

그레이 레벨 연결성 복원 하드웨어 구조

*김성훈, **양영일
경상대학교 전자재료공학과
경남 진주시 가좌동 900

e-mail : *shkim@cecc-1.gsnu.ac.kr , **yyang@nongae.gsnu.ac.kr

A Hardware Architecture for Retaining the Connectivity in Gray - Scale Image

Sung-Hoon Kim , Yeong-Yil Yang

Dept. of Electronic Materials Eng. GyeongSang Nat'l. Univ.
Gazwa-Dong 900, Chinju, KyungNam, 660-701, Korea

Abstract

In this paper, we have proposed the hardware architecture which implements the algorithm for retaining the connectivity which prevents disconnecting in the gray-scale image thinning. To perform the image thinning in a real time which find a skeleton in image, it is necessary to examine the connectivity of the skeleton in a real time. The proposed architecture finds the connectivity number in the 4-clock period. The architecture is consists of three blocks, *PS(Parallel to Serial) Converter* and *State Generator* and *Ridge Checker*. The *PS Converter* changes the 3×3 gray level image to four sets of image pixels. The *State Generator* examine the connectivity of the central pixel by searching the data from the *PS Converter*. the 3×3 gray level image determines. The *Ridge Checker* determines whether the central pixel is on the skeleton or not. The proposed architecture finds the connectivity of the central pixel in a 3×3 gray level image in the 4-clocks. The total circuits are verified by the design tools and operate correctly.

I. 서론

영상의 세선화(image thinning)는 영상내의 물체를 그의 골격선으로 변환하는 것으로서, 영상 처리에 의해 이루어지는 물체의 형태 분석과 패턴 인식에 있어서 매우 중요한 전처리 단계에 속한다. 물체의 골격선은 그 물체에 대한 구조적인 정보나 형태학적인 정보를 보유하고 있으므로 골격선을 이용하면 물체에서 원하는 특징을 용이하게 추출할 수가 있게 되어, 문자 인식, 문서 영상 처리, 구조 분석 등과 같은 분야에 사용된다. 영상의 세선화에는 많은 방법들이 제안되어 있다. *Dyer*^[1]는 이치 영상에 대하여 세선화를 일반화한 알고리즘을 제안하였고, *Peleg*^[2]는 local min/max 연산을 근방 연산으로 사용하여 골격선을 추출하였으며, *Salari*^[3]는 CGDT (Contextual Gray Distance Transformation)와 ridge 추적을 이용하여 세선화하는 알고리즘을 제안했다. *Wang*^[4]은 *Hilditch*^[5]의 이치 세선화 알고리즘을 개선한 그레이 레벨 알고리즘을 제안하였는데, 이는 근방의 화소값으로부터 연결수를 구한 후, 연결수 정보로 사용하여 골격선을 만들어 가는 세선화 기법이다. 이와 같은 영상의 세선화 알고리즘들은 영상에서 불림연산(dilation)과 녹임연산(erosion)과 같은 국부적 연산을 반복적으로 수행하여 영상을 세선화한다. 국부적인 연산을 수행하면 골격선이 분리되므로, 분리된 골격선을 연결시키기 위해 화소의 연결성을 확인한 후 복구해야 한다. 영상의 세선화는 골격선이 추출될 때까지 국부적 연산과정과 화소의 연결성 복구 과정을 반복 수행한다. 연결성을 복원하지 않으면 물체의 구조적인 정보를 잃어버리게 되므로, 물체의 형태를 분석하거나 패턴을 인식하는 것이 어렵게 된다.

세선화 알고리즘에서 사용되는 연산자인 불림연산(dilation)과 녹임연산(erosion)을 수행하는 하드웨어 구조에

대해서는 연구가 많이 이루어졌으나^[6,9,10] 화소의 연결성을 검사하는 하드웨어의 구조는 연구되지 않았다. 연결성 확인에서의 시간 지연은 전체적인 영상처리의 속도를 저하시키므로, 영상 세선화를 실시간으로 처리하기 위해서는 연결성을 검사하는 하드웨어가 필요하다. 국부적인 세선화를 수행한 후 연결성을 확인하는 방법으로는 *Wang*^[4]이 제안한 방법이 널리 사용되고 있다. 본 논문에서는 *Wang*이 제안한 알고리즘을 구현하는 하드웨어 구조를 제안하였다. 제안된 구조는 3×3 그레이 레벨에 대하여 가운데 화소의 연결성을 4-클럭에 검사한다. 본 논문에서 제안하는 구조는 *PS(Parallel to Serial) Converter* 블록, *State Generator* 블록과 *Ridge Checker* 블록으로 구성되어 있다. *PS Converter* 블록에서는 3×3 그레이 레벨 영상을 4개의 직렬영상으로 만들어 *State Generator* 블록으로 보낸다. *State Generator* 블록에서는 3×3 그레이 레벨의 가운데 화소가 골격선에 접하는지를 검사하고, *Ridge Checker* 블록에서는 가운데 화소가 골격선상에 있는지를 판단한다.

본 논문의 II장에서는 연결성 복원 알고리즘에 대해 기술하고, III장에서는 본 논문에서 제안된 4-클럭에 연결성을 검사하는 하드웨어 구조에 대해 설명한다. 그리고, IV장에서 결론을 보인다.

II. 연결수 복원 알고리즘

이 장에서는 영상의 세선화에서 연결성에 대하여 살펴보고, 용어들을 정의한 후 *Wang*이 제안한 연결성 복원 알고리즘^[4]에 대해 기술한다.

1. 영상의 세선화와 연결성

그림 1(a)는 손으로 쓴 "Y" 문자의 그레이 레벨을 3차원으로 표현한 것이다. 이 영상에 대하여 세선화 과정을 거쳐 골격선만 추출된 골격선 영상이 그림 1(b)에 나타나 있다. 이와 같이 추출된 골격선 영상은 물체의 원하는 특징을 용이하게 추출할 수가 있고, 문자 인식, 문서 영상 처리, 구조 분석과 같은 분야에 응용된다. 세선화를 통해서 추출된 골격선은 연결성과 세선도(thinness)가 좋아야 하며, 골격선의 위치가 영상의 가운데에 위치해야 한다.

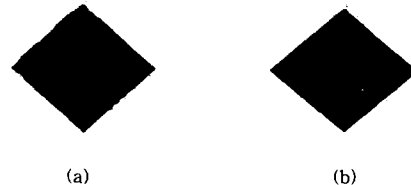


그림 1. (a) 3차원으로 표현된 "Y"자의 그레이 레벨 영상 (b) 세선화 과정에 의해 추출된 골격선 영상

Fig. 1. (a) The 3-D plot of the gray level image of the letter "Y" and (b) the skeleton image extracted by image thinning

그림 2(a)는 그레이 영상을 보여주는 데 진한 사각형으로 표시된 화소들은 물체가 존재하는 부분을 나타낸다. 그림 2(b)는 그림 2(a)의 영상에 연결성을 고려하지 않고 불림연산, 녹임연산과 local min/max와 같은 국부적인 연산만으로 세션화하였을 때 얻어진 골격선 영상을 보여준다. 그림 2(b)에서 진한 사각형으로 표시된 화소들은 추출된 골격선이고 점선으로 표시된 화소는 세션화 과정에서 골격선이 끊어진 화소들이다. 영상의 전체적인 구조 정보를 고려하지 않으면, 그림 2(b)와 같이 골격선이 끊어진 결과를 얻게된다. 이와 같이 연결성이 끊어지면 전체적인 영상의 구조 정보에 큰 영향을 주므로, 세션화 과정에서 끊어진 골격선을 복구해야 한다.

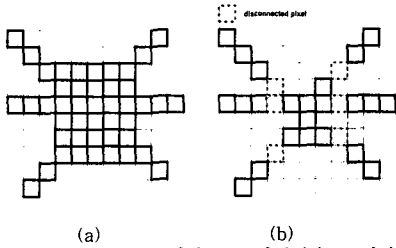


그림 2. (a) 그레이 레벨 영상 (b) 연결성을 고려하지 않고 세션화를 수행하였을 때 얻어진 골격선

Fig. 2. (a) The gray level image (b) The extracted skeleton by the image thinning without considering the connectivity. The shaded rectangles represent the disconnected pixels

그림 3(a)는 3×3 그레이 레벨 영상에서 화소를 나타내며, 각 화소를 N_i 로 표현한다. N_1, N_2, \dots, N_8 화소를 화소 N_0 에 대한 8-근방(8-neighbors)으로 정의한다. 이 때, 화소 N_i 의 화소값을 x_i 로 정의하며 ($i = 0, 1, \dots, 8$), 그림 3(b)는 각 화소에 대한 화소값을 보여준다. 적어도 1개 이상의 -1, 0 또는 1 요소들이 연속된 나열을 스트링이라 한다

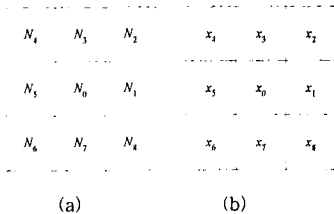


그림 3. (a) 3×3 그레이 레벨의 영상에서 화소 정의 (b) x_i 는 화소 N_i 의 그레이 레벨을 나타낸다.

Fig. 3. (a) The definition of the pixels in the gray level 3 × 3 pixels (b) x_i is the gray level of the pixel N_i , where $i = 0, 1, \dots, 8$.

2. 연결성 복원 알고리즘

Wang^[4]은 세션화 과정에서 골격선의 끊어지는 것을 복구하는 알고리즘을 제안하였다. 3×3 영상의 8-근방을 조사하여 식(1)과 식(2)의 정의에 의해 f 를 구한다. 화소 N_0 를 중심으로 N_1, N_2, \dots, N_8 순서대로 반시계 방향으로 일주하며 화소 N_0 의 화소에서 판단하였을 때, x_i 가 상승하는 면에 있으면 f_i 는 1이 되고, x_i 가 하강하는 면에 있으면 f_i 는 -1이고, 그 외의 경우는 0을 갖게 된다. 그림 4(a)는 식(1)과 식(2)에 의해 구해진 f_i 의 예를 보여준다. 화소 N_0 의 8-근방에 대해 반시계 방향으로 일주하며 식(1)과 식(2)에 의해 구해진 f_i 를 검사하여 화소 N_0 가 골격선과 만나는 수를 다음과 같이 구한다. 식(1)과 식(2)에 의해 구해진 f_i 를 순서대로 나열하여 스트링 $F1=f_1f_2 \dots f_8$ (그림 4(a)에서 스트링 $F1=-1,0,1,0, -1,0,1,0$ 이다.)을 만든다. 스트링 $F1$ 에서 0인 요소가 제거된 스트링을 $F2$ 라 하고, $F2$ 의 첫 번째 요소를 $F2$ 의 마지막 요소에 첨가하여 만들어진 스트링을 $F3$ 을 한다. 그림 4(a)의 예에서 $F2=f_1f_2f_3f_4f_5f_6f_7f_8=-1,1,-1,1$ 이 되고 $F3=f_1f_2f_3f_4f_5f_6f_7f_8f_1=-1,1,-1,1,-1,1$ 이 된다. 스트링 $F3$ 에서 f_i 가 -1에서 1로 증가한다는 것은 이 부분이

전체적인 영상에서 화소값이 낮은 영역에서 화소값이 큰 영역으로 증가하는 부분임을 의미하고, 연속된 두 요소가 1과 -1이면 화소값이 높은 영역에서 낮은 영역으로 감소하는 부분임을 의미한다. 스트링 $F2$ 에서 요소가 -1에서 1로 바뀌는 수 또는 1에서 -1로 바뀌는 수는 화소 N_0 의 연결수라 정의한다. 화소 N_0 대하여 계산된 연결수가 만일 2보다 크거나 같게 되면, 화소 N_0 의 화소값은 국부적인 연산을 거치기 전의 값으로 복원해야 한다.

0	1	0	0	1	1
-1	N_0	-1	1	N_0	0
0	1	0	0	-1	0

(a) (b)

그림 4. 각 화소에 표현된 수는 주어진 화소에 대하여 구해진 f_i 값을 나타내고, 이 예에서 (a)와 (b)에서 화소 N_0 의 연결수는 2이다.

Fig. 4. The number represents the f_i value of the pixels and the connectivity number of the pixel N_0 is 2 in the examples of (a) and (b).

$$f_{2l-1} = \begin{cases} -1, & \text{if } x_{2l-1} < x_0 \text{ and } (x_{2l-1} < x_{2l} \text{ or } x_{2l-1} < x_{2l+1}) \\ 1, & \text{if } x_{2l-1} > x_{2l+1} \text{ and } x_{2l+1} < x_0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$f_{2l} = \begin{cases} 1, & \text{if } x_{2l} > x_{2l+1} \text{ and } x_{2l+1} < x_0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

($l = 1, 2, 3, 4$)

여기에서 $x_9 \equiv x_1$ 이다.

그림 5에서 ○로 표시된 화소들은 골격선상의 화소를 나타내고 선은 화소들을 연결하는 골격선을 나타낸다. 그림 5(a)와 같이 골격선이 형성되어 있을 경우에 위에서 기술한 방법을 사용하여 구하여진 연결수가 그림 5(b)에 나타나 있다. 연결수가 1인 화소는 그 점이 골격선의 끝점에 있다는 것을 의미하고, 연결수가 2인 화소는 골격선상의 화소임을 의미한다. 연결수가 3인 화소는 골격선의 분기점에 있는 화소임을 나타내고, 연결수가 4인 화소는 골격선이 교차되는 지점의 화소임을 나타낸다. 그러므로 연결수가 2이상인 화소들은 모두 골격선상의 화소들일 가능성이 있다. 불림연산과 녹임연산 등의 국부적 연산을 수행한 후 골격선에서 제거될 화소중에서 연결성이 2보다 크거나 같으면 국부적 연산을 수행하기 전의 값으로 복원해야 한다.

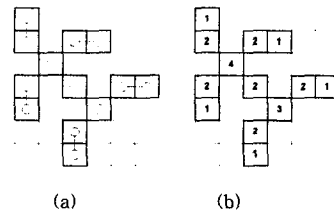


그림 5. (a) ○로 표시된 화소들은 골격선상의 화소들을 나타내고 선은 골격선을 나타낸다. (b) 골격선상의 화소에 대해 구해진 연결수를 보여준다.

Fig. 5. (a) The pixels denoted by ○ are the pixel on the skeleton and the line represents the skeleton of the image. (b) The number represents the connectivity number of the corresponding pixels.

III. 제안한 연결성 검사 하드웨어 구조

이 장에서는 본 논문에서 제안하는 하드웨어 구조를 설명한다. 제안한 구조는 3×3 그레이 레벨 영상으로부터 가운데 화소의 연결성을 4-클럭에 걸쳐 검사하는 하드웨어 구조를 설명한다. 그림 6은 3×3 그레이 레벨 영상의 연결성을 구하는 하드웨어의 블록 다이어그램이다. 입력 화소값들을 내부에 필요한 순서대로 화소값들을 변환하는 PS(Parallel to Serial) Converter 블록과 식(1)과 식(2)에 따라 f_i 의 값을 계산하여 골격선을 찾는 상태생성기(State Generator) 블록과 가운데 화소의 복원을 결정하는 Ridge Checker 블록으로 구성된다. 가운데 화소가 골격선상에 있어 제거하면 골격선이 끊어질 가능성이 있으면 Retention은 세트(Set)된다. 3.1절에서는 PS(Parallel to Serial) Converter 블록에 대하여 설명하고, 3.2절에서는 상태생성기(State Generator) 블록에 대하여 설명하고, 3.3절에서는 Ridge Checker 블록에 대하여 설명한다.



그림 6. 제안된 연결수 검사 하드웨어 구조의 블록 다이어그램
Fig. 6. The block diagram of the proposed architecture which checks the connectivity number

1. PS(Parallel to Serial) Converter 블록

그림 7은 식(1)과 식(2)를 계산하는데 필요한 화소들을 표현하였다. 식(1)과 식(2)에서 l 이 1인 f_1 과 f_2 를 계산하기 위해서는 사용되는 (x_0, x_1, x_2, x_3) 를 루프1의 화소들로 정의한다. 식(1)과 식(2)에서 l 이 2인 경우 f_3 과 f_4 의 계산에는 (x_0, x_3, x_4, x_5) 가 사용되므로 이 화소들은 루프2의 화소들로 정의한다. 또한 l 이 3인 경우인, f_5 과 f_6 의 계산에는 (x_0, x_5, x_6, x_7) 이, $l=4$ 인 경우인 f_7 과 f_8 의 계산에는 (x_0, x_7, x_8, x_1) 이 사용되고, 이 화소들은 각각 루프3과 루프4의 화소들로 정의한다.

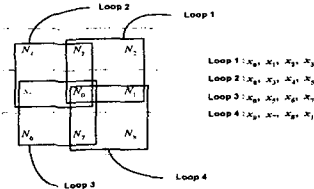


그림 7. 식(1)과 식(2)에서 f_{2l} 과 f_{2l-1} 을 계산하는데 필요한 x_i 값들의 집합

Fig. 7. The set of x_i values needed to compute the f_{2l} and f_{2l-1} in the equations (1) and (2)

그림 8은 PS(Parallel to Serial) Converter 블록의 내부 회로도도를 나타낸다. 그림 8에서 입력 영상의 그레이 레벨을 갖고 있는 R_0 는 4-클럭마다 갱신된다. 클럭의 주기는 t_1, t_2, t_3, t_4 이고, MUX1과 MUX2는 매 클럭마다 순차적으로 입력을 선택한다. 즉, MUX1은 4-클럭마다 $R_{21}, R_{12}, R_{23}, R_{32}$ 의 값인 x_1, x_3, x_5, x_7 을 출력하고 MUX2는 $R_{11}, R_{13}, R_{33}, R_{31}$ 의 값인 x_2, x_4, x_6, x_8 을 매 클럭마다 출력한다. MUX3는 t_1, t_2 ,와 t_3 주기에는 MUX1의 출력을 선택하고, t_4 클럭 주기에서는 D 값을 선택한다. D 는 첫 번째 클럭에만 동작하므로 언제나 x_1 값을 유지한다. 처음으로 클럭이 인가된 후 t_1 주기에는 SR_1 과 SR_2 는 MUX1과 MUX2의 출력인 x_1 과 x_2 를 갖게 된다. 그리고 MUX3은 t_1 주기에 MUX1의 출력을 선택하므로 x_3 이 출력된다. 그러므로 t_1 주기에서는 A, B, C 는 각각 x_1, x_2, x_3 를 출력하게 되므로, PS Converter 블록은 그림 7에서 루프1의 값을 출력한다. 다음 클럭 기간인 t_2 주기에서는 SR_1 과 SR_2 는 x_3 과 x_4 를 출력하고, MUX3은 MUX1의 출력인 x_5 를 선택한다. 그러므로 t_2 주기에서 PS Converter 블록은 그림 7의 루프2에 속하는 화소의 값(x_0, x_3, x_4, x_5)을 출력한다. t_3 주기에는 SR_1 과 SR_2 는 x_5 과 x_6 값을 갖게 되고, MUX3은 x_7 을 선택한다. 그러므로 그림 7의 루프3에 속하는 화소의 값

이 출력된다. 마지막 클럭인 t_4 주기에는 A 와 B 는 x_7 과 x_8 을 출력하고, MUX3은 D 를 선택함으로써 x_1 을 출력한다. 그러므로 t_4 주기동안에는 PS Converter 블록은 그림 7에서 4번째 루프에 속하는 화소의 값인 (x_0, x_7, x_8, x_1) 을 출력한다.

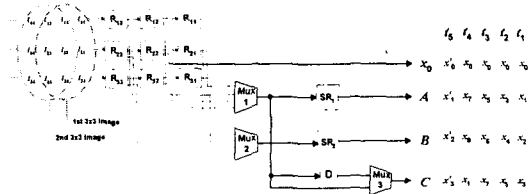


그림 8. PS(Parallel to Serial) Converter 블록
Fig. 8. The circuit diagram PS(Parallel to Serial) Converter

2. State Generator 블록

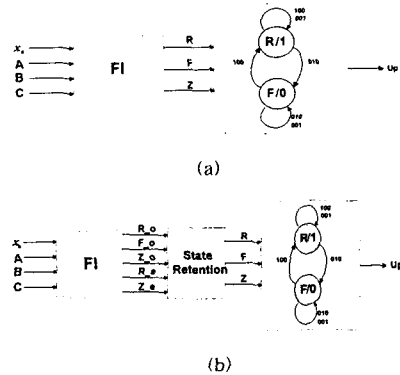


그림 9. 상태 생성기 블록
Fig. 9. The block diagram of the State Generator

그림 9는 그림 6의 State Generator의 내부 회로도이고, 회로도에서 FI 블록은 PS(Parallel to Serial) Converter 블록의 출력인 (x_0, A, B, C) 를 사용하여 식(1)과 식(2)의 계산을 하여 f_i 를 구한다. 그림 9(a)는 FI 블록에서 계산되는 f_i 값이 $f_1, f_2, f_3, \dots, f_8$ 까지 순차적으로 8-클럭동안 출력되는 경우에 연결수를 구하는 회로의 블록 다이어그램이므로, 그림 9(b)는 f_{2l} 과 f_{2l-1} (f_1 과 f_2, f_3 과 f_4, f_5 와 f_6, f_7 과 f_8)이 동시에 계산되어 출력되는 경우에 연결수를 구하는 블록 다이어그램이다. 우선 그림 9(a)의 경우에 대하여 먼저 설명하고, 그림 9(b)의 회로의 동작을 살펴본다. 그림 9(a)에서 FI는 매 클럭마다 f_1, f_2, f_3, \dots 의 계산결과가 순차적으로 출력된다. 그림 9(a)에서 FI의 계산 결과가 1이면 신호 R이 세트되고, -1이면 신호 F가 세트되고, 0이면 신호 Z가 세트된다. 그림 9(a)의 상태 생성기의 상태도의 동작은 다음과 같다. 그림 9(a)의 상태도는 값이 적어지는 상태를 나타내는 F(Falling) state와 값이 증가되는 상태를 나타내는 R(Rising) state로 이루어진 두 개의 상태를 갖고, 초기 상태로는 F state를 갖는다. 상태도에서 입력은 각각 FI의 출력인 RFZ를 의미한다. RFZ가 001인 경우, 즉 계산된 f_i 가 0인 경우에는, 클럭이 인가되었을 때 이전의 상태를 계속 유지한다. R state에서 RFZ가 100인 경우는, 현재 값이 증가되는 상태에 있고, 입력도 계속해서 증가됨을 의미하므로, 다음 상태는 R state를 유지한다. 또한 F state에서 RFZ가 010인 경우는, 현재 값이 감소되는 상태에 있고, 입력 또한 계속해서 감소됨을 의미하므로 다음 상태는 F state를 유지한다. 상태도의 천이는 다음과 같은 두 경우에만 발생한다. R state에서 RFZ가 010인 경우, 현재는 값이 증가하는 상태에 있고 계산된 f_i 값이 감소함을 의미하므로, 다음 상태는 F state로 바뀐다. 그리고 F state에서 RFZ가 100인 경우, 다음 상태는 R state가 된다. 상태 생성기의 출력인 up 신호

는 F state에서는 0을 출력하고, R state에서는 1을 출력하므로, 0에서 1로 변화한 수는 연결수가 된다.

그림 9(b)는 4-클럭을 주기로 연결도를 구하는 상태생성기 (State Generator) 구조를 보여준다. FI에서 식(1)과 식(2)에서 f_{2i-1} 과 f_{2i} 이 동시에 계산이 이루어지므로, f_{2i-1} 의 출력이 -1, 0, 1인 경우 각각 세트되는 신호인 R_o , F_o , Z_o 와 f_{2i} 의 출력이 0 또는 1인 경우에 세트되는 R_e 와 Z_e 신호가 출력된다. 그림 9(b)는 그림 9(a)의 FI와 상태도는 그대로 사용하고 R_o , F_o , Z_o , R_e , Z_e 신호를 R , F , Z 로 변환하는 State Reduction 회로가 첨가되어 있다.

4-클럭에 연결수를 구하기 위해서는 f_{2i-1} 과 f_{2i} 를 동시에 고려하여 상태를 결정하고 출력하여야 한다. 그림 10(a)는 FI 블록에서 동시에 계산되는 f_i 의 값들의 가능한 조합들을 보여준다. $f_{2i-1}(f_1, f_3, f_5, f_7)$ 은 1과 0값 외에 -1을 출력하지만 $f_{2i}(f_2, f_4, f_6, f_8)$ 은 1과 0의 값을 출력하고 -1을 출력하지 않는다. State Reduction 회로는 그림 10(b)의 각각의 경우에 대하여 다음과 같은 기능을 한다. Case1인 경우 상태를 유지해야하므로 신호 Z 를 세트하여 상태도가 현재상태를 유지하게 한다. Case2, Case3과 Case4인 경우 모두 f_{2i-1} 이 상승함을 나타내므로, 신호 R 를 세트하여 상태도에 f_{2i-1} 과 f_{2i} 를 고려하였을 때 상승함을 알려준다. Case5인 경우 f_{2i-1} 과 f_{2i} 를 고려하였을 때 값이 감소하므로 신호 F 를 세트하여 f_{2i-1} 이 감소함을 알려준다. Case6의 경우는 f_{2i-1} 이 1이므로 신호 R 를 세트한다. f_i 의 값이 -1에서 1로 변화하므로 상태도의 현재 상태와 관계없이 연결수가 증가하는 경우이다. 이와 같은 경우가 발생하면 상태도와 관계없이 up 신호를 발생하여 Ridge Checker에게 연결수가 증가됨을 알려준다. 이와 같이 f_{2i-1} , f_{2i} 에 의해 연결수가 증가할 때는 상태도에서는 up 신호를 발생하는 경우는 발생하지 않는다. 계산된 f_{2i-1} 과 f_{2i} 이 -1, 1인 경우 상태도가 F state에 있으면 다음 상태는 R state가 되고 연결수는 1이 증가하면 된다. 상태도가 R state이면 f_{2i-1} 과 f_{2i} 에 의해 연결수가 1 증가되어야 하고 다음 상태는 R state를 갖는 경우이다. 이 경우 State Reduction 회로로 신호 R 를 세트하므로 상태도는 R state를 유지하게 된다.

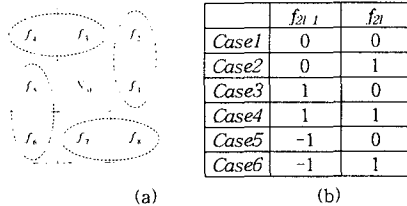


그림 10. (a) FI 블록에서 동시에 계산되는 f_i 의 값들 (b) f_{2i-1} 과 f_{2i} 값들이 갖는 가능한 경우

Fig 10. (a) f_i values computed concurrently in the block FI (b) The truth table shows the possible combinations of f_{2i-1} 과 f_{2i}

3. Ridge Checker 블록

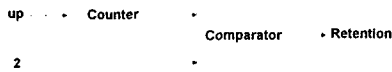


그림 11. Ridge Checker 블록
Fig 11. The block diagram of Ridge Checker

그림 11은 Ridge Checker 블록을 나타낸다. Ridge Checker 블록은 State Generator 블록의 출력인 up 신호가 세트(set)될 때, 1씩 증가하는 카운터와 카운터의 출력이 2보다 크거나 같은가를 비교하는 비교기로 구성된다. up 신호가 세트된다는 것은 연결수의 증가를 나타내며 그림 5에서 연결수가 2이상인 화소는 골격선상의 화소이므로, 4-클럭이 지난 후 카운터의 값이 2보다 크거나 같으면 그림 11의 출력인 Retention을 세트하여 이 화소는 이전의 화소값으로 복원해야 되는 화소임을 알려준다..

제한된 하드웨어 구조는 Max PlusII 설계 툴을 사용하여 동작 특성과 시뮬레이션을 수행하였다. 256 그레이 레벨 영상이 입력으로 주어질 때, 정상적으로 동작하였다.

IV. 결론

본 논문에서는 그레이 레벨 영상의 세션화 과정에서 발생하는 골격선의 끊어짐을 방지하는 연결성 검사 알고리즘을 구현하는 하드웨어 구조를 제안하였다. 영상에서 물체의 골격선을 찾아내는 세션화 과정을 할 때, 물체의 구조 정보를 잃어버리지 않기 위해 화소의 연결성을 조사해야 한다. 제안된 연결성 검사 하드웨어 구조는 기존의 불림연산과 녹임연산의 국부적 연산을 수행하는 하드웨어와 더불어 영상의 세션화를 실시간으로 구현하는 것을 가능하게 하였다.

제안한 하드웨어 구조는 PS Converter 블록과 State Generator 블록과 Ridge Checker 블록이 연속적으로 연결되어 있고, 제안된 구조는 3×3 그레이 레벨의 가운데 화소의 연결성을 4-클럭에 검사한다. 전체적인 회로는 설계 툴을 사용하여 정상적인 동작을 수행함을 검증하였다.

참고 문헌

- [1] C. R. Dyer and A. Rosenfeld, "Thinning algorithm for gray-scale pictures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 1, Jan., 1979.
- [2] S. Peleg and A. Rosenfeld, "A min-max medial axis transformation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, pp. 208-210, 1981.
- [3] E. Salari and P. Sly, "The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images," *IEEE Trans. on System, Man, and Cybernetics*, Vol. SMC-14, No. 3, pp. 524-528, 1984.
- [4] C. Wang and K. Abe, "A Method of Gray-scale Image Thinning : The Case without Region specification for Thinning," *IEEE 11th Int. Conf. On Pattern Recognition*, Vol. 3, pp. 404-407, 1992.
- [5] C. J. Hilditch, "Linear Skeletons from square cupboards," *Machine Intelligence IV*, University Press, Edingburgh, pp. 403-420, 1969.
- [6] 박종조, "Local min/max 연산에 의한 제조치 세션화 알고리즘," *대한전자공학회논문지*, 제 35권 S편, pp. 96-104, 1998.
- [7] K. Abe, F. Mizutani and C. Wang, "Thinning of gray-scale images with combined sequential and parallel conditions for pixel removal," *IEEE Trans. System, Man, and Cybernetics*, Vol. 24, No. 2, pp. 294-299, 1994.
- [8] Y. Nakagawa and A. Rosenfeld, "A note on the use of local min and max operation in digital picture processing," *IEEE Trans. System, Man, and Cybernetics*, Vol. SMC-8, No. pp. 632-635, 1978.
- [9] M. Sheu, J. Wang, J. Chen, A. Suen, Y. Jeang and J. Lee, "A Data-Reuse Architecture for Gray-Scale Morphologic Operations," *IEEE Trans. on Circuits and System-II*, Vol. 39, No. 10, pp. 753-756, 1992.
- [10] L. Lucke, C. Chakrabarti, "A Digit-serial Architecture for Gray-Scale Morphological Filtering," *IEEE Trans. on Image Processing*, Vol. 4, No. 3, pp. 387-391, 1995.