

뉴럴네트워크를 이용한 무인 전방차량 추적방법

이 희 만[†]

요 약

본 논문은 전방차량 추적에 있어서 스테레오 카메라 페럴렐 모델을 사용하여 전방차량과의 거리 및 헤딩앵글 데이터를 추출하고 이들 데이터를 이용하여 무인자동차 BART(Binocular Autonomous Research Team vehicle)를 제어하는 방법에 관한 것이다. 무인자동차의 제어는 2개의 역전달 뉴럴네트워크의 일종인 TDNN(Time Delay Neural Network)을 각각 독립적으로 사용하였다. 그중 하나는 S-TDNN으로 추적차량의 속도와 전방차량과의 거리를 제어하며, 다른 하나는 A-TDNN으로 무인차량의 스티어링 앵글을 전담 제어한다. 인간 운전자가 전방차량을 추적하면서 수집한 데이터를 이용하여 상기 뉴럴네트워크를 학습시키며, 학습된 뉴럴네트워크는 인간이 운전하였을 때와 같은 조건하에서 전방차량의 추적을 만족스럽게 수행하였다. 뉴럴네트워크를 이용한 제어프로그램은 이식성이 높아 다른 종류의 차량에도 쉽게 적용할 수 있어 타모델에 적용시에 개발경비와 소요시간을 줄일 수 있는 장점이 있다.

Autonomous Vehicle Tracking Using Two TDNN Neural Networks

Heeman Lee[†]

ABSTRACT

In this paper, the parallel model for stereo camera is employed to find the heading angle and the distance between a leading vehicle and the following vehicle, BART(Binocular Autonomous Research Team vehicle). Two TDNNs(Time Delay Neural Network) such as S-TDNN and A-TDNN are introduced to control BART. S-TDNN controls the speed of the following vehicle while A-TDNN controls the steering angle of BART. A human drives BART to collect data which are used for training the said neural networks. The trained networks performed the vehicle tracking function satisfactorily under the same driving conditions performed by the human driver. The neural network approach has good portability which decreases costs and saves development time for the different types of vehicles.

1. 서 론

현재 세계적으로 지능형 자동차 개발분야에 관한 연구가 활기를 띠고 있다. 차세대 자동차는 크게 네 분야, 즉 환경보존 관련기술, 에너지절약 관련기술, 안전관련기술, 그리고 사회기반시스템과의 결합관련

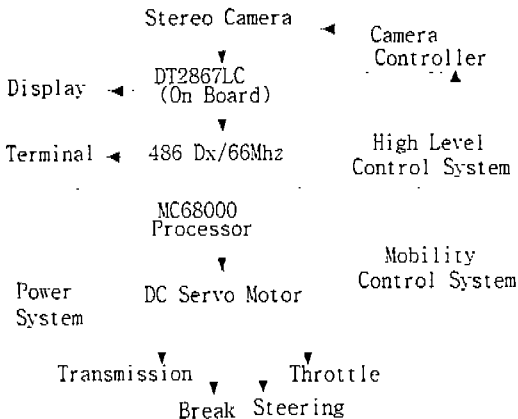
기술로 그 중 안전관련기술에 있어서 지능형 자동차(무인자동차 Autonomous Land Vehicle, ALV)를 개발하기 위한 많은 노력들이 집중되어 왔다[1]. 무인자동차 시스템의 궁극적인 목표는 인간의 조정 없이 장애가 있는 어떠한 공간에서라도 운전가능 하도록 하는 것이다. 이러한 무인자동차 시스템의 자동운전을 통한 항해(navigation)와 안내(guidance)를 이루기 위해서는 무엇보다도 우선하여 고성능 제어 알고리즘의 필요가 절실히 요구된다.

[†] 정 희 원:서원대학교 전자계산학과 전임강사
논문접수:1995년 9월 23일, 심사완료:1996년 1월 12일

본 논문은 전방차량(Leading Vehicle)의 무인 자동 추적방법에 관한 것으로, 보다 자세히는 전방차량을 스테레오 카메라를 이용하여 거리 및 위치를 계산하고 자차를 학습된 뉴럴네트워크를 이용하여 제어하는 방법에 관한 것이다. 기존의 제어방식은 자기 어느 특정 차량의 동특성에 맞게 프로그램되어 이중 차량인 경우 그 해당 차량에 맞게 재 프로그램하여야 하는 문제가 있었다. 본 논문은 이식성(Portability)이 좋은 제어 알고리즘 개발을 위한 것으로 자기 다른 모델의 제어 알고리즘 개발에 있어 개발단계에서 소요되는 시간과 경비를 절감할 수 있는 효과가 있다. 본 연구에 있어서 이식성이 높은 알고리즘 개발의 기본 출발점은 바로 인간이라는 점에 착안하였다. 즉 사람은 자기 다른 동특성을 갖는 차량에 쉽게 적응하기 때문이다. 본 논문에서는 무인차량 컨트롤에 있어 사람이 해당 차량을 운전하고 이에서 얻어진 데이터를 이용하여 뉴럴네트워크를 학습시키며 학습된 뉴럴네트워크를 이용하여 해당 차량을 제어한다.

2. BART 시스템 소개

본 연구를 위한 테스트베드인 BART (Binocular Autonomous Research Team vehicle) 시스템은 (그림 1)에 보였다[2]. BART는 시스템은 미국 Dodge에서 제작한 케러밴 자동차에 비디오 카메라 (Panasonic PK959)



(그림 1) BART시스템 개요
(Fig. 1) BART SYSTEM OVERVIEW

를 2대 부착하고 486/DX66를 탑재하며, 486/DX66 PC에서 운전 명령을 RS232C 통신을 통해 68000 마이크로 프로세서에 주면, 마이크로 프로세서는 서보 구동장치를 통해 트랜스미션과 액셀러레이터, 브레이크를 컨트롤하는 무인자동차 시스템이다. 서보(DC 모터)를 위한 별도의 24V 배터리를 부착하였고 PC전원 공급을 위해 1대의 가솔린 제너레이터(Kawasaki GA1400A A. C. Generator)를 부착하였다. 두대의 비디오 카메라는 스텝모터(NEAT-202 New England Affiliated Technologies Inc.)위에 부착되어 있으며 이들 카메라에서 포착한 이미지는 이미지프로세싱보드(DT2867LC, Data Translation Inc.)에서 캡처 및 프로세싱을 하여 상기 퍼스널 컴퓨터에 데이터를 준다.

3. 스테레오 카메라모델

카메라는 3차원 공간을 2차원 평면상으로 매핑하는 기구로서 한차원의 정보를 잃게 된다. 그러므로 사진촬영에 있어 입체감 있는 사진이 되기 위해서는 인간의 심리를 이용 즉, 크기의 대비, 평행선의 원리, 심도의 차이 등을 이용하여, 잃어버린 한차원을 추론할 수 있도록 촬영하는 기법이 널리 이용된다. 그러나 컴퓨터에서는 1대의 카메라만으로는 입체를 알 수가 없으므로 복수개의 카메라를 이용한다.

스테레오 비전에서는 2개의 비디오 카메라에서 각각 이미지를 캡처하고 두개의 화면에서 서로 대응되는 화소를 찾아 2개 화소의 좌표차이(Pixel Shift)를 이용하여 카메라로부터 화소에 해당하는 물체까지의 거리를 계산한다[3]. 이때 카메라의 기하학적 배치에 따라 물체까지의 거리를 계산하는 방법이 각각 다르다. 물론 2대 이상의 카메라를 사용하는 경우도 있다. 현재 많이 사용하고있는 스테레오 카메라의 기하학적 배치는 패러렐 모델과 컨버지 모델이 있다. 패러렐 모델은 다른 모델에 비해 거리계산식이 단순하며 아울러 이미지 평면 상 서로 대응하는 점이 같은 이미지 수평축에 위치하고 있기 때문에 대응점 검색에 있어 컴퓨팅 코스트가 작다는 장점이 있기 때문이다. 그러나 심도분해능(Depth Resolution)이 다른 모델에 비해 다소 떨어지는 단점도 있다[4].

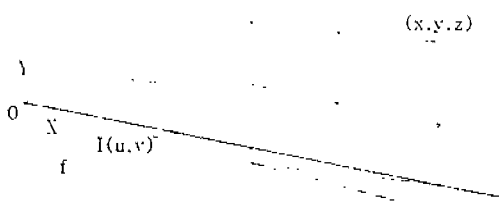
스테레오 카메라모델을 소개함에 앞서 투영변환(Perspective Transform)에 대해서 먼저 기술하기로

한다. 투영변환은 스테레오 카메라의 모델을 기술하는데 이용되기 때문이다.

3.1 투영변환

투영변환(Perspective Transform)은 비선형 변환으로 (그림 2)에 보인 것과 같이 3차원 공간 월드좌표 (x, y, z) 를 원점에서 f 떨어진 2차원 평면 (u, v) 에 투영하는 경우를 생각해보자. 좌표는 좌수식(Left Handed Convention)을 이용하고, 이미지 평면을 원점보다 전방에 위치한 이유는 도립영상을 피하기 위함이다. 그러면 삼각비례연산에 의해 월드좌표 (x, y, z) 와 평면상의 대응 좌표 (u, v) 간의 관계가 다음과 같음을 쉽게 구할수 있다.

$$(u, v) = \left[\frac{fx}{z}, \frac{fy}{z} \right] \tag{3.1}$$



(그림 2) 투영변환
(Fig. 2) Perspective Transform

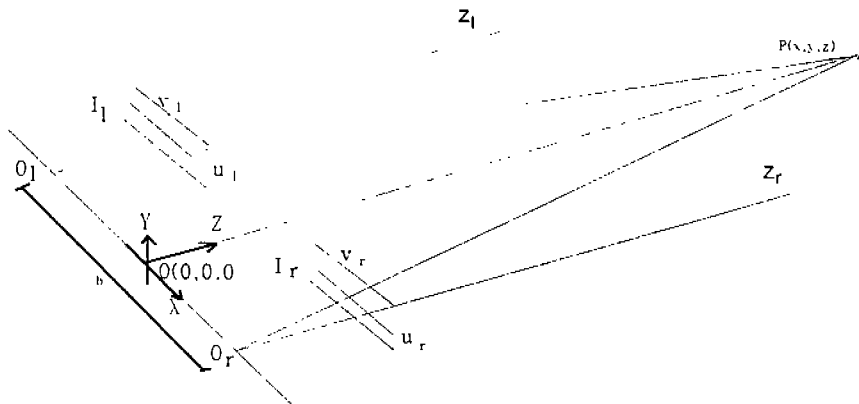
즉 한 물체의 월드좌표를 알면 이미지 평면상에서의 위치를 알수있으나 그 역과정은 성립되지 않는다. 한 차원의 정보가 없어졌기 때문이다. 카메라를 핀홀(Pin-hole)모델로 간주할 경우 위의 투영변환은 스테레오 카메라 모델을 분석에 적용할 수 있다.

3.2 패러렐 모델

2대의 카메라배치를 (그림 3)와 같이 배치해 보자. 즉 원점 $O(0, 0, 0)$ 에서 등거리 지점 $O_l(b/2, 0, 0)$ 과 $O_r(-b/2, 0, 0)$ 에 투영좌표의 기준점을 위치한다. 즉 원점 $O(0, 0, 0)$ 에서 등거리에 위치한 지점 $O_l(b/2, 0, 0)$ 과 $O_r(-b/2, 0, 0)$ 에 광축의 기준을 설정하고 광축 O_lZ_l 과 O_rZ_r 이 서로 평행이 되도록하고, O_l 과 O_r 에서 각각 Z 축으로 f 만큼 떨어진 위치에 광축과 수직이 되도록 이미지 평면 I_l 과 I_r 을 위치해 보자. 그러면 임의의 월드좌표 (x, y, z) 에 위치한 물체는 투영변환에 의해 이미지 평면상의 좌표 (u_l, v_l) 과 (u_r, v_r) 과의 관계는 다음과 같다.

$$\begin{bmatrix} u_l \\ v_l \end{bmatrix} = \begin{bmatrix} \frac{f(x+b/2)}{z} \\ \frac{fy}{z} \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} u_r \\ v_r \end{bmatrix} = \begin{bmatrix} \frac{f(x-b/2)}{z} \\ \frac{fy}{z} \end{bmatrix} \tag{3.3}$$



(그림 3) 패러렐 카메라모델
(Fig. 3) Parallel Camera Model

수평 디스페리티 $D_h = u_l - u_r$, 수직 디스페리티 $D_v = v_l - v_r$ 로 각각 정의하면 디스페리티는 다음과 같이 된다.

$$\begin{bmatrix} D_h \\ D_v \end{bmatrix} = \begin{bmatrix} fb \\ z \\ 0 \end{bmatrix} \quad (3.4)$$

즉 수직 디스페리티는 항상 제로가되어 두 이미지 평면상의 대응점 찾을 때 대응점이 같은 수직축에 위치하여 검색공간이 일차원이 되므로 수직축은 고려하지 않아도 되므로 대응점 찾기가 용이하다. 상기 D_h 로부터 월드좌표를 아래와 같이 복구할 수 있다.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{b}{2} \frac{u_l + u_r}{D_h} \\ \frac{b}{2} \frac{v_l + v_r}{D_h} \\ \frac{bf}{D_h} \end{bmatrix} \quad (3.5)$$

3.3 컨버지모델

컨버지 모델은 (그림 4)와 같이 광축 $\overline{O_r Z_r}$ 과 $\overline{O_l Z_l}$ 이 패러렐 모델에서처럼 평행하지 않고 서로 일정각도 θ 로 대향하고 월드좌표 $P(x, y, z)$ 에서 교차하는 경우이다. 즉 월드좌표 원점 $O(0, 0, 0)$ 에서 일정한 거리 $P1(-b/2, 0, 0)$ 과 $P2(b/2, 0, 0)$ 에 각각 이미지 평면의 원점 $I_l(0, 0, 0)$ 과 $I_r(0, 0, 0)$ 를 설정하고 광축 $\overline{O_r P_r}$ 과 $\overline{O_l P_l}$ 에 수직이 되도록 이미지 평면을 배치하고 이미지

평면에서 f 만큼 떨어진 위치에 투영변환의 기준점을 설정한다. 원점과 기준점의 위치는 계산의 편의성에 따라 설정한 것이지 반듯이 그 위치에 고정될 이유는 없다. 좌표는 패러렐 모델에서와 같이 좌수좌표계를 사용한다. 위의 기하학적 배치로부터 이미지 평면상의 좌표 (u_l, v_l) 와 (u_r, v_r) 과의 관계는 투영변환에 의해 다음과 같이 된다.

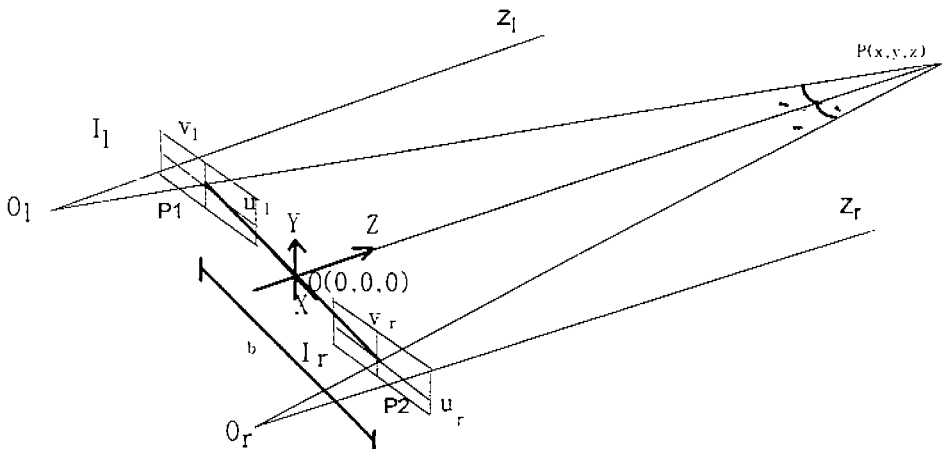
$$\begin{bmatrix} u_l \\ v_l \end{bmatrix} = \begin{bmatrix} \frac{f(x \cos \theta - z \sin \theta + (b/2) \cos \theta)}{x \sin \theta + z \cos \theta + (b/2) \sin \theta + f} \\ \frac{fy}{x \sin \theta + z \cos \theta + (b/2) \sin \theta + f} \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} u_r \\ v_r \end{bmatrix} = \begin{bmatrix} \frac{f(x \cos \theta + z \sin \theta - (b/2) \cos \theta)}{-x \sin \theta + z \cos \theta + (b/2) \sin \theta + f} \\ \frac{fy}{-x \sin \theta + z \cos \theta + (b/2) \sin \theta + f} \end{bmatrix} \quad (3.7)$$

패러렐 모델에서와 같이 디스페리티 D_h 와 D_v 를 정의하고 구하면 다음과 같다.

$$\begin{bmatrix} D_h \\ D_v \end{bmatrix} = \begin{bmatrix} \frac{-2f[x^2 \sin \theta \cos \theta + A(z \sin \theta - (b/2) \cos \theta)]}{A^2 - x^2 \sin^2 \theta} \\ \frac{-2xfy \sin \theta}{A^2 - x^2 \sin^2 \theta} \end{bmatrix} \quad (3.8)$$

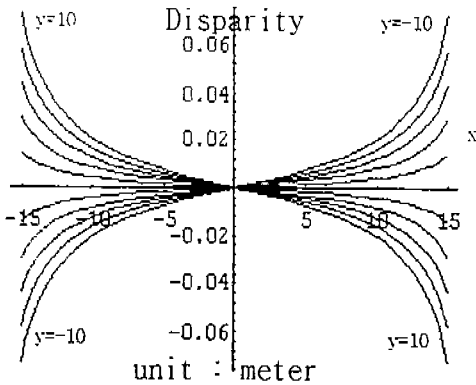
$A = z \cos \theta + (b/2) \sin \theta + f$



(그림 4) 컨버지 카메라모델
(Fig. 4) Converge Camera Model

페러렐 모델과는 달리 D_v 는 항상 제로가 되지 않는다. 즉 서로 대응점이 이미지 평면상의 같은 수평축에 위치하지 않으므로 대응점을 찾는 데 있어 컴퓨팅 코스트가 높게 된다. (그림 5)는 수직축 디스페리티가 이미지 평면상에서 어떻게 변하는가에 대해

다음과 같은 파라메터(Bart 시스템에서 이 모델을 적용한 경우)를 사용하여 도시하였다. 즉, $\theta=30$, $b=838.2\text{mm}$, $f=11\text{mm}$, $z=10\text{m}$, $-10\text{ meter} \leq y \leq 10\text{ meter}$, $-15\text{ meter} \leq x \leq 15\text{ meter}$. 본 모델의 장점은 심도분해능(Depth Resolution)이 페러렐 모델보다 우수하고, 우리 인간의 눈은 이 모델에 보다 더 가깝다.



(그림 5) 수직축 디스페리티 (Fig. 5) Vertical Disparities

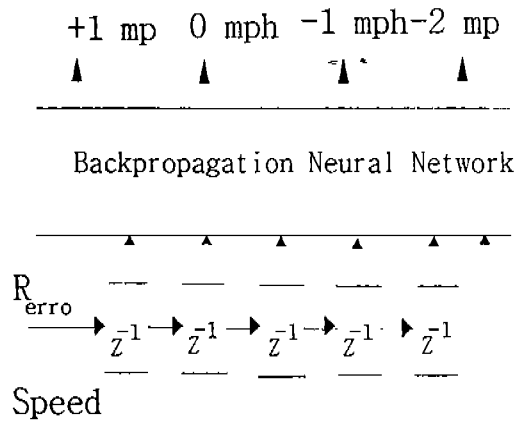
스테레오 비전 응용에 있어 중요한 사항은 카메라를 이용하여 거리를 측정하여야 함으로, 사용전에 반드시 캘리브레이션을 하여야 한다. 즉 기지의 타겟 위치를 모델에 의한 거리와 일치하는가를 확인하여야 한다. Bart 무인자동차에서는 페러렐 모델을 사용하므로 카메라의 광축이 서로 수평이 되도록 하는 것이 매우 중요하다. 이를 위해 카메라의 베이스라인과 타겟이 서로 수직선상에 위치하도록 하고, 타겟의 위치를 실측하여 컴퓨터에 입력하면, 카메라를 통해 계산한 위치와 기지의 값이 일치하지 않는 경우 양쪽 카메라의 팬각도를 자동 조절하여 광축이 서로 수평이 되도록 조절한다.

4. 시간지연 뉴럴네트워크(TDNN)

차량운전은 과거의 데이터, 예를 들면 속도변화, 스티어링 앵글 변화, 액셀러레이터 및 브레이크의 감압 변화 등에 영향을 받는다. 즉 과거의 데이터의 영향을 받아 현상황에 맞게 기구들을 조작 하여야 한다. 시간지연 뉴럴네트워크(Time Delay Neural Network)는 일종의 역전파 뉴럴네트워크로 입력데이터에 시간 지연된 데이터 $[x(t), x(t-1), x(t-1), x(t-2), x(t-m)]$ 를 사용하며 주로 음성인식에 이용되어왔다[5, 6]. 본 연구에서는 시간지연 뉴럴네트워크 (이하 TDNN이라 칭함)를 이용하여 시간지연된 전방차량과 추적차량과의 거리 및 헤딩앵글데이터를 입력으로 사용함으로써 무인차량제어에 사용되었다. 본 연구의 기본 개념은 역전파 뉴럴네트워크는 비선형 입출력 매핑 기능을 갖으므로 입출력간의 매핑관계를 교차학습시켜 제어에 이용하는 것이다. 뉴럴네트워크의 구조는 이론적으로 임의의 입출력 노드를 사용하여도 무방하나 성능과 효율의 최적화를 위해 이들 노드의 수는 실험적으로 구하였다. BART컨트롤을 위해 2개의 독립된 TDNN을 사용 하였다. 즉 스피드 컨트롤을 위한 S-TDNN과 스티어링 컨트롤을 위한 A-TDNN이라 명명된 뉴럴네트워크를 이용하였다.

4.1 S-TDNN

입력 데이터는 (그림 6)에 보인 바와 같이 현재의 스피드 데이터와 현재와 과거의 거리오차를 입력의 데이터로 사용한다. S-TDNN은 BART의 스피드 컨



(그림 6) S-TDNN의 구성 (Fig. 6) Structure of the S-TDNN

트를 전담하는 뉴럴네트워크로, 전방차량과 일정한 거리(Following Distance)를 유지하기 위해 거리가 좁혀지면 속도를 줄이거나 정지 시키며, 거리가 너무 벌어지면 속도를 증가시키는 역할을 담당한다. S-TDNN의 출력 값은 현재 차량속도의 가감속만을 지시하며 이의 지시에 따라 486/DX66는 현재 차속도에서 원하는 차속을 계산하고 이를 68000 마이크로 프로세서에 RS-232C를 통해 지령을 보내면 마이크로프로세서는 액셀러레이터와 브레이크용 서보모터를 제어하여 차량의 속도를 제어한다. 속도 제어를 위한 거리오차는 아래와 같이 계산한다.

$$R_{error} = \frac{\text{Current Distance-Following Distance}}{S.F.}$$

$$S.F. = \begin{cases} 1. \text{ Max. Distance} - \text{Following Distance} \\ \text{if current distance} \geq \text{following Distance} \\ 2. \text{ Following Distance} - \text{Min. Distance} \\ \text{Otherwise} \end{cases}$$

(4.1)

Max. Distance는 최대 허용 추적가능거리이며, Min. Distance는 충돌회피를 위한 최소한 필요한 거리이다. 스피드 입력 데이터는 다음과 같이 정규화 시킨다.

$$S_v = \frac{\text{Current Speed}}{\text{Max. Speed}}$$

(4.2)

Max. Speed는 최대허용 속도이다. S-TDNN의 출력 값은 속도의 변화량 만을 나타낸다. 즉 출력 값은 -2mph, -1mph, 0, +1mph로 고정 되었다. 출력값이 정수된 이유는 BART시스템(68000 마이크로프로세서)이 속도 지령을 정수형태로 받고 있기 때문이다. 감속을 위해 -2mph를 출력하는 이유는 전방차량이 급정차 한경우 충돌을 회피 하기 위함이다. 한편 BART는 0.5초 보다 빠르게 가감속을 할수가 없으므로 0.5초 보다 빠른 속도변화 요구 커맨드는 시스템을 불안정하게 므로 주의를 하여야 한다.

4.2 A-TDNN

A-TDNN은 (그림 7)에 보인 바와 같이 정규화 된 복수개의 헤딩앵글과 현재의 거리를 입력으로 하여 전방차량을 추적하기 위한 자차의 스티어링휠을 컨트롤한다. 즉 과거의 헤딩앵글의 변화와 현재의 거리

정보에 따라 스티어링휠의 각도를 증가/감소 시킨다. 현재의거리 R_d 는 다음과 같이 정규화한다.

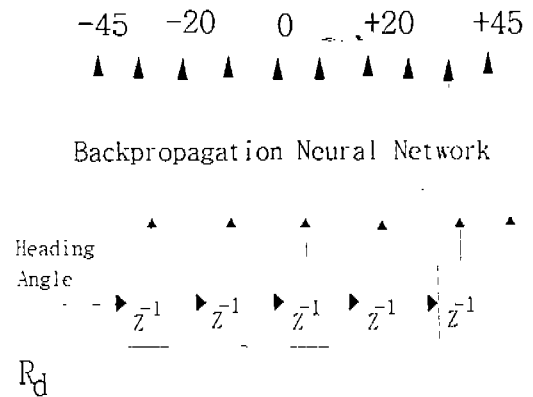
$$R_d = \frac{\text{Current Distance}}{\text{Max. Distance}}$$

(4.3)

A-TDNN은 15개의 출력 노드를 갖고며 그중 오직 한개 노드만이 주어진 입력데이터에 액티베이션 되도록 학습시킨며, 해당노드와 출력값과의 관계는 매핑 테이블을 이용한다. 출력노드는 각각 -45, -29, -20, -15, -10, -6, -3, 0, 3, 6, 10, 15, 20, 29, 45으로 이진(binary) 출력값을 갖는다. A-TDNN이 각도의 증감만 지시하므로 486/DX66 컴퓨터는 현재 각도로부터 원하는 각도를 계산하고 68000 마이크로 프로세서에 지령을 보내면 마이크로 프로세서는 스티어링휠 제어용 서보모터를 구동한다.

5. 실험결과

무인차량의 제어에 사용된 S-TDNN과 A-TDNN은 역전달 네트워크의 일종으로 교사학습이 필요하므로 이들 네트워크를 학습시키기 위해서는 먼저 학습데이터를 수집하여야 한다. 이를 위해 사람이 무인자동차인 BART를 운전하고 이에서 얻어진 거리정보와 전방차량과의 헤딩앵글 및 스티어링 각도 데이터를 추출하여 해당 네트워크를 학습시켰다. 학습된 네트워크를 이용하여 무인자동차를 제어할 때에는 데



(그림 7) A-TDNN의 구성
(Fig. 7) Structure of the A-TDNN

이터를 추출할 때와 같은 상황에서만 적용 가능하며 이 경우 학습시킨 사람의 운전습관이 무인차량의 제어에 적용된다.

전방차량과의 거리 및 헤딩앵글을 계산함에 있어 패러렐 스테레오 카메라 모델을 사용하였다. 두대의 비디오 카메라를 차량의 전방부분 운전석과 조수석 상단에 각각 광축이 평행이 되도록 설치하였으며, 베이스라인의 길이를 $b=838.2mm$ 로 설정하고 비디오 카메라의 초점거리를 수동모드 $f=11mm$ 로 고정하여 초점거리가 자동조절 되지 않도록 하였다. 초점거리가 변하면 거리계산이 달라지기 때문이다. 카메라를 이용하여 거리를 측정할 때 패러렐 스테레오 카메라 모델을 이용하므로 이 모델의 기본조건을 만족시켜야 정확한 거리측정이 가능하다. 패러렐 모델은 두대 카메라의 광축이 서로 평행이 되어야 한다. 두 광축의 평행을 확인하기 위해서 기지의 거리에 특정물체를 위치시키고 이를 카메라로 포착하여 카메라 모델에 따라 물체까지의 거리를 계산한다. 계산한 값과 기지의 값과의 비교에 의해 두 카메라의 광축을 평행하게 조절한다.

비디오 카메라로 포착한 영상을 512×512 프레임으로 DT2867LC보드를 사용하여 캡처하였다. 추적 대상이 되는 전방차량 후단부에 밝은 광원(light source)을 부착하여, 2대의 카메라에서 이를 포착하고 이미지 평면상에서 각각의 좌표를 구한후, 좌표의 편차로부터 거리를 계산하였다. 광원을 이용한 스테레오비전(Active Stereo Vision)을 사용한 이유는 대응점을 찾는 프로세싱 시간을 단축시킬 수 있고 또한 대응점 찾기가 비교적 용이하다는 장점이 있기 때문이다. 물론 단점도 있는데 예를들면, 강렬한 햇빛에 의한 주변 물체로 부터의 반사는 그릇된 거리정보 및 헤딩앵글정보를 얻게되어 무인차량을 원하는 전방차량이 아닌 그릇된 타겟으로 유인하는 원인이 되며, 또한 순간적 난반사 빛을 추적 할 경우 난반사가 사라지면 표적물을 잃고 방향하게 되는 경우가 있다. 이에 대한 해결방법으로, 물론 완벽하지는 않지만, 이미지 평면상에 윈도우를 설정하여 윈도우 범위내에서만 타겟을 찾는 것이다. 물론 윈도우의 크기를 시계(Field-of-View)가 일정하게 되도록 거리에 따라 다이나믹하게 조절하는 것이 더 효과적이다. 왜냐하면 전방차량이 근접한 경우에 설정한 윈도우는 전방차량이 멀리

떨어져 있는 경우에는 이 윈도우 크기가 지나치게 크게되어 윈도우를 사용하는 의미를 상실하기 때문이다. 윈도우를 설정함으로써 얻어지는 또 하나의 잇점은 프로세싱의 속도를 향상시킬수 있었다는 것이다.

<표 1>은 두개의 네트워크에 대한 각각의 파라미터를 정리한 것이다. TDNN의 설계에 있어 중요한 것은 각 레이어에서의 노드수이다. 입력노드수는 그 시스템의 시간 의존도 및 계산속도에 따라 가변되어야 하며 중간층 노드의 갯수는 Recall 계산시간 및 시스템에서 사용가능한 MSE(Mean Square Error)값 등에 따라, 출력노드는 시스템의 제어정도에 따라 달라짐으로 BART 시스템에 맞는 네트워크의 기하학적 구조는 실험적으로 구했다. S-TDNN은 7-21-4구조이며 A-TDNN은 4-12-15구조로 설계되었으며 최대반복루프 6000회수이며, MSE값이 0.1 이하로 내려가면 학습을 중단하게 된다. 학습율 (learning rate)은 다음과 같은 수식에 의해 루프(iteration)에 따라 가변한다.

$$\begin{cases} \eta = \eta_0 (0.5)^{Iteration/N} \\ N \text{ is some large Number} \\ \eta \geq \eta_{min} \quad \text{if } \eta \geq \eta_{min}. \end{cases} \quad (5.1)$$

여기서 N은 임의의 값으로 본 실험에서는 3000을 사용하였으며 최초 학습율 η_0 는 0.2, 최소 학습율 η_{min} 는 0.1을 사용하였다. 두개의 TDNN을 학습하기 위해 필요한 스티어링 각도 및 차속도 데이터를 얻기 위해서는 다음과 같은 방법을 사용하였다. 즉 BART를 사람이 운전핸들과 브레이크 및 액셀러레이터를 사용

<표 1> 뉴럴네트워크 파라미터
<Table 1> Neural Network Parameters

	S-TDNN	A-TDNN
입력노드수	7	4
출력노드수	4	15
중간레이어	1	1
중간노드수	21	12
최초학습율	0.2	0.2
최소학습율	0.1	0.1
요구 MSE값	0.1	0.1
최대반복루프	6000	6000

하지 않고 조이스틱으로 컴퓨터를 통해 제어하게 하였다. 조이스틱의 전후방 방향은 차량의 가속 또는 감속이 되도록 하고 조이스틱의 좌우방향은 스티어링 각도의 변화량을 제어하며 이때 전방차량의 거리와 헤딩앵글 정보와 함께 학습용 데이터로 디스크에 저장한다. 뉴럴네트워크의 노드수를 하드웨어의 제한상 무한정 크게 사용할 수 없으므로 차속의 가감속 및 스티어링의 가감각도는 양자화가 되어야 한다. 그러므로 조이스틱을 사용한 제어에서도 뉴럴네트워크의 출력 노드수에 부합되도록 양자화 시킨다. 즉 전후방향은 1, 0, -1, -2값을 사용하며, 좌우방향은 -45, -29, -20, -15, -10, -6, -3, 0, 3, 6, 10, 15, 20, 29, 45 값을 출력한다. 데이터는 3000개 정도 약 25분 동안 차선변경을 반복하며, 또한 커브길을 주행하며 수집하였다. 디스크 파일에 저장된 데이터는 네트워크의 학습을 위해 전처리 과정을 하며 또한 데이터의 순서를 랜덤하게 배치한다. 학습시간은 486/DX66 PC를 이용 24시간 이상이 소요되었지만 Recall시간은 342ms 정도이다.

<표 2>는 주행제어를 위한 파라미터를 요약한 것이다. 전방차량 추적거리(Following Distance)로 68 feet (20.73m), 최대 허용거리 128 feet (39m), 최소 안전유지거리 38 feet (11.58m), 최대정속 속도 20 mph (30.19 km/h)로 설정 하였다. 뉴럴네트워크를 이용한 BART의 컨트롤은 마치 사람이 운전하는 듯한 반응을 보였

다. 즉 커브길을 주행하는 경우에, 기존의 알고리즘은 전방차량이 커브를 도는 순간 곧 바로 따라도는 경향이 있으나, 뉴럴네트워크를 이용하면 전방차량이 커브를 시작한 지점까지 주행한 후 회전하여 차선을 유지하면서 추적하는 능력을 과시하여 실험대상자들을 놀라게 하였다. 그러나 학습하지 않은 곡물에서는 차선유지 추적이 이루어 지지않았다. 스피드 증가를 1 mph (1.61km/h)로 제한 하였으나, 감속인 경우 -2 mph (3.22km/h)까지 학습 시킴으로서 전방차량이 정지한 경우에도 충돌없이 충분히 정차할 수 있었다. 정지후 재출발시에는 다소 느린 반응을 보였는데 이는 속도 증가가 1 mph (1.61km/h)로 제한되었다는 점과 서보구동에 다소 시간이 필요하기 때문이다.

6. 결 론

스테레오비전을 이용하여 전방차량의 거리 및 헤딩앵글을 측정하고 이 데이터를 뉴럴네트워크에 입력하여 그의 출력값으로 차를 운전하여 전방차량을 무인 추적하는 시스템을 구축하였다. 패러렐 모델 스테레오 비전시스템은 68 feet (20.73m) 거리를 유지하며 추적하는 본 시스템에 적합하였다. 다만 급 커브길에서 전방 차량이 시계(Field of View)에서 벗어나는 문제점이 있다. S-TDNN은 속도를 전담 제어하는 네트워크로 최대속도 20 mph (30.19km/h)로 제한하였는데 이는 안전성의 제고에 기인한다. A-TDNN은 스티어링 휠 조작을 전담하는 뉴럴네트워크로 커브길에서도 차선을 유지하며 전방차량을 추적할 수 있었다. 스티어링 휠 조작을 최대 45도까지만 증감하도록 하였으며, 아울러 최대 변위도 제한하여 전복사고를 미연에 방지하였다. 학습데이터를 수집하기 위해서 조이스틱을 이용하였는데, 만일 스티어링휠에 변위센서를 부착하고 운전행동을 이용하여 데이터를 수집하였다면 좀더 부드러운 학습운행이 가능했을 것으로 생각한다. 뉴럴네트워크를 이용한 시스템 컨트롤 소프트웨어는 시스템의 다이내믹스를 알지 못하여도 각 시스템에 맞게 학습시키면 제어가 가능함으로 다른 환경에서도 쉽게 적용할 수 있어(portable) 소프트웨어 개발기간을 단축시킬수 있는 효과가 있다.

<표 2> 주행제어 파라미터
<Table 2> Driving Control Parameters

항 목	파라미터 값
최소 안전거리	38 feet(11.58m)
상시추적거리	68 feet(20.73m)
최대허용거리	128feet(39m)
최대 스티어링각도	45 degree
최대 윈도우폭크기	150 pixel
최소 윈도우폭크기	6 pixel
최대 윈도우높이	50 pixel
최소 윈도우 높이	5 pixel
상시 정속주행속도	15mph (24.24km/h)
최대 정속속도	20mph (30.19km/h)
속도 가감량	1mph (1.61km/h)

[1] Tomasaburo Yano, "Toward Advanced Motor Society of 21st Century", 통상산업성, 기계정보 산업국, 자동차과, 1989.

[2] Kelly M. Miller, "Portable Neural Net Approach to Vehicle Tracking", M. S. Thesis, Texas A&M Univ., College Station, Texas, 1994.

[3] Dana H. Ballard, Christopher M. Brown, "Computer Vision", Prentice Hall, pp88-93, 1982.

[4] Heeman Lee, "A Neural Network Using Wavelet for Stereo Vision Research", Ph. D Dissertation, Texas A&M Univ., College Station, Texas, 1994.

[5] Robert Hecht Nielsen, "Neuro-computing", Addison Wesley, pp124-137, 1990.

[6] A. Waibel, T. Hanazawa, G. Hinton, et al., "Phoneme recognition using time-delay neural network", IEEE Transactions on Acoustics, Speech and Signal Processing, 37(3), 328-339, 1989.



이 희 만

1984년 고려대학교 전자공학과 졸업(학사)
 1986년 한국과학기술원 전기 및 전자공학과 졸업(석사)
 1994년 미국 Texas A&M Univ. Electrical Eng. Dept. (Ph.D)

1986년~1990년 산업연구원(KIET) 연구원
 1994년~1996년 삼성중공업 중앙연구소 선임연구원
 1996년~현재 서원대학교 전자계산학과 전임강사
 관심분야: 멀티미디어, 음성합성/인식, 영상처리, 코딩 이론