by sensors. By using this trained networks, we can estimate the collision probability at each positions and velocities with high speed and low error rate. Computer simulations will be shown the validity of our proposed method.

Keywords: intelligent vehicle, monte carlo, neural networks, collision risk, monte carlo simulation

I. 서론

오늘날 지능형 자동차에 대한 연구는 발전을 거듭하면서 지능형 자동차 시스템이 양산형 차량 시장에 보급되기 시작함에 따라 지능형 자동차에 대한 연구는 더욱 더 활발해지고 있다.

특히 지능형 자동차 부분에서도 보행자의 안전을 위한 시스템이 각광 받고 있다. 유럽의 경우 2010년부터 보행자 안전 시스템이 의무화가 예정되어 있고[1], 국내에서도 2007년부터는 국토해양부 주관 신차안전도 평가에서 보행자 안전성에 대한 부분을 추가하여 시험하고 있다[2]. 차량과 보행자의 충돌 사고의 경우 보행자가 사망, 골절, 화상과 같은 치명적인 부상을 입을 수 있기 때문에 이러한 피해를 예방하는 것이 중요할 것이다.

보행자의 피해를 예방하는 방법으로는 사고가 일어난 후에 피해를 감소시키는 방법과 사고의 위험을 미연에 예측하여 사고의 가능성을 감소시키는 방법 이렇게 두 가지로 분류할 수 있다. 사고가 일어난 후에 피해를 감소시키는 방법으로는 차량 제작시 범퍼, 후드, 브러쉬 같이 보행자와 충돌 가능성이 있는 부분들을 곡선으로 제작하는 방법과 차량이 물체와 충돌할 때, 장애물의 종류를 구분하여 보행

자일 경우에는 보행자용 에어백 등과 같은 보행자 안전장치를 작동시킴으로서 보행자를 보호하는 시스템들이 개발되고 있다.

이러한 시스템이 장착된 차량이 도심과 같은 변화가 잦은 지역에서 보행자의 안전을 보호하기 위해서는 주변의 장애물을 정확히 인식하고[3-5] 추적하는 것이 중요하다 [6,7]. 또한 추적하고 있는 장애물이 자동차에 어느 정도의 충돌 위험성이 있는가 판단하는 것도 보행자 안전 확보에 큰 영향을 미칠 것이다[8].

추적하고 있는 장애물이 자동차에 어느 정도의 충돌 가능성이 있는가를 판단하기 위해서는 먼저 장애물을 특성에 따라서 분류하는 것이 좋은데 주로 보행자와 차량과 같은 동적 장애물과, 중앙분리대, 가드레일과 같은 정적 장애물 등으로 크게 분리할 수 있다. 정적 장애물의 경우에는 움직임이 없이 고정되어 있기 때문에 자동차가 특정 위치를 피해서 주행하면 큰 위험이 없지만, 동적 장애물의 경우에는 장애물이 계속적으로 움직이기 때문에 장애물을 피해서 주행하는 것이 조금 더 어렵다[9].

이러한 동적 장애물은 크게 차량과 보행자로 구분할 수 있는데, 차량의 경우에는 상대적으로 보행자보다 센서로 인식하기가 쉽고 인식 가능한 범위도 넓다[10]. 그리고 차량은 가질 수 있는 최대 속도의 범위가 매우 넓고 변화가 매우 큰 반면에 좌우 이동과 같은 이동 방향의 변화는 상대적으로 제한적이다. 보행자는 이와 반대로 센서가 인식할 수 있는 범위가 상대적으로 좁고, 오차도 크다. 그리고 보행자가 가질 수 있는 최대 속도의 범위는 상대적으로 작지만, 구조상의 한계가 없기 때문에 이동방향의 변화 폭이 매우 크다는 특성을 가진다.

* 책임저자(Corresponding Author)

논문접수: 2010. 6. 17., 수정: 2010. 7. 16., 채택확정: 2010. 12. 14.

김범성, 박성근, 최배훈, 김은태: 연세대학교 전기전자공학부

(battlubs@yonsei.ac.kr/keiny@yonsei.ac.kr/choibae@yonsei.ac.kr/etkim@yonsei.ac.kr)

이희진: 한경대학교 정보제어공학과(lhjin@hknu.ac.kr)

강형진: (주)만도 중앙연구소(hjkang@mando.com)

※ 본 연구는 (주)만도의 능동적 보행자 보호 시스템 개발 사업의 연구비 지원에 의해 수행되었음.

따라서 동적 장애물의 위험도를 판단할 때, 차량과 보행자를 나누어서 판단하는 것이 시스템의 성능을 향상시키는 데 유리하며, 특히 보행자의 경우에는 센서가 인식 가능한 범위가 좁고 오차가 매우 크며 조향 및 속도 변화가 크기에, 위험도를 판단하는 것이 매우 중요하다.

그래서 본 논문에서는 동적 장애물 중에서도 중요성이 상대적으로 높은 보행자와의 충돌 위험성을 판단하는 알고리즘을 제안한다. II 장에서는 센서가 장착된 자동차와 보행자의 위치, 속력에 따라서 충돌 유무 판단 방법을 보이고, III 장에서는 특정 입력에서의 충돌 확률을 판단하기 위한 Monte Carlo 모의실험과 충돌 확률 결과를 전체 영역으로 확장시키기 위한 신경망에 대하여 설명하고, IV 장에서는 실험을 통하여 제시한 알고리즘의 적합성을 보인다. 그리고 V 장에서는 결론 대하여 설명한다.

II. 충돌 사건 판단

도로상에서 보행자와 차량은 그림 1(a)와 같이 서로 독립적으로 움직인다. 여기서 $x_p, y_p, v_{x,p}, v_{y,p}$ 는 각각 보행자의 x 좌표, y 좌표, x축 속도, y축 속도를 나타내고 $x_v, y_v, v_{x,v}, v_{y,v}$ 는 차량의 x 좌표, y 좌표, x축 속도, y축 속도를 나타낸다.

이때 차량과 보행자가 충돌하기 위한 조건은 두 물체의 x, y 좌표 값의 차이가 차량의 길이의 1/2 이하로 줄어드는 것이다.

$$x_p - x_v \leq C_1/2 \tag{1}$$

$$y_p - y_v \leq C_2/2 \tag{2}$$

여기서 C_1 은 차량의 전폭(overall width, 너비)을 나타내고, C_2 는 차량의 전장(overall length, 길이)을 나타낸다. 따라서 (1)과 (2)가 동시에 만족하는 순간이 차량과 보행자가 충돌하는 순간이 될 것이다.

하지만 특정 시간에서 두 물체의 충돌 유무를 판단하기 위해서는 차량의 위치를 원점(0,0)으로 놓고 보행자의 상대적인 위치를 계산하여 차량의 크기와 비교해 보아도 된다. 즉 (3)과 같이 x축으로의 상대 좌표는 보행자의 x 좌표에서 차량의 x 좌표를 빼는 방법을 통해 구할 수 있고 구한 상대 좌표 x_{v-p} 는 앞으로 x로 표기한다. y축으로의 상대 좌표도 이와 같은 방법으로 구할 수 있고 앞으로는 y로 표

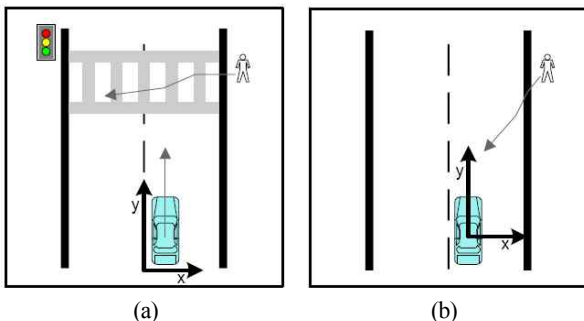


그림 1. 차량과 보행자의 움직임.
Fig. 1. Movement of vehicle and pedestrian.

기한다. (3)과 (4)가 두 물체의 위치를 상대 위치로 변환했기 때문에 차량과 보행자의 속도 또한 상대 속도로 변환하여 사용할 수 있다. 즉 (5)와 같이 x축 상대 속도 $v_{x,v-p}$ 를 구하고 이는 앞으로 v_x 로 표기한다. y축 상대 속도 $v_{y,v-p}$ 도 같은 방법으로 구하고 v_y 로 표기한다.

$$x = x_{v-p} = x_p - x_v \tag{3}$$

$$y = y_{v-p} = y_p - y_v \tag{4}$$

$$v_x = v_{x,v-p} = v_{x,p} - v_{x,v} \tag{5}$$

$$v_y = v_{y,v-p} = v_{y,p} - v_{y,v} \tag{6}$$

물체의 위치와 속도를 인지하기 위한 센서는 여러 가지가 있을 수 있지만, 레이더를 사용한다면 도플러 효과에 의해서 상대 위치와 상대 속도를 바로 알 수 있기 때문에 뒤에 나오는 실험에서는 레이더에서 받은 x축, y축의 상대 위치 x, y 와 상대 속도 v_x, v_y 4개를 충돌 판단을 위한 실험의 입력 값으로 사용한다.

III. 충돌 확률 계산

1. Monte Carlo 모의실험

임의의 상태에서 충돌 확률을 구하는 것은 일반적인 방법으로는 불가능하기 때문에 모의실험을 여러번 반복한 후에 충돌한 사건 수를 총 실험 수로 나누어 충돌 확률을 구한다[11]. 특정한 상대 위치와 속도에서 충돌 위험 확률을 구하기 위해서는 시간의 경과에 따라서 위치와 속도의 변화를 예측하는 것이 중요하다. 일정한 시간 간격 T초 후에 x, y 좌표의 위치 값의 변화는 다음 (7), (8)과 같이 나타나게 된다.

$$x(t+T) = x(t) + T \times v_x(t) \tag{7}$$

$$y(t+T) = y(t) + T \times v_y(t) \tag{8}$$

하지만 T초 동안 속도 v_x 와 v_y 는 일정하지 않고 계속 변화하고 있기 때문에 잡음 w 를 고려해 주어야 한다. (9)와 (10)과 같이 속도에 잡음을 주어서 속도 변화를 나타낸다.

$$v_x(t+T) = v_x(t) + w_1(t) \times T \tag{9}$$

$$v_y(t+T) = v_y(t) + w_2(t) \times T \tag{10}$$

위와 같이 (9)와 (10)을 이용하면 T 시간 후에 위치를 알 수 있다. 만약 이 위치가 차량의 크기와 같거나 작아지게 된다면 보행자와 차량이 충돌했다고 판단한다.

같은 실험을 일정 횟수 반복한 후에 충돌한 사건을 총 사건의 수로 나누면 특정 입력에서의 충돌 확률 값을 구할 수 있게 된다.

$$\text{충돌 위험 확률} = \frac{\text{충돌 사건}}{\text{총 사건}} \tag{11}$$

2. 신경망 시스템

Monte Carlo 모의실험을 통하여 특정 입력에서의 충돌 확률을 구할 수 있지만, 매번 입력 값이 들어 올 때마다 Monte Carlo 모의실험을 반복하는 것은 충돌 확률을 계산

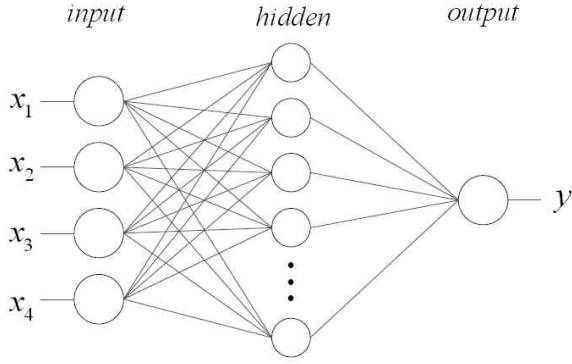


그림 2. 다층 퍼셉트론 신경망의 구조.

Fig. 2. Structure of multi-layer perceptron neural networks.

하는데 걸리는 시간이 매우 크기 때문에 실시간으로 위험을 판단하는 시스템에는 부적합하다. 이러한 문제는 최소 제곱법으로 구한 계수를 사용하여 해결할 수도 있지만, [11] 최소 제곱법은 넓은 입력 값의 영역을 표현하는 경우 오차가 크게 발생할 수 있다. 따라서 우리는 신경망을 통하여 Monte Carlo 모의실험을 수행하지 않고 입력에서의 충돌 확률을 계산하는 것을 제안한다.

신경망은 그림 2와 같이 입력 층(input layer)과 은닉 층(hidden layer)과 출력 층(output layer) 이렇게 세 개의 층으로 구성되어 있는데 은닉 층(hidden layer)의 층의 수와 노드(node)의 개수는 필요에 의해서 변형이 가능하다. 각각의 층 사이에는 가중치가 존재하며 입력에서 출력으로 가는 동안 이전 층의 값에 가중치를 곱해 주어 다음 층의 값을 결정하는 방식으로 각 층의 값을 갱신한다.

$$z_i = f_i \left(\sum_{k=1}^K w_{ki} \times x_k \right) \quad (12)$$

$$y = f \left(\sum_{l=0}^D w_l \times z_l \right) \quad (13)$$

x 는 입력 값, z 는 은닉 층의 값, y 는 출력 층의 값을 각각 나타낸다. K 는 입력 층의 차원 개수이고 D 는 은닉 층의 노드(node) 개수는 나타낸다. w 는 각 층과 층 사이의 가중치를 나타낸다. 은닉 층의 값 z 를 구하기 위해서는 입력 값 x 와 입력 층과 은닉 층사이의 가중치 값 w 를 곱한 값들의 합을 전달 함수를 통하여 구한다. 전달 함수로는 sigmoidal, 하이퍼탄젠트 함수 등을 주로 사용한다. 출력 값 y 는 은닉 층을 구하기 위한 방법과 같은 방법을 사용해서 구한다.

$$w_h^{k+1} = w_h^k + \Delta w_h^k = w_h^k + \Delta \left(-\frac{\partial \varepsilon_k}{\partial w_h^k} \right) \quad (14)$$

$$w_{ih}^{k+1} = w_{ih}^k + \Delta w_{ih}^k = w_{ih}^k + \Delta \left(-\frac{\partial \varepsilon_k}{\partial w_{ih}^k} \right) \quad (15)$$

학습 단계에서는 학습용 입력과 신경망을 통하여 나온 출력 값 y 와 학습용 데이터의 출력 값 d 를 비교하여 신경망의 출력 값 y 가 학습용 데이터의 출력 값 d 와 같아지도록

```

1: for  $i=1:N$ 
2:    $p(C_i) = MCS(x_i, y_i, V_{x,i}, V_{y,i})$ 
3: end
4:  $network_{train} = train(network_{initial}, [x \ y \ V_x \ V_y], p(C))$ 
5:  $p(C_k) = sim(network_{train}, [x_k \ y_k \ V_{x,k} \ V_{y,k}])$ 

```

알고리즘 1. 전체 알고리즘.

Algorithm 1. Entire algorithm.

록 신경망의 가중치 값 w 를 바꾸어 준다. 이와 같은 방법을 통하여 신경망이 학습용 데이터의 특성을 학습 할 수 있다. 이 실험에서는 하강 기울기 갱신 공식(gradient descent update equation)을 이용하여 가중치를 업데이트 한다.

우리가 사용할 신경망은 다층퍼셉트론 신경망 (multi-layer perceptron neural network)으로 하나의 입력 값이 x , y , v_x , v_y 이렇게 4개의 차원을 가지고 충돌 확률을 하나의 출력으로 가지는 신경망을 사용할 것이다. 학습을 통하여 신경망이 입력과 출력의 특성을 적은 오차로 표현하게 되면 입력에 학습 데이터 외의 다른 값을 넣어도 충돌 확률 값을 구할 수 있다.

따라서 우리는 입력 값의 범위 안에서 일정 간격으로 Monte Carlo 모의실험을 수행하여 견본 값을 알아내고 신경망을 통하여 일정 간격 이하의 범위에서의 충돌 확률 값을 추정함으로써 특정 입력이 들어왔을 때 충돌 확률을 빠르게 계산해 낸다.

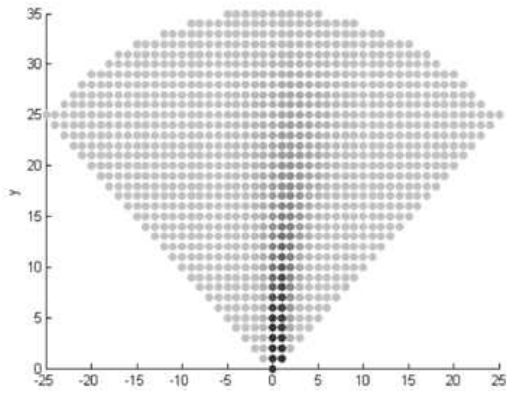
전체 과정을 알고리즘으로 표현하면 알고리즘 1과 같다. 라인 1-3은 위치와 속도를 N 개로 나누고 각각에 대해서 Monte Carlo Simulation을 통하여 충돌 확률을 구한다. 라인 4는 라인 1-3에서 입력과 구한 충돌 확률들을 가지고 신경망을 학습시킨다. 그리고 라인 5에서는 라인 4에서 구한 학습된 신경망을 가지고 구하고자 하는 위치와 속도 x_k , y_k , $V_{x,k}$, $V_{y,k}$ 에 대해서 충돌 확률을 구한다.

IV. 실험

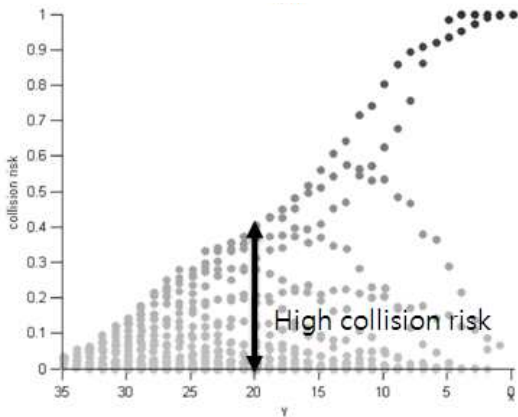
1. Monte Carlo 모의실험

입력 센서로는 레이더 센서를 이용하여 보행자를 인식한다고 가정했다. 레이더 센서의 범위는 그림 3(a)와 같은 $x = [-30 \ 30](m)$ 이고 y 는 전방 90° 이내의 부채꼴 모양으로 $[0 \ 35]$ 의 사이의 값을 갖는다. x 축으로의 속도 $v_x = [-10 \ 10](km/h)$ 를 $2km/h$ 간격으로 y 축으로의 속도 $v_y = [-80 \ 2](km/h)$ 를 $2km/h$ 간격으로 나눠서 각각의 v_x 와 v_y 값에 대하여 실험을 수행하였다. 이는 차량이 기준속도 $80km/h$ 로 앞으로 주행 중인 상황을 가정하고 보행자의 속도는 차량의 최대 속도에 비하면 상대적으로 작기 때문에 $[-80 \ 2]$ 의 범위를 설정 하였다.

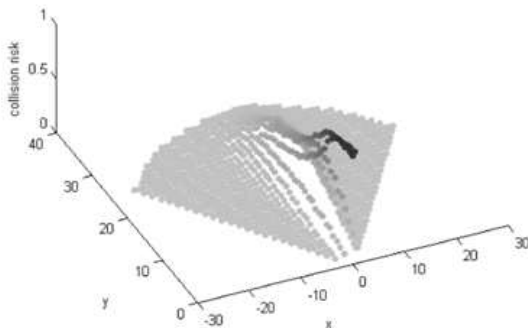
보행자가 센서의 인식 범위 이내로 들어오면 측정 값에 따라서 1000회의 Monte Carlo 모의실험을 수행하여 충돌 확률 값을 계산한다. 그림 3(a)는 $v_x = -6(km/h)$, $v_y = -80(km/h)$ 에서의 충돌 확률 분포를 $1m$ 간격으로 표현한 것이다. 부채꼴 모양은 레이더의 인식 범위를 나타내고, 각 위



(a) Collision probability in x and y axis.



(b) Collision probability in y axis.



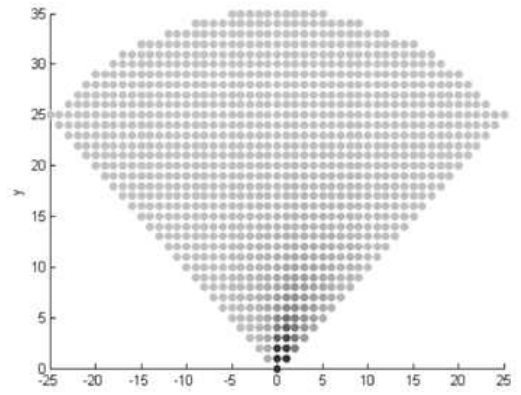
(c) 3D view of collision probability.

그림 3. $v_x = -6$ $v_y = -80$ 에서의 충돌 확률.

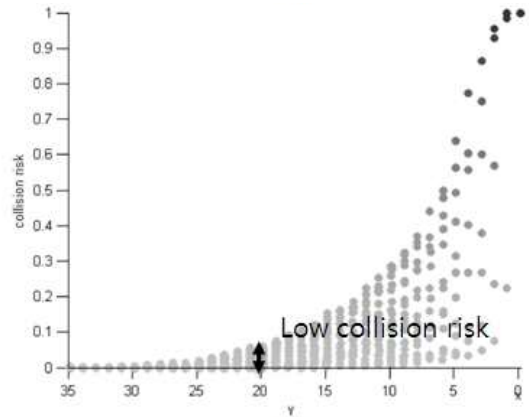
Fig. 3. Collision probability at $v_x = -6$ $v_y = -80$.

치에서의 충돌 확률이 높을수록 짙은 색의 점으로 표현하였다. v 가 $(-6, -80)$ 이기 때문에 전방에서 약간 오른쪽으로 치우친 부분에서 충돌 확률이 매우 높게 나타난다는 것을 확인 할 수 있다. 그림 3(b)는 y 좌표에 따른 충돌 확률 분포를 나타낸 것이다. 그림에서 볼 수 있듯이 y가 0에 가까워질수록 차량과 보행자 사이의 거리가 가까기 때문에 충돌 확률이 높게 나타난다.

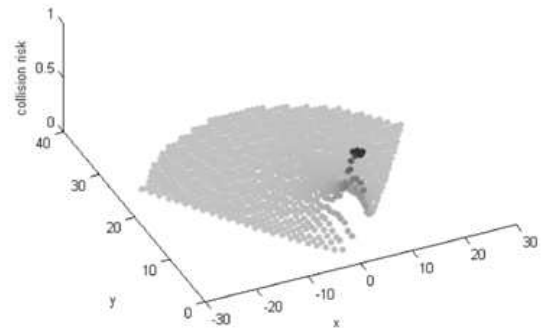
그림 4(a)는 $v_x = -6$ 이고 $v_y = -30$ 일 때의 충돌 확률 분포를 나타낸 것이다. 이를 그림 3(a)와 비교해보면 충돌 확률이 높아져 짙게 표시되는 부분이 y축 방향으로 더 짧게 나타나는 것을 볼 수 있다. 어느 정도의 거리가 유지되고 접근하는 물체의 속도가 낮을 경우 충돌 위험도를 낮게



(a) Collision probability in x and y axis.



(b) Collision probability in y axis.



(c) 3D view of collision probability.

그림 4. $v_x = -6$ $v_y = -30$ 에서의 충돌 확률.

Fig. 4. Collision probability at $v_x = -6$ $v_y = -30$.

판단하는 직관적인 결과와 일치한다. 그림 3(b)와 그림 4(b)를 비교해 보면 속도가 낮은 그림 4(b)가 y축 거리에 따라서 더 급격한 충돌 위험도를 가짐을 확인할 수 있다.

2. 신경망 모의실험

신경망을 이용하여 나온 결과 값의 유효성을 판단하기 위해서 그림 5에서와 같이 5m 간격으로 나온 Monte Carlo 모의실험 결과 데이터를 이용하여 신경망을 학습시키고 학습된 신경망을 이용하여 1m간격으로 충돌 확률을 구하고, 1m간격으로 구한 Monte Carlo 모의실험 결과와 각 점에서 비교해 보았다. 신경망을 통하여 나온 값은 오차가 존재하기 때문에 0 이하의 값이 존재하는 경우가 생기는데 이때는 위험 확률을 0으로 간주하였다.

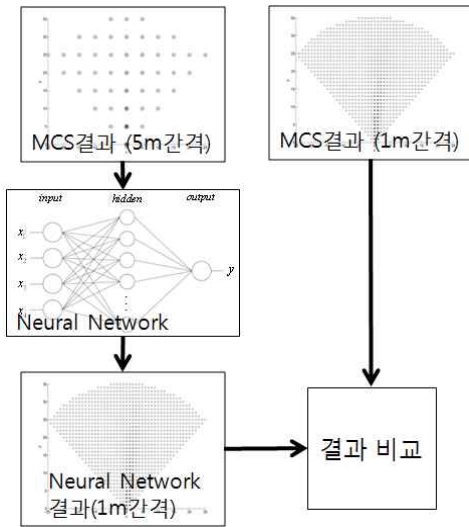


그림 5. Monte carlo와 신경망의 결과비교.

Fig. 5. Compare the output of monte carlo and neural networks.

표 1. 최소 제곱법 결과와 신경망의 결과의 오차 비교 표 ($v_x = -10, v_y = -6$).

Table 1. Comparative table of result and error of least square method and neural network ($v_x = -10, v_y = -6$).

x	y	MCS_out	LS_out	LS_error	NN_out	NN_error
0	0	0.98	0.4952	0.4848	0.9759	0.0041
-8	8	0.045	0.1915	0.1465	0	0.045
0		0.061	0.2014	0.1404	0.3473	0.2863
8		0.081	0.2077	0.1267	0.2099	0.1289
-16	16	0.011	0.0064	0.0046	0.0438	0.0328
8		0.024	0.0186	0.0054	0.079	0.055
0		0.028	0.0273	0.0007	0.0374	0.0094
8		0.044	0.0326	0.0114	0.0542	0.0102
16		0.037	0.0343	0.0027	0.0022	0.0348
-24	24	0.007	0	0.007	0	0.007
-16		0.013	0	0.013	0	0.013
8		0.02	0	0.02	0.0457	0.0257
0		0.023	0	0.023	0.0284	0.0054
8		0.02	0	0.02	0.0262	0.0062
16		0.029	0	0.029	0.0486	0.0196
24		0.033	0	0.033	0.0216	0.0114
-8	32	0.018	0.0322	0.0142	0	0.018
0		0.008	0.0387	0.0307	0.0168	0.0088
8		0.02	0.0417	0.0217	0.0213	0.0013
avg				0.05973		0.03805

표 1은 $v_x = -10, v_y = -6$ 일 때의 Monte Carlo 모의실험의 결과에 따른 최소 제곱법과 신경망의 오차를 정리한 것이다. 표 1에서 최소 제곱법과 신경망의 평균 오차 값을 비교해 보면 최소 제곱법의 오차 값보다 신경망의 오차 값이 약 64%로 줄어들었음을 볼 수 있다. 이때 모든 데이터를 정리할 수가 없어서 학습의 사용한 x, y 가 5의 배수가 되는 데이터가 아닌 8의 배수의 데이터를 정리하여 학습하지 않은 데이터에서도 충돌 확률을 잘 나타낸다는 것을 보였다. $v_x = -10, v_y = -6$ 일 때의 충돌 확률의 평균 오차는

0.024였다. 이러한 오차는 학습용 데이터의 간격을 줄임으로서 해결할 수 있겠다.

V. 결론

우리는 특정한 위치와 속력에 따른 충돌 위험도를 Monte Carlo 모의실험으로 구할 수 있음을 보았다. 하지만 Monte Carlo 모의실험 방법은 실시간으로 시스템을 구현하는 경우 계산량에 따른 필요 시간이 크다는 단점이 있다.

따라서 Monte Carlo 모의실험의 결과를 학습용 데이터로 사용하여 신경망을 학습시키고, 학습된 신경망을 이용하면 Monte Carlo 모의실험을 하는 것보다 더욱 빠른 시간에 충돌 위험도를 구할 수 있고, 이것의 오차가 매우 작음을 볼 수 있었다. 따라서 제안된 알고리즘을 사용하면 빠른 시간 안에 센서로 인식 가능한 영역에서의 충돌 위험도를 구할 수 있음을 확인할 수 있었다.

참고문헌

- [1] 서수민, “자동차, 보행자 안전 기술 개발 부산.” 한겨레 신문, 8. 11. 2006.
- [2] 이상원, “유럽연합, 보행자 보호위해 BAS 의무장착 추진,” 자동차 신문 AutoDaily (www.AutoDaily.co.kr), 10. 29. 2007.
- [3] H. Lee, J. Lee, and E. Kim, “Optimal classifier ensemble design for vehicle detection using GAVaPS,” *Journal of Institute of Control, Robotics and Systems*, vol. 16, no. 1, pp. 96-100, Jan. 2010.
- [4] P. Pascal, “Stereovision based vehicle classification using support vector machines,” *MIT master diploma paper*, 2004.
- [5] Z. SUn, G. Bebis, and R. Miller, “On-road vehicle detection using Gavor Filters and support vector machines,” *Proc. of IEEE DSP 2002*, vol. 2, pp. 1019-1022, 2002.
- [6] A. Petrovskaya and S. Thrun “Model based vehicle detection and tracking for autonomous urban driving,” *Auton Robot*, vol. 26, no. 2, pp. 123-139, 2009.
- [7] A. M. Johansson and E. A. Lehmann, “Evolutionary optimization of dynamics models in sequential monte carlo target tracking,” *IEEE Transactions on evolutionary computation*, vol. 13, no. 4, Aug. 2009.
- [8] M. Althoff, O. Stursberg, and M. Buss, “Model-based probabilistic collision detection in autonomous driving,” *IEEE Transactions on Intelligent Transportation System*, vol. 10, no. 2, Jun. 2009.
- [9] S. K. Park, J. P. Hwang, E. T. Kim, H. J. Lee, and H. G. Jung, “A neural network approach to target classification for active safety system using microwave radar,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 2340-2346, 2010.
- [10] J. P. Hwang, S. Park, E. Kim, and H. J. Kang “Camera and LIDAR combined system for on-road vehicle

detection,” *Journal of Institute of Control, Robotics and Systems*, vol. 15, no. 4, pp. 390-395, 2009.

[11] G. D. Nicloao, A. Ferrara, and L. Giacomini, “Onboard

sensor-based collision risk assessment to improve pedestrians’ Safety,” *IEEE Transaction on Vehicular Technology*, vol. 56, no. 5, pp. 2405-2413, Sep. 2007.



김 범 성

2009년 연세대학교 전기전자공학부 학사. 2009년~현재 연세대학교 전기전자공학부 석박 통합 과정. 관심분야는 Intelligent Vehicle, Computational Intelligence.



박 성 근

2004년 연세대학교 전기전자공학과 학사. 2004~현재 연세대학교 전기전자공학과 석박 통합 과정. 관심분야는 지능형차량, state estimation.



최 배 훈

2010년 연세대학교 전기전자공학부 학사. 2010년~현재 연세대학교 전기전자공학부 석박 통합 과정. 관심분야는 Intelligent Vehicle, Computational Intelligence.



김 은 태

1992년 연세대학교 전자공학과(공학사). 1994년 연세대학교 전자공학과(공학석사). 1999년 연세대학교 전자공학과(공학박사). 1999년~2002년 국립한경대학교 제어계측공학과 조교수. 2002년~현재 연세대학교 전기전자공학부 부교수. 2003년 University of Alberta, visiting researcher. 2008년 UC Berkeley visiting researcher. 2003년 대한 전자공학회 해동상 수상. 관심분야는 Computational Intelligence, machine learning.



이 희 진

1987년 연세대학교 전자공학과 졸업. 1989년 연세대학교 전자공학과 석사 졸업. 1989년~1993년 (주) 대우통신 종합연구소 연구원. 1993년~1995년 (주) 에스원 종합연구소 연구원. 1998년 연세대학교 전자공학과 박사 졸업. 1998년~현재 국립한경대학교 정보제어공학과 교수. 관심분야는 로보틱스, 강인제어, 지능제어, 전력제어 시스템.



강 형 진

1992년 연세대학교 전자공학과 졸업. 1994년 연세대학교 전자공학과 석사 졸업. 1999년 연세대학교 박사 졸업. 1999년~현재 (주)만도 수석연구원. 또한 2005년~현재 경북대학교 겸임교수. 관심분야는 automotive electronics, control system design and signal processing for intelligent vehicle.