

체인코드의 방향정보를 이용한 Filling과 Labelling

(Filling and Labelling Algorithm Using Directional Information of Chain-code)

沈載昌*, 河錦淑**, 玄基浩*, 河永浩*

(Jae Chang Shim, Keum Sook Ha, Ki Ho Hyun, and Yeong Ho Ha)

要 約

물체의 윤곽 내부를 채우는 filling과 내부가 채워져서 이웃하는 화소와 연결되어진 각각의 영역에 고유 번호를 매겨주는 labelling을 동시에 실시할 수 있는 새로운 방법을 제안하였다. 에지영상이 입력되면 각 영역의 영상을 반전하여 filling된 영상을 얻고 각 영역의 윤곽을 따라가며 주사선 마다의 영역 filling과 labelling할 주사선 마다의 시작점과 끝점을 할당하며 외부윤곽은 시계 방향으로, 내부윤곽은 반시계 방향으로 추적하여 정보를 추출한다. 제안된 알고리즘은 내부에 다른 영역이 중첩되어 있을 경우에도 예외없이 적용이 가능하며 labelling이 끝남과 동시에 물체의 구성tree도 동시에 얻을 수 있는 방법이다.

Abstract

A new algorithm for filling the interior of contours and labelling each filled region concurrently is presented. Filling is simply accomplished by inversion method. The labelling information in every scan lines is extracted directly from current direction of chain code so that the proposed algorithm needs less comparison and is more efficient. The contours are followed by two different directions, clockwise for the outer contour and counterclockwise for the inner contour to get filling and labelling information. This algorithm can be applied in case that contours are nested or regions are continous. Simulataneously the proposed algorithm can find the structure tree of object without additional post processing.

I. 서 론

Filling은 윤곽이 주어질때 윤곽으로 둘러 쌓여있는 영역의 내부를 채우는 영상처리 기법이며, labelling은

채워진 영역에 고유한 번호를 매겨주는 방법이다. 각 영역들의 윤곽은 에지중심법이나 영역중심법의 영상 분할 과정을 통하여 구할 수 있으며 그 다음 과정으로 물체의 특징을 찾거나, 정합, 인식 또는 분류를 위해서 서로 이웃이 연결된 영역마다 고유의 번호를 부여하는 labelling과정을 거치게 된다. 이는 컴퓨터 그래픽스에도 많이 응용되며, 영상합성등에서 윤곽을 그린 다음 내부를 특정한 색으로 채울경우 이용된다. 또한 벡터 방식의 문자폰트에서는 문자의 윤곽정보를 저장하고

* 正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)

** 正會員, 龜尾專門大學 電算科
(Dept. of Computer Science, Kumi Junior College)
接受日字 : 1991年 8月 17日

있다가 화면에 표시하거나 인쇄시 내부를 채우는 filling과정이 필요하다. 위의 두가지 영상처리 기법은 서로 깊게 연관을 가지며 대부분이 이 두가지를 연속하여 처리하므로 한꺼번에 수행할 수 있는 방법이 필요하다. Filling과 labelling에서 반드시 고려되어야 할 사항은 수행의 정확성, 빠른 처리속도 그리고 내부에 영역이 중첩되어 있을 경우에도 예외없이 적용이 가능하여야 한다.^[1]

Filling방법은 연속적인 윤곽을 이용하는 방법^[1,2]과 불연속적인 화소를 이용하는 경우로 나눈다. 화소단위 방법에는 패리티체크(parity check)^[3], 연결 해석법(connectivity filling)^[4], 다각형 분할법(decomposition of polygon)^[5], 체인코드법(chain-code)^[6-9], 블럭이용법(blocking technique)^[10], 쿼드트리법(quad tree)^[11-12]등의 방법이 제안되어 있다. 체인코드를 이용하는 방법은 적용의 간결성으로 인해 근간에 많이 연구되고 있다. 체인코드를 이용한 방법에서 Cai의 방법^[6]이나 Chang의 방법^[7]은 현재화소 방향과 이전화소 방향의 체인방향을 비교하므로 각 화소마다 특징값을 부여하기 위해서는 최대 64회의 비교과정을 거쳐야 한다. S. M. Ali방법도 체인의 두화소의 체인방향의 비교가 필요하므로 위와 동일한 비교과정이 필요하며 내부의 영역을 고려해주지 못한다. J. C. Shim^[9]의 방법은 현재화소의 방향만 고려하나 내부에 영역이 있을 때는 Long-wen Chang의 방법처럼 채워진 내부를 다시 없애야 하는 단점이 있다. 제안한 방법에서는 현재화소만의 체인방향으로 주사선상에서 영역의 출발지점, 끝지점, 내부지점의 좌표값을 구분하므로 최대 8회의 비교과정이 필요하며 위의 방법들^[6,7]보다 비교회수를 줄일 수 있고 처리속도가 빠르며 내부에 영역이 여러겹으로 겹쳐있어도 예외없이 영역을 채울 수 있다. 그리고 화소 중심법에서 뿐만 아니라 연속적인 경우에도 적용이 가능하다.

Labelling의 방법에는 병렬처리방법^[13-14]과 순차처리방법으로 나누며 순차처리 방법에는 화소labelling 방법^[15], 런레스 연결해석법(run length connectivity analysis)^[16], 쿼드트리 이용법^[17]등이 있다. 화소labelling 방법은 간결한 알고리즘인 반면 동일한 영역을 여러가지 고유번호로 할당할 수 있으므로 하나의 영역으로 고쳐주는 후처리 과정이 필요하다. 런레스 연결해석법과 쿼드트리 이용법은 효율적인 반면 프로그램이 복잡하다. 본 논문에서 제안한 방법은 영역을 반전하므로 간단히 filling 된 영상을 얻고 이 영상에 외부와 내부경계를 반시계방향으로 추적하므로 labelling된 영상을 구하며, labelling을 마침과 동시에 물체의 구성 관계를 알 수 있는 구성tree를 얻을 수 있다.

II. 체인코드의 데이터 표현 및 contour 추적방법

체인코드는 이진영상을 표현하거나 코딩할때 유용하게 사용되는 영상의 표현방법이다. 고려되는 방향에 따라 4방향과 8방향 체인코드가 있으며 예시를 표현하는 데는 8방향 체인코드가 유용하고 영역을 추적하는 데는 4방향 체인코드가 이용된다. 8방향성 체인코드는 그림 1과 같으며 위로 향하는 방향은 1, 2, 3 아래로 향하는 방향은 5, 6, 7 좌우 방향은 4, 0 이다.

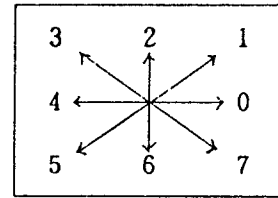


그림 1. 8방향 체인코.

Fig. 1. 8-directional chain coder

1. 윤곽 데이터의 표현방법

윤곽 데이터의 체인코드 표현방식은 Cai방법을 사용하며 식(1)과 같이 표현한다.

$$\begin{aligned} S X_{m0} Y_{m0} D_{m0} D_{m1} \cdots \cdots D_{m00} E \\ S X_{m0} Y_{10} D_{10} D_{11} \cdots \cdots D_{1m1} E \end{aligned} \quad (1)$$

$$S X_{m0} Y_{m0} D_{m0} D_{m1} \cdots \cdots D_{mmm} E \$$$

여기서 S, E는 체인의 시작과 끝을 나타내며, 첨자 i는 i번째 윤곽경로를 나타낸다. $X_{i0} Y_{i0}$ 는 i번째 윤곽의 출발 좌표값이며 (X_{i0}, Y_{i0}) 에서 시작한 윤곽은 원래 위치인 (X_{i0}, Y_{i0}) 에서 끝난다. D_j 는 i번째 윤곽에서 j번째 화소로 향하는 방향성 체인코드로서 그림 1과 같이 0-7중의 하나의 값을 갖는다. \$는 한영역의 끝을 표시하는 기호이며 다음 영역은 \$ 다음부터 계속하여 표시된다.

2. 추적방법

윤곽을 추적하는 방법은 왼쪽경계 추적법인 LML(left-most looking)^[11] 사용한다. 윤곽을 따라 사람이 걷고 있다고 가정하고 현재화소에서 옮겨야 할 다음 화소를 선정 할때 왼쪽화소부터 시계방향으로 차례로 돌면서 영역화소가 있는지 찾아보고 옮겨가는 방법이다.

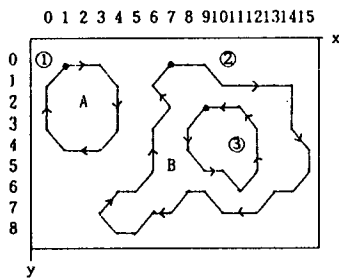
이방법은 항상 영역의 최외각 윤곽을 추적하게 된다. 본 논문에서 제안하는 추적방법은 영역의 외부윤곽을 시계방향으로 추적하고, 내부윤곽은 반시계방향으로 추적하면서 체인코드를 부여한다. 양방향 추적방법은 에지영상을 filling한후 각 영역을 labelling 할때 효율을 극대화 할 수 있게된다.

그림 2는 두 물체가 있는 영상에서의 추적 방향을 나타내고 있으며 그림 2(b)의 'o'은 '1'값을 갖는 화소이고 그외는 '0'을 갖는 화소에 해당하는 이진영상이다. 이에 대한 체인코드값은 식(2)와 같다. 외부윤곽인 ①②는 시계방향으로 추적하고 내부윤곽 ③은 반시계방향으로 추적한다.

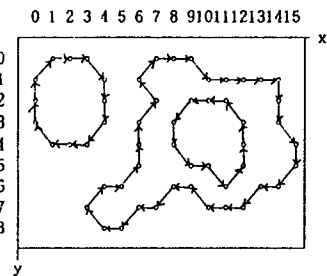
$$S 001 000 007665443221 E \text{ \textcircled{1}}$$

$$S 007 000 0070000667654544345454310122131 E \text{ \textcircled{2}}$$

$$S 009 002 567071223441 E \text{ \textcircled{3}}$$



(a)



(b)

그림 2. 두물체와 hole을 갖는 이진영상

- (a) 추적방향
- (b) 체인방향

Fig. 2. Binary image having two objects and a hole.

- (a) boundary following direction,
- (b) chain code direction,

III. 제안한 방법

에지영상이 입력되면 각각 독립된 영역의 외부경계를 찾고 이를 반전하므로 filling된 영역을 얻는다.(입력영상이 filling되어진 영상일 경우 반전과정은 생략됨) 각 영역을 순차 주사하면서 만나는 영역의 외부윤곽을 시계방향으로 LML 방법을 적용하여 돌면서 주사선상의 영역의 시작점과 끝점을 추출한다. 출발점으로 되돌아오면 주사선 마다 시작점에서 끝점 사이를 고유번호로 채워나간다. 이때 영역안에 내부윤곽이 있는지 조사하고 내부윤곽을 만나면 내부윤곽을 반시계방향으로 돌면서 시작과 끝 정보를 추출한다.

그림 2와 같이 한 영역을 갖는 이진영상에서 외부윤곽을 따라가며 아래로 향하는 체인코드수의 합과 위로 향하는 체인코드수의 합은 식(3)과 같이 동일하다. Green의 정리에 의하면 폐경로를 따라가며 위로 향하는 체인방향 만큼 아래로 향하는 체인의 방향이 존재하게 된다. 이를 이용하므로 한 주사선상에서의 각 영역의 시작지점과 끝지점의 정보를 추출할 수 있다.

$$\sum \text{1, 2, 3 방향갯수} = \sum \text{5, 6, 7 방향갯수} \quad (3)$$

체회로 체회로

체인이 아래로 향하는 5, 6, 7방향은 영역의 끝점이 되고 위로 향하는 1, 2, 3방향은 영역의 시작점이 된다. 내부에 다른영역이 중첩될 경우 내부경계에 시작지점과 끝지점 정보를 추출하여 추가하여도 위의 식은 성립되어진다. 위의 원리를 이용하여 외부경계는 시계방향으로 내부경계는 반시계방향으로 추적하여 각 주사선상의 영역 시작점과 영역의 끝점을 구별하는 방법이다.

1. 영상반전을 이용한 filling

입력영상인 에지영상을 반전하므로 각 영역이 에지에 의해 분리된 영상을 얻는다. 배경을 제외시키기 위해 독립된 영역이 존재하는 영상에서는 각각의 영역의 최외각 에지를 찾고 그 내부를 반전하므로 filling된 영상을 얻을 수 있다. 먼저 입력영상에서 주사선의 왼쪽 위 부터 순차주사 하면서 에지를 만나면 에지의 최외각을 II 장 2절의 LML추적방법으로 시계방향으로 돌면서 적용하여 'S' 'E' 'K'를 부여한다. 체인의 방향이 아래로 향하는 5, 6, 7이면 체인 도착화소에 각 주사선의 영역끝 표시인 'E'를 부여하고 체인의 방향이 위로 향하는 1, 2, 3방향이면 체인의 출발화소에 영역시작 표시인 'S'를 부여한다. 체인 출발화소와 체인 도착화



그림 3. 체인 출발 및 도착화소
Fig. 3. Chain code departure pixel and arrival pixel.

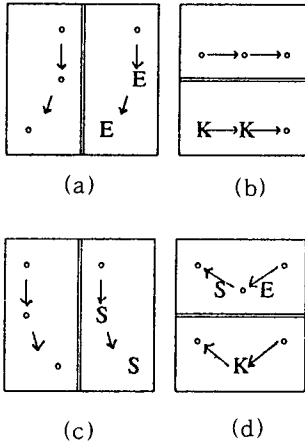


그림 4. 'S' 'E' 및 'K' 부여 좌표위치
(a) 아래로 향할때
(b) 좌우로 향할때
(c) 위로 향할때
(d) 아래꼭지점
Fig. 4. Pixel position to assign 'S' 'E' and 'K'
(a) down direction,
(b) left and right directions,
(c) upper direction,
(d) vertex point.

소는 그림 3에 나타나 있다. 이 두경우 이외는 체인 방향이 0이나 4로 접선화소(tangent pixel)이며, 'E'로 표시된 후 그위에 다시 'S'로 표시되는 경우는 꼭지점(vertex)이거나 영역만남지점(multi-junction) 이므로 체인의 출발지점 좌표에 'K'를 표시한다. 'S' 'E' 'K'를 부여하는 화소위치는 그림 4와 같다. 체인코드가 아래로 향할때와 위로 향할때 체인의 도착지점 좌표나 출발지점 좌표가 서로 한 화소 어긋나므로 이를 보상해 주기위해 위로 향할때는 체인의 출발 지점좌표에 'S'를, 아래로 향할때는 체인의 도착지점 좌표에 'E'를 각각 부여한다. 물체의 시작지점과 동일한 주사선상인 제일 윗줄은 모두 'K'가 부여 되어진다.

'S' 'E' 'K'가 부여된 영상에서 각 주사선을 따라가며 'S'에서 'E'사이의 화소를 반전 시킨다. 만약 화소값이 '1'이면 '0'으로 '0'이면 '1'로 바꾸고 'K'인 경우도 내부영역이므로 '1'로 바꾼다. 이때 각 영역의 시작

점은 다음 과정에서 이용하기 위해 위치를 저장한다. 각 영역은 서로 독립되게 되며 분리된 각각의 영역에 제안된 labelling방법을 적용한다. 독립된 영역은 영역의 구성 tree에서 최상위셀이 되며 반전된 영역내부에 또다른 영역이 있으며 하위셀을 구성되게 된다. 이에 대한 정보도 함께 저장한다.

2. 각영역의 labelling

영상반전시 찾아둔 영역의 시작점에서 시작하여 II 장 2절의 추적방법을 이용하여 영역의 외부윤곽을 시계방향으로 따라가면서 1.의 방법과 동일하게 'S' 'E' 'K'를 구한다. 각주사선을 따라가며 'S'에서 'E'까지를 고유번호 K로 채워 나간다. 이때 채울지점이 영역표시인 '1' 또는 'K'이면 K를 부여하나 경계표시인 '0'이면 영역내부에 또다른 영역이 중첩되어 있는 경우가 된다. '0'이 나타난 지점 바로 이전 화소가 현재 채우고 있는 영역의 내부경계 시작점이 되고 '0'이 나타난 바로 다음 화소가 중첩된 내부영역의 외부경계 시작점이 되어진다. 이 두지점을 저장시키고 현재의 labelling을 잠시 중단하고 내부경계의 시작점으로 부터 반시계방향으로 LML을 적용하면서 1.과 동일한 방법으로 'S' 'E' 'K'를 부여한다. 시작지점에 되돌아 오면 중단한 labelling작업을 계속 진행한다. 내부경계에 시작점과 끝점이 부여되므로 원래의 영역시작점과 끝점사이는 두개로 분리되며 내부의 중첩된 영역을 피해 영역채움이 이루어진다. 현재영역이 다 채워지고 나면 기억해둔 내부영역의 시작지점을 찾아가 위의 방법을 반복하여 영역을 채운다.

제안한 알고리즘을 요약하면 다음과 같다.

<영역반전을 이용한 영상의 filling>

A. 입력영상에서 서로 구분되어지는 외부경계를 찾는다.

(a) 왼쪽, 위 화소부터 에지를 만날때까지 주사.

(각영역의 시작점을 기억, 물체 구성의 최상위 셀 시작지점)

(b) 에지의 최외각을 시계방향으로 LML방법을 적용하면서 'S' 'E' 'K'부여

i) 체인 방향이 아래로 향하면 체인도착지점에 'E'

ii) 체인 방향이 위로로 향하면 체인출발지점에 'S'

iii) 체인 방향이 좌우로 향하면 체인출발지점에 'K'

iv) 'E'가 부여된후 다시 'S'가 부여되는 꼭지점은 'K'

B. 외부경계의 내부를 반전하여 filling된 영상을 얻는다.

(a) 시작점에 되돌아오면 각 주사선마다 'S'에서 'E'까지의 영상을 반전

- i) '1'은 '0'으로
- ii) '0'은 '1'로
- iii) 'K'는 '1'로

<각영역마다의 labelling (물체구성의 하위셀)>

A. 각영역의 시작점에서 외부경계를 시계방향으로 돌면서 'S'E'K'를 부여 한다.

B. 각주사선 마다 'S'에서 'E'를 고유번호 K로 채운다.

(a) 채울지점이 영역표시화소인 1이나 'K'이면 K를 부여

(b) 채울지점이 경계표시화소인 '0'이면 내부영역이 존재함.

('0'지점 바로이전 화소: 현재영역의 내부경계 시작점 기억, '0'지점 바로다음 화소: 내부영역의 외부경계 시작점 기억)

(c) 채우던 영역의 내부경계('0'이 있는 화소 바로이전 화소)를 따라 반시계 방향으로 'S'E'K' 부여

C. 내부영역 시작지점 부터 1, 2.과정 반복

(내부영역 속에 또다시 영역이 포함되어도 반복적으로 수행)

3. 그래픽에서의 적용

Filling과 labelling이 가장 많이 쓰이는 곳은 그래픽 시스템이다. 그래픽 시스템에서 메모리를 효율적으로 사용하면, 제공되는 그래픽 명령어를 적절히 이용할 수 있다. 34010 그래픽 시스템 프로세서¹¹⁾에서는 두좌표 사이를 채우는 그래픽 명령어인 fill이나 pixblt를 이용할 수 있고 MSC(MicroSoft-C)에서는 두좌표 사이를 채우는 -moveto(x, y)와 -lineto(x, y)명령과 한점을 채우는 -setpixel(x, y)명령을 이용하여 그래픽 화면에 filling시킬 수 있다. -moveto(x, y)와 -lineto(x, y)를 매크로로 묶어 -line(x₁, y₁, x₂, y₂)를 정의하여 사용한다. 체인의 방향코드를 이용하여 윤곽화소 마다 'S'E'K'를 표시하는 영상좌표에서 주사선(행)에 해당하는 y에 대해 체인방향이 위로 향할때는 체인 코드의 출발지점의 x좌표값을 버퍼에 기록하고, 체인방향이 아래로 향할때는 체인의 도착지점 x좌표값을 버퍼에 기록한다. 이 두지점은 선으로(line 명령) 연결하기 위한 좌표를 구하기 위한 버퍼이다. 'K'에 해당되는 화소는 출발지점의 x좌표를 점(-setpixel)명령을 위한 버퍼에 기억시킨다. 내부에 영역이 있는 경우 앞장의 설명과 같이 반시계 방향으로 추적하면서 위의 과정을 반복한다. 이때에는 각 y

행에 대한 버퍼마다 2개 x좌표값이 기록되어 있으므로 그 사이에 삽입한다. 이렇게 하므로 일반적인 방법에서의 sorting 과정을 피할 수 있으므로 처리속도를 향상할 수 있다. 각 y값에 대한 x값을 한쌍식 (y_i, x_{k1}), (y_i, x_{k2}) 묶어 선명령으로 -line(y_i, x_{k1}, y_i, x_{k2}) 화면을 labelling하고 점명령을 위한 버퍼는 -setpixel(x₁, y₁) 명령으로 해당화소를 labelling한다. 표 1은 그림 2의 A물체를 따라가며 선명령을 위한 버퍼와 점명령을 위한 버퍼에 좌표값을 기억시킨 결과이며 표 1-2는 B물체를 따라가며 버퍼에 기억시킨 결과이다. 명령과 명령으로 filling 시킨영상은 그림 6과 동일하다. 이 방법은 필요한 메모리를 최적화할 뿐만아니라 각 영역마다 병렬 분산처리가 가능하다. 각 영역들의 데이터표현은 II절에서 설명된 데이터 표현방식으로 되어 있다면 영역 마다 \$표시는 각영역을 완전히 독립적으로 분리시켜준다. 병렬 분산처리 시스

표 1. A물체를 위한 버퍼
Table 1. Buffer for object 'A'.

주사선	line 버퍼		pixel 버퍼		
	S	E	I1	I2	I3
0	-		1	2	3
1	0	4	-		
2	0	4	-		
3	0	4	-		
4	1	3	2		

표 2. B물체를 위한 버퍼
Table 2. Buffer for object 'B'.

주사선	line 버퍼				pixel 버퍼				
	S	E	S	E	I1	I2	I3	I4	
0	-				7	8	9		
1	6,	10			11	12	13	14	
2	7,	14			9	10	11		
3	8,		12,	14	-				
4	6,	8	12,	15	-				
5	6,	10	12,	15	9				
6	5,	14			13,	9,	8,	4,	11
7	3,	7	10,	12	6,				
8	4,	5			-				

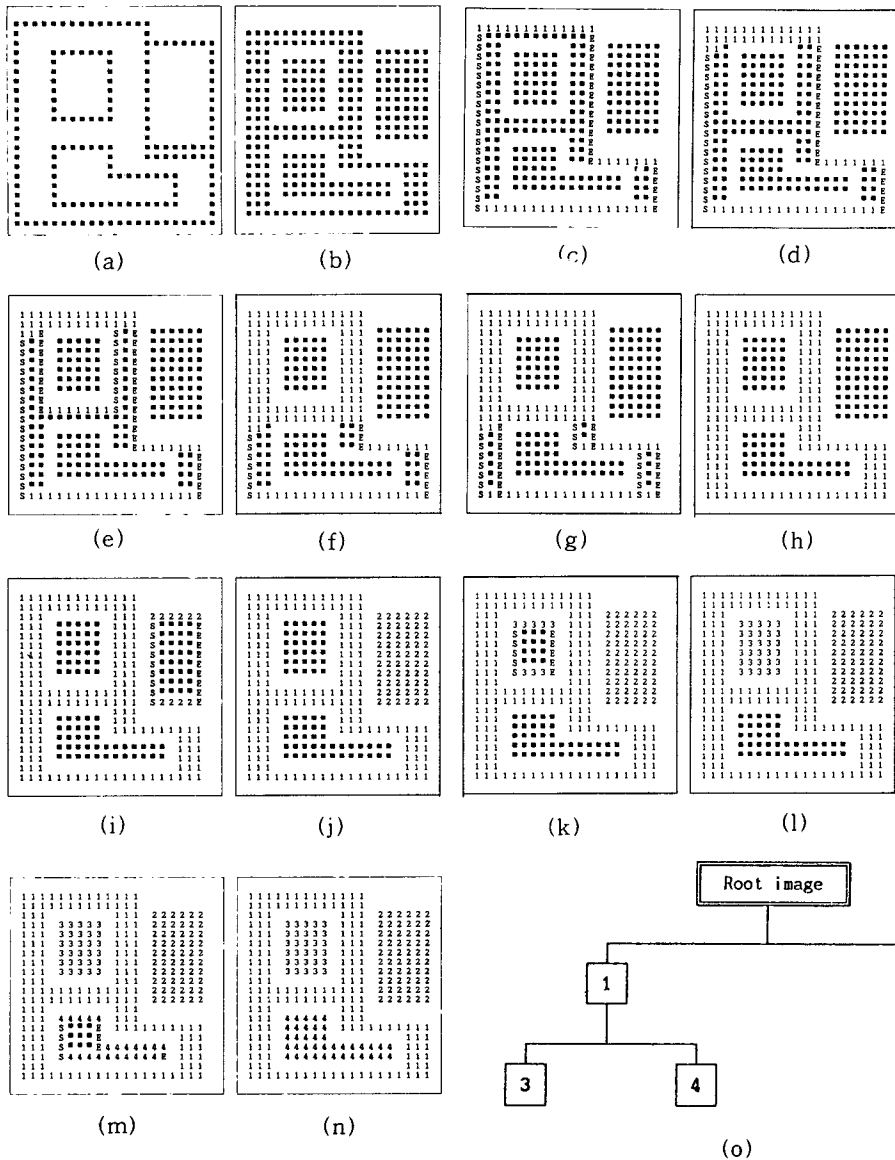


그림 5. 제안된 방법의 순차적 적용예

- (a) 입력 에지 영상 (b) 영역반전을 이용한 filling된 영상 (c) 첫영역의 외부경계 추적
- (d) 내부영역을 만났을 경우 (e) 내부경계를 반시계 방향으로 추적한 영상 (f) 두번째 영역을 만난 경우
- (g) 내부경계를 반시계 방향으로 추적 (h) 첫영역의 labelling 완성 (i) 두번째 영역의 외부윤곽을 시계방향 추적
- (j) 두번째 영역 labelling 완료 (k) 첫 내부영역의 시계방향 추적 (l) 첫 내부영역의 labelling 완료
- (m) 두번째 내부영역의 시계방향 추적 (n) 전체영역의 labelling 완료 (o) 물체의 구성 트리

Fig. 5. Example to show algorithm steps

- (a) edge image, (b) filled image using region inversion,
- (c) following cont clock-wise of 1st outer contour, (d) existing first internal region,
- (e) following count clock-wise of 1st inner contour, (f) existing second internal region,
- (g) following count clock-wise of 2nd inner contour, (h) complete labelling first region,
- (i) following clock-wise of 2nd outer contour, (j) complete labelling second region,
- (k) following outer contour first included region, (l) complete labeling first included region,
- (m) following outer contour second included region,
- (n) complete labelling, (o) object structure tree.

템을 이용하여 filling과 labelling할 경우 각 영역에 대한 데이터를 각 프로세서에 담당시켜 프로세서마다 제안한 방법으로 각각 filling과 labelling하면 된다. 이때 전체 처리속도는 병목현상으로 데이터 길이가 가장 긴 영역이 완전히 filling과 labelling을 끝내는 시간까지이다

IV. 실험 및 고찰

제안된 방법이 순차적으로 영상에 적용되는 과정을 살펴보기 위해 사각형으로 구성된 그림 5(a)의 입력영상과 영역별로 반전되어 filling된 영상이 그림 5(b)이며 labelling 단계를 자세히 나타내고 있다. 그림 5(c)는 첫영역의 외부윤곽에 'S' 'E' 'K'를 부여한 영상이며 그림 5(d)는 내부에 첫번째 물체가 있는 경우이며 그림 5(e)는 첫영역의 내부경계를 반시계 방향으로 추적한 영상이다. 그림 5(f)는 두번째 내부영역을 만난 경우이며 그림 5(g)는 첫영역의 내부경계를 반시계 방향으로 추적한 영상이다. 그림 5(h)는 첫영역이 K=1로 labelling된 영상이며 K값을 1증가 시킨다. (d)와 (f)에서 각각 한화소 앞쪽, 한화소 아래의 지점이 내부영역의 시작지점이 되므로 기억하게 된다. 그림 5(i)는 두번째 영역에 대한 외부윤곽에 'S' 'E' 'K'를 부여한 영상이며 그림 5(j)는 두번째 영역이 labelling된 영상이다. 그림 5(k)는 그림 5(d)에서 기억하고 있던 지점으로 와서 이지점으로 부터 외부경계를 추적한 영상이며 그림 5(l)는 첫번째 내부영역이 labelling된 영상이다. 그림 5(m)는 그림 5(f)에서 기억하고 있던 지점으로 와서 이지점으로 부터 외부경계를 추적한 영상이며 그림 5(n)은 모든 영역이 labelling된 결과 영상이다. 끝으로 위의 labelling과정에서 얻은 영역상호간의 구성tree는 그림 5(o)로 나타나진다.

복잡한 영상에 대한 적용을 실험하기 위한 그림 6의 입력영상은 4개의 독립된 영역을 포함하고 있고 각각의 영역은 다음과 같은 특징을 갖는다. 영상에서 좌, 상의 물체는 영역이 중첩되어 있는 예이며 우, 상의 물체는 에지가 서로 공유되어 기존의 방법을 적용하기 까다로운 영상이다. 좌, 하의 물체는 3차원 물체의 면을 나타내는 도형이며 각면에 대한 고유의 labelling이 필요한 경우이다. 마지막 영역은 천정과 바닥 그리고 문이 보이는 실내건축물의 경우에 대한 예이다. 에지 영상이 그림 6(a)에 나타나 있고 제안된 방법을 적용하여 처리한 결과가 그림 6(b)에 표현되어 있다. 실영상에서 제안된 방법의 적용을 실험하기 위해 house 영

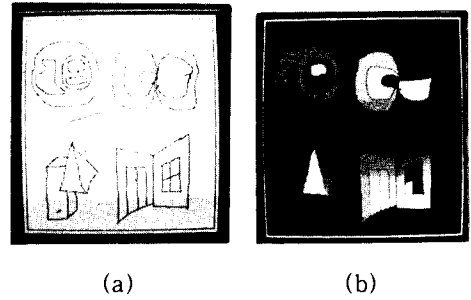


그림 6. 4개의 독립된 영역을 갖는 물체에 대한 실험
(a) 입력 에지 영상 (b) Labelling된 영상
Fig. 6. Experiment for image having 4-independent regions.
(a) Edge image, (b) Labeled image.

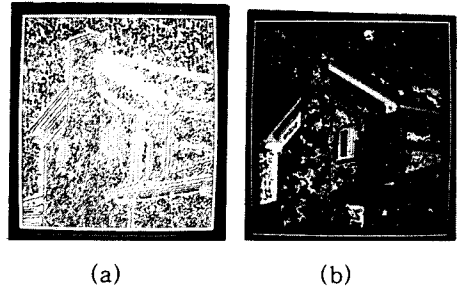


그림 7. House 영상에 대한 실험
(a) 입력 에지 영상 (b) Labelling된 영상
Fig. 7. Experiment for House image.
(a) Edge image, (b) Labeled image.

상에서 LOG 연산자를 적용하여 에지를 찾고 에지에 의해 구분되는 각영역을 labelling한 결과가 그림 7에 나타나 있다.

V. 결 론

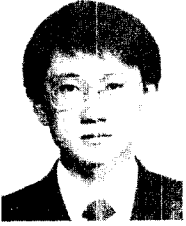
윤곽정보가 주어질때 영역을 반전하여 내부를 채우는 filling방법과 각 영역마다 고유의 번호를 정해주는 labelling방법을 제안하였다. 외부윤곽과 내부윤곽을 서로 반대방향으로 추적하여 각 주사선 마다 영역의 시작점과 끝점 정보를 추출하는 방법이다. 현재 체인 코드의 방향만을 이용하므로 체인코드를 이용한 기존의 알고리즘 보다 비교회수를 줄일 수 있으므로 계산 시간이 단축되며, 연속적인 윤곽에서도 적용이 가능하

다. Filling하면서 동시에 labelling할 수 있으므로 영상 해석시 유용하며 각각을 독립적으로 적용 시킬 수 있다. 각 분리된 영역의 데이터를 각각의 프로세서에 할당하여 filling시키므로 병렬 분산처리가 가능하며, labelling이 끝남과 동시에 구성tree를 얻음으로 영상 해석에 유용한 정보로 쓸 수 있다.

參 考 文 獻

- [1] T. Pavidis, "Algorithms for graphics and image processing," *Comput. Sci., Washington, D. C.*, 1982.
- [2] K. E. Brassel and R. Fegeas, "An algorithm for shading of regions in vector display graphics," in *Press. ACM SIGGRAPH '79*, pp. 126-133. 1979.
- [3] A. Distanto and N. Venesiani, "A two-pass filling algorithm for raster graphics," *Comput. Vision Graphics Image Process*, vol. 20, pp. 288-295, 1982.
- [4] U. Shain, "Filling regions in binary raster images," in *SIGGRAPH '80*, pp. 321-327.
- [5] B. D. Ackland and N. Weste, "The edge flag algorithm - A fill method for raster scan display," *IEEE Trans. Comput.* vol. 30, pp. 41-47, 1981.
- [6] Z. Cai, "Restoration of binary images using contour direction chain codes discription," *Comput. Vision Graphics Image Process*, 42, pp. 100-106, 1988.
- [7] Long-wen Chang and Kuen-Long Leu, "A fast algorithm for the restoration of images based on chain codes description and its applications," *Comput. Vision Grapics Image Process*, vol. 50, pp. 296-307, 1990.
- [8] S. M. Ali and R. E. Burge, "A new algorithm for extracting the interior of bounded regions based on chain coding," *Comput. Vision Graphics Image Process*, vol. 43, pp. 256-264, 1988.
- [9] 심재창, 조석제, 하영호, "양방향성 정보를 이용한 영역채움 알고리즘," 대한전자공학회 논문집, vol. 26 no. 11, pp. 238-242, 1989. 11.
- [10] I. Gargantini and H. H. Atkinson, "Linear quad trees: A blocking technique for contour filling," *Pattern Recognition* vol. 17, 289-293, 1984.
- [11] H. H. Atkinson, I. Gargantini, and T. R. S. Walsh, "Filling by Quadrants or Octants," *Comput. Vision Graphics Image Process*, vol. 33, 138-155, 1986.
- [12] G. Y. Tang and B. Lien, "Region filling with the use of the Discrete Green theorithm," *Comput. Vision Graphics Image Processing*, vol. 42, 297-305, 1988.
- [13] M. H. Sunwoo, B. S. Baroody, J. K. Aggarwal "A parallel algorithm for region labelling," *CS Workshop on Computer Architecture for Pattern Analysis and Machine Intelligence on Oct.* in Seattle, Wa., 5-7, 1987.
- [14] D. S. Hirschberg, A. K. Chandra and D. V. Sarwate "Computing Connected Components on Parallel Computers," *Communications of the ACM*, vol. 22, no. 8, 1979.
- [15] R. Lumia, L. Shapiro, and I. Zuniga, "A new connected components algorithm for virtual memory computers," *Computer Vision, Graphics and Image processing* vol. 22, 287-300, 1983.
- [16] A. K. Jain, "Fundamentals of digital image processing," Prentice-Hall International Editions, 1989.
- [17] H. Samet, "Connected component labelling using quadtrees," *Jurnal of the Association for Computing Machinery*, vol. 28, no. 3, pp. 487-501 July 1981.
- [18] M. Asal, G. Short, T. Preston, R. Simpson, Derek Roskell, and Karl Gutttag, "The Texas Instruments 34010 Graphics System Processor," *IEEE Trans CG&A*, 24-39, Oct 1986.

 著 者 紹 介



沈 載 昌(正會員)

1962年 3月 27日生. 1987年 2月 경북대학교 전자공학과 졸업. 1990年 2月 경북대학교 대학원 전자공학과 석사학위 취득. 1989年 11月~ 1990年 9月 서울대학교 반도체 공동연구소 조교. 1992年 8月 경북대학교 대학원 전자공학과 박사학위 수료. 주관심분야는 영상처리, 컴퓨터비전 등임.



玄 基 浩(正會員)

1964年 10月 19日生. 1986年 2月 경북대학교 전자공학과 졸업. 1990年 2月 경북대학교 대학원 전자공학과 석사학위취득. 1992年 2月 경북대학교 대학원 전자공학과 박사학위 수료. 주관심분야는 영상처리, 컴퓨터비전 등임.



河 錦 淑(正會員)

1962年 4月 10日生. 1983年 2月 경북대학교 전자공학과 졸업. 1990年 2月 경북대학교 대학원 전자계산기공학과 석사학위 취득. 1992年 1月~현재 구미전문대학 전산과 전임교수. 주관심분야는 데이터베이스, 멀티미디어, 영상처리 등임.

河 永 浩 (正會員) 第29卷 B編 第7號 參照
현재 경북대학교 전자공학과 교수