論文92-29B-1-13

확장된 퍼지 클러스터링 알고리듬을 이용한 다중 첨두 검출

(Multiple Peak Detection Using the Extended Fuzzy Clustering)

金 秀 桓,* 曺 彰 皓,** 姜 景 辰,*** 李 太 遠***

(Su Hwan Kim, Chang Ho Cho, Kyung Jin Kang, and Tae Won Rhee)

要 約

저자들은 특징 공간에서 데이타 분류에 널리 사용되고 있는 퍼지 클러스터링 알고리듬을 확장하여 분류하고자 하는 데이타의 중요도를 고려할 수 있는 새로운 알고리듬을 제안한 바 있다.

본 논문에서는 확장된 퍼지 클러스터링 알고리듬이 다중 첨두의 위치를 검출하는 문제에서도 효과적인 해결책을 제시할 수 있음을 증명하기 위해 Hough 변화시 발생하는 다중 첨두 검출에 적용하였고 하나의 최고값만을 고려하는 대신 모든 데이타의 높이를 고려하기 때문에 잡음에 강하고 첨두의 모양에 독립적이며 적응적으로 검출할 수 있음을 실험적으로 입증한다.

Abstract

We have already proposed an extended fuzzy clustering algorithm which considers the importance of the data to be classified in a previous paper.

In this paper, we suggest the extended fuzzy clustering algorithm based new method to slove a multiple peak detection problem, and prove experimently that this algorithm can detect the multiple peak adaptively to the noise and the shape of peaks.

I. 서 론

주어진 데이타와 영상의 분석 분야에서 첨두의 위치를 찾는 문제는 매우 중요하며 실제로 많은 알고리듬들이 제안되고 있다. 첨두의 위치를 찾는 알고리듬을 설계하는데 사용되는 첨두의 정의는 여러가지가 사용되고 있다. 가장 간단한 경우로는 주어진 데이타들의 최고값을 첨두로 정의하는 경우이다. 그러나 잡음이 존재하는 경우에는 이 정의는 유효하지 않다. 따라서 일반적으로 사용되는 첨두의 정의는 국소 지역의 데이타 값들의 적분에 첨두의 존재와 위치가 의존한다는 것이다. 특히 잡음이 심하게 존재하는 경우 이와 같은 첨두의 정의는 더욱 호소력을 갖는다.[3][5-7][9]

또한 여러개의 첨두가 동시에 존재할 때는 각 첨두의 존재와 위치는 절대적 높이가 아닌 주위에 대한 상대적 높이에 의해 결정되기 때문에 첨두는 적응적으로 찾아져야할 뿐 아니라 첨두의 존재와 위치가 국소 지역의 데이타 값들의 적분에 의존할 때 첨두들간의 경계 문제를 해결해야 한다.

한편 본 논문에서는 확장된 퍼지 클러스터링 알고리듬[2][10][15]을 다중 첨두를 검출하는데 적용하여 이 알고리듬이 잡음에 강하고 첨두의 모양에 독립적이며 적응적으로 검출할 수 있음을 증명하고자 한다.

*準會員, **學生會員, ***正會員,
高麗大學校 電子工學科
(Dept. of Elec. Eng., Korea Univ.)
接受日字 : 1991年 10月 10日

Ⅱ. 퍼지 집합의 특징

퍼지 집합이란 집합에 속하는 원소를 확실히 구분할 수 없는 집합을 말한다. 예를 들면 "집합 A는 10에 근접한 실수"라는 퍼지 집합은 다음과 같은 순서쌍으로 표현된다.

$$A=\{(x, u_A(x)) : x\in X\}$$
$$u_A(x)=(1+(x-10)^2)^{-1} \quad -\infty<x<+\infty \qquad (1)$$

여기서 $u_A(x)$는 소속함수 를 나타내고, 그림으로 표현하면 그림1과 같다.

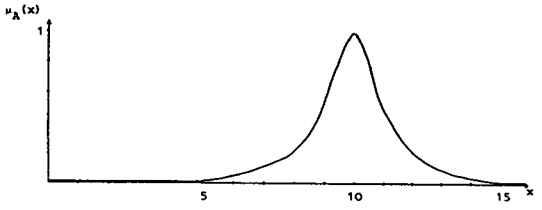


그림 1. "10에 근접한 실수"의 소속 함수
Fig. 1. Real numbers close to 10.

기존의 집합과의 차이점은 집합을 규정하는 특정함수(즉, 소속 함수)가 0과 1사이에 고르게 분포되는 특징이 있으며 다음과 같은 수식으로 나타낼 수 있다.

$$u_A(x) : A->\{0, 1\} \text{ 기존의 집합}$$
$$u_A'(x) : A->[0, 1] \text{ 퍼지 집합} \qquad (2)$$

퍼지 집합의 장점은 기존의 Crisp 집합이 표현하기 어려운 애매한 문제를 쉽게 설명할 수 있다는 것이다. 예를 들어, "오늘 날씨가 맑은가?"라는 문제에서 크리스프 집합의 경우는 하늘에 구름이 몇% 있을 때 맑다는 등을 상세히 정해 주어야 결론을 내릴 수 있다. 그러나 퍼지 집합은 하늘에 구름이 적으니까 맑다는 결론은 쉽게 내린다. 이러한 이유로 퍼지 집합은 시스템의 모델이 복잡한 경우에도 쉽게 제어할 수 있다.

현재 모든 시스템의 예측을 위해서는 주로 확률을 사용하여 결정을 내리고 있다. 그러나 통계처리를 할 수 없거나, 할 수 있어도 정확한 확률을 위해서 샘플을 많이 취해야 하는 경우에는 아무래도 확률을 가지고는 부족하다. 사실, 통계처리를 하는 것이 현재에는 가장 정확하다고 할 수 있지만, 전술한 문제를 위해서는 가능성이라는 개념을 도입하게 된다. 전문가의 충분한 경험을 바탕으로 예측한 가능성은 확률보다 정확하다고 할 수 있다. 쉽게 이야기해서, 결국 이러한 가능성은 퍼지집합에서의 소속함수가 되기 때문에 퍼지 이론의 장점이 되는 것이다. 즉, 확률적 처리가 불가능한 시스템에서도 퍼지 이론을 이용하면 간단하게 처리된다.

Ⅲ. 퍼지 클러스터링 알고리듬

클러스터링 알고리듬은 특정 공간내의 데이타들의 유사도를 계산하고 이를 근거로 하여 몇개의 부분집합으로 분할한다. 같은 클러스터에 속하는 데이타는 비슷한 성질을 가져야 하며, 다른 클러스터에 속하는 데이타는 가능한 한 상이한 성질을 가져야 한다.

이를 위해서는 먼저 데이타의 유사도를 측정하기 위해 어떤 수학적 성질을 사용할 것인가와 클러스터를 구분하기 위해 이러한 성질을 어떤 방법으로 사용할 것인가에 대한 문제가 대두된다. 이러한 문제는 주어진 문제에 따라 달라질 수 있다.

1. 퍼지 클러스터링 알고리듬의 특징

전통적인 클러스터링 알고리듬은 클러스터들간 경계가 명확하다는 가정하에서 하나의 데이타를 오직 하나의 클러스터에 할당한다. 그러나 이러한 가정은 클러스터간 경계가 애매하고 하나의 데이타가 하나의 클러스터에 속한다고 명확하게 말할 수 없는 실제 데이타의 특징을 반영하지 못한다. 이에 반해 최근 활발히 연구되고 있는 퍼지 클러스터링 알고리듬은 전통적인 평균과 공분산 대신 퍼지 평균과 퍼지 공분산을 사용하여 좀 더 적절한 데이타 분할 방법을 제공함으로써 퍼지 집합 이론의 성공적인 응용사례 중 하나로 받아들여지고 있다.

전통적인 클러스터링 알고리듬과 퍼지 클러스터링 알고리듬의 차이점은 나비문제라고 불리는 문제에서 찾아볼 수 있다.

그림2는 15개의 점으로 구성되는 데이타 집합 X를 표현하는 평면 공간을 나타내고 있다. 전통적인 클러스터링 알고리듬에 의해 이들 점들은 그림3에 나타낸 바와 같이 분류된다. 여기서 '1'은 왼쪽 클러스터에 속하는 것을 표시하며 '0'은 오른쪽 클러스터에 속하는 점을 표시한다. 반면 그림4와 그림5는 각각 퍼지 클러스터링 알고리듬에 의해서 각 점들에 대해 왼쪽과 오른쪽 클러스터에 속하는 귀속 정도를 나타낸다.

그림3에서는 한 가운데 점이 반드시 왼쪽 또는 오른쪽 클러스터중 하나에 귀속되어야 하기 때문에 비대칭이지만 그림4와 그림5에서는 이 점의 귀속 정도가 각각 0.5이므로 대칭이다.

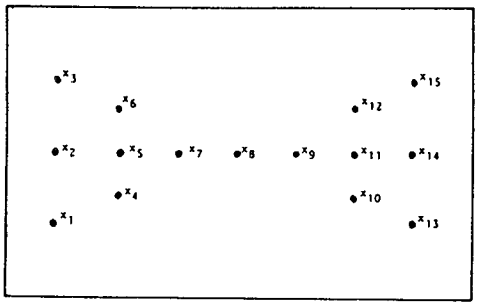


그림 2. 나비 모양 패턴
Fig. 2. A butterfly shape pattern.

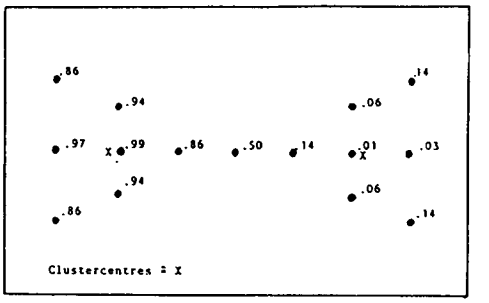


그림 3. 나비 모양 패턴의 크리스프 클러스터들
Fig. 3. Crisp clusters of butterfly shape pattern.

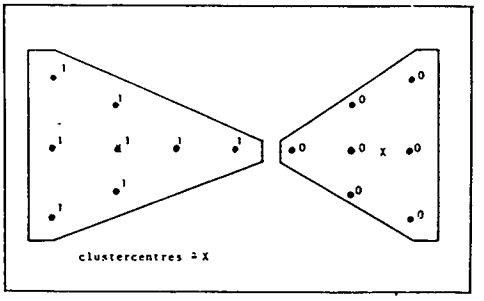


그림 4. 나비모양 패턴의 첫번째 퍼지 클러스터
Fig. 4. Fuzzy cluster 1 of butterfly shape pattern.

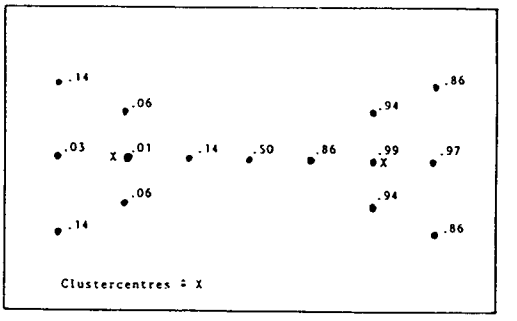


그림 5. 나비 모양 패턴의 두번째 퍼지 클러스터
Fig. 5. Fuzzy cluster 2 of butterfly shape pattern.

2. 퍼지 평균과 퍼지 공분산

확률 이론에서 사건A가 전체집합 R안에서 정확하게 정의된 점들의 집합이고 $\int_R dP=1$이라고 가정하자.

이때 사건A가 일반 사건이며 사건A의 확률값 P(A)은 다음과 같이 정의되고

$$P(A)=\int_A dP \qquad (3)$$

또는

$$P(A)=\int_R \chi_A(x)\,dP \qquad (4)$$

여기서 χ_A는 사건 A의 특성함수 ($\chi_A(x)=0$ 또는 1)이다.

사건 A가 퍼지 사건일 때 사건A의 확률값 P(A)은 다음과 같이 정의된다.

$$P(A)=\int_R f_A(x)\;dP \qquad (5)$$

여기서 f_A는 사건 A의 소속 함수 ($0\leq f_A(x)\leq 1$)이다.

이 식(5)는 식(4)의 일반화이다.

한편 확률값 P(A)와 관련있는 연속 형태의 퍼지 사건의 평균과 분산은 다음과 같이 표현될 수 있다.

$$v_A=\frac{1}{P(A)}\int_R x f_A(x)\;dP \qquad (6)$$

그리고

$$\sigma_A^2=\frac{1}{P(A)}\int_R (x-v_A)^2 f_A(x)\,dP \qquad (7)$$

식(6)과 식(7)에 대한 기본적인 생각은 많은 데이타들이 하나의 사건에 속해 있고 이들이 그 사건의 평균과 분산을 계산하는데 영향을 미친다는 것이다. 이러한 방법으로 계산되는 평균과 분산을 퍼지 평균과 퍼지 분산이라고 부른다. 식(6)과 식(7)로 부터 식(8)과 식(9)와 같이 표현되는 불연속 형태의 퍼지 평균과 퍼지 공분산 행렬이 얻어진다.

$$v_i=\frac{\sum_{k=1}^{n} f_i(x_k)\;x_k}{\sum_{k=1}^{n} f_i(x_k)} \qquad (8)$$

그리고

$$\sum_i = \frac{\sum_{k=1}^{n} f_i(x_k)\,(x_k - v_i)\,(x_k - v_i)^T}{\sum_{k=1}^{n} f_i(x_k)} \qquad (9)$$

즉, 퍼지 평균과 퍼지 공분산 행렬은 전통적인 평균과 공분산 행렬의 확장으로 이해될 수 있다. $f_i(x)$가 0 또는 1의 값을 가질 때 식(8)과 식(9)는 전통적 평균과 공분산 행렬의 정의이다.

3. 퍼지 c-means 알고리듬

퍼지 c-means 알고리듬은 식(10)과 같은 목적 함수를 최소화할 수 있도록 데이타 집합을 분할하는 알고리듬으로 순수 이론적 측면뿐 아니라 패턴 인식의 응용분야에서 많이 이용되고 있다.

$$J_m(U, V) = \sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ik})^m \| x_k - v_i \|^2,\ 1 \leq m < \infty \qquad (10)$$

m이 1보다 큰 경우에 모든 i, k에 대해서 $x_k \neq v_i$을 만족한다고 가정하면 다음 조건식들을 만족할 때만 (U, V)가 J_m의 최소화를 가능하게 한다.

모든 i, k에 대해서

$$u_{ik} = \left[\sum_{j=1}^{c}\left[\frac{\| x_k - v_i \|_A}{\| x_k - v_j \|_A}\right]^{2/(m-1)}\right]^{-1} \qquad (11)$$

모든 i에 대해서

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m\, x_k}{\sum_{k=1}^{n} (u_{ik})^m} \qquad (12)$$

이 알고리듬은 조건을 나타내는 식(11)과 식(12)를 포함하는 과정을 반복함으로써 J_m은 어떤 정해진 값에 수렴하게 된다. 즉, 퍼지 c-means 알고리듬은 아래의 1)부터 4)까지의 과정을 반복함으로써 수행된다.

1) 클러스터의 수c를 $2 \leq c \leq n$의 범위안에서 고정하고 $U^{(0)}$를 임의의 값으로 초기화한다. 또한p는 1로 초기화한다.

2) $U^{(p-1)}$와 식(12)를 이용하여 클러스터의 중심$\{v_i^{(p)}\}$를 계산한다.

3) $\{v_i^{(p)}\}$와 식(11)을 이용하여 $U^{(p)}$를 다시 개선한다.

4) 일반적인 norm을 이용하여 $U^{(p)}$와 $U^{(p-1)}$를 비교한다. 만약 $\| U^{(p)} - U^{(p-1)} \| \leq \varepsilon$를 만족하면 끝내고 그렇지 않다면 p를 1만큼 증가시키고 2)로 복귀한다.

4. 퍼지 MLE 알고리듬

퍼지 MLE 알고리듬의 소속 함수는 전형적인 평균이나 공분산 대신에 퍼지 평균 V와 퍼지 공분산 $\sum$를 사용한 MLE 알고리듬을 이용하여 구한다. 식(13)은 퍼지 MLE 알고리듬의 소속 함수를 나타낸다.

$$f_i(x) = \frac{P(x, v_i)}{\sum_{j=1}^{c} P(x, v_j)} \qquad (13)$$

여기서

$$P(x, v_i) = \frac{1}{(2\pi)^{1/D}|\sum_i|^{1/D}} \cdot exp\left[-\frac{1}{2}(x - v_i)^T \sum^{-1} (x - v_i)\right]$$

이고 D는 데이타를 나타내는 벡터차원이다. 또한 c는 미리 정해진 클러스터의 수이며 i의 범위는 $1 \leq i \leq c$이다.

그러나 일반적으로는 사용되는 퍼지 MLE 알고리듬은 소속 함수를 정할 때 주어진 데이타가 c개의 클러스터중 어느 클러스터에 속할지를 미리 결정할 수 있는 이전 확률값 B_i를 포함한다. 즉, 일반적으로 사용되는 퍼지 MLE 알고리듬은 아래의 1)부터 4)까지의 과정을 반복함으로써 수행된다.

1) 클러스터의 수c를 $2 \leq c \leq n$의 범위 안에서 고정하고 $U^{(0)}$를 임의의 값으로 초기화한다. 또한 p는 1로 초기화 한다.

2) $U^{(p-1)}$를 이용하고 식(14), 식(15), 그리고 식(16)을 계산함으로써 $\{B_i\}$, $\{v_i\}$, 그리고 $\{\sum_i\}$를 구한다.

모든 i에 대해서

$$B_i = \sum_{k=1}^{n} u_{ik}(x) \,/\, n \qquad (14)$$

$$v_i = \sum_{k=1}^{n} u_{ik}(x)\, x_k \,/\, \sum u_{ik}(x) \qquad (15)$$

$$\sum_i = \sum_{k=1}^{n} u_{ik}(x)\,(x_k - v_i)\,(x_k - v_i)^T \,/\, \sum u_{ik}(x) \qquad (16)$$

3) 위에서 구한 $\{B_i\}$, $\{v_i\}$ 그리고 $\{\sum_i\}$를 이용하여 식(17)에 따라 새로운 $U^{(p)}$를 구한다.

모든 i, k에 대해서

$$u_{ik}(x) = \frac{P(x_k, v_i)}{\sum_{j=1}^{c} P(x_k, v_j)} \qquad (17)$$

여기서

$$P(x_k, v_i) = \frac{B_i}{|\sum_i|^{1/D}} \cdot exp\left[-\frac{1}{2}(x_k - v_i)^T \sum^{-1} (x_k - v_i)\right]$$

이고 D는 데이타를 나타내는 벡터의 차원이다. 또한 c는 미리 정해진 클러스터의 수이며 i의 범위는 $1 \leq i \leq c$이다.

4) 일반적인 norm을 이용하여 $U^{(p-1)}$와 $U^{(p)}$를 비교한다. 만약 $\| U^{(p)} - U^{(p-1)} \| \leq \delta$를 만족하면 끝내고 그렇지 않다면 p를 1만큼 증가시키로 2)로 복귀한다.

5. 최적 퍼지 클러스터링 알고리듬

최적 퍼지 클러스터링 알고리듬을 두 단계로 이루어져 있다. 첫번째 단계에서는 대략적인 소속 함수를 구하고 두번째 단계에서는 데이타의 분포를 고려한 최적의 소속 함수를 구한다. 이를 위해 계산이 간단한 반면에 데이타의 분포를 고려할 수 없는 퍼지 c-means 알고리듬을 이용하여 대략적인 소속 함수를 구하고 데이타의 분포를 고려한 소속 함수를 구할 수 있으나 계산이 복잡한 퍼지 MLE 알고리듬을 이용하여 최적의 소속 함수를 구한다.

첫번째 단계에서 소속 함수의 초기값은 임의로 주어지며 퍼지 c-means 알고리듬을 수행함으로써 대략적인 소속 함수를 구한다. 두번째 단계에서는 첫번째 단계에서 얻어진 소속 함수값을 이용하여 퍼지 MLE 알고리듬을 수행함으로써 데이타의 분포를 고려한 최적의 데이타 분할을 얻을 수 있다.

그림6는 분포가 서로 다른 데이타를 분할한 예이다. 데이타는 (a)에 나타난 것처럼 한 부류는 분산이 매우 크고 데이타 밀도가 높지않으며 또 다른 부류는 분산이 매우 작고 데이타의 밀도가 매우 높다. 이들 두 부류사이에는 경계가 분명치 않다. 퍼지 c-means 알고리듬만을 사용하여 데이타를 분할 할 경우 (b)에 나타난 것처럼 경계 부분의 데이타를 잘못 분류한다. 즉, 분산이 크고 밀도가 낮은 부류의 데이타들 중 경계 부분의 데이타가 분산이 작고 밀도가 높은 부류에 포함된다. 그러나 퍼지 최적 클러스터링 알고리듬을 사용하여 데이타를 분할 할 경우 (c)에 나타난 것처럼 데이타를 제대로 분할한다.[10]

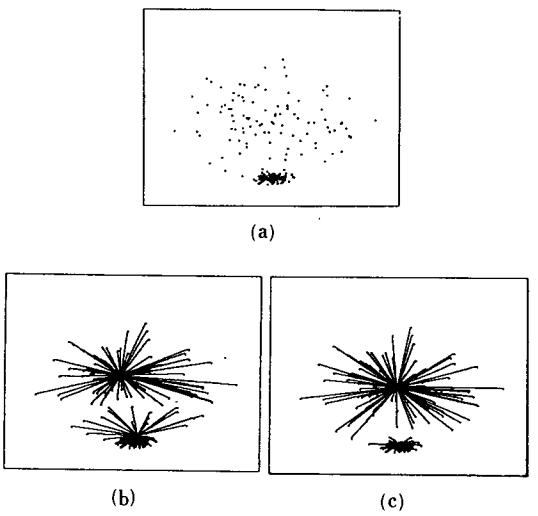


그림 6. 서로 다른 분포를 가진 데이타의 분할
(a) 가우시안 분포를 갖는 200개의 데이타 점들
(b) 퍼지 c-means 알고리듬
(c) 퍼지 최적 클러스터링 알고리듬

Fig. 6. Partition of simulated data with unequally variables.
(a) two hundrad data points generated form a Gaussian distribution,
(b) fuzzy c-means algorithm,
(c) fuzzy optimal clustering algorithm.

최적 퍼지 클러스터링 알고리듬은 기존의 방법에 비해 데이타의 분포를 고려할 수 있으며 최적의 소속 함수를 구할 수 있다는 장점 때문에 활발히 연구되고 있다. 본 논문에서도 이 알고리듬에 근거를 두어 데이타의 중요도를 고려할 수 있는 새로운 알고리듬을 제안하였다.

Ⅵ. 확장된 퍼지 클러스터링 알고리듬

기존의 퍼지 클러스터링 알고리듬을 확장하여 데이타의 중요도를 고려할 수 있도록 하기 위하여 데이타에 가중치를 주었으며 데이타의 중요도를 고려한 대표값의 의미를 갖는 각 클러스터의 중심값은 주위에 비해 단위 면적당 가중치가 상대적으로 높은 위치로 이동하는 특징을 갖도록 하였다. 물론, 데이타의 중요도가 모두 똑같은 경우에는 모든 데이타들의 가중치는 단위 크기를 가지며 기존의 퍼지 클러스터링 알고리듬과 같은 데이타 분할 효과를 갖도록 하였다. 또한 확률론에 근거한 초기 중심값 설정 알고리듬과 최적의 클러스터의 수를 정하기 위한 평가함수를 새롭게 제안하여 데이타 분할 성능을 증가시켰다. 한편 각 데이타들의 중요도는 각 데이타의 가중치로 나타낼 수 있으며 가중치의 정의는 응용하고자 하는 분야에 따라 다르게 정의될 수 있도록 하였다.

확장된 퍼지 클러스터링 알고리듬의 각 단계는 다음과 같다.

1) 초기 중심값 설정 알고리듬에 의해 각각 클러스터의 중심값을 가정한다.

2) 이 초기값을 확장된 퍼지 c-means 알고리듬에 이용함으로써 대략적인 소속 함수를 구한다.

3) 이 때 구해진 소속 함수에 확장된 퍼지 MLE 알고리듬을 적용함으로써 소속 함수를 최적화 한다.

4) 최적화된 정도에 대한 성능 평가를 한다.

5) 클러스터 수를 1만큼 증가시키고 주어진 데이타가 최적으로 분류될 때까지 위 과정을 반복한다.

1. 알고리듬의 수학적 해석

D차원의 벡터들로 이루어진 전체 데이타의 집합을 R, 집합R의 원소들 중 서로 다른 특징을 갖는 데

이타의 집합을 E라고 정의하고 집합R의 갯수를 N, 집합E의 갯수를 n이라고 가정할 때 집합E 원소들의 가중치의 합은 항상 집합 R의 갯수와 같다는 사실과 집합 E의 갯수는 항상 집합R의 갯수보다 작거나 같으며 같은 경우는 집합E 원소들의 가중치가 모두 단위 크기를 가질때이다는 사실은 매우 중요한 의미를 갖는다. 식(18)과 식(19)는 이러한 사실을 수식으로 나타낸 것이다.

$$N=\sum_{k=1}^{n} \text{가중치}(k) \tag{18}$$

$$N \geqq n \tag{19}$$

따라서 집합E의 원소 갯수가 집합R의 갯수의 D/(D+1)배 보다 작으면 기존의 방법에 비해 기억용량에 감소가 있고 집합E의 원소 갯수가 집합 R의 갯수보다 작으면 작을수록 기존의 방법에 비해 계산량의 감소가 커진다. 식(20)과 식(21)은 이러한 사실을 수식으로 나타낸다.

$$N \times D/(D+1) > n \tag{20}$$

$$N > n \tag{21}$$

2. 초기 중심값 설정 알고리듬

기존의 퍼지 클러스터링 알고리듬과 마찬가지로 확장된 퍼지 클러스터링 알고리듬도 각 클러스터의 중심값을 어떻게 예측하는가에 따라 데이타는 다르게 분류된다. 그러나 실질적인 상황에서 각 클러스터의 중심값을 예측하기는 불가능하며 주어진 정보를 최대한 이용하여 각 클러스터의 중심값을 예측하는 것이 필요하다.

주어진 데이타를 K+1개의 클러스터로 분할하기 위한 초기 중심값은 구하기 위해서는 우선 K개의 클러스터로 분할한 결과에서 각 클러스터에 포함되는 전체 데이타에 대해 일정 확률 밀도안에 포함되는 비를 구하였다. 그리고 가장 낮은 비를 갖는 클러스터의 최종 중심값을 제외한 나머지 클러스터의 최종 중심값을 그대로 초기 중심값으로 사용하였고 나머지 2개의 초기 중심값은 제외된 클러스터의 최종 중심값으로 부터 제외된 클러스터의 분산만큼 서로 반대 방향으로 떨어진 위치에 설정한다. 여기서 가장 낮은 비를 갖는다는 것은 최종 중심값이 클러스터간 경계에 위치할 가능성이 높다는 것을 의미한다.

한편 일정 확률밀도 경계는 주어진 데이타들의 평균이 v이고 분산이 $\sum$인 정규분포를 따른다고할 때 평균으로부터 $\{x:(x-v)^T\sum^{-1}(x-v) \leq \chi_D^2(\alpha)\}$의 거리안에 $(1-\alpha)\times 100\%$ 만큼의 데이타를 포함한다.[8]

다음은 초기 중심값 설정 알고리듬의 각 단계이다.

1) 주어진 데이타의 가중치를 고려하여 전체 평균과 공분산을 구한다.

2) 각 클러스터중 비율이 가장 낮은 클러스터를 선택한 후 이 클러스터를 두개로 분할하여 새로운 중심값으로 사용한다.

3) 확장된 퍼지 c-means 알고리듬과 확장된 퍼지 MLE 알고리듬을 차례로 적용하여 최적의 클러스터를 구한다.

4) 클러스터의 수가 주어진 최대 클러스터 수보다 작으면 단계2)로 돌아가고 그렇지 않으면 멈춘다.

3. 확장된 퍼지 c-means 알고리듬

확장된 퍼지 c-means 알고리듬은 기존의 퍼지 c-means 알고리듬이 사용하는 목적함수를 각 데이타의 중요도를 고려할 수 있도록 확장시켰으며 이 때 사용하는 목적함수는 식(22)와 같다.

$$J_{wm}(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{n} \text{가중치}(k)\times(u_{ik})^m \| x_k-v_i \|^2,\ 1\leqq m<\infty \tag{22}$$

m이 1보다 큰 경우에 모든 i, k에 대해서 $x_k \neq v_i$을 만족한다고 가정하면 다음 조건식들을 만족할 때만 (U, V)가 J_{wm}의 최소화를 가능하게 한다.

모든 i, k에 대해서

$$u_{ik}=\left[\sum_{j=1}^{c}\left[\frac{\| x_k-v_i \|_A}{\| x_k-v_j \|_A}\right]^{2/(m-1)}\right]^{-1} \tag{23}$$

모든 i에 대해서

$$v_i=\frac{\sum_{k=1}^{n} \text{가중치}(k)\times(u_{ik})^m x_k}{\sum_{k=1}^{n} \text{가중치}(k)\times(u_{ik})^m} \tag{24}$$

이 알고리듬은 조건을 나타내는 식(23)과 식(24)를 포함하는 과정을 반복함으로써 J_{wm}은 어떤 정해진 값에 수렴하게 된다. 즉, 확장된 퍼지 c-means 알고리듬은 아래의 1)부터 4)까지의 과정을 반복함으로써 수행된다.

1) 모든 데이타의 중요도에 따라 데이타에 가중치를 준다.

2) 클러스터의 수 c를 $2 \leqq c \leqq n$의 범위안에서 고정하고 $\{v_i^{(0)}\}$를 시작점 결정 알고리듬에 따라 초기화한다.

3) $\{v_i^{(0)}\}$를 이용하고 식(23)을 계산함으로써 $U^{(0)}$를 초기화 한다. 또한 p는 1로 초기화 한다.

4) $U^{(p-1)}$를 이용하고 식(24)를 계산함으로써 $\{v_i^{(p)}\}$를 다시 개선한다.

5) $\{v_i^{(p)}\}$을 이용하고 식(23)을 계산함으로써 $U^{(p)}$를 다시 개선한다.

6) 일반적인 norm를 이용하여 $U^{(p)}$와 $U^{(p-1)}$를 비교한다. 만약 $\| U^{(p)}-U^{(p-1)} \| \leq \varepsilon$를 만족하면 끝내고 그렇지 않다면 p를 1만큼 증가시키로 2)로 복귀한다.

4. 확장된 퍼지 MLE 알고리듬

확장된 퍼지 MLE 알고리듬은 기존의 퍼지 MLE 알고리듬에서 사용하는 퍼지 평균과 퍼지 공분산 대신 각 데이타의 중요도를 고려한 퍼지 평균과 퍼지 공분산을 사용한 알고리듬이다. 확장된 퍼지 MLE 알고리듬은 아래의 1)부터 4)까지의 과정을 반복함으로써 수행된다.

1) 모든 데이타의 중요도에 따라 데이타에 가중치를 준다.

2) 클러스터의 수c를 $2 \leq c \leq n$의 범위안에서 고정하고 확장된 퍼지 c-means 알고리듬의 최종적인 소속 함수를 이용하여 $U^{(0)}$를 초기화 한다. 또한 p는 1로 초기화 한다.

3) $U^{(p-1)}$를 이용하고 식(25), 식(26) 그리고 식(27)을 계산함으로써 $\{B_i\}$, $\{v_i\}$, 그리고 $\{\sum_i\}$를 구한다.
모든 i에 대해서

$$B_i=\sum_{k=1}^{n} \text{가중치}(k) \times u_{ik}(x) / \sum_{k=1}^{n} \text{가중치}(k) \qquad (25)$$

$$v_i=\sum_{k=1}^{n} \text{가중치}(k) \times u_{ik}(x)^m x_k / \sum_{k=1}^{n} \text{가중치}(k) \times u_{ik}(x)^m \qquad (26)$$

$$\sum_i=\sum_{k=1}^{n} \text{가중치}(k) \times u_{ik}(x)\,(x_k-v_i)\,(x_k-v_i)^T / \sum_{k=1}^{n} \text{가중치}(k) \times u_{ik}(x) \qquad (27)$$

4) 위에서 구한 $\{B_i\}$, $\{v_i\}$ 그리고 $\{\sum_i\}$를 이용하고 식(28)을 계산함으로써 새로운 $U^{(p)}$를 구한다.
모든 i, k에 대해서

$$u_{ik}(x)=\frac{P(x_k, v_i)}{\sum_{j=1}^{c} P(x_k, v_j)} \qquad (28)$$

여기서

$$P(x_k, v_i)=\frac{B_i}{|\sum_i|^{1/D}} \cdot exp\left[-\frac{1}{2}(x_k-v_i)^T \sum^{-1} (x_k-v_i)\right]$$

이고 D는 데이타를 나타내는 벡터의 차원이다. 또한 c는 미리 정해진 클러스터의 수이며 i의 범위는 $1 \leq i \leq c$이다.

5) 일반적인 norm을 이용하여 $U^{(p-1)}$와 $U^{(p)}$를 비교한다. 만약 $\| U^{(p)}-U^{(p-1)} \| \leq \delta$를 만족하면 끝내고 그렇지 않다면 p를 1만큼 증가시키고 3)으로 복귀한다.

5. 최적의 클러스터의 수 결정 문제

일반적으로 클러스터링 알고리듬은 클러스터의 수를 알고 있다는 가정하에서 주어진 데이타를 분할한다. 그러나 실제에 있어서 본질적으로 알려지지 않은 데이타 집합을 분할하는데 있어서 이러한 가정은 매우 부자연스러운 것이다. 따라서 데이타 분할 문제를 다룰때 항상 발생하는 중요한 문제는 얼마나 많은 클러스터가 존재하는지 결정하는 일이다.

이 문제를 해결하기 위한 많은 방법들이 제안되고 있으나 아직까지 일반화된 평가 함수는 제안되고 있지 않으며 분할하고자 하는 데이타의 분포에 따라 적절한 평가함수를 선택하여 사용하고 있다. 본 논문에서는 클러스터의 갯수를 점차 증가시키면서 클러스터 평가 함수의 값의 변화를 조사하는 방법을 선택하였다.

이 방법은 우선 주어진 클러스터 갯수에 대하여 목적함수가 최소가 되도록 주어진 데이타를 분할하고 분할된 결과에 대해 평가 함수를 적용한다. 그리고 이 과정을 클러스터의 갯수를 증가시켜 가면서 평가함수 값의 변화를 조사한다.

본 논문에서 사용한 에러 제곱합의 평가 함수 J는 c가 증가함에 따라 단조적으로 감소한다. 그러나 만약 n개의 데이타가 실제로 c개의 잘 분할된 클러스터로 구성되어 있다면 클러스터의 갯수가 c개일 때까지 J는 빠르게 감소하다가 클러스터의 갯수가 c개 이후 부터는 약간의 증감을 가지면서 매우 천천히 감소한다. 마지막에는 클러스터의 갯수가 n일 때 J 값은 0이 된다. 식(29)는 확장된 퍼지 클러스터링 알고리듬을 위한 에러의 제곱의 합으로 표시되는 평가 함수이다.

$$J_{we}=\sum_{i=1}^{c} \sum_{j=1}^{n} \text{가중치}(j) \times (u_{ij})^2 \| x_j-m_i \|^2 \qquad (29)$$

6. 모의 실험

모의 실험을 위한 집합R은 난수 발생 방법을 이용하여 0부터 199까지의 직선상에 누적시킨 1150개의 데이타이며 이는 N(39.52, 5.28)의 분포를 갖는 250개의 데이타, N(64.79, 10.48)의 분포를 갖는 300개의 데이타, N(99.99, 5.27)의 분포를 갖는 400개의 데이타, 그리고 N(170.05, 5.32)의 분포를 갖는 200개의 데이타로 이루어져 있다. 그리고 집합E는 발생된 난수의 종류로 이루어진 116개의 데이타이다. 집합E 원소들의 가중치는 각 난수의 발생빈도를 나타낸다.

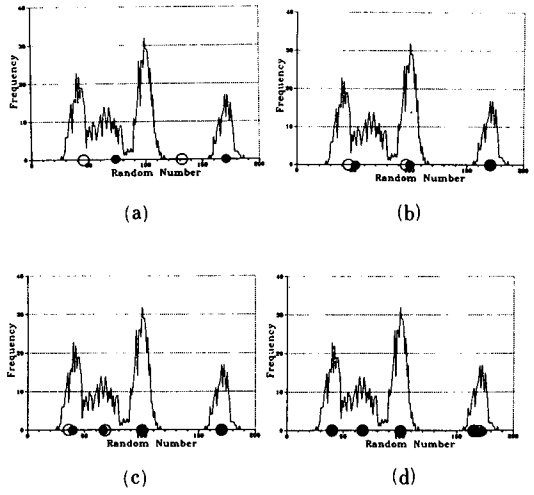


그림 7. 모의 실험 결과
(a) 2개의 클러스터 (b) 3개의 클러스터
(c) 4개의 클러스터 (d) 5개의 클러스터
Fig. 7. Simulation result.
(a) two clusters, (b) thcee clusters,
(c) four clusters, (d) five clusters.

그림7은 집합E 원소들과 집합E원소들의 가중치를 확장된 퍼지 클러스터링 알고리듬의 데이타로 사용하여 집합R를 분할하는 모의 실험 과정을 나타낸다. 모의실험 과정에서 클러스터의 수를 2개부터 5개까지 변화시키면서 단계적으로 설명하면 다음과 같다. (a)의 경우는 주어진 데이타를 2개의 클러스터로 나눈 결과이다. 초기 중심값은 전체 평균으로부터 전체 분산 만큼 서로 반대방향으로 떨어진 위치로 46.22와 133.58이었으며 최종 중심값은 73.03과 170.11이다. 이때 각 클러스터내 전체 데이타의 68%를 포함하는 일정 확률 밀도안에 포함되는 실제 데이타는 26.5%와 47.63%이다. 따라서 주어진 데이타를 3개의 클러스터로 나눌 경우 170.11은 초기 중심값으로 사용되며 나머지 2개의 초기 중심값은 73.03으로 부터 이 클러스터의 분산만큼 서로 반대방향으로 떨어진 위치에 존재하게 된다. (b)의 경우는 주어진 데이타를 3개의 클러스터로 나눈 결과이다. 초기 중심값은 48.00, 97.06, 그리고 170.11이었으며 최종 중심값은 53.06, 100.03, 그리고 170.05이다. 이 때 각 클러스터내 전체 데이타의 68%를 포함하는 일정 확률 밀도안에 포함되는 실제 데이타는 38.00%, 48.20%, 그리고 47.50%이다. 따라서 주어진 데이타를 4개의 클러스터로 나눌 경우 100.03과 170.05는 초기 중심값으로 사용되며 나머지 2개의 초기 중심값은 53.06으로 부터 이 클러스터의 분산만큼 서로 반대 방향으로 떨어진 위치에 존재하게 된다. (c)의 경우는 주어진 데이타를 4개의 클러스터로 나눈 결과이다. 초기 중심값은 38.00, 68.13, 100.03, 그리고 170.05이었으며 최종 중심값은 40.14, 67.14, 110.06, 그리고 170.05이다. 이때 각 클러스터내 전체 데이타의 68%를 포함하는 일정 확률 밀도안에 포함되는 실제 데이타는 50.46%, 48.83%, 48.26% 그리고 47.50%이다. 따라서 주어진 데이타를 5개의 클러스터로 나눌 경우 40.14, 67.14와 100.06은 초기 중심값으로 사용되며 나머지 2개의 초기 중심값은 170.05으로 부터 이 클러스터의 분산만큼 서로 반대 방향으로 떨어진 위치에 존재하게 된다. (d)의 경우는 주어진 데이타를 5개의 클러스터로 나눈 결과이다. 초기 중심값은 40.14, 67.14, 100.06, 165.00 그리고 170.11이었으며 최종 중심값은 40.12, 67.03, 100.02, 163, 77, 그리고 173.64이다. 각 과정의 마지막 단계에서 평가 함수를 적용하면 클러스터의 수가 4개일 때 부터 평가 함수는 일정한 값을 유지하여 최적의 클러스터의 수가 4개임을 나타낸다. 이 때 분할된 각 클러스터는 평균 40.14와 분산 5.66, 평균 67.14와 분산 9.28, 평균 100.0과 분산 5.25, 그리고 평균 170.00과 분산 5.33을 갖는다. 이 결과는 기존의 퍼지 클러스터링 알고리듬을 이용하여 집합R을 직접 분할한 결과와 같다. 한편 ○는 초기 중심값을 나타내고 ●는 최종 중심값을 나타낸다.

V. 확장된 퍼지 클러스터링 알고리듬을 이용한 다중 첨두 검출

1. 다중 첨두 검출의 개요

주어진 데이타와 영상의 분석 분야에서 첨두의 위치를 찾는 문제는 매우 중요하며 실제로 많은 알고리듬들이 제안되고 있다. 그러나 첨두의 위치를 찾는 알고리듬을 설계하기 위해서는 몇가지 해결해야 할 문제점이 있다. 우선 주어진 데이타에 적합한 첨두의 정의가 선택되어야 한다. 또한 여러개의 첨두가 동시에 존재할 때는 각 첨두의 존재와 위치는 절대적 높이가 아닌 주위에 대한 상대적 높이에 의해 결정되기 때문에 첨두는 적응적으로 찾아야 할 뿐아니라 첨두들간의 경계 문제를 해결해야 한다.

본 논문에서는 이러한 문제점을 해결하기 위하여 특징 공간내 데이타들의 높이를 데이타의 가중치로 사용하는 확장된 퍼지 클러스터링 알고리듬을 이용하였다. 특징 공간내 데이타들의 높이를 데이타의 가중치로 사용한다는 것은 주어진 위치에 높이 만큼

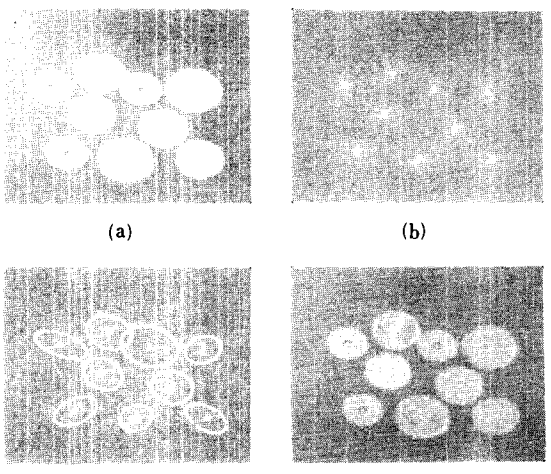
의 데이타가 누적되어 있는 것으로 해석하는 것을 의미하며 각 첨두의 위치는 면적당 데이타의 누적 정도가 많은 위치로 해석하는 것을 의미한다. 이 과정을 원의 중심을 찾기 위한 HT 계수 공간의 해석에 이용하였다.

2. 다중 첨두 검출의 실험 및 검토

다중 첨두의 검출 실험을 위하여 500원, 100원, 50원짜리 동전들과 버스 토큰들을 사용하였으며 이들 동전들은 완전한 원을 이루고 있다고 가정하였다.

임의로 배치된 동전들, 토큰들, 그리고 잡음을 실험 영상으로 받아들이고 이 실험 영상에 원의 중심을 찾기 위한 HT 변환을 적용하였다. 그리고 HT 계수 공간에서 다중 첨두의 위치를 정확하게 찾아내어 입력 영상에 원의 중심을 나타내었다.

동전들과 토큰들이 서로 겹쳐져 있지 않은 경우, 동전들과 토큰들이 서로 겹쳐져 있는 경우, 그리고 잡음이 포함되어 있는 경우에 대해 실험하였다. 그림8은 동전들과 토큰들이 서로 겹쳐 있지 않는 경우, 그림9는 동전들과 토큰들이 서로 겹쳐져 있는 경우, 그리고 그림10은 동전들과 토큰들 그리고 잡음들이 포함되어 있으며 겹쳐있는 경우의 입력 영상이다. 각 경우에 대해서 (a)는 입력 영상, (b)는 이 입력 영상에 대한 HT 계수 공간, 그리고 (c)는 (b)의 HT 계수공간

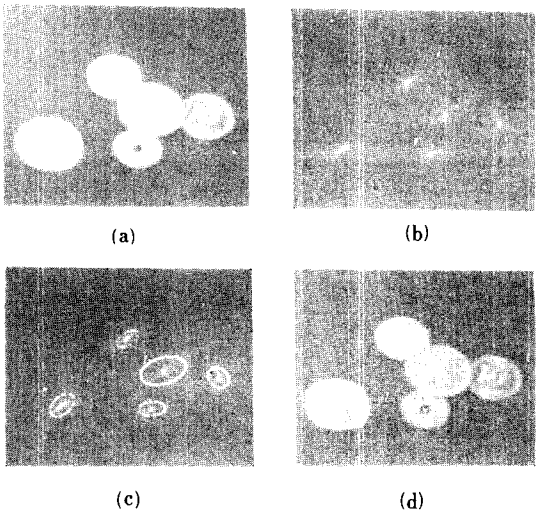


(a) (b)

(c) (d)

그림 8. 동전들과 토큰들이 겹쳐져 있지 않은 경우
(a) 입력 영상 (b) HT계수 공간
(c) 알고리듬 적용 결과 (d) 결과

Fig. 8. Coins and tokens not overlapped.
(a) input image, (b) HT parameter space,
(c) result of applying the alogrithm,
(b) result.

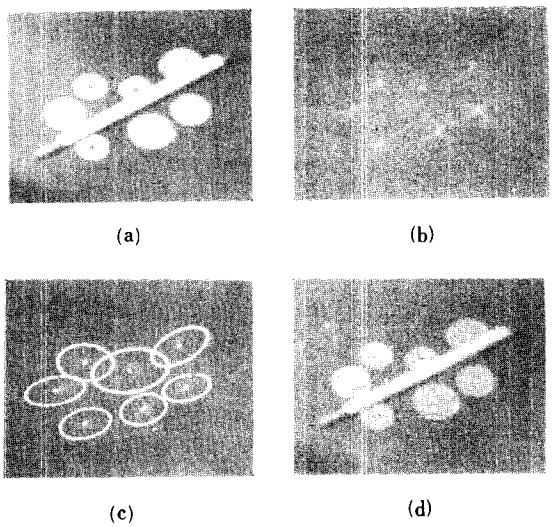


(a) (b)

(c) (d)

그림 9. 동전들과 토큰들이 겹쳐져 있는 경우
(a) 입력 영상 (b) HT 계수 공간
(c) 알고리듬적용 결과 (d) 결과

Fig. 9. Coins and tokens overlapped.
(a) input image, (b) HT parameter space,
(c) result of applying the alogrithm,
(d) result.



(a) (b)

(c) (d)

그림10. 잡음을 포함하는 경우
(a) 입력 영상 (b) HT 계수 공간
(c) 알고리듬 적용 결과 (d) 결과

Fig. 10. Noise included.
(a) input image, (b) HT parameter space,
(c) result of applying the algorithm,
(d) result.

에 확장된 퍼지 클러스터링 알고리듬을 적용한 결과로 타원들은 각 클러스터에 포함되는 전체 데이타의 68.3%를 나타내며 중심의 밝은 점은 각 클러스터의 중심이다. 찾고자 하는 첨두들의 중심이 각 클러스터의 중심과 일치하는 경우도 있으나 기본적으로는 하나의 타원 안에 하나의 첨두가 포함된다. 실제 첨두의 위치는 이 타원안의 최고점이며 각 클러스터의 중심값과 큰 차이가 나지 않는다. 마지막으로 (d)는 다중 첨두 검출의 결과의 정확도를 확인하기 위해 입력 영상에 원의 중심을 표시한 경우이다.

한편 첨두의 모양이 완전한 대칭일 때 각 클러스터의 중심과 첨두의 위치가 일치한다. 그러나 실제 데이타는 이러한 가정을 만족시키지 못하기 때문에 일정 확률 분포안에서 첨두의 위치를 보정하였다.

Ⅵ. 결 론

확장된 퍼지 클러스터링 알고리듬은 데이타 분류에 널리 사용되고 있는 기존의 퍼지 클러스터링 알고리듬에서 데이타의 성질에 따라 가중치를 줌으로써 분류하고자 하는 데이타의 성질을 고려할 수 있도록 한 알고리듬이다.

본 논문에서는 확장된 퍼지 클러스터링 알고리듬이 다중 첨두의 위치를 검출하는 문제에서도 효과적인 해결책을 제시할 수 있음을 증명하기 위해 HT변환시 발생하는 다중 첨두 검출에 적용하였다. 이를 위해 카메라를 통해 여러가지 경우의 데이타를 받아들였으며 이 실험 영상에 원의 중심을 찾기 위한 HT 변환을 적용하였다. 그리고 이때 형성된 HT 계수 공간내에 다중 첨두를 검출하기 위해 각 데이타의 높이를 가중치로 사용한 확장된 퍼지 클러스터링 알고리듬을 적용하였으며 다중 첨두 검출 결과의 정확도를 입증하기 위해 찾아진 첨두의 위치를 실험 영상에 표시하였다. 이렇게 함으로써 HT 변환 자체의 에러를 고려할 때 정확하게 다중 첨두의 위치를 찾을 수 있음을 입증했다.

參 考 文 獻

[1] R.V. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley-Interscience, 1973.

[2] James C. Bezdek and Joseph C. Dunn, "Optimal Fuzzy Partition: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions," *IEEE Trans.* vol. C-24, pp. 835-838, August 1975.

[3] Lawrence O'Gorman and Arthur C. Sanderson, "The Converging Squares Algorithm: An Efficient Method for Locating Peaks in Multidimensions," *IEEE Trans.* PAMI, vol. PAMI-6, no. 3, pp. 280-288, May 1984.

[4] Terano, Asai, Sugeno 지음, 박민용, 최항식 역, 퍼지 시스템의 응용 입문, pp. 139-158, 대영사, 1985.

[5] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," *IEEE Trans.* PAMI, vol. PAMI-9, no. 5, pp. 690-698, September 1987.

[6] E.R. Davis, "A high speed algorithm for circular object location," *PRL* 6, 323-333, 1987.

[7] E.R. Davis, "A modified Hough scheme for general circule location," *PRL* 7, 37-43, 1988

[8] Richard A. Johnson and Dean W. Wichern, Applied Multivariate Statistical Analysis, Prentice-Hall, 1988.

[9] J. Illingworth and J. Kittler, "Survey: A Survey of the Hough Transform," CGIP 44, 87-116, 1988.

[10] I. Gath and A.B. Geva, "Unsupervised Optimal Clustering," *IEEE Trans.* PAMI, vol. 11, no. 7, pp. 773-781, July 1989.

[11] Fangju Wang, "Fuzzy Supervised Classification of Remote Sensing Images," *IEEE Trans.* geoscience and remote sensing, vol. 28, no. 2, pp. 194-201, March 1990.

[12] Stephen P. Bonks, Signal Processing, Image Processing and Pattern Recognition, Prentice-Hall, 1990.

[13] H.-J. Zimmermann, Fuzzy set theory and its application, Kluwer Academic Publishers, 1991.

[14] 김수환, 임승민, 이규대, 이태원, "효율적 패턴 인식을 위한 순차적 GHT," 대한전자공학회 논문집 제28-B권, 제5호, pp. 327-334, 1991년 5월.

[15] 김수환, 강경진, 이태원, "확장된 퍼지 클러스터링 알고리듬을 이용한 자동 목표물 탐색," 대한전자공학회 논문지 제28-B권, 제10호, pp. 842-851, 1991년 10월.

著 者 紹 介

金 秀 桓 (準會員) 第28卷 B編 第5號 參照
현재 고려대학교 대학원 전자공학과 석사과정 재학중

●

曺 彰 晧 (學生會員)
1968年 10月 8日生. 1991年 2月 고려대학교 전자전산공학과 졸업 (공학사). 1991年 3月～현재 고려대학교 대학원 전자공학과 석사과정 재학중. 주관심분야는 패턴인식, 음성 인식 등임.

姜 景 辰 (正會員) 第28卷 B編 第10號 參照
현재 고려대학교 대학원 전자공학과 박사과정 재학중

●

李 太 遠 (正會員) 第25卷 第2號 參照
현재 고려대학교 전자공학과 교수