

셀룰라 병렬처리 회로망에 의한 동적계획법 설계와 자율주행 자동차를 위한 도로 윤곽 검출

論 文

53D-7-1

Cellular Parallel Processing Networks-based Dynamic Programming Design and Fast Road Boundary Detection for Autonomous Vehicle

洪昇完* · 金炯奭**

(Seung-Wan Hong · Hyong-Suk Kim)

Abstract - Analog CPPN-based optimal road boundary detection algorithm for autonomous vehicle is proposed. The CPPN is a massively connected analog parallel array processor. In the paper, the dynamic programming which is an efficient algorithm to find the optimal path is implemented with the CPPN algorithm. If the image of road-boundary information is utilized as an inter-cell distance, and goals and start lines are positioned at the top and the bottom of the image, respectively, the optimal path finding algorithm can be exploited for optimal road boundary detection. By virtue of the parallel and analog processing of the CPPN and the optimal solution of the dynamic programming, the proposed road boundary detection algorithm is expected to have very high speed and robust processing if it is implemented into circuits. The proposed road boundary algorithm is described and simulation results are reported.

Key Words : Cellular parallel processing networks(CPPN), Dynamic programming, Optimal path finding

1. 서 론

도로 상에서 차를 자율적으로 주행시키기 위해서는 컴퓨터가 도로 차선과 경계선을 정확하고 고속으로 인식하는 일이 매우 중요하다. 영상처리 기법을 이용해서 도로 차선의 윤곽을 추출할 경우에는 기상상태나 도로상태 및 가로수의 그림자 등을 포함한 각종 장애요인 때문에 도로 에지 윤곽의 절단된 선분들이 각종 잡음 성분에 섞인 영상으로 나타나게 된다. 인간 시각의 종합 판단 능력은 이와 같은 영상으로부터도 도로 선을 잘 찾아낼 수 있으나, 컴퓨터에 의존할 경우에 잘못된 도로 윤곽선이 검출되는 경우가 많다. 지금까지 도로 차선에 대해 많은 연구가 되었으나[10][11][12][13][14], 이들의 공통된 문제들은 곡선도로 구간에서와 같이 비정형 도로 구간에서 경계선 인식의 정확도가 떨어진다는 점[5]과 현재의 영상 뿐아니라 이전의 영상들을 연관시켜 처리해야하므로 처리시간의 과다 [8], 및 그림자, 눈, 비 그리고 다른 물체에 의한 도로 윤곽의 에러에 민감하여 인식의 오류를 유발한다는 문제점들[8]이 있었다.

본 연구는 이런 문제점들을 해결하기 위한 것으로서 도로 윤곽 검출을 셀룰라 병렬처리 회로망(Cellular Parallel Processing Networks, CPPN)의 병렬 계산 방식을 이용하여 고속화 할 뿐만 아니라 동적계획법(Dynamic Programming)의 최적화 개념을 도입함으로써 에지 선소들로 구성된 도로 영상으로부터 최적의 도로 윤곽을 찾는 방법을 제시하였다.

CPPN은 Chua등이 발표한 셀룰라 신경회로망 (Cellular Neural Networks, CNN)[2]의 다른 이름으로서 입력 영상을 아날로그 연산 셀들의 2차원 어레이를 사용하여 병렬 처리하므로 매우 고속의 처리가 가능한 새로운 처리 시스템이다. 이 CPPN에서의 처리는 템플릿(template 혹은 weight template) 값을 달리 인가할 경우 다양한 처리 효과를 얻을 수 있게 된다 [4]. 또한 여러 단계의 복잡한 연산을 위해서 다층구조 [2]를 가질 수 있으며 비선형 기능을 구현하기 위해 비선형 템플릿을 사용할 수도 있다[3]. 한편 동적 계획법(Dynamic Programming, DP)은 Bellman[9]등이 개발한 최적화 기법으로서 지역적인 연산을 수행하는 노드들이 인접 노드들에 연산결과를 제공함으로써 전체적으로 최적화 된 해를 찾는 방법이다. 이 전의 연구에서는 이 동적 계획법을 CPPN에 설치하여 최적의 경로를 고속으로 찾을 수 있는 알고리즘이 발표된 바 있다 [1]. 본 연구는 동적계획법이 설치된 CPPN을 도로 흔적 정보를 종합처리 하는데 이용함으로써 고속으로 최적의 도로경계선을 얻기 위한 알고리즘이다.

본 논문의 2절에서는 셀룰라 병렬처리 회로망(CPPN)에 대해서 설명하였고, 3절에서는 선형, 비선형 템플릿을 이용한 셀룰라 병렬처리 회로망(CPPN)에 대해 설명하였다. 또한 4절에서는 셀룰라 병렬처리 회로망(CPPN) 기반 동적 계획법을 이용한 도로 윤곽 검출에 대해 설명하였고, 5절에서는 제안한 알고리즘에 대한 컴퓨터 모의실험 및 고찰을 제시하였으며, 6절 이에 대한 결론이다.

2. 셀룰라 비선형 회로망(CPPN)

셀룰라 병렬처리 회로망(CPPN)은 CNN(Cellular Neural Networks)이라고도 불리우는 2차원 배열을 갖는 아날로그

* 學生會員 : 全北大 工大 制御計測工學科 碩士卒業

** 正 會 員 : 全北大 工大 電子情報工學部 副教授 · 工博

接受日字 : 2004年 1月 29日

最終完了 : 2004年 1月 29日

병렬처리 회로로서 그림 1(a)와 같은 격자망의 각 노드에는 지역적 연산을 담당하는 아날로그 연산처리 셀들이 배치되어 있다. 이 셀들은 주위의 8개 셀들과의 회로적 연결을 가지며 이 주변 셀들과 긴밀하게 정보 교환을 통하여 처리하는 구조이다. 그림 1(b)는 CNN 셀의 구조 및 주변 셀들과의 연결을 자세히 보이기 위한 1차원 CNN의 구조이다. 여기서, x 는 각 셀의 출력 상태이며 y 는 셀의 출력이고 $f(\cdot)$ 는 셀의 상태 값을 출력으로 변환하는 비선형 출력함수로서 비선형 함수인 시그모이드 함수를 주로 사용한다. 또한 u 는 입력을 표시한다. 그림 1(b)에서 각 연산 셀들의 입력으로는 자신의 위치에 해당하는 화소 값 뿐 아니라 인접 화소 값들을 시냅틱 가중치들과의 컨볼루션에 의하여 입력하는데 시냅틱 가중치 값들에 따라 다양한 종류의 연산이 가능하게 된다. 이 가중치 벡터를 **B** 템플릿이라고 부른다. 또, 회로의 입력에는 인접한 셀들의 출력들도 궤환시켜 입력하는데 이 때 사용된 시냅틱 가중치 벡터를 **A** 템플릿이라고 한다. 그림 1(a)의 2차원 구조에서는 시냅틱 가중치들이 8방향의 연결을 담당해야 하므로 3×3 의 매트릭스로 표현된다. 각 셀에는 또한 제작시의 불 균일을 고려하여 z 로 표현된 바이어스 값을 인가할 수 있게 되어있다. 각 셀의 출력은 입력된 모든 정보들을 합산하여 출력함수 $f(\cdot)$ 에 의해 연산한 결과이다.

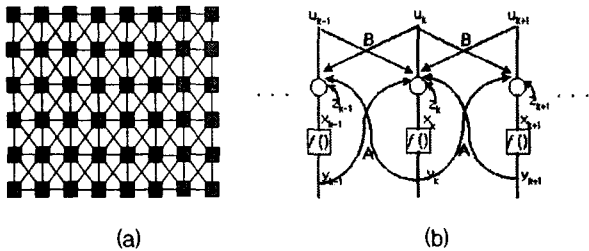


그림 1 CPPN 셀의 배열 (a) 2 차원 연결 (b) 1 차원 CPPN 셀들의 연결
Fig. 1 CPPN Array (a) connections of 2-D CPPN (b) connections of 1-D CPPN

이 연산 구조가 회로에 의해 구현되기 위해서는 입력정보를 전류로 변환하여 커패시터에 합산하는 구조가 효과적이다. 이와 같이 동작을 하는 CPPN 셀의 전기적 동가회로는 그림 2와 같다.

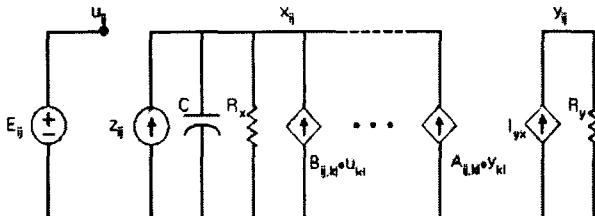


그림 2 CPPN 셀의 전기적 동가회로
Fig. 2 Electrical equivalent circuit of the CPPN cell

이 동가회로에서와 같이 CPPN 셀은 전형적인 RC회로 구조로 표현되므로 템플릿 A 및 B에 의한 입력 값과 내부 bias 값 z 를 고려한 회로의 동 특성식(dynamic)은 식 (1)처럼

1차 미분방정식으로 표현된다.

$$\frac{dx(i,j)}{dt} = -x(i,j) + \sum_{k \in N_r} A(ij,kl) y_{kl} + \sum_{k \in N_r} B(ij,kl) U(k,l) + z_{ij} \quad (1)$$

여기서, $x(i,j)$ 는 셀(i,j)의 출력측 상태를 의미하고 템플릿 $A(ij,kl)$ 와 $B(ij,kl)$ 는 각각 셀(i,j)와 셀(k,l)간의 출력측 및 입력측의 템플릿이며, z_{ij} 는 셀(i,j)의 bias이다. 이 셀의 출력함수는

$$y(i,j) = f(x(i,j)) \quad (2)$$

와 같이 표현되는데 f 는 비선형 함수(nonlinear function)이다.

3. CPPN 비선형 템플릿을 이용한 동적계획법의 구현

3.1 회로구현의 용이성을 위한 동적계획법의 수정

동적계획법(Dynamic Programming, DP)은 각 노드에서의 지역적인 최소값 연산을 통해서 전체적으로 최적의 경로를 계산하는 효율적 계산 알고리즘이다.

만약, $D(i,j)$ 와 $D(k,l)$ 를 각각 노드(i,j)와 노드(k,l)로부터 목적지에 이르는 데 소요되는 최소 거리라고 하자. 또, $d_{ij,kl}$ 를 노드(i,j)로부터 노드(k,l)에 이르는 지역적인 최소 거리라고 가정하면, $D(i,j)$ 는 $D(k,l)$ 와 $d_{ij,kl}$ 를 이용해서 다음과 같이 표현할 수 있다.

$$D(i,j) = \min \{ d_{ij,kl} + D(k,l), (k,l) \in R(i,j) \} \quad (3)$$

여기서, $R(i,j)$ 는 노드(i,j)주변의 모든 노드들의 집합이다. 식 (3)에서의 연산은 min과 summation 연산인데, 일반적으로 min연산은 max연산보다 회로가 복잡하므로, 식 (3)을 max를 사용한 연산으로 변환하는 것이 편리하다. 이를 위해서 기준값을 I_{max} 라 하고 $y(i,j)$ 를 I_{max} 와 $D(i,j)$ 간의 보수(Complement)로 표현하면, $y(i,j)$ 는

$$y(i,j) = I_{max} - D(i,j) \quad (4)$$

혹은

$$D(i,j) = I_{max} - y(i,j) \quad (5)$$

으로 표현된다. 식 (5)를 식 (3)에 대입해서 정리하면,

$$I_{max} - y(i,j) = \min \{ d_{ij,kl} + I_{max} - y(k,l); (k,l) \in R(i,j) \} \quad (6)$$

가 되며, I_{max} 가 상수이므로, 식 (6)은

$$y(i, j) = - \min \{ d_{ij,kl} - y(k, l); (k, l) \in R(i, j) \}$$

$$= - \min \{ -(y(k, l) - d_{ij,kl}); (k, l) \in R(i, j) \} \quad (7)$$

로 정리된다. 식 (7)에서 부의 부호가 min 연산식 밖으로 나오면 min이 max로 바뀌게 되므로,

$$y(i, j) = \max \{ y(k, l) - d_{ij,kl}; (k, l) \in R(i, j) \} \quad (8)$$

와 같이 max연산이 된다. 식 (8)을 이용해서 $y(i, j)$ 가 구해진다면, 목적지까지의 최소 거리 $D(i, j)$ 는 식 (5)를 이용해서 구할 수 있다. 그런데, 목적지에서는 자신의 위치까지의 거리 $D(i, j)$ 가 0이므로, 이 관계를 이용하면, 노드의 위치가 목적지 여부에 따라 다음과 같이 정리할 수 있다.

$$y(i, j) = \begin{cases} I_{max}; (i, j) \text{가 목적지로 지정되는 경우} \\ \max \{ y(k, l) - d_{ij,kl}; (k, l) \in R(i, j); \text{그 이외의 경우} \} \end{cases} \quad (9)$$

여기서, I_{max} 값은 주변의 어느 값에 비해 크다는 점을 고려하여, 식 (9)를 간략화하면,

$$y(i, j) = \max \{ u(i, j), (y(k, l) - d_{ij,kl}, (k, l) \in R(i, j)) \} \quad (10)$$

이고, $u(i, j)$ 는

$$u(i, j) = \begin{cases} I_{max}; (i, j) \text{가 목적지인 경우} \\ 0; \text{그 이외의 경우} \end{cases} \quad (11)$$

로 표현할 수 있다. 이 식에서 확인할 수 있는 바와 같이 식 (1)에서의 min함수가 회로구현이 용이한 max함수로 표현되었음을 알 수 있다.

3.2 다층 CPPN

식 (10)으로 표현된 변형된 동적계획법의 원리는 목적지에 부여한 기준값 I_{max} 가 모든 노드에 전파되는데 이 때, 각 노드에는 인접 노드 출력으로부터 지역적 거리값 만큼을 감한 결과가 입력되게 되며, 각 노드는 이 결과 중에서 최대값을 취하는 연산을 수행한다. 이 전파된 값은 목적지에서 최대가 되고, 목적지에서 멀어질수록 감소하게 된다. 우리는 이 전파되는 값을 거리 기준값(Distance Reference, DR)이라고 하고, 목적지에 인가한 원래의 기준값을 최대 기준값이라고 명명한다. 또한, 이 거리 기준값이 전파되어 0보다 큰 영역을 DR영역이라고 한다. 목적지에 인가한 거리 기준 최대값 I_{max} 값은 가능하면 큰 것이 좋다.

이렇게 DR필드가 형성되어 정상상태(Steady state)에 이르면 DR필드 내에서는 어떤 점에서도 노드의 출력 y 를 이용해서 목적지의 값까지의 최소 거리를 식 (5)에 의해 구할 수 있다. 또한, (i, j) 를 최적 경로상의 한 점이라면, 최적 경로상의 다음 점은 식 (12)를 만족하는 점 (k, l) 로 결정하면

된다. 이와 같은 과정을 반복하여 목적지까지의 최소 거리 경로를 구할 수 있다.

$$y(k, l) = y(i, j) + d_{ij,kl} \quad (12)$$

CPPN에서 많은 셀들을 한 칩에 포함시킬 수 있게 하기 위해서 셀의 구조는 간단할수록 좋을 것이다. 따라서, 본 논문에서 사용한 방법은 식 (10)을 단순한 여러 개의 연산들로 분해하여 각기 다른 층에서 처리하게 하는 것이다. 이를 위해서 식 (10)의 양변에 y_{ij} 를 더하면, 식 (10)은

$$2 y_{ij} = \max \{ y_{ij} + y_{kl} - d_{ij,kl}, (k, l) \in R(i, j) \} \quad (13)$$

혹은

$$y_{ij} = \max \left\{ \frac{y_{ij} + y_{kl} - d_{ij,kl}}{2}, (k, l) \in R(i, j) \right\} \quad (14)$$

가 된다. 식 (14)에서 CPPN의 한 층에서는 max연산만 수행하고 다른 층에서는 괄호 내부의 감산과 2로 나누는 일을 분담해서 처리하게 할 수 있다. 즉, max연산을 수행하는 층을 거리 계산층(Distance Computing layer, DC)이라고 하고, 괄호 내부의 연산을 하는 층을 중간층(Intermediate layer)이라고 하면, 각 층에서는 각각

$$y_{DC}(i, j) = \max \{ y_l(k, l), (k, l) \in R(i, j) \} \quad (15)$$

과

$$y_l(k, l) = \frac{1}{2} (y_{DC}(i, j) + y_{DC}(k, l)) - \frac{d_{ij,kl}}{2} \quad (16)$$

의 연산을 한다. 여기서, $y_{DC}(ij)$ 과 $y_l(ij)$ 는 거리 계산(DC)층과 중간(I)층의 출력이다. DC층의 (i, j) 위치에서 (15)식의 계산을 위해서는 8방향의 (k, l) 에서의 식 (16)과 같은 계산 결과가 필요하며, (i, j) 와 인접한 셀에서도 식 (15)의 계산을 하기 위해서는 식 (16)의 계산이 필요하다. 즉, I층에서 DC층의 (i, j) 와 그 인접 셀 (k, l) 에 대한 계산 결과는 DC층의 (i, j) 위치에서 (15)식 계산에 이용될 뿐 아니라, DC층의 (k, l) 위치에서 (15)식 계산 시에도 I층의 계산 결과가 중복 이용되므로 연산이 효율적이다. 또한, 이 I층의 값은 최종 경로 결정시에도 한번 더 사용되어질 수 있다. 즉, 경로 결정을 위한 조건식 (12(b))는 DC층에서의 y 값이므로 이 값을 (16)식에 대입하면, 경로 결정 조건식은 (17)식으로 절묘하게 단순화된다.

$$y_l(k, l) = y_{DC}(i, j) \quad (17)$$

이 식의 의미는 DC층의 출력과 같은 값을 갖는 I층의 셀이 있다면, 그 방향이 경로의 방향이다. 이와 같이 경로의 여부를 판단하는 일은 경로 결정층(PF layer)라고 하는 또 하나의 층에서 수행한다. 그림 3은 위의 3층의 위치와 연결을 보여준다.

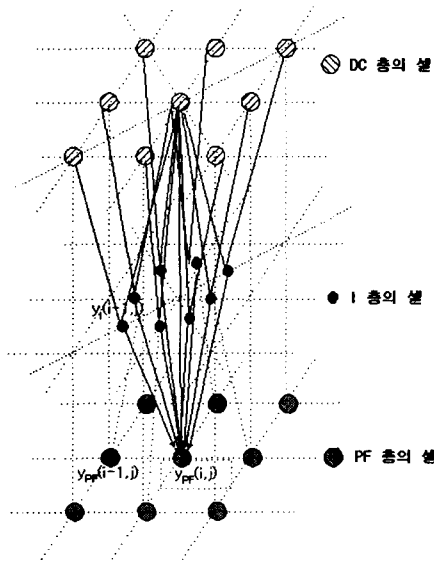


그림 3 제안한 CPPN 구조에서의 셀 연결
 Fig. 3 Connections of a cell in the proposed multi-layer CPPN structure

3.3 최적 경로 계산을 위한 CNN의 비선형 템플릿

최적 경로 계산을 위해서 각 층에서의 연산은 DC층에서 max연산, I층에서 행렬, 나눗셈 및 PF층에서의 비교 연산 등이다. CPPN에서는 이들의 연산 기능을 템플릿을 이용해서 수행해야 하는데, 다행히 max기능은 비선형 템플릿으로써 [1]에서 그 템플릿을 제안했으며, 행렬 및 나눗셈 기능은 CPPN의 선형 템플릿[7]에 의해 해결할 수 있다.

나머지 템플릿은 PF층에서 경로를 표시하기 위해 식 (17)로 표현한 동등 여부 비교기능이다. 기준 거리값들의 분포가 완성되어 있는 한, 식 (17)을 만족하는 셀은 필히 존재하기 마련이며, 그 방향이 경로의 진행 방향이다. 그러나, 식 (17)을 만족했다고 해서 모든 셀의 위치가 경로 상의 셀의 위치가 되는 것은 아니며, 인접 셀이 시작점의 셀 위치이거나 경로 상의 셀이어야 한다는 조건을 만족해야 한다. 즉, 최적 경로는

$$(k, l) \in S(i, j) \text{ and } y_i(k, l) = y_{PF}(i, j) \quad (18)$$

의 조건을 만족하는 셀 (k,l) 로 결정한다. 여기서, $S(i,j)$ 는 (i,j) 셀 주변에 있는 셀들의 집합을 의미한다.

이를 위해서 비선형 템플릿을 사용해야 하며, 이를 위한 CPPN의 비선형 동특성 식은

$$\frac{dx_{PF}(i, j)}{dt} = \sum_{k,l \in N} D_1(y_i(k, l) - y_{DC}(i, j))(y_{PF}(k, l) + 1) + D_2(u_{PF}(i, j) - y_{PF}(i, j)) + y_{PF}(i, j) + z_0 \quad (19)$$

이고, z_0 는 bias값으로써, (0.0, -1.0)의 범위이다. 여기서, D_1 은 조건 $(k,l) \in S(i,j)$ 의 만족 여부를 검사하기 위한 비선형 템플릿이고 D_2 는 $y_i(k,l) = y_{DC}(i,j)$ 여부를 검사하기 위한 비선형

템플릿이다. 이 비선형 템플릿들은 그림 4 및 5와 같이 정의한다.

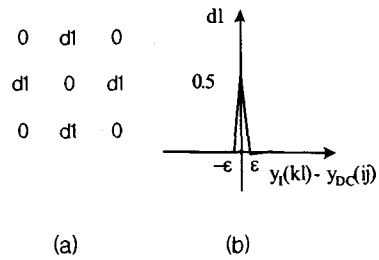


그림 4 경로 결정을 위한 D_1 비선형 템플릿
 (a) 템플릿 (b) 비선형 함수

Fig. 4 Nonlinear template D_1 for the path determination
 (a) template (b) nonlinear function

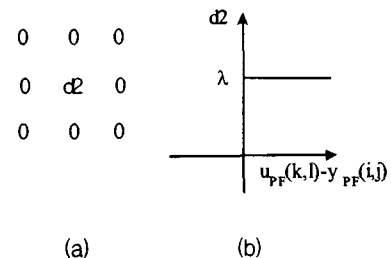


그림 5 경로 결정을 위한 D_2 비선형 템플릿
 (a) 템플릿 (b) 비선형 함수

Fig. 5 Nonlinear template D_2 for the path determination
 (a) template (b) nonlinear function

4. CPPN 기반 동적계획법을 이용한 도로 윤곽 검출

제안한 알고리즘에서 도로 윤곽을 검출하는 원리는 최적화에 효과적인 동적계획법 알고리즘을 병렬고속 처리 구조인 CPPN 상에 구현한 병렬최적화 계산구조를 이용하는 것이다. 인간이 그림 6(a)와 같이 잡음이 많은 도로 에지 영상에서 도로 윤곽을 찾아내려고 하는 경우에는 가능한 다양한 정보를 모두 활용하여 결정을 하기 때문에 오인식이 작은 최적의 결정을 얻게 된다. 제안한 알고리즘도 인간의 눈처럼 도로 윤곽을 찾아내기 위해서 최적화 알고리즘을 사용하는 것을 기본 원리로 한다. 그림 7은 제안한 알고리즘의 순서도를 보여주는데, 먼저 도로 영상으로부터 그림 6(a)처럼 에지를 추출하여, 도로 에지 영상을 제안한 도로 윤곽 검출 알고리즘을 위한 화소간의 거리 값으로 사용한다. 다음 단계는 그림 6(b)과 같이 영상의 위 부분에 긴 목적지 선을 설정하고 아래 부분에는 짧은 2개의 시작선을 설정하기 위해, CPPN의 입력으로 인가한다. 이 영상에서 밝은 화소들은 화소간의 거리 값이 큰 위치를 나타내고 어두운 화소들은 거리 값이 작은 위치를 표현한다. 또, 목적지 선 내부에 있는 점들 상호 간 거리나 시작 점 내의 점들 상호 간의 거리

는 공히 0으로 부여한다. 따라서, 목적지 선에 있는 모든 점들은 하나의 목적지로 작용하며, 시작선에 있는 모든 점들은 하나의 시작 선으로 작용한다. 다음 단계는 높은 거리 기준 값(Distance Reference (DR) Value)을 목적지 선에 인가하는 것이다. 제안한 CPPN에서는 큰 DR 값이 주어지면 이 정보가 전파 확산되어 DR 필드를 만든다. 이 과정에서, 각 셀에서는 식 (10)에서와 같이 입력 값으로부터 거리 값만큼 감해진 값들 중 최대값이 선택되므로, DR 값은 화소간의 거리 값이 작은 도로 윤곽을 따라 큰 값으로 확산하게 된다. 이렇게 생성된 DR 필드가 시작선 근방에 이르면 즉, DR 필드가 목적지 선으로부터 확산되어 시작 선에 도달하면 시작 선에서는 식 (12)의 조건을 만족하는 경로를 따라 목적지 선에 이르기까지 최적의 도로 윤곽선을 생성한다. 이 경우, 두 개의 시작선이 영상 하단의 좌 우측에 설정되어 있으므로 도로의 좌우 경계선을 찾을 수 있게 된다.

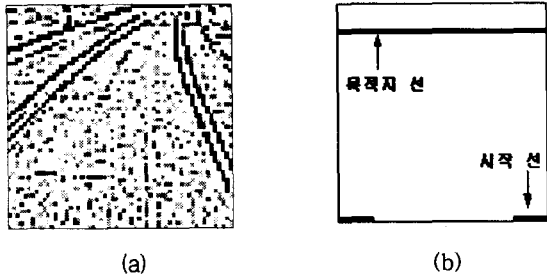


그림 6 (a) 도로 에지 영상의 예 (b) 목적지 선과 시작 선
Fig. 6 (a) An example of the road edge image (b) goal / start lines

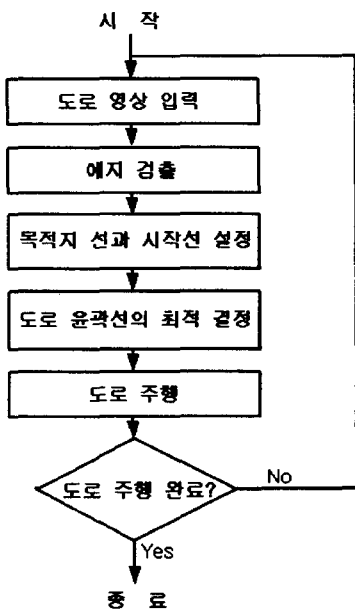


그림 7 제안한 알고리즘 순서도
Fig. 7 Flow diagram of the proposed algorithm

5. 모의실험 및 고찰

도로 윤곽 추출 모의실험

제안한 알고리즘을 포장도로, 그림자가 있는 비포장도로, 눈 덮인 도로 등 여러 상황의 도로 영상에 적용하여 시뮬레이션 하였다. 이 시뮬레이션에서 사용한 영상의 크기는 59x59 였다. 이 시뮬레이션을 위해서 4 절의 알고리즘과 그림 3과 같은 구조의 CPPN을 CNN 시뮬레이터[7]를 이용하여 구현하였다. 셀 간의 거리 값 정보는 도로의 에지 영상을 사용하였으며, I layer에 인가하였다. 또, 목적지 선과 시작 선은 그림 6(b)와 같은 선을 DC layer와 PF layer의 입력으로 인가함으로써 설정하였다.

그림 8(a)는 포장도로의 영상이고 그림 8(b)는 에지 검출 템플릿 [7]을 CPPN에 인가하여 얻은 에지 영상이다. 이 영상에서 보는 바와 같이 도로 영상에는 차선뿐만 아니라 매우 많은 잡음이 함께 존재함을 알 수 있다. CPPN의 거리 가중치로는 그림 8(b)의 에지 영상을 I 층에 인가하고, 그림 6(b)와 같은 목적지 선을 DC 층에 그리고 시작 선을 PF 층에 인가하면 거리 기준 값 (DR source)이 목적지 선으로부터 확산하게 된다. 그림 8(c)는 확산된 DR 필드를 보여준다.

도로 윤곽선은 DR 필드가 그림 6(b)에 있는 시작선에 도달한 후에 생성되기 시작한다. 그림 8(d)에서의 검은 선은 2개의 시작 선으로부터 최적의 도로 윤곽선이 생성된 최종 결과이다. 그림 8(d)에서 회색으로 표현된 선은 시작 선으로부터 찾아진 최적 경로의 일부이지만, 도로 윤곽선이 화면의 가장자리와 만나기 이전의 부분만을 유효한 도로 윤곽선으로 간주한다.

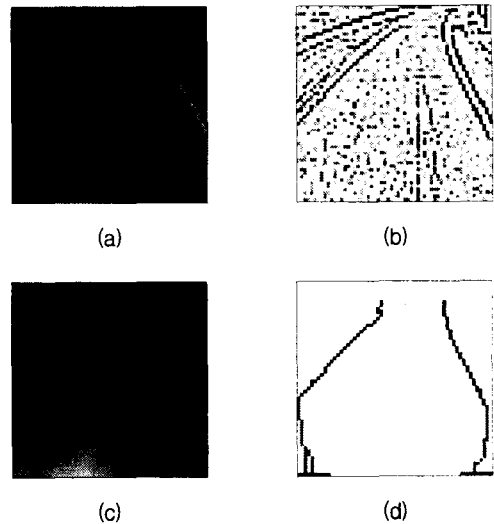


그림 8 포장도로에서의 도로 윤곽 검출 (a) 포장도로 원 영상 (b) 포장도로에서 얻은 에지 영상 (c) 거리 기준 값의 전파 (d) 추출된 도로 윤곽

Fig. 8 Road boundary detection on the paved road image (a) paved road image (b) edge image (c) Propagated Distance Reference(DR) Value (d) determined road boundaries

비포장도로 영상은 도로에 인위적인 차선이 그려져 있지 않기 때문에 도로 윤곽선 검출이 훨씬 어려운 영상이다. 특히 그림 9(a)와 같이 도로 중앙에 타이어 자국이 있고, 가로수의 그림자가 도로에 가로질러 있는 경우는 문제가 더욱 어려워진다. 이 비포장도로 영상에 에지 검출 템플릿[7]을 적용하여 얻은 에지 영상은 그림 9(b)이다. 비포장도로의 에지 영상인 그림 9(b)는 도로의 윤곽선 뿐만 아니라 그림자 윤곽, 타이어 자국 등 많은 잡음 요소들이 존재하므로 도로의 윤곽선만을 검출해 내기가 쉽지 않다. 그림 9(c)는 DR 필드가 목적지 선으로부터 확산하는 과정을 나타내는 그림이고 그림 9(d)는 2개의 시작 선으로부터 최적의 도로 윤곽선이 생성된 최종 결과 영상인데 원하는 도로 윤곽이 잘 검출되었음을 보여준다.

상당히 다른 종류의 영상으로는 그림 10(a)와 같은 눈 덮힌 도로 영상이 있다. 이 영상에 대해 앞에서와 같은 시물레이션을 거쳐서 얻은 에지 영상, 전파된 거리 기준 값 영상, 및 검출된 도로 윤곽선 영상은 각각 그림 10(b), 10(c), 및 10(d)와 같다.

이상의 시물레이션에서 보는 바와 같이, 제안한 알고리즘을 이용하면 최적의 도로 윤곽이 검출될 수 있음을 알 수 있다. 이 결과는 아직 소프트웨어적 시물레이션에 의해 얻은 것이지만 CPPN이 하드웨어적으로 구현되면 고속의 처리 효과도 확보하게 될 것이다.

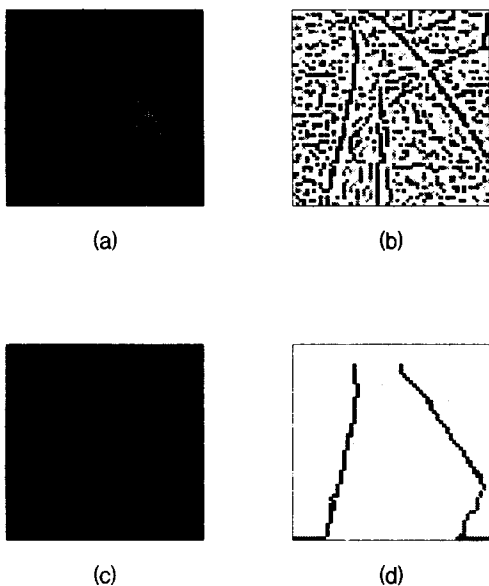


그림 9 비포장도로에서의 도로 윤곽 검출 (a) 비포장도로 원 영상 (b) 비포장도로에서 얻은 에지 영상 (c) 거리 기준 값의 전파 (d) 추출된 도로 윤곽
Fig. 9 Road boundary detection on the unpaved road image (a) unpaved road image (b) edge image (c) Propagated Distance Reference(DR) Value (d) determined road boundaries

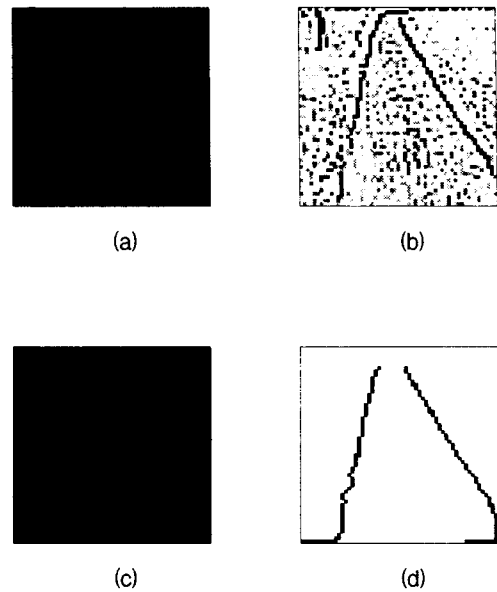


그림 10 눈 덮힌 도로에서의 도로 윤곽 검출 (a) 눈 덮힌 도로의 원 영상 (b) 눈 덮힌 도로에서 얻은 에지 영상 (c) 거리 기준 값의 전파 (d) 추출된 도로 윤곽
Fig. 10 Road boundary detection on the snowy road image (a) snowy road image (b) edge image (c) Propagated Distance Reference(DR) Value (d) determined road boundaries

수행 속도 고찰

본 알고리즘은 아날로그 병렬처리 하드웨어로 구성된 시스템을 위한 병렬처리 알고리즘이다. 따라서, 제안한 시스템이 칩으로 구성되어 처리 됨을 가정하여 그 속도를 비교한다.

각 CNN 셀의 처리 속도는 각 셀의 시정수를 T라고 하는 경우, 약 3T의 시간이면 처리가 완료되는 것으로 간주한다. 그런데, 제안한 시스템에서는 3 층 CNN을 사용하므로 입력이 3 층의 템플릿 연산을 거쳐 출력되는데는 3x3T의 시간이 소요된다. CNN에서의 이와같은 연산 처리는 모든 셀에서 동시에 병렬로 일어나기 때문에 한 프레임의 영상을 처리하는데는 9T 이면 완료되게 된다. 현재 개발된 ACE 64K CNN 칩을 예로 들면, 시정수 T는 7 μsec이다. 따라서, 이들이 3 층의 연산을 거쳐 안정화되려면 9x7μsec=63 μsec의 시간이 소요된다. 이것은 초당 16,000 프레임의 영상을 처리할 수 있는 속도로서 현존하는 어떤 영상처리 시스템에 비해 매우 고속이다.

경로 계산 능력 고찰

제안한 알고리즘은 앞에서 주어진 모든 에지 영상에 대해서 육안으로 판단한 최적의 도로 윤곽선들을 잘 찾아냄을 알 수 있다. 그러나, 도로 윤곽선 상의 화소들 중 적은 수의 화소를 이용하여 최적의 도로 윤곽선을 복원할수록 우수한 성능의 최적 경로 계산 능력이라고 할 수 있다. 따라서 이 최적 경로 계산능력을 비교하기 위해서 다음과 같이 Performance Index (PI)를 정의하였다.

$$PI = 1 - \frac{\text{(에지화소 점 중 도로 윤곽선 상의 화소 점의 수)}}{\text{(총 에지 화소 수)}}$$

만약 에지로 주어진 총 화소의 수는 많지만, 도로 윤곽선 상의 화소 수가 작은 경우에도 최적의 도로 윤곽선을 찾았다면, 우수한 능력의 경로계산 알고리즘으로 평가되어 1에 가까운 값을 갖게 된다. 반대로 총 에지 화소의 수는 적으면서 도로 윤곽선 상의 점이 많다면 낮은 평가 값을 갖게 되며 이 경우에는 작은 PI 값을 갖게 된다.

표1은 앞에서 언급한 3 종류의 영상에 대한 PI 값을 보이는 것으로서 그림자있는 비 포장도로에서의 환경이 열악하며, 이런 환경에서도 최적경로를 찾을 수 있었다는 점에 때문에 가장 큰 평가 값(PI)을 얻고 있음을 보여준다.

표 1. 앞의 영상들에 대한 Performance Index (PI)
Table 1. Performance Index for the previous images

영상종류	포장도로	그림자있는 비 포장도로	눈 덮인 도로
Performance Index	0.46	0.83	0.58

6. 결 론

빠르고 정확한 도로 윤곽 검출은 자동차의 자율 주행 기술에 있어서 핵심적 중요 요구 사항이다. 기존의 도로 윤곽 기법들은 그림자, 눈, 비 그리고 다른 물체 에지 들의 영향 때문에 자주 도로 윤곽선의 오인식이 유발되는 문제가 있었다. 또한, 자율주행자동차에 활용하는 데는 고속처리가 필수적이지만 기존의 순차적 처리 방식의 컴퓨터에 의존하면 속도가 매우 느리다는 문제가 있었다.

본 논문에서 제안한 알고리즘은 동적계획법의 최적화 기법을 사용하여 각종 잡음에 섞여있는 도로 에지 선소들을 종합 활용하므로 최적의 도로 윤곽선을 찾아낼 수 있게 된다. 또한, CPPN의 아날로그 병렬처리 능력을 사용하므로 초고속의 연산이 가능하다.

제안한 알고리즘이 다양한 도로 상황에서 적용될 수 있음을 보이기 위해서, 포장도로, 그림자가 있는 비포장도로, 그리고 눈 덮인 도로 등을 대상으로 도로 윤곽검출 시뮬레이션을 수행 한 결과 정확하게 도로 윤곽선들이 검출될 수 있음을 제시하였다. 본 논문에서 수행한 시뮬레이션은 CPPN의 소프트웨어적 구현에 의한 것이지만, 최근 CPPN의 비선형 템플릿들이 회로로 구현되고 있기 때문에 제안한 CPPN의 하드웨어적 구현이 곧 이뤄질 것으로 예측된다. 따라서, 제안한 방법은 보다 정확하고 고속의 도로 윤곽 검출 수단을 제공할 수 있을 것으로 기대된다.

참 고 문 헌

[1] H. Kim, H. Son, T. Roska, and L. O. Chua, "Optimal path finding with space- and time-variant metric weights with Multi-layer CNN," International

Journal of Circuit Theory and Applications, vol. 30, pp.247-270, 2002.

[2] L. O. Chua and L. Yang, "Cellular neural networks: theory," IEEE Tr. on Circuits Systems, vol.35, pp. 1257-1272, 1988.

[3] S. Espejo, R. Domínguez-Castro, G. Liñán, and A. Rodríguez-Vázquez "A 64x64 CNN universal chip with analog and digital I/O," 5th International Conference on Electronics, Circuits and Systems (ICECS-98), Lisbon, Portugal, pp. 203-206, 1998.

[4] T. Roska and L. O. Chua, "The CNN universal machine: an analogic array computer," IEEE Tr. on Circuits Systems II, CAS-40, pp. 163-173, 1993.

[5] J. D. Crisman and C. E. Thorpe, "SCARF: A color vision system that tracks roads and intersection," IEEE Tr. Robotics and Automation, vol. 9, pp. 49-58, 1993.

[6] K. Soumyanath and V. S. Borkar, "An analog scheme for fixed-point computation-part II: application," IEEE Tr. on Circuits and Systems-I: Fundamental Theory and Applications, vol.46, no. 4, pp. 442-451, 1999.

[7] T. Roska, K. Laszlo, L. Nemes, C. Rekeczky, I. Szatmari, and Zarandy, SimCNN-Multi-layer CNN Simulator for Visual Mouse Platform : Reference Manual version 2.2. Computer and Automation Institute (MTA SzTAKI) of the Hungarian Academy of Sciences, 1998.

[8] J. Tsao, J., R. W. Hall, and S. E. Shladover, "Design options for operating automated highway systems," IEEE-IEE Vehicle Navigation and Information Systems Conference, pp. 494-500, 1993.

[9] Bellman, R., Dynamic Programming, Princeton, NJ: Princeton Univ. Press, 1957.

[10] C. Thorpe and H. Martial, "Vision and Navigation for the Carnegie-Mellon Navlab," IEEE Tr. on Pattern Analysis and Machine Intelligence. Vol. 10, no. 3, May 1988.

[11] E. Mizutani, T. Kozek, and L. O. Chua, "Road lane marker extraction by motion-detector CNNs." IEEE World Congress on Computational Intelligence, vol.1, pp. 503-508, 1998.

[12] D. Kuan and G. Phipps, "Autonomous Robotic Vehicle Road Following," IEEE Tr. on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, September 1988.

[13] K. C. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," Intelligent Vehicles Symposium, pp. 109-114, 1994.

[14] D. Pomerleau and T. Jochem, "Rapidly Adapting Machine vision for Automated Vehicle Steering." IEEE Expert, vol, 11, No. 2, pp. 19-27, 1996.

저 자 소 개



홍 승 완 (洪昇完)

1974년 4월 4일생.
2000년 전북대학교 제어계측공학과 졸업.
2002년 전북대학교 대학원 제어계측공학과 석사 졸업.
2002. 12~현재 서울대학교 의학연구원인구 의학연구소 연구원 (주)젠덱스 연구원
Tel : 02-3673-1992
Fax : 02-3673-1993
E-mail : mirhong@chonbuk.ac.kr



김 형 석 (金炯奭)

1956년 1월 21일생.
1980년 한양대학교 전자공학과 졸업
1982년 전북대학교 대학원 전자공학과 졸업
1992년 University of Missouri, Columbia, Dept. of Electrical and Computer Eng. 박사
1993. 09-현재 전북대학교 부교수
Tel : 063-270-2477
Fax : 063-270-2451
E-mail : hskim@moak.chonbuk.ac.kr