

# 시작점 및 끝점 y축 분할표를 이용한 체인코드 기반의 영상복원 알고리즘

正會員 심재창\*, 이준재\*\*

## An Algorithm for the Restoration of Images Based on Chain Codes Description Using y-axis Partition of the Starting and Ending Points

Jae Chang Shim\*, Jun Jae Lee\*\* *Regular Members*

### 요 약

시작점과 끝점의 y축 분할표를 이용하여 체인코드 표현의 영상을 복원하는 빠른 방법을 제안하였다. 제안된 참조표는 시작점과 끝점이 구분되어 지며 이를 이용해 체인코드 표현을 두 개의 y축 분할표로 변환한다. 이 방법은 순서정렬 해야할 y축 분할표의 요소수가 반으로 줄어 처리 속도가 Shih의 방법보다 훨씬 빠르다. Shih의 방법과 처리 속도를 비교하였다.

### ABSTRACT

A fast algorithm for the restoration of an images based on chain codes description using y-axis partition of the starting and ending points: The algorithm is to convert the chain codes description into the two y-axis partitions by proposed look-up table. It reduces the size of column elements as a half in y-axis partition. This technique is computationally faster than Shih's methods. Comparison of the proposed method and Shih method is also provided.

### I. 소 개

체인코드 기반의 영상복원 알고리즘은 빠른 처리가 가능하고 쉽게 구현할 수 있어 많이 연구되었다.<sup>[1-3]</sup> 연속되는 두 방향 정보를 이용하는 방법<sup>[4-5]</sup>과 한 방향만을 이용하는 방법<sup>[6-7]</sup>으로 분류할 수 있다. 두 방향 정보를 이용하는 방법 중 [1]은 윤곽표시 참조표를 이용하여 영상을 복원하는 방법으로 다양한 문자 정보의 복원에 적용하였다. 이 방법은 영상 크기와 동일한 2개의 버퍼가 필요한 단점을 갖는다. [2]는 [1]의 방

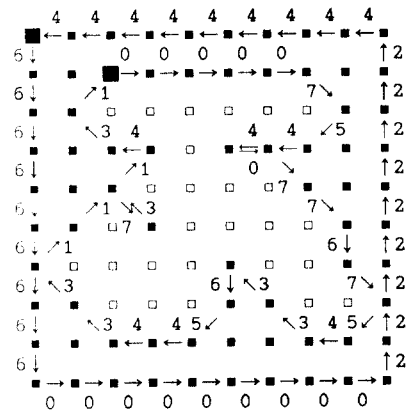
\*안동대학교 컴퓨터공학과  
Dept. of Computer Engineering, Andong National University  
\*\*동서대학교 컴퓨터공학과  
Dept. of Computer Engineering, Dong Seo University, Pusan, Korea  
論文番號: 95253-0801  
接受日字: 1995年 8月 1日

법을 개선하여 내부 및 외부 윤곽표시 참조표를 제안하고 y축 분할표를 이용해 다양한 특징을 추출할 수 있는 효과적인 방법이다. 그러나 내부 윤곽표시 참조표에서 4가지 경우에 오류가 발생된다. [3]에서는 [2]에서 사용한 외부 및 내부 윤곽용 참조표중 외부 윤곽표시 참조표를 내부윤곽에 그대로 적용하여 오류가 없도록 알고리즘을 보완하였다. [2-4]의 방법은 연속되는 체인코드의 두 방향 정보로 윤곽표시 참조표를 참조하여 y축 분할표를 만들고 이 표로부터 원 영상을 복원하는 방법이다. y축 분할표로 변환하는 방법은 영상의 복원뿐만 아니라 다양한 영역 특징을 추출할 수 있어 패턴인식 등에 매우 유용하다. 그러나 복잡한 영상이나 내부 윤곽이 많은 영상의 경우에 y축 분할표 순서정렬(sorting)에 시간이 오래 걸린다.

본 논문에서는 하나의 y축 분할표를 시작 y축 분할표와 끝 y축 분할표로 분리하여 순서 정렬할 요소를 반으로 줄일 수 있는 방법을 제안한다. 두 방향 체인코드 조합으로부터 패리티를 조사하여 [2]에서 사용한 참조표의 표시점(Marking)에 해당하는 부분을 시작점(Starting)과 끝점(Ending)으로 나누었다. 연속하는 두 체인 코드의 방향을 해석하여 시작점의 경우 시작점 y축 분할표에 끝점의 경우 끝점 y축 분할표에 이를 저장한다. 사용된 윤곽추적 규칙은 [1-4]와 마찬가지로 8방향 체인코드<sup>[1]</sup>과 오른쪽소 우선 윤곽추적 규칙<sup>[2]</sup>이다. 오른쪽소 우선 윤곽추적은 체인코드로 경계를 추적할 때 다음 이동지점을 체인코드 진행 방향에 대해 오른쪽의 화소를 미리 고려하는 법칙으로 물체의 외부와 내부 가장자리를 추적하는 알고리즘이다. 외부윤곽은 반시계 방향으로 추적하고 내부윤곽은 시계방향으로 추적한다. 그림 1과 같은 물체의 경우 추적된 체인코드 표현<sup>[1]</sup>은 다음과 같다.

s(1,1)666666666000000002222222244444444e  
s(3,2)00000754407767543365443311731431e\$

여기서 's'는 윤곽의 시작 표시며 'e'는 윤곽의 끝 표시다. 괄호 속의 두 수는 시작점의 열과 행 좌표이다. '\$'는 하나의 물체에 대한 끝 표시다. 내부에 구멍이 없는 경우 하나의 윤곽으로 표시되나 내부에 구멍이 여러 개 있을 경우 여러 개의 시작점을 가지나 하나의 '\$'표시를 가진다.



물체화소, □: 배경화소, 숫자: 방향코드, ■: 시작지점)

그림 1. 하나의 내부윤곽을 가진 물체의 예와 체인코드의 방향

Fig. 1 An example of object with an in-contour and chain codes

## II. 기존의 방법

[3]에서는 체인코드 데이터로부터 연속하는 두 체인코드로 윤곽표시 참조표를 이용해 표시점과 특이점에 해당하는 행 좌표를 y축 분할표에 삽입한다. 이 표에 저장된 원 좌표(x 좌표)는 임의의 순서로 저장되므로 이 표를 순서 정렬한 후 원 영상을 복원한다. 표 1의 윤곽표시 참조표는 연속되는 체인코드의 두 방향 정보인 현재화소를 중심으로 이전 체인코드와 다음 체인코드를 해석하여 작성한다. 이 표는 물체의 윤곽과

표 1. 윤곽표시 참조표  
Table 1. Look up-table for contour marking

d(i, j)	0	1	2	3	4	5	6	7	
d(i, j-1)	0	2	1	1	1	1	3	3	2
1	2	1	1	1	1	0	3	2	
2	3	1	1	1	1	0	0	3	
3	3	1	1	1	1	0	0	0	
4	1	3	3	2	2	1	1	1	
5	1	0	3	2	2	1	1	1	
6	1	0	0	3	3	1	1	1	
7	1	0	0	0	3	1	1	1	

0: singular point      1: marking point  
2: skipping point      3: unsuitable point

주사선이 만나는 회수인 패리티에 기반을 두고 있다.

이 표에서 '0'은 특이점(singular point)으로 주사선(가로 방향)에 대해 화소가 단독으로 존재하는 경우이다. '1'은 주사선에 대해 영역의 시작 또는 끝지점에 해당하는 표시점(marking point)이 되며, '2'는 주사선에 대해 시작지점과 끝지점 사이에 존재하는 무시점(skipping point)이다. '3'의 경우는 위에서 설명된 추적법칙에 의해 추적된 영역일 경우 발생되지 않는 부적합점(unsuitable point)이다. 현재지점을 기준으로  $d(i, j-1)$ 은 이전 체인코드의 방향이며  $d(i, j)$ 는 다음 체인코드의 방향이다.  $i$ 는 몇 번째 물체인가를 나타내고  $j$ 는 몇 번째 체인코드 인지를 나타낸다.

[3]의 방법은 체인코드 표현을 y축 분할표로 변환하는 알고리즘이다. y축 분할표는 각 주사선마다 물체의 윤곽 중 영역의 시작지점 및 끝지점과 특이점을 저장한 표이다. 이 표의 행은 주사선 번호를 의미하고 열은 표시점 및 특이점의 x좌표다. y축 분할표의 작성법은 연속되는 두 체인코드에 따라 표시점인 경우에 해당 좌표 값을 한번 분할표에 기록하고 특이점의 경우 두 번 기록한다. 복잡한 영상에서는 당연히 하나의 주사선에 대해 여러 개의 시작지점과 끝지점이 발생할 수 있다. 영상복원은 y축 분할표에서 각 주사선별로 홀수 번째 좌표로부터 짝수 번째 좌표까지 채운다. Shih의 알고리즘을 요약하면 다음의 세 단계로 이루어진다.

**Shih의 알고리즘**

**Step 1.** 연속하는 두 방향 체인코드로 윤곽표시 참조표를 이용하여

- 1) 표시점 'M' 이면 y축 분할표에 x 좌표를 저장
- 2) 특이점 'I' 이면 y축 분할표에 x 좌표를 2회 저장

**Step 2.** y축 분할표를 순서 정렬

**Step 3.** 각 줄마다 홀수 번째 지점에서 짝수 번째 지점까지 채움.

실 영상에서 내부 윤곽이 많거나 영상이 복잡할수록 y축 분할표의 각 행마다 저장된 열 좌표 수가 많아진다. 저장된 좌표의 수가 많을 수록 순서정렬에 시

간이 오래 걸린다. 이를 개선하기 위해 윤곽표시 테이블을 바꾸어 시작점표와 끝점표를 이용하여 행 좌표수가 반으로 줄어든 2개의 표를 순서 정렬하여 처리속도를 빠르게 한 알고리즘을 제안한다.

### III. 제안된 방법

8-방향 체인코드와 오픈화소 우선 윤곽추적 규칙이 적용될 경우 그림 2에서  $[d(i, j-1), d(i, j)]$ 가 [4, 5], [5, 6], [6, 6], [6, 7], [7, 0]의 경우 시작점이 되고 [0, 1], [1, 2], [2, 2], [2, 3], [3, 4]의 경우 끝점이 된다. 영상의 윤곽을 관찰해 보면 아래로 향하는 체인의 조합은 시작점이며 위로 향하는 체인의 조합은 끝점이 된다. 내부윤곽에서도 동일한 원리가 적용된다. 두 방향 체인코드에 의해 표현될 수 있는 64가지 경우의 수 중  $d(i, j-1)$ 이 0, 1, 2, 3 중의 하나이고  $d(i, j)$ 가 1, 2, 3, 4중

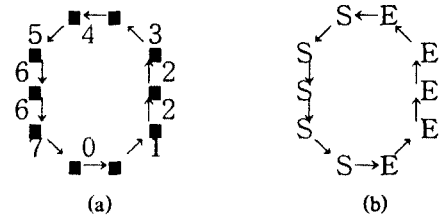


그림 2. 시작 및 끝지점의 분리 예 (a) 체인코드와 체인의 방향 (b) 시작 및 끝 지점표시

Fig. 2 An example of the starting and ending points (a) chain codes and its directions (b) marking of the starting and ending point.

표 2. 제안된 참조표

Table 2. A proposed look-up table

$d(i, j)$	$d(i, j-1)$	0	1	2	3	4	5	6	7
0		D	E	E	E	E	X	X	D
1		D	E	E	E	E	I	X	D
2		X	E	E	E	E	I	I	X
3		X	E	E	E	E	I	I	I
4		S	X	X	D	D	S	S	S
5		S	I	X	D	D	S	S	S
6		S	I	I	X	X	S	S	S
7		S	I	I	I	X	S	S	S

S : starting point    E : ending point    X : unsuitable point  
 D : don't care point(skipping point)    I : singular point

의 하나가 되는 16가지의 경우가 시작점이 되고,  $d(i, j-1)$ 이 4, 5, 6, 7 중의 하나이고  $d(i, j)$ 가 5, 6, 7, 0 중의 하나가 되는 16가지의 경우가 끝점이 된다. 이를 바탕으로 표 1을 바꾸어 표 2와 같이 새로 제안한다. 표 1에서 'I'로 표현된 표시점이 시작점 'S'와 끝점 'E'로 분리된다. 단순히 표시점으로만 인식하던 지점들이 시작점과 끝점으로 분리되어 시작점은 시작점 y축 분할표에 끝점은 끝점 y축 분할표에 저장할 수 있다.

여기서 'S'는 영역 시작점, 'E'는 영역 끝점, 'I'는 좌우가 고립된 특이점, 'D'는 무시점, 그리고 'X'는 부적합점이다. 'D'는 주사선을 따라가며 시작점과 끝점 사이에 존재되고 'X'는 정상적인 체인코드 표현일 경우 발생되지 않으므로 y축 분할표 작성에는 고려하지 않아도 되는 정보이다. 이 두 가지를 제외한 영상복원에 꼭 필요한 정보만을 포함한 참조표는 표 3과 같다.

표 3. 복원에 필요한 윤곽표시 참조표

Table 3. A look-up table for contour marking for image restoration

$d(i, j)$ $d(i, j-1)$	0	1	2	3	4	5	6	7
0	.	E	E	E	E	.	.	.
1	.	E	E	E	E	I	.	.
2	.	E	E	E	E	I	I	.
3	.	E	E	E	E	I	I	I
4	S	.	.	.	.	S	S	S
5	S	I	.	.	.	S	S	S
6	S	I	I	.	.	S	S	S
7	S	I	I	I	.	S	S	S

S: Starting point    E: Ending point    I: Singular point

여기서 'S'는 시작점 'E'는 끝점 그리고 'I'는 특이점을 의미한다. 위의 표를 적용하는 제안된 방법을 요약하면 다음과 같다.

제안된 알고리즘

Step 1. 연속하는 두 방향 체인코드로 윤곽표시 참조표를 이용하여

- 1) 시작점 'S'면 시작점 y축 분할표에 x 좌표를 저장
- 2) 시작점 'E'면 끝점 y축 분할표에 x 좌표를 저장

- 3) 특이점 'I'면 시작점 및 끝점 y축 분할표에 각각 좌표를 저장

Step 2. 시작 및 끝 y축 분할표를 순서 정렬

Step 3. 각 줄마다 시작 y축 분할표의 각 지점에서 끝 y축 분할표의 각 지점까지를 채움 (1:1로 대응됨)

#### IV. 실험 및 고찰

그림 1의 영상에 Shih의 알고리즘과 제안한 알고리즘의 윤곽 표시 범을 적용해 메모리에 표시점을 기록한 영상은 각각 그림 3(a)와 그림 3(b)이다. 그림 3(b)를 보면 시작점과 끝점이 명확히 구분되어 처리가 용이해 진다.

M K K K K K K K K M	S . . . . . E
M . K K K K K K . M	S . . . . . E
M M . . . . . M M	S E . . . . . S E
M . K M . M K . M	S . . E S . . . E
M . M . . . . . M . M	S . E . . . . . S . E
M M I . . . . . M M	S E I . . . . . S E
M . . . . . I . M M	S E . . . . . I S E
M M . . . . . M M . M	S E . . . . . S E . M
M . K K K . . . K K M	S . . . . . E
M K K K K K K K K M	S . . . . . E

M: Marking Point    S: Starting Point  
 K: Skipping Point    E: Ending Point  
 I: Singular Point    I: Singular Point

그림 3. 윤곽표시 예 (a) Shih의 알고리즘 (b) 제안된 알고리즘

Fig. 3 An example of contour marking (a) the Shih's algorithm (b) the proposed algorithm

Shih의 방법은 윤곽을 복원하면서 표 1을 이용해 표 3(a)와 같이 y축 분할표를 만들고, 이를 순서 정렬하여 표 3(b)를 얻는다. 이 표에서 홀수 번째 지점에서 짝수 번째 지점까지 채워 원 영상을 복원하게 된다. 이 표에서 1, 2, 9, 10열에는 각각 2개의 좌표 값이 3, 4, 5열에는 4개의 좌표 값이 그리고 6, 7, 8열에는 6

개의 좌표 값이 기록되어 있다. 모두를 순서 정렬해야 한다.

표 4. 그림 1의 y축 분할된 결과 (a) 순서정렬 이전 (b) 순서 정렬 이후

Table 4. The Result of Fig. 1 in y-axis partition (a) before sorting, (b) after sorting

Y(row)	X(column)	Y(row)	X(column)
1	1, 10	1	1, 10
2	1, 10	2	1, 10
3	1, 10, 9, 2	3	1, 2, 9, 10
4	1, 10, 6, 4	4	1, 4, 6, 10
5	1, 10, 8, 3	5	1, 3, 8, 10
6	1, 10, 9, 2, 4, 4	6	1, 2, 4, 4, 9, 10
7	1, 10, 9, 6, 6, 1	7	1, 1, 6, 6, 9, 10
8	1, 10, 10, 7, 6, 2	8	1, 2, 6, 7, 10, 10
9	1, 10	9	1, 10
10	1, 10	10	1, 10

제안된 방법을 적용하는 과정은 표 3을 이용해 시작점은 시작점 y축 분할표 표 4(a)의 X-Start에 끝점은 끝점 분할표인 X-End에 기록한다. 표 4(a)를 순서 정렬하여 표 4(b)를 구한다. 여기서는 1, 2, 9, 10열은 좌표가 1개뿐이므로 순서 정렬이 필요 없고 3, 4, 5열은 두 개의 좌표 그리고 6, 7, 8은 세 개의 좌표가 순서 정렬하게 된다.

표 5. 그림 1의 시작 및 끝 y축 분할된 결과 (a) 순서정렬 이전 (b) 순서정렬 이후

Table 5. The Result of Fig. 1 in starting and ending y-axis partition (a) before sorting, (b) after sorting

Y(row)	X-Start	X-End	Y(row)	X-Start	X-End
1	1	10	1	1	10
2	1	10	2	1	10
3	1, 9	10, 2	3	1, 9	2, 10
4	1, 6	10, 4	4	1, 6	4, 10
5	1, 8	10, 3	5	1, 8	3, 10
6	1, 9, 4	10, 2, 4	6	1, 4, 9	2, 4, 10
7	1, 9, 6	10, 6, 1	7	1, 6, 9	1, 6, 10
8	1, 10, 6	10, 7, 2	8	1, 6, 10	2, 7, 10
9	1	10	9	1	10
10	1	10	10	1	10

제안된 방법과 Shih의 방법을 비교해 보면 Shih의 y축 분할표의 내용을 시작 및 끝 분할표로 정확히 2개로 나누어지므로 순서정렬에서 시간을 많이 단축할 수 있다.

단순한 4개의 물체가 포함된 영상에 대한 입력영상 및 체인코드 그리고 복원된 영상의 예가 그림 4와 같다. 처리되는 과정을 살펴보면 원 영상에 대해 윤곽을 추적하여 (c)와 같은 체인코드를 구한다. 실제 전송 받거나 저장된 데이터의 형태는 (b)와 같은 체인코드 형태이며 영상의 복원은 복원 알고리즘에 의해 체인코드를 해석하여 처리한다. 체인코드 데이터 (b)로부터 복원된 영상은 (c)와 같다. 여기서 비교되는 윤곽 추적에 의한 압축기법은 압축과 복원 과정에서 손실이 없는 무손실 압축이어야 하므로 반드시 (a)와 (c) 영상은 동일하다.

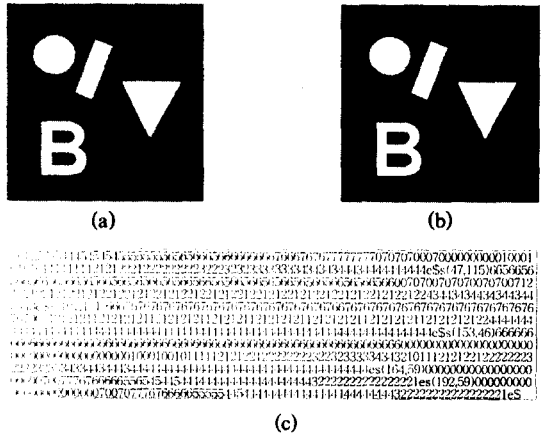


그림 4. 단순한 4개의 물체를 가진 영상 (a) 입력 영상 (b) 복원된 영상 (c) 영상 윤곽의 체인코드

Fig. 4 A simple example image with 4 objects (a) input image, (b) restored image (c) chain coded data

윤곽을 추적하여 구한 체인코드 표현으로 바꾼 (c)와 같은 시작점과 끝점 그리고 방향 정보를 가진 데이터를 하드 디스크에 저장하고, 이 데이터로부터 원 영상을 복원하는데 걸리는 시간을 비교하였다. 시간의 측정은 하드 디스크로부터 데이터를 읽는 시간을 제외하고 체인코드를 해석하여 영상을 복원하는 프로그램이 시작되는 지점과 처리가 끝나는 지점에서

각각 클럭을 호출하여 그 차이를 계산하였다. 실험에서는 제안된 방법과 Shih 방법의 처리 시간을 비교하였다. 실험에 사용된 영상은 그림 5와 같다. 물론 복원은 체인코드 추출에 사용된 입력영상과 동일하며 영상의 크기는  $256 \times 256$ 이다. (a)는 Ghill영상이며 (b)는 임계값 100을 가해 만든 이진 영상이고 (c)는 Canny 연산자를 적용해 구한 굵은 영역이다. (d)는 Canny 연산자를 적용해 구한 세밀한 에지이다. 실험에서는 먼저 (b), (c) 및 (d) 영상에서 체인코드 정보를 추출하여 하드디스크에 저장하고 이 체인코드로 데이터로부터

두 방법을 적용하여 영상을 복원하여 비교하였다. 두 방법에 대한 수행 시간의 비교는 표 4와 같으며 시간의 단위는 초이다. 사용된 컴퓨터는 IBM 486/50이고 그래픽 카드는 WD Super VGA를 사용하였다. 제안



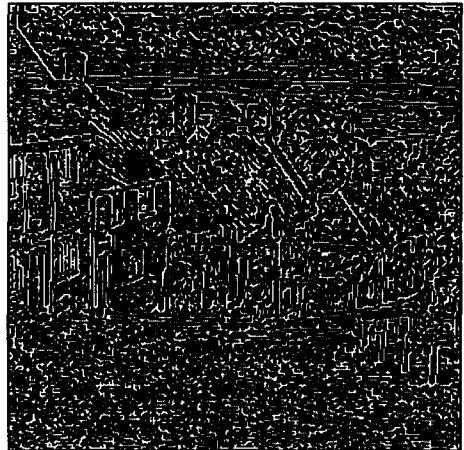
(a)



(b)



(c)



(d)

그림 5. 실험에 사용된 영상 (a) 256 명암도의 원 영상 [Ghill.img] (b) 이진화된 영상[Ghill.thr] (c) 굵은 영역 영상[Ghill.blob] (d) 세밀한 에지 영상[Ghill.edg]

Fig. 5 Images for experiments (a) input 256 gray-level image, (b) thresholded image (c) blob image and (d) fine edge image.

된 방법이 항상 Shih의 방법보다 빠르다. 그리고 영역의 특징추출 등에서 제안된 방법도 y축 분할표 방식에서와 동일하게 적용되어 진다. y축 분할표에서 홀수지점과 짝수지점을 구분하여 처리하는 대신 시작점표와 끝점표를 1:1로 대응시키면 처리가 가능하다. 영상에서 순차정렬해야할 내용이 많을수록 제안된 방법의 적용이 우수해 진다.

표 6. 제안된 방법과 Shih의 방법의 처리 시간 비교표  
Table 6. Performance comparison of the Shih's method and the proposed method

(단위: 초)

Method	Image	Ghill.thr	Ghill.blo	Ghill.edg
Shih		0.22	0.54	1.70
Proposed		0.10	0.38	1.54

### V. 결 론

Shih의 체인코드로부터 영상복원 방법은 영역의 특징 추출에는 매우 효과적인 방법이나 복잡한 영상에서 y축 분할표를 순서정렬하기 위해서 시간이 오래 걸린다. 본 논문에서는 두개의 y축 분할표를 이용하여 영상을 빠르게 복원하는 방법을 제안하였다. 기존의 윤곽표시 참조테이블의 표시점을 시작점과 끝점으로 분리한 새로운 y축 분할표를 이용해 시작점 및 끝점 y축 분할표로 변환하므로 순서정렬에 걸리는 시간을 많이 단축시켰다. 다양한 실 영상에 적용해본 결과 제안된 방법이 우수함이 입증되었다.

### 참 고 문 헌

1. Z. Cai, "Restoration of Binary Images Using Contour Direction Chain Codes Description," *Computer Vision, Graphics, and Image Processing* Vol. 41, pp. 101-106, 1988.
2. L. W. Chang and K. L. Leu, "A Fast Algorithm for Restoration of Images Based on Chain Codes Description and Its Applications," *Computer Vision, Graphics, and Image Processing* Vol. 50, pp. 296-307, 1990.

3. F. Y. Shih and W. T. Wong, "An Improved Fast Algorithm for the Restoration of Images Based on Chain Codes Description," *CVGIP: Graphical Models and Image Processing* Vol. 56, No. 4, pp. 348-351, 1994.
4. G. Y. Tang and B. Lien, "Region Filling with the Use of the Discrete Green Theorem," *Computer Vision, Graphics, and Image Processing* Vol. 42, pp. 297-305, 1988.
5. S. M. Ali and R. E. Burge, "A New Algorithm for Extracting the Interior of Bounded Regions Based on Chain Coding," *Computer Vision, Graphics, and Image Processing* Vol. 43, pp. 256-264, 1988.
6. 심재창, 새로운 윤곽추적 기법을 이용한 영상처리 및 질감인식, 경북대학교 박사학위논문, 1993.
7. 심재창, 이준재, 하영호, "윤곽추적 영역채색 기법," 대한전자공학회 논문지 제 31권 B편 제5호, pp. 95-107, 1994.
8. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 2, Academic Press, New York, 1982.
9. T. Pavlidis, *Algorithm for Graphics and Image Processing*, Comput. Sci., Washington, D. C., 1982.



심재창(Jac Chang Shim) 정회원  
1962년 3월 27일생  
1987년 2월: 경북대학교 전자공학과(학사)  
1993년 8월: 경북대학교 전자공학과(박사)  
1993년 1월~4월: IBM T.J.Watson 연구소(방문연구원)

1993년 3월~현재: 안동대학교 컴퓨터공학과 조교수  
1997년 2월~1998년 2월: IBM T.J.Watson 연구소 Post Doc.(예정)

※주관심분야: 영상처리, 패턴인식, 컴퓨터비전, Video Indexing, VOD



이 준 재(Jun Jae Lee) 성회원

1962년 5월 31일생

1986년 8월:경북대학교 전자공학과 학사

1990년 8월:경북대학교 전자공학과 석사

1994년 8월:경북대학교 전자공학과 박사

1991년 9월~1993년 7월:경북대학교 조교

1995년 3월~현재:동서대학교 컴퓨터공학과 전임강사

※주관심분야:영상처리, 컴퓨터 비전, 신경회로망 등임