

論文2000-37SP-7-5

동영상에서의 모양 정보 부호화를 위한 정점 선택 알고리즘

(Inter-frame vertex selection algorithm for lossy coding of shapes in video sequences)

徐琮烈*, 金慶重**, 姜文基*

(Jong-Yeul Suh, Kyong Joong Kim, and Moon Gi Kang)

요약

본 논문에서는 동영상에 효과적으로 적용 가능한 정점 기반 모양 정보 부호화 알고리즘을 제안한다. 제안한 방법은 영상 사이에 존재하는 시간적 중복성을 해결하기 위해 현재 영상의 외곽선으로부터 정점을 선택할 뿐 아니라 이미 근사화 되어 부호화된 이전 영상으로부터도 정점을 선택하여 상대적으로 부호화하였으며 고정된 오차 한계 이내에서 비트 발생량이 최소가 되도록 정점을 선택한다. 이는 코드표현 방법이 정해져 있을 때 비율-왜곡 측면에서 최적이다. 정점 선택은 가중치를 갖는 방향그래프에서의 최단 경로 탐색으로 모델링되어 최적의 정점을 선택하도록 하였으며 실험 결과 기존의 모양 정보 부호화 방법보다 적은 비트율을 요구하였다. 제안한 방법은 실시간 부호화를 필요로 하지 않는 저장 매체를 위한 객체 기반 부호화 시스템 및 웹을 통한 동영상 전송에 이용될 수 있다.

Abstract

The vertex-based boundary encoding scheme is widely used in object-based video coding area and computer graphics due to its scalability with natural looking approximation. Existing single framebased vertex encoding algorithm is not efficient for temporally correlated video sequences because it does not remove temporal redundancy.

In the proposed method, a vertex point is selected from not only the boundary points of the current frame but also the vertex points of the previous frame to remove temporal redundancy of shape information in video sequences. The problem of selecting optimal vertex points is modeled as finding shortest path in the directed acyclic graph with weight. The boundary is approximated by a polygon which can be encoded with the smallest number of bits for maximum distortion. The temporal redundancy between two successive frames is efficiently removed with the proposed scheme, resulting in lower bit-rate than the conventional algorithms.

* 正會員, 延世大學校 電氣컴퓨터工學科

(Department of Electric & Computer Engineering, Yonsei University)

** 正會員, 現代電子 通信部門 端末機

(Telecommunication Business personal Equipment Division, Hyundai Electronics Industries Co., Ltd.)

接受日字:1999年6月3日, 수정완료일:2000年6月9日.

I. 서론

최근 정보 통신 기술과 영상 압축 기술의 발달로 정보 통신의 대상도 화상을 포함한 멀티미디어 통신 중심으로 변화하고 있다. 이러한 화상 정보의 효율적인 처리를 위해 객체 단위의 조작이 가능한 형태가 요구되고 있다. 이러한 객체 단위의 조작을 위한 방법으로

객체 기반 부호화 기법이 탄생하였다^[1-6]. 객체 기반 부호화기에서 모양 정보는 부호화 모드에 관계 없이 항상 전송해야 하는 정보이므로 이에 대한 효율적인 부호화는 필수적이다. 또한 컴퓨터 그래픽을 통해 만들어진 합성 영상의 효과적인 표현과 저장에도 효율적인 모양 정보 부호화(shape coding) 알고리즘에 대한 연구가 필요하며 CAD 및 영상 인식 시스템에서도 객체의 모양 정보를 효과적으로 표현하고 저장하는 알고리즘이 요구된다.

모양 정보 부호화 방식은 크게 영역 부호화 방식과 정점 부호화 방식으로 나뉘어진다. 영역 부호화 방식은 영역을 비트맵이나 사진트리로 표현하여 부호화 하는 방식이다. 이 방법의 장점은 복잡한 형태를 쉽게 표현할 수 있으며, 하드웨어 구현이 용이하다는 장점이 있다. MPEG4에서는 구현의 용이성 때문에 영역 표현방식의 일종인 CAE 알고리즘이 채택되었다^[7]. 그러나 객체의 확대나 축소가 이루어지는 경우에 시각적으로 심각한 왜곡을 수반한다.

정점 부호화 방식(vertex-based shape coding)은 객체의 외곽선을 직선을 이용한 다각형 또는 스플라인 곡선으로 표현하여 이러한 직선이나 곡선을 이루는 정점을 부호화하는 방식이다^[8-11]. 정점 부호화 방식은 상당부분 왜곡을 허용하는 경우에도 시각적으로 자연스러운 근사화가 가능하고, 객체의 확대나 축소가 이루어지는 경우에도 시각적인 왜곡을 최소화할 수 있다는 장점이 있다. 정점 선택은 먼저 객체 내에서 가장 길이가 긴 축을 탐색한 뒤 축을 구성하는 두 개의 점으로부터 시작한다^[10,11]. 이 방법은 근사화 오차가 정해진 최대 허용 오차 이내가 되도록 반복적으로 정점을 선택하는 방식으로 근사화 오차를 최소화 하는데 초점을 맞추고 있으며 정점 선택으로 인한 객체의 모양 정보의 비트율은 고려하지 않았다. 따라서 객체의 모양 정보를 부호화할 때 부호화 효율을 높여주는 효과적인 방법이 필요하다.

모양 정보 부호화에 비율-왜곡 이론(rate-distortion theory)을 적용한 경우로는 G. Schuster의 정점 부호화 알고리즘을 들 수 있다^[12,13]. 이 알고리즘은 직선을 이용하여 외곽선을 근사화 할 때 주어진 허용오차 이내에서 최적의 정점을 선택하는 알고리즘으로 정점의 선택과정을 방향 그래프(directed acyclic graph)로 모델링하고 여기에 최단 경로 알고리즘을 적용하였다. 이 방법은 하나의 영상에서의 모양 정보 부호화에는 효율

적이지만 시간적인 중복성을 제거하지 않기 때문에 동영상에 적용하기에는 적합하지 않았다. F. W. Mayer 등에 의해 직선 대신 스플라인 곡선을 사용하여 부호화 효율을 높이는 시도가 있었지만 근본적인 정점 선택 알고리즘은 변화가 없었다^[14]. 또한, 시간적 예측을 통해 시간적 중복성을 제거하는 방법을 제시한 연구가 있었지만 성능향상이 크지 않았고, 최적 정점 선택 방법은 제시되지 않았다^[15]. MPEG-4에서도 동영상에서의 정점 부호화에 대한 연구가 진행되었지만 최소 비트율을 발생시키는 정점 선택 방법은 제시되지 않았다^[16].

본 논문에서는 동영상 부호화기에서 주어진 허용 오차 이내에서 현재 영상에서 최적의 정점을 선택할 뿐 아니라 이미 부호화된 이전 영상의 최적 정점에 대응시켜 시간적 중복성을 제거하는 프레임간 모양 정보 부호화 알고리즘을 제안한다. 제안한 정점 선택 알고리즘은 직선을 이용한 다각형 근사화에 근거하고 있지만 스플라인 곡선 모델에도 적용시킬 수 있다.

II. 정지 영상을 위한 최적 정점 부호화

영상에서 객체의 모양 정보를 부호화하는 방법으로 외곽선을 다각형으로 근사화 (polygonal approximation)하여 외곽선 대신 다각형의 정점을 부호화하는 방법이 있다. 효율적인 부호화를 위해서는 다음과 같은 방법을 통하여 적은 차수의 다각형으로 근사화 할 수 있다.

$B = \{b_0, \dots, b_{N_B-1}\}$ 를 외곽선의 점을 나타내는 순서집합이라 할 때, b_j 는 B의 j 번째 점이며, N_B 는 B에 속한 점의 개수이다. 폐곡선의 경우 $b_0 = b_{N_B-1}$ 이 된다. $P = \{p_0, \dots, p_{N_P-1}\}$ 은 B를 근사화한 다각형의 정점으로 구성된 순서집합이다. p_k 는 P의 k 번째 정점이며, n_p 는 정점의 개수이다. 다각형의 K 번째 변은 정점 p_{k-1} 에서 시작하여 p_k 에서 끝난다. P는 순서집합이므로 배열 방법과 정점 집합이 주어지면 다각형을 유일하게 정의할 수 있다.

정점 p_{k-1} 이 주어졌을 때, p_k 를 부호화하는데 필요한 비트량은 $r(p_{k-1}, p_k)$ 로 정의한다. 전체 다각형을 부호화하는 데 필요한 비트량 $R(p_0, \dots, p_{N_P-1})$ 은 다음 식으로 정의할 수 있다.

$$R(p_0, \dots, p_{N_P-1}) = \sum_{k=0}^{N_P-1} r(p_{k-1}, p_k) \quad (1)$$

이 때, $r(p_{-1}, p_0)$ 는 시작점의 절대 좌표를 부호화하는 데에 필요한 비트량이다. 폐곡선의 경우 마지막 점과 시작점은 동일하므로 부호화할 필요가 없다. 따라서, $r(p_{N_p-2}, p_{N_p-1})$ 은 0의 값을 갖는다.

근사화 이전의 외곽선의 점들로부터만 정점을 선택하도록 $p_k \in B$ 로 제한한다면, 다각형의 정점 p_{k-1} 와 $p+k$ 를 연결하는 선분인 k 번째 변은 외곽선의 일부분인 $l+1$ 개의 점으로 이루어진 $\{b_j = p_{k-1}, b_{j+1}, \dots, b_{j+l} = p_k\}$ 를 근사화 한다. 따라서, 근사화 오차는 다음에 정의할 선분의 근사화 오차 $d(p_{k-1}, p_k)$ 로 나타낼 수 있다. 다각형의 근사화 오차는 모든 선분의 근사화 오차의 최대값으로 나타낼 수 있다^[2]. $d(p_{k-1}, p_k, t)$ 가 p_{k-1} 와 p_k 를 잇는 선분과 임의의 점 t 와의 최단 거리라고 하면, 거리는 다음의 식으로 나타낼 수 있다.

$$d(p_{k-1}, p_k, t) = \frac{|(t_x - p_{k-1,x})(p_{k,y} - p_{k-1,y}) - (t_y - p_{k-1,y})(p_{k,x} - p_{k-1,x})|}{\sqrt{(p_{k,x} - p_{k-1,x})^2 + (p_{k,y} - p_{k-1,y})^2}} \quad (2)$$

이 때, 점 x 와 y 는 각각 어떤 점의 x 축과 y 축 값을 의미한다. 부분 외곽선 $\{b_j = p_{k-1}, b_{j+1}, \dots, b_{j+l} = p_k\}$ 와 선분 (p_{k-1}, p_k) 의 최대 절대 거리는 다음과 같다.

$$d(p_{k-1}, p_k) = \max_{t \in \{b_j = p_{k-1}, b_{j+1}, \dots, b_{j+l} = p_k\}} d(p_{k-1}, p_k, t) \quad (3)$$

다각형 전체의 오차는 최대값 연산자에 근거하여 다음과 같이 정의할 수 있으며

$$D(p_0, \dots, p_{N_p-1}) = \max_{k \in [0, \dots, N_p-1]} d(p_{k-1}, p_k) \quad (4)$$

이 때, $d(p_1, p_0)$ 는 0이며 이 최대값은 주어진 최대 허용 오차 D_{max} 보다 작아야 하며 결과적으로 정점의 선택 문제는 다음과 같은 조건을 갖는 최적화 문제로 볼 수 있다.

$$\begin{aligned} & \min \\ & p_0, \dots, p_{N_p-1} \quad R(p_0, \dots, p_{N_p-1}), \text{ subject to:} \\ & D(p_0, \dots, p_{N_p-1}) \leq D_{max} \end{aligned} \quad (5)$$

III. 제안한 방법

G. Schuster가 제안한 최적 정점 선택 알고리즘^[12,13]은 단일 프레임 내에서 모양을 부호화하는데 있어서는 최적이지만 영상간 중복성을 고려하지 않았기 때문에 동영상에 적용할 경우 효과적이지 못한 단점이 있다. 본 논문에서는 동영상에서 정점을 이용하여 모양을 부호화할 때 시간축상의 영상의 중복성을 고려한 최적 정점 선택 알고리즘을 제안한다.

1. 프레임간 모양 정보 부호화 방법

본 논문에서는 다음과 같은 표기를 사용한다. $B' = \{b'_j: j=0, \dots, N'_B-1\}$ 은 시간 t 에서의 외곽선의 점을 나타내는 순서 집합이다. b'_j 는 B' 의 j 번째 점이며, N'_B 는 B' 에 속한 점의 개수이다. 폐곡선의 경우 $b'_0 = b'_{N'_B-1}$ 이 된다. $P' = \{p'_k: k=0, \dots, N'_P-1\}$ 는 B' 를 근사화 한 다각형의 정점으로 구성된 순서 집합이다. p'_k 는 P' 의 k 번째 정점이며, N'_P 는 정점의 개수이다. 다각형의 k 번째 변은 정점 p'_{k-1} 에서 시작하여 p'_k 에서 끝난다. P' 는 순서 집합이므로 배열 방법과 정점 집합이 주어지면 다각형을 유일하게 정의할 수 있다.

시간 t 에서의 순서집합 $B' = \{b'_j: j=0, \dots, N'_B-1\}$ 를 근사화 하여 부호화 하고자 할 때, 어떤 모양 B^{01} 을 이루는 점 $\{b'^{-1}_j: j=0, \dots, N'^{-1}_B-1\}$ 이 정점 $P'^{-1} = \{p'^{-1}_k: k=0, \dots, N'^{-1}_P-1\}$ 로 부호화 되어 있다고 가정한다. 이 때, B' 를 부호화하기 위해 선택되는 새로운 정점의 순서집합 P' 는 B' 와 P'^{-1} 의 원소로부터 선택 된다. 만일 어떤 정점 $p'^{-1}_k \in P'^{-1}$ 가 $b'_j \in B'$ 를 대신하여 오차 한계 D_{max} 를 만족시키면서 근사화 시킬 수 있으면, p'^{-1}_k 는 순서집합 P' 에 그대로 사용될 수 있다. 이 때, 정점 p'^{-1}_k 는 순서집합 P'^{-1} 에서의 순서로서 부호화할 수 있다. 정점의 순서를 부호화하는 것은 정점을 절대적 또는 상대적 좌표로 부호화하는 것 보다 일반적으로 효율적이다. P'^{-1} 의 원소로 대치할 수 없거나 정점의 좌표를 직접 부호화하는 것이 효율적인 경우에는 b'_j 가 사용된다. 이와 같은 영상간 모양 정보 부호화 방법은 이전 영상에서의 정점을 부호로 표현하는 방법에 따라 정점 선택 알고리즘은 크게 달라지게 된다.

다각형의 한 정점은 B' 로부터 선택되었을 경우 이전 정점의 위치와 상관도가 매우 높기 때문에 상대적인 좌표로 부호화하는 것이 효율적이며, P'^{-1} 로부터 선택

된 경우에도 가장 최근에 선택된 $p_k^{i-1} \in P^{i-1}$ 에 대한 P^{i-1} 에서의 상대적인 순서로 부호화하는 것이 효율적이다. 따라서 모든 정점은 상대적인 값으로 부호화하는 것으로 가정한다. 정점 P_{k-1}^i 이 주어졌을 때, P_k^i 를 부호화하는데 필요한 비트량은 $r(p_{k-1}^i, p_k^i)$ 로 정의한다. 전체 다각형을 부호화하는 데 필요한 비트량 $R(p_0^i, \dots, p_{N_p^i-1}^i)$ 은 다음 식으로 정의할 수 있다.

$$R(p_0^i, \dots, p_{N_p^i-1}^i) = \sum_{k=0}^{N_p^i-1} r(p_{k-1}^i, p_k^i) \quad (6)$$

이 때, $r(p_{k-1}^i, p_k^i)$ 는 시작점의 절대 좌표를 부호화하는 데에 필요한 비트량이다. 폐곡선의 경우 마지막 점은 시작점과 같으므로 부호화할 필요가 없다. 따라서, $r(p_{N_p^i-2}^i, p_{N_p^i-1}^i)$ 은 0의 값을 갖는다. 비트량은 세부적인 부호화 방법에 따라 크게 달라질 수 있으며 결과적으로 영상간 정점 선택도 다음과 같은 최적화 문제로 모델링 될 수 있다.

$$\begin{aligned} \min_{p_0^i, \dots, p_{N_p^i-1}^i} R(p_0^i, \dots, p_{N_p^i-1}^i), \text{ subject to:} \\ D(p_0^i, \dots, p_{N_p^i-1}^i) \leq D_{\max} \end{aligned} \quad (7)$$

2. 최적 정점 선택 알고리즘

정점 선택 알고리즘의 목표는 왜곡이 허용 최대 오차 이내이며 최소한의 비트량으로 부호화 될 수 있는 정점을 갖는 다각형을 찾는 것이다.

다각형의 어떤 한 정점이 p_{k-1}^i 이고 이 정점까지 다각형을 부호화하기 위해 필요로 하는 비트량이 $R_{k-1}(p_{k-1}^i)$ 이라면, 다음의 정점 p_k^i 의 선택은 이전에 선택된 정점들인 p_0^i, \dots, p_{k-2}^i 와는 독립적이다. 비트 발생량은 한 선분에 의해 발생하는 비트량 $r(p_{k-1}^i, p_k^i)$ 와 선분의 오차 $d(p_{k-1}^i, p_k^i)$ 의 함수인 점화식으로 표현할 수 있으므로 이 성질이 사실임을 알 수 있다. 이를 식으로 표현해보면,

$$R(p_k^i) = R_{k-1}(p_{k-1}^i) + w(p_{k-1}^i, p_k^i) \quad (8)$$

$$w(p_{k-1}^i, p_k^i) = \begin{cases} \infty & : d(p_{k-1}^i, p_k^i) > D_{\max} \\ r(p_{k-1}^i, p_k^i) & : d(p_{k-1}^i, p_k^i) \leq D_{\max} \end{cases} \quad (9)$$

이 점화식은 $R_i(p_k^i)$ 을 0으로 초기화 시켜 주어야 한

다. $w(p_{k-1}^i, p_k^i)$ 의 정의는 최대 허용 오차를 만족시키지 못하는 다각형의 비트 발생량은 무한대임을 나타낸다. 위 식으로부터 전체 다각형의 비트 발생량은 다음과 같음을 알 수 있다.

$$R_{N_p^i-1}(p_{N_p^i}^i) = R(p_0^i, \dots, p_{N_p^i}^i) \quad (10)$$

이 때, 외곽선의 첫번째 점인 b_0^i 나 이전 영상의 첫번째 정점 p_0^{i-1} 이 첫번째 정점인 p_0^i 로 선택되어야 한다.

식 (8)를 이용하면 식 (7)에서 주어진 문제는 가중치를 갖는 방향 그래프 $G=(V, E)$ 에서 최단 경로를 찾는 문제로 생각할 수 있다. 이 때, V 는 그래프 정점의 집합이며, E 는 그래프 연결의 집합이다. 영상 내 정점 선택 알고리즘에서는 $V=B$ 이나, 영상간 정점 선택 알고리즘에서는 외곽선 상의 점과 이전 영상을 근사화 한 다각형의 정점이 현재 영상의 정점 후보가 되므로, 그래프의 정점 집합이 다르게 정의되어야 한다. 또한 그래프의 정점을 잇는 각각의 연결 비용은 유일하게 정의되어야 한다. 만일 그래프의 연결이 유일하게 정의되지 않고 경로에 중속하는 함수가 된다면 최단 경로 알고리즘을 적용할 수 없다. 행 방향으로 N_p^{i-1} 개의 노드 (node)와 열 방향으로 N_B^i 개의 노드가 2차원적으로 배열되어 있는 총 $N_p^{i-1} \times N_B^i$ 개의 노드를 갖는 그래프를 생각해보자. 이 그래프의 노드 $v_{(i,j)} \in V$ 는 i 행 j 열에 존재하는 노드이며, p_i^{i-1} 또는 b_j^i 를 의미한다. 이 때, 한 노드가 반드시 한 점을 대표하도록 해야 한다. 따라서 모든 노드 $v_{(i,j)}$ 는 대표하는 점에 따라 p_i^{i-1} 를 대표하는 부노드 $v_{(i,j)}^C$ 와 b_j^i 를 대표하는 부노드 $v_{(i,j)}^I$ 로 분리되며, 이는 최단 경로 알고리즘에 의해 최적의 정점 선택을 하는 것을 가능하게 한다. 그림 1에 최단 경로 알고리즘의 구현을 위한 그래프를 보여주고 있다. 각 그래프의 모든 노드는 2개의 부노드로 분리되며 각 부노드는 각 정점으로서의 연결 모드, 즉 대응 모드 (correspondence mode)와 삽입 모드(insertion mode)를 의미한다. C 와 I 는 대응과 삽입을 의미하는 부호화 모드 $M=C, I$ 의 원소로 정의한다. p_0^{01} 에 해당하는 첫번째 행에서는 임의의 i 에 대해 p_i^{i-1} 를 대표하는 노드는 존재할 수 없으므로, 첫번째 행을 제외한 모든 노드에서 노드의 분리가 일어난다. 따라서, 모든 부노드의 수는 $2N_p^{i-1} \times N_B^i$ 개가 된다. 부노드를 포함하여 노드를

다시 정의하면 노드 $v_{(i,j)}^m$ 은 $N_p^{i-1} \times N_B$ 그래프에서 i 행 j 열에 위치하며 부호화 모드가 m 인 노드이다.

$$v_{(i,j)}^m = \begin{cases} p_i^{i-1} & : m=C \\ b_j^i & : m=I \end{cases} \quad (11)$$

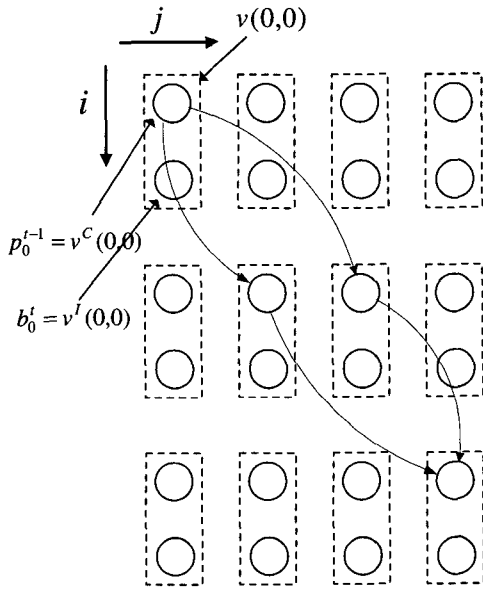


그림 1. 최적 정점의 선택을 위한 그래프 표현
Fig. 1. Graph for selecting optimal vertex points.

모든 그래프의 정점은 다각형의 정점이 될 수 있으며, 그래프상의 정점 사이를 잇는 연결은 다각형에서 변에 해당한다. 방향을 갖는 연결 $(v_{(i,j)}^m, v_{(k,l)}^n) \in E$ 는 그래프의 정점 $v_{(i,j)}^m$ 에서 시작하여 $v_{(k,l)}^n$ 에서 끝나는 연결을 의미한다.

정점 $v_{(i,j)}^m$ 에서 정점 $v_{(k,l)}^n$ 까지의 차수가 인 경로는 순서집합 $\{v_0, \dots, v_H\}$ 이며, 이때 $h=1, \dots, H$ 에 대하여 $v_{(i,k)}^m = v_0, v_{(k,l)}^n = v_H$ 이다. 경로의 차수는 경로상에 존재하는 연결의 수로 정의한다. 경로의 길이는 다음과 같이 정의된다.

$$\sum_{h=1} w(v_{h-1}, v_h) \quad (12)$$

이 때, $w(v_{(i,j)}^m, v_{(k,l)}^n) : E \rightarrow R$ 은 가중치 함수이며, 식 (9)와 같이 다음과 같이 정의된다.

$$w(v_{(i,j)}^m, v_{(k,l)}^n) = \begin{cases} \infty & : d(v_{(i,j)}^m, v_{(k,l)}^n) > D_{\max} \\ r(v_{(i,j)}^m, v_{(k,l)}^n) & : d(v_{(i,j)}^m, v_{(k,l)}^n) \leq D_{\max} \end{cases} \quad (13)$$

위의 정의에 의해 D_{\max} 이상의 근사화 오차를 갖는 선분을 포함하는 모든 경로의 길이는 무한대가 된다. 따라서 최단 경로 알고리즘은 이러한 경로는 선택하지 않는다. p_0^i 에서 시작하여 $p_{N_p}^i$ 에서 끝나는 길이가 무한대가 아닌 모든 경로의 길이는 다각형으로 표현할 때 필요한 비트량과 같다. 따라서, 이러한 경로 중 최단경로는 가장 적은 비트로 표현될 수 있는 다각형에 대응되며, 이는 식 (7)의 해가 된다.

식 (13)에서 연결비용 $r(v_{(i,j)}^m, v_{(k,l)}^n)$ 은 연결의 종류에 따라 달라진다. 즉, 두 종류의 연결에 의해 발생하는 코드의 표현 방법이 다르며, 비용 또한 크게 달라진다. 부호화 모드에 따라 연결 비용은 다음과 같이 구할 수 있다.

$$r(v_{(i,j)}^m, v_{(k,l)}^n) = \begin{cases} r_C(k-i) & : n=C \\ r_i(p_i^{i-1}, b_j^i) & : m=C, n=I \\ r_l(b_j^i, b_l^i) & : m=I, n=I \end{cases} \quad (14)$$

여기서 $r_C(a)$ 는 정수 a 를 가변 길이 부호화 하는 것이다. 이전 영상의 정점이 현재 영상의 정점에 모두 그대로 대응된다면 $a=1$ 이 된다. $i \neq k$ 이므로, $a=0$ 이 되는 경우는 발생하지 않는다. $r_l(b, c)$ 는 두 점 b 와 c 가 있을 때, c 를 b 에 대한 상대적인 좌표로 나타내고 이를 부호화할 때 발생하는 코드의 길이이다.

다각형을 순서집합으로 정의하였으므로, 마지막으로 선택된 외곽선 상의 점이 b_j^i 일 때, $j > l$ 인 점 b_l^i 는 선택한다면 이것은 최적의 선택이 아니라는 것은 명백하다. 마지막으로 선택한 이전 영상의 정점이 p_i^{i-1} 일 때, $i > k$ 인 정점 p_k^{i-1} 를 선택한다면 이 또한 최적의 선택이 아니다. 반면에 연속된 정점이 외곽선 상의 순서와 다르게 할당된 다각형에서는 근사화 이후 급격한 변화를 보일 수 있다. 따라서, $j < l$ 가 아니거나 $i < k$ 인 모든 순서쌍 $(v_{(i,j)}^m, v_{(k,l)}^n)$ 은 부적절한 연결이라는 제한 조건이 덧붙여져야 한다. 따라서 연결 집합 E 는 다음과 같이 정의할 수 있다.

$$E = E_C \cup E_I$$

$$E_C = \{(v_{(i,j)}^m, v_{(k,l)}^n) \in V^2 : \forall ((i < k) \wedge (j < l) \wedge (n = C))\} \quad (15)$$

$$E_l = \{(v_{(i,j)}^m, v_{(k,l)}^n) \in V^2 : \forall ((i=k) \wedge (j < l) \wedge (n = l))\}$$

이 때, E_C 에 속하는 연결을 대응 모드 연결이라고 정의하고, E_l 에 속하는 연결을 삽입 모드 연결이라고 정의한다. 식 (15)과 같은 제한 조건은 주어진 정점 집합에 의해 유일한 다각형을 얻을 수 있게 하며, 이와 같이 정의된 그래프에 DAG 최단 경로 알고리즘을 적용하여 최적의 정점을 선택할 수 있다.

시작 노드에서 어떤 노드 $v_{(i,j)}^m$ 에 이르는 최소 경로 비용은 $R(v_{(i,j)}^m)$ 으로 정의할 때, 시작 노드에서 $v_{(k,b)}^n$ 에 이르는 경로 비용 $R(v_{(k,b)}^n)$ 는 식 (8)에 따라, 다음과 같은 점화식으로 나타낼 수 있으며

$$R(v_{(k,l)}^n) = R(v_{(k,b)}^m) + w(v_{(i,j)}^m, v_{(k,l)}^n) \quad (16)$$

이 점화식이 성립하기 위해서는 시작점을 부호화 하는데 필요한 비트량인 시작 노드의 비용이 정의되어 있어야 한다. 영상 내 부호화 시에는 항상 경계선을 이루는 점의 집합 $B = \{b_0, \dots, b_{N_b-1}\}$ 에서 첫번째 원소인 b_0 를 시작점으로 선택하였으나, 영상 간 부호화 시에는 $B' = \{b'_j : j=0, \dots, N_b'-1\}$ 의 첫번째 원소인 b'_0 를 시작점으로 선택하거나, $P^{l-1} = \{p_k^{l-1} : k=0, \dots, N_p^{l-1}-1\}$ 의 첫번째 원소인 p_0^{l-1} 를 선택하여 부호화한다. 따라서 대응 모드에서의 시작점은 $v_{(0,0)}^C = p_0^{l-1}$ 이며, 삽입 모드에서의 시작점은 $v_{(0,0)}^I = b'_0$ 이다.

최적의 정점 선택을 위해서는 위와 같은 두개의 시작 노드를 고려하여야 하지만 두개의 시작 노드를 갖는 그래프 상에서는 최단 경로 알고리즘을 적용할 수 없으므로, 두개의 시작 노드에 대해 각각 최단 경로 알고리즘을 따로 적용시켜야 한다. 이 때, 시작 노드 $v_{(0,0)}^S$ 는 $v_{(0,0)}^C$ 또는 $v_{(0,0)}^I$ 가 된다. 가상의 시작 노드를 v_{source} 라고 할 때, 시작 점의 부호화 비용은 다음과 같이 초기화 된다.

$$R(v_{(0,0)}^S) = r(v_{source}, v_{(0,0)}^S) \quad (17)$$

$v_{(0,0)}^C$ 의 경우 $v_{(-1,-1)}^C$ 이후에 나타나는 대응 코드로서 표현 가능하며, $v_{(0,0)}^I$ 는 원점에 대한 코드로서 나타낼 수도 있으며, p_0^{l-1} 에 대한 상대 좌표로도 나타낼 수 있다. 대부분의 경우 시작점에 대해서도 영상의 시간적

중복성이 존재하므로 $v_{(0,0)}^I$ 는 p_0^{l-1} 에 대한 상대 좌표로 나타내는 것이 좀 더 효율적이다. 따라서 각각의 경우 실제 비용은 다음과 같다.

$$r(v_{source}, v_{(0,0)}^C) = r(v_{(-1,-1)}^C, v_{(0,0)}^C) = r_C(l) \quad (18)$$

$$r(v_{source}, v_{(0,0)}^I) = r_I(p_0^{l-1}, b'_0) \quad (19)$$

DAG 최단 경로 알고리즘을 적용하기 위해서는 탐색의 시작 노드와 종료 노드가 결정 되어 있어야 한다. 제안한 알고리즘에서는 시작 노드는 $v_{(0,0)}^I$ 또는 $v_{(0,0)}^C$ 의 두 가지 경우가 존재하므로 각각의 시작노드에 대해서 최단 경로 알고리즘을 별도로 수행하여야 하며, 하나의 시작 노드에 대하여 종료 노드는 여러 개가 존재할 수 있다. 따라서 종료노드는 $(k=0, \dots, N_p'-1)$ 에 대하여 $v_{k, N_b'-1}^S$ 가 될 수 있다. 최단 경로 알고리즘에서는 $v_{(0,0)}^S$ 에서 $v_{k, N_b'-1}^S$ 에 이르는 경로 중에서 최적의 정점을 선택하는 경로를 각각 찾을 수 있다. 이러한 복수개의 종료노드까지 이르는 최단 경로 중에서 경로 비용이 가장 작은 하나의 최단 경로를 선택한다. 따라서, 최종적인 최단 경로의 경로 비용 R^* 는 다음과 같다.

$$R^* = \min_{k=0, \dots, N_p'-1} R(v_{(k, N_b'-1)}^S) \quad (20)$$

이 비용은 최적의 정점을 선택하였을 경우 발생하는 비트량이 된다. 최종적인 종료 노드는 다음과 같이 구할 수 있다.

$$(n^*, k^*) = \arg \min_{(k=0, \dots, N_p'-1)} R(v_{(k, N_b'-1)}^S) \quad (21)$$

이 때, $v_{k^*, N_b'-1}^S$ 가 최종적인 종료 노드가 된다.

$q(v_{(i,j)}^m)$ 이 어떤 노드에 이르는 최적 경로를 기억하기 위한 역추적 포인터라고 하면, 알고리즘은 다음과 같다.

먼저 모든 노드에 대해서 시작점으로부터 그 노드까지의 최소 경로 비용을 무한대로 설정한다. 다음에는 식 (18)을 이용하여 대응 모드의 시작점을 부호화하는데 필요한 비트량을 구하여 그 비용을 설정한다. 다음 단계에서는 다각형의 변을 이루는 선분이 시작하는 정점이 될 수 있는 후보 노드를 차례로 선택한다. 이 때, 실제로 경로가 존재하지 않는 노드는 항상 경로비용이

무한대가 되므로 최단 경로의 후보로부터 제외될 수 있으며, 이를 통해 연산량을 크게 줄일 수 있다. 다음으로 연결 집합(edge set) E 에 속하는 모든 연결에 대해 선분이 끝을 이룰 수 있는 정점 후보를 선택하여 각각의 연결의 비용 $w(v_{(i,j)}^m, v_{(k,l)}^n)$ 를 계산한다. 가장 최근의 정점이 $v_{(i,j)}^m$ 일 때 후보 정점 $v_{(k,l)}^n$ 까지 이르기까지의 새로운 비트 발생량 $R^*(v_{(i,j)}^m) + w(v_{(i,j)}^m, v_{(k,l)}^n)$ 가 지금까지 점 $v_{(k,l)}^n$ 까지 이르는 데 드는 최소 비트율 보다 작다면 $R^*(v_{(k,l)}^n) = R^*(v_{(i,j)}^m) + w(v_{(i,j)}^m, v_{(k,l)}^n)$ 라는 식을 통해 그 값이 점 $v_{(k,l)}^n$ 에 이르는 최소 비트율로 설정된다. 이 때, $R^*(v_{(k,l)}^n)$ 를 발생시키는 데 사용한 이전 정점이 $v_{(i,j)}^m$ 이므로, $v_{(k,l)}^n$ 의 역추적 포인터 $q(v_{(k,l)}^n)$ 를 $v_{(i,j)}^m$ 로 설정한다. 이 알고리즘은 정점 $v_{(i,j)}^m$ 의 비트율 $R^*(v_{(i,j)}^m)$ 가 주어졌을 때, 이후의 정점 선택 ($v_{(k,l)}^n, i \leq k < N_p, j < l < N_B'$)는 이전의 정점 선택 ($v_{(e,f)}^r, 0 \leq e < i, 0 \leq f < j$)과는 독립적이기 때문에 최적의 해를 도출한다.

최적 경로 $\{p'_0, \dots, p'_{N_p-1}\}$ 는 식 (21)에 의해 얻어진 종료 노드로부터 다음과 같은 재귀적인 방법을 통하여 포인터 $q(v_{(i,j)}^m)$ 를 추적함으로써 구할 수 있다.

$$(p'_{N_p-1} = v_{(k', N_b-1)}, p_0 = b_0)$$

$$p'_{k-1} = q(p'_k), \quad k = N_p' - 1, \dots, 1 \quad (22)$$

제한한 방법에 의한 영상간 모양 부호화의 예를 그림 2에 나타내었으며, 이에 대한 그래프 모델링을 그림 3에 나타내었다. 그림에서 알 수 있듯이 그래프의 각 노드는 2개의 부노드로 분리되며 위쪽의 부노드는 대응 모드를 의미하며 아래쪽의 부노드는 삽입 모드를 의미한다. DAG 최단 경로 알고리즘을 적용할 때 시작 점은 p_0 로 부호화 된다. 대응 모드로 부호화할 때 이전 프레임에서 이미 부호화된 p_1, p_2, p_3 이 선택되며, 대응 모드로 부호화할 경우 주어진 오차 조건을 만족할 수 없는 점들, 즉 삽입 모드로 부호화할 점들은 b_{12} 과 b_{16} 이 된다. 다각형은 폐곡선이므로 마지막 점은 시작 점과 일치하게 된다. 따라서 b_{16} 를 부호화하기 위한 비트량이 0이며 이 점은 부호화되지 않는다. 모양을 근사화하는데 사용된 다각형의 각각의 선분은 그래프 상에서의 연결과 일치한다.

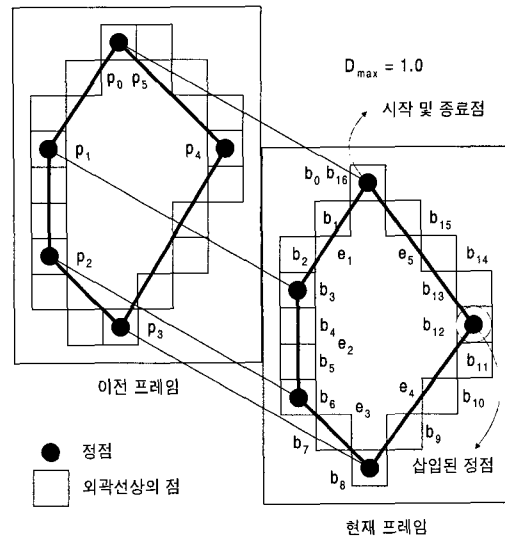


그림 2. 영상간 정점 부호화의 예
Fig. 2. Example of the inter-frame vertex coding.

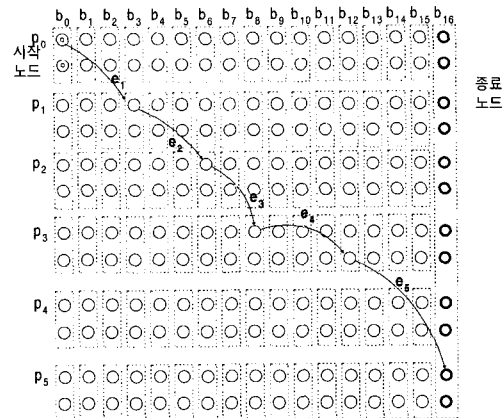


그림 3. 그림 2에서의 정점 선택을 위한 그래프 모델링
Fig. 3. The graph representation of the shape coding of frame 2 in Fig. 2 using the optimal method.

IV. 실험결과

정점 선택 알고리즘의 성능을 평가하기 위해 다음과 같은 실험 방법을 사용하였다. 실험 대상 영상은 176×144 크기의 MPEG4 표준 실험 영상을 사용하였으며, 모양 정보는 MPEG4 표준 실험 마스크 영상을 사용하였다. 그림 4과 5에 이 실험에서 사용한 "news"와 "weather" 동영상과 각 영상을 부호화하기 위한 마스크 영상을 나타내었다. 이 실험에서는 각 영상에 대하여

30장의 프레임을 사용하여 부호화하였다. 마스크 영상에 내측 외곽선 추적 알고리즘을 적용하여 8방향 체인 코드를 구하였다^[17]. BMA를 사용하여 움직임 보상을 수행하였으며, 탐색 범위는 $(-7 \leq x \leq 7, -7 \leq y \leq 7)$ 이다.

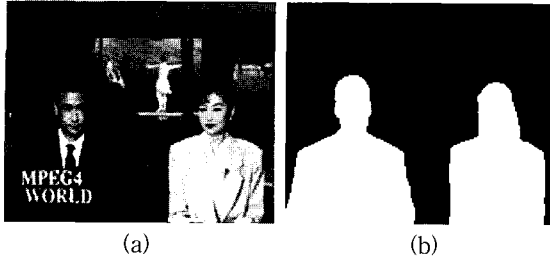


그림 4. "News" 영상 (a) 원영상 (b) 마스크 영상
Fig. 4. "News" sequence (a) original image (b) segmentation mask.



그림 5. "Weather" 영상 (a) 원영상 (b) 마스크 영상
Fig. 5. "Weather" sequence (a) original image (b) segmentation mask.

기존의 영상 내(intra-frame) 정점 부호화 알고리즘(알고리즘 1), MPEG4 FDIS의 CAE 알고리즘(알고리즘 2)과 제안한 방법(알고리즘 3)을 비교하였다. 다각형 근사화를 통한 정점 부호화 방법의 경우에는 $D_{max} = \sqrt{2}$ 로 설정하였고, CAE 알고리즘의 경우 오차의 측정 방법이 정점 부호화 방법과 다르므로 동일한 오차한계를 부여할 수 없었으나, Alpha 임계치를 128로 설정함으로써 정점 부호화 방법과 비슷하거나 좀 더 큰 오차를 허용하도록 하여 비교 방법에 객관성을 부여하였다.

표 1. 상대 좌표의 부호화 길이
Table 1. The code length of relative address.

상대 좌표	부 호	부호화 길이
~-9	'10'+(URA)	10
-8 ~+7	'0'+(4 bits)	5
+8~	'11'+(URA)	10

* URA : Unsigned Relative Address(상대 좌표의 절대값)

표 2. 대응 순서의 코드표

Table 2. Codes for correspondence order.

부호(Code word)	상대적 대응순서	부호(Code word)	상대적 대응순서
0	1	1110110	9
10+상대좌표	삽입	1110111	10
110	2	1111000	11
1110000	3	1111001	12
1110001	4	1111010	13
1110010	5	1111011	14
1110011	6	1111100	15
1110100	7	1111101	16
1110101	8	1111110	17

알고리즘 1은 각 프레임을 독립적으로 정점 부호화한 방법이며 여기서 사용하는 코드는 x 방향의 상대좌표와 y 방향의 상대좌표를 각각 표 1에 의하여 표현한다. 제안한 방법(알고리즘 3)에서는 대응과 삽입 모드 모두 상대적인 값을 사용하여 부호화 하였으며 여기에 사용된 코드는 대응 모드의 경우 표 2와 같으며, 삽입 모드의 코드에 대해서는 표 1의 코드의 앞에 삽입 모드 구분자 '10'(2bit)를 붙여준다. 즉, 대응 모드의 경우 상대적인 대응 순서값을 사용하였으며 삽입 모드의 경우 이전 정점에 대한 상대적인 좌표값을 사용하여 부호화 하였다. 본 논문의 목적은 부호 표현 방법 자체가 아니라 정해진 부호 표현 방법 내에서 최적의 정점을 선택하는 방법을 제시하는 데 있으므로, 다른 부호 표현 방법이 사용되었을 때에도 제안한 정점 선택 알고리즘을 적용할 수 있다. 따라서 보다 개선된 부호 표현 방법을 사용할 경우 더 향상된 결과를 얻을 수 있다.

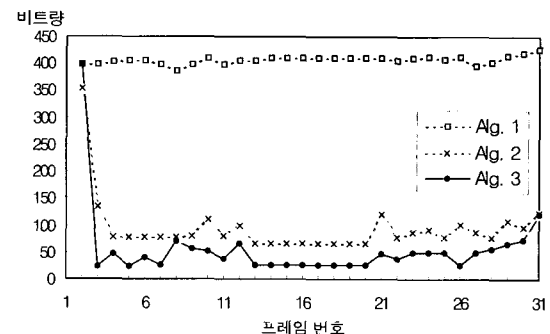


그림 6. 'news' 영상에서의 비트 발생량 비교
Fig. 6. The comparison of the number of coded bits for 'news' sequence.

그림 6에서 'news' 영상에 대하여 알고리즘 1, 2, 3을 적용했을 때의 모양 정보에 대한 비트 발생량을 그래프로 나타내었다. 'news' 영상은 객체의 모양의 변화가 적어 시간적 중복성이 높은 영상이다. 알고리즘 1은 시간적 중복성을 제거하지 않으므로 전체 프레임에 걸쳐 비슷한 비트발생률을 나타내었다. 알고리즘 2와 알고리즘 3은 시간적 중복성을 제거하므로 2번째 프레임부터 비트 발생량이 현격하게 줄어드는 것을 알 수 있다. 알고리즘 2보다는 제안한 알고리즘 3이 더 적은 비트를 발생시키는 것을 알 수 있다.

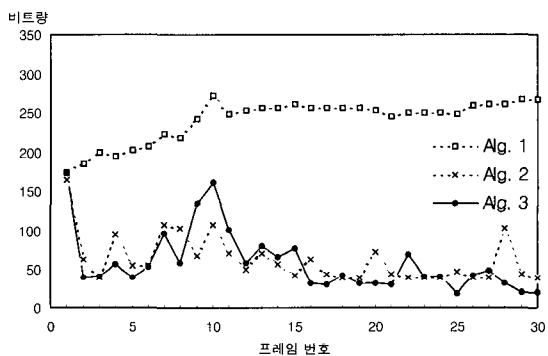


그림 7. 'weather' 영상에서의 비트 발생량 비교
Fig. 7. The comparison of the number of coded bits for 'weather' sequence.

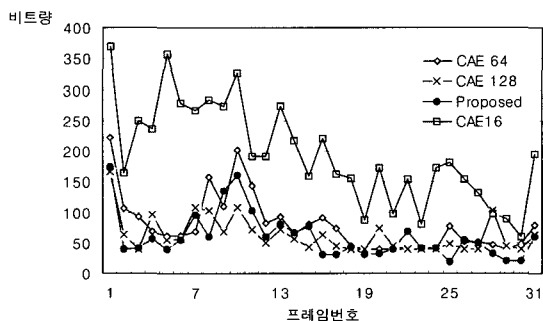


그림 8. CAE의 Alpha-Threshold의 변화에 따른 성능 변화와 제안한 방법과의 비교 (Weather sequence)
Fig. 8. The comparison among the proposed method and CAE algorithms with different Alpha-Threshold.

그림 7에서는 'weather' 영상에 대하여 알고리즘 1, 2, 3을 적용했을 때의 모양 정보에 대한 비트 발생량을 그래프로 나타내었다. 'weather' 영상은 'news' 영상에 비하여 모양 정보의 변형이 심한 영상이다. 'news' 영상과 마찬가지로 알고리즘 1에 비하여 알고리즘 2, 3의

비트 발생량이 매우 적게 나타난다. 제안한 알고리즘 3이 알고리즘 2보다 비트발생량이 약간 적게 나타나지만, 그 차이는 'news' 영상에서 보다 좁혀졌음을 알 수 있다.

CAE 알고리즘과의 보다 다각적인 비교를 위해 CAE 알고리즘에서 Alpha 임계치를 다른 값을 사용하는 경우에 대해서도 실험을 하였다. Alpha 임계치가 작을수록 허용 왜곡 수치가 작아지므로 보다 많은 비트량이 요구된다. 그림 8의 그래프에서 이와 같은 결과가 잘 나타나고 있다. CAE64는 Alpha 임계치가 64인 경우이며, CAE16, CAE128 역시 Alpha 임계치가 각각 16, 128인 경우이다.

표 3. 각 알고리즘의 총 발생 비트량 비교(30 프레임)
Table 3. The comparison of the total encoded bits. (30 frames)

실험 영상	News 영상	Weather 영상
intra-frame 정점 선택	12928	7241
CAE (Alpha 임계치 : 128)	2794	1823
제안한 방법	1674	1773

표 3에서는 두 가지 영상의 30프레임에 대한 총 비트 발생량을 알고리즘 별로 비교하였다. 제안한 알고리즘 3은 기존의 단일 영상의 정점을 이용한 외곽선 근사화 방법(알고리즘 1)에 비하여 월등하게 우수한 성능을 나타내는 것을 알 수 있다. 시간적 중복성이 큰 'news' 영상의 경우 변형이 심한 'weather' 영상 보다 제안한 알고리즘의 성능이 더 우수하다. MPEG-4의 CAE 알고리즘에 비해서도 같은 왜곡 조건하에서 비트 발생량 측면에서 제안한 방법이 우수한 성능을 나타내었음을 알 수 있다.

V. 결론

본 논문에서는 동영상에서 객체의 모양 정보를 부호화하기 위해 주어진 왜곡에 대하여 최소량의 비트를 발생시키는 정점 선택 알고리즘을 제안하였다. 동영상의 모양 정보는 일반적으로 시간적 중복성이 크므로 이를 효과적으로 제거하면 부호화 효율을 높일 수 있

다. 본 논문에서는 현재 영상을 부호화할 때 이전 영상에서 이미 부호화된 정점을 사용함으로써 영상의 중복성을 고려한 영상간 정점 선택 알고리즘을 제안하였다. 영상간 정점 선택은 주어진 정점 집합을 노드로 한 DAG에서의 최소값을 갖는 연결을 선택하는 문제로 모델링하였다. 모양을 근사화 하는 다각형의 각 정점은 그래프에서의 노드로 나타낼 수 있으며 각 노드는 부호화 방식에 따라 두개의 부노드로 나뉘게 된다. 이 때, 최소값을 갖는 연결에 선택된 정점이 다각형의 정점이 되며 이러한 방법으로 선택된 정점은 비율-왜곡 측면에서 최적의 성능을 보인다.

제안한 방법은 실험 결과 다른 모양 정보 부호화 알고리즘에 비해 상당한 수준의 비트율의 감소를 보였으며 이는 제안한 방법의 우수성을 나타낸다고 할 수 있다. 본 논문에서 제안한 방법을 사용하여 선택된 정점에 대하여 보다 효과적인 부호화 방법을 적용할 경우 객체의 모양 정보의 비트 발생량은 더욱 줄어들 수 있다. 따라서 선택된 정점에 대한 부호화 표현 방법의 최적화 문제가 향후 과제로 남아있다.

참 고 문 헌

- [1] H. Musmann, M. Htter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, Vol. 1, pp.117-128, Oct. 1989.
- [2] M. Htter, "Object-oriented analysis-synthesis coding based on moving two-dimensional objects," *Signal Processing: Image Communication*, Vol. 2, pp.409-428, 1990.
- [3] M. Htter, "Optimization and efficiency of an object-based analysis-synthesis coder," *IEEE Trans. Circ. Syst. Video Technol.*, Vol. 4, No. 2, pp.181-194, April 1994.
- [4] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit-rates," *IEEE Trans. Circ. Syst. Video Technol.*, Vol. 4, No. 3, pp.228-235, June 1994.
- [5] J. Ostermann, "Object-based analysis-synthesis coding based on the source model of moving rigid 3D objects," *Signal Processing: Image Communication*, Vol. 6, pp.143-161, 1994.
- [6] J. Ostermann, "Object-based analysis-synthesis coding (OBASC) based on the source model of moving flexible 3-D objects," *IEEE Trans. Image Processing*, Vol. 3, No. 5, Sep. 1994.
- [7] "Final Draft of International Standard", ISO/IEC JTC1/SC29/WG11 Document N2502, Oct. 1998.
- [8] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, EC-10:260-268, June 1961.
- [9] J. Saghi and H. Freeman, "Analysis of the precision of generalized chain codes for the representation of planar curves," *IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI* Vol. 3, No. 5, pp. 533-539, Sept. 1981.
- [10] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Trans. Circuits and Systems*, Vol. 4, pp. 228-235, June 1994.
- [11] K. J. O'Connell, "Object-adaptive vertex-based shape coding method," *IEEE Trans. Circ. Syst. Video Technol.*, Vol. 7, pp. 251-255, Feb. 1997.
- [12] G. Schuster and A. Katsaggelos, "An optimal lossy segmentation encoding scheme," *Proceedings of the Conference on Visual Communications and Image Processing, SPIE*, pp.1050-1061, Mar. 1996.
- [13] G. Schuster and A. Katsaggelos, "An Optimal Polygonal Boundary Encoding Scheme in the Rate Distortion Sense," *IEEE Trans. Image Processing*, Vol. 7, No. 1, Jan. 1998.
- [14] F. W. Meier, G. M. Schuster and A. K. Katsaggelos, "An efficient boundary encoding scheme using B-spline curves which is optimal in rate-distortion sense," *Proceedings of International Conference on Image Processing*, 1997.
- [15] G. Heckner, "Redundance reducing coding of moving object shapes," *Signal Processing: Image Communication*, Vol. 9, No. 1, pp.91-98, 1997.

- [16] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and Rate-Distortion-Based Shape-Coding Techniques," Proceedings of IEEE, Vol. 86, No. 6, June 1998.
- [17] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall, 1993.

저 자 소 개



徐琮烈(正會員)

1994년 2월 연세대학교 전자공학과 졸업 (공학사). 1996년 8월 연세대학교 대학원 전자공학과 졸업 (공학 석사). 1996년 9월 현재 연세대학교 대학원 전기·컴퓨터 공학과 박사과정. 주관심분야 : 디지털 비디오 처리, 파노라마 합성, 컴퓨터 비전

리, 파노라마 합성, 컴퓨터 비전



金慶重(正會員)

1992년 2월 연세대학교 전자공학과 졸업 (공학사). 1994년 2월 연세대학교 대학원 전자공학과 졸업 (공학 석사). 1998년 8월 연세대학교 대학원 전자공학과 졸업 (공학 박사). 1998년 10월 1999년 10월 현대전자

정보통신 연구소 선임연구원. 1999년 10월 현재 현대전자 통신부문 단말기 SBU 선임연구원. 주관심분야 : 영상압축, 영상신호처리, 멀티미디어통신, MPEG-4



姜文基(正會員)

1986년 2월 서울대학교 전자공학과 졸업(공학사). 1988년 2월 서울대학교 대학원 전자공학과 졸업(공학 석사). 1994년 2월 Northwestern University 졸업(공학 박사). 1994년 3월 1994년 10월 Northwestern

University Research Fellow. 1994년 11월 1997년 2월 University of Minnesota, Duluth 조교수. 1997년 3월 1999년 2월 연세대학교 전자공학과 조교수. 1999년 3월 현재 연세대학교 전기·컴퓨터 공학과 부교수. 주관심분야 : 영상복원, 초고해상도 영상복원, 디지털 비디오 처리