

논문96-1-1-06

## 윤곽 상태를 이용한 여섯 방향 윤곽부호화

홍 원 학\*, 허 진 우\*\*, 김 남 철\*\*\*

### Six-Connected Contour Coding Using Contour States

Won Hak Hong,\* Jin Woo Heo,\*\* and Nam Chul Kim\*\*\*

#### 요 약

본 논문에서는 복호화시 정확한 복원이 가능하며 윤곽 상태에 따라 효율적으로 부호화하는 여섯 방향 윤곽부호화 기법을 제안하였다. 먼저 방향벡터의 확률적 분포가 수평 수직 진행 방향과 대각선 진행 방향에서 서로 다르다는 사실에 근거하여 현재의 진행 방향을 두 상태로 나누고, 각 상태별로 다른 코드를 이용하여 부호화하였다. 그리고 부호화 효율을 더욱 높이기 위해 윤곽 상태를 현재 방향벡터와 이전 진행 방향벡터에 따라 다섯 가지로 구분하여 부호화하였다. 또한, 단말 코드워드를 제거함으로써 전체 코드워드 발생 횟수를 줄였다. 실험을 통하여 단말 코드를 제거하고 두 상태 및 다섯 상태로 나누었을 때의 전체 엔트로피는 실영상에 대해서는 각각 12%, 14% 정도, 합성영상에 대해서는 각각 10%, 26% 정도로 크게 감소함을 확인할 수 있었다.

#### Abstract

In this paper, we present efficient six-connected contour coding algorithms which can uniquely reconstruct any contour image and efficiently compress the contour data. We first design chain difference codes using two onward direction states, based on the fact that the probability distribution of the direction vectors of horizontal/vertical direction state is different from that of the direction vectors of diagonal direction state. In order to increase coding efficiency, we also design chain difference codes using five states which are classified according to current and previous onward direction vectors. In addition, we also remove the END codeword to reduce total codeword occurrence. Experimental results show that when using 2 states and 5 states without END codeword total entropy decreases by about 12% and 14% for real images and by about 10% and 26% for a synthetic image, respectively.

#### I. 서 론

분할기반 부호화 기법에서는 영상을 연결성과 동질성을 만족하는 영역들로 분할한 뒤, 각 영역의 경계를 표현하는 윤곽(contour)과 영역 내부의 질감을 분리하여 부호화한다<sup>1),2)</sup>. 일반적으로

윤곽정보는 인간 시각 인식에 있어 질감보다 중요할 뿐만 아니라 부호화에 소요되는 비트량도 질감에 비해 매우 크기 때문에, 윤곽 정보를 손실없이 압축하는 기법이 요구된다. 기존에 연구된 윤곽 정보의 무손실 부호화 기법들은 대부분 체인 차 코드(chain difference code)에 기반하고 있다<sup>3),4),5)</sup>. 이들은 윤곽화소의 연결 상태에 따라 체인 코드의 네 방향 또는 여덟 방향 벡터의 진행 방향을 부호화한다.

일반적으로 영상은 화소(pixel)를 나타내는 직사각형 형태의 셀(cell)들이 바둑판 구조로 배열해 있는 것으로 생각할 수 있다. 따라서, 분할 영역의 경계를 나타내는 윤곽은 엄밀히 말해서 셀 사

\*울산전문대학 전자과

Dept. of Electronic Eng., Ulsan Junior College

\*\*경북대학교 전자공학과

Dept. of Electronic Eng., Kyungpook National University

이의 가상적인 위치에 존재해야 한다. 그러나 기존의 윤곽부호화 방법들에서는 각 영역들의 가장자리(boundary) 화소를 윤곽의 위치로 사용한다. 이러한 이유로 각 영역의 가장자리 화소들은 복호화시 정확하게 복원되지 않는 경우가 발생한다.

한편, 컴퓨터 비전 분야에서는 분할된 영역들의 면적이나 형태를 정확하게 추출하기 위해 mid-crack 코드를 사용한 바<sup>[6]</sup> 있는데, 이는 화소와 화소사이를 윤곽화소로 하여 여섯 방향 연결성을 갖고 진행되는 체인 코드의 일종이다. 최근 Sanson 등<sup>[7]</sup>은 여섯 방향 윤곽부호화의 필요성은 언급하였지만, 이에 관한 실제 부호화 기법은 제시하지 못하였다. 여섯 방향 윤곽화소를 체인 차 코드로 부호화 할 경우 네 방향 및 여덟 방향과는 달리 수평 수직 진행 방향에서의 방향벡터들과 대각선 진행 방향에서의 방향벡터들이 진행되는 각도가 서로 다르다. 따라서 방향벡터들을 진행 방향에 따라 구분하면, 진행 방향에 따라 확률 분포가 다르게 나타나는 점을 부호화에 효과적으로 이용할 수 있다. 또한 이전 진행 방향벡터도 상태 구분의 기준으로 함께 사용한다면 윤곽화소간에 존재하는 상관성까지 이용할 수 있다.

본 논문에서는 복호화시 정확한 복원이 가능하며 윤곽 상태에 따라 효율적으로 부호화하는 여섯 방향 윤곽부호화 기법을 제안한다. 제안된 방법은 먼저 전체 방향벡터들을 수평 수직 또는 대각선 진행 방향에서의 방향벡터들에 따라 두 가지 상태로 구분한다. 다음 각 상태에서의 방향벡터의 조건부 확률을 구하고, 이를 사용하여 각 상태에 독립적으로 코드를 부여한다. 또한 더욱 부호화 효율을 높이기 위해 이전 방향벡터가 어떻게 진행해 왔는가에 따라 윤곽 상태를 다섯 가지로 나누어 부호화한다. 그리고 윤곽화소에서 현재 탐색방향에 대해 부호화할 윤곽화소가 없음을 나타내는 단말 코드워드(END codeword)를 효과적으로 제거함으로써 코드워드의 종류 및 전체 코드워드수를 줄인다. 제안된 여섯 방향 윤곽부호화 방법의 성능을 평가하기 위해 실제 영상과 합성영상에 대해 Pavlidis의 분할법<sup>[8]</sup>에 의해 분할된 윤곽 영상을 사용하였다.

## II. 체인 차 코드에 기반한 윤곽부호화

영역의 가장자리 화소를 윤곽화소로 간주하여 진행되는 윤곽부호화는 윤곽화소의 연결 방향에 따라 네 방향 또는 여덟 방향 윤곽부호화로 나눌 수 있다. 체인 차 코드를 이용한 네 방향 또는 여덟 방향 윤곽부호화에서는 초기 절대 좌표를 부호화하고 현재 윤곽화소에서 다음 윤곽화소에 대한 상대적 위치를 그림 1의 체인 차 코드를 이용하여 부호화한다. 세 영역 이상이 만나는 경계 부분에서의 윤곽화소는 두 개 이상의 탐색 방향으로 진행 가능한데, 이러한 윤곽화소를 교점(vertex)이라 한다. 일반적으로 세 영역 이상이 만나는 경계보다 두 영역이 만나는 경계가 대부분이므로 교점수는 윤곽화소의 수보다 매우 적다. 따라서 윤곽화소의 방향벡터에 작은 비트의 코드워드를 부여하고 교점에 큰 비트의 코드워드를 부여하여야 한다.

네 방향 윤곽부호화에서는 그림 1(a)에서와 같이 0, 1, 2의 체인 차 코드를 이용하여 방향벡터를 나타내며, 진행 방향에 대해 두 개 이상의 탐색 방향으로 진행 가능한 교점은 세 가지 형태로 나타나고 세 개 이상의 탐색 방향으로 진행 가능한 교점은 한 가지로 나타난다. 여덟 방향 윤곽부호화에서는 그림 1(b)에서와 같이 0, 1, 2, 3, 그리고 4의 체인 차 코드를 이용하여 방향벡터를 나타낸다. 여덟 방향 윤곽부호화에서는 네 방향 부호화보다 윤곽화소 수는 적지만 방향벡터의 가지수는 많다.

네 방향 또는 여덟 방향 윤곽부호화에서 윤곽화소의 체계적 탐색 및 부호화를 위해 그래프 알고리즘 중 하나인 DFS(depth first search) 알고리즘을 이용하면 효율적으로 부호화할 수 있다.<sup>[5,6]</sup> 이 방법은 초기 절대 좌표의 전체에 대한 비율을 줄이면서 윤곽화소의 중복 부호화를 피하는 장점을 가지고 있다. DFS를 윤곽부호화에 이용하기 위해 윤곽화소를 그래프의 각 노드(node)로 한다. 진행 방향이 하나인 각 노드에서는 진행 방향에 따른 링크(link)의 방향벡터를 체인 차 코드로 부호화하고, 진행 방향이 둘 이상인 노드에서는 교점의 형태를 나타내는 코드워드를 부호화하

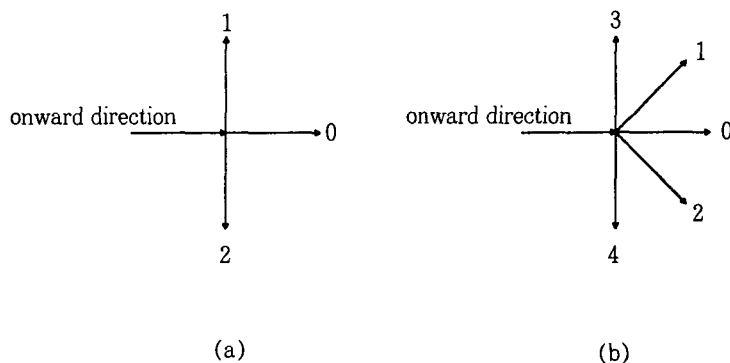


그림 1. 방향 벡터 (a) 네 방향 체인 차 코드 (b) 여덟 방향 체인 차 코드

Fig. 1. Direction vectors: (a) four-connected chain difference code; (b) eight-connected chain difference code.

1	1	1	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2	2	2
1	1	1	①	③	④	4	4	4	2
1	1	3	3	3	3	4	4	4	2
1	3	3	3	3	3	4	4	4	2
3	3	3	3	3	3	3	3	3	2

그림 2. 여덟 방향 윤곽부호화시 유일한 영역맵으로 복원되지 못하는 영역맵의 예  
 Fig. 2. Example of a region map which is not uniquely reconstructed by eight-connected contour coding.

고 교점의 정보 즉, 교점의 위치, 진행 방향, 형태 및 교점에서 진행 방향에 대한 탐색 여부를 스택에 저장해 둔다. 다음 이 교점에서 여러 진행 방향 중 미리 정해진 순서에 따라 한 진행 방향으로 부호화를 계속한다. 이 진행 방향으로 영상의 가장자리를 만나거나 이미 부호화된 노드를 만나면 스택에 저장해 둔 교점의 정보를 가져와 그 교점에서 남아있는 다른 진행 방향으로 부호화를 계속 수행한다. 분할된 영역의 윤곽은 폐경로(closed path)를 이루므로 DFS 탐색 알고리즘을 적용시, 교점정보 중 교점에서 진행 방향의 탐색 여부를 조사하여 교점정보가 변화하였다면 교점의 형태를 바꾸어주거나 교점을 없애으로써 효율적으로 부호화할 수 있다.

그러나 불행하게도 네 방향 또는 여덟 방향 윤곽부호화는 가장자리 화소를 윤곽화소로 사용하기 때문에 복원시 유일한 영역맵을 얻을 수 없는 문제점이 있다. 그림 2에서 보는바와 같이 원으로 표시된 ①, ③, 그리고 ④의 윤곽화소는 순서대로 영역 1, 3, 4

의 화소이지만 모두 영역 1의 화소이거나 영역 4의 화소인 경우에도 그림 2와 같은 윤곽화소로 표현될 수 있기 때문에 이들 화소들은 여러 가지 형태로 복원될 수 있다.

이는 화소와 화소사이의 가상점을 윤곽화소로 하는 여섯 방향 윤곽부호화에서는 발생하지 않는다. 그림 3은 윤곽화소의 수직 및 수평 진행 방향의 연결 상태를 가질 때 여섯 방향 벡터들을 보여준다.<sup>[6,7]</sup>

그림 4는 그림 3의 여섯 방향 벡터에 체인 차 코드를 적용한 예이다. 그림 4(a)는 수평 수직 진행 방향에서의 방향벡터 및 코드워드를 나타낸다. 대각선 진행 방향에서는 방향벡터들이 그림 4(b)와 (c)의 두 가지 형태로 나누어지는데, 그림 3에서처럼 수직 진행 방향의 연결 상태를 가지는 경우 방향벡터 및 코드워드는 그림 4(b)와 같으며, 수평 진행 방향의 연결 상태를 가지는 경우 방향벡터 및 코드워드는 그림 4(c)와 같다.

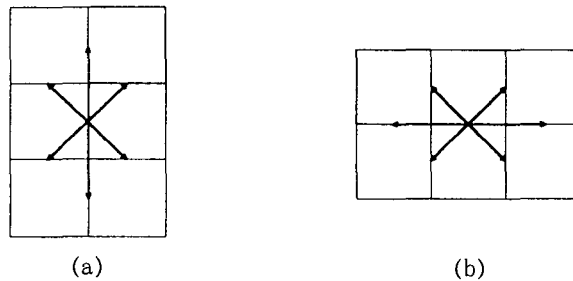


그림 3. 여섯 방향 벡터 (a) 수직 진행 방향 연결 벡터 (b) 수평 진행 방향 연결 벡터  
 Fig. 3. Six-connected vectors: (a) vertical connected vectors; (b) horizontal connected vectors.

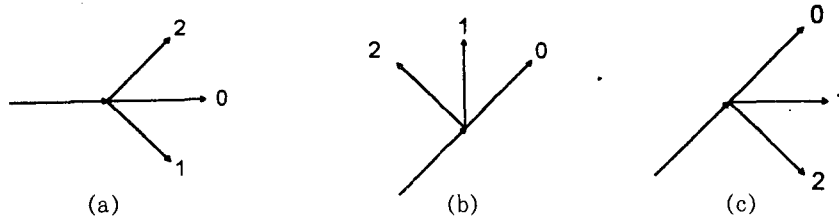


그림 4. 방향벡터 (a) 수평/수직 방향 상태에서의 방향벡터(0°, ±45°) (b),(c) 대각선 방향 상태에서의 방향벡터(0°, ±45° ±45°)  
 Fig. 4. Direction vectors: (a) direction vectors(0°, ±45°) in horizontal/vertical direction state; (b),(c) direction vectors(0°, ±45° ±45°) in diagonal direction state.

III. 여섯 방향 윤곽부호화

본 장에서는 먼저 수평 수직 진행 방향 및 대각선 진행 방향의 두 상태를 이용한 윤곽부호화를 설명하고, 이전 방향벡터의 진행 상태에 따른 다섯 상태를 이용한 윤곽부호화 기법을 설명한다. 다음, 단말 코드워드의 제거 방법과 제안된 여섯 방향 윤곽부호화의 진행 과정의 예를 보여준다.

1. 두 상태를 이용한 윤곽부호화

그림 5는 수평 수직 방향 그리고 대각선 방향의 상태가 그림 4의 방향벡터에 의해 천이하는 것을 보여준다. 수평 수직 방향 상태에서 방향벡터가 0이면 그 상태를 유지하고 방향벡터가 1 또는 2이면 대각선 방향 상태로 천이하게 된다. 대각선 방향 상태에서의 방향벡터가 0 또는 2일 경우 그 상태를 유지하며, 방향벡터가 1이 되면 수평 수직 방향으로 천이하게 된다. 이렇게 천이하면서 각 상태별로 독립적인 코드로서 부호화한다.

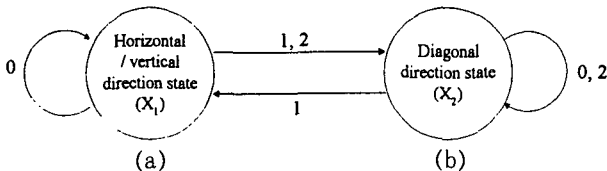


그림 5. 두 가지 진행 방향 상태에 대한 상태 천이도  
Fig. 5. State transition diagram for two onward direction states.

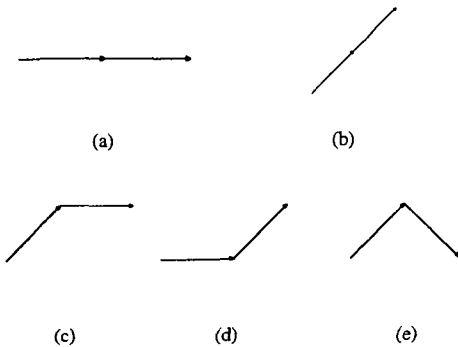


그림 6. 다섯 가지 방향 벡터 (a) 0° 수평 수직 벡터  
(b) 0° 대각선 벡터 (c) 45° 수평 수직 벡터  
(d) 45° 대각선 벡터 (e) 90° 대각선 벡터

Fig. 6. Five direction vectors: (a) 0° horizontal/vertical vector; (b) 0° diagonal vector; (c) 45° horizontal/vertical vector; (d) 45° diagonal vector; (e) 90° diagonal vector.

2. 다섯 상태를 이용한 윤곽부호화

그림 5에서와 같이 현재의 진행 방향벡터만으로 상태를 구분하여 부호화하면 0°, 45°, 또는 90° 변화 형태를 가진 윤곽화소에

서의 방향벡터들의 특성이 다름에도 불구하고 같은 특성을 지닌 것으로 간주되어 부호화의 효율이 떨어진다. 따라서 효율적인 부호화를 위하여 이전 방향벡터의 진행 상태에 따라 다섯 가지로 구분하여 부호화한다. 즉, 현재 윤곽화소의 현재 진행 방향벡터와 이전 윤곽화소의 이전 진행 방향벡터에 의해 방향벡터들이 어떻게 진행해 왔는가에 따라 그림 6과 같이 0° 수평 수직 방향 상태( $X_1$ ), 0° 대각선 방향 상태( $X_2$ ), 45° 수평 수직 방향 상태( $X_3$ ), 45° 대각선 방향 상태( $X_4$ ), 그리고 90° 대각선 방향 상태( $X_5$ )의 다섯 가지로 나누어 부호화한다.

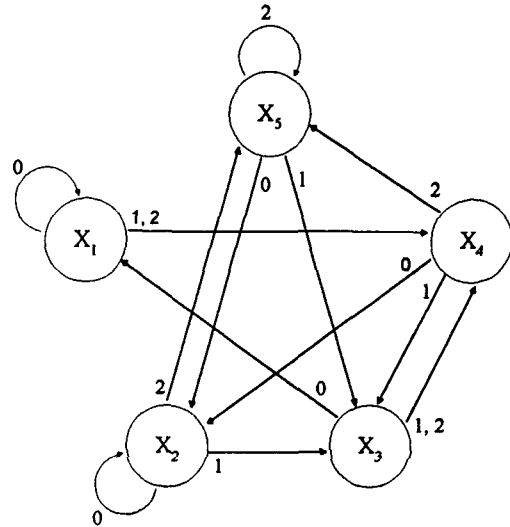


그림 7. 다섯 가지 진행 방향 상태에 대한 상태 천이도  
Fig. 7. State transition diagram for five onward direction states.

그림 7은 다섯 상태의 상태 천이도이다. 각 윤곽 상태에서 그림 4의 방향벡터에 의해 윤곽 상태가 천이된다. 0° 수평 수직 방향의  $X_1$  윤곽 상태에서는 방향벡터 0에 의해 원래 상태가 되고, 방향벡터 1 또는 2에 의해 45° 대각선 방향의  $X_4$  상태로 천이한다.  $X_4$  상태에서 방향벡터 0에 의해 0° 대각선 방향의  $X_2$  상태로 천이하며, 방향벡터 1에 의해 45° 수평 수직 방향의  $X_3$  상태로 천이하며, 방향벡터 2에 의해 90° 진행 대각선 방향의  $X_5$  상태로 천이한다

3. 단말 코드워드의 제거

단말 코드워드는 현재 윤곽화소에서 탐색 방향에 대해 탐색시 다음 윤곽화소가 없음 나타내고, 이전 교점의 정보 또는 새로운 초기 윤곽화소의 정보를 스택에서 가져오는 역할을 한다. 부호화된 현재 윤곽화소가 윤곽 영상의 가장자리 화소일 경우, 초기 윤곽화소로 다시 돌아온 경우, 그리고 탐색시 이미 부호화된 윤곽화소를 만나는 경우에 현재 윤곽화소에서 탐색 방향에 대해 다음 윤곽화소를 찾을 수 없으므로 단말 코드워드를 생성해야 한다.

이러한 단말 코드워드는 복원시 단말 코드워드가 생성되는 각

경우의 상황을 고려함으로써 제거할 수 있다. 먼저 현재 복호된 윤곽화소의 위치를 구함으로써 윤곽화소가 가장자리 화소인 경우를 알 수 있고, 또한 현재 복호된 윤곽화소에서 탐색 방향에 대해 이미 복호된 윤곽화소를 만나게 되면 단말 코드워드가 나오게 됨을 알 수 있다.

4. 여섯 방향 윤곽부호화의 예

여섯 방향 윤곽부호화에서는 네 방향 또는 여덟 방향 윤곽부호화에서의 마찬가지로 초기 절대 좌표, 방향벡터 및 교점에 관한 정보들을 부호화한다. 이때 교점의 가지수는 네 방향 윤곽부호화에서의 마찬가지로 네 가지이다. 표 1은 제안된 여섯 방향 윤곽부호화에서 사용된 코드표이다.

표 1. 여섯 방향 윤곽 코드워드 표.

Table 1. Six-connected codeword table.

types of codewords		codewords
direction vectors		0, 1, 2
END codeword		3
vertex type	vertex with direction vectors 0, 1	4
	vertex with direction vectors 0, 2	5
	vertex with direction vectors 1, 2	6
	vertex with direction vectors 0, 1, 2	7

그림 8은 여섯 방향 윤곽 영상을 부호화하는 과정의 예를 보여 준다. 그림 8에서 보듯이 윤곽화소를 중복없이 탐색 가능함을 알 수 있다. 좌측 상단 좌표를 (0, 0)로 하고, 그림 8의 방향벡터와 표 1의 코드워드 표를 이용하여 윤곽화소를 순서대로 부호화하면 다음과 같다.

[5, 0], 0, 2, 5, 2, 1, 2, 0, 0, 0, (3), 4 → 0, 0, 0, 0, 1, 1, 0, 4, (3), 1, 0, 1, 1, 2, 0, (3)

[5, 0]은 첫 번째 윤곽화소의 초기 절대 좌표이며, 그 다음 0은 두 번째 윤곽화소에 대한 방향벡터를 나타낸다. 그리고 3은 열 번째 윤곽화소에서 탐색의 끝임을 나타내는 단말 코드워드이다. 실제 부호화되지 않으므로 (3)로 표현하였다. 4→0은 교점의 형태 4를 교점이 아닌 방향벡터 0으로 바꾸어 부호화함을 의미한다. 즉, 열 한 번째 윤곽화소가 탐색 방향에 대해 열 두 번째와 스물 여섯 번째 윤곽화소 각각에 대한 방향벡터 0과 1을 가지는 교점이므로 교점의 형태 4를 부여하고 열 두 번째 윤곽화소 위치에서 열 세 번째 윤곽화소 방향으로 부호화를 진행한다. 스물 여섯 번째 윤곽화소가 부호화된 후 탐색방향에 더 이상 부호화할 윤곽화소가 없으므로 스택으로부터 교점정보를 얻게 된다. 교점인 열 한 번째 윤곽화소에서는 교점정보를 이용하여 스물 여섯 번째 윤곽화소 방향으로 부호화를 수행하게 된다. 그러나 이미 부호화되어 있어 중복하여 부호화할 필요가 없으므로 열 한 번째 윤곽화소를 교점으로 부호화하지 않고 열 두 번째 윤곽화소에 대한 방향벡터 0을 대신 부호화한다. 각 방향벡터 및 교점의 형태는 윤곽 상태에 따라 독립적으로 수행한다.

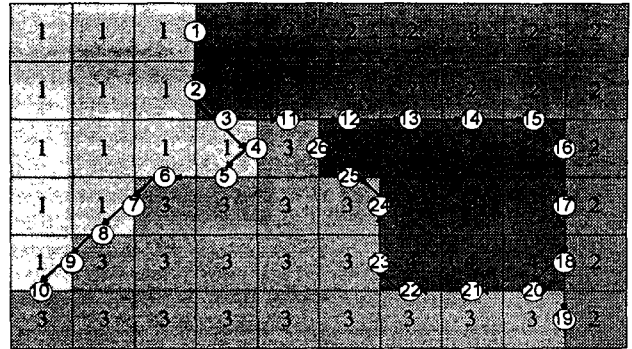


그림 8. 여섯 방향 윤곽부호화의 예

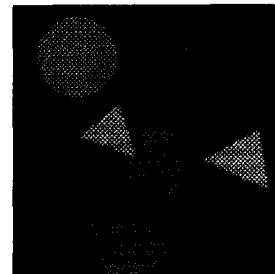
Fig. 8. Example of six-connected contour coding.

IV. 실험 결과 및 고찰

제안된 여섯 방향 윤곽부호화의 성능을 알아보기 위하여 256 명암도의 256×256 크기를 갖는 Lena 영상 및 합성영상에 대해 실험하였다. 그림 9와 그림 10은 각각 실험에 사용된 원 영상들과 윤곽 영상들이다. 그림 10의 (a), (c)는 Pavlidis의 분할법(split-and-merge)을 사용하여 1×1 화소단위까지, (b)는 2×2화소 단위까지 분할한 윤곽 영상이다. 이 그림에서 분할 결과의 영역수는 (a), (b), (c)의 순으로 각각 312, 262, 7이며 윤곽화소수는 각각 13782, 11522, 1509개이다.



(a)



(b)

그림 9. 원 영상들 (a) Lena 영상 (b) 합성영상

Fig. 9. Original images: (a) Lena; (b) synthetic image.

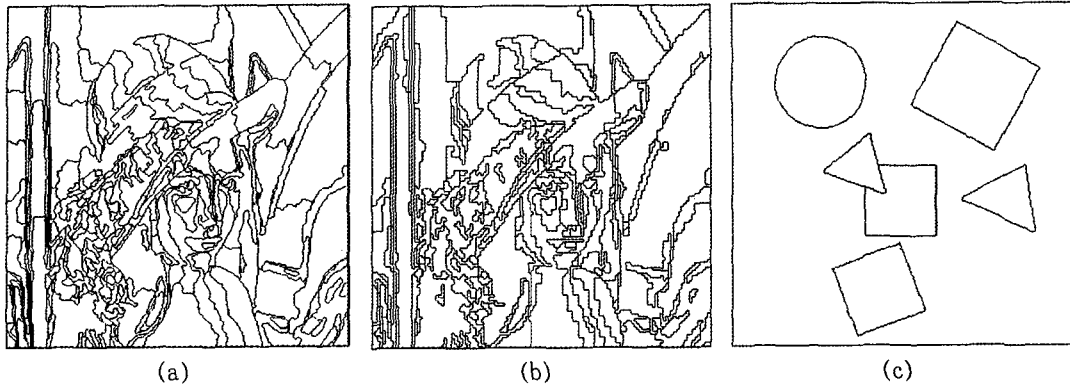


그림 10. 윤곽 영상들 (a) Lena의 분할 윤곽 영상 (b) Lena의 분할 윤곽 영상(2×2 화소단위로 분할) (c) 합성영상의 분할 윤곽 영상  
 Fig. 10. Contour images: (a) contour image for Lena; (b) contour image for Lena(segmented by 2×2 pixel unit); (c) synthetic image.

본 논문에서는 제안된 각 윤곽부호화의 성능을 평가하기 위한 척도로 전체 엔트로피와 윤곽화소당 평균 비트수를 사용하였다. 몇 가지 상태로 나누었을 경우, 코드워드들의 전체 엔트로피  $H(Y)$ 는 각 상태  $X_i$ 에서의 조건부 엔트로피  $H(Y|X_i)$ 들의 평균으로 다음과 같이 표현된다.

$$H(Y) = \sum_{i=1}^N H(Y|X_i)P(X_i) \quad (1)$$

여기서  $N$ 은 상태수를,  $P(X_i)$ 는 각 상태의 확률을 나타낸다. 그리고, 윤곽화소당 평균 비트수는 전체 엔트로피 × (전체 코드워드 빈도수) / (윤곽화소수)로 표현된다.

먼저, 아래와 같은 네 가지의 경우로 나누어 각각에 대해 코드워드 발생 횟수 및 전체 엔트로피를 조사하여 비교하였다.

- (1) 상태를 나누지 않은 경우의 여섯 방향 윤곽부호화
  - 1) 단말 코드워드가 있는 경우
  - 2) 단말 코드워드를 제거한 경우
- (2) 단말 코드워드를 제거하고 두 상태로 나눈 경우의 여섯 방향 윤곽부호화
- (3) 단말 코드워드를 제거하고 다섯 상태로 나눈 경우의 여섯 방향 윤곽부호화

표 2부터 표 4까지는 그림 10(a) Lena 윤곽 영상에 대해 각 방법으로 여섯 방향 윤곽부호화하였을 때의 실험 결과이다. 표 2에서는 방법 (1)을, 표 3에서는 방법 (2)을 그리고 표4에서는 방법 (3)을 사용하였을 때 구해진 코드워드들의 발생 횟수 및 전체 엔트로피를 나타내고 있다. 먼저 표 2에서 보듯이, 단말 코드워드의 발생 횟수가 방향벡터 코드워드들의 발생 횟수보다는 작지

표 2. 그림 10(a)에 대하여 방법 (1)을 사용하였을 때의 코드워드들의 발생 횟수 및 전체 엔트로피  
 Table 2. Codeword occurrency and total entropy when using method (1) for Fig. 10(a).

with/without END	with END	without END
codeword occurrency		
total codewords	13782	13471
codeword 0	8468	8461
codeword 1	3351	3348
codeword 2	1362	1366
codeword 3	307	0
codeword 4	119	122
codeword 5	112	119
codeword 6	49	47
codeword 7	14	8
total entropy	1.53	1.41

만, 교점 형태 코드워드들의 발생 횟수보다는 더 크게 나타남을 볼 수 있다. 따라서 단말 코드워드만을 제거할 경우에도 전체 엔트로피가 8% 정도 감소한다.

표 3과 4는 두 상태와 다섯 상태로 상태를 나누었을 때의 실험 결과이다. 이들 표에서 보듯이 각각의 상태로 나누었을 경우 각 상태에서의 코드워드들의 발생 횟수 분포가 서로 다름을 알 수 있다. 따라서 각 상태마다 다른 코드를 이용하여 부호화함으로써 상태를 구분하지 않고 하나의 코드로 부호화한 표 2의 경우보다 전체 엔트로피가 각각 2%, 4% 정도 감소되었다.

표 3. 그림 10(a)에 대하여 방법 (2)를 사용하였을 때의 코드워드들의 발생 횟수 및 전체 엔트로피

Table 3. Codeword occurrency and total entropy when using method (2) for Fig. 10(a).

two states codeword occurrency	horizontal/vertical direction state	diagonal direction state
total codewords	13471	
codeword 0	4140	4321
codeword 1	1124	2224
codeword 2	986	380
codeword 4	54	68
codeword 5	59	60
codeword 6	33	14
codeword 7	4	4
conditional entropy	1.43	1.33
total entropy	1.38	

표 4. 그림 10(a)에 대하여 방법 (3)을 사용하였을 때의 코드워드들의 발생 횟수 및 전체 엔트로피

Table 4. Codeword occurrency and total entropy when using method (3) for Fig. 10(a).

five states codeword occurrency	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>
total codewords	13471				
codeword 0	2995	2369	1145	1606	346
codeword 1	588	1663	536	399	162
codeword 2	488	267	498	87	26
codeword 4	40	41	14	24	3
codeword 5	34	29	25	27	4
codeword 6	13	9	20	5	0
codeword 7	2	3	2	1	0
conditional entropy	1.26	1.39	1.66	1.13	1.24
total entropy	1.35				

표 5는 그림 10의 (a), (b), (c) 세 윤곽 영상에 대해 각 방법들로 부호화 효율을 나타내는 윤곽화소당 평균 비트수를 구한 것이다. 표에서 보듯이 1×1화소단위와 2×2화소 단위까지 분할한 각각의 윤곽 영상에 대해 방법 (3)의 결과는 방법 (1)에서 단말 코드를 제거한 경우에 비해서 각각 5%, 4% 정도 감소되었다. 또한 합성영상에 대해서도 방법 (3)을 사용한 결과는 방법 (1)에서 단말 코드를 가진 경우에 비해서 25% 정도 감소되었다. 또한 표에서 보듯이 상태를 구분했을 때에 비해 구분하지 않았을 때의 윤곽화소당 평균 비트수의 감소는 실영상에 비해 합성영상에서 크게 나타난다. 이는 실영상보다 합성영상에서 코드워드들의 발생횟수의 분포가 상태에 따라 잘 분류되기 때문이다. 전체적으로는 단말 코드가 있고 상태를 나누지 않은 경우에 비해 단말 코드를 제거하고 두 상태 및 다섯 상태로 나누었을 때의 전체 엔트로피는 실영상에 대해서는 각각 12%, 14% 정도, 합성영상에 대해서는 각각 10%, 26%로 크게 감소한다.

표 5. 윤곽 영상들에 대해 방법 (1), (2), (3)을 사용하였을 때의 윤곽화소당 평균 비트수

Table 5. Average bits per contour pixel when using method (1), (2), and (3) for contour images.

contour images methods	Lena image	Lena image (segmented by 2×2)	synthetic image
method(1) with END	1.53	0.89	1.36
method(1) with END	1.38	0.76	1.33
method(2)	1.35	0.75	1.22
method(3)	1.31	0.73	1.00

### V. 결 론

본 논문에서는 원 영역의 형태를 정확히 복원할 수 있는 여섯 방향 윤곽부호화를 구현하고, 윤곽화소들에 존재하는 상관도를 이용하기 위해 윤곽 상태에 따라 구분하여 부호화하는 방법을 제안하였다. 먼저 방향벡터의 확률적 분포가 수평 수직 진행 방향과 대각선 진행 방향에서 서로 다르다는 사실에 근거하여 현재의 진행 방향을 두 상태로 나누고, 각 상태별로 다른 코드를 이용하여 부호화하였다. 다음 부호화 효율을 높이기 위해 윤곽 상태를 현재 방향벡터와 이전 방향벡터가 어떻게 진행해 왔는가에 따라 다섯 가지로 구분하여 부호화하였다. 마지막으로 단말 코드워드가 생성되는 각 경우의 상황을 복원시 고려할 수 있도록 부호화하여 단말 코드워드를 제거함으로써 코드워드의 종류 및 전체 코드워드 발생 횟수를 줄였다.

실험에서 먼저 단말 코드만을 제거한 경우 상태 구분의 유무에 관계없이 전체 엔트로피가 7~14% 감소함을 확인할 수 있었다. 또한 상태를 구분하지 않았을 때에 비해 두 상태 및 다섯 상태로

나누었을 때 전체 엔트로피는 실영상에 대해서는 각각 2%, 4% 정도, 합성영상에 대해서는 각각 8%, 25%로 감소하였다. 전체적으로는 단말 코드가 있고 상태를 나누지 않은 경우에 비해 단말 코드를 제거하고 두 상태 및 다섯 상태로 나누었을 때의 전체 엔트로피는 실영상에 대해서는 각각 12%, 14% 정도, 합성영상에 대해서는 각각 10%, 26%로 크게 감소함을 확인할 수 있었다.

이로써, 제안된 윤곽부호화 기법은 실영상 뿐만 아니라 합성영상이나 컴퓨터 그래픽 영상과 같은 합성영상의 윤곽부호화에 적합하리라 기대된다. 특히 합성영상의 경우 방향벡터들의 확률적 분포가 각 윤곽 상태에 따라 잘 구분되기 때문에 보다 효율적으로 부호화가 가능함을 알 수 있었다.

#### 참 고 문 헌

- [1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding technique," *Proc. IEEE*, vol. 73, pp. 549-574, Apr. 1985.
- [2] M. Eden, M. Kocher, and M. Eurasip, "On the performance of a contour coding algorithm in the context of image coding, Part I: contour segment coding," *IEEE Trans. Signal Processing*, vol. 8, pp. 381-386, May 1985.
- J. elected Areas* mm. pp.86-96, 1992
- [3] H. Freeman, "Computer processing of line drawing images," *Computer Surveys*, vol. 6, no. 1, May, 1974.
- [4] S. M. Ali and R. E. Burge, "A new algorithm for extracting the interior of bounded regions based on chain coding," *CVGIP*, vol. 43, pp. 256-264, 1988.
- [5] 김종훈, 김한수, 김성대, 김재균, "Depth first search 이용한 윤곽선 영상의 효과적인 부호화 기법," 대한전자공학회 논문
- [6] F. Y. Shih, and W. T. Wong, "Reconstruction of binary and gray-scale images from mid-crack code descriptions," *Journal of Visual Communication and Image Representation*, vol. 4, no. 2, pp. 121-129, June 1993.
- [7] H. Sanson, "Region-based motion analysis for video coding at low bitrates," *MPEG'94 Contribution*, March 1994.
- [8] S. L. Horowitz, and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *JACM*, vol. 23, Apr. 1976.
- [9] E. Horowitz, S. Sahni, and S. Anderson-Freed, *Fundamentals of Data Structures in C*, Computer Science Press, 1993.



저 자 소 개

洪元學

1991년 2월 경북대학교 대학원 전자공학과 졸업(공학석사).  
1996년 2월 경북대학교 대학원 전자공학과 졸업(공학박사).  
1995년 3월 ~ 현재 울산전문대학 전자과 전임강사.  
주관심 분야는 영상처리, 영상압축, 컴퓨터비전.



許眞佑

1995년 2월 경북대학교 전자공학과 졸업(공학사).  
현재 경북대학교 대학원 석사과정.



金南哲

1955년 7월 29일생.  
1978년 2월 서울대학교 전자공학과 졸업(공학사).  
1980년 2월 한국과학원 전기 및 전자공학과 졸업(공학석사).  
1984년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학박사).  
1991년 8월 ~ 1992년 8월 미국 Syracuse 전기 및 전산학과 객원교수.  
1990년 1월 ~ 1994년 12월 생산기술연구원 HDTV 신호처리 분과위원.  
1994년 1월 ~ 현재 대한전자공학회 및 한국통신학회 편집위원.  
1996년 ~ 현재 한국방송공학회 편집위원  
1984년 3월 ~ 현재 경북대학교 전자 전기공학부 교수.  
주관심 분야는 영상처리, 영상압축, 컴퓨터비전.