

Semantic Information 을 이용한 사용자 개입 영상분할 알고리즘

김 대 회, 호 요 성^o

광주과학기술원 정보통신공학과

A User-Assisted Video Segmentation Method Using High-Level Semantic Information

Daehee Kim and Yo-Sung Ho^o

Kwangju Institute of Science and Technology (K-JIST)

dhkim@gogh.kjist.ac.kr, hoyo@kjist.ac.kr

요 약

일반적인 자동 영상분할 알고리즘은 한 화면내의 개별적인 비디오 객체를 정확하게 분리해 내지 못한다. 더구나 이런 알고리즘들은 다양한 종류의 영상에 대한 객체의 수학적 모델이 제대로 설정되어 있지 않다. 그러나 Active Contour 알고리즘과 같이 사용자가 객체 추출 과정에 직접적으로 또는 간접적으로 개입하여 객체의 정의나 Semantic 정보를 부여한다면 훨씬 더 좋은 영상분할의 결과를 얻을 수 있다. 이러한 사용자 개입 영상분할 알고리즘은 동영상에서 움직임 객체의 VOP(Video Object Plane)를 생성하는 매우 현실적인 방법이다. 본 논문에서는 영상의 상위 레벨 Semantic 정보를 이용한 새로운 사용자 개입 영상분할 방법을 제안한다.

1. 서 론

최근 멀티미디어 통신 서비스가 확대됨에 따라 MPEG-4 표준에 대한 관심이 증대되고 있다. 기존의 MPEG-1이나 MPEG-2 영상부호화 방식과는 달리, MPEG-4 표준에서는 VOP(Video Object Plane)라는 영상 객체의 개념을 도입하여 객체기반 영상부호화를 시도하고 있다. 따라서 영상을 처리하기 위해서는 우선 영상을 구성하고 있는 객체들로 나누어야 한다. 이때 영상 객체의 모양에 대한 정보는 객체기반 부호화에서 중요한 내용이 된다. 즉, 입력되는 영상시퀀스의 각 화면은 임의의 모양을 갖는 영역으로 분할되고, 분할된 각 영역은 의미있는 비디오 객체를 나타낸다.

MPEG-4 표준의 버전 1에서 권장하는 영상분할 방법은 시공간(Spatio-Temporal) 영상분할 알고리즘이다 [1]. 이 방법은 움직임 검출 마스크(Change Detection Mask, CDM)과 형태학적(Morphological) 접근 방법을 사용한다. 비디오 객체의 정확한 경계는 형태학적인 방법을 이용하여 검출하고, 비디오 객체의 영역은 CDM을 이용하여 찾는다. 그러나 이 방법은 단일 프레임에서 복수의 비디오 객체를 구분하기 곤란하며, 비디오 객체에 대한 정의 자체가 주관적이고 정확한 수학적 모델을 제시할 수 없기 때문에 MPEG-4 응용에 실제 사용하기에 부적절하다.

만약 사용자가 초기 화면에서 비디오 객체에 대한 정보를 부여하고 이를 적절히 활용한다면, 이후에 연속되는 화면에서는 정확한 영상분할의 결과를

효과적으로 얻을 수 있다 [2]. 이러한 사용자의 도움을 이용하는 영상분할 방법이 MPEG-4 부호화 방법에서 VOP를 생성하는 데 보다 실질적인 대안이 될 수 있다.

본 논문에서는 사용자의 정보를 입력받아 이를 활용하는 방법으로 Active Contour 알고리즘을 사용한다. Active Contour를 이용하여 초기 객체를 정의하고, 정의된 객체로부터 연속되는 프레임에서 객체의 윤곽선을 추적하여 동영상의 영상분할을 수행하는 알고리즘을 제안한다.

2. User-Assisted Segmentation

사용자가 부여하는 정보를 활용하는 반자동 영상분할 방법에서는 좋은 성능을 얻기 위해 사용자의 상위레벨(High-Level) Semantic 정보를 사용한다. 그림 1에 나타낸 것과 같이, 사용자 개입 영상분할 알고리즘은 Intra-Frame 분할과 Inter-Frame 분할의 두 단계로 구분된다.

Intra-Frame 분할 단계에서는 사용자가 초기 프레임의 객체나 임의의 프레임에서 관심있는 객체에 대한 정보를 마우스와 같은 입력 장비를 이용하여 입력시킨다. 이때 주어지는 정보는 객체 윤곽선 주변의 제어점들이다. 이러한 제어점들로부터 Active Contour 알고리즘을 적용하여 객체의 정확한 경계를 추출한다. 이렇게 얻어진 객체를 사전에 정의된 객체(Predefined Video Object)라고 말한다. 사전에 정의된 객체의 경계는 Inter-Frame 분할 단계로 전달된다.

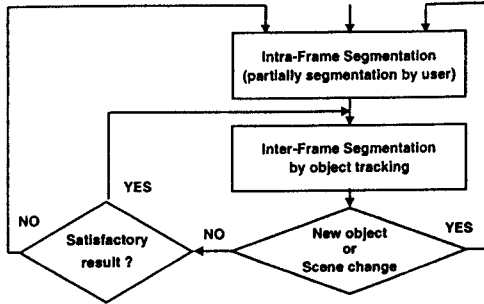


그림 1. 사용자 개입 영상분할 알고리즘

Intra-Frame 분할 단계에서 비디오 객체가 정의되면, Inter-Frame 분할 단계에서는 객체 추적(Tracking) 알고리즘을 이용하여 이후의 프레임에서 이미 정의된 객체를 연속된 시간축의 프레임 상에서 추적한다. 이때 만족스럽지 못한 결과가 나오거나 관심있는 새로운 객체가 등장하면 Intra-Frame 분할 단계로 되돌아 간다. Inter-Frame 분할 단계는 곡선 추적(Contour Tracking)과 곡선 조절(Fitting) 과정으로 구성된다.

3. Intra-Frame Segmentation

본 논문에서는 Intra-Frame 분할 단계에서 Active Contour를 사용한다. Active Contour는 Kass가 처음 제안한 방법으로 Active Contour가 가지는 위치 에너지를 이용한다 [3]. 이 방법은 위치 에너지가 큰 높은 위치의 물체가 낮은 위치로 자연스럽게 이동하는 현상을 모델화한 것이다.

기본적으로 Active Contour 알고리즘은 초기에 사용자가 입력한 곡선으로부터 에너지를 최소화하는 Spline 곡선을 찾는다. 이 알고리즘에서는 에너지 함수를 어떻게 정의하느냐가 매우 중요하다.

Active Contour의 모양은 내부 에너지, 외부 에너지, 제한 조건의 세 항목에 의해 조절된다. 외부 에너지는 Active Contour 영상의 특징을 추적하도록 도와주고, 내부 에너지는 곡선의 부드러운 변화를 보장한다. Active Contour 알고리즘을 전개할 때, 제한적인 조건은 물론 사용자의 개입이 허용된다. Active Contour에 대한 수식은 다음과 같다.

$$E_{snake}^* = \int_0^1 E_{snake}(v(s)) ds$$

$$= \int_0^1 [E_{int}(v(s)) + E_{ext}(v(s)) + E_{con}(v(s))] ds \quad (1)$$

여기서 첫째 항은 내부 에너지를, 둘째 항은 외부 에너지를, 마지막 항은 제한 조건을 의미한다. Active Contour의 목표는 식(1)의 에너지 함수를 주어진 영역에서 국부적으로 최소화하는 곡선 $v(s)$ 를 구하는 것이다.

내부 에너지는 곡선의 구부러지는 정도를 나타내며, 식(2)와 같이 표현된다.

$$E_{int} = (\alpha(s)|v_x(s)|^2 + \beta(s)|v_{xx}(s)|^2) \quad (2)$$

여기서 첫째 항은 곡선의 1차 미분값 $v_x(s)$ 를 이용하므로 곡선의 불연속적인 곳에서 큰 값을 갖는다. 둘째 항은 곡선이 급격하게 휘어지는 지점에서 큰 값을 갖는다. 그러므로 각 점에서의 $\alpha(s)$ 와 $\beta(s)$ 의 값은 그 점에서의 객체의 윤곽선 특성을 잘 반영해야 한다. 예를 들면, 큰 값의 $\beta(s)$ 는 곡선을 부드럽게 만들므로, 폐곡선(Closed Contour)의 경우에는 원을 생성하고, 개곡선(Opened Contour)의 경우는 직선을 만든다. 만약 $\alpha(s)$ 가 0이면, 그 점에서 불연속성이 일어날 수 있다. 반면에 $\beta(s)$ 가 0이면, 그 점이 코너가 될 수 있다. 따라서 $\alpha(s)$ 와 $\beta(s)$ 는 영상에 따라서 적절한 값을 가지도록 선택해야 한다.

외부 에너지는 영상에 의해서 주어지는 에너지로서, 일반적으로 영상 객체의 윤곽선을 반영할 수 있는 Laplace 연산자나 Sobel 연산자와 같은 방법을 이용하여 구하며, 경사도(Gradient)가 큰 곳에서 작은 에너지를 갖도록 한다. 즉, 영상의 Edge에서 작은 값을 가지므로, 곡선이 영상의 윤곽선에 놓이는 것을 보장한다. 보통 영상에서 단순 마스크 연산을 통해 객체의 윤곽선을 추출하기 위해서는 배경이 비교적 단조로워야 효과적인 성능을 발휘할 수 있지만, 대부분의 영상은 배경이 복잡하므로 효과적인 객체의 특징을 추출하기가 어렵다. 또한 Active Contour의 최소화 과정(Minimizing Process)을 수행할 때, 복잡한 영상의 경우는 마지막 곡선이 우리가 원하지 않는 국부 최소값으로 수렴할 수 있다.

3.1 형태학적 외부에너지를 이용한 Active Contour 알고리즘

기존의 Active Contour 알고리즘은 단순한 배경(Homogeneous Background)을 가지는 영상에서는 제대로 동작하지만, 복잡한 배경에서는 잘 동작하지 않는다. 본 논문에서는 복잡한 배경을 가진 영상에도 효과적으로 적용할 수 있도록 다음과 같은 새로운 방법을 제안한다.

식(1)을 이산화하기 위해서는 유한 개의 제어점을 가지는 곡선으로 표현해야 하며, 식(1)은 식(3)과 같이 표현될 수 있다. 여기서 n 은 곡선 위에 있는 전체 제어점(Control Point)의 개수이고, λ_i 는 정규화 변수(Regularization Parameter)이다.

$$E_{snake}^* = \sum_{i=1}^n \lambda_i E_{int}(v_i) + (1 - \lambda_i) E_{ext}(v_i) \quad (3)$$

식(3)의 내부 에너지를 적용하기 위해서는 식(2)를 이산화해야 하며, 식(4)와 같이 표현할 수 있다.

$$E_{int}(v_i) = (\alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i+1} - 2v_i + v_{i-1}|^2) \quad (4)$$

그러나 식(4)를 사용할 경우에는 곡선이 점차 흐트러져 외부 에너지가 없는 경우에는 곡선이 하나의 점으로 수렴한다. 이것을 막기 위해 여러 가지 변형된 내부 에너지의 공식이 사용되었다. 본 논문에서는 내부 에너지를 다음과 같이 정의하였다.

$$E_{im}(v_i) = \frac{1}{l(V)} |v_i - \alpha(v_{i-1} + v_{i+1})|^2 \quad (5)$$

여기서 α 값은 개곡선의 경우는 0.5로 설정하고, 폐곡선의 경우는 $0.5\cos^{-1}(2\pi/n)$ 으로 설정한다. 식(5)는 개곡선과 폐곡선에서 제어점 사이의 평균 길이를 이용하므로, 내부 에너지를 이산화할 때 곡선이 움추러드는 문제를 해결하였다. 식(5)에서 $l(V)$ 는 Active Contour의 제어점 사이의 평균 거리이다.

$$l(V) = \frac{1}{n} \sum_{i=1}^n |v_i - v_{i-1}|^2 \quad (6)$$

본 논문에서는 두 종류의 외부 에너지를 정의하였는데, 그 첫번째는 영상에 윤곽선 검출기(Edge Detector)를 적용하여 얻은 결과를 에너지로 이용하는 것이다. 윤곽선 검출기에서 구한 외부 에너지는 영상의 각 제어점으로부터 이웃하는 점과의 차분에 의한 경사도 벡터(Gradient Vector)를 구하여, 식(7)과 같이 적용하였다.

$$E_{grad}(v_i) = 1 - |n_i^T g(v_i)| \quad (7)$$

여기서 $g(v_i)$ 는 제어점 v_i 에서의 2×1 경사도 벡터이고, $n_i(v_i)$ 은 v_i 에서의 곡선의 방향에 직교하는 방향의 단위 법선 벡터(Unit Normal Vector)이다. 따라서 윤곽선의 경사도 벡터가 곡선의 법선의 벡터 같은 방향일 때 에너지는 최소값을 가진다. 곡선의 법선 벡터는 식(8)과 같은 접선 벡터(Tangent Vector)를 통해 얻어진다.

$$t_i = \frac{v_i - v_{i-1}}{|v_i - v_{i-1}|} + \frac{v_{i+1} - v_i}{|v_{i+1} - v_i|} \quad (8)$$

만약 외부 에너지를 기존의 Active Contour 알고리즘에서 사용하는 Laplace 연산자나 Sobel 연산자와 같은 단순 마스크 연산자를 이용하여 생성하면, 복잡한 배경에 대해서 좋은 성능을 얻지 못한다. 따라서 본 논문에서는 이를 해결하기 위해 형태학적 처리 방법을 사용하여 두 번째의 외부 에너지를 정의하였다.

영상의 형태학적 처리에 기반한 에너지를 산출하기 위해서는, 먼저 형태학적 연산(Morphological Operation)을 이용하여 영상의 복잡성을 제거한다. 이 절차는 뒤에 언급할 Watershed Line이 너무 많이 생기는 것을 억제하기 위해서도 필요하다. 이때 영상을 단순화시키기 위해 사용한 형태학적 필터는 식(9)과 식(10)에 나타내었다.

Open by reconstruction of erosion: $\gamma^{(rec)}(\epsilon_n(f), f)$ (9)

Close by reconstruction of dilation: $\phi^{(rec)}(\delta_n(f), f)$ (10)

식(9)의 경우, 우선 침식(Erosion) ϵ_n 에 의해 형태요소(Structuring Element)보다 작은 영역을 제거한다. 그 다음 재구성(Reconstruction)을 이용하여 침식에서

완전히 제거되지 않은 영역의 경계를 복원한다. 재구성에 의한 close 필터도 위와 비슷한 결과를 가진다. 재구성에 의한 open 필터는 형태 요소보다 작은 밝은 영역을 제거하고, 재구성에 의한 Close 필터는 어두운 상세 영역을 제거하는 차이점을 가진다. 이들 필터들은 경계 정보를 잘 보존하는 장점을 가지고 있다 [5].

재구성에 의한 open_close 필터와 close_open 필터는 앞에서 정의된 재구성에 의한 open 필터와 close 필터의 조합으로 정의되며, 밝은 상세 영역과 어두운 상세 영역을 동시에 제거할 수 있다. 재구성에 의한 open_close 필터와 close_open 필터는 원영상보다 단순한 결과를 만들지만, 영상의 객체를 명확하게 드러나게 한다.

다음으로 형태학적 경사도(Morphological Gradient)를 사용하여 영상의 특징을 추출한다. 이 과정은 밝기가 균일한 평지 영역을 구분하는 Watershed 알고리즘의 입력 영상을 만들기 위한 것이다. 형태학적 경사도가 작은 영역들이 Watershed 알고리즘에 의해 초기 평지 지역으로 설정된다. 주어진 임계값보다 낮은 경사도를 가지는 영역들은 너무 많은 영역으로 분할되는 것을 막기 위해 0으로 설정하였다.

이와 같이 구한 형태학적 경사도 영상은 Watershed 알고리즘[6]에서 생성된 Watershed Line을 통해 정교한 영역으로 분할된다. Watershed 알고리즘은 집수 지역들(Catchment Basins)과 그들을 나누는 경계선인 Watershed Line을 결정하는 알고리즘이다. 본 논문에서는 여러 종류의 Watershed 알고리즘 중에서 침수 실험(Immersion Simulation)을 이용한 방법을 사용하였다. 침수 실험은 각 국부 최소점에서 표면을 침수시켜 가면서 Watershed Line을 찾는 방법이다. 그림 2는 Watershed 알고리즘의 개념을 직관적으로 일차원 공간에 나타낸 것이다.

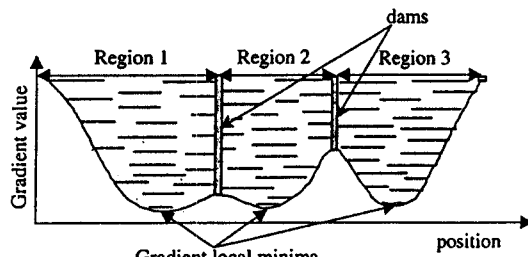


그림 2. Watershed 알고리즘

여기서 찾아진 국부 최소점은 영역 확장의 출발점이 된다. 그리고 경사도에 의해 나누어진 골짜기 조각들은 점진적으로 침수되어 집수장(Catchment Basin)을 이루어 물이 채워진다. 물의 표면이 경사도의 산마루를 만나면 인접 집수장과 구분하기 위해서 댐을 만들고, 마지막에 가서 모든 집수장의 물의 높이가 같아질 때까지 지속적으로 물을 채워 나간다. 결국 모든 영역은 댐이나 산마루에 의해 둘러싸이게 되어 각 영역이 분할된다. 이때 형성된 댐과

산마루를 연결하는 선을 Watershed Line 이라고 한다. 이때 분할된 각 영역은 유일한 값으로 표시된다.

Watershed 알고리즘에서 얻은 영상에서 Watershed Line 의 위치는 1로, 다른 위치는 0으로 치환하여 새로운 이진 영상(W_{img})을 구한다. 이렇게 구한 이진 영상에 형태학적 외부 에너지를 식(11)과 같이 정의한다. 여기서 3×3 의 작은 마스크 윈도우를 이용하는 이유는 Watershed Line 이 한 화소 두께의 직선이기 때문이다.

$$E_{mor}(v_i) = 1 - \sum_{m=-1}^1 \sum_{n=-1}^1 W_{img}(v_{ix} + m)(v_{iy} + n) \quad (11)$$

여기서 v_{ix} 와 v_{iy} 는 벡터 v_i 의 두 요소성분이다.

3.2 최소화 알고리즘

이 절에서는 앞에서 정의한 에너지와 초기에 사용자가 제공한 곡선을 이용하여 식(3)의 함수를 최소화하는 제어점의 집합을 찾아내는 방법을 다룬다. 본 논문에서는 Euler 방정식에 기초한 접근 방법과 LMS(Least Mean Square)에 기초한 방법에서 발생할 수 있는 수치적 오류를 극복하기 위해 Amimi가 처음으로 Active Contour에 적용한 Dynamic Programming의 접근 방법을 사용한다 [8].

본 논문에서는 매개변수 λ 값을 미리 조절할 수 없으므로, 식(3)을 최소화하기 위해 각 제어점에서 임의의 λ 값에 대해 최대값이 나오는 경우를 최소화하는 곡선을 구한다. 이것은 식(12)의 국부적인 Minimax 비용함수(Cost Function)로 나타낼 수 있다.

$$E_{snake}^{final} = \min_v \sum_{i=1}^n \max(E_{int}(v_i), E_{ext}(v_i)) \quad (12)$$

최소화 과정은 n 개의 독립적인 단계로 분할되어 수행될 수 있다. 그러므로 Dynamic Programming에 기반한 Active Contour 알고리즘의 최소화 과정을 식(14)와 같이 기술할 수 있다. 각 점에서의 에너지는 내부 에너지의 항을 가지므로, 연속된 3개의 제어점이나 제어점 주변의 후보점들로 정의된다. 본 논문에서는 두 종류의 외부 에너지를 사용하므로, 임의의 위치에 있는 제어점의 에너지를 다음과 같이 정의한다.

$$E_i = \max(E_{int}(v_{i-1}, v_i, v_{i+1}), E_{mor}(v_i), E_{grd}(v_i)) \quad (13)$$

Dynamic Programming의 최소화 절차를 간단히 정리하면 다음과 같다.

$$\begin{aligned} s_1(v_2, v_3) &= \min_{v_1} E_2(v_1, v_2, v_3) \\ s_2(v_3, v_4) &= \min_{v_2} (s_1(v_2, v_3) + E_3(v_2, v_3, v_4)) \\ &\vdots \\ s_{n-2}(v_{n-1}, v_n) &= \min_{v_{n-2}} (s_{n-3}(v_{n-2}, v_{n-1}) \\ &\quad + E_{n-1}(v_{n-2}, v_{n-1}, v_n)) \end{aligned} \quad (14)$$

Dynamic Programming의 최소화 절차는 많은 연산량을 요구하지만, 수치적으로 안정적이다. 최소화 절차를 수행한 후에는 Intra-Frame 분할의 결과로 사전에 정의된 객체(Predefined Video Object)를 얻는다.

3.3 Intra-Frame 영상분할 실험결과

그림 3은 Intra-Frame 영상분할의 실험결과를 보여주고 있다. 그림 3(a)는 사용자가 마우스를 이용하여 표시한 초기 곡선의 제어점을 나타낸다. 그림 3(b)는 형태학적 도구를 이용하여 Watershed 알고리즘으로 얻은 실험결과로써, 이것은 형태학적 외부 에너지를 산출하기 위해 사용된다. 그림 3(c)는 주어진 초기 곡선으로부터 Lai가 제안한 Generalized Snake 알고리즘[11]을 이용하여 얻은 실험결과이다. 그림 3(d)는 본 논문에서 정의한 형태학적 에너지를 가지고 최소화 과정을 이용하여 얻은 실험결과이다. 앞서 언급한 최소화 과정은 많은 연산량을 요구하므로, 본 논문에서는 3단계 움직임 벡터 탐색 알고리즘[12]과 같이 점진적으로 탐색 영역을 축소하여 찾아가는 방법을 사용하였다.



그림 3. Intra-Frame 영상분할 실험결과

그림 3(c)에서 볼 수 있듯이, 같은 영역이라도 영상의 경사도가 국부적인 최소값(Local Minimum)을 가지면 그곳으로 수렴한다. 그림 3(d)는 형태학적 에너지의 차이가 없는 지역에서는 식(7)의 경사도를 이용한 에너지가 최소화되는 방향으로 수렴하고, 영상의 경사도가 적은 지역에서는 형태학적 에너지를 최소화하는 방향으로 수렴하여 객체의 윤곽을 명확하게 추출한다. 이러한 특징이 형태학적 접근 방법을 이용하면 복잡한 배경에 대해서도 향상된 성능을 보일 수 있도록 한다.

4. Inter-Frame Segmentation

본 장에서는 첫 프레임 이후의 프레임에 위해서 사전에 정의된 객체(Predefined Video Object)를 추적하는 방법을 제안한다. 이 단계는 사전 정의된 객체의 윤곽선 추적과 추적된 윤곽선의 보상의 두 단계로 구성된다.

4.1 윤곽선 추적 알고리즘

일반적인 동영상의 연속된 화면 사이에는 상관도가 매우 높다. 그러므로 Intra-Frame 영상분할 이후의 프레임들에 대해서는 이렇게 높은 상관도를 이용하여 객체를 추적할 수 있다. 일반적으로 움직임 벡터 탐색 알고리즘은 연속된 영상의 유사도를 이용하여 임의의 영역이 다음 프레임의 어느 곳으로 이동하였는지를 추정하는 방법이므로 객체 추적 알고리즘으로 사용할 수 있다. 비디오 객체의 추적을 위해 이전 프레임에서 분할된 객체의 제어점을 선택한다. 그리고 움직임 추정 방법에 의해서 현재 프레임으로 제어점을 투사한다. 따라서 움직임 추정 방법은 Intra-Frame 분할의 결과로 생성된 비디오 객체의 제어점을 따라서 순차적으로 수행된다.

움직임 추정 방법을 단순히 제어점 단위로 수행할 경우에는 화소 순환 방법(Pel-Recursive Method)에서 가지는 문제점인 잡음에 민감한 특성을 가진다 [12]. 따라서 본 논문에서는 제어점에 대한 움직임 벡터를 찾는 것이지만, 화소 단위의 방법이 아닌 블록 정합 방법에 기초하고 있다. 즉, 제어점을 중심으로 한 블록을 설정하여, 블록 정합 방법을 이용하여 움직임 벡터를 찾으며, 이를 그 블록 중심점의 움직임 벡터로 간주한다. 본 논문에서는 보다 더 안정적인 결과를 얻기 위해서는 인접하는 4개의 블록을 설정하고, 각각의 움직임 벡터를 구한다. 이 인접 블록들의 중심은 원래 블록의 중심으로부터 두 화소 거리에 있다. 이러한 다섯 개의 움직임 벡터로부터 중간값을 제어점의 움직임 벡터로 설정하였다. 겹침 움직임 추정 방법(Overlapped Motion Estimation Method)은 좀더 안정적인 성능을 보여준다 [9, 10].

일반적인 블록정합 방법은 전경과 배경을 동등하게 고려한다. 따라서 객체의 이동에 의해 가려진 영역이 나타날 때 구해진 움직임 벡터가 부정확할 수 있다. 이것을 해결하기 위해 본 논문에서는 일반적인 블록정합 방법을 변형하여 배경을 고려하지 않았다.

본 논문에서는 임의의 모양을 가지는 블록(Arbitrary Shaped Block, ASB)을 설정하여 움직임 벡터를 탐색한다. 이전 프레임의 사전에 정의된 객체의 제어점에 정방형 블록의 중심을 위치시키고, 배경 영역을 무시하였다. 따라서 블록의 형태는 객체의 개 부분에서 객체의 모양과 일치한다. 배경을 포함한 전체 블록의 크기는 7x7 이고, 탐색 영역은 -8에서 8로 설정하였다. 이 움직임 탐색 단계에서는 현재 프레임의 ASB에 가장 잘 정합되는 이전 프레임

의 ASB를 찾는다. 여기서 객체가 비강성체(Non-Rigid Body)인 경우에는 문제점을 발생시킬 수 있으나, 본 논문에서는 비강성체의 경우 연속된 프레임 사이의 변화가 적고, 움직임 추정 단계 이후에 윤곽선 보상 단계를 거치므로, ASB 단위의 블록정합 방법이 유효하다.

본 논문에서는 네 개의 인접 블록을 고려하였으므로 다섯 개의 ASB로부터 다섯 개의 움직임 벡터가 얻어진다. 여기서 중간값을 제어점의 움직임 벡터로 설정한다. 움직임 벡터는 식(15)와 같은 차이값의 절대값 평균이 최소인 점을 벡터로 설정한다.

$$MAD = \frac{1}{M} \sum_{(x,y) \in A} |F_{k-1}(x,y) - F_k(x + MV_x, y + MV_y)| \quad (15)$$

여기서 A는 ASB의 화소들의 집합이고, M은 집합 A의 화소의 수이다. 그리고 MV는 움직임 벡터를 나타내고, F는 밝기값을 나타낸다.

4.2 윤곽선 보상 알고리즘

Inter-Frame 분할의 두 번째 단계는 추정된 객체의 윤곽선을 보상하는 과정이다. 단순한 움직임 추정 동작으로 얻은 현재 프레임의 비디오 객체 윤곽선은 실제의 윤곽선과 차이를 보일 것이다. 또한 앞서 언급한 비강성체에 대한 문제가 발생할 수 있다. 따라서 이전 프레임의 모든 제어점을 현재 프레임으로 투사시키고, 투사된 제어점을 현재 프레임에서 Active Contour를 위한 초기 제어점으로 재설정한다.

이렇게 재설정된 제어점을 Intra-Frame 분할 방법에서 제안한 Active Contour 알고리즘을 이용하여 정확한 객체의 윤곽선을 보상한다. 이와 같이 보상된 윤곽선은 다시 다음 프레임에 움직임 추정 방법을 통해 투사되고, 윤곽선이 Active Contour 알고리즘에 의해 보상되는 과정을 반복적으로 수행한다.

4.3 Inter-Frame 영상분할 실험결과

본 논문에서는 Mother_Daughter의 CIF(352x288) 동영상을 사용하여 Inter-Frame 영상분할 알고리즘의 동작을 실험하였다. 그림 4는 제안된 알고리즘의 실험 결과이다. 그림 4의 실험에서도, Intra-Frame 분할의 경우처럼, 최소화 과정에서 점진적으로 탐색 영역을 축소하여 찾아가는 방법을 사용하였다.

그림 4(a)는 사용자가 마우스를 이용하여 입력한 초기 제어점들이다. 그림 4(b)는 Intra-Frame 분할에서 Active Contour 알고리즘을 사용하여 얻은 실험 결과이다. 그림 4에서도 앞 절에서 언급한 형태학적 에너지를 이용으로 얻은 장점을 보여준다.

그림 4의 다른 그림들은 Inter-Frame 영상분할의 실험결과이다. 즉, 그림 4(b)의 제어점을 이용하여 1번 프레임에서 객체의 제어점을 추정하고, Intra-Frame에서와 같이 추정된 제어점을 초기 곡선으로 설정하여, Active Contour 알고리즘을 이용하여 1번

프레임의 객체를 구한다. 그리고 이 과정을 반복적으로 수행한다. 그림 4는 전체 영상 중에서 7프레임 간격으로 실험결과를 보여준 것이다. Dynamic Programming 알고리즘에서 계산량을 줄이기 위해, Intra-Frame 분할에서 언급한 것처럼, 3 단계로 탐색 영역을 축소하는 방법을 사용하였다.

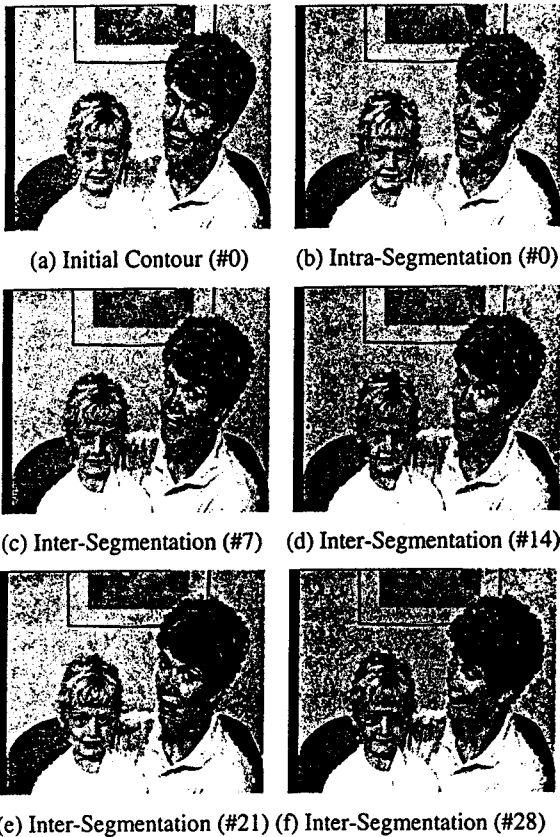


그림 4. 사용자 개입 영상분할 실험결과

5. 결론

본 논문에서는 영상 신호에서 비디오 객체를 추출하기 위해 사용자 개입 영상분할 알고리즘을 제안하였다. 제안된 알고리즘은 Intra-Frame 분할과 Inter-Frame 분할 두 단계로 구성된다. 본 논문에서 제안된 Active Contour 알고리즘은 평균 거리를 이용한 내부 에너지를 정의하여 곡선이 일반적으로 축소되는 것을 막았고, 외부 에너지는 경사도 벡터와 형태학적 윤곽선을 사용하여 정의하였다. 이 두 종류의 외부 에너지는 영상이 단순한 지역과 복잡한 지역에서 서로 상보적으로 작용하는 특성을 가지고 있다. 본 논문에서 제시한 Active Contour 알고리즘은 단순한 배경 뿐만 아니라, 복잡한 배경에서도 객체의 윤곽선을 정확하게 추출할 수 있다. 그리고 Inter-Frame 분할을 위해 ASB (Arbitrary Shaped Block)를 이용한 움직임 추정 방법과 겹침 움직임 추정

방법을 이용하여 객체를 추적하는 방법을 제시하였다. 추정된 객체의 정확한 윤곽선을 보상하기 위해 Intra-Frame 분할에서 사용된 알고리즘을 반복적으로 적용하여 효율적인 영상분할을 수행하였다.

감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC) 지원금에 의한 것입니다.

6. 참고문헌

- [1] M. Kim, J.G. Choi, D. Kim, H. Lee, M.H. Lee, C. Ahn and Y.S. Ho, "A VOP generation tool: Automatic Segmentation of moving objects in image sequences based on spatio-temporal information", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp.1216-1226, 1999.
- [2] 김대회, 김민호, 호요성, "MPEG-4 동영상 부호화를 위한 영상 객체 추출 기법", *전자공학회지*, 제 26 권, 제 7 호, pp. 714-723, 1999.
- [3] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models," *Proc. First Int. Conf. on Computr Vision*, pp. 259-269, 1987.
- [4] K. F. Lai and R. T. Chin, "Deformable contours: modeling, extraction," *EEE Trans. Patt. Anal. Mach. Intel.*, Vol. 17, No 11, pp. 1084-1090, Nov. 1994.
- [5] P. Salembier and M. Pardas, "Hierarchical morphological segmentation for image sequence coding," *IEEE Trans. Image Processing*, Vol. 3, No. 5, pp. 639-651, Sep. 1994.
- [6] P. Salembier, "Morphological multiscale segmentation for image coding," *Signal Processing*, Vol. 38, No. 3, pp. 359-386, Jan. 1994.
- [7] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Patt. Anal. Mach. Intel.*, Vol. 13, No.5, pp. 583-598, June 1991.
- [8] A. A. Amimi, T. E. Weymouth and R. C. Jain, "Using dynamic programming for solving variational problems in vision", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855-867, Sep. 1990.
- [9] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach", *IEEE Trans. Image Processing*, vol. 3, no.5, pp. 693-699, 1994.
- [10] C. Auyeung and J. Kosmach, "Overlapped block motion compensation", *Visual Communications and Image Processing'92*, vol. 1818, pp. 561-572, 1992.
- [11] K. F. Lai and R. T. Chin, "Deformable contours: modeling, extraction", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no 11, pp. 1084-1090, Nov. 1994.
- [12] A. M. Tekalp, *Digital Video Processing*, Prentice-Hall, 1995.