

B-Spline 곡선을 이용한 반자동 영상분할 알고리즘

김대희, 호요성
광주과학기술원 정보통신공학과
광주광역시 북구 오룡동 1번지

A Semi-Automatic Segmentation Algorithm using B-Spline Curves

Daehee Kim and Yo-Sung Ho
Kwangju Institute of Science and Technology (K-JIST)
E-mail: kimdh@kjist.ac.kr

요약

대부분의 자동 영상분할 방법은 한 화면에서 개별 객체를 추출하기가 어렵고, 비디오 객체의 명확한 모델이 없어 자동분할 방법에는 한계가 존재한다. 본 논문에서는 이러한 문제점을 극복하기 위해 Active Contour를 이용한 반자동 영상분할 방법을 제안한다. 일반적인 Active Contour 알고리즘은 유한개의 제어점을 설정하고 그 차이로 곡선의 특성을 묘사하므로 곡선위의 제어점 사이의 모양 정보를 표현하는데 불충분하다. 또한, 대부분의 Active Contour 알고리즘은 단순한 배경을 갖는 객체에는 잘 적용되도록 설계되었으나, 복잡한 배경을 갖는 객체에는 부적절한 단점을 가지고 있다. 따라서 본 논문은 B-Spline을 이용하여 곡선을 표현하고, 복잡한 영상에서도 좋은 성능을 갖도록 곡선의 외부 에너지는 SUSAN 연산자를 이용하여 추출하였다.

1. 서론

최근 멀티미디어 통신 서비스가 확대됨에 따라 MPEG-4 표준에 대한 관심이 크게 높아지고 있다. 기존의 영상부호화 방식과는 달리, MPEG-4 표준에서는 VOP(Video Object Plane)라는 개념을 도입하여 객체기반 영상부호화를 시도하고 있다. 따라서 우선 영상을 구성하고 있는 객체들로 나누어야 한다. 이때 영상객체의 모양에 관한 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 표준에서 권장하는 영상분할 방법은 시공간(Spatio-Temporal) 영상분할 알고리즘이다 [1, 2]. 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 복

수의 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 [3]. 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. 반자동 영상분할 알고리즘

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

Intra-Frame 분할 단계에서는 사용자가 초기 프레임의 객체나 임의의 프레임에서 관심이 있는 객체에 대한 정보를 입력시킨다. 이때 주어지는 정보는 객체 윤곽선 주변의 제어점들이다. 이러한 제어점들로부터 초기 곡선을 만들고, Active Contour 알고리즘을 적용하여 객체의 정확한 경계를 추출한다.

Intra-Frame 분할 단계에서 비디오 객체가 정의되면, Inter-Frame 분할 단계에서는 객체 추적 알고리즘을 이용하여 이후의 프레임에서 이미 정의된 객체를 시간축으로 뒤따라 오는 프레임 상에서 추적한다. 이때 만족스럽지 못한 결과가 나오거나 관심있는 새로운 객체가 등장하면 Intra-Frame 분할 단계로 되돌아 간다. Inter-Frame 분할 단계는 곡선 추적(Tracking) 과정과

모양 조절(Fitting) 과정으로 구성된다.

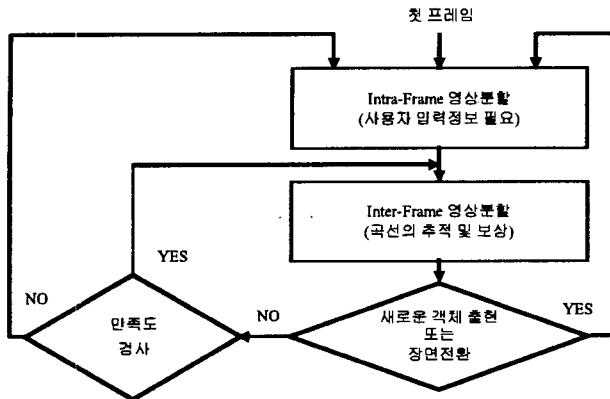


그림 1. 반자동 영상분할 알고리즘

3. 반자동 영상분할 알고리즘

3.1 Active Contour

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. 사용자의 개입은 제한 조건을 통해 허용된다. Active Contour에 대한 수식은 다음과 같다 [4].

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

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

첫번째 항은 내부 에너지를, 두번째 항은 외부 에너지를, 마지막 항은 제한 조건을 의미한다. Active Contour의 목표는 식 (1)의 에너지 함수를 주어진 영역에서 국부적으로 최소화하는 곡선 $\mathbf{r}(s)$ 를 구하는 것이다. 내부 에너지는 곡선의 구부러지는 정도를 나타내며, 식 (2)와 같이 표현된다

$$E_{int} = (\alpha(s)|r_s(s)|^2 + \beta(s)|r_{ss}(s)|^2)/2 \quad (2)$$

첫번째 항은 곡선의 1차 미분값을 이용하므로 곡선의 불연속적인 곳에서 큰 값을 갖는다. 두번째 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. 외부 에너지는 영상에 의해서 주어지는 에너지로서, 일반적으로 영상 객체의 윤곽선을 반영할 수 있도록 정의되어야 한다.

식 (1)을 최소화하는 곡선 $\mathbf{r}(s)$ 를 구하는 것은 Euler 방정식을 이용한 Variational Calculus를 구하는 문제로 간주할 수 있다. Kass는 재귀적인 접근으로 이것을 구현하였으나 [4], 이 방법은 수치적으로 불안정한 방법으로 밝혀졌다. 이러한 문제를 해결하기 위해 Dynamic Programming 방법을 이용한 접근 방법이 많이 사용되고 있다 [5]. 그러나 이 방법은 연산량이 많다

는 단점을 가지고 있다. 이를 개선하기 위한 고속 Active Contour 알고리즘이 제안되었으나, 이 방법은 초기 곡선의 제어점들의 간격을 정교하게 설정해야 하는 단점을 가지고 있다 [6].

3.2 B-Spline Snake

$E(s)$ 의 실질적인 계산은 디지털 영역에서 수행되므로 곡선도 역시 이산화되어야 한다. 따라서 이산화된 내부에너지는 식 (3)과 같다.

$$E_{int} = \alpha_i |r_i - r_{i-1}|^2 + \beta_i |r_{i+1} - 2r_i + r_{i-1}|^2 \quad (3)$$

외부에너지 없이 식 (3)만 사용하면, 곡선은 하나의 점으로 수렴한다. 이런 문제를 해결하기 위해 식 (3)의 변형들이 많이 사용될 수 있다. 그러나 유한 점들의 차이를 이용한 식 (3)은 점들 사이에서의 곡선에 대한 정보를 완벽하게 제공하지 못한다. 따라서 연속적인 곡선 $\mathbf{r}(s)$ 를 완벽하게 복원할 수 있는 제어점 $\mathbf{r}(s_i)$ 을 이용하는 방안이 고려되어야 한다. 다항식의 Spline으로 $\mathbf{r}(s)$ 를 형성하면 곡선의 부드러운 변화를 얻을 수 있다. 그러나 Spline 곡선은 영상의 특성을 고려하지 않기 때문에 Spline 곡선은 영상의 구석점 위치를 지나가지 않을 수도 있다. 이런 문제는 다중 연결점 또는 다중 제어점을 이용하여 Spline 곡선의 부드러운 변화를 조절하는 NURBS 곡선을 사용하여 해결할 수 있다 [7].

매개변수를 이용한 곡선 $\mathbf{r}(s)=(x(s), y(s))$ 은 s 의 함수이다. 차수 d 의 B-Spline의 기저함수는 연결점에서 서로 연결된 d 개의 다항식 곡선 조각으로 구성된 함수이다. 복잡한 모양은 높은 차수의 다항식으로 근사화될 수 있다. 일반적으로 다항식의 차수는 3차(Quadratic)나 4차(Cubic)이 사용된다.

B-Spline 함수 $x(s)$ 는 N_B 개의 기저함수 $B_n(s)$ 의 가중합(Weighted Sum)으로 나타낼 수 있다. 다항식의 차수를 3으로 설정함으로써 곡선은 연속적인 기울기를 갖는다. 따라서, 이렇게 구성된 Spline 함수는 내부에너지의 요구 조건을 자연스럽게 충족시킨다.

Spline 함수는 식 (4)와 같이 표현된다.

$$x(s) = \sum_{n=0}^{N_B-1} x_n B_n(s) \quad (4)$$

여기서 x_n 은 기저함수 $B_n(s)$ 의 가중치이다. 식 (4)는 행렬을 이용하여 식 (5)처럼 나타낼 수 있다.

$$x(s) = \mathbf{B}(s)^T \mathbf{Q}_x \quad (5)$$

여기서 $\mathbf{B}(s)$ 와 \mathbf{Q}_x 의 의미는 다음과 같다.

$$\mathbf{B}(s) = [B_0(s), B_1(s), \dots, B_{N_B-1}(s)]^T, \quad (6)$$

$$\mathbf{Q}_x = [x_0, x_1, \dots, x_{N_B-1}]^T \quad (7)$$

매개변수를 이용한 곡선 $\mathbf{r}(s)$ 은 행렬을 이용하여 식 (8)처럼 표현된다.

$$\mathbf{r}(s) = \mathbf{U}(s)\mathbf{Q} \quad (8)$$

여기서 $U(s)$ 와 Q 는 식 (9)의 의미를 가지고 있다.

$$U(s) = I_2 \otimes B(s)^T = \begin{pmatrix} B(s)^T & 0 \\ 0 & B(s)^T \end{pmatrix}, Q = \begin{pmatrix} Q_x \\ Q_y \end{pmatrix} \quad (9)$$

식 (9)에서 연산자 \otimes 는 Kronecker Product를 의미한다. 제어점 벡터 Q 는 제어점들의 x 좌표 값들로 구성된 Q_x 와 y 좌표 값들로 구성된 Q_y 로 이루어진 열벡터이다.

객체의 경계를 찾기 위해서 추정된 곡선 $r(s)$ 와 목표 곡선 $r_d(s)$ 사이의 모양 차이를 나타내는 식 (10)과 같은 곡선의 왜곡 정도를 정의하였다.

$$\|r(s) - r_d(s)\|^2 = \frac{1}{L} \int_0^L |r(g(s)) - r_d(s)|^2 ds \quad (10)$$

여기서 L 은 s 축 위에서의 전체 길이이다. 그러므로, 추정된 곡선이 목표 곡선과 정확히 같다면 식 (10)은 이론적으로 0의 값을 가질 것이다. 추정된 곡선과 목표 곡선의 매개 변수에 따라 비교하려는 곡선의 위치가 일치하지 않는다면, 임의의 Active Contour 알고리즘은 부적합해 질 수 있다. 이러한 문제를 해결하기 위하여 식 (10)에는 $g(s)$ 를 사용하였다. 두 곡선의 비교되는 부분이 서로 일치하도록 조절하기 위한 추상적인 함수이다. 이런 매개변수에 관한 문제는 두 곡선의 위치를 서로 비교하여 식 (10)을 최소화하는 가능한 매개변수를 주어진 조건에서 찾는 것이다. 이 문제는 3.4 절에서 다룬다.

3.3 모양 공간 모델(Shape Space Model)

초기 곡선이 목표 곡선으로 수렴해 갈 때, 곡선의 변화를 기술하기 위해서 제어점 벡터의 차원보다 작은 차원을 갖는 모양 공간을 이용하는 것이 효과적이다. 모양 공간은 $N_x \ll N_Q$ 차원의 벡터들로 구성되며, 모양 공간은 기본 곡선에서 가능한 곡선의 변형들의 집합으로 정의된다. 모양의 골격을 유지하면, 제어점의 변이를 낮은 차원의 모양 공간으로 제한하는 것은 타당하다. 제한되지 않은 제어점 벡터 Q 는 불안정한 Active Contour를 생성시킬 수 있다 [8].

모양 공간에서 곡선의 변화는 모양 공간 벡터 X 의 제어점 벡터 Q 로의 선형 사상으로 나타낼 수 있다.

$$Q = WX + Q_0 \quad (11)$$

W 는 $N_Q \times N_x$ 의 모양 행렬이며, Q_0 는 초기 곡선의 제어점 벡터이다. 모양 공간 벡터 X 는 초기 곡선으로부터의 변화를 묘사한다.

본 논문에서는 6 변수의 Affine 모델($N_x = 6$)을 사용하여 곡선의 변화를 기술한다. Affine 모델은 사용자에게 의해서 정의된 초기 곡선 $r_0(s)$ 로부터 가능한 모든 선형 변환을 표현할 수 있다.

$$r(s) = u + Mr_0(s) \quad (12)$$

$u = (u_1, u_2)^T$ 는 2 차원 벡터로 이동을 나타내며, M 은 2×2 행렬이다. 그러므로, M 과 u 는 초기 곡선 $r_0(s)$ 의 변환을 기술하는데 6개의 자유도를 제공한다. 식 (12)는 초기 곡선 Q_0 와 식 (13)의 모양 행렬을 이용하여

식 (11)로 나타낼 수 있다.

$$W = \begin{pmatrix} 1 & 0 & Q_{x0} & 0 & 0 & Q_{y0} \\ 0 & 1 & 0 & Q_{y0} & Q_{x0} & 0 \end{pmatrix} \quad (13)$$

W 의 각 컬럼은 모양 공간을 구성하는 기저 벡터를 형성한다. X 의 요소와 식 (12)의 요소들과의 관계는 식 (14)로 표현된다.

$$X = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T \quad (14)$$

3.4 최소화 알고리즘

본 절에서는 식 (10)의 비용함수를 최소화시켜 객체의 경계를 발견하는 Active Contour 알고리즘을 설명한다. 모양 공간은 곡선의 변화를 제한하고, 곡선이 영상의 특징점과 정합될 때 곡선의 부드러운 변화를 유지하도록 한다. 영상의 특징점으로부터 Spline 곡선 형태의 목표 곡선 $r_d(s)$ 을 얻을 수 있다고 가정하면, 목표 곡선은 $r_d(s) = U(s)Q_d$ 로 표현될 것이다. 추정된 곡선 $r(s)$ 를 6 변수의 Affine 모델의 모양 공간에 나타내고 근사화된 특징점 곡선을 외부 에너지로부터 얻는다면, Active Contour 알고리즘의 문제는 식(15)으로 나타낼 수 있다.

$$\min_x \|r - r_d\|^2 = \min_x \|WX + Q_0 - Q_d\|^2 \quad (15)$$

식 (15)의 해 $X = X^*$ 는 의사역행렬(Pseudo-Inverse) $W^+ = (W^T W)^{-1} W^T$ 에 의해서 얻을 수 있다.

$$X^* = W^+ (Q_d - Q_0) \quad (16)$$

식 (16)에서 X^* 는 모양 공간에서 제어점 벡터 Q 와 Q_d 와의 차이를 최소화 한다. 이것은 X^* 가 모양 공간으로의 투사(Projection)를 의미한다.

식 (10)을 최소화하는 곡선 정합은 매개변수의 변화에 따른 곡선의 변화에 매우 민감하다. 두개의 곡선이 서로 매우 비슷하더라도, 두 곡선이 매개 변수에 따라 곡선의 변화가 다르다면 식 (10)은 큰 값을 갖게 된다. 따라서 $g(s)$ 를 통하여 매개변수를 재조절하여 식 (10)을 계산하는 방법으로 문제를 변형하여 매개변수에 따른 문제를 해결할 수 있다. 그러나 식 (10)의 최적의 $g(s)$ 를 구하는 것은 매우 어렵다. 식 (10)의 $g(s)$ 에 관한 문제를 풀기 위하여 두 곡선이 매우 유사하다고 가정하면, $\|r(g(s)) - r_d(s)\|^2$ 는 $[r(s) - r_d(s)] \cdot n(s)$ 로 대치하여 매개변수에 따른 문제를 해결 할 수 있다 [8]. 여기서 $n(s)$ 는 $r(s)$ 의 접선방향에 수직인 벡터이다. 식 (10)의 적분을 함으로 바꾸고 $n(s)$ 를 모양의 차이를 구하는데 사용한다면, 최소화되어야 하는 비용함수는 식 (17)로 표현된다.

$$\|r - r_d\|^2 \approx \frac{1}{N} \sum_{i=1}^N [(r_d(s_i) - r(s_i)) \cdot n(s_i)]^2 \quad (17)$$

N 은 s 축에서 등간격으로 표본화된 점들의 수이다. 식 (17)을 초기 곡선 r_0 와 모양 공간 벡터로 표현하면 식 (18)로 재정의된다.

$$\begin{aligned} \|r - r_d\|^2 &\approx \frac{1}{N} \sum_{i=1}^N [(r_d(s_i) - r_0(s_i)) \cdot n(s_i) \\ &\quad - n^T U(s_i) W(X - X_0)]^2 = \frac{1}{N} \sum_{i=1}^N \delta_i^2 \end{aligned} \quad (18)$$

가중치 w_i 를 식 (18)에 포함한 일반적인 형식은 식 (19)로 표현된다.

$$\|r - r_d\|^2 = \left(\sum_{i=1}^N w_i \right)^{-1} \sum_{i=1}^N w_i \delta_i^2 \quad (19)$$

특징점 곡선을 동일한 호길이 간격으로 표본 추출을 하여야 한다. 그러나 실질적으로는 s 축 위에서 등간격으로 추출되므로 곡선 상에서 등간격을 보상하기 위하여 식 (20)과 같은 가중치를 사용한다.

$$w_k = |r'_k(s_k)| \quad \text{where } r'(s) \equiv \frac{dr}{ds} \quad (20)$$

식 (19)를 최소화하고 추정된 모양 공간 벡터 X^* 를 구하기 위해 Least Square Method를 사용한다. 최소값 X^* 는 $\partial \|r - r_d\|^2 / \partial X = 0$ 을 만족할 때 식 (21)을 이용하여 구한다.

$$\begin{aligned} X^* &= \left(\sum_{i=1}^N \rho_i W^T U^T n n^T U W \right)^{-1} \\ &\quad \times \left(\sum_{j=1}^N \rho_j W U^T n (r_d - r_0)^T n \right) \end{aligned} \quad (21)$$

식 (11)을 이용하여 식 (19)를 최소화하는 제어점 벡터 Q 를 얻는다. 보다 정확한 결과를 얻기 위하여 구해진 $r(s)$ 를 다시 초기 곡선 $r_0(s)$ 로 설정하여 최소화 알고리즘을 반복 수행한다.

4. 특징점 추출

Active Contour 알고리즘은 곡선의 기본적인 성질인 내부 에너지 조건을 유지하면서 외부에너지가 나타내는 영상의 특징점에 곡선을 정합시키는 것이다. 기존의 Active Contour 알고리즘은 배경이 없는 영상에서 개발되었기 때문에 복잡한 배경이 있는 영상에는 효과적인 성능을 얻지 못한다. 본 논문에서는 복잡한 배경에도 적용 가능한 외부 에너지 요소를 적용한다.

Active Contour의 외부 에너지는 영상에 의해서 주어지는 에너지로서, 일반적으로 영상 속의 객체의 윤곽선을 반영할 수 있는 Laplace 연산자나 Sobel 연산자와 같은 방법을 이용하여 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 Processing)을 수행할 때, 복잡한 영

상의 경우는 마지막 곡선이 우리가 원하지 않는 국부 최소값으로 수렴할 수 있다.

본 논문에서는 영상의 특징점을 추출하기 위하여 SUSAN (Smallest Univalve Segment Assimilating Nucleus) 연산자를 사용하여 추출된 특징점을 외부에너지로 정의 하였다. SUSAN 윤곽선 추출기는 형태학적 (Morphological) 윤곽선 추출기 또는 Canny 윤곽선 추출기에 비해 연산량이 적으면서도 복잡한 영상에 대하여 좋은 성능을 갖고 있다.

SUSAN 윤곽선 추출기의 동작은 3단계로 구성된다 [9]. 첫번째 단계는 원형 마스크의 중심을 검사되는 화소의 위치로 이동시킨다. 두번째 단계는 식 (22)에 주어진 식을 이용하여 마스크의 중심과 유사한 화소의 숫자를 구한다.

$$c(p, p_0) = \exp\left(-\frac{I(p) - I(p_0)}{t}\right)^6 \quad (22)$$

$$n(p_0) = \sum_p c(p, p_0) \quad (23)$$

여기서 p 는 원형 마스크 안에 있는 임의 화소의 위치 벡터이고 p_0 는 마스크 중심의 위치 벡터이다. 마스크 중심과 비슷한 화소의 숫자를 USAN 영역이라고 정의 한다. 그리고 식 (22)의 연산 속도를 개선하기 위하여 $I(p) - I(p_0)$ 의 모든 값에 대하여 식(22)의 연산 결과를 Lookup Table로 만든다. 마지막 단계는 영상으로부터 윤곽선을 검출하기 위하여 기하학적 임계값에서 USAN 영역을 빼는 것이다.

$$R(p_0) = \begin{cases} g - n(p_0) & \text{if } p_0 < g \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

$R(p_0)$ 는 영상의 윤곽선을 나타내고, g 는 기하학적 임계값이다. 만약 잡음이 없을 때 윤곽선을 검출하고자 한다면, 임계값은 필요가 없다. 그러나 최적의 잡음 제거를 위한 임계값은 $3n_{max}/4$ 이다 [10].

그림 2는 SUSAN 연산자의 개념을 일차원 공간에 나타낸 것이다. 그림 2에서 곡선은 영상의 밝기 변화를 나타내고, 흰색 영역은 마스크의 중심과 유사한 화소를 갖는 지역을 나타낸다. 그림 2에서 보는 것처럼 영상이 윤곽선의 중심에 있을 때 가장 적은 흰색영역을 갖는다. 따라서 윤곽선의 중심에서 $R(p_0)$ 는 제일 큰 값을 가지게 된다.

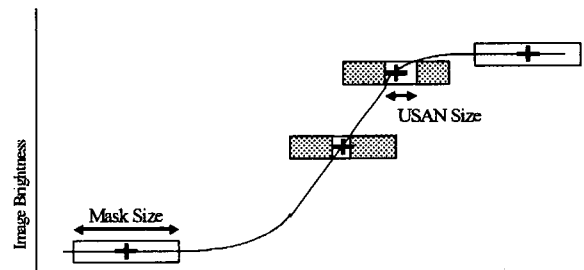


그림 2. SUSAN 알고리즘

식 (10)의 $r_d(s)$ 는 영상으로부터 추출되어야 한다.

Active Contour 알고리즘의 목적은 초기 곡선을 객체의 특징점에 정확 시키는 것이므로, 초기 곡선 주변의 영역에서 특징점을 찾는다. 식 (17)에 따라서 초기 곡선의 법선 위에서 특징점을 추출한다. 법선이 $r(s)$ 를 따라 구해진다면, 가상의 특징 곡선 $r_d(s)$ 을 따라서 $r_d(s_i)$ 의 시퀀스를 구할 수 있다.

$r(s_i)$ 는 부동 소수점의 좌표를 가지고 있으므로, 법선 벡터도 부동 소수점 요소로 구성된다. 따라서 $r_d(s_i)$ 의 밝기는 양방향 선형보간법에 의해서 구해지고, $r_d(s_i)$ 는 각 법선에서 최대값을 갖는 점으로 설정된다. 이 특징점들은 Intra-Frame 분할에 적용된다.

5. 실험 결과 및 분석

본 논문에서는 30 Hz의 프레임율을 갖는 CIF (352x288) 동영상을 사용하여 제안된 영상분할 알고리즘의 동작을 실험하였다. 그림 3은 MPEG-4 Visual Part 표준의 Annex에서 제시된 자동 분할 알고리즘 [1]의 실험 결과이다. 일반적으로 자동 분할은 수십 프레임 동안 과도응답(Transient Response)을 갖는다. 이 과도응답 시간 동안 비디오 객체를 찾기 때문에, 과도응답 시간 동안의 분할 결과는 의미가 없다. 또한, 자동 분할 알고리즘은 복수의 객체를 갖는 영상으로부터 개별 객체를 분리할 수 없다.

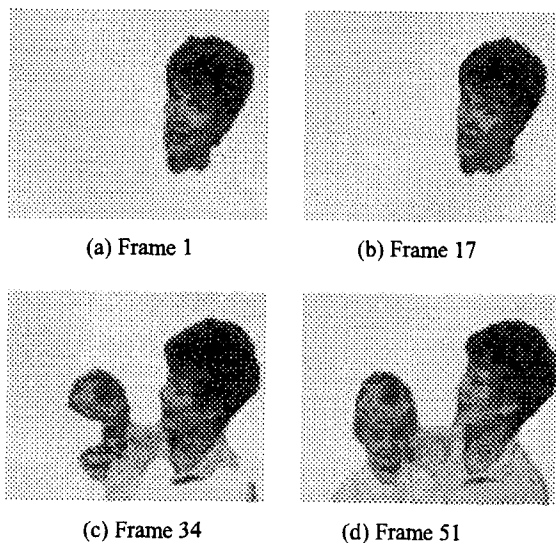


그림 3. 과도응답 시간 동안의 자동분할 결과

그림 4(a)는 사용자가 마우스와 같은 입력장치를 통하여 입력한 초기 곡선이다. 그림 4(b)는 SUSAN 연산자를 사용하여 얻은 경사도 영상이다. 기하학적 임계값에서 USAN영역의 크기의 값의 차를 표시하기 때문에 윤곽선 부분에서 가장 밝은 색을 보인다. 그림 4(c)에 보여진 검색 영역은 전체 초기 곡선을 법선 벡터 방향으로 확장하고 축소하여 얻은 특징점 추출을 위한 검색영역이다. 그림 4(c)에서 백색 선 위의 점들은 $r(s_i)$ 에 대한 $r_d(s_i)$ 의 후보점들이다. 그림 4(d)는 제안된 B-Spline Snake 알고리즘을 사용하여 얻은 Intra-Frame 분할의 최종 결과이다. 제안된 알고리즘은 곡선

의 변화를 기술하기 위하여 모양 공간 벡터를 사용하여, 저수준 영상처리로부터 생성되는 불필요한 특징점을 무시할 수 있다. 이는 곡선이 잡음에 의해 원하지 않는 위치로 이동하는 것을 막아주므로, 기존의 Active Contour 알고리즘이 주변의 잡음에 너무 민감한 특성 때문에 단순 배경 영상에 적용할 수 밖에 없었던 점을 해결하는 요인이 된다.



그림 4. Intra-Frame 분할 결과

그림 5의 그림들은 Inter-Frame 분할의 실험 결과이다. 그림 5에서는 그림 4(d)의 제어점을 이용하여 1번 프레임에서 객체의 제어점을 추정하고, 추정된 제어점을 1번 프레임의 초기 제어점으로 설정한 다음 Active Contour 알고리즘을 이용하여 1번 프레임의 객체를 구한다. 이 과정을 영상 시퀀스의 끝까지 반복적으로 수행한다. 그림 5는 전체 영상 중에서 18번 프레임과 36번 프레임에서 정합된 곡선으로부터 얻어진 비디오 객체이다. 연속된 영상의 작은 변화는 그림 4(c)의 검색영역 안에서 감지 되므로, 본 논문에서는 Inter-Frame의 분할을 위하여 제어점을 다음 프레임으로 투사할 때 움직임벡터 정보를 사용하지 않는다. 그림 4와 그림 5에서 반자동 영상분할 알고리즘은 과도응답을 생성하지 않는다.

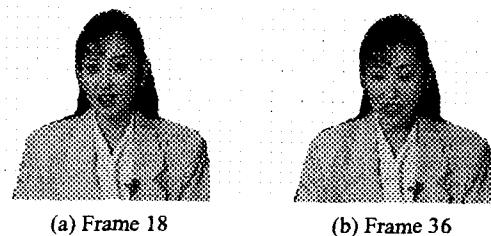
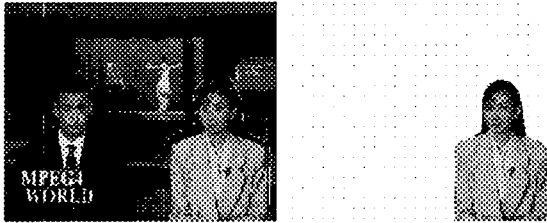


그림 5. Inter-Frame 분할 결과

그림 3의 결과에서처럼 자동 분할 방식은 프레임 간의 차이를 이용하므로, 여러 개의 움직임 객체가 한 화면에 나타나면 개별 객체를 분리해낼 수가 없다. 그

러나 반자동 분할 알고리즘에서는 원하는 객체의 주변에 초기곡선을 설정하고, Active Contour 알고리즘을 적용하면 원하는 객체만을 얻을 수 있다. 그림 6은 NEWS 영상에서 여자 아나운서 주변에 초기 곡선을 그리고 제안된 B-Spline Snake 알고리즘을 통하여 얻은 결과이다.



(a) 초기 곡선 (b) 추출된 객체
그림 6. 복수의 객체로부터 단일 객체 추출

그림 7은 그림 5에서와 같이 그림 6에서 Intra-Frame 분할을 통해서 정의된 객체에 대해 Inter-Frame 영상분할을 수행한 결과를 매 18프레임마다 표시한 것이다.

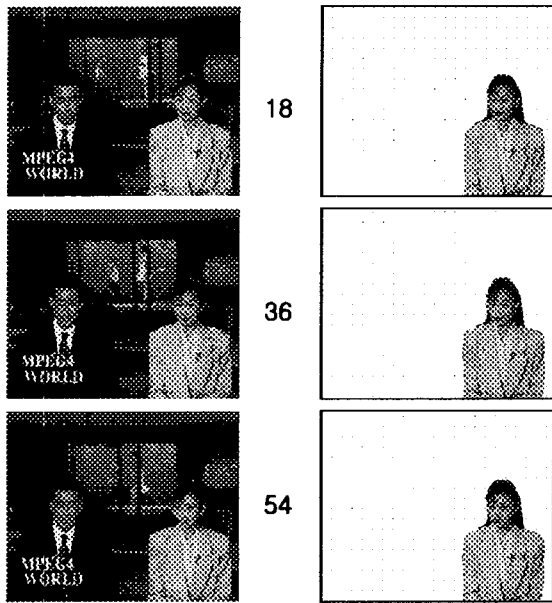


그림 7. NEWS 영상에 대한 Inter-Frame 분할 결과

6. 결론

본 논문에서는 영상 시퀀스에서 비디오 객체를 추출하기 위하여 Intra-Frame 분할과 Inter-Frame 분할로 구성된 새로운 반자동 알고리즘을 제안하였다. 제안된 알고리즘은 복잡한 배경으로부터도 객체의 윤곽선을 추출하기 위하여 SUSAN 연산자를 사용하였다. 또한 곡선의 변화를 묘사하기 위하여 모양 공간 벡터를 사용하였기 때문에 저수준 영상 처리에서 발생할 수 있는 잘못 추출된 특징점을 효과적으로 무시할 수 있다.

제안된 반자동 영상 분할 알고리즘은 영상 객체를 찾기 위하여 과도응답 시간을 요구하지 않으며, 단일 프레임에 여러 개의 움직임 객체가 있을지라도 원하는 객체를 선별하여 추출할 수 있다. 본 논문은 효과적으로 객체를 추출하기 위해 B-Spline 함수를 이용한 Active Contour 알고리즘을 사용하였다.

감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트 워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

참고 문헌

- [1] ISO/IEC FDIS 14496-2: "Information technology - generic coding of audio-visual objects, Part 2: visual," *ISO/IEC JTC1/SC29/WG11*, Oct. 1998.
- [2] M. Kim, J. Choi, D. Kim, H. Lee, C. Ahn and Y. S. Ho, "VOP generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuit and System for Video Technology*, vol. 9, no. 8, pp. 1216-1226, Dec. 1999.
- [3] 김대희, 김민호, 호요성, "MPEG-4 동영상 부호화를 위한 영상 객체 추출 기법", 전자공학회지, 제26권, 제7호, pp. 714-723, 1999.
- [4] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models," *First International Conference on Computer Vision*, pp. 259-269, 1987.
- [5] A. Amimi, T. Weymouth and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. 12, no. 9, pp. 855-867, Sept. 1990.
- [6] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol. 55, No. 1, pp. 14-26, Jan. 1992
- [7] J. D. Foley, A. Dam, S. K. Feiner, J. F. Hughes and R. L. Phillips, *Introduction to Computer Graphics*, Addison-Wesley, New York, 1995.
- [8] A. Blake and M. Isard, *Active Contours*, Springer, London, 1998.
- [9] S. M. Smith, "Flexible filter neighborhood designation," *Proc. 13th Int. Conf. on Pattern Recognition*, vol.1, pp. 206-212, 1996.
- [10] S. M. Smith, "SUSAN Nonlinear Noise Reduction," <http://www.fmrib.ox.ac.uk/fsl/susan>.
- [11] D. Kim and Y.S. Ho, "A User-Assisted Segmentation Algorithm using B-Spline Curves," *SPIE Visual Communications and Image Processing*, pp. 734-744, Jan. 2001.
- [12] D. Kim and Y.S. Ho, "Shape Training for Video Object Segmentation," *Int'l Workshop on Very Low Bitrate Video (VLBV) Coding*, pp. 186-190, Oct. 2001.