MPEG-4 객체의 브라우징 및 학습에 의한 추출 기법

MPEG-4 Object Browsing and Extraction by Learning

양만석, 오상욱, 설상훈
고려대학교 전자공학과
Mansuk Yang, Sangwook Oh, and Sanghoon Sull
Dept. of Electronics Engineering, Korea University

요 약

본 논문*은 MPEG-4 비디오 객체의 브라우징(browsing) 및 학습을 통한 객체 추출 기법을 제안한다. 제안된 학습에 의한 객체 추출 기법은, 객체 브라우징 시 임의 접근한 프레임에서 사용자가 내용 기반의 객체를 검색하기 위해 선택한 영역에 대한 인지적인 정보를 특징벡터(feature vector)로 history에 저장, 활용함으로써 프레임 내 객체의 계층적인 군집화(clustering)를 수행한다. 이러한 기법으로 인지적 개념과 근접하게 객체를 인식할 수 있음을 실험을 통해 확인하였다.

본문의 순서

I. 서 론

MPEG-4는 객체 기반의 독립적인 부호화 방식을 채택하여 내용 기반 객체들 간의 관계를 장면(scene)으로 표현하고 있다. 특히 비디오 프레임으로부터의 객체 추출 및 임의 형상의 비디오 객체로 구성된 내용 기반 데이터 구조가 MPEG-4에서의 주된 요구 사항들(requirements) 중의 하나이다[6]. 또한 MPEG-4 Version 1 Visual 부분에서는 "내용 기반 기능성(Content-Based functionality)", 즉 비디오 시퀀스 내의 객체에 대한 임의 접근을 용이하게 하고, 비디오 정보의 쉬운 조작을 가능케 하여 멀티미디어 서비스에 사용자와의 상호작용(user interaction)이라는 기능성(functionality)을 제공한다[7]. 이러한 기능성은 MPEG-4 객체의 브라우징 및 검색을 위한 필수 요소가 될 수 있다. 그러므로 본 논문에서는 MPEG-4 객체의 브라우징 및 학습을 통한 객체 추출 기법을 제안함으로써 이러한 요구사항 및 기능성을 만족시키고자 하였다.

영상분할(segmentation) 기법은 MPEG-4 표준에 포함되어 있지는 않으나 임의 형상의 신뢰성 있는 비디오 객체가 영상분할 과정을 통해 추출된다는 점에서 MPEG-4 코딩을 효과적으로 수행하는데 중요한 역할을 한다고 할 수 있다. 그러나 의미론적 객체 추적에 적용하여 외곽선 안팎의 불확실 영역을 분할하는 "morphological watershed algorithm"[3], "active contour model: snake"에 기반하여 객체의 움직이는 윤곽선 추출에 이용된 "energy-minimizing elastic contour model"[4] 등 기존의 계산에만 의존하는 자동 객체 추출 기법은 인간의 인지적인 개념을 반영하기 어렵기 때문에 의미 있는 객체의 추출은 사용자와의 상호작용을 통해 이루어져야 한다. 현재 멀티미디어 데이터의 급증으로 영상/동영상 등의 내용기반 검색에 대한 필요성이 증대되고 있으며, 특히 MPEG-4 표준화로, 객체에 대한 내용기반 브라우징 및 검색 방법이 요구되고 있으므로 MPEG-4 객체의 브라우징 및 검색에 필수 요소라 할 수 있는 사용자와의 상호작용에 대한 중요성이 더해지고 있다. 그러나 사용자가 직접 전반적인 객체 추출 과정에 참여하는 방법은 매우 비효율적이다. D.

* 본 연구는 한국과학재단 주관의 특정기초연구과제 (98-0102-04-01-3) 연구지원비 지원 하에 수행되었음.

Zhong과 S.-F. Chang은 이러한 문제점을 해결하고자 AMOS (Active system for MPEG-4 video object segmentation)을 이용하여 사용자로부터 받은 정보를 하위 레벨의 자동 영역분할 과정에 적용하려 하였지만[1][2], 본 논문에서 중점을 두고 있는, MPEG-4 객체의 브라우징 및 검색을 위한 사용자와의 상호 작용성을 고려할 때, 한 사용자로부터의 복잡하고 상세한 정보보다는 다수 사용자로부터 받은 간단한 정보의 통계적인 자료가 더 유용하다. 그러므로 본 논문에서는 내용 기반의 객체를 검색하기 위해 사용자가 선택한 영역에 대한 인지적인 정보를 특징벡터로 history에 저장, 이를 이용하여 군집화를 수행함으로써 각 클러스터(cluster) 내의 대표벡터를 최적의 객체 표현 값으로 설정하는, 학습에 의한 객체 추출 방법을 제안한다.

본 논문의 구성을 살펴보면, II에서 MPEG-4 비디오 객체에 대한 임의 접근 방식을 통한 브라우징 방법을, III에서는 사용자와의 상호작용 및 학습이 이루어지는 전반적인 과정을, 그리고 사용자로부터 얻어낸 특징벡터를 군집화하여 대표벡터를 얻어냄으로써 인지적인 객체 인식에 접근하는 방법을 IV에서 설명한다.

II. MPEG-4 비디오 객체의 임의 접근 방법

MPEG-4 Version 1 Visual 부분에서 제공하는 "내용 기반 기능(Content-Based functionality)" 중, 저장된 비디오 객체의 내용에 대한 임의 접근(Random Access of content in video sequences)을 가능케 하기 위해, "Microsoft FDIS Version 1.0"의 형태로 제공되는 MPEG-4 비디오 복호기(video decoder)를 MPEG-4 IM1-2D Player에 DLL 형태로 삽입함으로써 BIFS(Binary Format for Scenes) 부호화와 Multiplexing을[7] 통해 얻어진 비디오 스트림을 포함하는 "MPG4" 형태의 비트 스트림을 IM1 2D Player에서 동작 가능한 형태로 복원할 수가 있었다.

이렇게 해서 복원된 비디오 스트림에 대해서 그림 1에서 보는 바와 같이, 새로이 MPEG-4 비디오 브라우저(browser)를 구현함으로써 임의의 프레임(frame)에 대한 접근이 용이해졌다. 임의 접근 (random access) 기능을 위해 초기 복호화 과정에서 각 프레임에 대한 정보 및 시작 포인터(pointer)를 색인(index)으로 구성하였으며, 색인을 이용, 임의의 프레임에서의 일시정지(pause), 재시동(play), 특정 프레임의 검색(seek)

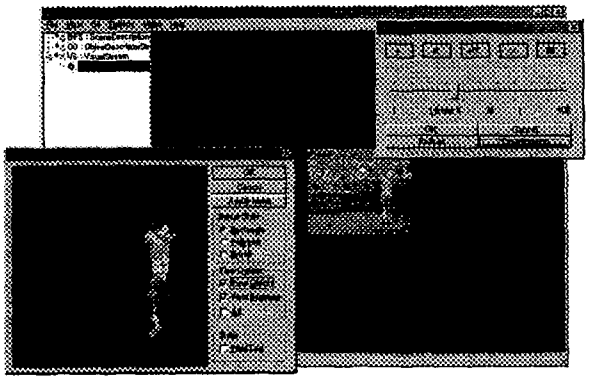

그림 1. MPEG-4 비디오 브라우저 및 사용자 인터페이스

이 가능하게 되었다. 이러한 임의 프레임에 대한 접근 기능 및 사용자 인터페이스(user interface)를 이용하여 비디오 스트림 내의 어느 프레임에서도 사용자의 인지적 정보를 받을 수 있으며, 또한 부가적으로 객체 표현자(object descriptor), 장면 표현자(scene descriptor)의 복호화 결과를 트리 구조로 보여주는 기능을 추가, 현재의 비트스트림에 포함된 객체의 종류를 쉽게 파악할 수 있도록 하였다.

III. 사용자와의 상호작용 및 학습

구현한 사용자 인터페이스 및 브라우저를 활용함으로써, 사용자가 원하는 프레임에서 관심있는 영역을 직접 표현할 수 있으며, 내용 기반 검색을 위해 여러 사용자들이 입력한 정보를 history에 저장, 이를 학습함으로써 어떠한 객체가 관심 있는지를 알아낼 수 있다. 또한 자동 영상분할 알고리즘을 사용하여 객체에 해당하는 근사 영역을 얻어낼 수 있다. 물론 특정 사용자가 객체 추출을 위한 과정에 적극적으로 참여하여 영역분할 정보를 상세히 부여할 수도 있지만[1][2], 사용자에게 편의성의 제공해야 하는 MPEG-4 비디오 객체의 브라우징 및 검색에는 비효율적이다. 그러므로 일반 사용자들로부터 받은 간단한 정보들을 효율적으로 저장, 활용함으로써 객관적이고 인지적 개념의 객체를 얻어낼 수 있다. 현재 자동 영역분할을 통한 객체 추출 기법은 인지적인 개념을 적용하기 힘들기 때문에, 다수 사용자로부터 받아들인 정보를 활용하는 것이 신뢰성 있는 객체 추출에 보다 유용하다.

본 논문에서는 사용자가 검색할 객체로 선택한 영역을 특징벡터로 history에 저장, 각 프레임 내에서 객체의 군집화에 이용한다. 이때, 사

용자의 관심 영역을 선택함에 있어 객체의 세부적인 윤곽선(contour)보다는 간단한 기하학 형태의 도형인 bounding box를 활용한다.

사용자와의 상호작용 및 학습을 통한 영역분할 및 객체 추출 기법을 사용하여 기존의 자동 영역분할 기법에서 나타나게 되는 과분할 영역(over-segmented region)을 객체에 해당하는 의미 있는 영역으로 통합함으로써 내용 기반 객체 추출이 가능하며, 소분할 영역(under-segmented region) 또한 보다 객체에 근접한 작은 영역으로 분할할 수 있다.

IV. 특징벡터의 군집화 및 객체 인식

III의 과정을 거쳐 받아들인 사용자의 인지적인 정보, 즉 4차원 공간 상의 특징벡터들은 각 객체별로 대표적인 특징벡터를 얻어내기 위해 군집화 과정을 거쳐야 한다.

본 논문에서 제안하는 학습을 통한 객체 추출 기법에서는 Hierarchical clustering의 NN(Nearest-Neighbor) 알고리즘, FN(Furthest-Neighbor) 알고리즘의[8] 단점을 보완할 수 있는 d_{mean}을 이용한 군집화 알고리즘을 사용하며 다음의 단계를 거쳐 군집화를 수행한다.

1. i 번째 사용자가 선택한 bounding box의 좌상점 $\mathbf{a}_i = (x_{i1}, y_{i1})$, 우하점 $\mathbf{b}_i = (x_{i2}, y_{i2})$을 4차원 특징벡터 $\mathbf{z}_i = (\mathbf{a}_i, \mathbf{b}_i)$ 로 4차원 공간 상의 한 점으로 둔다.
   $(i = 1, 2, \ldots\ldots, N)$

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{a}_i \\ \mathbf{b}_i \end{bmatrix} = \begin{bmatrix} x_{i1} \\ y_{i1} \\ x_{i2} \\ y_{i2} \end{bmatrix}$$

2. 특징벡터 $\mathbf{z}_i$와 M개의 기존 클러스터들의 각 대표벡터 $\mathbf{r}_j$ 간의 거리가 최소가 되는 클러스터 $\mathbf{C}$를 추출하고, 두 벡터 사이의 거리가 특정화된 임계값(threshold) 이하일 경우에는 새로운 특징벡터 $\mathbf{z}_i$를 클러스터 $\mathbf{C}$에 추가하고, 임계값 이상일 경우에는 새로운 클러스터를 생성한다. 즉, 모든 클러스터 $\mathbf{C}_j$에 대해,

$$\mathbf{r}_j \in \mathbf{C}_j \quad (j = 1, 2, \ldots\ldots, M)$$
$$d_{min} = \min(\| \mathbf{r}_j - \mathbf{z}_i \|), \quad (j = 1, 2, \ldots\ldots, M)$$

을 구하고, $\mathbf{z}_i$와 d_{min}의 거리인 $\mathbf{r}_j$의 대표벡터를 가지는 클러스터가 $\mathbf{C}$라면 새로운 특징벡터가 속하게 되는 클러스터 $\mathbf{C}_n$은,

$$\mathbf{C}_n = \begin{cases} \mathbf{C}, & \text{if } d_{min} < t \\ \mathbf{C}_{M+1}, & \text{if } d_{min} > t \end{cases}$$

이 된다.

3. $\mathbf{C}_n = \mathbf{C}$인 경우, 변경된 클러스터들을 최적화하기 위해서, 구성된 클러스터들의 대표벡터들 간의 거리를 순환적으로(recursively) 구해 나가면서 클러스터들을 merge시킨다. 즉, i번째 클러스터를 $\mathbf{C}_i$, 그 대표벡터를 $\mathbf{m}_i$, 그리고, j번째 클러스터를 $\mathbf{C}_j$, 그 대표벡터를 $\mathbf{m}_j$라고 두면,

$$d_{mean} = \min(\| \mathbf{m}_i - \mathbf{m}_j \|), \ (i = j = 1, 2, \ldots\ldots, M)$$
$$\text{if } i \neq j, \mathbf{C}_i \neq \varnothing, \text{and } \mathbf{C}_j \neq \varnothing$$
$$\mathbf{C}_j = \begin{cases} \mathbf{C}_i \cup \mathbf{C}_j, & \text{if } d_{mean} < t \\ \mathbf{C}_j, & \text{if } d_{mean} > t \end{cases}$$

4. 사용자로부터의 새로운 인지적 정보가 들어올 때마다 1~3의 과정을 실행하여 클러스터들을 재구성한다.

위의 4단계 과정을 거침으로써 각 프레임에 포함된 객체들을 표현할 수 있는 정보들이 군집화된다. 이러한 군집화된 정보들은 사용자가 검색하고자 하는 객체를 추출해 낼 수 있는 척도가 된다. 그러나 이러한 군집화의 결과는 사용되는 임계값 t에 따라 달라질 수 있다. 임계값에 영향을 미치는 요소 및 임계값과의 관계를 살펴 보면,

- **특징벡터의 특성** : 특징벡터를 구성하는 좌상점, 우하점 간의 거리에 비례하여 군집화의 에러율이 커진다. 따라서,

$$t \propto f_{z_i}$$

  t : threshold
  fz_i : characteristic function of feature vector

- **특징벡터들의 분산** : 프레임 내 객체들의 밀집 정도는 특징벡터들의 분산에 반비례한다. 군집화에 적용되는 임계값은 특징벡터들의 분산에 비례해야 한다. 이 때 특징

벡터들의 분산을 측정하기 위해 분산함수(scatter function)를 이용한다. 이 분산함수는 클러스터 내의 분산과 클러스터들 간의 분산을 종합적으로 측정하는데 사용된다. 클러스터 내의 분산은,

$$S_w = \sum_{j=1}^{M} S_j \qquad (j=1,2,.....,M)$$

$$S_j = \sum_{z_i \in Cj} (\mathbf{z}_i - \mathbf{m}_j)(\mathbf{z}_i - \mathbf{m}_j)^t \quad (j=1,2,.....,M)$$

가 되며, 여기에서 S_j는 j번째 클러스터 내에서의 분산, $\mathbf{z}_i$는 이 클러스터에 포함되어 있는 N개의 특징벡터이며, $\mathbf{m}_j$는 j번째 클러스터 내 특징벡터들의 평균이다. M이 클러스터의 전체 개수이므로 $j \le M$가 되며 그리고 S_w는 각 클러스터 내 분산 S_j들의 합을 나타낸다. 또한 클러스터들 사이의 분산은 다음과 같이 구할 수 있다[8].

$$S_B = \sum_{j=1}^{M} n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^t \quad (j=1,2,.....,M)$$

n_j는 각 클러스터에 포함되어 있는 특징벡터의 개수이며, $\mathbf{m}$은 모든 특징벡터들에 대한 평균벡터이다. 따라서 구하고자 하는 종합 분산값 S_T는 다음과 같고 임계값은 S_T에 비례한다.

$$S_T = S_w + S_B$$
$$t \propto S_T$$

- 특징벡터의 에러율 : 특징벡터의 에러율에 따라서 임계값 또한 변해야 한다. 이것은 일반 사용자가 입력하는 정보에 대한 에러율이 시스템 관리자가 입력하는 정보에 대한 에러율보다 크기 때문이다. 따라서, 입력하는 사용자에 대해서 가중치(weighting value)를 부여하며 가중치에 따라 임계값도 변해야 한다.

$$t \propto W_U$$

위의 세가지 요소는 독립적으로 임계값에 영향을 주기보다는 동시에 작용한다.

$$t = (\alpha f_{z_i} + \beta S_T + \gamma W_U)$$

여기서 α, β, γ는 실험을 통하여 구해지는 실험치라 할 수 있다. 이 식에서 임계값에 영향을 미치는 요소 중 사용자 가중치 W_U는 본 논문의 범위를 넘어서는 것이므로 실험에서는 모든 사용자의 가중치를 0으로 고정하여 사용했다.

V.실험 결과 및 고찰

본 논문에서는 제안한 학습을 통한 객체 추출 기법에서는 MPEG-4 비디오 시퀀스만을 사용하였다. II에서 언급했듯이, 현재 비디오 시퀀스를 포함하고 있는 MPEG-4 스트림이 부족하므로, IM1 Player에 MPEG-4 비디오 복호기를 삽입하고, "YUV"형태의 시퀀스를 부호화를 거쳐 압축된 비디오 스트림으로 만들었다. 이 과정에 의해 만들어진 비디오 스트림의 프레임 크기는 352 X 288의 "CIF" 형과, 176 X 144의 "QCIF" 형의 두 가지 SPEC을 따른다. 실험을 위해 BIFS 부호화, Multiplexing 과정에 의해 구성한 "MPG4" 형태의 스트림으로는 QCIF 형의 비디오 스트림이 포함된 Stefan, CIF 형의 비디오 스트림이 포함된 Weather, Coastguard, News, Singer, Dancer, Kids, Birthday 가 있으며 이 중 Stefan, Coastguard 스트림에 대한 프레임별 학습에 의한 객체 추출을 수행하였다.

본 실험을 위해서 사용한 자동 영역 분할 알고리즘은, "Image Segmentation Using Local Variation"[5]이며 객체를 추출해 내기 위해서 "region growing" 기법을 사용하였다.

그림 2에서는 Coastguard 스트림의 33번 째 프레임으로부터의 객체 추출 결과를 보여준다. 즉, 그림 2(a)를 자동 영역분할한 결과를 그림 2(e)에, 본 논문에서 제안한 학습에 의한 군집화에 의해 얻어진 객체 영역을 "region growing" 한 결과를 그림 2(b), (c), (d)에 보였다. 특히 그림 2(c), (d)에서는 개별적인 객체를, 그림 2(b)에서는 프레임 내의 전체 객체를 나타낸다.

보는 바와 같이 자동 영상분할 기법만을 적용했을 경우보다 인지적 개념에서의 객체를 효과적으로 추출해 내고 있다. 또한 본 논문에서 제안한 군집화 알고리즘의 효율성을 평가하기 위해 4차원 공간 상의 벡터를 표현하는 데 있어서, 사용자 선택 영역을 나타내는 좌상점과 우하점 각각의 동일 성분을 2차원 평면상에 나타냄으로써, 군집화된 결과를 그림 3에서 보여준다. 이 때, 비디오 스트림은 Stefan을 사용하였으며, 그림 3(a) 프레임이 포함하는 네 개의 객체들을 그림 3(b)에서, 그림 3(c) 와 그림 3(d)는 2차원 평면 상의 군집화 결과를 각각

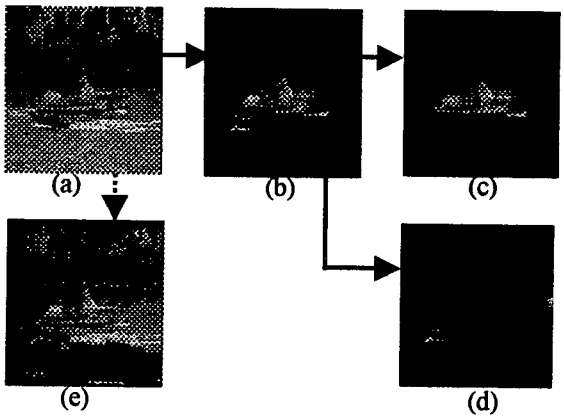

그림 2. 자동 영역 분할 기법 및 학습에 의한 객체 추출 결과 (a)원래 프레임 (stream : coastguard) (b)학습에 의해 추출된 모든 객체 (c),(d)학습에 의해 추출된 각 객체 (e)자동 영역 분할 기법에 의한 분할 영역

보여준다. 그림 3(d)에서 비록 객체 2, 3 이 겹쳐 있긴 하나, 그림 3(c)에서는 분리되어 있으므로 종합적으로 4 차원 벡터의 군집화 결과에 대한 효과적인 표현이라고 할 수 있다.

또한 모든 객체에 대해 동일한 임계값을 적용했을 때 발생할 수 있는 에러를 그림 4(a), (b)에서, 가변적 임계값(t = 60)을 적용하여 군집화한 결과를 그림 4(c), (d)에서 보여준다. 비교해 보면 가변적인 임계값 60 을 적용했을 경우 그림 2(b)에서 보는 바와 같이 의미 있는 두 개의 객체로 집중되어, 고정 임계값보다 가변적 임계값 적용이 보다 효과적임을 알 수 있다.

실험을 통해, 본 논문의 학습을 통한 객체 추출 기법을 이용함으로써 인간의 인지적 개념과 근사하게 의미 있는 객체를 추출할 수 있음을 확인할 수 있었다.

VI. 결 론

본 논문에서는 MPEG-4 비디오 브라우징 및 학습을 통한 객체 추출 기법을 제안하였다. 비디오 브라우저를 통해 선택된 한 프레임에 대해 사용자가 객체 검색을 위해 선택한 영역을 특징벡터로 하여 history 에 저장함으로써 군집화에 활용할 수 있으며, 학습을 통해 각 객체에 대한 클러스터들을 얻어낼 수 있다. 이 때, 그 에러율을 최소화하기 위해서는 가변적인 임계값이 적용되어야 한다. 제안한 학습을 통한 객체 추출 기법을 이용함으로써 기존의 자동 영역분할 기법에서 나타나게 되는 과분할 영역(over-segmented region)을 객체에 해당하는 의미

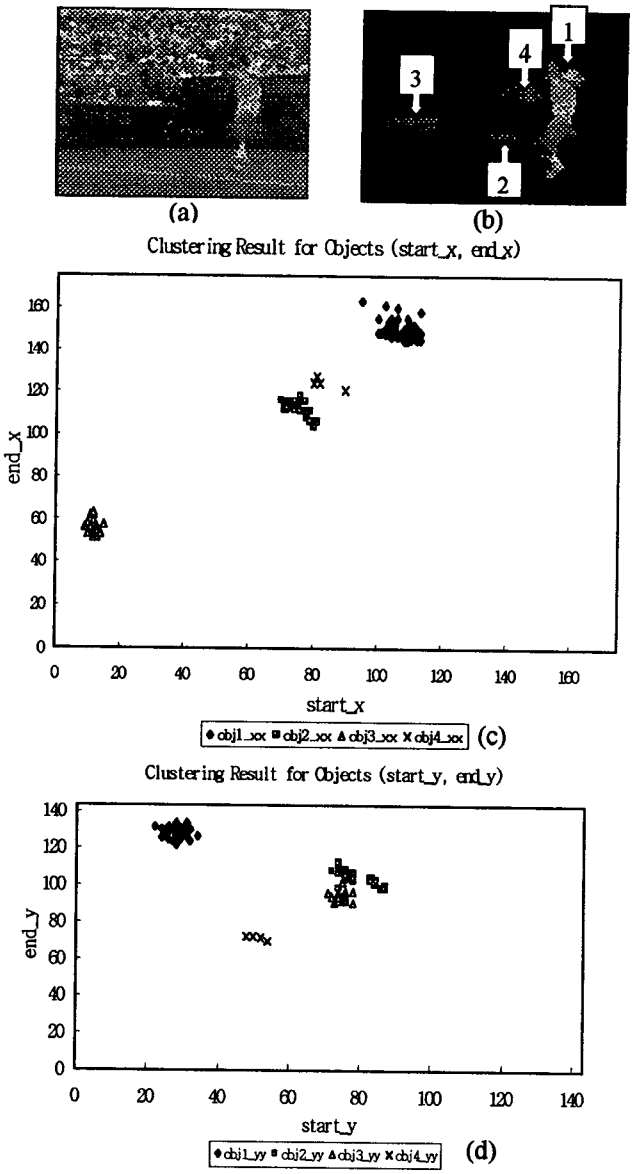

그림 3. Stefan #36 프레임의 군집화 결과 (a)원래 프레임 (b)프레임 내의 모든 객체 (c)특징벡터의 (x_{i1}, x_{i2}) 성분의 군집화 결과 (d)특징벡터의 (y_{i1}, y_{i2}) 성분의 군집화 결과

있는 영역으로 통합함으로써 내용 기반 객체 추출이 가능하며, 소분할 영역(under-segmented region) 또한 보다 객체에 근접한 작은 영역으로 분할할 수 있다. 또한 본 논문에서 중점을 둔 사용자와의 상호작용이라는 기능성은 MPEG-4 객체의 내용기반 브라우징 및 검색에 효과적으로 사용될 수 있다.

현재 본 논문에서는 특정 프레임에서의 학습에 의한 객체 인식 및 추출 기법을 제안하고 있지만, MPEG-4 비디오 객체를 다루기 위해서는 특정 프레임에 국한되지 않고 전체 프레임

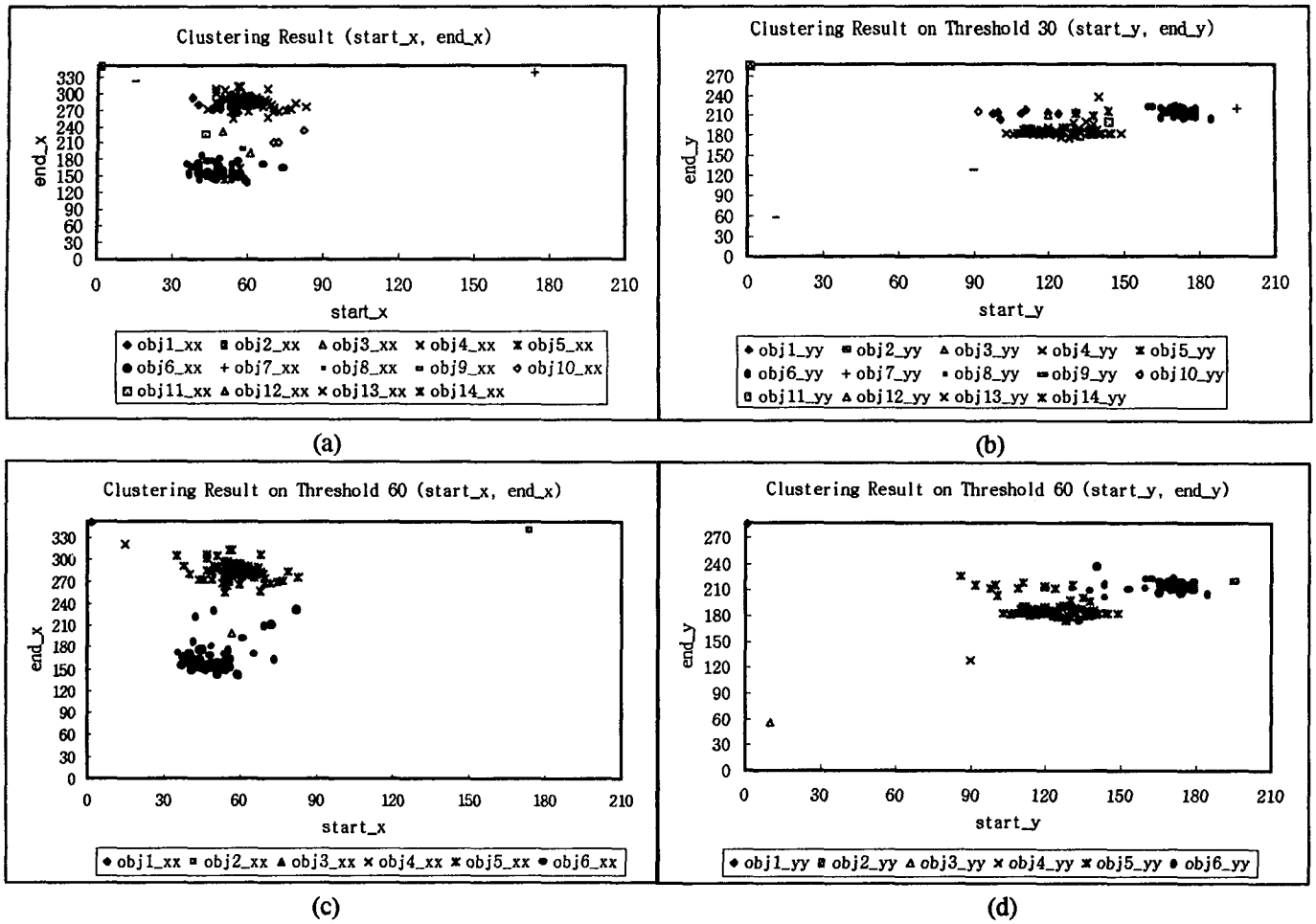

그림 4. Coastguard #33. 가변적인 임계값에 따른 군집화 결과 비교
(a) 임계값 30 (x_{i1}, x_{i2}) (b) 임계값 30 (y_{i1}, y_{i2}) (c) 임계값 60 (x_{i1}, x_{i2}) (d) 임계값 60 (y_{i1}, y_{i2})

에 적용되어야 하므로, 움직임(motion)을 이용한 객체 추적(object tracking) 알고리즘 개발을 연구 과제로 남겨두고 있다. 또한 임의의 프레임보다는 각 shot을 대표할 수 있는 대표 프레임을 추출하고 이를 이용하는 방법에 대한 연구 또한 필요하다.

참 고 문 헌

[1] D. Zhong and S.-F. Chang, "AMOS : An Active System For MPEG-4 Video Object Segmentation", IEEE Intern. Conference on Image Processing, October 1998, Chicago, IL.

[2] D. Zhong and S.-F. Chang, "Video Object Model and Segmentation for Content-Based Video Indexing", ISCAS'97, HongKong, June 9-12, 1997.

[3] G. Gu and M.-G. Lee, "Semantic Video Object Segmentation and Tracking Using Mathematical Morphology and Perspective Motion Model", ICIP'97, October 26-29, 1997. Santa Barbara, CA.

[4] N. Ueda and K. Mase, "Tracking moving contours using energy minimizing elastic contour models", Computer Vision-ECCV'92, Vol 588, pp 453-457, Springer-Verlag, 1992.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Image Segmentation Using Local Variation", Proc. of IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), 1998.

[6] Document ISO/IEC JTC1/SC29/WG11 N2723, "MPEG-4 Requirements, version 11 (Seoul revision)", Seoul MPEG meeting, March. 1999.

[7] Document ISO/IEC JTC1/SC29/WG11 MPEG97/N1730, "Overview of the MPEG-4 Standard", Stokholm MPEG meeting, Jul. 1997.

[8] Richard O. Duda, Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 2nd edition, 1999.