

자기조직화지도를 이용한 직선 추출 알고리즘 A Straight-Line Detecting Algorithm Using a Self-Organizing Map

이 문 규 (Moon-Kyu Lee)

계명대학교 기계·자동차공학부 산업공학전공

Abstract: The standard Hough transform has been dominantly used to detect straight lines in an image. However, massive storage requirement and low precision in estimating line parameters due to the quantization of parameter space are the major drawbacks of the Hough transform technique. In this paper, to overcome the drawbacks, an iterative algorithm based on a self-organizing map is presented. The self-organizing map can be adaptively learned such that image points are clustered by prominent lines. Through the procedure of the algorithm, a set of lines are sequentially detected one at a time. Computational results for synthetically generated images are given. The promise of the algorithm is also demonstrated with its application to two natural images of inserts.

1. 서론

래스터(raster) 영상(image)으로부터 직선을 추출하는 작업은 오래 전부터 연구되어 오던 과제인데, 이를 위하여 가장 대표적으로 사용되는 방법은 허프(Hough) 변환(Hough, 1962)을 이용하는 것이다. 기존의 허프변환은 연속적인 값을 갖는 2개의 변환 모수(parameter)를 일정구간으로 나누어서 이산적인 모수셀(cell)로 설정하여 직선을 추정하고 있다. 이 경우, 하나의 영상상의 점은 다수의 모수 셀로 정합(mapping)되는데 모든 영상점들을 정합하였을 때 정합 빈도가 높은 셀들이 각각의 직선에 대응된다.

이와 같은 방법은 결국 연속적인 값을 갖는 모수들을 이산적인 값들로 변환시켜 직선을 추출하게 되므로 추출된 직선식의 정밀도가 저하된다. 또한, 일정한 정밀도를 만족시키기 위해서는 고려해야 할 모수셀의 수가 너무 많게 되어 엄청난 량의 컴퓨터 기억공간을 요구하게 되기 때문에 기억공간 및 계산상의 부담이 너무 크다는 단점이 지적되고 있다. 따라서, 이러한 문제를 극복하기 위해서는 모수들이 연속적인 값을 갖고 기억공간을 적게 사용하는 효과적인 직선추출방법이 요구된다. 본 연구에서는 이를 위하여 잡음(noise)에 강하고, 연속적인 수치를 다룰 수 있으며 대상 패턴들의 자연스런 특성을 반영할 수 있어 군집화(clustering) 성능이 뛰어난 자기조직화 지도

(self-organizing map: SOM) 기반의 직선추출 기법을 제안한다.

2. 표준 허프 변환 (Standard Hough Transform: SHT)

허프 변환은 영상자료로부터 직선(Yang and Lee, 1997; Bhattacharya et al., 2000)이나 곡선(Ioannou et al., 1999) 등을 추출하기 위한 방법으로 널리 사용되어 왔고 그 동안 수많은 연구가 수행되어 왔다. 이에 대한 기존의 연구는 Illingworth and Kittler(1988) 와 Leavers(1993)에 분야별로 잘 정리되어 있다. 이러한 허프변환은 다양한 분야에 응용될 수 있는데 직접적으로는 기존의 설계도면 영상에서 자동적으로 직선을 추출하여 CAD 도면화하거나(Park and Kim, 1992; Dorl, 1997) 나아가서는 움직이는 물체의 동작을 인식하기 위한 특징추출(Tian and Shah, 1997) 등에 사용되기도 한다. 최근에는 흉부 X-ray 영상으로부터 허프변환을 이용하여 갈비뼈를 추출하는 기법(Orlov et al., 2000)이 발표된 바 있다.

허프변환의 기본 개념은 직선식의 경우 (x, y) 를 영상상의 점이라 할 때

$$y = ax + b \quad (1)$$

로 표현되는 양함수식(explicit equation)을 변환하여 2개의 모수 a, b 대신에 ρ, θ 를 사용하여

$$x \cdot \cos\theta + y \cdot \sin\theta = \rho \quad (2)$$

로 변환하는 것이다. (1)식의 경우에는 모수 a 와 b 의 값이 $(-\infty, \infty)$ 의 무한한 범위를 갖게 되어 수리적으로 처리하기 어려울 때가 많다. 반면에, 허프변환된 (2)식의 경우에는 N_r 를 대상 영상의 장축의 화소수라 할 때, $\theta \in [-\pi/2, \pi/2]$, $\rho \in [-\sqrt{2}N_r/2, \sqrt{2}N_r/2]$ 의 유한한 값을 가지므로 탐색공간을 확실하게 정의할 수 있는 좋은 성질이 있다.

영상상의 하나의 점 (x, y) 는 (ρ, θ) 공간상에 있는 다수의 점에 정합되는데 정합되는 점들은 일반적으로 사인곡선(sinusoidal curve)을 이루게 된다. 특히, (x, y) 공간상에서 직선을 이루는 점들은 모두 (ρ, θ) 공간내의 동일한 점으로 정합되는 성질을 이용하여 영상상의 직선을 추출하게 된다. 즉, 배경을 제외한 영상상의 모든 (x, y) 점들로부터 정합되는 (ρ, θ) 점들을 확인하고 이 중에서 동일한 (ρ, θ) 점에 정합되는 회수가 특정의 임계치 이상이 면 직선으로 간주하게 된다. 다음은 이러한 허프변

한국과학기술원(KAIST) 2002년 5월 3일~4일
 환을 이용하여 직선을 추출하는 절차를 단계적으로 기술한 것이다.

< SHT에 의한 직선추출 절차 >

- 단계 0. 영상으로부터 윤곽선들의 좌표를 다음 집합 D로 놓는다:
 $D = \{P_i=(x_i, y_i), i=1, 2, \dots, N\}$
- 단계 1. ρ 와 θ 의 요구정밀도를 설정하고 이를 만족시키도록 각각의 모수범위를 N_r, N_t 개의 구간으로 분할하여 누적배열 $H(\rho_j, \theta_k), j=1, \dots, N_r, k=1, \dots, N_t$ 를 설정하고 초기치를 모두 0으로 한다. 직선에 해당하는 모수셀을 선정하는데 사용되는 임계치 T를 설정한다.
- 단계 2. D에 속한 점 P_i 를 차례로 선택하여 $x_i \cos\theta + y_i \sin\theta = \rho$ 를 만족하는 모든 (ρ, θ) 셀의 배열값을 1만큼 증가시킨다.
- 단계 3. 배열 $H(\rho, \theta)$ 의 값 중에서 가장 큰 것을 선택하여 그 값이 임계치 E 이상이 되면 해당 셀의 각 모수범위의 중간값을 취하여 선택된 직선의 모수값으로 저장하고, 그렇지 않으면 종료한다. 선택된 직선에 속한 모든 점들을 집합 D에서 제거한 후 단계 2로 간다.

이러한 표준허프변환은 영상상의 점들이 화소(pixel) 단위로 표현되는 이산적인 정수 좌표값을 갖고 있어서 실물상에서는 동일한 직선위에 놓여 있는 점들이라 하더라도 이들에 대한 (ρ, θ) 공간상의 점은 서로 다를 수가 있다. 따라서 정밀도를 만족시키기 위하여 설정된 모수셀의 크기에 따라 추출된 직선식의 정밀도와 정확도가 크게 달라지게 되는 단점이 있다.

따라서, 기억공간의 요구량이 많지 않으면서 정밀도를 유지할 수 있는 방법이 필요한데 이러한 방향으로 최근에 개발된 수정 허프변환(Chutatape and Guo, 1999)은 모수셀로 미리 나누지 않고 한쌍의 점들로부터 직선식을 만족시키는 모수셀에게만 기억공간만을 할당하는 기법을 사용하고 있다. 또한 한가지 방법은 직선으로 추정되는 셀에 대하여 셀을 해상도가 점차 높아지도록 계층적으로 분할하여 탐색하는 것이다. 예로서 일정 임계치 이상의 정합회수를 가지는 단위 공간을 quadtree 형태로 분할해 가는 방법(Li et al., 1986)과 배열의 크기를 매 사이클 마다 적절하게 조절해 나가는 방법(Illingworth and Kittler, 1987) 등을 들 수 있다. 그러나, 영상이 비교적 다양한 물체를 포함하고 있어서 복잡한 경우에는 임계치에 따라 그 결과가 크게 달라지게 되므로 이산적인 영상에 포함된 직선식을 제대로 표현하기 어렵다는 단점이 지적되고 있다. 또한, Chutatape and Guo, Li et al., Illingworth and Kittler의 방법들 모두 직선식이 모수구간으로 표현되는 비연속적인 모수값으로 나타나는 점과 잡영(noise)에 많은 영향을 받는다는 허프변환의 기본적인 단점을 갖고 있다. 이외에도 이러한 단점들을 극복하기 위한 여러 가지 접근방법이 시도된 바 있지만 아직까지 확실하게 극복할 수 있는 방법이 발표되지 않은 것으로 보인다.

한편 이산적인 영상자료의 특성 때문에 동일한 직선에 속하는 점들이라도 그들의 허프변환 모수 값은 다르게 나타난다. 따라서 이들을 어떻게

효과적으로 군집화하는가 하는 문제가 발생한다. 결국, 표준허프변환 방법은 미리 모수 (ρ, θ) 공간을 직사각형의 단위 공간으로 구분하여 군집화하는 단순한 방법을 사용하고 있어서 영상 자료 자체에 내재된 고유의 특성에 따라 군집화되기가 원천적으로 불가능하다. 그런데, 주어진 자료들의 특성에 따라 자연스럽게 군집화할 수 있는 특성을 가진 기법중의 하나가 자기조직화지도 기반의 신경망이다.

3. 자기조직화 지도기반의 직선추출 알고리즘 (SOM-based Line Extracting Algorithm: SOMLEA)

3.1 알고리즘

자기조직화 지도는 Kohonen(1982)에 의하여 제안된 신경망(neural network)의 하나로서 입력공간을 잘 반영하도록 스스로 형성된, 출력공간을 표현하는 지도이다. 즉, 입력패턴의 유사성을 공간상의 관계로 변환된 지도를 의미한다. 이 때 입력패턴의 구조(차원)와 출력패턴의 구조는 서로 상이할 수가 있는데 본 연구의 경우는 영상의 (x, y) 좌표를 1차원 이상의 격자구조의 뉴런(neuron)들로 변환시키게 된다. 즉, 입력공간의 각 단위공간에 $m (\geq 1)$ 대 1로 대응되는 출력공간의 뉴런이 존재하고 그 뉴런들은 각기 해당하는 영상좌표들의 단위공간을 대표한다.

하나의 직선식은 두점으로 정의될 수 있으므로 영상 상에 있는 임의의 두 점 $P_i(x_i, y_i)$ 와 $P_j(x_j, y_j)$ 를 선정하여 각 점에 대한 허프변환 식인 (2)식을 세운 다음 이 두식을 만족하는 (ρ_{ij}, θ_{ij}) 를 구한다. 자기조직화 지도의 입력 공간을 (ρ, θ) 로 하고 이들을 효과적으로 정합하여 주는 출력공간은 단순한 1차원 또는 2차원의 뉴런들로 구성된다. 입력공간의 한점 V_{ij} 가 출력공간의 한점 r 로 정합되는 과정은 입력점 V_{ij} 와 거리가 가장 짧은 연결강도(synaptic strength) W 를 갖는 출력공간의 점을 선택하는 것이다. 이렇게 선정된 출력공간의 점 r 에 대한 W 를 입력점 V_{ij} 의 방향으로 수정하게 된다.

이러한 방법으로 자기조직화 지도를 학습하게 되면 유사한 (ρ_{ij}, θ_{ij}) 들의 군집을 구할 수 있고 그것은 결국 유사한 직선상에 있는 점들의 군집을 의미하게 되는 것이다. 이렇게 해서 추출된 군집들이 원 영상상에 있는 직선들에 속한 점들을 군집한 결과라고 해석할 수 있고 그때의 뉴런들의 연결강도 벡터 W 가 군집된 점들을 대표하는 직선식에 해당되는 것으로 해석할 수 있다. 그러나 이러한 연결강도 벡터들은 대체적인 직선식을 나타내기 때문에 그 군집에 속한 점들에 대한 정확한 직선식을 구하기 위해서는 최소자승법을 이용한다. 즉, 주어진 영상점들을 자연스럽게 그 특징이 살려지도록 군집화하기 위해서 자기조직화 지도를 사용하는 것이다. 또한, 가장 두드러진 군집군들을 찾아내기 위하여 가장 많은 점들을 포함하고 있는 하나의 군집을 찾아낸 후, 그에 해당하는 영상점들을 대상 입력점들로부터 제거한 후 남은 영상점들만을 대상으로 하여 자기조직화 지도를 다시 학습한다. 이 과정을 반복하여 일정 수준의 임계치 이상의 영상점들을 포함하는 군집들이 남아 있지 않을 때까지 새로운 군집들을 반복적으로 찾아낸다. 이상의 자기조직화 지도를 이용한 직선추출 과정을 단계별로

다음에 정리되어 있다.

우선 기술상의 편의를 위하여 사용되는 용어를 정의한다:

- N = 영상상에 있는 점의 개수,
- n = 뉴런 수(이차원의 경우에는 $N_x \cdot N_y$ 로서 N_x 는 X축의 뉴런수, N_y 는 Y축의 뉴런수를 나타냄),
- V_{ij} = 입력벡터 (ρ_{ij}, θ_{ij}), $i, j=1, \dots, N$,
- W_k = 뉴런 k 의 입력층과의 연결강도 = (W_{pk}, W_{0k}), $k=1, \dots, n$,
- ϵ = 학습계수(learning coefficient),
- L_r = 뉴런 r 에 의하여 정의되는 직선에 속한 영상점들의 집합,
- E = 직선으로 정의되기 위하여 최소로 필요한 영상점들의 수,
- t = 진화 사이클 인덱스,
- T = 총 진화 사이클 수.

단계 0. 초기화

출력공간의 구조와 규모(뉴런의 수)를 설정한다. 다음 뉴런 k 에 대한 연결강도벡터 W_k 의 요소값인 W_{pk} 과 W_{0k} 를 각각 $U[-\sqrt{2}N_f/2, -\sqrt{2}N_f/2]$ 와 $U[-\pi/2, \pi/2]$ 의 일양분포(uniform distribution)로부터 임의로 추출한 값으로 초기화한다. 즉,

$$W_k = (W_{pk}, W_{0k}), k=1, \dots, n$$

$$W_{pk} \in [-\sqrt{2}N_f/2, -\sqrt{2}N_f/2], W_{0k} \in [-\pi/2, \pi/2].$$

진화 사이클 $t = 1$.

단계 1. 입력벡터 선택

임의의 두 영상점을 선택하여 다음 식에 의하여 입력벡터 $V_{ij}=(\rho_{ij}, \theta_{ij})$ 를 산출한다:

$$x \cdot \cos\theta_{ij} + y_i \cdot \sin\theta_{ij} = \rho_{ij};$$

$$x_j \cdot \cos\theta_{ij} + y_j \cdot \sin\theta_{ij} = \rho_{ij}$$

의 두식으로부터

$$\theta_{ij} = \arctan((x_j - x_i)/(y_i - y_j));$$

$$\rho_{ij} = ((x_i + x_j)\cos\theta_{ij} + (y_i + y_j)\sin\theta_{ij})/2.$$

단계 2. 반응 뉴런 결정

입력벡터 V_{ij} 와 가장 근접한 뉴런 q 를 결정한다:

$$q = \phi(V_{ij}),$$

$$|W_{\phi(V_{ij})} - V_{ij}| = \min_k \|W_k - V_{ij}\|.$$

단계 3. 연결강도 조정 (Adaptation Step)

뉴런 r 과 근접한 뉴런일수록 수정이 많이 될 수 있도록 연결강도의 값들을 아래 식에 따라 수정한다:

$$W_k^{new} = W_k^{old} + \Delta W_k, k=1, \dots, n,$$

$$\Delta W_k = (V_{ij} - W_k) \cdot \epsilon(t) \cdot h(t)$$

여기서 $\epsilon(t)$ 는 t 가 커질수록 그 값이 작아지는 감소함수로서 학습율을 줄이기 위한 목적으로 사용하고, $h(t)$ 는 r 과의 근접도를 나타내는 함수로서 r 의 출력공간에서의 위치벡터 $r=(X_r, Y_r)$ 을 중심으로 표준편차 σ 를 갖는 정규분포를 따르는 것으로 다음과 같이 가정한다:

$$\epsilon(t) = \epsilon_0 \cdot \exp(-t/T), \epsilon_0 = \text{상수};$$

$$h(t) = \exp(-\|k-r\|^2/2\sigma^2(t)), k = \text{뉴런 } k \text{의 위치벡터 } (X_k, Y_k), \sigma(t) = \sigma_0 \cdot \exp(-t/T), \sigma_0 = \text{상수};$$

$\sigma(t)$ 가 t 가 증가함에 따라 감소하기 때문에 $h(t)$ 는 뉴런 r 의 중심부가 상대적으로 점차 증가하게

된다. 이것은 학습이 진행됨에 따라 대표성이 있는 뉴런에 대한 선택성(selectivity)을 높이는 작용을 한다.

단계 4. 사이클 반복

$t = t+1$ 로 재정의하고 단계1-3을 $t \geq T$ 를 만족할 때까지 반복 수행한다. 반복이 완료되면 단계5로 간다.

단계 5. 군집 선택

뉴런의 연결강도를 직선의 모수로 하여 가장 많은 영상점들을 포함하는 뉴런을 찾는다. 영상점들이 직선에 포함하는지에 대한 여부는 해당직선으로부터의 직선거리를 계산하여 다음과 같이 평가한다:

$$d_{ir} = P_i \text{로부터 뉴런 } r \text{에 대한 직선거리}$$

$$= |x_i \cdot \cos W_{0r} + y_i \cdot \sin W_{0r} - W_{0r}|;$$

만약, $d_{ir} \leq d_0$ 이면 P_i 를 L_r 에 추가한다. $r^* = \text{argmax } |L_r|$ 라 할 때 $|L_{r^*}| \geq E$ 이면 하나의 직선으로 선택하고, 그 L_{r^*} 에 속한 영상점들을 대상으로 최소자승법을 이용하여 구한 직선식의 모수를 저장하고 단계 6으로 간다. 한편, $|L_{r^*}| < E$ 이면 그동안 구해진 직선식들을 출력하고 종료한다.

단계 6. 추출된 직선에 속한 영상점 제거

영상점 집합

$$D = D - L_{r^*}.$$

로 재정의하고 단계 0으로 간다.

위 알고리즘의 구현에 필요한 기억용량은 뉴런 수에 전적으로 비례하여 입출력과 관계되는 용량을 제외하면 $2 \cdot n$ 이 된다.

3.2 예제

3.1절에 정의된 기술된 직선추출알고리즘의 수행과정의 이해를 돕기 위하여 간단한 예제를 풀어보기로 한다. 그림 1-a에 주어진 영상은 512X512 크기의 영역내에 세 개의 직선을 설정한 후 잡음을 첨가하여 임의로 만든 것이다. 직선상에 놓인 영상점 수는 각각 90, 45, 30으로 총 165개이다. 이 영상으로부터 세 개의 직선을 차례로 추출하게 되는데 SOMLEA를 적용하기 위한 조건은 크기가 $n=9$ 인 1차원 구조의 자기조직화지도, $\epsilon_0 = 0.3$, $\sigma_0 = 1$, $d_0 = 4.0$, $T=100,000$, $E=20$ 으로 주어졌다.

단계 0: 총 9개의 뉴런에 대한 연결강도 벡터 W 를 $W_{pk} \in U[-256\sqrt{2}, 256\sqrt{2}]$, $W_{0k} \in U[-\pi/2, \pi/2]$ 의 범위에서 임의적으로 초기화한 결과가 그림 1-b에 주어져 있다.

단계 1-4: 단계 0에서 초기화된 연결강도 벡터 W 를 총 100,000회 학습한 결과가 그림 1-c에 나타나 있다.

단계 5: 그림 1-d는 위에서 학습된 결과를 가지고 뉴런별로 영상점들을 군집화하여 가장 많은 영상점을 가진 것으로 선택된 직선을 도시한 것이다. 마찬가지로 방법으로 단계 0-4를 거쳐 학습하고 군집화하는 절차를 반복하여 두 번째 직선과 세 번째 직선을 추출한 결과가 그림 1-e, 1-f, 1-g, 1-h에 주어져 있다.

그림 1-d, 1-f, 1-h에서 보면 각 직선에 포함된 영상점들을 제거하고 난 후에도 그 직선에 속한

한국과학기술원(KAIST) 2002년 5월 3일~4일

는 것으로 설정된 영상점들의 일부가 남아있는 것을 볼 수가 있는데 이는 판정기준이 되는 거리 임계치인 d_0 값의 크기가 너무 작게 설정되었기 때문으로 해석할 수 있다. 즉, 주어진 영상점들은 표준편차 2화소로 주어진 정규분포에 따라 선정된 직선으로부터 벗어나도록 생성되었으므로 최대 6화소(=3 σ)만큼의 차이가 있다. 그런데 $d_0 = 4.0$ 로 설정하였기 때문에 그 이상 멀리 떨어진 점들은 포함되지 못하였다. 그럼에도 불구하고 SOMLEA 적용결과 추출된 직선들이 주어진 영상점들의 내재된 특성을 잘 표현하고 있는 것으로 보인다.

4. 성능 평가

본 연구에서 개발된 SOMLEA의 성능을 검증하기 위하여 임의로 생성된 직선들로 이루어진 영상과 실제영상을 대상으로 하여 성능평가를 시행한다.

4.1 SHT의 특성

SOMLEA의 상대적 성능을 알아보기 위하여 SHT에 의한 직선추출절차와 비교하여 보기로 한다. 그러기 위해서는 SHT의 특성을 우선 알아볼 필요가 있다. 2절에 기술된 SHT 절차는 직선식의 정밀도를 높이기 위하여 대용량의 기억장치를 요구하므로 실제 적용하는 데 무리가 있다. 그러므로 본 연구에서는 이를 피하기 위하여 Li et al.(1986)의 방법을 인용하여 다음과 같이 수정하여 사용하였다:

초기에 가능한 크기의 모수배열을 설정한 후 SHT 단계 2를 수행하고 단계 3에서는 임계치 이상의 배열값을 갖는 모수셀 중에서 가장 큰 것을 2개의 셀로 이분한다. 이러한 과정을 임계치 이상의 배열값을 갖는 모수셀이 나오지 않을 때까지 반복 수행한다.

일차적으로 이러한 SHT의 특성을 파악하기 위하여 다음과 같이 주어지는 여러 경우에 대하여 예비시험을 실시하였다:

$N_r = 50, 100; N_t = 50, 150, 250; E = 5, 25, 50$
 직선추출의 대상으로 사용된 영상들은 영상에 존재하는 직선들의 수 N_L 과 잡영 수준(noise amplitude) λ 가 다양하게 반영될 수 있도록 하였는데 다음의 각 경우를 고려하였다:

$N_L = 1, 3, 5; \lambda = 0, 3.0, 5.0.$

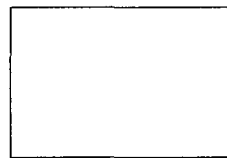
서로 다른 9가지의 (N_L, λ)의 경우에 대하여 각각 20개의 영상(총 180개)을 임의로 생성(각 직선에 속하는 영상점들의 수는 90로 주어짐)하여 위에 정의된 총 18(=2X3X3) 경우의 (N_r, N_t, E)에 대하여 SHT를 적용하였다. λ 는 주어진 직선으로부터 수직으로 벗어난 거리를 의미하는데 여기서는 잡영을 생성할 때 편의상 평균이 0이고 표준편차가 λ 인 정규분포를 사용하였다. 그림 2는 $N_L=5$ 이고 $\lambda=5.0$ 인 경우에 임의로 생성된 직선들과 그것들로부터 만들어진 영상점들이 함께 도시된 영상레이다.

한편, 그동안 문헌에서는 개발된 직선추출 절차의 상대적인 우월성을 평가하는 척도로서 추출된 직선들의 정확도를 판단하는 기준이 모호한 경우가 대부분이었다. 특히, 잡영이 많이 들어 있는 영상의 경우에는 그 동안 정확도의 평가척도로

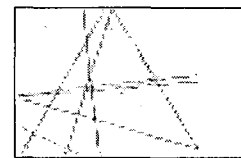
사용해왔던 "모수구간의 폭"이 직선의 정확도를 제대로 반영하지 못하는 경우가 대부분이다. 그 이유는 잡영이 많은 직선은 대부분의 영상점들이 직선상에 있지 않고 직선 주위로 흩어져서 분포하기 때문에 모수 구간의 폭을 너무 줄이면 전혀 엉뚱한 직선을 추출하게 되기 때문이다. 따라서, 본 연구에서는 추출된 직선들의 정확도에 대한 절대적인 척도로서 [0,1]의 범위를 갖는 오차, E_p 를 새로이 정의하여 사용한다. 즉, (ρ, θ)와 (ρ', θ')를 각각 영상생성시 원래 정의된 직선과 추출된 직선의 모수라고 할 때

$$E_p = [E_r^2 + E_t^2]^{1/2}$$

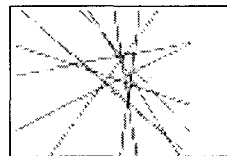
단, $E_r = (\rho' - \rho) / \sqrt{2} N_t; E_t = (\theta' - \theta) / \pi$



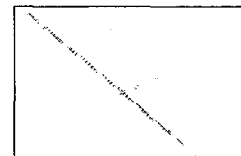
a. 세 개의 직선에 대한 영상점들이 주어진 영상



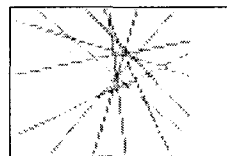
b. 연결강도 초기화 결과



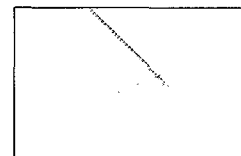
c. 학습이 완료된 상태의 모습



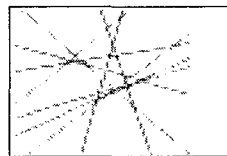
d. 추출된 첫 번째 직선



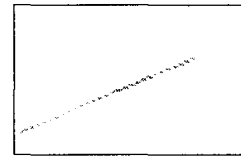
e. 첫번째 직선에 속한 점들을 제거한 후 2차 학습이 완료된 상태의 모습



f. 추출된 두 번째 직선



g. 두번째 직선에 속한 점들을 제거한 후 3차 학습이 완료된 상태의 모습



h. 추출된 세 번째 직선

그림 1. 세 개의 직선이 있는 영상에 대한 SOMLEA에 의한 적용례

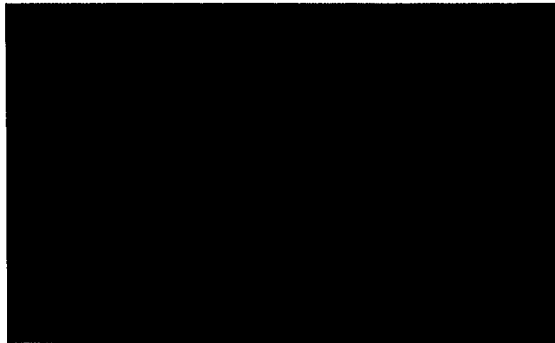


그림 2. 임의로 생성된 직선들의 영상례($N_L=5, \lambda=5.0$)

로 정의한다. 이렇게 정의하면 오차의 크기가 항상 0과 1사이의 값을 갖게 되기 때문에 절대적인 오차의 크기를 이해하기 쉽고 또한 상대 비교에도 효과적이다.

이러한 오차의 평균을 평가척도로 하여 표 1은 상기의 실험결과를 정리한 것인데 (N_r, N_t, E)의 변화에 따른 SHT의 성능변화가 뚜렷이 나타나 있다. 즉, 동일한 N_r 에 대해서는 N_t 와 E 의 값이 커질수록 성능이 좋아지고, 마찬가지로 N_r 이 증가함에 따라 성능도 향상되는 것으로 나타나고 있다. 또한, E 값이 커짐에 따라 해를 구하지 못하는(즉, 주어진 모든 직선들 중에서 추출되지 못한 직선이 있는) 문제의 수가 대폭적으로 늘어나는 것으로 보아 영상에 주어진 직선을 모두 제대로 추출하기가 어려워짐을 나타내고 있는데 이는 잡영이 포함된 영상의 경우 더욱 심해진다. 결국, 앞에서도 기술된 바와 같이 우수한 결과를 얻기 위하여 N_r, N_t, E 값을 적절하게 선택하는 것이 대단히 중요한 것임을 확인할 수 있는데 이것은 SHT가 지니고 있는 특성을 그대로 반영하는 당연한 결과로 보여진다.

표 1. (N_r, N_t, E)의 변화에 따른 SHT의 성능 변화 (수치는 평균오차이며, 괄호 안의 값은 해를 구하지 못한 문제의 수임)

Nr	Nt	E		
		5	25	50
50	50	0.02718(0)	0.01078(18)	0.04650(83)
	150	0.02875(0)	0.01387(11)	0.00359(75)
	250	0.02491(0)	0.02245(6)	0.00431(78)
100	50	0.02233(0)	0.00354(64)	0.00354(140)
	150	0.02649(0)	0.00302(61)	0.00236(115)

4.2 SOMLEA의 특성

본 연구에서 제안하는 SOMLEA는 SHT가 갖고 있는 단점을 극복하고는 있으나 역시 출력공간의 구조와 학습방법, 특히 학습회수에 따라 성능이 달라진다. 출력공간은 1차원일 수도 있고 2차원 또는 그 이상으로 보다 복잡한 구조를 가질 수 있는데 일반적으로 최적의 구조는 SOM의 응용 대상의 특성에 따라 달라진다(Ritter et al., 1992). 여기서는 다양한 경우의 영상에 대하여 이러한 변수들이 알고리즘의 성능에 미치는 영향을 살펴보기로 한다.

이를 위하여, 직선의 수가 비교적 많고 상당한 잡영을 포함하고 있어서 정확한 직선 추출이 어려울 것으로 예상되는 경우($N_L = 5, \lambda = 5.0$)에 대하여 실험을 하였다.

한편, SOMLEA의 학습회수는 주어진 영상점들의 개수나 서로 다른 입력벡터 V_{ij} 의 수에 비례하는데 여기서는 학습효과를 효과적으로 평가하기 위해서 서로 다른 입력벡터의 수를 고려하였다. 즉, 하나의 입력벡터는 2개의 점들로 정의되기 때문에, 총 학습회수 TLI는 서로 다른 입력벡터가 모두 선택될 때까지의 학습회수인 1주기(=NC2)의 곱으로서 다음과 같이 표현된다:

$$TLI = R * NC2 \quad (\text{단, } R = \text{임의의 상수}).$$

표 2는 1차원과 2차원(X축과 Y축의 뉴런 수를 동일한 것으로 가정함: 즉, $N_x=N_y$)의 각 출력공간에 대하여 뉴런 수의 변화와 R값의 변화에 따른 SOMLEA를 적용하여 구한 평균오차를 정리한 것이다. 이 표에서 보면, 우선 뉴런의 개수가 동일할 때, 단순한 1차원 출력공간의 구조가 2차원 구조보다는 월등히 우수하고, 전체 뉴런 개수가 커질수록 성능이 향상됨을 보여준다. 이것으로부터 3차원 이상의 복잡한 출력구조와 비교하여도 유사한 결과를 얻을 것으로 예상된다.

1차원의 출력공간 구조는 뉴런들을 일렬로 늘어놓는 구조이기 때문에 뉴런수가 동일할 경우 2차원보다는 뉴런들의 거리가 훨씬 멀게 된다. 이는 영상점들이 군집화될 때 뉴런들이 가급적 서로 멀리 떨어져 있는 것이 유효하다는 것을 보여주고 있다. 그러한 구조가 유사한 성질을 가진 영상점들을 각 뉴런들에게 서로 배타적으로 군집화하는데 효과적인 것으로 해석된다.

한편, 표 2에서 보는 것처럼 학습회수는 대체적으로 많을수록 좋으나 군집화에 우수한 출력공간구조를 갖고 있고 뉴런 수가 충분한 크기인 경우, 예로서 1차원의 $n=36, 81$ 인 경우 등에는 통계적으로 유의차가 없는 것으로 나타나고 있다. 즉, 그러한 출력공간의 구조를 갖는 SOM에서는 학습을 1주기만 시켜도 충분한 학습효과를 나타낼 수 있는 것으로 판단된다. 그러므로, 전체적으로 보았을 때 출력공간의 구조 선택이 학습회수보다 상대적으로 보다 중요한 결정변수임을 알 수 있다.

일반적으로 다양한 형태의 여러 직선을 포함되어 있는 영상에서는 직선추출의 성능이 떨어지게 된다. 알고리즘의 성능에 영향을 주는 것으로 이러한 영상의 복잡성과 함께 잡영의 수준도 주요요인이 된다. SOMLEA의 적용상 예상되는 성능의 추이를 관찰하기 위하여 영상의 복잡성(즉, 영상에 존재하는 직선의 수)과 잡영의 수준의 변화에 대한 평균오차의 추이를 분석하였다.

표 3은 학습시간이 적게 걸리면서도 우수한 성과를 보이는 1차원의 출력공간, $R = 1, n = 81$ 의 조건하에서 $N_L = 1, 3, 5; \lambda = 0, 3.0, 5.0$ 로 주어지는 각 9가지 경우의 실험결과를 정리한 것이다. 여기서 보면, 잡영이 전혀 없고 영상이 극히 단순하여 1개의 직선만이 존재하는 경우에는 회귀분석(linear regression)의 결과와 동일하여 오차가 0에 가깝다. 그러나, 잡영이 많고 영상의 복잡성이 증가할수록 초기의 예상과 같이 평균오차가 커지는 현상을 보이고 있다. 그럼에도 불구하고, $N_L = 1$ 이고 $\lambda = 0.5$ 의 경우를 제외하고는 평균오차

한국과학기술원(KAIST) 2002년 5월 3일~4일

가 최대 0.00171로서 본래의 직선식과의 오차가 거의 없는 우수한 성능을 보이고 있다.

$N_L = 1$ 이고 $\lambda = 0.5$ 의 경우에는 오차가 상대적으로 크게 그 이유는 잡영이 많은 반면 영상점들의 수($N=90$)는 매우 적어서 학습회수가 적게 되고 그 결과로 학습효과가 잘 나타날 수 없었기 때문이다. 이것은 학습회수를 증가시키기 위하여 $R = 3$ 과 5 로 설정하고 구한 평균오차가 각각 0.00279, 0.00226으로 급격히 감소한 사실로부터 확인할 수 있었다. 그러므로 직선 수가 작고 잡영이 많은 영상의 경우에는 학습회수를 증가시키는 것이 바람직할 것으로 생각된다.

4.3 SHT와의 비교

다음으로 본 연구에서 제안하는 SOMLEA와 SHT의 성능을 서로 비교하여 보았다. 우선 SHT를 적용하기 위해서는 적절한 N_r , N_t , E 값을 결정해야 한다. 이를 위하여 많은 실험을 하였는데 그 결과, 주어진 모든 문제를 풀 수 있으면서 동시에 상대적으로 우수한 성능을 보이는 값으로서 $N_r = 110$, $N_t = 180$, $E = 10$ 이 선정되었다. 표 4에는 이렇게 선정된 조건하에서 (N_L, λ)의 각 9가지 조합에 대하여 임의로 생성된 20개씩의 문제를 풀어서 구한 SHT와 SOMLEA의 성능이 비교되어 있다(SOMLEA의 경우에는 표 3의 자료를 그대로 사용함). 이 표에서 보는 바와 같이 $N_L=1$ 이고 $\lambda=5.0$ 인 경우를 제외하고는 SOMLEA의 성능이 월등히 우수함을 알 수 있다. $N_L=1$ 이고 $\lambda=5.0$ 인 경우에도 앞절 4.2에서 기술되어 있는 것처럼 학습회수를 증가시키면 평균오차가 현저히 감소하여 SHT보다 좋은 결과를 가져오기 때문에 전체적으로 SOMLEA가 우월한 성능을 내는 것으로 판단된다.

또한, 필요한 기억용량은 SOMLEA의 경우 최대 $2*n=162$ 에 불과하나 SHT의 경우에는 $N_r*N_t=19,800$ 으로 현저하게 증가하게 된다.

4.4 SOMLEA의 실제 응용례

마지막으로 SOMLEA의 응용성을 평가하기 위하여 실제 영상에 적용하여 보았다. 평가에 사용된 영상은 Chutatape and Guo(1999)에서 사용한 실제 영상과 유사한 것으로서 선반이나 밀링가공용으로 많이 사용되는 인서트(insert) 영상이다. 평가의 다양성을 위하여 인서트가 한 개인 경우와 보다 복잡한 것으로서 3개인 경우의 두가지 영상을 고려하였다.

그림 3.a의 영상은 인서트가 한 개인 경우이고 3.b는 그 영상으로부터 추출된 이진화된(binazed) 윤곽선(edge)의 모습이 주어졌다. 윤곽선의 추출은 잡영을 제거하기 위한 중위수여과(median filtering), Sobel 필터 처리, 이진화 및 세선화(thinning) 등의 일련의 영상처리 과정을 거쳐서 이루어 졌다. 3.c에 주어진 직선 추출결과는 일차원 구조의 SOM에 $R=1$ 의 조건하에서 SOMLEA를 적용하여 구한 것이다. 그림에서 보듯이 4각형의 모습을 이루는 4개의 긴 직선과 코너에 있는 4개의 짧은 직선들이 모두 완벽하게 추출되었음을 알 수 있다.

또한, 3개의 인서트를 포함하고 있는 보다 복잡한 영상에 적용한 결과가 그림 4에 주어졌다. 이 경우에도 모두 10개의 직선이 제대로 추출되었

다. 이러한 결과들로부터 SOMLEA가 인공적으로 만든 가상의 직선이 아닌 실제 영상에 존재하는 직선들도 효과적으로 추출해내고 있기 때문에 다수의 직선이 포함된 일반 영상에도 우수하게 적용될 수 있는 것으로 평가된다.

표 2. 5개의 직선이 주어진 문제에 대한 SOMLEA에 의하여 구한 평균오차($\lambda=5.0$)

출력공간 R n	1차원			2차원		
	9	36	81	9	36	81
1	0.0951	0.0025	0.0017	0.10929	0.0112	0.01096
3	0.0939	0.0024	0.002	0.14407	0.00771	0.00935
5	0.0876	0.0026	0.0019	0.1363	0.01050	0.00632

표 3. 영상의 복잡성과 잡영 수준에 따른 평균오차의 변화

직선수 λ	λ		
	0	3	5
1	0.00001	0.00027	0.00538
3	0.00011	0.00037	0.00109
5	0.00026	0.00092	0.00171

표 4. SHT와 SOMLEA의 평균오차 비교

N_L λ	λ		
	0	3.0	5.0
1	0.00001*/0.00034@	0.00027/0.00244	0.00538/0.00393
3	0.00017/0.01357	0.00037/0.02586	0.00109/0.02674
5	0.00026/0.02238	0.00092/0.03557	0.00171/0.03675

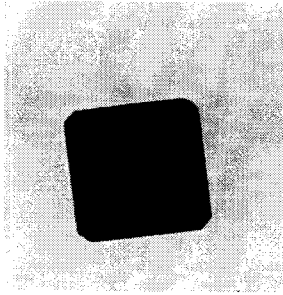
* SOMLEA의 평균오차; @ SHT의 평균오차

5. 결론

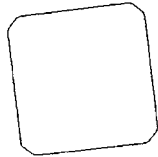
본 연구에서는 인공신경망의 일종인 자기조직화 지도를 이용하여 영상에 내재된 직선을 효과적으로 추출하는 알고리즘을 제안하였다. 제시된 자기조직화 지도는 1차원 또는 2차원 구조의 뉴런들로 이루어져 있는데 각 뉴런은 직선을 정의하는 모수에 대응되는 2개의 연결강도로 표현된다. 따라서, 하나의 직선에 속한 영상점들은 개념적으로 하나의 뉴런에 군집화되게 된다. 이러한 성질을 이용하여 개발된 SOMLEA는 기존의 SHT 기법이 갖고 있는 단점인 대량의 기억용량이 필요하다든지 또는 모수들의 영역을 이산적인 구간으로 분할함으로써 야기되는 직선식의 정밀도 저하 등의 단점을 극복할 수 있는 대안으로 평가된다. SHT와의 비교결과 기억용량이 훨씬 적게 필요할 뿐만 아니라 직선식의 정확도 측면에서도 월등히 우월한 성능을 나타내었고 또한 실제 영상에도 효과적으로 응용될 수 있음을 알 수 있었다. 다만, 현재에는 컴퓨터를 이용하여 모사된 프로그램을 이용하여 구현하였기 때문에 학습시간이 비교적 오래 걸리는 단점이 있으나 병렬처리가 가능한 신경망 회로를 사용하게 되면 시간상의 제약은 그다지 문제가 되지 않을 것이 다.

대한산업공학회/한국경영과학회 2002 춘계공동학술대회
 한국과학기술원(KAIST) 2002년 5월 3일~4일

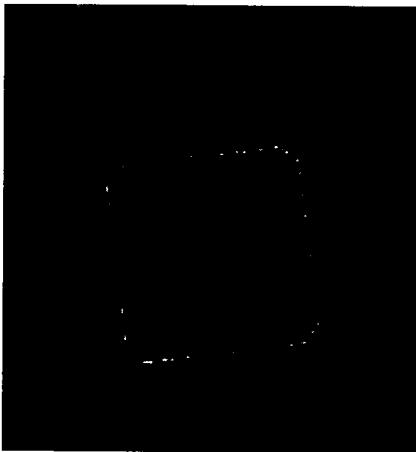
본 연구에서는 여러 개의 직선이 겹쳐져 있거나 연결되어 있는 경우는 고려하지 않았으나 이는 대상영역을 점차 확대하는 방법을 사용하므로써 쉽게 해결될 수 있을 것으로 보인다. 또한, 이러한 SOMLEA는 직선뿐만 아니라 보다 복잡한 여러 가지 도형들, 예로서, 원, 타원, 곡선 등의 2차원 도형과 나아가서는 사면체 등의 3차원 물체 등도 추출할 수 있도록 쉽게 확장될 수 있을 것으로 보인다.



a. 사각 인서트 영상

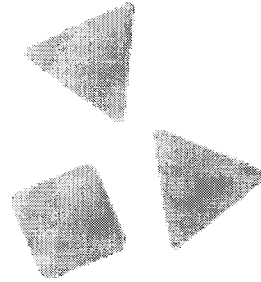


b. 영상 a에서 윤곽선을 추출한 모습

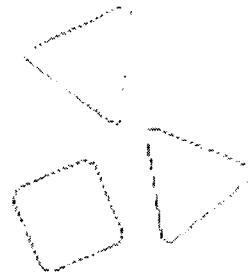


c. 영상 b에서 추출된 직선을 도시한 모습

그림 3. 한 개의 인서트가 있는 영상에 대한 직선 추출 결과



a. 2개의 삼각 인서트와 한 개의 사각 인서트가 있는 영상



b. 영상 a에서 윤곽선을 추출한 모습



c. 영상 b에서 추출된 직선을 도시한 모습

그림 4. 세 개의 인서트가 있는 영상에 대한 직선 추출 결과

대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일~4일
참고문헌

- Bhattacharya, P., Liu, H., Rosenfield, A., and Thompson, S. (2000), Hough-Transform Detection of Lines in 3-D Space, *Pattern Recognition Letters*, **21**, 843-849.
- Chutatape, O. and Guo, L. (1999), A Modified Hough Transform for Line Detection and its Performance, *Pattern Recognition*, **32**, 181-192.
- Dori, D. (1997), Orthogonal Zig-Zag: an Algorithm for Vectorizing Engineering Drawings Compared with Hough Transform, *Advances in Engineering Software*, **28**, 11-24.
- Hough, P. V. C. (1962), Method and Means for Recognizing Complex Pattern, U.S. Patent No. 3069654.
- Illingworth, J. and Kittler, J. (1987), The Adaptive Hough Transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**(5), 690-698.
- Illingworth, J. and Kittler, J. (1988), A Survey of the Hough Transform, *Computer Vision, Graphics, and Image Processing*, **44**, 87-116.
- Ioannou, D., Huda, W., and Laine, A. F. (1999), Circle Recognition through a 2D Hough Transform and Radius Histogramming, *Image and Vision Computing*, **12**, 15-26.
- Jia, X. and Nixon, M. S. (1995), Extending the feature Vector for Automatic Face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(12), 1167-1176.
- Ker, J.-I., Chen, F. F., and Lu, J. (1996), A Quick Hough Transform Parameter Search Algorithm for Flexible Part Inspection, *Journal of Manufacturing Systems*, **15**(6), 404-418.
- Kierkegaard, P. (1992), A Method for Detection of Circular Arcs based on the Hough Transform, *Machine Vision and Applications*, **5**, 248-263.
- Kohonen, K. (1982), Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, **43**, 59-69.
- Leavers, V. F. (1993), Survey - Which Hough Transform, *Computer Vision, Graphics, and Image Processing -Image Understanding*, **58**, 250-264.
- Li, H., Lavin, M. A., and LeMaster, R. J. (1986), Fast Hough Transform: A Hierarchical Approach, *Computer Vision, Graphics and Image Processing*, **36**, 139-161.
- Orlov, A. A., Sadykov, S. S., and Zhiznyakov, A. L. (2000), Using the Hough Transform for Extraction and Suppression of Ribs in X-rays of the Thorax, *Pattern Recognition and Image Analysis*, **25**(2) 365-367.
- Palmer, P. L., Kittler, J., and Petrou (1995), M, An optimizing Line Finder Using a Hough Transform Algorithm, *Computer Vision, and Image Understanding*, **17**(12), 1167-1176.
- Park, J. W. and Kim, K. D. (1992), A Study on Efficient Image Processing and CAD-Vision System Interface, *Journal of the Korean Institute of Industrial Engineers*, **18**(2), 11-22.
- Tian, Y. T. and Shah, M. (1997), Recovering 3D Motion of Multiple Objects Using Adaptive Hough Transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(10), 1178-1183.
- Tsai, D.-M. (1996), Detecting and Locating Partly Occluded Planar Parts by Clustering, *IIE Transactions*, **28**, 531-544.
- Yang, M. C. K. and Lee, J.-S. (1997), Hough Transform Modified by Line Connectivity and Line Thickness, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(8), 905-909.