

Depth First Search 알고리즘을 이용한 윤곽선 영상의 효과적인 부호화 기법

(An Efficient Contour Coding Method Using Depth First Search Algorithm)

金 鍾 勳,** 金 漢 樹,* 金 聖 大,* 金 在 均*

(Jong Hun Kim, Han Soo Kim, Seong Dae Kim and Jae Kyoon Kim)

要 約

본 논문에서는 최근들어 활발히 연구되고 있는 영역화를 이용한 영상 부호화 방식의 데이터 감축효과를 높이기 위하여 새로운 윤곽선 영상 부호화 기법을 제안한다.

일반적으로 윤곽선 부호화는 chain coding 혹은 chain difference coding 이 많이 이용되는데, 이 방법들을 따른다면 윤곽선이 중복되어 부호화되며 chain 의 시작을 나타내는 초기 절대 좌표를 표시하는데 필요한 데이터량이 매우 많다.

제안 방식은 이러한 문제점들을 해결하기 위하여 depth first search 를 윤곽선 부호화에 이용하였다. 제안방식을 split-merge 기법을 이용하여 얻어진 윤곽선 영상에 적용해 본 결과 중복을 피한 chain difference coding 에 비해 60% 정도의 데이터 감축효과가 있었다.

Abstract

In this paper, a new contour coding algorithm is investigated for use in region based image coding. Generally the contour data may be encoded by its chain codes or chain difference codes. But the data compression efficiency is low because of heavy burden for initial absolute coordinates of each chain. To alleviate this problem, the depth first search in graph traversal algorithm is applied to the chain difference coding method. The proposed coding scheme is shown to be very efficient for contour images obtained by split-merge segmentation. Finally, we can reduce data about 60% in comparison with modified chain difference coding.

I. 서 론

*正會員, **準會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)
接受日字: 1988年 7月 28日

새로운 영상 부호화 방법의 하나로 영상 영역화를 이용한 부호화 방법이 있다.^{1),2)} 영역화를 이용한 부호화는 그림 1 처럼 원 영상을 밝기 값이나 영역내의

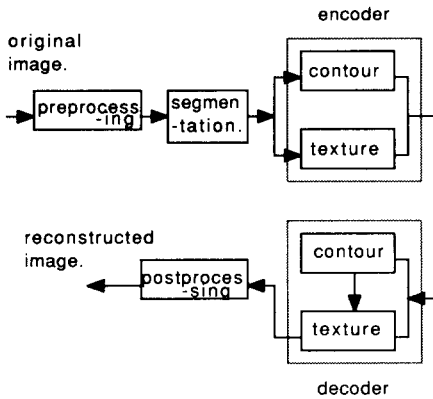


그림 1. 영역화를 이용한 부호화기의 구성도
Fig. 1. Region based image coding.

특성이 비슷하도록 분할한후, 그 영역들의 경계를 표시하는 윤곽선과 영역들의 특징을 표현하는 대표값을 전송하여 수신단에서 원영상에 근사한 영상을 재구성하는 방법이다.

본 논문에서는 이 부호화 방법의 데이터 감축효과를 높이기 위하여 그림 1의 부호화기단에서 윤곽선 영상에 할당되는 데이터량을 줄이는 한 방안을 제안한다.

일반적으로 윤곽선 영상은 chain code 또는 chain difference code로 부호화한다.^{1,3,4} Chain code란 윤곽선의 임의의 한 점에서 시작하여 화소들간의 연결 상태에 따라 경계선을 일정한 방향으로 그림 2(a)와 같은 방향 벡터들의 순열로서 표시하는 방법이다. 그러나 이러한 code로 부호화 한다면 chain의 link, 즉 윤곽선에 데이터가 많이 중복되어 부호화된다. 또한 chain의 시작을 나타내는 초기 절대 좌표의 표현에 많은 bit 수가 소요된다. 이와 같은 chain code의 데이터 중복성을 줄이기 위하여 최근에 많은 연구가 시도되었다.^{3,4,5}

중복성을 줄이기 위한 방법의 하나인 근사화를 이용하는 방식은 데이터 감축은 크게 이루어 지지만 근사화로 인하여 영상에 일그러짐이 생기고 일그러짐의 정도를 가늠하는 임계치가 요구된다. 윤곽선 영상은 영역의 경계로서 영역화를 이용한 부호화에서는 그림 1에서처럼 영상 재구성의 기본이 되므로 가능한한 정확하게 나타낼 필요가 있다.¹¹

그리고 이 분야에서 많이 이용되는 M. Eden, M. Kocher 방식은 실질적인 부호화 방식이 아니라 부호화에 필요한 데이터량을 단순히 구하는 것이기 때문에 부호화 방식으로 이용하기가 어렵다. 또한 윤

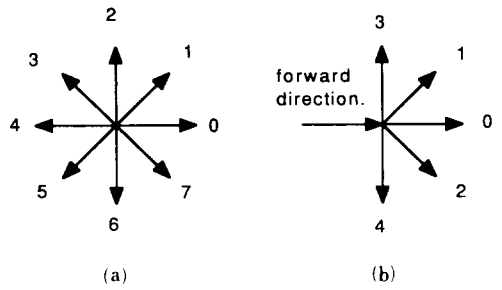


그림 2. 방향 벡터
(a) chain code
(b) chain difference code

Fig. 2. Direction vector.
(a) chain code,
(b) chain difference code.

곽선 영상을 표현하는데 필요한 데이터량을 계산한다고 하더라도, 초기 절대 좌표에 해당하는 데이터량은 고려하지 않고 구한 값이므로 정확한 값과는 많은 차이가 난다.

제안방식은 기존 방식의 문제점을 보완하여 윤곽선의 데이터 중복성을 피하고, 초기 절대 좌표의 전체에 대한 비율을 줄이며, 무손실 부호화하는 것을 기본으로 하고 있다. 이를 위하여 그래프상에서 연결된 구성원들을 체계적으로 찾아가는 그래프 traversal 알고리즘중에서 가장 많이 알려져 있는 depth first search를 부호화에 이용하였으며, codeword의 분포를 데이터 감축에 이용하기 위하여 chain difference code로 부호화 하였다.^{4,6}

다음 장에서는 먼저 depth first search를 이용한 부호화 방식을 살펴보고, 부호화를 위하여 필요한 교점(vertex 혹은 node)의 표현 및 특수 codeword의 사용, 그리고 제안 방식의 전체적인 구조에 대하여 알아본다.

II. Depth first Search (DFS)를 이용한 부호화

DFS는 그래프상에서 연결된 구성원들을 체계적으로 찾아가는 방법으로 그림 3(a)의 그래프상의 모든 교점(vertex 혹은 node)들을 DFS로 찾아가면 이의 탐색 순서는 V₁, V₂, V₄, V₅, V₃, V₆, V₇로서, 그림 3(b)와 같이 그래프에 대한 연결된 탐색 경로가 형성된다.⁶

이와 같이 모든 교점들을 체계적으로 찾아가는 DFS를 이용하여 부호화하려면, 그림 3(b)의 탐색 경로를 윤곽선으로 가정하고, 교점은 탐색할 방향이 2

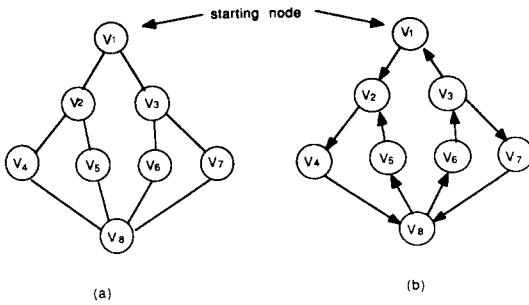


그림 3. Depth first search의 수행순서
 (a) 그래프 (b) 탐색경로
 Fig. 3. Illustration of depth first search.
 (a) graph, (b) search path.

개이상인 윤곽선 화소로 가정해야 한다. 또한 DFS의 수행과정과 달리 모든 교점뿐만 아니라 교점과 교점을 연결하는 직선(edge)도 모두 찾아가야 하기 때문에 그림 3(b)의 이러한 직선성분도 부호화 해야한다.

즉, 그림 4(a)를 그림 2(b)의 방향 벡터로 그림 4(b)의 순서를 따라 부호화하면,

(3, 1), 0, 0, 0, 2, 0, X₁, (5, 1), 0, 0, 0, X₂, 0, 0, X₂, 0, 0, X₃, 0, 2, 0, X₄, 0, 0, 3, 0, 1, X₅, 4, 1, X₆, X₇

이다.

구해진 code에는 초기 절대 좌표, link를 나타내는 codeword 이외에 영상의 재구성을 위해 필요한 codeword로, 앞에서 언급한 것처럼 교점에 대한 정보를 나타내는 특수 codeword(X₂, X₃, X₅)와 탐색의 끝을

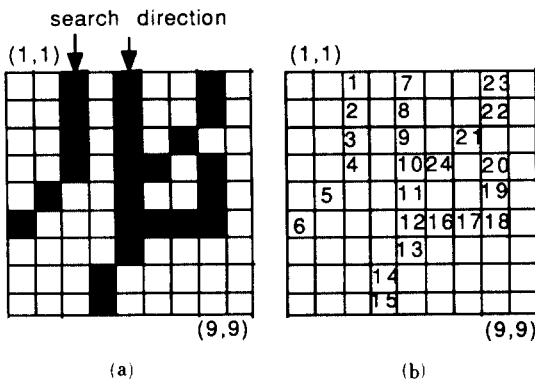


그림 4. 부호화의 예제
 (a) 윤곽선 영상 (b) 부호화 과정
 Fig. 4. Contour coding using DFS.
 (a) image, (b) the order of contour pel search.

나타내는 terminal codeword (X₁, X₂, X₆, X₇)가 있다. 교점에 대한 정보를 나타내는 이유는, 영상을 재구성할 때 code상에서 스택에 push 해야하는 교점의 위치를 결정해야 하고(prefix), pop할 때는 교점의 위치에서 탐색할 방향을 결정해야 하기 때문이다(type). 그러므로 데이터 감축을 위해서는 교점의 정보를 가능한 적은 bit수로 만들 필요가 있다. Terminal codeword는 영상의 재구성을 위하여 code상에서 어떤 부분이 탐색의 끝이라는 것을 표시하는 부분이다. 또한 “X₂”처럼 스택으로부터 데이터를 pop 하였으나 부호화에 불필요하게 사용된 교점도 존재하므로 제거하도록 해야한다. 이에 관한 해결 방안을 3.4절에서 제안한다.

III. 교점(Vertex 혹은 Node) 정보의 부호화

본 절에서는 DFS의 수행을 위해 필요한 교점에 관한 정보를 체계적으로 표현하는 방법을 제안한다.

DFS를 부호화에 이용하려면 반드시 교점의 정보를 부호화 해야한다. 교점으로 정의되는 화소는 한 위치에서 탐색할 수 있는 방향이 2개 이상인 윤곽선 화소로서, 부호화를 위해서는 code상에서 교점의 위치 및 스택으로부터 pop하였을 때 탐색할 방향을 기억 시켜야 한다. 이와같은 정보의 표현은 다른 방법으로도 할 수 있으나 제안 방식에서는 교점의 형태(type)를 부호화하여 이에 대한 정보를 표현 하였다.

그런데 교점으로 취급할 수 있는 형태는 임의의 윤곽선 화소로부터 탐색할 수 있는 방향은 진행 방향에 대하여 5방향(그림2)이므로 이의 모든 경우의 수는

$$\begin{aligned}
 & {}_5C_2 + {}_5C_3 + {}_5C_4 + {}_5C_5 \\
 & = 10 + 10 + 5 + 1 \\
 & = 26
 \end{aligned}$$

이다.

만일 8방향(chain code의 방향 벡터) 각각에 대하여 생각한다면, 208(26×8)개의 교점 형태가 생긴다. 그러나 부호화 하면서 부호화 진행방향은 알 수 있으므로, 26개의 교점 형태만 생각할 수 있다. 이를 codeword로 표현한다면 5bit가 필요하다. 그런데, 모든 교점에 대하여 이와 같은 bit수를 할당한다면, 전체 데이터 량에서 차지하는 비율이 매우 커진다. 데이터 감축효과를 기대하려면 교점 형태의 종류를 체계적으로 줄임으로서 bit수를 줄일 필요가 있다.

교점 형태의 종류를 줄이는 과정은 윤곽선 화소가 주위에 존재하는지를 탐색할 때, 진행 방향을 수평, 수직방향 그리고 대각선 방향인 2가지로 나누고, 탐

색순서를 다르게 조절함으로써 가능해진다. 즉, 그림 5와 같이 2 가지 진행 방향에 대하여 윤곽선 화소 탐색순서를 다르게 하고, 만일 윤곽선 화소가 존재한다면 탐색 방향에 대한 교점 형태를 그림 6의 형태로 제한시킨다. 이렇게 할 경우 진행 방향은 현재 교점 이전 이후의 위치로부터 알 수 있으므로 그림 6처럼 4 개의 그룹인 총 2bit (2^2)로 표현된다. 교점의 종류를 이렇게 구성할 수 있는 이유는 탐색 순서를 결정할 때 대각선 방향보다 수평-수직 방향에 우선권 (priority)을 주었기 때문이다. 결과적으로 교점을 표현하는 과정에서 3bit 가량 줄어들게 된다.

IV. 전체 흐름도

제안방식은 기존 윤곽선 부호화 방식의 데이터 중복성 및 초기 절대 좌표의 데이터 비율을 줄이기 위하여 depth first search 알고리즘을 기본적으로 따르고 있다. DFS를 부호화에 이용하기 위하여 교점을 특수 codeword로 부호화 한다. 즉, 윤곽선 화소를 부호화 할 때마다 교점의 여부를 검색하고, 교점인 경우에는 앞의 그림 6의 하나로 대응시켜 특수 codeword (prefix, type)를 할당한다. 또한 더 이상 부호화할 윤곽선 화소가 없을 경우에는 탐색의 끝임을 나타내는 terminal codeword를 할당하고 있다.

그런데 제안방식에서는 앞서서와 같은 특수 codeword의 체계적인 사용뿐만 아니라, 데이터 감축효과를 높이기 위하여 부호화에 불필요한 데이터를 제거하고 있다. 이 과정은 교점 정보를 표현하는 부분에서 이루어진다. 이는 스택에서 데이터를 pop하여 그 위치에서의 교점 형태가 push할 때의 교점 형태와 다를 때이다. 많은 경우에 pop한 후의 교점 형태를 살펴보면, 주위의 탐색할 화소들이 모두 부호화

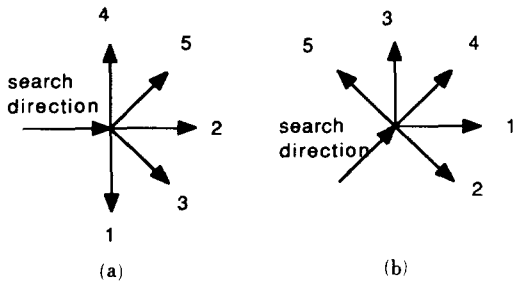


그림 5. 진행 방향에 따른 이웃화소의 탐색순서
 (a) 수평-수직 방향
 (b) 대각선 방향
 Fig. 5. The order of neighboring point search.
 (a) Horizontal & vertical direction.
 (b) diagonal direction.

Priority. →	Group 1	Group 2	Group 3	Group 4
	1	2	3	4
1				
2		.	.	.
3			.	.
4,5

this number means the order of Fig-5.

(a) continue

Priority. →	Group 1	Group 2	Group 3	Group 4
	1	2	3	4
1			.	.
2				
4		.	.	.
3,5

this number means the order of Fig-5.

→ : search direction on the current pel.
 → : search direction at pop being occurred.

(b)

그림 6. 진행방향에 따른 교점의 종류
 (a) 수평-수직 방향
 (b) 대각선 방향

Fig. 6. Vertex types when search direction is
 (a) horizontal or vertical,
 (b) diagonal.

되어 있기 때문에 교점으로 생각할 필요가 없다. 이런 교점의 prefix, type codeword는 제거할 수 있다. 그림 7은 이상의 내용을 종합한 제안방식의 전체 흐름도이다.

제안방식의 전체적인 구성을 나타내는 앞의 흐름도는 8-connected 윤곽선 영상의 부호화 알고리즘

V. Simulation결과 및 고찰

제안 방식의 특성을 알아보기 위하여 split-merge 방법으로 contour 영상을 얻었다. 영역화 이전의 원 영상은 256×256크기, 256밝기 값(grey level)을 갖는 Girl 및 Cronkite 영상이다.

윤곽선 영상의 복잡성은 영역화과정으로 이루어진 윤곽선 영상내의 영역수에 의존한다. 그러므로, 제안 방식의 특성을 알아보기 위하여 원 영상을 여러가지 영역수를 갖도록 영역화를 수행하였다. 표1은 2가지 윤곽선 영상에 대한 특성이며 그림8은 영역화과정으로 이루어진 윤곽선 영상이다.

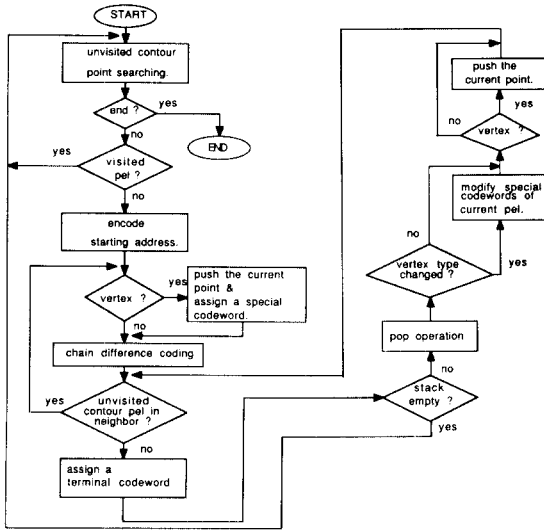
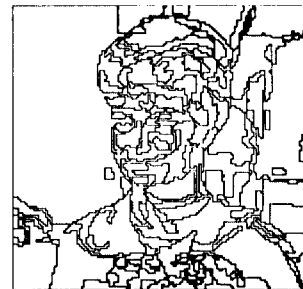


그림 7. 전체적인 부호화 흐름도

Fig. 7. Coding flow diagram.

(assuming that visited contour pels are deleted).



(a)



(b)

그림 8. 윤곽선 영상

(a) girl

(b) cronkite

Fig. 8. Contour images.

(a) girl.

(b) cronkite.

로 어떤 종류의 윤곽선 영상에 적용이 가능하다. 그러나 영역화를 이용한 부호화기의 윤곽선 영상은 split-merge, RAG 등에 의하여 얻어지므로 일반적으로 4-connected 윤곽선 영상으로 생각할 수 있다.^{2,7,8,9} 4-connected 윤곽선 영상은 8-connected 영상과 달리 현 위치에서 대각선 방향으로는 윤곽선 화소를 고려하지 않아도 된다.^{9,10} 즉 4-connected 영상에 대해서는 대각선 방향으로는 탐색하지 않고, 단순히 수평-수직 방향으로만 탐색을 하여 그림 7과 동일한 흐름도를 따라 부호화 할 수 있다.

표 1. 윤곽선 영상의 특성(종류)

Table 1. Contour images.

contour image	case	# of contour points	# of regions	case	# of contour points	# of regions
Girl	1	8575	153	4	11810	308
	2	9453	203	5	11647	343
	3	10934	251	6	12582	403
Cronkite	1	8592	156	4	10431	299
	2	9510	204	5	10553	355
	3	10131	269	6	11076	394

1. 제안 방식의 감축 성능비교

표 2는 제안 방식의 성능을 나타내는 표로서, 비교 방식으로는 중복을 피한 chain difference code이다. 중복을 피한 chain difference code란 한 영역에 대하여 closed path를 형성하면서 chain difference code

로 부호화 하고, 매 영역마다 이를 반복하여 부호화 하는 방식이다.

비교 특성으로는 초기 절대좌표의 갯수, 제안방식에 사용된 특수 code의 비율, bit/contour pel, 그리고 객관적인 비교 방법인 entropy이다. 여기서 특수

표 2. 성능비교 표

Table 2. The comparison of performance.

(GIRL)

case	coding method	# of starting coordinates	# of vertex	special codewords	total bits	bit/contour point	entropy
1	Modified-chain difference coding prosed method	194	135	2.75%	19866	2.317	1.398
		8			17446	2.035	1.149
2	Modified-chain difference coding Proposed method	254	186	3.33%	22462	2.376	1.461
		8			19220	2.033	1.149
3	Modified-chain difference coding Proposed method	315	230	3.57%	26278	2.403	1.477
		7			22270	2.037	1.140
4	Modified-chain difference coding Proposed method	372	278	4.02%	28828	2.441	1.524
		8			24102	2.04	1.15
5	Modified-chain difference coding Proposed method	413	316	4.52%	29076	2.496	1.580
		9			23830	2.046	1.162
6	Modified-chain difference coding Proposed method	488	365	4.96%	31996	2.543	1.636
		13			25850	2.055	1.174

(CRONKITE)

case	coding method	# of starting coordinates	# of vertex	special codewords	total bits	bit/contour point	entropy
1	Modified-chain difference coding Proposed method	185	143	2.75%	19774	2.301	1.262
		4			17442	2.029	1.027
2	Modified-chain difference coding Proposed method	242	191	3.22%	22408	2.356	1.333
		5			19342	2.033	1.042
3	Modified-chain difference coding Proposed method	318	249	4.02%	24714	2.439	1.427
		5			20660	2.039	1.064
4	Modified-chain difference coding Proposed method,	346	283	4.54%	25706	2.464	1.454
		5			21334	2.045	1.073
5	Modified-chain difference coding Proposed method	413	324	5.1%	26888	2.548	1.546
		6			21618	2.049	1.085
6	Modified-chain difference coding Proposed method	458	360	5.43%	28564	2.579	1.590
		5			22708	2.05	1.097

code란 초기 절대 좌표와 link에 할당되는 데이터 량을 제외한 데이터이고, entropy는 여러 부호화 방식으로 부호화 하였을 때 code에서 codeword가 각각 독립이라고 가정하고 code의 entropy를 윤곽선 화소당 평균 bit으로 환산한 값이다.

즉,

$$\text{entropy} \left(- \sum_{\text{codewords}} p_i \log_2 p_i \right) \times \frac{\text{code의 길이}}{\text{영상내의 윤곽선 화소의 개수}}$$

이다.

표 2의 결과를 보면, 제안방식은 초기 절대 좌표의 개수가 중복을 피한 chain difference code에 비해 상당히 줄어들었다. 이와 반면에, 10% 이내의 특수 code가 생겼는데 대부분이 교점들을 표현하는데 필요한 데이터 량이며, 중복을 피한 chain difference code에 필요한 초기 절대좌표의 데이터 량에 비교된다고 할 수 있다.

영역의 개수가 많아지는 것은 영상의 복잡성이 커지는 것을 의미한다. 영상이 복잡해지면 그만큼 그 영상을 표현하는데 필요한 데이터 량은 많아질 것이다. 표 2에서도 영역의 개수가 많아질수록 entropy 값이 커졌다.

M. Eden M. Kocher 방식에서는 윤곽선 화소당 1.2bit를 할당 시킨다.¹⁾ 물론 이 방법에 의한 entropy값은 윤곽선 영상을 표현하기 위한 정확한 값은 아니지만 표 2의 entropy 값에 비교될 수 있다.

2. 감축 기법의 적용결과

제안방식은 depth first search와 chain difference code를 이용하고, 가능한 한 한개의 codeword에 확률 분포가 집중되도록 특수 codeword를 사용하였기 때문에, "0"run이 많이 나타난다(표3). 만일, 이와 같은 code상의 특성과 영상의 특성을 이용한다면 좋은 감축 효과를 얻을 수 있을 것이다. 본 논문에서는 runlength coding과 Huffman coding, 그리고 영상의 특성을 제안방식에 적용시켜 감축 특성을 알아 보았다.

첫째 방법으로, codeword "0"의 runlength 확률 분포를 구하여 Huffman coding을 적용하였다. 이와 함께 초기 절대 좌표를 제외한 나머지 데이터와 runlength에 대한 prefix codeword도 Huffman coding을 하였다.

둘째, 방법은, split-merge 영역화 방법으로 영역화를 수행할 때 2 × 2 화소들을 1개의 화소로 가정하는 것이다. 이러한 방법으로 영역화를 수행한다면,

표 3. Codeword 표(그림2 이용)
Table 3. Codeword table(Use Fig2).

codeword의 종류	4-connected영상	8-connected영상
방향벡터(link)	0, 1, 2	0, 1, 2, 3, 4
node 정보의 표현 (prefix codeword+ type codeword)	2 0 2 1 2 2 1 1	5 0 5 1 5 2 5 3
terminal codeword		6
절대 좌표의 prefix codeword	3	6

1개의 방향 벡터로 2개의 contour 화소의 위치를 알 수 있다.⁴⁾¹⁰⁾¹¹⁾ 즉, 이렇게 얻어진 code에 Huffman coding을 적용 하였다. 이 방법에 의한 code에는 첫째 방법과는 달리 runlength의 길이가 1/2로 줄어들기 때문에 Huffman runlength coding보다는 Huffman coding이 효과적이다. 그림 9는 2가지 방법의 simulation 결과를 나타낸 그래프이다.

이상의 2가지 감축 방법을 제안방식에 적용시켰을 때, 최종적으로 중복을 피한 chain difference code로 부호화하는 경우보다 60% 이상 감축 되었다.

3. 제안방식이 영역화를 영상 부호화기법에 미치는 효과

제안방식은 윤곽선 영상을 효율적으로 데이터 감축하여 영역화를 이용한 부호화기의 감축 성능을 높인다. 이는 동일한 감축율이라면, 영역 부호화(texture coding)로 좀더 자세히 표현할 수 있는 여유를 제공하게 된다.^{12, 13)}

본 절에서는 한 영역을 표현하는 데이터량은 동일하다는 가정하에, 윤곽선 부호화에 M. Eden-M. Kocher 방식을 이용하는 경우와 비교하였다. 영역을 부호화하는 방법은 2차 다항식 근사화로 6개의 계수가 필요하다. 감축률 관계는 다음과 같다.

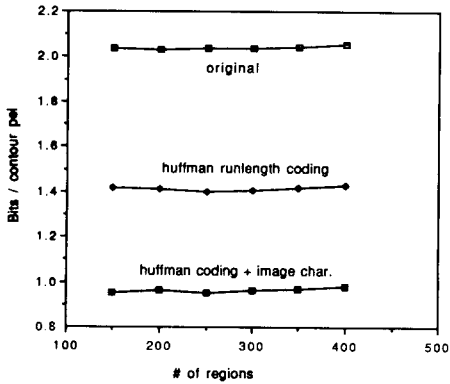
M. Eden, M. Kochdr의 방식이용 :

$$\text{감축률} = \frac{256 \times 256 \times 8}{1.2 \times \text{contour pel의 주+영역수} \times 6 \times (\text{bits/계수})}$$

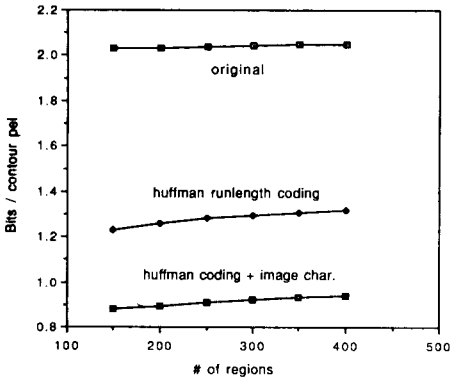
제안방식 이용 :

$$\text{감축률} = \frac{256 \times 256 \times 8}{\text{제안 방식의 contour code의 데이터량} + \text{영역수} \times 6 \times (\text{bits/계수})}$$

그림10은 윗 식들을 근거로 다항식 근사화의 계수당 10bit, 12bit 균등 양자기(uniform quantizer)를 이



(a) continue



(b)

그림 9. 제안방식에 감축 기법을 적용한 결과
(a) girl (b) cronkite

Fig. 9. The results of compression technique applied to the proposed method.

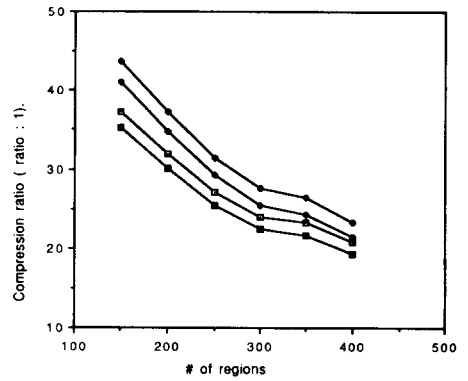
용한 감축률 비교 그래프이다.

그림10을 보면 제안방식은 기존의 방식보다 감축률이 높아졌다. 특히, 영역의 갯수가 적어질수록, 즉 감축률이 높아질수록 영역화를 이용한 부호화 방식에 미치는 영향력은 상당히 커진다. 감축율이 높아질수록 감축 효과가 20% 정도까지 크게 상승 되었다.

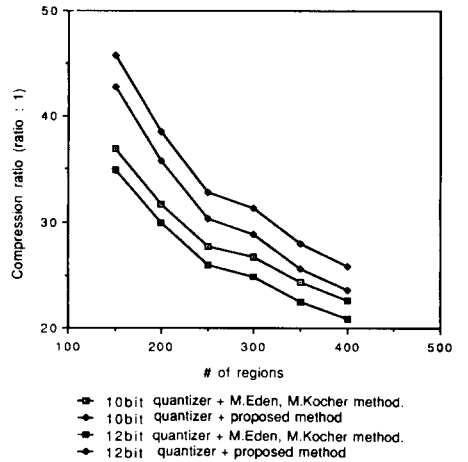
VI. 결 론

본 논문에서는 영역화를 이용한 부호화기의 윤곽선 영상을 데이터 감축하기 위하여 그래프 traversal 알고리즘의 하나인 depth first first search를 부호화에 이용하였다.

DFS를 이용함으로써 데이터 중복성 및 초기 절대



(a) continue



(b)

그림10. 감축율 비교 그래프

(a) girl (b) cronkite

Fig. 10. Comparisons of compression ratio.

좌표의 데이터 비율을 크게 줄였다. DFS를 부호화에 이용하기 위해 본 논문에서는 부호화에 필요한 교점 정보를 간략화하고, 특수codeword를 체계적으로 사용하였으며, Huffman coding에 의한 데이터 감축 효과를 높이기 위해 codeword의 분포를 고려하였다. 물론 중복을 피한 chain difference code와 비교한다면 약간의 복잡성과 2배가량의 시간이 소요되지만 데이터 감축효과를 고려한다면 효율적이라고 할 수 있다.

최종적으로 split-merge에 의한 윤곽선 영상에 적용한 결과, 중복을 피한 chain difference coding에 비해 60% 이상 데이터 감축이 되었다. 또한 영역화를 이용한 부호화 방식의 데이터 감축율은 M. Eden, M. Kocher 방식을 이용하는 경우보다 최대 20% 가량 증가 되었다.

参 考 文 献

[1] M. Kunt, A. Ikonomopoulos, M. Kocher, "Second generation image coding technique," *Proc. IEEE*, vol. 73, 4, 1985.

[2] M. Kocher, M. Kunt, "Image data compression by contour texture modelling," *SPIE*, vol. 397, 1983.

[3] H. Freeman, "Boundary encoding and processing," in the picture processing and psychopictories, Academic Press, New York, 1970.

[4] H. Freeman, "Computer processing of line drawing images," *Computer surveys*, vol. 6, no. 1, 3, 1974.

[5] M. Eden, M. Kocher, M. Eurasip, "On the performance of a contour coding algorithm in the context of image coding, Part I: contour segment coding," *Signal processing* 8, 1985.

[6] E. Horowitz, S. Sahni, "Fundamentals of data structures in pascal," Computer science press, 1982.

[7] S.W. Zuker, "Survey-region growing: childhood and adolescence," *CGIP* 5, 1976.

[8] S.L. Horowitz, T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *JACM*, vol. 23, 4, 1976.

[9] T. Pavlidis, "Structural pattern recognition," Springer-Verlag Berlin Heidelberg, New York, 1977.

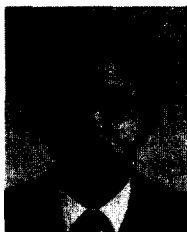
[10] H. Murakami, etc, "Algorithm for construction of variable length code with limited maximum word length," *IEEE Trans. comm*, vol. COM-32, no. 10, 10, 1984.

[11] D.A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, 9, 1952.

[12] M. Kocher, "Image representation by means of two dimensional polynomials," *SPIE*, vol. 594, 1985.

[13] M. Kocher, "Adaptive region growing technique using polynomial functions for image approximation," *Signal processing* 11, 1986. *

著 者 紹 介



金 鍾 勳(準會員)

1964年 1月 15日生. 1986年 2月 서강대학교 전자공학과 졸업. 1986年 3月~1988年 2月 한국과학기술원 전기 및 전자공학과 공학석사학위 취득. 1988年 2月~ 현재 한국과학기술원 전기 및 전자공학과 박사과정. 주관심분야는 영상부호화 및 영상처리 등임.



金 聖 大(正會員)

1953年 12月 26日生. 1977年 2月 서울대학교 전자공학과 학사학위 취득. 1979年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1983年 12月 E. N. S. E. E. I. H. T 박사학위 취득. 1984年 9月~현재 한국과학기술원 전기 및 전자공학과 조교수. 주관심분야는 영상인식, 영상부호화 및 영상처리 3차원 물체인식, 이동물체추적 등임.

金 漢 樹(正會員)

1960年 9月 19日生. 1983年 2月 고려대학교 전자공학과 졸업. 1983年 3月~1985年 2月 한국과학기술원 전기 및 전자공학과 공학석사학위 취득. 1985年~현재 한국과학기술원 전기 및 전자공학과 박사과정. 주관심분야는 영상부호화 및 영상해석등임.



金 在 均(正會員)

1938年 9月 17日生. 1962年 2月 한국항공대학 응용전자공학과 학사학위 취득. 1967年 2月 서울대학교 전자공학과 석사학위 취득. 1971年 8月 미국 남가주대 전기공학과 박사학위 취득. 1973年 4月~현재 한국과학기술원 전기 및 전자공학과 교수. 주관심분야는 영상데이터 처리 및 감축, 디지털 통신 시스템 전자통신 네트워크 등임.