論文98-35S-1-13

공간 영역에서의 평활한 완전 8방향 윤곽선 부호화 기법

(A Coding Technique for the Smoothly Perfect 8-connected Contours in the Spatial Domain)

趙 聖 鎬 *, 金 麟 澈 **, 李 商 郁 *

(Seong Ho Cho, Rin Chul Kim, and Sang Uk Lee)

요 약

본 논문에서는 평활한 완전 8방향(SP8C) 윤곽선이라는 새로운 개념을 도입하고, SP8C 윤곽선을 효율적으로 부호화하는 기법들에 대하여 고찰하였다. SP8C 윤곽선은 영상 영역화된 영상에 대하여 Gu[2]가 제안한 majority filter를 이용하여 윤곽선을 간략화한 다음, 윤곽선 격자상에서 윤곽선을 추출함으로써 얻을 수 있다. SP8C 윤곽선은 윤곽선을 따라 단지 3 방향으로만 움직인다는 사실을 참조하여 본 논문에서는 NDSC (neighbouring direction segment coding) 기법과 entropy 기법을 이용하여 SP8C 윤곽선을 부호화하였다. 본 논문에서 제안한 SP8C 윤곽선은 Gu등이 사용한 4방향 윤곽선에 비해 대각선 방향 윤곽선에 대해 낮은 엔트로피를 보임을 입증하였다. 또한, 모의실험 결과, 본 논문에서 고려한 윤곽선 부호화 기법은 SP8C 윤곽선을 효과적으로 부호화함을 알 수 있었다.

Abstract

In this paper, we introduce the notion of the smoothly perfect 8-conntected (SP8C) contour and describe a coding technique for the SP8C contours. Based on the simplification using the majority filter proposed by Gu[2], SP8C contours are extracted on the contour lattice from the segmented image. By noting that the SP8C contours are restricted to travel in only 3 different directions along the contours, we attempt to encode the SP8C contours using the following two techniques. The one is the neighbouring direction segment coding, while the other is to employ the notion of the entropy encoding. Computer simulations reveal that the contours can be efficiently encoded by the two techniques.

* 正會員, 서울大學校 電氣工學部 信號處理 硏究室
(Signal Processing Lab., School of Electrical Eng., Seoul National University)
** 正會員, 漢城大學校 情報電算學部
(School of Information and Computer Eng. Hansung University)
※ 본 연구는 1995년도 한국학술진흥재단의 대학부설 연구소 연구과제 연구비에 의해 연구되었음.
接受日字: 1997年9月5日, 수정완료일: 1997年11月29日

I. 서 론

초 저속 영상 전송을 위하여 객체지향 부호화 (object-oriented coding) 기법[1]이 활발히 연구되고 있다. 객체지향 부호화 기법에서는 영상 정보를 객체로 나누고, 각 객체에 대한 모양(혹은 윤곽선), 움직임, 색정보 등을 부호화하는 것이다. 이 중에서 윤곽선은 객체지향 부호화 기법에서 핵심적인 정보이므로, 이를 효율적으로 부호화하는 기법들에 대하여 많은 연구가 이루어져왔다[2,3].

윤곽선 부호화는 일반적으로 영역화된 영상(segmented image)으로부터 윤곽선 표현, 윤곽선 간략화,

그리고 부호화 등을 포함하게된다. 먼저, 영역화된 영상으로부터 윤곽선을 표현하는 기법은 크게 다음 두 가지 방법으로 나누어진다. 하나는 화소 영역(pixel domain)에서 윤곽선을 표시하는 방법이고, 다른 하나는 윤곽선 격자(contour lattice)를 이용하여 표시하는 방법이다. 화소 영역에서 윤곽선을 표시하는 방법은 윤곽선을 처리함과 동시에 윤곽선상의 화소가 어느 영역에 속하는 지를 판단해야 하는 단점이 있다. 따라서, 실제적으로 영역의 수가 많은 경우에는 화소 영역에서 윤곽선을 추출하려고 한다면, 각 윤곽선 픽셀이 어느 영역에 속하는 지에 관한 정보가 추가적으로 필요하다는 문제점이 있다. 그러나, 윤곽선 격자로 표시하는 방법은 화소와 화소 사이에 윤곽선 격자를 그림 1과 같이 정의하고, 윤곽선을 격자 1과 격자 2로 표시하는 방법으로 윤곽선 격자에 의해 화소들이 속한 영역이 명확하게 나뉘어 진다. 그러나, 윤곽선 격자로 표시하는 방법은 화소 영역에서 표시하는 방법에 비해 윤곽선을 표시하는 격자 수가 많아져 윤곽선 부호화 효율이 저하된다는 단점이 있다. 이러한 문제를 해결하기 위해, Gu [2]는 격자 1로 표시된 격자들만을 이용하여 윤곽선을 표시하였다. 이때, 윤곽선은 4방향 (4-connected) 윤곽선으로 제한된다. 그리고, [4]에서는 윤곽선 격자상에서 윤곽선을 6 방향 윤곽선으로 정의하고, 이를 효율적으로 부호화할 수 있는 방법을 제시하였다.

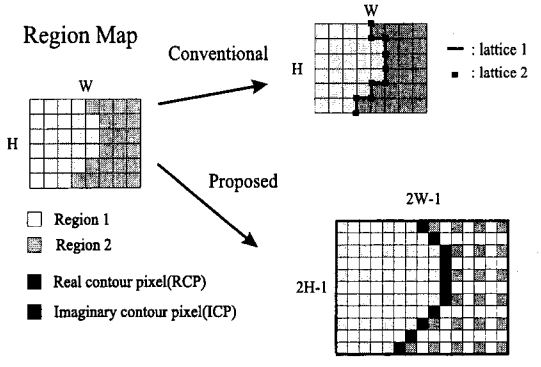


그림 1. 격자 변환
Fig. 1. Lattice conversion.

한편, 영역화된 영상으로부터 추출된 윤곽선이 지나치게 상세한 모양을 묘사하는 부분들을 포함하는 경우, 이러한 부분으로 인해 전체적인 부호화 효율이 저하될 뿐만아니라 시각적으로 좋지못한 영향을 주게된다. 따라서, 객체의 모양을 충실히 표현하면서도 상세한 윤곽선을 간략화시킴으로써 부호화 효율을 향상시키는 방법에 대해서 많은 연구가 이루어져 왔다[2, 3, 5, 6]. 윤곽선을 간략화시키는 방법으로는 윤곽선 부호화의 전처리로서 간략화를 수행하는 방법들이 있고[2, 3] 윤곽선 부호화시 어느 정도의 오차를 허용하면서 부호화하는 방법들[5, 6] 또한 소개되고 있다.

추출된 윤곽선을 부호화하는 기법은 다양하게 소개되고 있는데, 대표적인 방법은 윤곽선 방향을 따라가면서 그 방향을 부호화하는 체인 부호화이다[7, 8, 9 10, 11] 기존의 체인 부호화들은 4방향 혹은 8방향 윤곽선을 부호화하는 기법들로, 이 가운데 완전 8방향(perfect 8-connected) 윤곽선을 세그먼트(segment) 단위로 부호화하는 기법들[8, 9]이 우수한 성능을 보이는 것으로 알려져 있다.

그러나, 지금까지 소개되고 있는 윤곽선 부호화 기법들은 윤곽선 간략화, 윤곽선 표현, 부호화등을 개별적으로 처리하였다는 단점이 있다. 본 논문에서는 윤곽선 간략화 후 새로운 특성을 가지는 윤곽선을 평활한 완전 8방향(SP8C; smoothly perfect 8-connected) 윤곽선으로 정의하고, SP8C 윤곽선을 효율적으로 부호화하는 기법들에 대하여 고찰한다. SP8C 윤곽선은 majority filter [2]를 이용하여 윤곽선을 간략화한 다음, 이를 윤곽선 격자상에서 8방향 윤곽선으로 표현함으로써 얻을 수 있다. SP8C 윤곽선은 기존의 완전 8방향 윤곽선과는 달리, 윤곽선의 이동 방향이 항상 3 방향으로 제한된다. 본 논문에서는 이러한 특성을 반영하여 부호화함으로써 윤곽선 부호화 효율을 개선시킬 수 있음을 보이겠다.

Ⅱ. 평활한 완전 8방향 윤곽선

본 절에서는 영역화된 영상으로부터 평활한 완전 8방향 윤곽선(SP8C)을 얻는 방법을 설명하고, 이의 특성을 검토한다. SP8C 윤곽선은 majority 필터링을 수행한 다음 격자 변환 (lattice conversion)함으로써 얻는데, 각 과정에 대한 자세한 설명은 다음과 같다.

1. Majority 필터링

일반적인 영역화된 영상은 잡음 등의 영향으로 영역의 경계에 고주파 성분이 다수 포함되어 있다. 특히, 그림 2에 보인 바와 같은 돌출 영역(spark region)이 존재하면, 복호화된 영상의 주관적 화질이 저하될 뿐

만 아니라, 윤곽선 부호화에 많은 정보량이 요구된다는 단점이 있다. 이러한 단점을 해결하기 위해, [2]에서는 일종의 비선형필터인 majority 필터를 제안하였다.

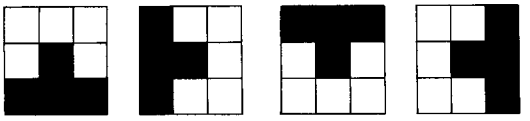


그림 2. 돌출 영역의 예
Fig. 2. Example of spark regions.

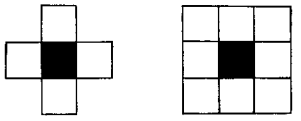


그림 3. 크기가 1인 십자형 및 정방형 구조체
Fig. 3. Cross and square structuring elements of size one.

Majority 필터링은 다음과 같이 수행된다. 먼저, 영역화된 영상에서 각 객체 (object)에게 일정한 밝기 값을 할당함으로써 라벨 영상(label image)을 얻는다. 예를 들면, 배경과 하나의 객체로 이루어져 있을 때, 배경영역에 0, 객체 영역에 255라는 값을 할당함으로써 라벨 영상을 만든다. 공간 위치 x 에서의 라벨 값을 $P(x)$ 라고 하고, x를 둘러싸는 일정한 영역을 구조체(structuring element) B 라고 하자. 그림 3에는 대표적인 구조체로서 크기가 1인 십자형(cross) 및 정방형(square)의 구조체를 예로 들었다. 만약, 구조체 B 영역내에 k 개의 서로 다른 라벨 값이 존재하고, L_i라벨 값을 가지는 화소 수를 N_i 라고 하면, majority 연산자(operator)는 다음과 같이 구조체내에 최대 빈도수를 보이는 라벨 값으로 대체하는 것으로 정의된다.

$$M(P(x))=\begin{cases} L_i & N_i > N_j, 1\le j\le k, j\neq i \\ P(x) & \text{otherwise} \end{cases} \quad (1)$$

이때, majority 필터링 σ_B 는 식 (1)로 정의된 majority 연산자 M 을 다음과 같이 수렴할 때까지 반복 적용함으로써 이루어진다.

$$\sigma_B(P(x))=M^{\infty}(P(x)) \quad (2)$$

여기서 M^{∞} 는 majority 필터링을 수렴할 때까지 반복 적용함을 나타낸다. 본 논문에서는 크기가 1인 정방형 구조체를 이용하여 식 (2)로 정의되는 majority 필터를 영역화된 영상에 적용하였다. 그 결과, 그림 2에 보인 예와 같은 1 화소 폭을 가지는 돌출 윤곽선을 없앨 수 있었다. 뿐만 아니라, 다음 절에서 설명하는 바와 같이 윤곽선의 방향 이동이 일정하게 제한되기 때문에 윤곽선을 효율적으로 부호화할 수 있다.

2. 격자 변환

본 논문에서는 다수의 객체가 존재하는 경우에도 각 객체 영역을 명확히 구분하기 위하여 윤곽선 격자 상에 윤곽선을 표시한다. 이를 위해서는 격자 변환을 통하여, 화소 영역에서 표시된 라벨 영상을 윤곽선 격자상에서 표시되는 윤곽선 영상으로 변환하여야한다. 폭이 W이고 높이가 H인 라벨 영상에 대한 윤곽선 영상은 각 화소 사이에 윤곽선 격자를 삽입하여 폭과 높이가 각각 2W-1, 2H-1인 크기로 만들어 진다. 이때, 그림 1에 보인 바와 같이 가로, 세로 모두 짝수번째에 있는 격자는 화소 영역을 나타내는데, 인접한 화소 영역간의 레이블이 바뀌는 윤곽선 격자에 255 값을 지정함으로써 윤곽선을 표시한다. 여기서 윤곽선을 나타내는 화소들을 RCP(real contour pixel)라고 정의한다. 만약, 윤곽선이 가로 혹은 세로 방향으로 이동한다면, 매 두번째 윤곽선 격자는 윤곽선 표시가 되지 않으므로 이러한 격자들을 ICP(imaginary contour pixel)라고 정의하고, RCP와 동일하게 255값을 지정하여 연속된 윤곽선을 얻는다. ICP는 단지 연속된 윤곽선을 얻기 위한 것으로, 부호화시 ICP를 부호화하지 않더라도 모양 정보를 충실히 표현할 수 있다.

본 논문에서는 이와 같은 방법을 이용하여, 그림 1에 검은색 격자로 표시한 것과 같이, 윤곽선을 완전 8방향 윤곽선으로 표현한다. 한편, Gu [2]는 그림 1의 우측 상단부에 보인 것과 같은 윤곽선 격자 구조에서 격자 1만을 이용하여 4방향 윤곽선으로 표시하였다. Gu의 방법과 제안한 방법의 차이점은 다음 절에서 구체적으로 비교하겠다.

3. SP8C 윤곽선의 특성

앞서 설명한 바와 같이, majority 필터링 후 윤곽선 격자상에서 완전 8방향 윤곽선으로 윤곽선을 표시하면, 일견, 화소 영역에서 윤곽선을 표시한 것에 비해 2배 정도 많은 격자들로 윤곽선을 표시한 것으로 보인다. 그러나, 제안한 윤곽선 표현 방법은 윤곽선을 효율적

으로 부호화할 수 있는 여러가지 특성들을 가지게되어 기존의 방법에 비해 부호화 효율을 크게 향상시킬 수 있다. 이러한 특성들은 제안한 윤곽선 표현 기법이 기존의 완전 8방향 윤곽선에 비해 보다 평활한 특성을 보이기 때문이다. 따라서, 본 논문에서는 제안한 기법에 의해 표현된 윤곽선을 평활한 완전 8방향(SP8C) 윤곽선으로 명명한다. SP8C 윤곽선의 특성을 살펴보면, 다음과 같다.

- 라벨 영상을 크기가 1인 정방형 구조체로 majority 필터링하면, 그림 2에 보인 바와 같은 돌출 영역이 제거되므로, 윤곽선 영상은 대각선 방향 이동 후에도 90° 방향 성분이 없어지게 된다. 따라서, 그림 4에 보인 바와 같이 수평/수직 방향 이동 후는 물론 대각선 방향 이동 후에도 3가지 방향으로만 이동한다.

v: SP8C 윤곽선에는 존재하지 않음
(a) after horizontel or vertical (b) after diagonal

그림 4. 평활한 완전 8 방향 윤곽선
Fig. 4. Smoothly perfect 8-connected contours.

- 윤곽선 이동은 항상 대각선 방향과 수평/수직 방향 이동이 번갈아 발생한다. 즉, 서로 다른 방향의 대각선 방향 이동, 혹은 서로 다른 방향의 수평/수직 방향 이동등은 연속적으로 발생할 수 없다.
- 동일한 대각선 방향으로 이동하는 윤곽선 화소의 갯수는 대각선 방향 이동 이전과 이후의 이동 방향과 관련되어 있다. 만약, 대각선 방향 이동 이전과 이후의 이동 방향이 같으면, 대각선 방향으로 이동하는 윤곽선 화소의 갯수는 짝수이다. 그렇지 않은 경우에는 윤곽선 화소의 갯수는 홀수이다.

Ⅲ. Entropy 비교

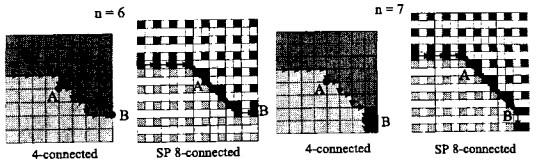
본 절에서는 majority 필터링 후 윤곽선 격자상에서 표현된 SP8C 윤곽선을 체인 부호화하는 개념을 살펴보면서, 엔트로피 측면에서 majority 필터링 후 표현되는 4방향 윤곽선[2]과 부호화 효율을 비교하겠다.

체인 부호화는 윤곽선을 따라가면서 그 방향 성분을 부호화하는 것이다. 따라서 체인 부호화 기법의 부호화 효율은 윤곽선이 이동 가능한 방향 성분을 살펴보면 알 수 있다. Gu [2]가 사용한 4방향 윤곽선과 본 논문에서 제안한 SP8C 윤곽선은 수직 및 수평 방향 이동 후, 다음 이동 방향은 모두 3 방향으로 제한되어, 방향 성분당 log 3 비트로 부호화할 수 있다. 그리고, SP8C 윤곽선과 4방향 윤곽선은 동일한 갯수의 방향 성분을 부호화하게 된다. 즉, SP8C 윤곽선은 ICP를 포함하여 4방향 윤곽선에 비해 윤곽선을 표현하는 격자 수가 2배이지만, SP8C 윤곽선의 수평 및 수직 방향 성분은 항상 2 격자 단위로 이동하므로, 실제 부호화되는 방향 성분의 갯수는 동일하다. 따라서, 수평 및 수직 방향 이동에서는 두 종류의 윤곽선은 동일한 부호화 효율을 보이게된다.

한편, 4방향 윤곽선 및 SP8C 윤곽선은 majority 필터링을 수행하기 때문에, 대각선 방향 이동 전후에는 수평 혹은 수직 방향으로만 이동할 수 있다. 예를 들면, 4방향 윤곽선 표현에서는 대각선 방향 이동은 수직 및 수평 방향 이동이 연속하여 나타나는 것이다. 이러한 경우, 이전 이동 방향과 현재 이동 방향이 다르면, 윤곽선의 다음 이동 방향은 이전 이동 방향, 현재 이동 방향중의 하나로 제한된다. 따라서, 4방향 윤곽선 표현에서 대각선 방향으로 이동할 때에는 2 방향으로만 이동하므로, log 2 비트의 정보량으로 부호화된다. 즉, 대각선 방향으로 이동하는 n 격자로 구성된 4방향 윤곽선을 부호화하는데 요구되는 비트량은 다음과 같다.

$$B_4(n) = n\log 2 \tag{3}$$

SP8C 윤곽선 표현에서는 그림 4에 제시한 바와 같이, 대각선 방향으로 이동할 때에 방향 성분은 3 방향으로 표현된다. 그런데, 대각선 방향 이동 전후의 이동은 수평 혹은 수직 방향으로의 이동으로 제한되므로, 그림 5에 나타낸 바와 같이,대각선 방향 이동 전후의 이동이 같은 경우와 같지 않은 경우 등 2경우로 나눌 수 있다. 만약, 대각선 방향으로의 이동이 시작될 때, 1비트의 flag 를 도입하여 두 경우를 구분한다면, 대각선 방향 이동 후의 다음 이동 방향을 알 수 있다. 이때, 대각선 방향 이동은 진행중인 대각선 방향, flag에 의해 표시되는 방향 등 2 방향으로 제한시킬 수 있다. 즉, SP8C 윤곽선 표현에서 대각선 방향 이동시, 1 비트의 flag를 나타낸 다음, 각 방향 성분을 log 2 비트로 부호화할 수 있다.



(a) 짝수인 경우(flag=0) (b)홀수인 경우(flag=1)

그림 5. 윤곽선 표현 기법 비교

Fig. 5. Comparison of the contour representation method.

그림 5에서는 4방향 윤곽선 및 SP8C 윤곽선 표현법으로 대각선 방향 이동시의 두가지 경우를 각각 도시한 것이다. 그림 5를 참조하면, 대각선 방향 이동으로 부호화되는 구간은 A 점 부터 B 점까지인데, 4방향 윤곽선 및 SP8C 윤곽선 표현은 동일한 갯수의 격자로 윤곽선이 표시된다. 그러나, SP8C 윤곽선의 경우에는 수직 및 수평 방향 이동과 마찬가지로 2격자 단위로 대각선 방향 이동을 표현할 수 있다.

만약, 대각선 방향으로 이동하는 윤곽선 격자 수를 n이라고 하면, 대각선 방향 이동 전후의 이동 방향이 같은 경우(flag=0)에는 n이 짝수가 되고, 전후 이동 방향이 다른 경우(flag=1)에는 n이 홀수가 된다. 이 두 경우는 flag에 의해 분별할 수 있으므로, n이 짝수인 경우 2격자단위로 $\frac{n}{2}$ 개의 방향 성분을 부호화하고, n이 홀수인 경우 $\frac{(n-1)}{2}$ 개의 방향 성분을 부호화한 다음 1 격자 만큼 추가적으로 동일한 대각선 방향으로 이동시킨다. 즉, 1 비트의 flag는 대각선 방향 이동 전후의 이동 방향에 관한 정보뿐만 아니라, 대각선 방향으로 이동하는 격자의 수에 관한 정보도 포함하게된다. 이상에서 살펴본 바에 의해, 대각선 방향으로 이동하는 n 격자로 구성된 SP8C 윤곽선을 부호화하는데 요구되는 비트량은 다음과 같다.

$$B_{SP8}(n) = 1+(n-A)\log 2 \qquad (4)$$
$$\text{where, } A=\begin{cases}0, & \text{if } n(\bmod 2)=0\\ 1, & otherwise\end{cases}$$

식 (3)와 식 (4)를 참조하면, 제안하는 SP8C 윤곽선으로 표현하는 것이 기존의 4방향 윤곽선 표현에 비해 적은 정보량으로 부호화할 수 있음을 알 수 있다. 수식적으로 대각선 방향 변화에서 약 50% 정도 비트가 적게 드는데, 이것은 SP8C 윤곽선의 장점을 이용하여 2 격자 단위로 윤곽선을 부호화하기 때문이다.

Ⅳ. 윤곽선 부호화

본 절에서는 SP8C 윤곽선을 효율적으로 부호화할 수 있는 두가지 기법들에 대하여 살펴보겠다. 첫번째 방법은 entropy 부호화[10]에 근거한 방법이고, 두번째 방법은 NDSC 기법[8]을 수정한 것이다. 본 논문에서는 SP8C 윤곽선의 성능 검증에 주안점을 두었기 때문에 영상을 배경 속에 하나의 객체만 존재하는 영상에 적용하였다. 따라서, 실제 윤곽선 부호화에 필요한 윤곽선 분기점에 대한 고려는 하지 않았다.

두가지 부호화 방법에서 시작점은 다음과 같이 부호화하였다. 제안하는 SP8C 윤곽선 표현 기법에서 시작점은 그림 1에 보인바와 같이 화소 격자 사이에 있는 윤곽선 격자로 제한된다. 따라서, 원 영상이 QCIF(176×144)인 경우, 수평방향으로 8비트, 수직방향으로 9비트의 정보로 시작점을 부호화한다. 또한, 처음 시작 방향을 가리키는 팔분 부호(octant code)는 3비트로 부호화한다.

1. Entropy 부호화

Entropy 부호화 기법에서는 시작점 및 처음 시작 방향을 부호화한 다음, SP8C 윤곽선의 특성에 따라 직선 방향 후와 대각선 방향 후의 두 가지 mode로 나누어 부호화한다. j 번째 체인 부호 μ_j 를 부호화하였을 때, entropy 부호화에서는 $j+1$번째 체인 부호 μ_{j+1} 를 다음과 같이 부호화한다.

• 직선 방향 후 ($\mu_j(\bmod 2)=0$ 인 경우)

$$Code=\begin{cases}0, & \text{if }, \mu_{j+1}=\mu_j, (\bmod 8)\\ 10+\text{flag}, & \text{if }, \mu_{j+1}=\mu_j+1, (\bmod 8)\\ 11+\text{flag}, & \text{if }, \mu_{j+1}=\mu_j-1, (\bmod 8)\end{cases} \qquad (5)$$

• 대각선 방향 후 ($\mu_j(\bmod 2)=1$ 인 경우)

$$Code=\begin{cases}0, & \text{if }, \mu_{j+1}=\mu_j, (\bmod 8)\\ 1, & \text{if }, \mu_{j+1}=\mu_j\pm 1, (\bmod 8)\end{cases} \qquad (6)$$

실험에서 부호화의 끝을 나타내는 부호는 삽입하지 않았다. 즉, 부호화하는 중, 윤곽선이 영상의 가장자리에 도달하였거나, 이미 부호화된 윤곽선을 다시 부호화하고자 할 때 부호화할 윤곽선이 끝난 것으로 판단하였다.

2. NDSC

Kaneko가 제안한 NDSC[8] 기법은 SP8C 윤곽선을 가장 효율적으로 부호화할 수 있는 방법중의 하나

이다. 본 절에서는 SP8C 윤곽선의 특성을 효과적으로 이용할 수 있도록 적절히 수정한 NDSC 기법을 설명한다. 기존의 NDSC 기법은 이웃하는 두 방향의 체인 부호로 표현되는 윤곽선 픽셀들을 하나의 세그먼트로 분리하여 세그먼트 단위로 부호화하는 알고리듬이다. 한 세그먼트에서 다른 세그먼트로 이동할 때, 다음 세그먼트의 처음 방향을 지정해주는 확장 팔분 부호(extended octant code)를 표시한 다음, 세그먼트내의 각 화소들의 방향 성분을 1 비트로 부호화한다.

그런데, SP8C 윤곽선 영상은 항상 3가지 방향만을 가지므로 이웃하는 2개의 방향으로 구성된 세그먼트 단위 부호화 기법에서 다음 세그먼트의 방향은 자동적으로 정해지게된다. 따라서, 확장 팔분 부호가 필요없다. k번째 세그먼트의 팔분 부호(octant code)를 α_k라고 하고, 이 이 세그먼트의 마지막 체인 부호를 a라고 하자. 그러면, a는 세그먼트의 팔분 부호에 의해 α_k 혹은 α_k+1 (mod 8) 중의 하나가 된다. 이때, $k+1$번째 팔분 부호 α_{k+1}는 다음과 같이 결정된다.

$$\alpha_{k+1}=\begin{cases}\alpha_k-1,(\bmod 8), & \text{if } d=\alpha_k\\ \alpha_k+2,(\bmod 8), & \text{if } d=\alpha_k+1,(\bmod 8)\end{cases} \tag{7}$$

한편, 대각선 방향 이동에서도 다음 세그먼트의 방향을 알 수 있으므로, entropy 부호화 기법과는 달리 flag를 전송할 필요가 없다. 즉, 대각선 방향 이동 직전의 체인 부호를 a라 하고, 직후의 체인 부호를 d^*라 할때, 대각선 방향의 윤곽선 화소 수 L_a는 다음 식(8)과 같이 정해진다.

$$L_d(\bmod 2)=\begin{cases}0 & \text{if } d=d^*\\ 1 & \text{otherwise}\end{cases} \tag{8}$$

따라서, NDSC 기법을 이용하여 SP8C 윤곽선을 부호화할 때, 별도의 flag를 표시할 필요없이, 항상 2격자 단위로 이동 방향을 부호화하고, 이에 따라 세그먼트 길이 또한 2격자 단위로 부호화한다. 세그먼트 길이 부호화는 [8]의 방법을 따랐다.

V. 모의 실험

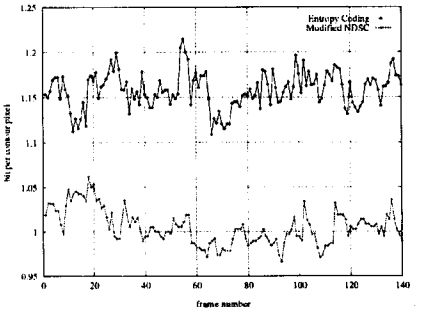
본 절에서는 제안한 기법의 성능을 평가하기 위하여, 3가지 MPEG-4 실험영상, Claire, Foreman, Mother and Daughter 을 [12] 기법으로 객체와 배경의 2 영역으로 나눈 영역화된 영상에 대하여 모의실험을 하였다. 성능은 프레임당 윤곽선을 부호화하는데 소요된 비트 수, 그리고 실제 부호화해야하는 윤곽선 화소인 RCP당 비트 수(즉, 부호화 비트수 / RCP 수)를 측정하였다.

모의실험에서 본 논문에서 제안한 윤곽선 간략화후, 윤곽선 격자상에서 표현된 SP8C 윤곽선은 제 4 절에서 설명한 두가지 부호화 기법, 즉 엔트로피 부호화 기법과 NDSC 기법을 이용하여 부호화하였다. 그림 6에서는 모의실험에서 사용한 SP8C 윤곽선 영상의 예를 보였다.

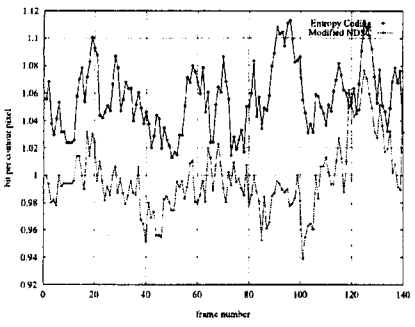
(a) Claire(10th) (b) Foreman(10th)

그림 6. SP8C 윤곽선 영상의 예
Fig. 6. Examples of the SP8C contour images.

성능 비교 대상으로는 다양한 모의 실험을 수행한 후, 다음의 두 방법을 선정하였다. 첫번째 방법은 윤곽선 간략화 과정을 수행하지 않은 완전 8방향 윤곽선을 효율적으로 부호화할 수 있는 NDSC [8] 기법이다. 두번째 방법은 제 2절에서 설명한 윤곽선 간략화 과정 후, 4방향 윤곽선을 추출한 다음 entropy 부호화 [10] 하는 기법이다. 첫번째 방법과의 비교에서는 윤곽선 간략화 과정의 이득을 살펴볼 수 있고, 두번째 방법은 Gu [2] 가 사용한 윤곽선 표현 방법과 동일하므로, 제안한 기법과 Gu의 방법을 간접적으로 성능을 비교할 수 있다.



(a) Claire sequence



(b) Foreman sequence

그림 7. 윤곽선 부호화 기법의 성능
Fig. 7. Performance of the contour coding techniques.

그림 7에는 QCIF 크기의 동영상 Claire 와 Foreman 140 프레임을 8.33 Hz로 부호화하였을 때 윤곽선을 표현하는데 소요된 비트수를 프레임 별로 제시한 것이다. 그리고, 표 1과 표 2에는 첫번째 프레임을 부호화하는데 필요한 비트량과 140프레임을 8.33Hz로 부호화하였을 때의 평균 RCP당 비트량을 각각 제시하였다.

표 1. 첫번째 프레임에 대한 윤곽선 부호화 기법들의 성능
Table 1. Performance of the contour coding techniques on the first frame.

(단위: bits)

Ifnage	Claire	Foreman	M&D
1st frame CP No.	371	482	574
Entropy Coding(4방향 윤곽선)	453	582	679
NDSC(완전8방향 윤곽선)	494	689	791
Entropy Coding(SP8C 윤곽선)	428	509	575
Modified NDSC(SP8C 윤곽선)	378	482	530

표 2. 윤곽선 부호화 기법들의 평균 성능
Table 2. Average performance of the contour coding techniques.

(단위: bits)

Image	Claire	Foreman	M&D
Entropy Coding(4방향 윤곽선)	1.232	1.178	1.209
NDSC(완전8방향 윤곽선)	1.366	1.378	1.372
Entropy Coding(SP8C 윤곽선)	1.158	1.057	1.087
Modified NDSC(SP8C 윤곽선)	1.005	1.0	0.968

실험 결과에서 보면, 완전 8방향 윤곽선을 NDSC 기법으로 부호화한 경우 RCP 당 평균 1.3 ~ 1.4 비트의 정보량이 필요한 것으로 나타났다. 윤곽선 간략화된 4방향 윤곽선을 entropy 부호화하는 방법은 완전 8방향 윤곽선을 부호화하는 방법에 비해 우수한 것으로 나타났다. 이로 부터 윤곽선 간략화를 통하여 부호화 효율을 향상시킬 수 있음을 알 수 있다. 한편, 본 논문에서 제안한 SP8C 윤곽선을 부호화하는 두가지 방법들은 기존의 4방향 혹은 완전 8방향 윤곽선을 부호화하는 방법들에 비해 우수한 성능을 보이는 것으로 나타났다. 즉, SP8C 윤곽선의 특성을 적절히 이용하여 부호화함으로써 RCP 당 약 0.97 ~ 1.16 비트의 정보량으로 윤곽선을 부호화할 수 있어 기존의 방법에 비해 RCP 당 약 0.2 비트 정도 비트량 감축을 얻을 수 있었다. SP8C 윤곽선을 부호화하는데 있어서는 entropy 부호화보다 NDSC 기법이 더 뛰어남을 알 수 있는데, 이는 NDSC에서 flag를 추가적으로 전송할 필요가 없을 뿐만아니라, 세그먼트 단위로 부호화하는 것이 SP8C 윤곽선에 대해서도 효과적이기 때문이다. 한편, 본 논문에서 제시한 방법은 majority 필터링이라는 윤곽선 간략화 후 윤곽선을 부호화하기 때문에 제안한 기법에서 부호화되는 윤곽선은 원래의 것과 차이가 있을 수 있다. 이 차이를 알아보기 위하여 원래의 윤곽선 영상에서의 객체 크기에 대한 mismatched pixel의 비율[6]을 측정한 결과, 그 비율이 약 0.000343 ~ 0.003650으로 나타났다. 따라서, 본 논문에서 제안한 SP8C 윤곽선 표현은 원래의 윤곽선을 비교적 충실히 표현함과 동시에 윤곽선을 효율적으로 부호화함을 알 수 있었다.

VI. 결 론

본 논문에서는 기존의 윤곽선 추출 기법보다 효율적으로 윤곽선을 추출하고, 정보를 압축할 수 있는 기법을 제안하였다. 기존의 윤곽선 추출, 부호화 기법의 성능과 제안하는 기법을 이용하여 윤곽선을 추출하고, 윤곽선이 가지는 방향 제한의 성질을 이용하여 부호화를 하였을 때의 성능을 entropy 측면에서 비교하였다. 제안하는 기법을 이용하였을 때, 대각선 방향 이동시 약 50% 정도 적은 비트량으로 부호화됨을 알 수 있었다. 또한, 실제 영상에 대한 모의 실험을 통하여 제안하는 SP8C 윤곽선 표현 기법이 매우 우수한 성능을 보임을 확인하였다. 본 논문에서 제안한 기법을 이용하여, 현재 연구가 활발히 진행되고 있는 객체지향

영상 부호화기의 성능을 향상시킬 수 있을 것으로 기대된다.

참 고 문 헌

[1] H. G. Musmann, M. Hötter and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, Vol. 1, No. 2, pp. 117-138, October 1989.

[2] C. Gu and M. Kunt, "Contour simplification and motion compensated coding," *Special Issue of Signal Processing: Image Communication on Coding Techniques for Very Low Bitrate Video*, Vol. 7, No. 4-6, pp. 279-296, Nov. 1995.

[3] V.A. Christopoulos, C.A. Christopoulos, J. Cornelis, A.N. Skodras, "A new contour simplication filter for region-based coding," *EUSIPCO-96*, September, 1996.

[4] 홍원학, 김남철, "윤곽상태를 이용한 6방향 윤곽 부호화," 제9회 영상처리 및 이해에 관한 워크샵, 부산, pp.173-178, 1996년 1월

[5] S. H. Cho, R. C. Kim, and S. U. Lee, "Contour coding technique using motion information." in *Proc. of SPIE-VCIP '97*, San Jose, CA, vol. 3024, pp. 435-446, Feb. 1997.

[6] J. L. Kim, J. H. Kim, H. M. Jung and B. U. Lee, "Contour Coding Using DST," in *ISO/IEC JTC1/SC29/WG11 MPEG-4 Related Document*, March 1995.

[7] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, Vol. EC-10, pp. 260-268, June 1961.

[8] T. Kaneko and M. Okudaira, "Encoding of arbitrary curves based on chain code representation," *IEEE Trans. on Communications*, Vol. COM-33, pp. 697-707, July 1985.

[9] B. B. Chaudhuri and S. Chandrashekhar, "Neighboring direction runlength coding: An efficient contour coding scheme," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 20, pp. 916-921, July 1990.

[10] C. C. Lu and J. G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representations," *IEEE Trans. on Communications*, Vol. 39, No. 10, pp. 1511-1514, Oct. 1991.

[11] M. Eden and M. Kocher, "On the performance of contour coding algorithm in the context of image coding. part I: Contour segment coding," *Signal Processing*, Vol. 8, pp. 381-386, 1985.

[12] P. Kauff, U. Gölz, S. Kruse and S. Rautenberg, "Improved image segmentation technique for hybrid waveform/object-oriented coding,"in *Proc. of SPIE-VCIP '94*, Chicago, IL, pp. 1987-1998, September 1994.

저 자 소 개

趙 聖 鎬(正會員)

1995년 서울대학교 전기공학부 공학사. 1997년 서울대학교 전기공학부 공학석사. 1997년 ~ 현재 신세기 이동통신 근무 주관심분야는 영상부호화, 디지털 통신 등

李 商 郁(正會員) 第 33卷 B編 第 6號 參照

현재 서울대학교 전기공학부 교수

金 鱗 澈(正會員)

1985년 2월 서울대학교 제어계측공학과 졸업(공학사). 1987년 2월 서울대학교 제어계측공학과 졸업(공학석사). 1992년 2월 서울대학교 제어계측공학과 졸업(공박사). 1992년 10월 ~ 1993년 10월 서울대학교 자동화시스템 공동 연구소(특별연구원). 1992년 8월 ~ 1994년 2월 대우전자 영상연구소(선임연구원). 1994년 3월 ~ 현재 한성대학교 정보전산학부 (조교수). 주관심분야는 신호처리, 영상처리, VLSI 신호처리 등