

論文2002-39SP-1-3

정지영상/동영상에서 non-rigid object의 효율적인 영역 분할 방식에 관한 연구

(Effective segmentation of non-rigid object in a still
picture and video sequences)

李仁宰*, 金龍虎**, 金重圭**, 李明浩***, 安致得*

(Injae Lee, Yongho Kim, Joongkyu Kim, Myoung Ho Lee, and Chieteuk Ahn)

요 약

멀티미디어 표준안으로 제안된 MPEG-4는 객체기반 부호화 방식으로서, 객체를 효율적으로 분할하는 것은 MPEG-4에 있어 중요한 관건이다. 지금까지 이 분야에 대한 연구는 주로 rigid object를 대상으로 하였으나, 본 논문에서는 non-rigid object, 특히 구름이나 연기와 같은 non-rigid object를 대상으로 하여 효율적인 영역 분할 방식을 연구하였다. Non-rigid object는 모양이나 크기가 일정치 않으며 시간에 따라 형태도 변형되므로 정확히 분할해내는 것은 쉽지 않다. 따라서 본 논문에서는 이를 효율적으로 극복하기 위해 정지 영상에서는 watershed 알고리즘을 사용하여 non-rigid object를 분할해 주었다. 그리고 동영상에서는 intra-frame segmentation과 inter-frame segmentation을 통해 연속되는 프레임 내 관심 있는 객체의 경계선을 자동으로 추출해 주었다. 이 때 영상 내 경계 정보와 영역 정보 각각에 가중치를 두어 원하는 객체를 보다 정확히 추출해 주었다.

Abstract

The new MPEG-4 video coding standard enables content-based functionalities. Image segmentation is an indispensable process for it. This paper addresses an effective segmentation of non-rigid objects. Non-rigid objects are deformable objects with fuzzy, blurred and indefinite boundaries. So it is difficult to segment deformable objects precisely. In order to solve this problem, we propose an effective segmentation of non-rigid objects using watershed algorithms in still pictures. And we propose an automatic segmentation through intra-frame and inter-frame segmentation process in video sequences. Automatic segmentation preforms boundary-based and region-based segmentation to extract precise object boundaries.

* 正會員, 韓國電子通信研究院 無線放送研究所,
(Radio & Broadcasting Laboratory Electronics and
Telecommunications Research Institute)

** 正會員, 成均館大學校 電氣電子및컴퓨터工學部,
(Department of Electrical & Computer Engineering
Sungkyunkwan University)

*** 正會員, 關東大學校 情報技術工學部
(Dept. of Information and Communication Engineering
Devision of Information Technology Engineering
Kwandong University)

※ 본 연구는 정보통신부의 “객체기반 AV 편집/제작
도구 개발” 사업의 일환으로 수행된 연구 결과입니다.
接受日字:2001年1月17日, 수정완료일:2001年11月20日

I. 서 론

디지털 비디오 신호 처리에 관련된 영역 분할 방식은 꾸준히 주요 연구 분야로서 관심의 대상이 되어왔다. 이 분야에 대한 효율적인 분석과 표현은 나날이 새로운 기술이 적용되고 있으며, 오늘날 많은 멀티미디어 적용 분야에 있어 없어서는 안될 분야가 되었다. 지금까지 이 분야에 대한 연구는 주로 rigid object를 대상으로 하였으나, 최근에 위성 영상, 적외선 영상, 감마선 영상 등을 이용, 구름이나 기포와 같은 non-rigid object에 대한 연구의 필요성도 서서히 제기되고 있다.

본 논문에서는 구름이나 연기와 같은 non-rigid object의 효율적인 영역 분할 방식에 대해 기술하였다. Non-rigid object란 일정한 모양없이 시간에 따라 형태가 변하는 물체로서 시간에 따른 형태 변화를 예측하기 어려운 물체를 말한다. 본 논문에서 연구 대상으로 하는 non-rigid object의 경우 퍼지한 패턴을 가지고 있고, 원하는 객체와 배경 사이에 밝기차가 전반적으로 작으며, 일정한 크기나 모양이 없으면서 시간에 따라 형태가 변화하므로 rigid object와 동일한 방법을 그대로 적용해 줄 경우 잘못된 결과를 얻을 수 있다. 따라서 non-rigid object에 적합한 방법을 적용해 주어야 한다.

본 논문에서는 먼저 공간적인 정보를 이용한 watershed 알고리즘을 적용함으로써 non-rigid object를 효율적으로 영역 분할해 내는 방법에 대해 언급하였다. 그리고 이 결과를 이용해 첫 번째 프레임에서 관심 있는 객체의 경계선을 선택, 추출해주는 intra-frame segmentation과 이를 기반으로 다음 프레임에서 선택되어진 객체의 경계선을 자동으로 추출해 주는 inter-frame segmentation에 대해 기술하였다.

II. 정지영상에서의 non-rigid object segmentation

영상의 효율적인 영역 분할은 영상 분석에 있어 필수적인 과정으로, 특히 영상의 가시화, 인식, 압축 등 많은 방법들이 영역 분할 결과에 크게 의존한다. 따라서 본 연구에서는 정지 영상 내에 존재하는 non-rigid object를 효과적으로 영역 분할해 주기 위하여 watershed 알고리즘을 적용해 주었다. Watershed 알고리즘은 경사 영상을 입력 영상으로 하여 영상 내 그레이 레벨 중 최소값을 찾아낸 후 이 값을 증가시켜가면서 영역들을 분할해 내는 방법이다^[1,2]. 이 방법은 seeded region growing 알고리즘^[3]과 같이 원영상으로부터 영역을 병합해 내는 방법에서 적절치 못한 임계값을 적용했을 때 흔히 발생할 수 있는 under-segmentation 현상을 막아줄 수 있다. 특히 구름이나 연기와 같은 non-rigid object는 다른 영상에 비하여 경계선 주위에서의 밝기차가 작으므로 watershed 알고리즘을 사용해 줄으로써 객체의 경계선을 보다 정확하게 파악해 줄 수 있다. 하지만 이 알고리즘은 객체가

많은 영역으로 분할되는 과분할 현상이 발생한다. 따라서 전처리 과정과 후처리 과정을 통해 이를 극복하였다. 이 과정을 간단한 블록도로 구성해 보면 그림 1과 같다.

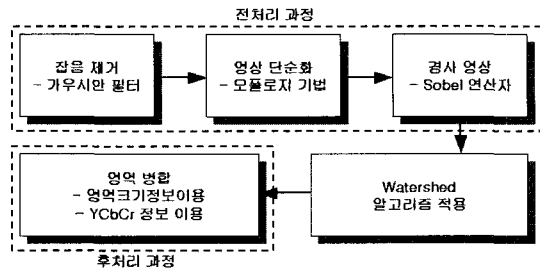


그림 1. 공간적인 정보를 이용한 영역분할 알고리즘의 블록도

Fig. 1. Block diagram of the spatial segmentation process.

1. Watershed 알고리즘

Watershed 알고리즘은 영상의 그레이 레벨값을 높이로 생각하여 최소 영역에서부터 차츰 물을 채워나가듯이 영역을 합쳐 나가는데 최종적으로 하나의 테두리로 둘러싸여진 부분을 균일 영역으로 판단한다. 이때 그림 2에서 보여지듯이 두 개 이상의 영역이 인접할 경우 watershed 라인으로서 영역 사이의 경계를 구분해 준다.

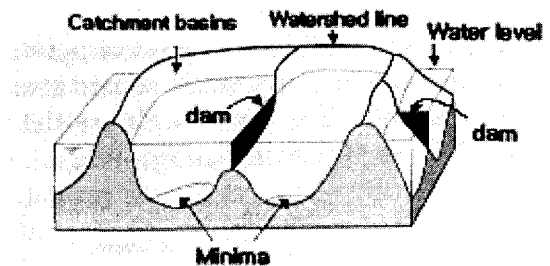


그림 2. Watershed 알고리즘의 개념도

Fig. 2. Minima, catchment basins, watersheds and dams.

Watershed 알고리즘은 입력 영상으로 원영상을 그대로 사용하는 것이 아니라 원영상에 대한 경사 영상을 이용한다. 따라서 이 알고리즘은 경사 영상에 대한 에지 정보와 영역 정보를 적절히 이용하여 폐곡선으로 분할된 영역들을 얻을 수 있기 때문에 주요한 결정도 구로써 많이 이용되어지고 있다.

Algorithm : Watershed Algorithm

i) 초기화 단계 : 경사 영상 내 가장 작은 그레이 레벨값을 찾아 최소지역을 설정하고 레이블링. 이 때 서로 연결되어 있는 픽셀에는 동일한 레이블 할당, 떨어져 있는 최소지역들에는 각각 다른 레이블 할당.

ii) 침수 단계 : 그레이 레벨값을 증가시켜 가면서 이미 레이블된 픽셀, 즉 기존 지역에 접해 있는 경우에는 기존 지역의 레이블 할당, 새로운 극부 최소 지역 존재시 새로운 레이블 할당.

2. 전처리 과정

전처리 과정의 목적은 watershed 알고리즘을 적용했을 때 발생하는 과분할 현상을 줄이는데 있다. 또한 분할되어진 영역의 개수가 많을수록 watershed 알고리즘의 적용 후 행해지는 후처리 과정의 처리 시간도 길어지므로 watershed 알고리즘을 적용해 주기에 앞서 영역을 단순화시켜줄 필요가 있다. 그리고 보다 효율적인 watershed 결과를 얻기 위해서는 에지를 부각시킬 필요가 있다. 따라서 전처리는 그림1에서 보여지는 것처럼 세 가지 과정으로 나누어 수행된다.

연구 대상인 구름이나 연기의 경우 퍼지한 패턴으로 인해 내부에서도 밝기차가 존재하므로 watershed 알고리즘을 원영상에 그대로 적용하였을 때 원하지 않는 미세 영역까지 분할 되어지는 문제점이 나타난다. 따라서 먼저 필터링 과정을 통해 잡음을 줄여주고, 모폴로지를 이용하여 영상을 단순화 시켜줌으로써 대상이 되는 객체의 에지는 보존해 주면서 불필요한 영역은 제거시켜 줄 수 있다.

영상의 잡음 성분을 줄여 주기 위해 일반적으로 linear filter, median filter, Gaussian 등 여러 가지 방법들이 사용되고 있다^[4].

영상에서 가능한 에지의 손실 없이 원영상을 단순화시켜 주기 위해서는 수리적 모폴로지 기법이 많이 사용된다. 이 기법의 기본 연산자로는 팽창(dilation), 침식(erosion) 연산이 있고, 더 나아가 제거(opening), 채움(closing) 연산도 보편적으로 많이 사용되고 있다^[5]. 여기서 제거 연산자는 구성 성분을 이용하여 주위에 비해 밝기값이 큰 미세한 부분들을 제거해 주는 반면 채움 연산자는 밝기값이 작은, 주위에 비해 어두운 미세한 부분들을 제거시킨다. 즉 이 연산들을 통해 미세한 부분에 영향을 주어 미세 영역에 대한 과분할 현상을 줄여줄 수 있다.

또한 일반적으로 그레이 레벨이 빨리 변하는 지점들이 영역들의 에지 부분과 일치하므로 watershed 알고리즘의 입력 영상으로 경사 영상을 사용하면 영역 분할을 하는데 있어 보다 효율적이다. 따라서 첫 번째 단계에 의해 단순화된 영상에 대한 경사 영상을 구해 이것을 watershed 알고리즘에 적용시키기 위한 입력 영상으로 사용해준다. 일반적으로 Sobel, Prewitt, Gaussian, Gaussian derivative 등을 이용해 경사 영상을 구해줄 수 있다^[6].

3. 후처리 과정

후처리 과정에서는 컬러 정보와 통계적인 분석을 적용한 영역 병합을 통해 영역들을 최적의 상태로 분할하고자 하였다. 영역을 병합시키는 최종 목적은 하나의 객체에 하나의 레이블을 할당시키는 데 있다. 이 과정에서 판단 기준이 되는 각 영역의 유사성을 결정하기 위해 영역들의 크기, 평균, 분산 등 통계적인 방법들을 이용한다.

본 논문에서는 HSWO(Hierarchical stepwise optimization) 알고리즘을 응용하여 영역을 병합시켜 주었다. HSWO 알고리즘은 stepwise criterion C_{ij} 의 값이 가장 작은 두 개의 영역을 선택해 병합시키는 최적화 기법이다^[7,8]. 원영상 $f(x, y)$ 에서 분할된 i 번째 영역에 대해 평균값은 μ_i , 크기값은 N_i 라 할 때, 각 영역간의 유사성을 판단하기 위한 조건인 C_{ij} 는 식 (1)과 같이 정의된다.

$$C_{i,j} = \frac{N_i \cdot N_j}{N_i + N_j} (\mu_i - \mu_j)^2 \quad (1)$$

구름과 같은 non-rigid object는 퍼지한 패턴으로서 객체 내부의 밝기차로 인해 크기가 작은 영역들이 여전히 존재한다. 따라서 구름의 패턴 특성으로 인해 발생하는 이런 크기가 작은 영역들에 우선 순위를 두어 병합 시켜줄 필요가 있다. 즉, 영역의 평균값 외에 크기값도 반영시켜주어야 한다. 또한 일반적으로 영역을 병합시키기 위한 조건으로서 밝기값을 사용해 설정해 주는데, RGB 혹은 YCbCr 등 다양한 컬러 정보를 조건 설정시 적절하게 첨가시켜 주면 보다 효율적인 영역 병합이 가능하다. 따라서 유사성을 판단할 때 이웃하는 영역간의 RGB나 YCbCr 등의 평균값의 차이를 이용해 보다 효율적인 결과를 얻고자 하여 식 (2)를 사용하였다.

$$C_{i,j} = \frac{N_i \cdot N_j}{N_i + N_j} [w_Y (\mu_i^Y - \mu_j^Y) + w_{Cb} (\mu_i^{Cb} - \mu_j^{Cb}) + w_{Cr} (\mu_i^{Cr} - \mu_j^{Cr})] \quad (2)$$

여기서 $\mu_i^Y, \mu_i^{Cb}, \mu_i^{Cr}$ 은 분할 영역 S_i 의 Y, Cb, Cr의 각 평균값이며, w_Y, w_{Cb}, w_{Cr} 은 Y, Cb, Cr에 대한 가중치를 나타내는 것으로, Y정보가 Cb, Cr 정보에 비해 중요하므로 Y 정보에 대한 가중치를 더 크게 준다. 이 식을 이용하여 이웃하는 모든 영역들에 대해 C_{ij} 를 계산한 후 이 값이 가장 작은 것들을 찾아 해당되는 두 영역을 병합시켜준다. 이 과정을 반복해 나가면서 객체를 최적의 상태로 찾아준다.

III. 동영상에서 non-rigid object의 user-assisted segmentation

User-assisted segmentation^[9]은 첫 번째 프레임에 대해 수동으로 관심 있는 객체의 경계 영역을 선택해 준 후 연속되는 프레임에서 자동으로 객체의 경계선을 추출해내는 데 목적이 있다. 이를 수행하기 위한 과정은 크게 intra-frame segmentation과 inter-frame segmentation 두 가지 단계로 나누어 줄 수 있다.

1. Intra-frame segmentation

Intra-frame segmentation은 첫 번째 프레임에서 GUI를 통해 수동으로 원하는 객체의 경계 영역을 선택해 준 후 실제 경계선을 추출해 내는 과정이다. 이는 초기 단계에서 보다 정확한 경계선을 찾아내 줌으로써 이 정보를 기반으로 연속되는 프레임에 존재하는 객체의 경계선을 정확히 찾아내는데 그 목적이 있다. 이 과정을 수행해 주기 위해 첫 번째 프레임내에서 GUI(Graphic User Interface)를 통해 원하는 객체의 경계 영역, 즉 UR(Uncertain Region)을 선택하고 이를 포함하는 ROI(Region Of Interest)를 설정해 준다. 이때 ROI 이외의 영역은 단순화를 시켜준다. 이 영상을 입력으로 하여 그림 1에서 제시되었던, 공간적인 정보를 이용한 영역 분할 알고리즘을 적용하여 각 영역에 따른 local labeling을 설정해 준다. 그리고 UR을 기반으로 global labeling을 부여해 준다. 이 두 가지 정보를 이용해 객체의 경계선을 추출해 준다. 이를 간략한 흐름도로 나타내면 그림 3과 같다.

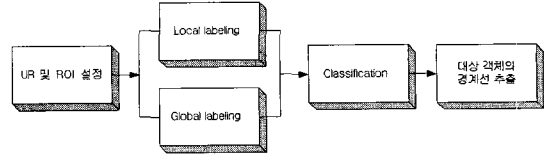


그림 3. Intra-frame segmentation의 블록도
Fig. 3. Block diagram of intra-frame segmentation process for non-rigid object

1) Global labeling 및 local labeling

Global labeling은 GUI를 통해 원하는 객체의 경계선을 일정한 두께로 사용자가 선택함으로써 수행된다. 이때 선택되어진 영역을 UR이라 정의하며 해당 영역에 '0'으로 label을 부여해준다. 나머지 객체의 내부에 해당되는 영역은 '1'로, 외부에 해당되는 영역은 '2'로 label을 할당한다. 이를 위해 다음과 같은 알고리즘을 사용한다.

- i) 모든 영역을 TRUE로 설정. (값이 TRUE인 pixel의 집합을 S_T 라고 함)
- ii) UR을 FALSE로 설정. (값이 FALSE인 pixel의 집합을 S_F 라고 함)
- iii) Background 영역에 시작점을 잡고, FALSE로 설정.
- iv) 시작점에서부터 4-neighbor로 확장해가면서 TRUE를 FALSE로 변경.
- v) S_T 에 label '1' 할당, $S_F \cap \overline{UR}$ 에 label '2' 할당, UR에 label '0' 할당.

본 연구에서 얻고자 하는 결과가 선택된 객체의 경계선이고, UR에서 많이 벗어난 곳의 정보는 필요없게 되므로 관심 영역을 축소하였다. 따라서 UR에서 ± 10 범위 이내의 좌표에 대해서만 모든 처리를 해 주었다.

Global labeling의 알고리즘 iii)에서 시작점은 $(x_{min} - 1, y_{min} - 1)$ 로 하였으며, 이를 위해 영상의 경계를 한 픽셀 더 크게 하였다. 또한, 처리 속도를 더 빠르게 하기 위해서 global labeling 결과가 0, 1인 영역의 최대값, 최소값을 구한 후, 영역 2에서 이 값을 이용하여 임계화해 줌으로써 보다 단순화 시켰다.

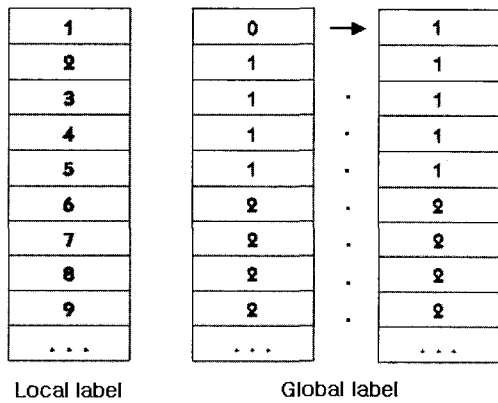
Local labeling은 그림1에서 제시했던 공간적인 정보를 이용한 알고리즘을 적용하여 영역 분할을 행함으로써

각 영역에 따른 label 값을 구해주었다. 이 두 가지 레이블링 결과를 이용하여 UR내에서 실제 경계선을 분류한다.

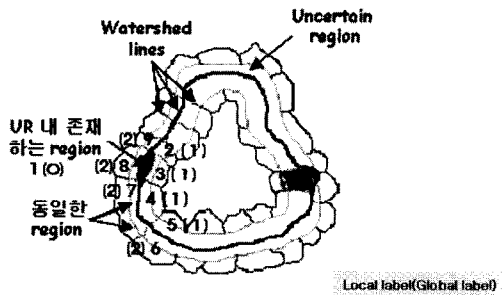
2) Classification

① 영역 기반의 classification

Global labeling과 local labeling이 끝난 후 최대 local label만큼의 크기를 가진 배열을 만들고, '0'으로 초기화시킨다. 그림 4(a)처럼 UR을 제외한 영역을 탐색해가면서 해당 픽셀의 local label을 방번호로 가지는 배열값에 해당 픽셀의 global label값을 넣어 준다. 즉, 이것은 watershed에 의해 분할된 영역들이 객체의 내부에 존재하는지 외부에 존재하는지 판단해주는 것이다.



(a)



(b)

그림 4. 영역 기반에서의 경계선 추출
Fig. 4. Region-based boundary extraction.

그리고 global label이 '0'인 영역들에 대해서는 이 영역들이 객체의 내부와 유사한지 외부와 유사한지 판단해 주어야 한다. 즉, 그림 4(b)에 표시된 두 영역처럼 분할된 영역이 UR 내에 존재할 경우나 UR을 벗어날 경우가 이에 해당된다. 이 경우에는 해당 영역에 이웃하는 영역들에 대해 식 (2)에 해당되는 C_{ij} 를 계산해 준

다. 이것은 해당 영역과 유사한 영역을 찾아줄 때 구름의 퍼지한 패턴 특성에 따라 컬러 정보와 영역의 크기 정보를 고려해주는 것이다. 이 C_{ij} 값이 가장 작은 것을 찾아 이에 해당되는 이웃 영역의 global label값을 현재 영역의 global label값으로 할당해 준다.

이렇게 각 영역들에 global label을 부여해 준 후 UR을 탐색해 나가면서 watershed 라인에 해당하는 픽셀이 있으면 주위 영역의 global label값을 검사하여 '1'과 '2' 사이에 있을 경우에는 그대로 두고, 주위값들이 모두 '1'이거나 '2'일 경우 watershed 라인을 제거 시켜 준다. 그림 4(b)의 굵은 라인이 이에 해당하는 결과이다. 즉, 이것은 watershed 라인이 객체의 내부 영역과 외부 영역 사이에 존재할 경우 그대로 남겨두고 내부에 혹은 외부에 존재하는 경우에는 제거시켜 주는 과정으로서, 이를 통해 원하는 객체의 경계선을 찾아준다.

② 경계 기반의 classification

UR 내에서의 정확한 watershed 라인을 찾기 위해서 본 연구에서는 객체의 경계선 정보를 기반으로 하였다. 이를 위해서는 그림 5에서 보여지는 것처럼 먼저 UR 내의 watershed 라인들을 최종 경계선이 될 확률의 높고 낮음에 따라 CANDI, CANDI2, SPRIG의 세 가지 종류로 나눈다. CANDI는 확실히 최종 경계가 되는 watershed 라인이고, CANDI2는 최종 경계선이 될 확률이 높은 것, 그리고 SPRIG는 확률이 거의 없는 경우이다. 앞에서 설정한 global labeling 결과가 이번 단계에서 사용되게 된다. CANDI는 주위에 '1', '2'의 레이블이 모두 있는 것이고, CANDI2는 주위에 '0'이 있는 것, 그리고 '1' 또는 '2'만 있는 경우가 SPRIG에 해당된다.

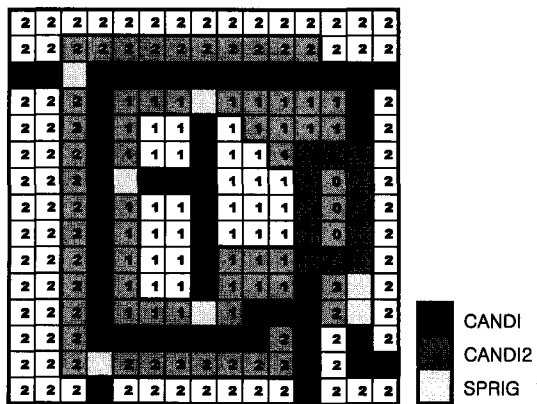


그림 5. 경계 기반에서의 경계선 추출
Fig. 5. Edge-based boundary classification.

여기에서 CANDI는 확실히 경계선이 되고, SPRIG는 거의 경계선이 될 확률이 없으며, CANDI2는 그 중 어느 부분만이 최종 경계선이 됨을 알 수 있을 것이다. 따라서 CANDI2에서 어느 부분이 최종 경계선이 될 것 인지를 결정하여야 한다. 최종 경계선을 찾기 위해 본 연구에서는 두 가지 factor를 사용하였다.

첫 번째는 여러 개의 선들 중에서 어떤 선이 경계선이 될 확률이 높은가 하는 것이다. 이를 위해 본 연구에서는 경사 정보를 이용하였다. 경사값이 클수록 경계선이 될 확률이 높다. 여기에서는 경사 정보로 일반적인 Sobel edge detector의 결과를 사용하였다.

두 번째는 모양에 관한 정보이다. 영역을 이용한 분류에서는 영역들만 분리해 주면 이 영역을 둘러싼 확실한 경계선이 나오게 된다. 그러나 경계선을 이용하면 순서대로 검색하지 않을 경우 중간에서 끊길 수가 있다. 이를 위해 순서대로 검색을 하더라도 시작점과 끝점이 연결되지 않거나 경계선의 모양이 원하는 형태로 나오지 않을 수도 있다. 따라서 각 픽셀에 방향벡터를 주어서 방향벡터의 방향에 대해 더 큰 값을 주도록 하였다. Boundary following을 통해 안과 밖 두 개의 경계선을 찾고, 이를 순서대로 정렬시켜준 후, 정렬된 점들에서 i 번째 점과 $i+1$ 번째 점의 차를 이용하여 i 번째 점의 방향벡터를 설정하였다. 내부 벡터는 외부 벡터의 평균으로 하였으며, 이에 대한 그림 6에 나와 있다.

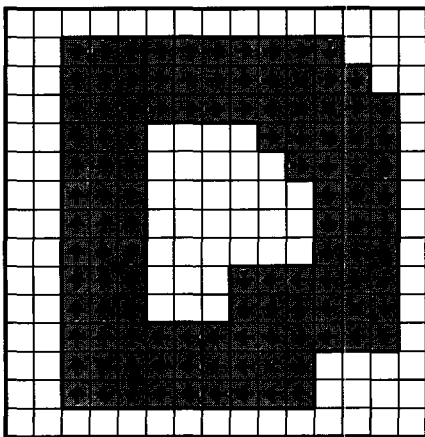


그림 6. 방향 벡터 설정

Fig. 6. Creation process of directional vector.

최종 경계선을 선택하는 알고리즘의 순서도는 그림 7과 같다. 여기에서 EDGE는 최종 경계선을 의미하고, 이웃 픽셀을 찾는 과정에서 시작점이 나오면 종료하게

된다.

경계선 탐색에 있어서 시작점 설정은 매우 중요하다. 시작점이 잘못 설정될 경우에 폐곡선을 만들 수 없거나 잘못된 폐곡선을 만들 수 있기 때문이다. 따라서 EDGE가 확실한 부분이 시작점이 되어야 한다. 그러므로 여기에서는 시작점을 CANDI중의 하나로 선택하였다. 만약 CANDI가 없다면 CANDI2 중에서 경사 정보가 가장 큰 것으로 하였다. 이 경우에 잘못된 결과를 낼 수도 있을 것이다.

CANDI가 이웃 픽셀 중에 있으면 무조건 그것이 EDGE가 되게 된다. 그렇지 않으면 CANDI2가 있는지를 검색하게 되는데, 이 경우에 주위에 2개 이상의 CANDI2가 있을 수 있으므로 이 중의 하나를 선택하여야 한다. 이를 위해 앞에서 설정하였던 방향 벡터와 경사 정보를 이용하였다. 기본적으로 경사 정보가 가장 큰 곳으로 이동하도록 하였으며, 만약 방향 벡터의 방향과 일치할 경우에 경사 정보에 10배를 해 주었다. 그리고 방향 벡터의 방향과 $\pm 90^\circ$ 이내의 차이가 나면 4배를 해 주었고, 그렇지 않으면 1배를 사용하였다.

기본적으로 이러한 방법을 이용하여 탐색을 하게 되는데, 여기에서 몇 가지 오류에 대한 방지책이 필요하다. 첫 번째 오류는 EDGE가 부분적으로 폐곡선이 되어 버리는 경우이다. 이 경우, 주위에 CANDI나 CANDI2가 없게되므로 더 이상 탐색을 할 수가 없게 된다. 이렇게 되면 BackTrack을 함으로써 다시 원래 위치로 되돌아 갈 수 있도록 하였다. 두 번째 오류는 UR을 잘못 지정함으로써 생기는 오류이다. 이 경우에는 UR의 바깥쪽에서 탐색을 해 주어야 한다.

영역을 이용해도 마찬가지로, 경계를 이용해도 마찬가지로 watershed 정보를 이용한 탐색의 최대 약점은 watershed 라인이 UR내에 오지 않는 경우가 될 것이다. 따라서 이 경우에 밖에서 EDGE를 찾아줄 수 있는 기능이 필요하게 된다. Outer search는 이러한 경우에 바깥 부분의 watershed line을 찾아주게 되며, 최종적으로 다시 UR내에 들어와야 한다. 바깥에서 쪼이거나 다시 UR내로 들어오지 않는 것을 방지하기 위해서 방향 벡터를 이용하였다. Outer search를 하기 전의 점에서 방향 벡터의 방향으로 가다보면 결국 다시 돌아와야 할 점을 찾을 수 있다. 따라서 이 점을 기준으로 chamfering 알고리즘^[4]을 사용하였고, chamfering 값이 작은 곳으로 이동을 하게 되면 이 점으로 돌아올 수 있게 된다.

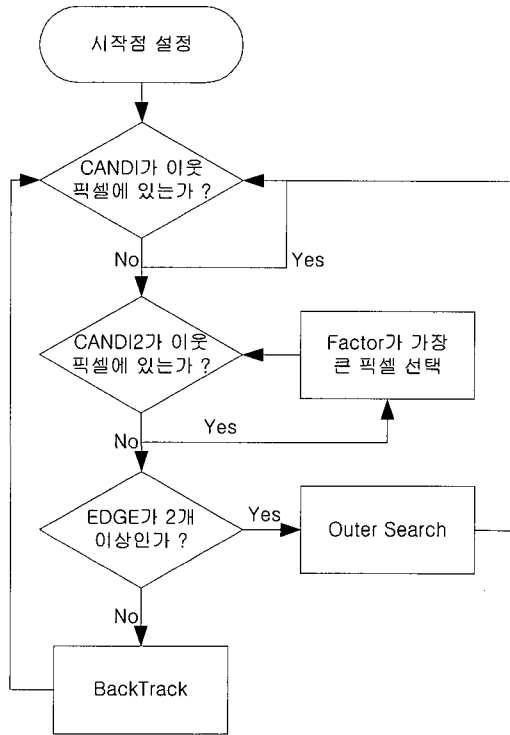


그림 7. 경계선 탐색 알고리즘의 흐름도
Fig. 7. Flow chart of boundary searching algorithm.

2. Inter-frame segmentation

Inter-frame segmentation은 첫번째 프레임에 대해 intra-frame segmentation으로 찾아낸 객체의 경계선 정보를 이용하여 자동으로 현재 프레임 내 객체의 경계선을 추출해 내는데 그 목적이 있다. 첫 프레임으로부터 얻어낸 경계선에 대한 정보를 이용하여 연속되는 다음 프레임에 대해 BM(block matching) 알고리즘을 적용함으로써 매칭되는 점들을 찾아준다. 이 매칭되는 점들을 중심으로 해당 프레임내 객체의 경계선이 존재할 가능성이 있는 UR을 구해준다. 그리고 이를 기반으로 intra-frame segmentation을 다시 한번 수행함으로써 보다 정확한 non-rigid object의 경계선을 얻게 된다.

1) BM을 통한 움직임 추정

객체의 경계 부분에 대한 움직임 추정(motion estimation)을 구하기 위해 BM 알고리즘을 적용해 주었다. BM은 현재 프레임을 일정한 크기의 블록들로 나누어 준 후 이 블록들 중 이전 프레임에서 설정된 블록과 가장 유사한 블록을 찾아주는 알고리즘이다^[10,11]. 본 연구에서는 좀더 효율성을 높이기 위해 full search로 매칭되는 점들을 찾아주는 방법보다 앞서 얻어진

객체의 경계선에 해당하는 각 점들을 중심으로 블록을 설정해 주고, 현재 프레임의 특정 탐색 영역 내에서 매칭 함수를 이용하여 해당 블록을 찾아주었다.

본 연구에서는 매칭시키기 위한 조건으로 일반적으로 많이 사용되는 MAD를 이용하였다. 이를 식으로 표현하면 식 (3), (4)와 같다.

$$MAD(u, v) = \sum_{i=-N/2}^{N/2} \sum_{j=-N/2}^{N/2} |f^t(i+u, j+v) - f^{t-1}(i, j)| \quad (3)$$

$$MV = (u, v) |_{\min MAD(u, v)}, \quad -p \leq u, v \leq p \quad (4)$$

여기서 $f^{t-1}(i, j)$ 는 이전 프레임 내 경계선을 중심으로 한 블록의 픽셀값이고 $f^t(i, j)$ 는 현재 프레임의 조사대상 블록의 픽셀값을 나타낸다. N은 블록의 크기가 되고, $\pm p$ 는 탐색 영역을 의미한다. 후보대상이 되는 블록마다 MAD를 계산하여 이 값이 가장 작은 블록이 해당 블록으로서 선택된다. 하지만 non-rigid object 특성상 정확히 매칭되는 점들을 얻기 힘들므로 이를 기반으로 경계선이 존재할 가능성이 있는 UR과 ROI를 설정해 주어 앞에서 언급되었던 intra-frame segmentation과정을 반복해 준다. 그림 8은 user-assisted segmentation의 전반적인 흐름도로 첫 번째 프레임에 대해서는 intra-frame segmentation을 적용해 주고, 그 이후 프레임에 대해서는 inter-frame과 intra-frame segmentation을 적용해 줌으로써 보다 정확한 경계선을 찾아줄 수 있다.

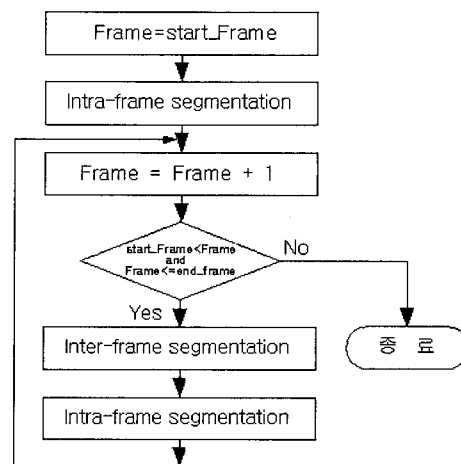


그림 8. User-assisted segmentation의 흐름도
Fig. 8. Flow chart of a user-assisted segmentation method.

IV. 실험 결과 및 고찰

1. 정지 영상에 대한 실험 결과 및 고찰

176×144 크기의 Y, Cb, Cr 구름 정지 영상에 대해 실험하였다. 그림 9(b)는 전처리 과정을 거친 결과 영상으로서 Gaussian filter를 이용하여 잡음을 제거시켜주고 closing을 이용하여 미세한 영역을 줄여준 후 sobel edge detector를 통해 경계 영상을 구해준 결과이다. 그림 9(c)는 그림 9(b)를 입력영상으로 하여 watershed 알고리즘을 적용시켜준 결과로서, 613개의 영역들로 분할되었다. 그림 9(d)는 원영상에 watershed 라인을 흰

색으로 표시한 결과 영상으로, 각 영역 컬러 정보의 평균값을 반영하여 병합해 줄 경우 316개의 영역으로 분할되어지고, 컬러 정보와 영역의 크기 정보를 같이 반영해 줄 경우 156개의 영역으로 감소됨을 알 수 있다. 그림 10의 경우에도 525개의 영역이 71의 영역으로 현저히 감소됨을 알 수 있다. 즉, 같은 객체 임에도 불구하고 객체 내부의 밝기차에 의해 세세히 분할되었던 작은 영역들이 효율적으로 병합되었음을 알 수 있다. 또한 원영상과 결과 영상을 비교하였을 때 중요한 경계 부분의 손실 없이 효율적으로 과분할 현상을 극복하였음을 알 수 있다.

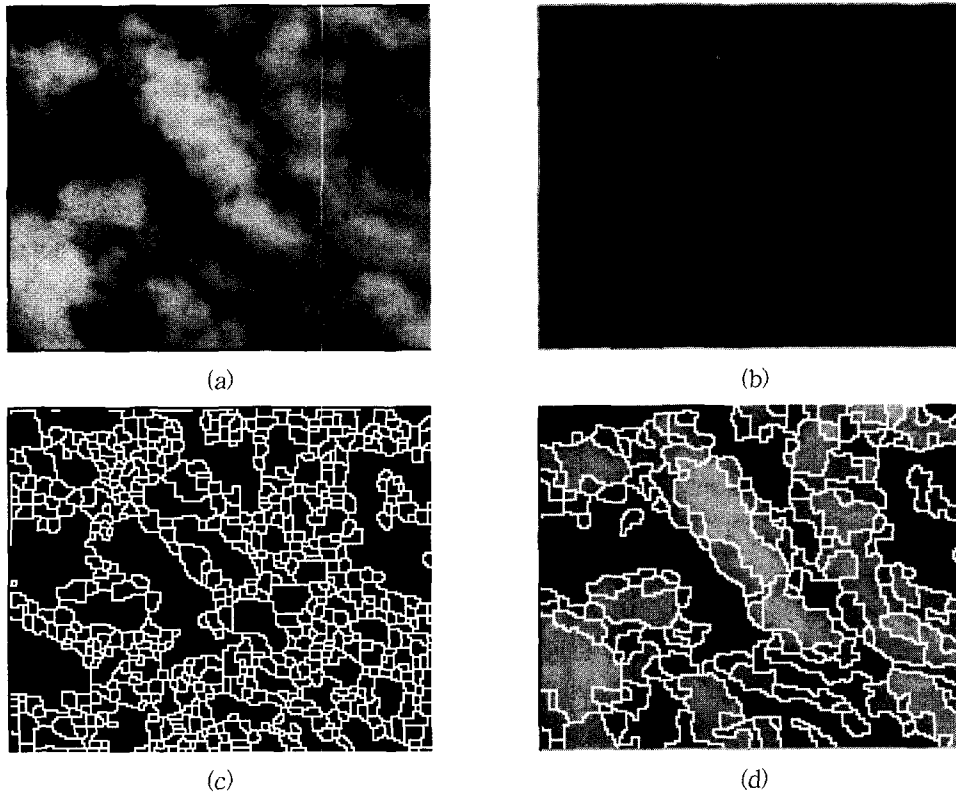


그림 9. 구름 영상 I 에 대한 image segmentation 실험 (a) 구름 원영상 (b) 단순화시킨 영상에 대한 경사 영상 (c) Watershed 알고리즘을 적용한 결과 영상(613 regions) (d) Region merging한 결과 영상(156 regions)
 Fig. 9. Image segmentation of a cloud image I (a) original image (b) result of mathematical morphology and sobel edge detector (c) Result of Watershed segmentation(613 regions) (d) Result of Region merging Overlaid on original image (156 regions).

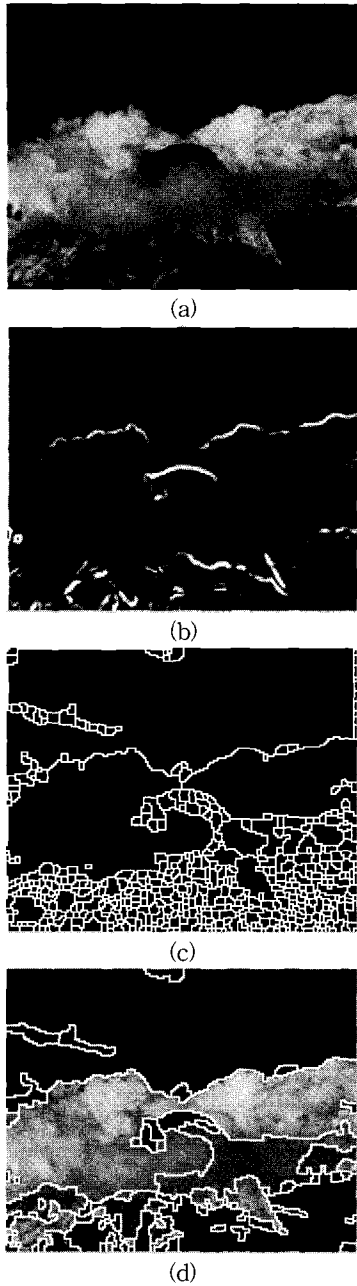
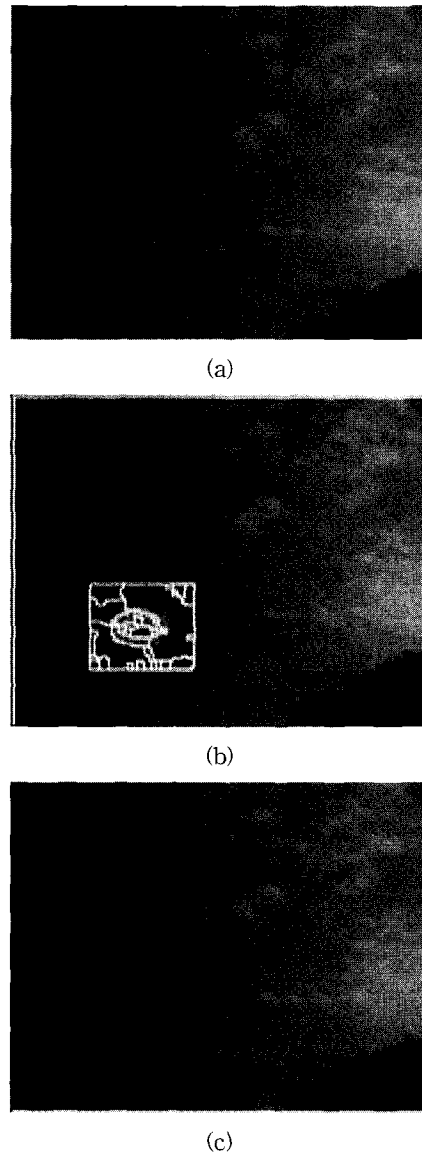


그림 10. 구름 영상Ⅱ에 대한 image segmentation 실험 (a)구름 원영상 (b) 단순화시킨 영상에 대한 경사 영상 (c) Watershed 알고리즘을 적용한 결과 영상(525 regions) (d)Region merging한 결과 영상(71 regions)

Fig. 10. Image segmentation of a cloud image II (a) original image (b) result of mathematical morphology and sobel edge detector (c) Result of Watershed segmentation(525 regions) (d) Result of Region merging Overlaid on original image (71 regions).

2. 동영상에 대한 실험 결과 및 고찰

QCIF format의 구름과 연기 동영상을 대상으로 실험하였다. 그림 11은 구름 영상에 대한 실험 결과로서, 그림 11(a)와 (e)는 각기 첫 번째 프레임과 두 번째 프레임의 원영상이다. 그림 11(b)는 global labeling 결과를 이용, 단순화시킨 영상을 입력으로 하여 그림 1에서 제시한 알고리즘을 적용한 결과이다. 그리고 그림 11(c)와 (d)는 첫 번째 프레임에서 intra-frame segmentation 과정을 거쳐 각각 영역 기반과 경계 기반으로 분할해 준 결과 영상이다. 그림 11(f)는 연속되는 프레



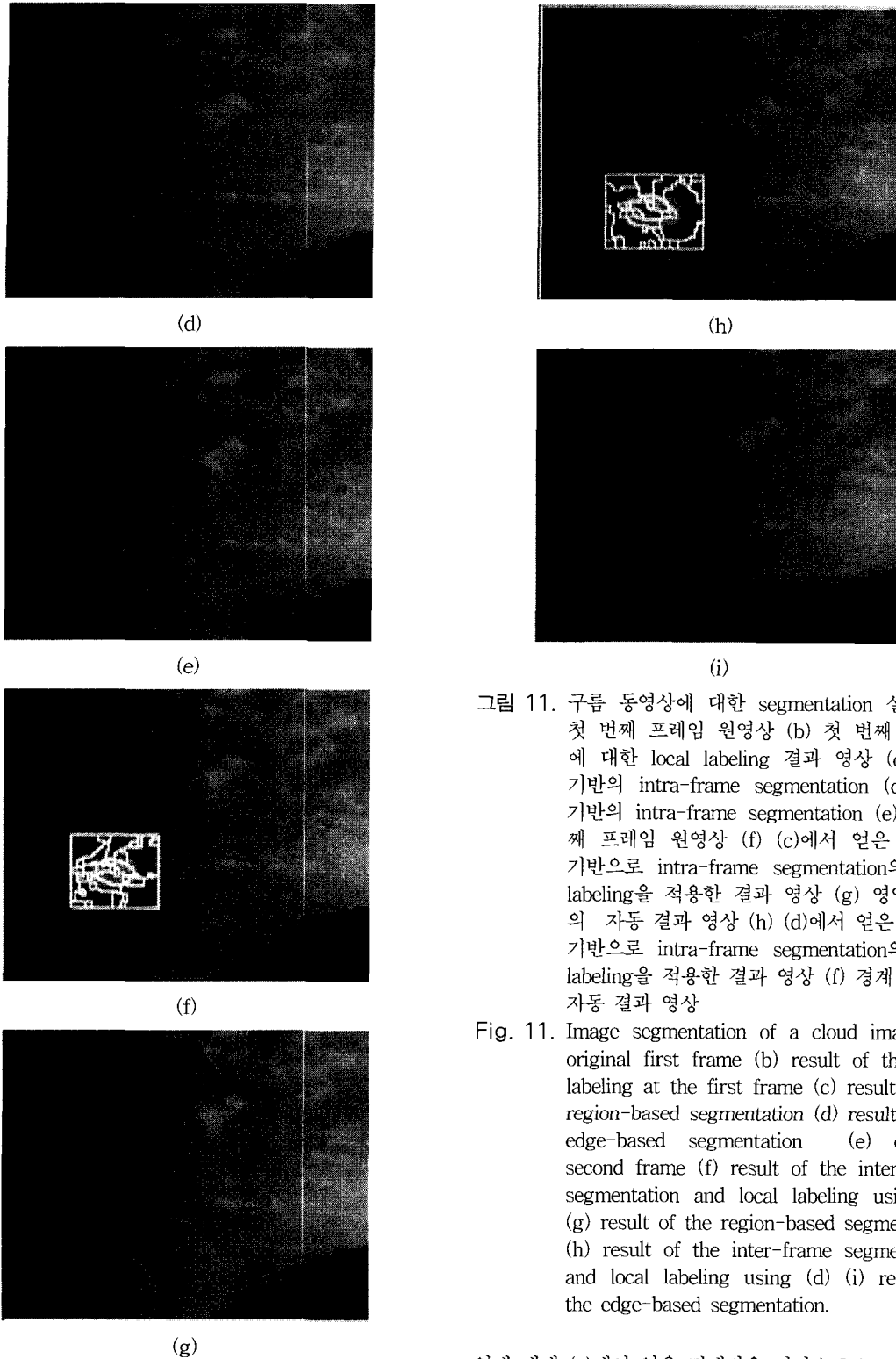
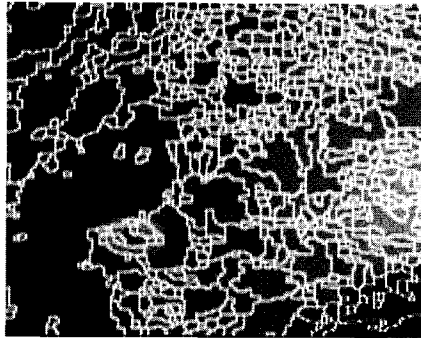


그림 11. 구름 동영상에 대한 segmentation 실험 (a) 첫 번째 프레임 원영상 (b) 첫 번째 프레임에 대한 local labeling 결과 영상 (c) 영역 기반의 intra-frame segmentation (d) 경계 기반의 intra-frame segmentation (e) 두 번째 프레임 원영상 (f) (c)에서 얻은 경계를 기반으로 intra-frame segmentation의 local labeling을 적용한 결과 영상 (g) 영역 기반의 자동 결과 영상 (h) (d)에서 얻은 경계를 기반으로 intra-frame segmentation의 local labeling을 적용한 결과 영상 (f) 경계 기반의 자동 결과 영상

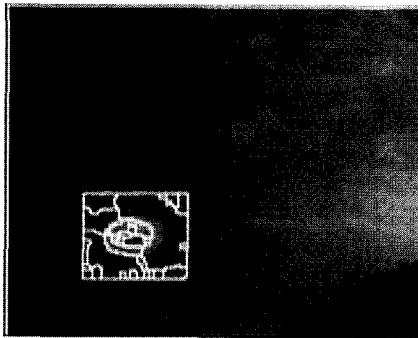
Fig. 11. Image segmentation of a cloud image (a) original first frame (b) result of the local labeling at the first frame (c) result of the region-based segmentation (d) result of the edge-based segmentation (e) original second frame (f) result of the inter-frame segmentation and local labeling using (c) (g) result of the region-based segmentation (h) result of the inter-frame segmentation and local labeling using (d) (i) result of the edge-based segmentation.

임에 대해 (c)에서 얻은 경계값을 기반으로 inter-frame segmentation을 적용한 후 이에 intra-frame segmentation의 local labeling을 수행해 준 결과이다. 그림

11(g)는 (f)에서 실제 경계선을 찾아준 결과이다. 그림 11(h) 또한 (d)를 기반으로 inter-frame segmentation과 intra-frame segmentation의 local labeling을 수행시켜 준 결과이고, (i)는 이렇게 자동 분할해 준 결과이다.



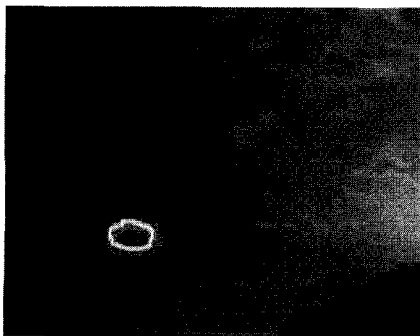
(a)



(b)

그림 12. ROI 설정 결과 비교 (a) ROI를 설정하지 않고 local labeling을 한 결과 (b) ROI를 설정하여 단순화시킨 영상에 대해 local labeling 한 결과

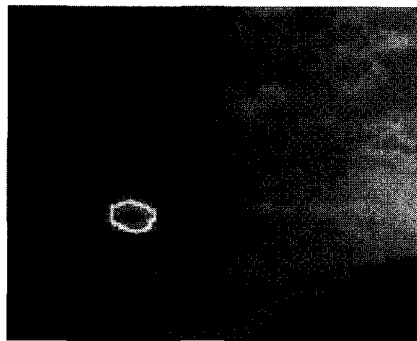
Fig. 12. Comparison between local labeling results (a) nonapplication of ROI (b) application of ROI.



(a)



(b)



(c)

그림 13. Intra-frame segmentation 재적용 비교 (a) 첫 번째 프레임에 대해 intra-frame segmentation을 적용한 결과 영상(b) 다음 프레임에 대해 inter-frame segmentation을 적용한 결과 영상 (c) inter-frame segmentation 값을 기반으로 intra-frame segmentation을 적용한 결과 영상

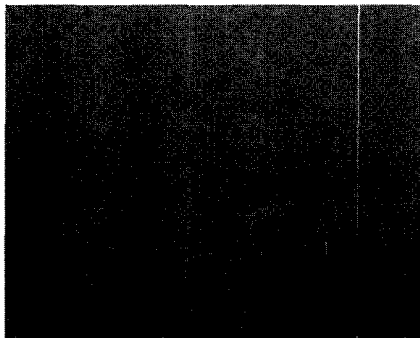
Fig. 13. Comparison about reapplication of intra-frame segmentation (a) application of inter-frame segmentation (b) application of the intra-frame segmentation after inter-frame segmentation.

그림 12(a)는 ROI를 설정하지 않은 원영상에 local labeling을 적용한 결과이고, 그림 12(b)는 ROI를 기반으로 단순화시킨 영상에 local labeling을 적용한 결과이다. 그림 12에서 보여지는 것처럼 ROI를 기반으로 단순화시킨 영상을 사용함으로써 보다 빠르고 효율적인 영역 분할이 가능하다.

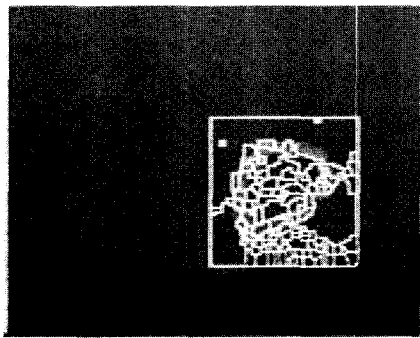
그림 13(a)는 첫 번째 프레임에 intra-frame segmentation을 적용해 얻은 결과이다. 그림 13(b)는 연속된 프레임에 대해 inter-frame segmentation만을 적용해준 결과이고, 그림 13(c)는 inter-frame segmentation을 적용한 후에 이를 기반으로 UR을 설정하고 intra-frame segmentation을 적용하여 얻은 결과 영상이다.

앞서 기술한 것처럼 구름 영상과 같은 non-rigid object는 퍼지한 패턴을 가지고 있고, 원하는 객체와 배경 사이에 밝기차가 전반적으로 작으며, 일정한 크기나 모양이 없으면서 시간에 따라 형태가 변한다는 특성을 가지고 있다. 따라서 이런 특성을 극복하기 위해, 그림 13의 결과 영상에서 보여지는 것처럼 inter-frame segmentation을 적용해 준 후 intra-frame segmentation을 적용해 준 것이 그렇지 않은 것보다 정확한 객체의 경계선을 얻을 수 있다.

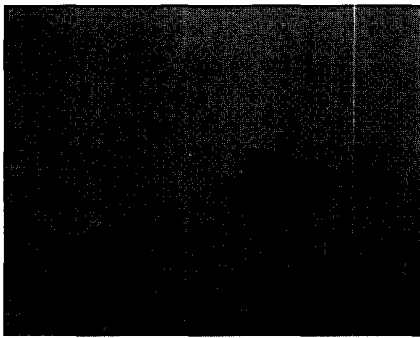
그림 14는 연기 영상에 대한 것으로 첫 번째 프레임과 두 번째 프레임을 통해 분할되어진 객체를 얻은 결과이다. 결과에서 보여지듯이 객체의 형태에 변화가 있어도 그에 따른 객체의 경계선을 찾아주었다.



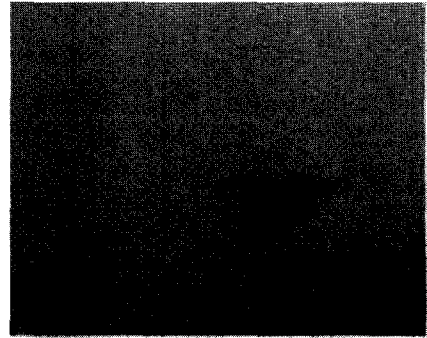
(a)



(b)



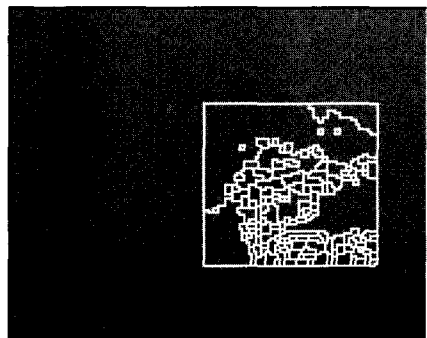
(c)



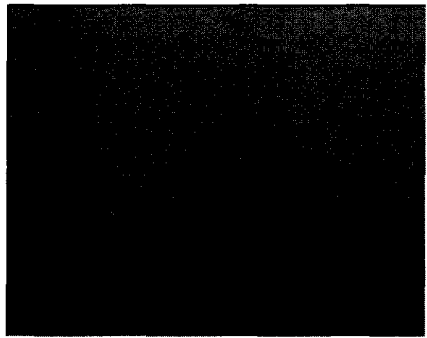
(d)



(e)



(f)



(g)

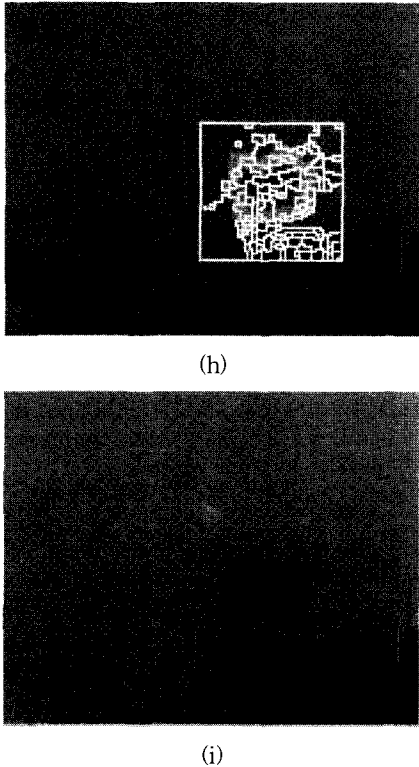


그림 14. 연기 동영상에 대한 segmentation 실험 (a) 첫 번째 프레임 원영상 (b) 첫 번째 프레임에 대한 local labeling 결과 영상 (c) 영역 기반의 intra-frame segmentation (d) 경계 기반의 intra-frame segmentation (e) 두 번째 프레임 원영상 (f) (c)에서 얻은 경계를 기반으로 intra-frame segmentation의 local labeling을 적용한 결과 영상 (g) 영역 기반의 자동 결과 영상 (h) (d)에서 얻은 경계를 기반으로 intra-frame segmentation의 local labeling을 적용한 결과 영상 (f) 경계 기반의 자동 결과 영상

Fig. 14. Image segmentation of a smoke image (a) original first frame (b) result of the local labeling at the first frame (c) result of the region-based segmentation (d) result of the edge-based segmentation (e) original second frame (f) result of the inter-frame segmentation and local labeling using (c) (g) result of the region-based segmentation (h) result of the inter-frame segmentation and local labeling using (d) (i) result of the edge-based segmentation.

그림 11과 14에서 intra-frame segmentation은 영역 기반과 경계 기반으로 나누어 실행하였다. 결과에서 보여지듯이 영역 기반 classification은 원하는 객체의 실

제 경계선이, 선택해 준 영역 내에 존재할 경우 좋은 성능을 보인다. 반면 경계 기반 classification은 실제 경계선이 선택해 준 영역 외에 존재할 경우에도 경계 정보를 따라 찾아 줄 수 있다. 경계 기반에 있어서 그림 15와 같이 's' 픽셀에서 이어질 실제 경계선을 찾아 줄 때, a, b 픽셀 위치에서의 edge값으로 어느 watershed 라인을 실제 경계선으로 따라가 줄 것인지 판단해야 한다. 이때 판단 위치에서의 edge값을 반영하여 경계선을 따라가므로, 그림 15의 a처럼 해당 판단 위치에서 edge값이 크고 나머지 부분이 약한 경우 경계 기반 classification은 이를 실제 경계선으로 따라간다는 약점이 있다.

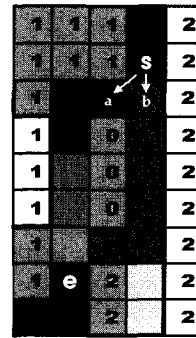


그림 15. 경계 기반 classification의 예제
Fig. 15. Example of the edge-based boundary classification.

V. 결 론

본 연구에서는 구름과 같이 형태가 일정하지 않은 non-rigid object를 동영상, 정지영상에서 보다 정확하면서 효율적으로 영역 분할하는 방법에 대해 연구하였다. 먼저 정지영상에서 watershed 알고리즘을 이용하여 영역 분할해 주는 과정은 크게 두 가지로서 전처리 과정에서는 영상의 잡음을 제거, 단순화시키고 좀 더 정확한 에지를 찾아내 이를 watershed 알고리즘을 적용하기 위한 입력영상으로 사용하였다. 후처리 과정에서는 컬러정보, 영역의 크기 그리고 영역간의 평균값의 차이를 반영하여 차이가 작은 것부터 병합할 수 있도록 하였으며, 또한 동적 메모리 할당과 배열을 이용함으로써 속도와 메모리를 줄일 수 있도록 하였다. 이 과정을 통해 중요한 에지의 손실 없이 과분할을 효율적으로 극복해 보다 정확한 결과를 얻을 수 있었다.

앞서 영역 분할한 결과를 토대로 intra-frame segmentation과 inter-frame segmentation 과정을 통해 연속되는 프레임에 대해 자동으로 관심있는 객체의 경계를 찾아 주었다. Intra-frame segmentation시에는 관심있는 대상의 영역을 포함하는 ROI를 설정하고 나머지 영역은 단순화시킨 영상을 입력으로 사용함으로써 효율성 있게 하였다. 또한 영역 그리고 경계 각각에 중점을 두어 intra-frame segmentation 단계에서 경계선을 보다 정확히 찾고자 했다. 또한 non-rigid object는 시간에 따라 불규칙한 패턴으로 변화되므로 inter-frame segmentation에서 매칭되는 점들로부터 UR을 얻어내어 ROI를 설정해 주고 intra-frame segmentation을 수행해 줌으로써 객체의 보다 정확한 경계선을 얻어낼 수 있다.

이런 non-rigid object에 대한 영역 분할 기술은 위성영상 분석 등의 기상분야, 혈류 분석과 속도 측정 및 피부병의 병위 진행 정도 파악 등의 의료분야, 적외선 감시 시스템 등의 방위 분야, 대기 오염 측정 등의 환경 분야에 다방면으로 이용될 수 있다. 따라서 현재 다양한 응용 가능성 분야에 대한 필요성을 비추어 볼 때 이 과제는 앞으로 더욱 활발히 연구될 것이다.

참 고 문 헌

- [1] Munchurl Kim, Jae Gark Choi, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 9, No. 8, pp. 1216-1226, 1999.
- [2] Luc Vincent and Pierre Soille, "Watersheds in Digital Spaces : An Efficient Algorithm Based on Immersion Simulations", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 6, pp. 583-597, 1991.
- [3] Rolf Adams and Leanne Bischof, "Seeded Region Growing", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 16, No. 6, pp. 641-647, 1994.
- [4] Milan Sonka, Vaclav Hlavac, Roger Boyle, *Image Processing, Analysis, and Machine Vision*, Brooks/Cole Publishing Company, 1999.
- [5] P. Soille, *Morphological Image Analysis : Principles and Applications*, Springer, 1999.
- [6] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, *Machine Vision*, McGraw-Hill Inc. 1995.
- [7] Kostas Haris, Serafim N. Efstratiadis, "Hybrid Image Segmentation using Watersheds and Fast Region Merging", *IEEE Trans. Image Processing*, Vol. 7, No. 12, pp. 1684-1699, 1998.
- [8] Jean-Marie Beaulieu and Morris Goldberg, "Hierarchy in Picture Segmentation : A Stepwise Optimization Approach", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, No. 2, pp. 150-163, 1989.
- [9] Munchurl Kim, Jun Geun Jeon, Jinsuk Kwak, Myoung Ho Lee, Chieteuk Ahn, Roland Mech, Michael Wollborn, "Text for VOP generation tools in MPEG-4 version 2 Visual Committee Draft", *ISO/IEC JTCL/SC29/WG11 MPEG99/m4478*, March 1999.
- [10] Christoph stiller, Janusz konrad, "Estimating motion in image sequences", *IEEE signal processing magazine*, pp. 70-91, July 1999.
- [11] A. Murat Tekalp, *Digital video processing*, Prentice Hall PTR, 1995.

저 자 소 개



李仁宰(正會員)

1999년 2월 : 성균관대학교 전자공학과 졸업(공학사). 2월 : 성균관대학교 전기전자및컴퓨터공학부 졸업(공학석사). 2001년 1월~현재 : 한국전자통신연구원 연구원.

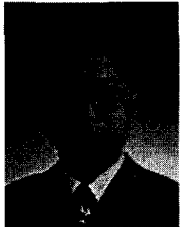
<관심분야> 영상처리, 컴퓨터비전, 멀티미디어 통신, 컴퓨터그래픽스 등



李明浩(正會員)

1983년 2월 : 송실대학교 공과대학 전자공학과 졸업(공학사). 1985년 2월 : 송실대학교 대학원 전자공학과 졸업(공학석사). 1996년 3월 : 일본 오사카대학 통신공학과 졸업(공학박사). 1985년 7월 ~ 2001년 2월 : 한국전자통신연구원 책임연구원. 2001년 3월 ~ 현재 : 관동대학교 정보기술공학부 조교수.

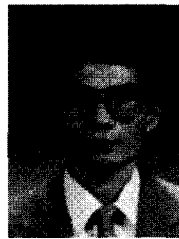
<관심분야> 영상부호화, 멀티미디어 통신, 컴퓨터 그래픽스, 디지털 방송, 디지털 워터마킹



金龍虎(正會員)

1998년 2월 : 성균관대학교 전자공학과 졸업(공학사). 2000년 2월 : 성균관대학교 전기전자및컴퓨터공학부 졸업(공학석사). 2000년 3월~현재 : 성균관대학교 전기전자및컴퓨터공학부 박사과정.

<관심분야> 영상처리, 컴퓨터비전, 의용공학, 컴퓨터그래픽스 등



安致得(正會員)

1980년 2월 : 서울대학교 공과대학 전자공학과 졸업(공학사). 1982년 2월 : 서울대학교 대학원 전자공학과 졸업(공학석사). 1991년 8월 : 미국 University of Florida 대학원 전기공학과 졸업(공학박사). 1982년 12월 ~ : 한국전자통신연구원 책임연구원 (방송시스템연구 부장). 1996년 7월 ~ : MPEG-Korea 의장. 1997년 5월 ~ : SC29-Korea 의장.

<관심 분야> 신호처리, 영상통신

金重圭(正會員) 第36卷 S編 第4號 參照