

論文94-31B-5-12

윤곽추적 영역채색 기법

(Connected-component Labeling using Contour Following)

沈 載 昌*, 李 駿 宰** , 河 永 浩**

(Jae Chang Shim, Jun Jae Lee and Yeong Ho Ha)

要 約

본 논문에서는 윤곽추적에 기반한 새로운 영역채색 기법을 제안한다. 제안된 알고리즘은 폐 윤곽을 따라 한바퀴 추적할 때 위로 향하는 체인코드의 수와 아래로 향하는 체인코드의 수가 같다는 원리에 근거한다. 윤곽을 추적하면서 체인코드 방향이 위로향하면 체인코드의 출발지점에 영역의 시작지점 표시를 하고, 아래로 향하면 체인코드의 도착지점에 영역의 끝점표시를 하면 한 주사선상에서 출발화소와 도착화소가 일대일 대응되므로 전영역을 탐색할 수 있다. 제안된 방법은 체인코드의 한 방향 만으로 윤곽정보를 추출 하므로 알고리즘이 매우 간단하고, 수행이 빠르며 적은 버퍼를 사용하는 효과적인 방법이다. 제안된 방법은 영역채움 및 영상복원에 바로 적용될 수 있고 영역채색과 동시에 영역특징들을 추출 할 수 있는 장점을 가진다.

Abstract

A new efficient contour following algorithm for connected-component labeling processing is proposed. The basic idea of the algorithm is that the total number of downward chain codes is the same as one of upward chain codes along the closed contour. If the chain code direction is upward, then region start mark is assigned at the chain code departure pixel and if the chain code is downward, then region end mark is assigned at the chain code arrival pixel. The proposed algorithm extracts directly the contour information from only the current direction informatin of chain. This makes the algorithm simple and fast and requires less memory with comparison to the conventional algorithms. The proposed contour following algorithm can be applied to the various kind of image processing such as region filling, restoration and region feature extraction.

I. 서 론

* 正會員, 安東大學校 컴퓨터工學科
(Dept. of Computer Eng., Andong Univ.)

** 正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)
接受日字 : 1993年 7月 21日

영역채색(labeling) 기법은 2차원 및 3차원 영상처리에서 많이 이용되는 기본적인 처리과정이므로 활발한 연구가 이루어 지고 있다.¹⁾³⁾ 영역채색이란 영상에서 분리되어진 각 영역에 고유한 부호를 부여하는

영상처리 기법으로 [4] 이는 이진 영상의 분할 [4-7], 영역의 특징 [8] 추출 및 영상의 복원 [2] 등에 이용된다.

영역채색 방법에는 주사선법(raster scan), [13-9] 시드확산법(seed filling) [2], 선길이 연결법(run length connectivity) [11], 4진목법(quadtree) [12-13], 윤곽추적법(contour following) [4, 7, 14-19] 등이 있다.

이중 체인코드 이용법 [4, 7, 19] 은 영역의 윤곽을 먼저 추적하고 그 내부를 채색하는 방법으로 윤곽의 특징 추출이 용이하므로 영상인식 등에 유용한 방법이며, 이중 심 [9] 의 방법은 외부 및 내부윤곽을 서로 반대 방향으로 추적하면서 주사선상의 영역 시작화소 및 끝화소를 부여하고 그 사이를 채우는 효과적인 방법이다. 특수하게 연결된 영상에서 채색이 잘못될 경우가 발생될 수 있다.

본 논문에서는 심의 방법을 개선한 체인코드를 기반으로 새로운 윤곽추적과 윤곽추적시 방향 정보를 활용하는 알고리즘을 제안하였다. 영역의 윤곽을 따라 한바퀴 추적할 때 위로 향하는 지점들의 높이가 합은 아래로 향하는 지점들의 높이 합과 항상 동일하다. 이 정보를 영역내부 추적을 위한 각 주사선에서 영역의 시작화소와 끝화소로 활용하며 외부윤곽은 시계 방향, 내부윤곽은 반시계 방향으로 추적하도록 하였다. 이를 기반으로 하여 4방향 연결성을 적용하는 윤곽추적 방법과 8방향 연결성을 적용하는 윤곽추적법을 제안하였다. 4방향 연결성 추적은 왼쪽화소 우선추적법(left most looking method) [2] 으로 추적하면서 윤곽정보를 이용하고, 8방향 연결성 추적은 왼쪽화소 우선추적을 적용하되 4방향 연결성을 4대각 방향 연결성보다 우선하여 추적한다.

영상을 순차주사 하다가 영역을 만나면 순차주사를 멈추고 윤곽을 추적 하면서 체인의 방향정보를 이용하여 윤곽정보를 부여한다. 이 윤곽정보를 이용하여 영역의 내부를 추적하면서 추적되어진 화소임을 표시함과 동시에 내부윤곽이 있는가를 찾는다. 윤곽정보는 윤곽추적시 체인코드가 위로 향하는 지점은 동일 주사선에서 배경과 영역이 만나는 시작점으로, 아래로 향하는 지점은 영역과 배경이 만나는 끝점으로 표시하며 영역내부 추적은 표시된 영역 시작점과 영역 끝점을 연결하면서 내부 윤곽의 유무를 확인하고 추적한다.

II. 기존의 영역채색 기법

디지털 영상에서 한 화소를 둘러싼 주변화소들 중에 하나 이상의 배경화소를 포함하면 이를 윤곽화소(contour pixel)라 정의한다. 디지털 영상에서 윤곽

을 추적하기 위해서는 연결화소를 고려하는 방법 [2] 에 따라 4방향 연결성, 4대각방향 연결성 또는 8방향 연결성을 그림 1과 같이 정의한다.

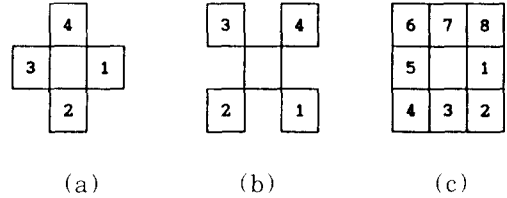


그림 1. 연결성 (a) 4방향 연결성 (b) 4 대각방향 연결성 (c) 8방향 연결성

Fig. 1. Connectivity (a) 4-connectivity, (b) 4-diagonal connectivity, (c) 8-connectivity.

영역채색은 배경과 분리되고 서로 연결된 각각의 영역에 고유한 부호를 부여하여 각 영역의 특징을 추출하기 위해 부호화된 영상을 얻는 과정이다. 4방향 연결성과 8방향 연결성은 적용에 따라 영역의 구성이 달라지게 된다. 4방향 연결성을 적용할 경우 그림 2(a) 영상은 그림 2(b)와 같이 두개의 영역으로 구성되나, 8방향 연결성을 적용하면 그림 2(c)처럼 하나의 영역으로 취급 된다. 고유 부호가 2부터 시작한 이유는 배경화소값에 '0'을 영역화소 값에 '1'을 사용하므로 이와 구분하기 위함이다.

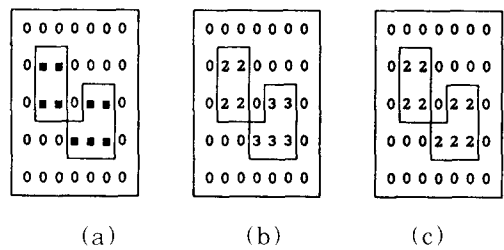


그림 2. 영상의 연결성을 이용한 영역채색 (a) 이진영상 (b) 4방향 연결성 적용할 경우 (c) 8방향 연결성 적용할 경우

Fig. 2. Region labeling using connectivity. (a) Binary image. (b) Labeled with 4-connectivity. (c) Labeled with 8-connectivity.

영역채색 방법에는 주사선법, 선길이 연결법, 4진목법, 윤곽추적법 등이 있다. 일반적으로 가장 많이

이용되는 주사선법은 영상을 2회 순차 주사하는 방법으로 첫 주사시 그림 3(a)와 같은 한 영역에 그림 3(b) 같이 서로 다른 부호인 2와 3이 부여 될때 “?” 지점에서 발생하는 두개의 부호를 버퍼에 저장한 다음 두번째 주사에서 동일 부호로 바꾸는 방법이며, 부호를 저장하기 위해 버퍼가 많이 필요하고 바꾸는 처리과정에서 버퍼를 모두 검색해야 하므로 수행시간이 오래 걸린다. 또한 영상이 커지고 동일 영역에 너무 많은 부호가 부여될 때 버퍼의 부족으로 처리가 중단될 수 있다.¹⁰⁾

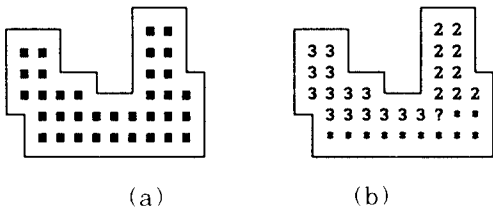


그림 3. 주사선 해석법 적용시 문제점 (a)이진영상 (b)일부가 채색된 영상

Fig. 3. Problems in raster scan method. (a) Binary image, (b) Partially labeled image.

이 방법을 개선한 버퍼를 사용하지 않는 반복수행법⁶⁾이 있으나 역시 시간이 오래 걸린다. 확산법은 영상을 순차주사 하면서 만나는 영역의 첫화소를 시작점으로 하여 윤곽을 만날 때까지 확산시키는 방법으로 확산지점을 연속하여 저장하기 위한 버퍼가 많이 필요할 뿐만 아니라 영역이 너무 클때 버퍼부족으로 채색에 실패할 경우가 발생된다. 4진목법은 4진목으로 표현된 영상으로 부터 각 트리를 검색하여 채색하는 방법으로 윤곽정보를 구하기 어려워 윤곽추적법을 추가하여 적용하여야 하는 단점이 있다. 선 길이 연결법은 이전 주사선의 부호와 영역이 나누어지는 지점 및 합해지는 지점의 정보를 표에 저장하고 이를 참조하여 현재 주사선에 부호를 부여하는 방법으로 해석표가 필요하고 처리과정이 복잡하다. 체인코드를 이용한 Morrin⁵⁾의 방법은 외부윤곽을 추적하고 한 겹씩 순차적으로 내부를 채색하는 방법으로 처리시간이 오래 걸리며, Kruse⁶⁾와 Danielsson⁷⁾의 방법은 윤곽을 추적하면서 영역의 끝점정보를 표에 저장하고 해석하여 채색하는 방법으로 표를 연속적으로 참조 해야하며, 표를 저장하기 위한 버퍼가 필요하다. 심⁹⁾의 방법은 표를 필요로 하지 않은 효율적인 방법이나 특수하게 연결된 영상에서 채색이 잘못될

경우가 발생할 수 있다. 그림 4(a)의 영상에 심의 방법을 적용할 때 그림 4(b)는 외부윤곽을 추적하여 윤곽정보를 부여한 영상이며 그림 4(c)는 내부윤곽을 추적하여 윤곽정보를 부여한 영상이다. 이때 ‘*’ 표시된지점의 왼쪽화소가 내부윤곽 추적시 ‘E’가 추적점 ‘T’로 바뀌에 따라 영역끝점이 없어지게 되고 ‘S’로 부터 ‘E’를 만날때 까지 채색하므로 ‘...’로 표시된 지점들이 계속하여 잘못 채색된다.

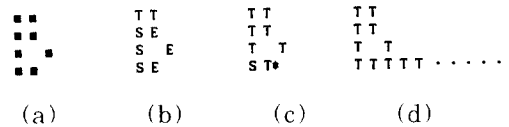


그림 4. 심의 방법에서 영역채색에 오류가 발생하는 경우 (a) 입력영상 (b) 외부윤곽이 추적된 영상 (c) 내부윤곽이 추적된 영상 (d) 채색에 오류가 발생한 영상

Fig. 4. Labeling error is occurred by Shim's method. (a) Input image. (b) outer contour following. (c) inner contour following. and (d) incorrect labeled image.

III. 새로운 윤곽추적 채색기법

영역영상이나 에지영상이 주어질 때 윤곽추적법을 응용하므로 인식에 필요한 정보를 추출하거나, 영상의 저장 또는 전송을 위해 영상을 압축하고 쉽게 복원 할 수 있다. 윤곽추적 방법에는 Bug의 추적법¹¹⁾과 체인코드 추적법²⁾ 등이 있으며 Bug추적법은 영역의 시작점으로 부터 전진하는 방향의 화소가 영역 화소이면 왼쪽 화소로 추적점을 이동하고, 배경 화소이면 오른쪽 화소로 추적점을 옮겨가는 방법이다. 매우 간단하여 윤곽정보만을 추출할 때 자주 이용되나 완전한 윤곽추적을 위해서는 역방향 추적을 실시하여야 하므로 영상이 복잡할 경우 체인코드 추적법보다 시간이 더 걸린다. 체인코드 추적법은 윤곽정보 추출과 영역의 면적, 갯수, 구멍수, 외부윤곽 및 내부윤곽의 주변길이 등의 일반적인 특징추출을 병행할 수 있어 영상인식에 유용하게 이용될 수 있다. 기존의 체인코드²⁾를 이용한 추적법은 내부구멍을 찾기 위해서 반드시 연속되는 두 체인코드를 이용하여 검색해야 하는 번거로움이 있으며, 외부 및 내부윤곽이 동일 지점으로 구성된 경우 등에는 해석이 복잡하다.

그러나 체인코드의 방향 정보를 이용하면 연속되는 두 체인코드 대신 하나의 체인코드 만으로도 처리가 가능하다.

본 논문에서는 체인코드의 방향정보를 이용하여 한번의 순차추사 및 하나의 체인코드만을 사용하면서 어떠한 경우라도 매우쉽고 빠르게 추적하면서 영역을 채색하는 새로운 윤곽추적 영상처리 기법을 제안한다. 제안된 알고리즘의 핵심되는 아이디어는 윤곽을 한바퀴 돌때 위로 향하는 체인코드의 수와 아래로 향하는 체인코드의 수가 같다는 Green의 정리^[21, 22]에 근거한다. 윤곽을 추적하면서 체인코드가 위로 향하는 화소를 각 주사선에서 영역의 시작점으로, 체인코드가 아래로 향할때 체인코드의 도착지점을 영역의 끝 지점으로 표시하면 영역 시작과 영역 끝 지점의 수가 각 주사선상에서 일대일로 대응되므로 이정보를 이용하는 방법이다. 윤곽내부에 또 다른 윤곽을 포함하고 있을 때는 외부윤곽은 시계 방향으로 내부 윤곽은 반시계 방향으로 추적하면 된다. 영상에서 영역의 연결성을 4방향과 8방향에 따라 두가지 윤곽추적법을 제안하여 영역채색을 실시하였다.

1. 4방향 연결성 추적채색기법

4방향 추적 체인코드는 그림 5와 같이 중심으로 부터 4방향 연결성을 고려한다. 그림에서의 숫자 1, 2, 3, 4는 각각 시계방향으로 동 남 서 북의 방향을 나타낸다. 4방향 윤곽추적은 왼쪽우선 윤곽추적법과 동일하나 체인코드의 한 방향만으로 영역의 시작과 끝을 구분하는 방법이다. 윤곽정보는 체인코드의 방향에 따라 영역시작점, 영역끝점 및 추적점으로 분류된다. 윤곽정보 중 체인코드가 아래로 향하는 방향의 화소는 영역 끝점이 되고 위로 향하는 방향의 화소는 영역 시작점이 된다.

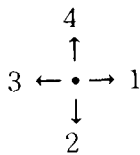


그림 5. 4방향 체인코드
Fig. 5. 4-directional chain code.

4방향 체인코드에서 각 체인코드의 진행방향에 대한 검색 우선 순위는 왼쪽화소를 우선하여 시계방향으로 순차적으로 검색한다. 그림 6(a)에서 서쪽화소에서 현재화소인 중심화소로 체인코드가 진입 했을때 다음 진행할 화소의 위치를 검색하기 위해서는 북쪽

①, 동쪽 ②, 남쪽 ③의 순으로 살펴보고, 이때 앞의 세방향 화소가 영역화소가 아닐 경우, 서쪽 ④로 되돌아가는 순서로 검색한다. ○ 안의 숫자는 검색 우선순위를 의미하며 검색 할때 ①의 방향에 영역화소가 있으면 ①의 방향으로 진행하여 추적하고 그렇지 않으면 ②의 방향 및 ③의 방향 순으로 추적한다. 동일한 방법으로 그림 6(b)는 북에서 중심화소로, (c)는 동에서 중심화소로, (d)는 남에서 중심화소로, 진입한 체인코드에 대한 각각의 추적 우선 순위를 나타내고 있다

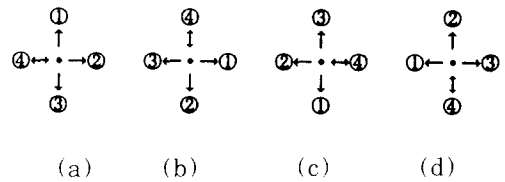


그림 6. 4방향추적시 검색 우선순위 화소진행 방향이 (a) 서에서 중심으로 향하는 경우 (b) 북에서 중심으로 향하는 경우 (c) 동에서 중심으로 향하는 경우 (d) 남에서 중심으로 향하는경우 이때 중심은 현재화소를 나타냄

Fig. 6. Searching priority of 4-connectivity boundary following. Chain code direction is (a) west to center, (b) north to center, (c) east to center, and (d) south to center. Center is current pixel.

체인코드의 출발화소는 그림 7(a)에서 화살표가 시작되는 지점이며, 도착화소는 화살표가 도착하는 지점이다. (b)와 같이 수직으로 연결된 경우 체인코드가 적용될 때 아래로 향하는 체인코드의 도착화소 ①은 위로 향하는 체인코드의 도착화소 ②와 서로 한 주사선 엇갈려 있으므로 이를 동일한 주사선에 일치시키려면 아래로 향할 경우 체인코드 도착화소 지점을, 위로 향할 경우 체인코드 출발지점 화소를 고려해야 한다.

윤곽정보 부여 방법을 자세히 살펴보면 체인코드의 방향이 아래로 향하는 그림 7(c)의 경우 체인코드 도착지점에 영역의 끝점(end point) 표시인 'E'를 부여하고, 체인코드의 방향이 위로 향하면 그림 7(d)와 같이 체인코드 출발지점에 영역의 시작점(start point) 표시인 'S'를 부여한다. 'S'나 'E'를 부여할 때 부여될 화소에 이미 'S'나 'E'가 부여되어 있는

경우는 그림 7(e)와 같은 꼭지점이거나 그림 7(f)와 같은 세로 직선 등의 경우이므로 이미 추적된 화소임을 나타내는 표시인 'T'(trace)를 부여한다. 마찬가지로 체인코드 출발지점에 'S'나 'E'가 부여되지 않는 좌, 우로 향하는 그림 7(g)와 같은 수평성분도 'T'를 부여한다.

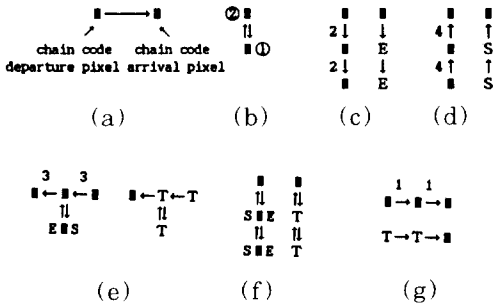


그림 7. 체인코드 방향 및 윤곽정보 부여지점 (a) 체인코드 출발화소 및 도착화소 (b) 도착 및 출발화소가 주사선상 불일치 (c) 체인코드가 아래로 향할 경우 (d) 체인코드가 위로 향할 경우 (e) 꼭지점의 경우 (f) 한화소 두께의 경우 및 (g) 수평축의 경우

Fig. 7. The pixel positions assigning contour informations according to the chain code direction. (a) Chain code departure and arrival pixel. (b) Arrival pixel and departure pixel are not on the same scan line. Pixel position assigned with (c) 'E' at chain code arrival pixel, (d) 'S' at chain code departure pixel, (e) 'T' at vertex point, (f) 'T' at vertical line of one pixel width, and (g) 'T' at chain code departure pixel in the horizontal direction.

4방향 연결성을 따라 윤곽을 추적하며 체인의 방향이 위로 향할 경우와 아래로 향할 경우 영역시작 및 끝점을 부여하면서 윤곽을 추적하고 각 주사선별로 영역시작점에서 끝점사이를 채색하는 방법은 알고리즘 1과 같다.

4방향 영역채색 알고리즘: 4방향 연결성을 고려한 영역채색 알고리즘
 영역값 = 1, 배경값 = 0, T = 2
 영상을 순차주사

단계 1 : 영역을 만나면

1. 시계 방향으로 왼쪽우선 윤곽추적법으로 추적
 - 1) 체인코드 방향이 아래로 향하면 체인코드 도착 화소에 'E' 부여 (화소값이 'S'나 'E'이면 'T')
 - 2) 체인코드 방향이 위로 향하면 체인코드 출발 화소에 'S' 부여 (화소값이 'S'나 'E'이면 'T')
 - 3) 각 체인 출발 화소에 아무값도 부여되지 않으면 'T'

2. 추적 시작점으로 되돌아오면 시작점 바로 아래 화소 연결 확인

- 1) 윤곽 시작지점화소 다음줄 화소값이 추적되지 않은 영역이면 단계 1-1을 계속
- 2) 그외의 경우 단계 2 수행

단계 2 : 각 줄마다 'S'에서 'E'사이를 'T'로 채움

- 1) 'S'에서 'E'사이에 'U'인 값을 만나면 채움을 일시중단
- 2) 윗 주사선 동일 위치부터 반시계 방향으로 추적 ('S', 'E', 'T' 부여는 단계 1과 동일)
- 3) 시작점으로 되돌아오면 바로 이전화소로 부터 단계 2를 계속

단계 3 : 한 영역의 추적이 끝나면 T를 1 증가

단계 4 : 모든 영역이 채색될때 까지 1-3 반복 수행

윤곽추적 알고리즘은 영상을 왼쪽에서 오른쪽으로, 위에서 아래로 순차주사 하면서 만나는 영역을 순서대로 한 영역씩 처리한다. 단계 1은 각 영역의 시작지점을 순차주사 하면서 찾고 외부윤곽을 추적하면서 윤곽정보를 부여하는 과정이며, 1단계 2는 추적 시작점으로 되돌아 왔을때 아래방향으로 추적되지 않은 영역연결이 있는 경우 이를 추적하는 과정이다. 이와 같은 영역구조는 시작지점의 오른쪽 화소가 한 화소 두께로 가로방향으로 이어지고, 시작지점의 바로 밑에 화소가 있고 왼쪽으로 영역이 이어지는 그림 8과 같은 경우이다.

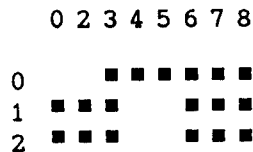


그림 8. 영역 시작지점 바로 밑에 영역연결이 있는 경우

Fig. 8. In case of the region connected below region start point.

두번째 단계는 윤곽추적에서 부여된 윤곽정보를 이용하여 내부영역을 채색하고 이를 마친 지점을 표시한 후 내부구멍이 있는지 확인하는 과정이다. 각 주사선별로 영역시작점에서 끝점사이를 채색 표시인 'T' (trace)로 채우면서 내부구멍이 있는가를 확인한다. 'T'를 채울 지점화소값이 '0'인 경우 내부구멍이 존재하므로 내부영역채색을 잠시 중단하고 바로 이전 주사선 동일 위치로부터 반시계 방향으로 영역을 추적한다. 내부윤곽 시작지점으로 되돌아오면 중단된 내부영역 채색 중단지점 한 화소 오른쪽 지점부터 내부영역 채색을 계속한다. 내부구멍 둘레에 왼쪽은 영역끝점 정보가 오른쪽은 영역시작점 정보가 부여되므로 구멍을 제외하고 채색이 이루어지며 내부구멍이 여러개 있어도 그때마다 위와 동일한 방법으로 내부구멍을 추적하므로 구멍은 영역 채색에서 모두 제외될 뿐만 아니라 내부구멍의 수를 쉽게 구할 수 있다. 한 영역에 대한 윤곽추적과 내부채색이 끝나면 중단된 순차주사 지점에서 부터 다음 영역을 검색하는데 이미 채색된 영역은 모두 'T'로 바뀌어 있으므로 새로운 영역과 쉽게 구별된다. 두번째 단계를 적용할 때 'S'를 찾아야 하는데 'S'를 찾는 방법에 따라 다음과 같이 세가지 방법이 가능하다.

첫째: 윤곽추적시 물체의 외접최소 사각형(MBR)을 찾아두고 그 사이를 순차주사 하여 'S'에서 'E'사이를 추적하는 방법

둘째: 윤곽추적시 'S'가 부여되는 화소의 x, y좌표를 버퍼에 저장하였다가 이지점에서 부터 'E'를 만날 때 까지 추적하는 방법

이 두방법은 한 영역의 윤곽을 추적한 후 내부를 채색하고 다음영역으로 옮겨가므로 중복순차되는 영역이 발생되어 시간이 걸린다. 반면 영상크기와 동일한 버퍼를 이용하여 중복주사를 피할 수 있는 최적의 방법은

셋째: 영상의 크기와 동일한 버퍼를 준비하고 윤곽추적시 'S'가 부여되는 화소에 해당하는 새 버퍼에 채색 고유번호를 부여한다. 그리고 순차주사시 'S'를 만나면 채색 고유번호는 저장하고 새버퍼의 값으로 'E'를 만날때 까지 채운다음 채색 고유번호로 다시 바꾸어 새로운 영역을 검색하는 방법이다.

이 세방법을 알고리즘으로 기술하면

윤곽내부채색 및 내부윤곽확인법 1:

외부윤곽추적시 : 영역의 최소외접 사각형을 찾고

내부추적 : 최소 외접사각형 내를 순차주사 한다.

1. 'S'를 만나면 'E'를 만날때 까지 고유번호

'T'를 부여한다.

- 1) 만약 부여될 지점값이 '0'이면 바로 윗주사선 동일지점 화소부터 반시계방향으로 윤곽을 추적한다. (윤곽정보 부여는 외부 윤곽추적시와 동일함)
- 2) 시작점으로 되돌아 오면 바로 이전화소에서 부터 계속하여 내부추적을 수행한다.

윤곽내부채색 및 내부윤곽확인법 2:

외부윤곽추적시 : 'S'가 부여되는 지점의 x, y 좌표값을 버퍼에 저장한다.

내부추적 : 버퍼의 좌표값을 읽고 해당하는 화소의 값을 읽어

1. 만약 'S'이면 'E'를 만날때 까지 고유번호 'T'를 부여한다.
 - 1) 만약 부여될 지점값이 '0'이면 바로 윗주사선 동일지점 화소부터 반시계방향으로 윤곽을 추적한다. (윤곽정보 부여는 외부 윤곽추적시와 동일함)
 - 2) 시작점으로 되돌아 오면 바로 이전화소에서 부터 계속하여 내부추적을 수행한다.
2. 'S'가 아닌값이면 해당되는 좌표값은 무시함 (한 화소두개의 세로방향으로 이어지는 경우나 두영역이 한 점을 중심으로 만나는 경우등)

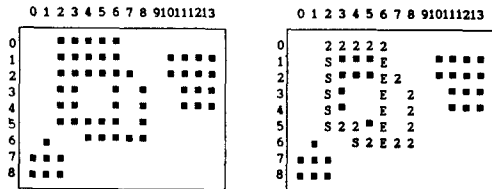
윤곽내부채색 및 내부윤곽확인법 3:

1. 외부윤곽 추적시 'S'가 부여되는 지점에 해당하는 새버퍼에 고유번호 'T'를 저장
2. 만약 화소값이 'S'이면 고유번호는 저장하고 새 버퍼의 고유번호 'T'를 읽는다.
3. 'E'를 만날때 까지 'T'로 채움.
 - 1) 만약 부여될 지점값이 '0'이면 바로 윗주사선 동일지점 화소부터 반시계방향으로 윤곽을 추적한다. (윤곽정보 부여는 외부 윤곽추적시와 동일함)
 - 2) 시작점으로 되돌아 오면 바로 이전화소에서 부터 계속하여 채움을 수행한다.
4. 'E'를 만나면 원래의 고유번호를 가지고 검색

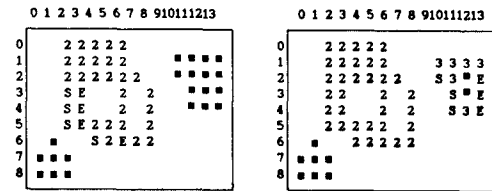
세가지 방법들은 각각 다음과 같은 장단점을 갖는다. 먼저 버퍼 사용량을 고려할 때 첫째 방법은 추가적인 메모리가 전혀 필요하지 않는 반면 셋째 방법은

표 1. 주사 횟수 및 요구되는 버퍼량
 Tabel 1. Scan numbers and required buffers.

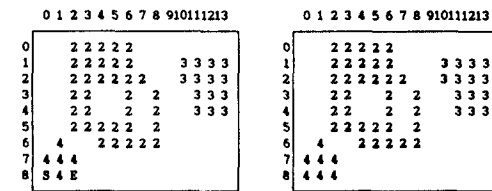
Algorithm	Scan Number	Required buffer
Algorithm1	$N \times N \cdot \text{region MBR} \cdot \text{contour number}$	0
Algorithm2	$N \times N \cdot \text{region number}$	Number of 'S's
Algorithm3	$N \times N$	$N \times N$



(a) (b)



(c) (d)



(e) (f)

그림 9. 영역채색 알고리즘 1의 적용예 (a) 입력영상 (b) 첫 외부윤곽의 추적 (c) 첫 내부윤곽의 추적 (d) 둘째 외부윤곽의 추적 (e) 셋째 외부윤곽의 추적 (f) 영역채색이 완료된 영상

Fig. 9. Example to show region labeling algorithm 1. Part (a) shows input image (b) contour following along (b) 1st outer contour, (c) 1st inner contour, (d) 2nd outer contour, (e) 3rd outer contour, and (f) labeled image.

원영상과 동일한 크기의 버퍼가 반드시 필요하다. 둘째 방법은 영상의 복잡도에 따라 필요한 버퍼의 크기도 가변되나 시작점의 좌표를 저장하기 위한 약간의 버퍼가 필요하다. 다음은 중복하여 주사되는 화소를 고려해보면 셋째 방법은 윤곽화소만 두번에 걸쳐 주사하는 반면 두번째 방법은 영역화소는 두번 주사하게 된다. 첫째 방법은 외접사각형 내부를 주사하게 되므로 중복하여 주사하게 되는 수가 가장 많다. 이 세가지 방법의 수행시간을 실험에서 비교하였다. 표 1은 세가지 방법의 영상전체에서 필요한 주사횟수 및 요구되는 버퍼수를 비교하였다.

그림 9(a)영상에 4방향 연결성을 고려할 경우 세개의 영역으로 분리되고 첫 영역은 하나의 구멍을 갖는 구조의 영상이다. 영역채색 과정을 순차적으로 살펴보면 그림 9(a)는 원 영상, (b)는 외부윤곽을 추적하여 'S', 'E' 및 'T'를 부여한 영상이며 그림 9(c)는 내부윤곽을 추적한 결과이고, 그림 9(d)와 (e)는 각각 두번째 및 세번째 물체의 윤곽을 추적한 결과이며, 그림 9(f)는 영역채색을 끝마친 영상이다.

2. 8방향 연결성 윤곽추적 채색기법(4방향 연결성 우선 8방향 추적)

8방향 왼쪽화소 우선추적법의 추적방향은 그림 10과 같고 체인코드가 다음 진행할 방향을 검색할 때 왼쪽부터 시계방향으로 순차적으로 검색한다.

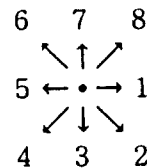


그림 10. 8방향 체인코드의 방향

Fig. 10. 8-directional chain code.

제한된 4방향 연결성 우선 8방향 추적법은 왼쪽화소 우선추적법과 유사하게 왼쪽화소부터 검색하지만 검색순서를 이우한 두방향 중 4방향 연결성을 4대각방향 연결성보다 우선하여 추적하는 방법이다. 체인의 방향에 대한 추적 우선 순위는 그림 11과 같다. 먼저 그림 11(a)와 같이 체인코드가 서쪽 화소에서 현재화소인 중심화소로 진입하여 다음으로 추적하여 나갈 경우 기존의 8방향 왼쪽화소 우선추적법은 북쪽화소 ①, 북동쪽화소 ③, 동쪽화소 ②, 남동쪽화소 ⑤, 남쪽화소 ④의 순으로 검색한다. 만약 이들중 이어질 영역 화소가 없으면 체인코드의 진행방향의 반대쪽인 서쪽화소 ⑥을 검색한다.

그러나 제안된 방법은 4대각방향 연결보다 4방향연결이 추적 우선순위가 높아 검색순서는 먼저 북쪽 ①을 다음은 동쪽 ②, 동북쪽 ③, 남쪽 ④, 남동쪽 ⑤, 그리고 서쪽 ⑥을 순차적으로 검색한다. 여기서 남서쪽 및 북서쪽 두 방향은 추적에서 제외되므로 각 방향마다 최대로 6 방향만을 검색하면 된다. 마찬가지로 그림 11(b)는 체인코드가 남동쪽으로 향할 경우, 그림 11(c)는 남쪽으로 향할 경우, 그림 11(d)는 서남쪽, 그림 11(e)는 서쪽, 그림 11(f)는 서북쪽, 그림 11(g)는 북쪽, 그림 11(h)는 북동쪽으로 향할 경우의 검색순서를 나타낸다.

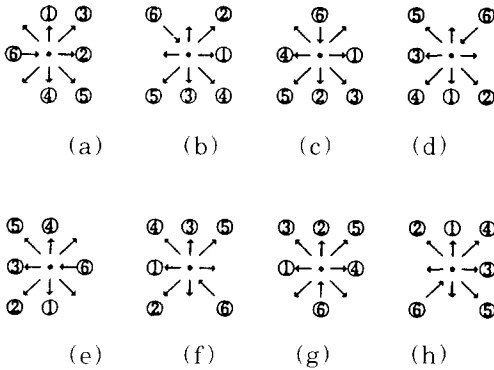


그림 11. 제안된 4방향 연결성 우선 8방향 윤곽추적법의 검색 우선순위, 체인코드의 진입 방향이 (a) 서에서 중심으로 (b) 북서에서 중심으로 (c) 북에서 중심으로 (d) 북동에서 중심으로 (e) 동에서 중심으로 (f) 남동에서 중심으로 (g) 남에서 중심으로 (h) 남서에서 중심으로 진입할 경우, 중심은 현재화소의 위치를 나타냄

Fig. 11. The directional priority in the proposed boundary following in case that input chain code direction is (a) west to center, (b) northwest to center, (c) north to center, (d) northeast to center, (e) east to center, (f) southeast to center, (g) north to center, and (h) northwest to center, respectively. Center is the current pixel position.

추적하는 방법은 영상을 왼쪽에서 오른쪽으로, 위에서 아래로 순차 주사하면서 영역을 만날때까지 검색을 실시한다. 영역을 만나면 시계방향으로 추적하고 체인코드는 저장한다. 체인코드 방향 정보를 이용

하여 각 줄마다의 영역의 시작 지점과 영역 끝지점을 표시하고 이 사이에 내부윤곽의 유무를 확인한다.

방향정보를 이용한 윤곽정보는 주사선상에서 영역의 시작화소는 위로 향하는 체인코드인 6, 7, 8로 그림 12(a)와 같이 출발화소에 'S'를 부여하고, 영역 끝화소는 아래로 향하는 체인코드인 2, 3, 4로 그림 12(b)과 같이 도착화소에 'E'를 부여한다.

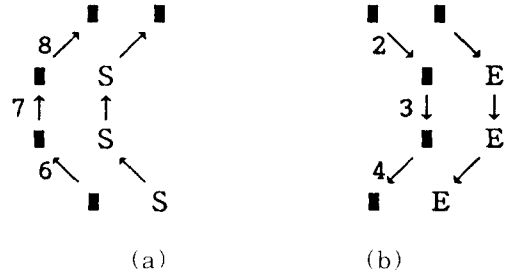


그림 12. 8방향 연결성 추적의 윤곽정보 'S' 및 'E' 부여에 (a) 위로 향할 경우 (b) 아래로 향할 경우

Fig. 12. The example assigned with contour information in 8 connectivity. Pixel position to assign 'S' and 'E' in case of chain code is (a) up direction, (b) down direction.

윤곽을 한바퀴 추적하고 출발지점으로 되돌아 오면 그림 13과 같이 왼쪽 아래 대각방향(225°방향 대각방향) 부분에 영역의 존재 유무를 확인한다. 그림 13에서 [0,4] 지점에서 시작하여 시계방향으로 오른쪽 영역을 추적하고 출발지점으로 되돌아 왔을때 왼쪽아래 영역이 추적되지 않은 상태이므로 추적해야 한다.

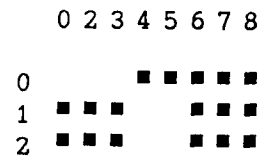


그림 13. 왼쪽아래 (225°방향) 대각연결
Fig. 13. The diagonal connection in lower-left corner(225° direction).

4대각방향 연결성 우선 추적 채색기법을 정리하면 알고리즘 2와 같다.

8방향 영역채색 알고리즘: 8방향 연결성을 고려한 영역채색 추적시 4방향 우선 8방향으로 추적하면서

단계 2 에서 아래방향대신에 왼쪽아래 대각연결 영역 확인하고 나머지는 알고리즘 1과 같음

그림 9(a)의 영상에 8방향 연결성을 고려할 경우 2개의 영역으로 나누어지고 첫 영역은 두개의 구멍을 가진다. 영역채색 과정을 순차적으로 나타내면 그림 14(a)는 첫 영역의 외부윤곽을 추적한 영상이고 그림 14(b) 및 (c)는 첫내부윤곽 및 둘째 내부윤곽을 각각 추적한 결과이다. 그림 14(d)는 두번째 영역의 외부 윤곽을 추적한 영상이며 그림 14(e)는 영역채색이 완료된 영상이다. 제안된 8방향 영역채색 방법은 그림 4와 같은 영상에 대해서도 오류없이 영역채색이 가능하다.

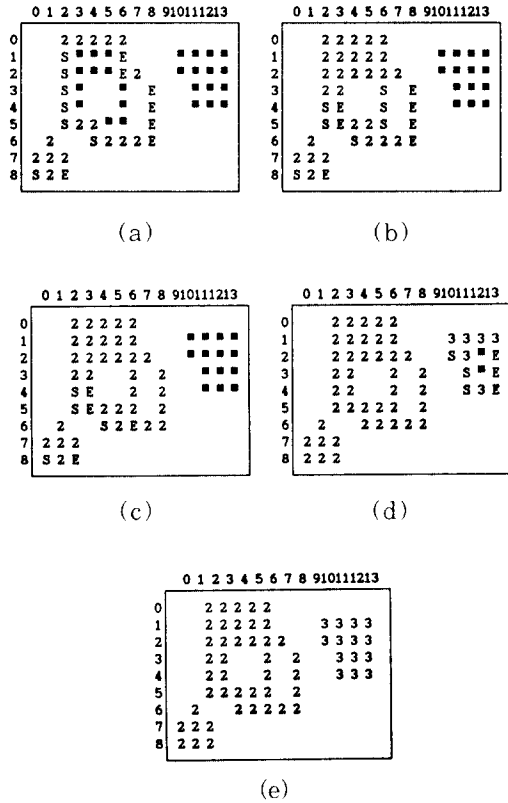


그림 14. 영역채색 알고리즘 2의 적용에 (a) 첫 외부윤곽의 추적 (b) 첫 내부윤곽의 추적 (c) 둘째 내부윤곽의 추적 (d) 둘째 외부윤곽의 추적 (e) 영역채색이 완료된 영상

Fig. 14. Example to show region labeling algorithm 2. Part (a) shows contour following along 1st outer contour, (b) 1st inner contour, (c) 2nd inner contour, (d) 2nd outer contour, and (e) labeled image.

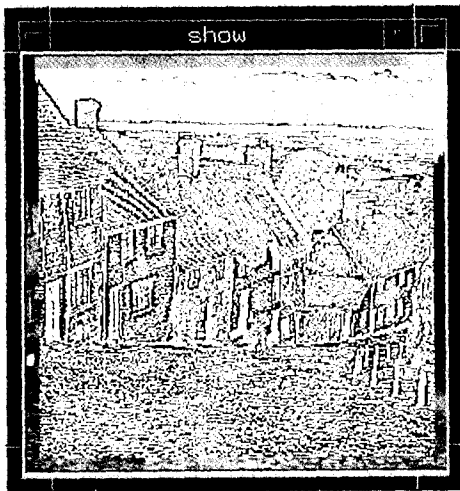
V. 실험 및 고찰

영역채색법의 수행시간을 비교하기 위해 256x256 크기의 영상을 선택하였고 이로 부터 블롭영상과 에지영상을 각각 2개씩 추출하였다. 블롭영상은 영역으로 구성된 영상이며, 에지영상은 가는 경계로 구성된 영상으로 제안된 방법을 다양한 영상에 적용해 보기 위해 선택하였다.

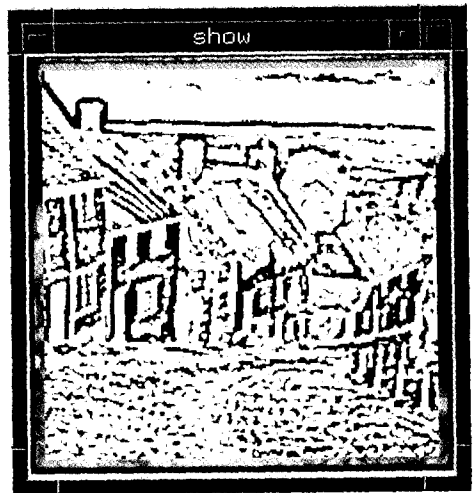
그림 15(a)는 Ghill영상이며 이로 부터 블롭영상을 구한 과정은 먼저 Burt의 계층적 이산 상관법으로 영상을 몽롱화 시키는데, 이는 DOG와 비슷한 특성의 영상을 추출하게 된다. 그 다음은 2차와 3차 몽롱화된 영상의 차 영상을 구하고 절대값을 취한 후 임계치가 0.9 이상인 화소를 블롭으로 선정하였다. 계층적 이산 상관법에 의한 몽롱화는 DOG와 특성이 거의 비슷한 반면 수행시간이 빨라서 실 시간 구현에 적합한 방식이므로 이를 선택하였다. 1과 2차의 차로 구한 블롭영상은 그림 15(b)와 같이 많은 수의 작은 블롭으로 구성된 영상이고 2와 3차 몽롱화 영상의 차로 부터 구하면 그림 15(c)의 굵은 블롭 영상이 된다. 영상의 영역수는 그림 15(b)와 15(c)가 각각 1464개 및 504개이다. 에지영상을 얻기 위해 이용한 영상처리법은 Canny에지 추출법이며 IBM AIX Image Assistant/6000 시스템에서 제공하는 표준 라이브러리를 이용하였다. 구해진 에지영상이 그림 15(d)와 같고 17388개의 에지를 가지며 매우 복잡한 영상이 얻어진다. 이 에지영상으로 부터 연결성이 약한 에지를 제외한 7916개의 에지를 선택하면 배경부분의 에지가 거의 없어진 평탄한 그림 15(e)영상이 얻어진다. 그림 15(f)에서 (i)는 영역채색 알고리즘 1을 이용하여 영역채색한 그림 15(b)에서 (e)의 영상이다. 채색된 영상에서 동일밝기의 명암도는 8 연결성으로 연결된 영상이며 오류없이 고유부호를 부여하고 있다. 실험에서 비교한 영상은 8개이며 블롭 1과 블롭 2는 세밀 및 듬성한 블롭영상을 각각 나타내고, 에지 1과 에지 2는 세밀 및 듬성한 에지영상을 각각 나타낸다. Iblob1 및 Iblob2는 블롭1과 블롭2를 각각 반전한 영상이며 Iedge1 및 Iblob2는 각각 에지1 및 에지2를 반전한 영상이다. 비교에 사용된 방법은 영역채색 기법에서 제안된 일반적으로 사용되는 주사선법 및 이의 연속적 수행으로 버퍼를 필요로 하지 않는 반복수행법과 확산법 및 제안된 방법의 3가지 윤곽추적을 이용하는 방법이며 채색하는데 걸리는 CPU 시간은 표 2와 같다. 실험에 사용된 시스템은 IBM-PC 486/33이며 MSC7.0언어로 프로그램 하였다.



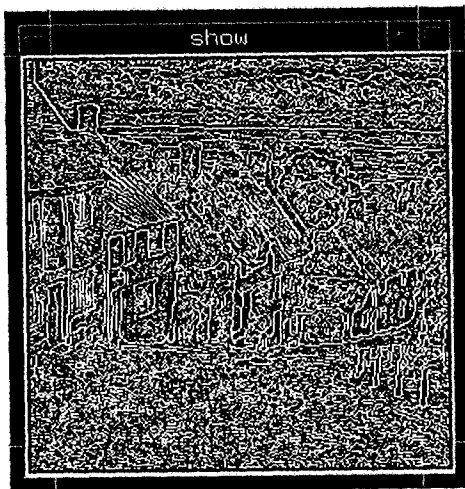
(a)



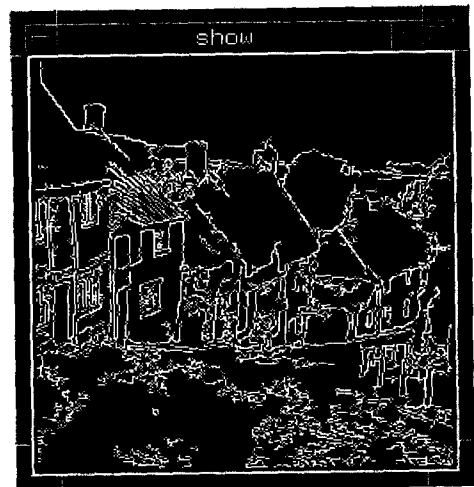
(b)



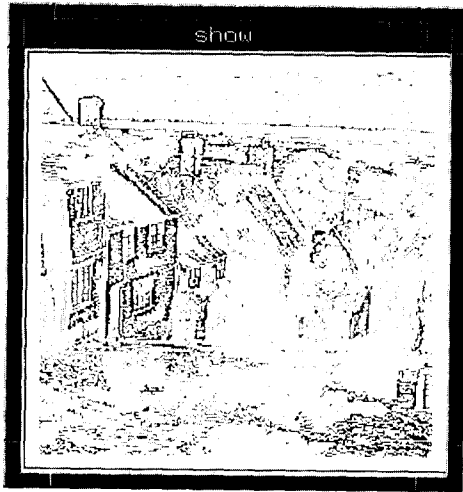
(c)



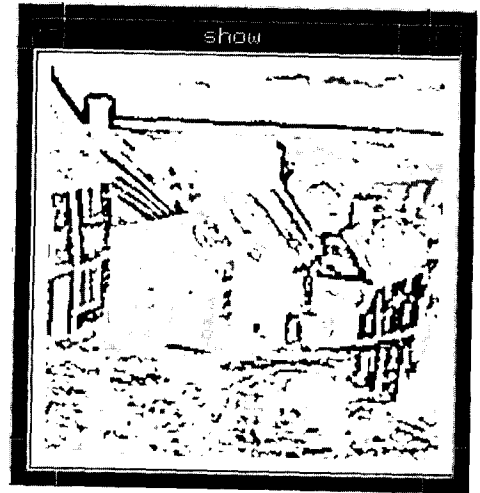
(d)



(e)



(f)



(g)



(h)



(i)

그림 15. Ghill 영상 (a) 원영상 (b) 세밀한 블럽영상 (c) 굵은 블럽영상 (d) 세밀한 에지영상 (e) 연결성이 강한 에지영상 (f)에서 (i)는 (b)에서 (e)영상을 영역채색한 결과

Fig. 15. Ghill image. Part (a) shows original image, (b) fine blob image, (c) coarse blob image, (d) fine edge image, (e) edge image with strong connection, (f) through (i) are labeled image of (b) through (e), respectively.

표 2에서 주사선법은 반전된 4개의 영상에서 배경에 해당하는 아주 큰 영역으로 인해 채색시간이 오래 걸리며, 확산법도 큰영역이 많을 경우 채색하는데 시간이 많이 소요된다. 제안된 첫방법은 영역의 최소외접사각형 내부를 중복하여 주사하므로 인해 제안된 둘째 방법보다 약간의 시간이 더 필요하다. 제안된

세번째 방법은 반전된 영상에서와 같은 큰영역이 많이 존재할 때 시간이 가장적게 걸리는 방법이다. 결론적으로 주사선법은 동일 영역에 부여된 서로 다른 부호를 동일 부호로 부여하기 위한 검색과정의 구현이 매우 어렵고, 1024x1024이상의 큰 영상에는 10,000여개 이상의 서로 다른 영역부호가 발생할 수 있

표 2. 영역채색 방법의 속도비교

Table 2. CPU time comparison in connected-component labeling.

(unit : sec)

Image	Raster scan Labeling	Iterative labeling	seed labeling	Proposed labeling Algorithm1	Proposed labeling Algorithm2	Proposed labeling Algorithm3
blob1	7.31	32.03	2.47	2.36	1.97	2.42
blob2	7.63	23.45	1.32	1.32	1.10	1.70
Edge1	4.39	19.28	2.20	2.14	1.81	2.14
Edge2	1.96	11.87	1.92	2.31	1.54	1.87
fblob1	34.33	94.30	4.34	3.79	3.57	3.62
fblob2	16.14	29.38	5.33	2.75	2.91	2.30
fdge1	79.86	107.21	4.50	3.29	3.19	3.02
fdge2	30.21	94.36	4.56	2.53	2.56	2.15

어 버퍼를 많이 필요로 하며 때로는 검색이 실패할 경우도 있다. 반복수행법은 버퍼가 필요없는 반면 실험에 사용된 영상의 경우 약 10회 정도를 반복하여야 채색이 완료되므로 시간이 오래 걸린다. 확산법은 영상이 커짐에 따라 버퍼가 대단히 많이 요구되며 버퍼량을 초과하여 채색에 실패할 경우가 발생할 경우가 발생될 수 있으며 주변길이 등의 특징추출을 위해서는 추가적인 윤곽추적이 필요한 반면, 제안된 첫째 방법의 영역채색 알고리즘은 버퍼가 필요없고, 둘째 방법은 약간의 버퍼가 요구되나 수행시간이 가장 빠르다. 셋째 방법은 각 주사선별 병렬처리에 적합한 처리방법이다.

VI. 결론

본 논문에서는 새로운 윤곽추적법을 제안하여 영역 채색에 적용하는 알고리즘을 제안하였다. 체인코드 기반의 새로운 윤곽추적법은 윤곽을 따라 한바퀴 추적할 때 위로 향하는 윤곽의 높이 합은 아래로 향하는 윤곽의 높이 합과 항상 동일하다는 원리를 근거로 외부윤곽은 시계 방향, 내부윤곽은 반시계 방향으로 추적하면서 체인의 방향에 따라 윤곽정보를 추출하고 이를 활용하는 방법이며 4방향 연결성을 적용하는 윤곽추적법과 8방향 연결성을 적용하는 두 가지의 윤곽추적법을 제안하였다. 제안된 방법은 처리속도가 빠르고, 메모리가 적게 필요하며 구현이 용이하다. 또한 영역의 중요한 특징들인 영역수, 구멍수, 주변길이, 모멘트, 면적 및 구성구조 등을 채색과 동시에 구할 수 있다. 제안된 윤곽추적법은 심의 방법에서 채색에 오류가 발생하는 경우를 보완할 뿐만 아니라 윤곽의 내부를 추적하는 3가지 방법을 추가하여 연구하였다. 그리고 지금까지 발표된 체인코드를 이용하는 여러가지 다른 방법과 비교해 볼 때 체인코드의 한 방향 정보만으로 윤곽정보를 추출하므로 두 개의

보조 테이블이 필요 없고, 추가적인 버퍼가 적게 필요한 장점을 가진다.

參考文獻

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. Second Edition, Academic Press, 1982.

[2] T. Pavlidis, *Algorithm for Graphics and Image Processing*. Comput. Sci., Washington, D.C., 1982.

[3] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Addison-Wesley publishing company, 1992.

[4] R. L. T. Cederberg "Chain-link coding and segmentation for raster scan devices," *Comput. Vision Graphics Image Process.*, 10, pp. 224-234, 1979.

[5] T. H. Morrin, "Chain-link compression of arbitrary black-white images," *Computer Graphics and Image Processing*, vol. 5, no. 2, pp. 225-273, 1976.

[6] B. Kruse, "A fast algorithm for segmentation of connected components in binary image," *Proceedings of the first Scandinavian conference on Image Analysis*, 1980.

[7] P. Danielsson, "An improved segmentation coding algorithm for binary and nonbinary images," *IBM J. Res. Develop.*, vol. 26, no. 6, pp. 698-707, 1982.

[8] J. L. C. Sanz and D. Petkovic, "Machine vision algorithms for automated inspection of thin-film disk heads," *IEEE Pattern Analysis and Machine Intelligence*, vol. 14, no. 6, pp. 830-848, 1988.

[9] R. Lumia, L. Shapiro, and O. Zuniga, "A new connected components algorithm for virtual memory computers," *Comput. Vision Graphics Image Process.*, 22, pp. 287-300, 1983.

[10] I. Dinstein, D. W. L. Yen, and M. D. Flickner, "Handling memory overflow in connected component labeling

- applications," *IEEE Pattern Analysis and Machine Intelligence*, vol. Pattern Analysis and Machine Intelligence-7, no. 1, January 1985.
- [11] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall International Inc., 1989.
- [12] H. Samet, "Connected component labeling using quadtrees," *Journal of the Association for Computing Machinery*, vol. 28, no. 3, pp. 487-501, July 1981.
- [13] H. Samet, and M. Tamminen, "Component labeling of images of arbitrary dimension represented by linear bintrees," *IEEE Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, July 1988.
- [14] M. B. Dillencourt, H. Samet, and M. Tamminen, "A general approach to connected-component labeling for arbitrary image representations," *Journal of the Association for Computing Machinery*, vol. 39, no. 2, pp. 253-280, April 1992.
- [15] J. Hecquard, and R. Acharya, "Connected Component labelling with linear octree," *Pattern Recognition*, vol. 24, no. 6, pp. 515-531, 1991.
- [16] G. Grant and A. F. Reid, "An efficient algorithm for boundary tracking and feature extraction," *Comput. Vision Graphics and Image Process.*, vol. 17, pp. 225-237, 1981.
- [17] A. K. Agrawala, and A. V. Kulkarni, "A sequential approach to the extraction of shape features," *Comput. Vision Graphics Image Process.*, 6, pp. 538-557, 1977.
- [18] 심재창, 조석제, 하영호, "이진 경계코드를 이용한 새로운 영역채움 알고리즘," 대한전자공학회 논문지, 제26권 11호, 238-242쪽, 1989
- [19] 심재창, 현기호, 하영호, "체인코드의 방향정보를 이용한 Filling과 Labeling," 대한전자공학회 논문지, 제29권 B편 11호, 50-58쪽, 1992
- [20] M. Kunt, et. al, "Second generation image coding techniques," *Proc. IEEE*, vol. 73, pp. 549-574, 1985.
- [21] M. R. Spiegel, *Theory and Problems of Vector Analysis*, McGraw-Hill Inter., 1959.
- [22] G. Y. Tang, "A discrete version of green's theorem," *IEEE vol. Pattern Analysis and Machine Intelligence-4*, no. 3, May 1982.
- [23] J. D. Cappelletti and A. Rosenfeld, "Three-dimensional boundary following," *Comput. Vision Graphics Image Process.*, 48, 80-92, 1989.
- [24] R. Lumia, "A new three-dimensional connected components algorithm," *Comput. Vision Graphics Image Process.*, 23, pp. 207-217, 1983.

 著者紹介

沈 載 昌(正會員) 第 31卷 B編 第 4號 參照

현재 안동대학교 컴퓨터공학과 전임
강사

李 駿 宰(正會員) 第 31卷 B編 第 4號 參照

현재 경북대학교 전자공학과 박사
과정 수료

河 永 浩(正會員) 第 31卷 B編 第 4號 參照

현재 경북대학교 전자공학과 부교수