
대화형 멀티미디어 방송을 위한 객체 저장 방법

Object Store Method for Interactive Multimedia Broadcasting

한대영, 황부현, 김대인, 김재인, 나철수
전남대학교 전자컴퓨터공학부

Dae-Young Han(nara9yo@gmail.com), Bu-Hyun Hwang(bhhwang@chonnam.ac.kr),
Dae-In Kim(dikim@chonnam.ac.kr), Jae-In Kim(sereno3@naver.com),
Choul-Su Na(choulsu@daum.net)

요약

통방융합으로 인한 데이터 방송이 상용화됨에 따라 멀티미디어 객체에 대한 다양한 부가 정보를 제공하는 대화형 멀티미디어 방송 서비스가 가능하게 되었다. 대화형 멀티미디어 방송의 중요한 요소 중 하나는 방송 제공자와 시청자와의 데이터 상호 운용성이다. 상호 운용성이 높을수록 사용자의 요청이 있을 때 즉각적으로 시청자 관심 객체에 대한 정보를 제공할 수 있다. 본 논문에서는 대화형 멀티미디어 방송의 상호 운용성을 높이기 위한 객체 저장 방법을 제안한다. 제안 방법은 객체의 마스크 영상에서 객체 영역을 균등 분할하고, 각 영역에 대하여 객체정보 포함 최소 비율 검사를 통해 저장 영역을 비트합 형태로 군집화하여 저장한다. 그리고 실험을 통하여, 제안하는 방법이 기존의 대화형 멀티미디어 방송에 비하여 객체 영역 정보 저장을 위한 공간 사용을 줄임으로써 보다 효율적임을 확인한다.

■ 중심어 : | 객체추출 | 객체저장 | 마스크 영상 | 비트합 클러스터링 |

Abstract

Interactive multimedia broadcasting can serve various additional information of object in multimedia because of the commercialized data broadcasting by communication and broadcasting convergence. One of the most important factors in interactive multimedia broadcasting is User-Centric Interoperability. The higher User-Centric Interoperability, the more information of user-interest objects are served quickly by user request. This proposed method finds own area of the object in mask video and divides the area into equal parts. And then it store as a form of bitsum after clustering the area. As a result of experiment, We confirm the method is efficient to use space for storing position information of the object.

■ keyword : | Extraction of Object | Store of Object | Mask Video | BitSum Clustering |

1. 서론

산업의 새로운 패러다임인 컨버전스는 디지털 기술과 방송 서비스가 결합된 통방융합이라는 새로운 방식

의 멀티미디어 서비스를 가능하게 하였으며 디지털 방송 보급과 인터넷 회선의 발달로 HD급의 고품질 멀티미디어 방송 및 IP-TV 등의 데이터 방송 서비스가 가능하게 되었다. 또한 디지털기술이 방송기술과 결합하

여 사용자 중심 방송(Personalized Broadcasting), 전문가 채널, 리턴 채널을 이용한 대화형 멀티미디어 방송과 같은 사용자의 실시간 요구를 반영하는 사용자 중심의 방송에 대한 연구가 활발하게 진행되고 있다[1].

대화형 멀티미디어 방송 서비스는 리모콘을 통하여 사용자 요구에 대한 On-Demand 방식으로 영상 및 텍스트 기반의 교육, 날씨 및 쇼핑 정보와 같은 다양한 부가 서비스를 제공한다. 그러나 리모콘에 기반한 인터페이스는 고정된 메뉴에서 제한적으로 동작하며 TV 화면에 있는 시청자의 관심 객체에 접근할 수 없는 문제가 있으며 이러한 문제를 해결하기 위한 방법으로 터치스크린 인터페이스가 제안되었다. 터치스크린 인터페이스는 전통적인 메뉴 검색 방식을 벗어나 실시간 영상에서 시청자가 관심 있는 객체를 선택함으로써 시청자 중심의 다양한 서비스를 제공할 수 있는 기반을 마련하였다[2][3].

영상안의 객체는 다양한 형태, 질감과 색상으로 구성되며, 하나의 영상은 다수의 객체를 포함할 수 있다. 그리고 영상 내 객체 추출 결과를 마스크 영상이라 하며 마스크 영상은 원본 영상에서 해당 객체의 위치 정보 및 영역 정보를 포함한다. 그러므로 마스크 영상은 영상 내 객체 인식을 위한 기본 데이터로 사용된다. 또한 마스크 영상의 크기는 원본 영상의 해상도와 객체 크기에 비례하므로 추출하는 객체의 수가 많고, 해상도가 클수록 마스크 영상을 저장하기 위한 공간이 증가되는 문제가 있다. 본 논문에서는 이러한 문제를 해결하기 위하여 대화형 멀티미디어 방송에서 마스크 영상이 표현하는 객체의 영역정보를 효율적으로 저장하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 대화형 방송 및 객체 추출에 대한 관련 연구를 살펴보고, 3절에서는 제안하는 객체 정보 저장방법을 기술한다. 4절에서는 시뮬레이션을 통하여 제안방법의 우수함을 보이고, 마지막 5절에서는 본 연구에 대한 결론 및 향후 연구방향을 기술한다.

II. 관련 연구

1. 대화형 방송

대화형 방송시스템은 디지털 방송에 부가 서비스를 포함하고, 셋톱박스를 이용하여 시청자의 반응에 대하여 즉각적으로 결과를 반영 할 수 있는 시스템을 의미한다[4-6].

이러한 대화형 방송시스템은 멀티미디어 데이터뿐만 아니라 부가 서비스 정보를 함께 송출한다. 그리고 시청자는 화면상 객체 중 객체에 대한 보다 다양한 정보를 보고자하는 특정 객체를 선택하고 셋톱박스를 통하여 객체에 대한 부가적인 서비스를 요청한다. 시청자가 요청하는 부가 정보는 시청중인 프로그램 및 화면에 포함된 특정 객체에 대한 세부 정보로 구성된다. 그리고 이러한 서비스가 더욱 발전하여 사용자가 실시간으로 화면상 특정 객체에 대한 세부 정보를 요청하고 해당 객체에 대한 세부 정보를 방송하는 대화형 객체정보 제공서비스로 발전하고 있다. 대화형 멀티미디어 방송 시스템은 [그림 1]과 같이 구성된다.

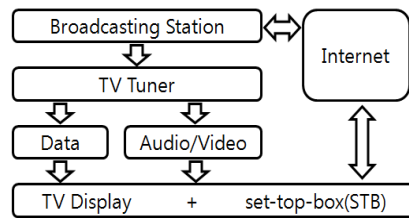


그림 1. 대화형 멀티미디어 방송 시스템

2. 객체 추적

시청자가 선택한 화면상의 객체정보를 분석 방법은 객체 추적 방식과 객체 추출 방식으로 나누어진다. 그리고 객체 추적 방식과 객체 추출 방식의 대표적인 방법으로 각각 움직임 기반 객체 추적방법과 마스크 영상 오버레이 기법이 사용된다.

1.1 움직임 기반 객체 추적방법

움직임 기반 객체 추적방법에는 화소 차이 값 분석방법, 배경 보정 방법 및 블록 정합 방법이 있다.

▪ 화소 값 차이 분석방법

화소 값 차이 분석방법은 고정된 카메라와 같이 고정된 배경에서의 객체 추적 방법으로 현재 프레임에서 이전 프레임의 화소 값을 빼면 고정된 배경은 모두 제거되고 움직임이 발생한 위치의 화소 값만 남게 된다. 화소 값 차이 분석방법으로 객체를 추적하는 방법은 식 1과 같다[7].

$$\Delta_n = |I_n - I_{n-1}|$$

$$M_n(x, y) = \begin{cases} I_n(x, y), & \Delta_n(x, y) \geq T \\ 0, & \Delta_n(x, y) < T \end{cases} \quad (1)$$

I_n : nth 프레임의 히스토그램, Δ : difference function, M : 움직인 영역, 색상의 숫자가 255인 경우 $T \approx 40$ 으로 정의된다.

▪ 배경 보정 방법

배경 보정 방법은 카메라가 이동하는 경우 배경이 이동되며 카메라의 이동에 따라 이동된 배경을 보정하는 방법으로 카메라 움직임을 보정함으로써 배경은 고정된 것으로 간주된다. 배경 보정 방법에서 실제로 움직임이 발생한 영역을 추출하여 객체를 추적하는 방법은 식 2와 같다[8].

$$x_{t-1} = f \frac{x_t + a \sin \theta y_t + f^a \cos \theta}{-a \cos \theta x_t + Y y_t + f}$$

$$y_{t-1} = f \frac{-a \sin \theta x_t + y_t - f Y}{-a \cos \theta x_t + Y y_t + f} \quad (2)$$

a : pan angle, Y : tilt angle,
 θ : initial inclination of the camera
 (x_t, y_t) : 현재 프레임에서의 화소 위치
 (x_{t-1}, y_{t-1}) : 이전 프레임에서 화소 위치

▪ 블록 정합 방법

블록 정합 방법은 화면 내 동작 정보를 추출하기 위한 방법으로, 블록 정합 방법에서 동작 정보는 카메라 이동에 따른 배경 영역의 움직임과 객체의 움직임으로 구분된다. 그리고 블록 정합 방법에서 객체의 움직임은 블록 정합 알고리즘을 적용하여 얻어진 동작 벡터를 분

석함으로써 영상 내 객체의 동작 정보를 추출한다. 블록 정합 방법의 종류에는 전역 탐색 알고리즘(FSBMA: Full Search BMA), 전역 탐색 알고리즘의 시간 복잡도를 개선한 3단계 탐색 알고리즘(TSS: Three Step Search), 그리고 탐색 공간을 적절하게 조절하는 적응적 전역 탐색 알고리즘(AFSBMA: Adaptive FSBMA) 등이 있으며 이 중 가장 정확한 블록 정합을 할 수 있는 전역 탐색 알고리즘이 가장 많이 사용된다[9-12].

움직임 기반 객체 추적방법은 객체의 형태가 단순할수록 빠른 추적이 가능하다는 장점이 있다. 하지만 객체의 형태가 복잡하고 하나의 영상에서 동시에 두 개 이상의 객체를 추적하는 경우 시간 복잡도가 높아져 많은 계산을 필요로 하므로 다수의 객체를 갖는 대화형 방송에는 부적합하다.

1.2 마스크 영상 오버레이 기법

마스크 영상 오버레이 기법은 기존 방송에서 사용되는 크로마키 기법을 응용한 것으로 크로마키 기법은 단일 색의 배경 공간을 설정하고 카메라를 통하여 입력 받은 화면에서 배경색을 제거하여 원하는 객체정보를 추출하는 기법을 말한다[13].

마스크 영상 오버레이 기법은 화면의 객체를 인식하기 위하여 마스크라는 특수한 영상을 사용한다. 마스크는 객체의 영역 정보만이 포함된 흑백의 영상으로 크로마키 기법을 사용하여 추출된 순수한 객체 영역 정보가 마스크 파일이 된다. 만약 시청자가 방송 화면에서 특정 객체를 선택하면 현재 화면과 마스크 영상을 오버레이하여 선택한 좌표가 마스크 영상의 영역에 포함되는지를 판별하여 시청자가 선택한 객체를 인식한다. 마스크 영상 오버레이 기법에서의 객체 추출 과정은 [그림 2]와 같다.

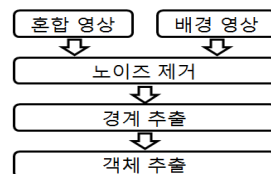


그림 2. 객체 추출 과정

마스크 영상 오버레이 기법은 원본 화면과 비교할 때 거의 손실 없이 객체 영역 정보를 저장함으로써 높은 정확성을 보인다. 또한 객체 인식 과정에서 객체의 좌표 정보만을 가지고 포함 유무를 판별하여 객체를 인식하므로 매우 빠른 객체 인식이 가능하다. 반면에 마스크 영상 오버레이 기법에서 사용하는 마스크 파일의 크기는 방송되는 화면의 해상도에 비례한다.

예를 들어 1080p의 HD 영상은 1,920(H) x 1,080(V)의 해상도를 가지므로 하나의 프레임에서 하나의 객체에 대한 마스크 파일은 약 2MB 정도의 저장 공간을 필요로 한다. 또한 HD 영상의 최소 프레임을 24프레임으로 가정할 때 1초의 객체 영상을 저장하기 위하여 마스크 파일 저장을 위하여 약 48MB의 디스크 공간이 필요하다. 그리고 하나의 화면에 다수의 객체가 존재하는 경우 각각의 객체에 대한 마스크 파일이 필요하므로 객체 수에 비례하여 많은 디스크 공간이 필요하다. 따라서 보다 현실적이고 효율적인 대화형 멀티미디어 방송 서비스를 위해서는 원본 손실이 적으면서도 저장 공간을 효율적으로 사용할 수 있는 객체 정보 저장 알고리즘 개발이 필요하다.

III. BSCM 알고리즘

본 논문에서는 하나의 영상에 포함된 객체정보 분석을 위하여 마스크 영상 오버레이기법에 기반한 BSCM(BitSum Clustering for Mask) 알고리즘을 제안한다. BSCM 알고리즘은 균등 분할 기법을 적용하여 저장 공간 낭비를 최소화함으로써 마스크 파일을 사용하여 객체 정보를 분석하는 마스크 영상 오버레이 기법의 저장 공간 문제를 해결할 수 있다. 본 논문에서 사용하는 용어는 다음과 같다.

3.1 용어 정의

마스크 영상

마스크 영상은 원본 영상과 동일한 해상도를 갖으며, 객체의 영역 정보만을 포함한 흑백 영상을 의미한다.

프레임

프레임은 동영상의 특정 시점의 정지된 하나의 장면을 의미한다.

사용자 선택 좌표

사용자 화면을 2차원 평면으로 가정 할 때, 화면상의 임의의 지점은 x축과 y축에 대한 좌표로 표현된다. PT 는 화면에서 사용자가 선택한 x축과 y축 상의 순서쌍 $\langle x_pos, y_pos \rangle$ 로 정의된다.

$$PT = \langle x_pos, y_pos \rangle \quad (3)$$

$$0 \leq x_pos \leq width$$

$$0 \leq y_pos \leq height$$

PT : 사용자가 선택한 x축과 y축 상의 좌표

OOA: 객체 영역

BSCM 알고리즘은 마스크 영상을 $\langle 0, 0 \rangle$ 좌표부터 $\langle width, height \rangle$ 좌표까지 순차적으로 탐색하여 화면상의 객체 정보를 손실 없이 포함하는 최소 크기의 사각형에 대한 좌표를 저장한다.



그림 3. 객체 영역

OOA(Object's Own Area)는 마스크 영상에서 객체 영역을 포함하는 최소 크기의 사각형으로 [그림 3]과 같이 좌측 상단과 우측 하단을 나타내는 두 개의 좌표 $\langle PT_{top}, PT_{bottom} \rangle$ 로 정의된다.

$$OOA = \{PT_{top}, PT_{bottom}\} \quad (4)$$

$MIN(A, B)$

$MIN(A, B)$ 함수는 두 인자 중 더 작은 쪽의 값을 반환한다.

D: 객체 영역의 분할 단위

BSCM 알고리즘은 객체 영역 정보를 저장하기 위하여 객체 영역을 균등한 크기로 분할하고 분할된 영역을 객체별로 균집화한다. *D*는 균집화 단계 이전에 객체 영역을 균등한 크기로 분할하기 위하여 임의로 지정한 크기이다. 객체 영역의 가로, 세로 크기를 각각 *Object_W*, *Object_H*라 할 때 객체 영역의 분할 크기는 $MIN(Object_W, Object_H)$ 이하의 값을 갖는다.

$$0 < D \leq MIN(Object_W, Object_H) \quad (5)$$

Min_Rate: 최소 객체정보 포함 비율

Min_Rate 이상의 포함 비율을 갖는 분할된 객체 영역들은 해당 객체의 균집화 대상에 포함된다. *Min_Rate*는 사용자가 지정한 0과 1사이의 실수 값으로 *Min_Rate*가 0.5(50%) 일 때 그림 7의 0~7번 분할 영역은 객체 정보를 50% 이하로 포함하므로 균집화 대상에서 제외된다.

$$0 \leq Min_Rate \leq 1 \quad (6)$$

BN: 분할 영역 식별 번호

*BN*은 분할된 객체 영역을 빠르게 접근하기 위한 인덱스 값을 의미한다. 객체 영역의 각 분할 영역은 [그림 4]와 같이 순차적으로 부여된다.

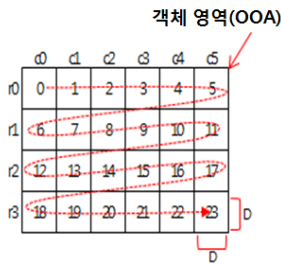


그림 4. 분할 영역 식별 번호

$$BN = c + (cs \cdot r) \quad (7)$$

c: 열의 위치, *cs*: 한 행당 열의 개수, *r*: 행의 위치

SC: 분할 영역의 식별번호 집합

*SC*는 *Min_Rate* 이상의 객체 정보를 포함하는 분할 영역의 식별번호를 저장하기 위한 배열이다.

$$SC = \{BN / BN \geq Min_Rate\} \quad (8)$$

각 분할 영역이 *Min_Rate*를 만족하는 지를 판별하는 방법은 다음 식 9와 같다.

$$\text{객체정보 포함 비율} = \frac{\text{객체 영역 내 1의 개수}}{\text{객체 영역의 넓이}} \quad (9)$$

FI: 프레임 건너뛰기 간격

BSCM 알고리즘은 마스크 영상의 빠른 탐색을 위하여 일정 크기만큼의 프레임을 건너뛰며 수행된다. 마스크 영상이 의미하는 객체의 움직임이 정적이라면 상대적으로 *FI*를 크게 설정한다.

$$0 < FI < \text{영상의 총 프레임} \quad (10)$$

3.2 마스크 저장 알고리즘

BSCM 알고리즘에서 객체 분석을 위한 마스크 저장 알고리즘은 다음 [그림 5]와 같은 단계로 수행된다.

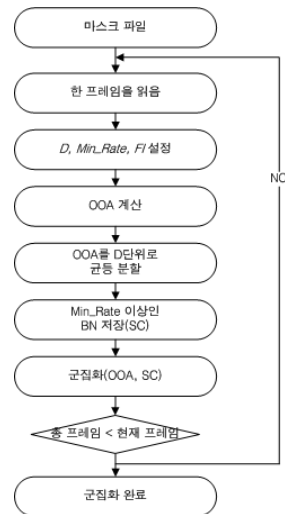


그림 5. BSCM 알고리즘

BSCM 알고리즘 설명을 위하여 1080p의 해상도를 갖는 [그림 6]과 같은 객체에 마스크 저장 알고리즘을 적용한다. 마스크 영상은 움직임 없이 30프레임 동안 연속된다고 가정한다.



그림 6. 포도 객체의 마스크 영상

<단계 1> D , Min_Rate , FI 설정

설명을 위하여 적용하는 D , Min_Rate , FI 는 다음과 같다.

상수값 설정
$D=20$
$Min_Rate=0.3$
$FI=10$

<단계 2> 마스크 영상에서 객체 영역 탐색

객체를 포함하는 최소 크기의 사각형에 대한 좌표 정보를 다음과 같이 PT_{top} 와 PT_{bottom} 에 대한 순서쌍으로 저장한다. 탐색의 시작점은 마스크 영상의 왼쪽 좌측 상단의 점 $\langle 0,0 \rangle$ 이다.

입력값: 마스크 영상
출력값: OOA
$PT_{top}=\langle 300,700 \rangle$
$PT_{bottom}=\langle 500,800 \rangle$
$OOA=\{PT_{top}, PT_{bottom}\}$

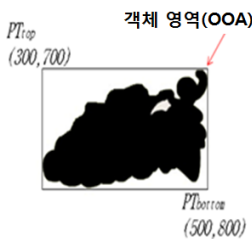


그림 7. 발견된 객체 영역

<단계 3> 순 객체 영역 균등 분할 및 식별 번호 부여

균등 분할 및 식별번호 부여
① OOA를 D 단위로 분할
② 식별번호 부여



그림 8. 균등 분할 및 식별번호 부여

<단계 4> 객체정보 포함 비율 검사 및 저장 영역의 식별번호 저장

저장 영역의 식별번호 저장
① 객체정보 포함 비율 계산
② 객체정보 포함 비율이 Min_Rate 이상인 경우 해당 SC에 추가

정의된 Min_Rate 는 0.3이므로 [그림 7]의 영역에서 0.3 이상의 객체 정보 포함 비율을 갖는 영역은 [그림 8]과 같은 영역으로 구성된다.



그림 9. 지지도가 30% 이상인 영역

$$SC=\{8, 9, 13-19, 22-39, 40-46, 48, 49\}$$

<단계 5> BitSum Clustering

설명을 위하여 각 군집당 비트열 길이는 32bit으로 지정한다. 군집화 방식은 SC에 저장된 영역 식별 번호에 해당되는 비트열에서의 해당 영역의 위치를 찾아 논리합을 취하여 각 군집별 비트열을 생성 한다. 이 단계의 수행을 위해 몇 가지 변수들을 사용하는데 다음과 같다.

nGroup: 군집번호
nMyPos: 분할 영역의 식별 번호의 비트 위치
nShifted: 1을 *nMyPos* 만큼 좌측 쉬프트 한 값
BitSum: 군집

```

입력값: OOA
BitSum=0;
for(i=0;i<(number of BN;i++){
    nGroup=OOA[i]/32;
    nMyPos=OOA[i]%32;
    nShifted=(1<<nMyPos);
    BitSum|=nShifted;
}
    
```

예에서 사용된 마스크 영상의 군집화 결과는 [표 1]과 같다.

표 1. 군집화 결과

군집 No.	비트열	Bit Sum
0	1111111111001111110001100000000	2 4,291,814,144
1	0000000000000011011111111111111	2 229,375

<단계 6> 군집화 결과 저장

BSCM 알고리즘에서 객체 마스크 영상 정보는 데이터베이스에 저장되며 데이터베이스의 MASK 테이블 스키마는 [표 2]와 같다.

표 2. MASK 테이블 스키마

필드명	타입	비고
video_id	int	영상 ID
object_id	int	객체 ID
s_frame	int	시작 프레임
e_fame	int	끝 프레임
ooa	varchar	객체 영역
d	int	분할 크기
group_num	int	군집 번호
bit_sum	unsigned long	군집내 비트 누적 합

[표 2]의 MASK 테이블에서 video_id 필드는 영상 ID이며 object_id 필드는 특정 영상에 존재하는 하나 이상의 객체 구분을 위한 객체 ID이다. s_frame과 e_frame 필드는 특정 객체가 존재하는 시작 프레임과 과 끝 프

레이미간의 구간 정보이며, boundary 필드는 객체 영역 정보를 나타낸다. d 필드는 객체 영역의 분할 단위, group_num 필드는 저장되는 영역의 군집 번호, bit_sum 필드는 군집화 된 비트열의 누적 합을 의미한다. [표 1]의 군집화 결과를 사용하여 데이터베이스화 한 내용은 [표 3]과 같다.

표 3. MASK 테이블

video_id	object_id	s_frame	e_frame	ooa	min_size	group_num	bit_sum
1	1	1	10	300,700,500,800	20	0	4,291,814,144
1	1	21	30	300,700,500,800	20	1	229,375

IV. 실험 및 분석

제안한 BSCM 알고리즘의 성능을 평가하기 위하여 1280(W) * 720(H)의 해상도를 갖는 다수의 마스크 영상에서 마스크 영상을 추출하여 사용했다.

4.1 Min_Rate별 군집 수 변화

Min_Rate가 군집 수에 미치는 영향을 알기위해 D, Min_Rate, FI값을 두 가지 케이스로 정의한다.

군집1: D=20, Min_Rate=03, FI=10

군집2: D=20, Min_Rate=0.5, FI=10

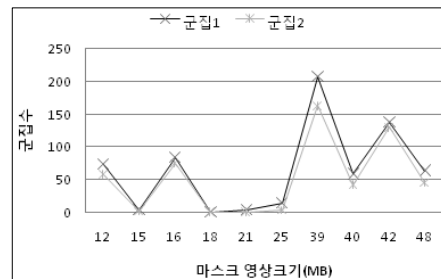


그림 10. Min_Rate를 달리한 BSCM 결과

[그림 10]은 Min_Rate를 달리하여 BSCM 알고리즘을 적용하는 경우 서로 다른 Min_Rate에 대하여 군집 수가 변화함을 확인 할 수 있었다. Min_Rate의 값이

클 수록 군집수가 줄어들을 알 수 있다. 객체의 테두리 영역은 객체 영역의 안쪽보다 객체 정보가 상대적으로 적게 포함되어 있으므로 *Min_Rate* 값을 증가시킬수록 객체 테두리 영역은 군집화 대상에서 제외될 것이므로 정확도를 높이고자 한다면 객체의 테두리 영역을 고려해 *Min_Rate* 값을 적절히 조절해야 할 필요가 있다.

4.2 저장 공간 사용에 대한 효율성

[그림 11]은 BSCM 알고리즘의 공간 효율성에 대한 그래프이다.

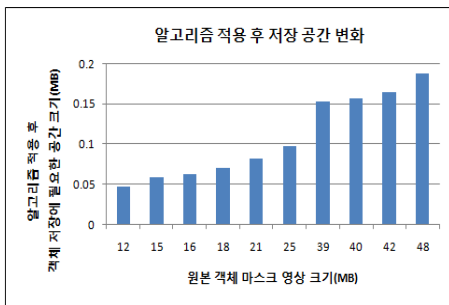


그림 11. 알고리즘 적용 후 저장 공간 변화

실험에서 적용한 *FI*가 10이므로 원본 마스크 영상에 비하여 BSCM 방법에서 추출되는 마스크 정보는 1/10의 크기로 줄어든다. 또한 가로 20 × 세로 20의 영역을 하나의 식별 번호로 나타낼 수 있으므로 분할된 부분 영역 정보가 1/40의 크기로 줄어든다. 그러므로 BSCM 방법은 저장 공간 사용의 효율성 측면에서 볼 때 기존의 마스크 영상 저장 방식에 비하여 효율적임을 알 수 있다.

V. 결론 및 향후 연구 방향

본 논문에서는 대화형 멀티미디어 방송의 상호 운용성을 높이기 위한 객체 저장 및 인식 방법으로 BSCM 방법을 제안하였다. BSCM 방법은 객체의 마스크 영상에서 해당 객체 영역을 찾아 균등 분할하고, 분할된 영역에 대하여 객체 정보 포함 비율을 계산하여 마스크

영상 저장 영역을 클러스터링 한다. 또한 BSCM 방법은 마스크 영상에서 실제 객체가 포함된 영역 중 최소 객체정보 포함 비율 이상의 영역만을 선별하여 저장하므로 기존의 마스크 영상 저장 방식에 비하여 디스크 공간 사용이 효율적임을 실험을 통하여 확인하였다. 향후 연구로 한 화면에 다수의 객체가 존재하는 경우 객체 상하 관계를 인식하여 추출하는 방법과 제안한 BSCM 알고리즘을 개선하여 다양한 영상 추출 과정에 적용하는 방법에 관하여 연구하고자 한다.

참고 문헌

- [1] 김진용, *인터넷방송에서의 데이터방송 서비스*, ETRI, 2001.
- [2] 안종구, "한국디지털 TV 방송의 광고효과와 방안에 관한 연구", *광고학연구*, 제13권, 제5호, pp.73-95, 2002.
- [3] <http://www.atsc.org>
- [4] <http://www.dvb.org>
- [5] <http://www.opentv.com>
- [6] P. McLauchlan, D. Beymer, B. Coifman, and J. Mali, "A Real-time Computer Vision System for Measuring Traffic Parameters," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp.495-501, 1997.
- [7] K. Kanatani, "Camera Rotation invariance of image characteristics," *Computer Vision, Graphics and Image Processing*, Vol.39, No.3, pp.328-354. 1987.
- [8] L. M. Po and W. C. Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.6, pp.313-317, 1996.
- [9] X. Jing and L. P. Chau, "An Efficient Three-Step Search Algorithm for Fast Block

