

# 필기체 문자 영상의 이진화에 관한 연구

## (A Study on Binarization of Handwritten Character Image)

최 영 규\*                      이 상 범\*\*  
(Young-Kyoo Choi) (Sang-Burm Rhee)

### 요 약

온라인 필기체 문자 인식은 필기의 순서와 획의 위치를 알 수 있어 신경망을 이용한 자소의 효과적인 분할로 큰 성과를 이루었다. 그러나 오프라인 필기체 문자 인식은 동적인 정보와 시간적인 정보를 가지고 있지 않고, 다양한 필기와 자소의 겹침이 심하며 획 사이의 잡영을 많이 가지고 있어 불완전한 전처리를 수행하여야 하는 어려움을 가지고 있다. 따라서 오프라인 필기체 문자 인식은 다양한 방법의 연구가 필요하다.

본 논문에서는 Watershed 알고리즘을 오프라인 필기체 한글 문자 인식 전처리에 적용하였다. 여기서 Watershed 알고리즘의 수행 시간과 결과 영상의 품질을 고려해 Watershed 알고리즘 4단계에서 효과적인 적용 방법을 제시하였다. 효과적으로 구성된 Watershed 알고리즘을 전처리에 적용함으로써 영상 향상과 이진화에 좋은 결과를 얻었다. 실험에서는 기존의 방법과 본 논문 방법을 수행 시간과 품질로써 평가했다. 실험 결과 기존의 방법은 평균 2.08초, 본 논문 방법은 평균 0.86초의 수행 시간이 걸렸다. 결과 영상의 품질은 본 논문 방법이 기존의 방법에 비하여 문자의 획 사이의 잡영을 효과적으로 처리하였다.

### ABSTRACT

On-line handwritten character recognition be achieved successful results since effectively neural networks divided the letter which is the time ordering of strokes and stroke position. But off-line handwritten character recognition is in difficulty of incomplete preprocessing because has not information of motion or time and has frequently overlap of the letter and many noise occurrence. consequently off-line handwritten character recognition needs study of various methods.

This paper apply watershed algorithm to preprocessing for off-line handwritten hangul character recognition. This paper presents effective method in four steps in watershed algorithm as consider execution time of watershed algorithm and quality of result image. As apply watershed algorithm with effective structure to preprocessing, can get to the good result of image enhancement and binarization. In this experiment, this paper is estimate the previous method with this paper method for execution time and quality in image. Average execution time on the previous method is 2.16 second and Average execution time on this paper method is 1.72 second. While this paper method is remove noise effectively with overlap stroke, the previous method does not seem to be remove noise effectively with overlap stroke.

\* 정희원 : 단국대학교 대학원 전자공학과 박사수료

논문접수 : 2002. 3. 21

\*\* 정희원 : 단국대학교 공학부 컴퓨터공학전공 교수

심사완료 : 2002. 4. 15

## 1. 서론

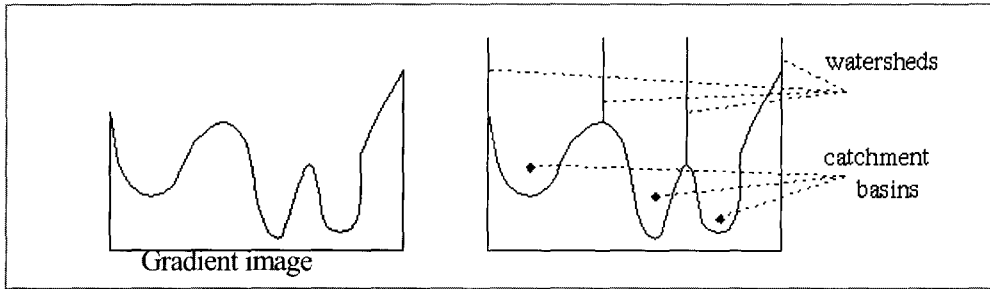
문자 인식에 관한 연구는 급속한 산업의 발달과 정보량의 증가로 대용량 정보의 관리 및 처리 시간과 비용 감소의 필요성으로 시작되었다. 또한 문자 인식은 산업전반에 응용 가치가 매우 큰 분야이다. 지금까지의 다양한 문자 인식 연구는 온라인 필기체 문자 인식 분야에 있어서는 상용 시스템이 등장하는 등 많은 결과를 보였지만, 오프라인 필기체 문자 인식 분야에서는 필기자의 다양성과 동적 정보의 손실로 데이터 처리의 어려움이 있어 상용 시스템이 등장하기까지 많은 연구를 필요로 하고 있다[1-2]. 오프라인 필기체 문자 인식에서는 데이터 획득 시 포함될 수 있는 잡음과 필기 시 발생하는 잡음을 처리하는 전처리 과정을 필요로 한다. 하지만 문자의 방대함과 필체의 다양성, 자소간의 빈번한 접촉으로 효과적인 전처리 방법의 연구를 필요로 하고 있다[3]. 일반적으로 문자 영상의 향상을 위해 필터링 처리와 모폴로지 연산을 적용한다. 가장 많이 사용되는 평균 필터링 처리는 문자 영상 정규화의 크기에 많은 수행 연산을 필요로 하게 된다. 또한 모폴로지 연산은 영상 전체에 여러 번의 반복적인 처리로 수행 시간이 증가하게 된다. 그리고 결과 영상에 있어서 여전히 자소의 접촉 부분의 미세한 잡음 처리가 불안정하다.

본 논문에서는 Watershed 알고리즘을 적용하여 Watershed 문자 영상으로 변환하였다. 문자 영상에 적합하게 Watershed 알고리즘을 효과적으로 구성하여, 수행 시간을 상당히 줄일 수 있었다. Watershed 알고리즘의 구성 단계에서 수행 시간과 관련된 Minima의 수를 적절히 추출할 수 있는 방법을 적용하였고, 각 단계에서 Watershed 문자 영상 질의 신뢰성과 수행 시간을 고려한 방법들로 구성하여 좋은 결과를 얻을 수 있었다. 본 논문에서의 Watershed 문자 영상은 효과적인 영역 분할과 영역 병합의 결과로, 이미지 분석 및 처리가 용이하게 구성된다. Watershed 문자 영상은 정규화 된 형태의 히스토그램(histogram) 분포로 구성되어, 이 영상에 기본 문턱치를 적용하여 자소의 접촉 부분의 미세한 잡음을 효과적으로 처리하였다.

## 2. Watershed 알고리즘

Watershed 알고리즘은 X-ray 영상, 도로 표지 영상, 항공 사진 등의 영상 분할 연구에 사용되어지고 있다. 또한 변환된 Watershed 영상 분석을 통한 다양한 연구가 이루어지고 있다[4-5]. 문자 영상에 있어 자소간의 높은 유사성으로 적절한 Watershed 알고리즘의 적용이 어렵고, 일반적인 Watershed 알고리즘 적용 시 수행 시간이 길다는 단점을 가진다. 여기서 수행 시간은 Watershed 알고리즘의 Minima의 수와 관련지어지는데, Minima의 수가 적절하게 추출되지 못하면 결과 영상에 있어 좋지 않은 결과를 가져온다. 반면 Minima의 수가 너무 많으면 과분할의 결과와 함께 수행 속도도 길어진다. 따라서 Minima의 적절한 추출은 Watershed 알고리즘에 있어 중요한 요소이고 수행 속도와 관련되어진다. Watershed 알고리즘은 영상 처리 분야에서 인접한 영역을 효과적으로 군집화 하는 과정을 통해 영상을 분할하는 목적으로 사용되어지는 기법이다. Watershed 알고리즘은 Lantuejoul 와 Beucher 에 의해 영상 분할의 목적으로 처음 소개되었다[6]. Watershed 알고리즘은 4단계로 구성된다. 1단계는 원 영상으로부터 고도를 가진 경사 영상(gradient image)으로 변환한다. 2단계는 경사 영상으로부터 지역적 낮은 고도(local minima)를 가진 픽셀들을 추출한다. 3단계는 지역적으로 낮은 고도로부터 높은 고도로 범람 과정(flooding)을 수행하여 유사한 영역을 분할한다. 마지막 4단계에서 유사한 영역에 대한 영역 병합(region merging)을 수행한다[4].

일반적으로 Watershed 알고리즘 기본적인 구성은 Vincent 와 Soille에 의한 방법과 Mayer에 의한 방법으로 대표된다. 영상의 명도를 하나의 위상학적 고도로 생각함으로써 이 위상학적 표면에 물을 떨어뜨리면 물은 낮은 고도(minima altitude)에서부터 흘러 들어가 높은 고도(maxima altitude)까지 물이 차 들어가는 원리를 이용하였다. 가장 낮은 높이에서부터 수면 높이를 점차로 증가시키면서, 고도를 넘히는 과정으로 서로 다른 catchment basin이 서로 접하게 될 경우 가상의 댐을 쌓아서 물이 찬 부분을 서로 분리하고, 인접한 영역의 유사성을 조사하여 유사한 영역을 서로 합치는 과정이다[7-8].



[그림 1] Watershed와 Catchment basin  
 [Fig. 1] Watershed and Catchment basin

### 2.1 Catchment basin

Catchment basin은 지역적인 minima들로 구성된 클래스들의 집합을 의미한다[9]. Catchment basin은 식 (1)과 같이 모든 영역  $D$  에서 minima  $m_i$  가 다른 지역적 minima 클래스  $m_j$  에 접할 때까지의 클래스로 정의한다.

$$CB(m_i) = \frac{1}{2} x \in D \mid \forall_j \in \setminus \frac{1}{2} i \neq j : f(m_i) + T_f(x, m_i) < f(m_j) + T_f(x, m_j) \quad (1)$$

Catchment basin은 [그림 1]에서와 같이 지역적인 영향권 내의 집합으로써 영상 분할의 기준을 이룬다.

### 2.2 Watershed

Watershed(  $f_w$  )는 Catchment basin 영역이 아닌 부분을 의미한다. Watershed(  $f_w$  )는 식 (2)와 같이 영상의 전체 영역  $D$  에서 Catchment basin에 보수를 취한 값으로 정의한다[9].

$$f_w = D \cap \left( \bigcup_{j \in I} CB(m_j) \right)^c \quad (2)$$

Watershed는 [그림 1]에서와 같이 영상 분할의 기준이 되는 Catchment basin들의 경계면을 이루면서 Catchment basin들의 상호 작용의 기준이 된다.

Watershed 알고리즘 수행 과정은 4단계로 이루어진다. 1단계는 경사 영상을 구성하는 과정이고, 2단계는 지역적인 낮은 고도를 추출하는 과정이고, 3단계는 범람 과정을 통해 영역을 분할하는 과정이고, 4단계는 유사도가 높은 인접 영역들을 병합하는 과정이다[4].

#### 2.2.1 경사 영상 추출

고도를 가진 영상을 구성하기 위해 문자 영상에 기울기 연산자들을 적용하여 얻어진다. 대표적인 기울기 연산자로 로버츠 연산자(Roberts operator), 소벨 연산자(Sobel operator), 프리윗 연산자(Prewitt operator) 3가지 연산자를 적용하여 수행하였다. 2차 미분 연산자인 라플라스(Laplacian) 연산자는 잡영에 대해 아주 민감한 연산자로서 잡영 부분에 있어 효과적인 처리를 기대할 수 없었다[10]. 로버츠 연산자는 크기가 작지만 돌출 된 값들을 잘 평균화하는 장점을 가지고 있다. 하지만 라플라스 연산자와 같이 잡영에 대해 민감하고, 불필요한 기울기 값을 나타내기도 한다. 로버츠 연산자는 기울기 연산 속도는 빠르나 많은 Minima 부분을 추출함으로써 Watershed 알고리즘의 수행속도는 상대적으로 느리게 나타났다. 소벨 연산자는 대각 방향에 더욱 민감한 기울기 연산자로서 모서리 부분 처리에 중점을 두고 있다[11]. 이 연산자는 로버츠 연산자에 비해 기울기 연산 속도는 느리나 적절한 Minima를 추출한다.

<표 1> 기울기 연산자를 적용한 수행 시간 및 Minima 수  
 <Table 1> The running time and minima number applied gradient operator

| C   | O       | M   | G     | W      |
|-----|---------|-----|-------|--------|
| 1   | Roberts | 198 | 5.552 | 88.702 |
|     | Prewitt | 183 | 6.493 | 88.726 |
|     | Sobel   | 202 | 6.728 | 89.532 |
| 2   | Roberts | 176 | 4.527 | 86.359 |
|     | Prewitt | 189 | 6.350 | 89.487 |
|     | Sobel   | 202 | 6.680 | 89.788 |
| 3   | Roberts | 181 | 4.434 | 83.966 |
|     | Prewitt | 175 | 6.321 | 87.694 |
|     | Sobel   | 189 | 7.175 | 89.378 |
| 98  | Roberts | 167 | 4.519 | 86.379 |
|     | Prewitt | 147 | 6.316 | 86.072 |
|     | Sobel   | 166 | 6.568 | 87.175 |
| 99  | Roberts | 151 | 4.372 | 82.985 |
|     | Prewitt | 121 | 6.572 | 81.829 |
|     | Sobel   | 135 | 6.649 | 83.533 |
| 100 | Roberts | 129 | 4.399 | 80.728 |
|     | Prewitt | 121 | 6.362 | 81.867 |
|     | Sobel   | 132 | 6.440 | 82.664 |

C : Handwritten Character Dataset Number  
 O : Operator  
 M : Minima Detection Counter  
 G : 기울기 연산 수행 시간  
 W : Watershed Transform 수행시간  
 시간단위 : 1/100 초

프리윗 연산자는 영상에서 대각 방향 보다 수직, 수평 방향에 대해 더욱 민감하다[13]. 이 연산자는 로버츠 연산자에 비해 기울기 연산 속도는 느리나 적절한 Minima를 추출한다. 또한 프리윗 연산자 보다 수직, 수평 방향에 중점을 두고 기울기 영상을 변환하므로 한글 특성에 가장 적합하다고 본다. 필기체 한글 문자 영상 샘플 10벌에 대해 기울기 연산자를 적용하여 수행 시간과 Minima 수를 측정된 것을 <표 1>에 보여준다. Minima의 수가 많을수록 상대적으로 Watershed 이미지 변환 수행 시간이 오래 걸린다. 로버츠 연산자는 많은 Minima의 수를 가지나 기울기 연산 속도가 빠르므로, Watershed 이미지 변환 수행 시간이 두 연산자 보다 빠르다. 하지만 많은 Minima의 추출로 영역 분할 및 영역 병합의 신뢰성이 떨어지고, 과분할의 결과를 가질 확률이 대단히 높다.

2.2.2 지역적 저고도 추출

1단계에서 얻어진 경사영상의 각 픽셀에 대해 주변 픽셀과의 경사 값을 조사하여 낮은 경사 값을 갖는 픽셀에 대하여 지역적인 저고도로 추출하는 과정이다. 또한 유사한 픽셀간의 거리를 고려하여 마커(marker)를 순차적으로 초기화한다[7-8].

$$X_{h_{min}} = \{p \in D \mid f(p) = h_{min}\} = T_{h_{min}}$$

$$h_{min} = \min_{r \in N_G(p)} \{f(r) \mid f(r) < f(p)\}$$

$N_G$  : neighbor pixel. (3)

수행 과정을 식 (3)에 표현하였다. 전체 영상에 대해 이웃 픽셀(neighbor pixel)을 고려하여 Minima 부분  $X_{h_{min}}$ 을 마커로서 초기화한다. 이웃 픽셀을 설정하여 주변 픽셀과 마커의 값을 비교하여 local minima를 찾아 인접한 마커와의 관련성을 가진 픽셀들을 추출한다. 여기서  $N_G$ 는 이웃 픽셀에 대한 방향성으로 8방향과 4방향으로 수행하였다. <표 2>는  $N_G$ 의 수에 따른 수행 시간을 측정된 것이다.  $N_G$ 가 4방향 주변 픽셀 고려 시에는 Minima 추출의 수가 많아 수행 속도 저하의 원인이 되기도 하였으며 과분할도 나타났다.  $N_G$ 가 8방향 주변 픽셀 고려 시 적절한 Minima의 추출로 수행 속도가 빠르게 나타났다.

<표 2>  $N_G$ 에 따른 수행 시간  
 <Table 2> The running time by  $N_G$

| C   | $N_G$ | M   | W       |
|-----|-------|-----|---------|
| 1   | 4     | 267 | 98.994  |
|     | 8     | 183 | 91.726  |
| 2   | 4     | 277 | 102.329 |
|     | 8     | 189 | 89.489  |
| 3   | 4     | 241 | 98.143  |
|     | 8     | 175 | 87.694  |
| 98  | 4     | 217 | 98.026  |
|     | 8     | 147 | 86.072  |
| 99  | 4     | 182 | 93.647  |
|     | 8     | 121 | 84.867  |
| 100 | 4     | 167 | 90.459  |
|     | 8     | 121 | 81.829  |

C : Handwritten Character Dataset Number  
 M : Minima Detection Counter  
 W : Watershed Transform 수행시간  
 시간단위 : 1/100 초

2.2.3 영역 분할

지역적으로 분류된 낮은 경사 값을 가지는 픽셀로부터 경사 값이 높은 픽셀로 영역을 확장해 나가는 범람 과정(flooding)을 수행하여 인접한 영역을 분할한다. Minima의 추출 수에 따라 영역의 분할 수도 비례해서 이루어진다[9].

수행과정을 식 (4)에 표현하였다. 전 영역  $D$  에서 지역적으로 수행되어진다. 가장 낮은 고도 ( $h_{min}$ )를 가진 영역에서부터 가장 높은 고도 ( $h_{max}$ )를 가진 영역으로 하나의 catchment basin으로 분할한다. 또 다른 지역에 이와 같은 과정이 평행하게 수행되어진다.

$$X_{h_{min}} = \{p \in D \mid f(p) = h_{min}\} = T_{h_{min}}$$

$$X_{h+1} = MIN_{h+1} \cup IZ_{T_{h+1}}(X_h), \quad h \in [h_{min}, h_{max}] \quad (4)$$

전 단계에서 Minima의 수가 너무 많으면 전 영상에 대해 catchment basin의 영역들도 많이 분할되어져 수행 시간이 길어진다. 또한 과분할의 경우 catchment basin의 분할에 있어 신뢰성이 감소하게 된다.

2.2.4 영역 병합

분할된 영역들은 각각의 영역의 유사성 비교로서 영역을 병합하게 된다. 인접한 영역의 유사성 비교 척도로 영상처리에서 많이 사용되어지는 3가지 방법을 이용하였다[9]. 첫 번째 방법으로 주변 화소의 기울기 값을 이용하여 유사성을 측정하는 것이다. 인접한 두 영역  $R(p)$ 와  $R(r)$ 이 서로 다르고  $R(p)$ 에 대한 영역 개수가 문턱치( $T$ )보다 작으면 두 영역은 유사성을 가지고 있다고 판단하고 병합을 수행한다. 이 과정에 대한 알고리즘을 식 (5)에 표현하였다.

*if*  $R(p) \neq R(r)$   $p, r \in D$  /\* 인접한 두 영역이 다르고 \*/  
 $R_{GC}(p) < T$  /\* 기울기 값이 문턱치 이하일 때 \*/  
*then begin*

$R(p) \cup R(r)$  /\* 인접한 두 영역을 병합한다 \*/  
*end*  
 $p, r = \text{segment region pixel}$   
 $R_{GC} = \text{segment region pixel counter} \quad T = \text{threshold}$  (5)

두 번째 서로 다른 인접 영역에 대해 두 영역의 평균값의 차를 이용하였다. 영상에 있어 명도의 평균값은 외형과의 관련성 때문에 자주 사용되어진다. 인접한 두 영역  $R(p)$ 와  $R(r)$ 이 서로 다르고  $R(p)$ 에 대한 영역 평균값과  $R(r)$ 의 평균값의 차이가 문턱치( $T$ )보다 작으면 두 영역은 유사성을 가지고 있다고 판단하고 병합을 수행한다. 이 과정에 대한 알고리즘을 식 (6)에 표현하였다.

*if*  $R(p) \neq R(r)$   $p, r \in D$  /\* 인접한 두 영역이 다르고 \*/  
 $(R_{sv}(p) / R_s(p)) - (R_{sv}(r) / R_s(r)) < T$  /\* 평균값의 차가 문턱치 이하일 때 \*/  
*then begin*  
 $R(p) \cup R(r)$  /\* 인접한 두 영역을 병합한다 \*/  
*end*  
 $p, r = \text{segment region pixel} \quad T = \text{threshold}$   
 $R_{sv} = \text{Sum Value} \quad R_s = \text{size}$  (6)

마지막으로 서로 다른 인접 영역에 대해 두 영역의 결합된 영역의 분산 값을 이용하였다. 영상에 있어 명도의 변화량은 두 영역의 유사성 비교에 사용되어진다. 인접한 두 영역  $R(p)$ 와  $R(r)$ 이 서로 다르고  $R(p)$ 와  $R(r)$ 이 결합된 영역의 분산 값이 문턱치( $T$ )보다 작으면 두 영역은 유사성을 가지고 있다고 판단하여 병합을 수행한다. 이 과정에 대한 알고리즘을 식 (7)에 표현하였다.

*if*  $R(p) \neq R(r)$   $p, r \in D$  /\* 인접한 두 영역이 다르고 \*/  
 $R_{CM} / (R_{CM} \times R_{CM}) < T$  /\* 결합된 영역의 분산값이 문턱치 이하일 때 \*/

then begin

```

R(p) ∪ R(r) /* 인접한 두 영역을 병합한다 */
end
p, r = segment region pixel T = threshold
RCM = mean of combined region (7)
    
```

영역 병합 과정 또한 영역 분할 과정과 같이 전 단계에서의 Minima의 수가 적은 경우 수행 시간이 빠르게 나타난다.

<표 3> 영역 유사도 측정 방법에 따른 수행 시간  
 <Table 3> The running time by region similarity measurement method

| C   | L                 | W       |
|-----|-------------------|---------|
| 1   | Gradient method   | 576.595 |
|     | Mean difference   | 85.705  |
|     | Combined Variance | 91.726  |
| 2   | Gradient method   | 579.318 |
|     | Mean difference   | 87.068  |
|     | Combined Variance | 89.487  |
| 3   | Gradient method   | 574.797 |
|     | Mean difference   | 83.033  |
|     | Combined Variance | 87.694  |
| 98  | Gradient method   | 576.933 |
|     | Mean difference   | 82.226  |
|     | Combined Variance | 86.072  |
| 99  | Gradient method   | 572.195 |
|     | Mean difference   | 78.861  |
|     | Combined Variance | 81.829  |
| 100 | Gradient method   | 572.308 |
|     | Mean difference   | 78.662  |
|     | Combined Variance | 84.867  |

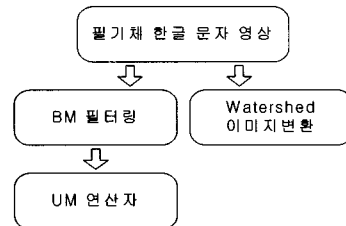
C : Hand-Written Character Dataset #  
 L : 영역유사도 측정 방법  
 W : Watershed Transform 수행시간  
 시간단위 : 1/100 초

<표 3>에서와 같이 기술기에 의한 유사도 측정 방법은 Watershed 문자 영상 변환 수행 시간이 상당히 길어진다. 그리고 평균값의 차를 이용하였을 경우, 수행 시간이 빠르게 나타났고, 영역 병합에 있어서 자소의 접촉 부분에서의 미세한 잡영을 처리하기 용이하게 구성되었다.

### 3. 영상 향상 기법

필기체 한글 문자는 자소간의 접촉과 필체에 있어 획의 홀림이 심해서 미세한 잡영을 흡수할 수 있는 영상 향상 과정을 필요로 한다. 영상 향상 과정은 필기체 한글 문자의 효과적인 영상 분할을 수행하기 위해 필수적인 과정으로 영상 강화 필터링 및 모폴로지 마스크 연산을 통해 효과적인 영상 향상의 결과를 얻게 해준다. 하지만 필터링 처리와 모폴로지 연산은 필기체 한글 문자 영상에 지나친 침식이나 팽창의 결과를 가져올 수 있어 경계선 부분이 불거나 연결 선분이 끊어지는 경우 기존 필기체 문자의 패턴이 훼손 될 경우도 있다. 또한 반복적인 연산을 통한 방법은 수행 속도에 영향을 준다[12].

영상 향상 과정은 전처리 과정에 있어 기본적인 잡영을 처리함으로써 다음 영상 분할을 보다 효과적으로 수행할 수 있게 하고, 영상 분할의 오류를 줄이기 위함이다. 또한 영상 향상 과정에 있어서 수행 시간은 기존의 평균 필터 처리로 많은 시간을 필요로 하고 있다. 이러한 수행 시간의 단점을 보완하기 위해 여러 다양한 방법을 사용한 연구 있으나, 결과 영상에 있어 좋지 않은 결과를 가져와 결국 평균 필터를 이용한 방법을 주로 사용한다.



[그림 2] 영상 향상 기법

[Fig. 2] Image enhancement method

#### 3.1 BM 필터링

BM(Brightness correction+Mean filter) 필터링은 먼저 전체적인 영상의 밝기를 조절하고 평균 필터를 적용하여 잡영을 처리하는 방법이다. 전체적인 히스토그램의 조정을 통해 일부의 잡영에 대한 향상 효과에 남아있는 잡영 제거를 위해 평균 필터를 사용한다[13]. 평균 필터는 모서리 부분에 있어 처리가

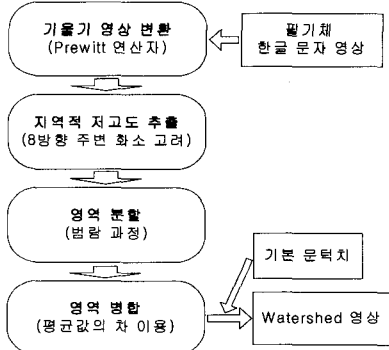
효과적인 특징을 가지고 있다. 또한 경계선이 불거나 뚜렷한 부분이 사라지는 단점을 가지고 있다.

### 3.2 UM 연산자

UM(Unsharping Masking)연산자는 경계선 부분을 강조하는 처리로, 잡영에 민감하다는 단점을 가지고 있으나 독립적인 처리로 사용되지 않으며 평균 필터나 다른 영상 개선 필터와 함께 많이 사용되어진다 [13].

### 3.3 Watershed 문자 영상 변환

Watershed 알고리즘을 적용하여 Watershed 문자 영상으로 변환을 한다. 변환된 문자 영상은 Watershed 알고리즘 수행과정에서 영역 분할 및 영역 병합과정을 통해 효과적인 영상 향상의 효과를 보여준다.

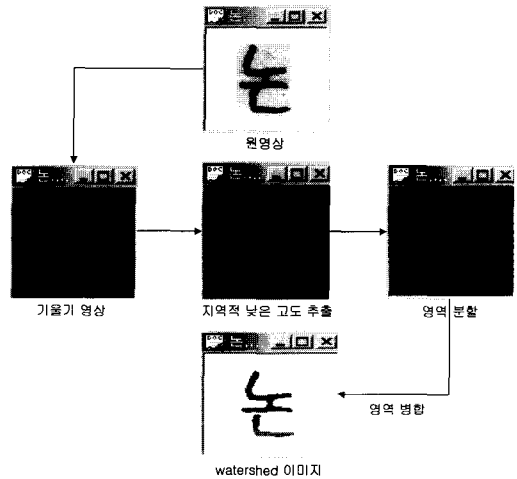


[그림 3] Watershed 영상 변환 과정

[Fig. 3] Watershed image transformation process

Watershed 문자 영상 변환 과정은 [그림 3]에서 보듯이 필기체 한글에 대해 Watershed 알고리즘을 적용하였다. 기울기 변환에서는 기울기 연산자 중 가장 효과적인 연산을 보인 프리윗 연산자를 적용하였고 지역적 낮은 고도 추출에 있어 8방향 주변 픽셀을 고려함으로써 적절한 Minima 추출을 수행하였고, 영역 병합에 있어 인접 픽셀간의 평균값의 차를 이용하였다. 최종적으로 기본 문턱치를 영역 병합 결과 이미지에 적용함으로써 영상 향상 과정을 구성

하였다. 기본 문턱치는 자소의 접촉부분의 잡영을 처리하였다.



[그림 4] Watershed 영상 변환 과정의 예

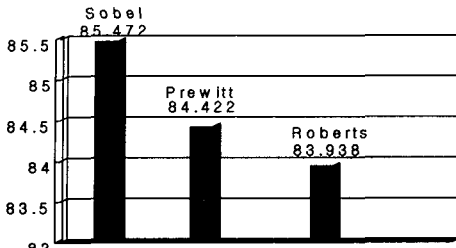
[Fig. 4] An example of Watershed image transformation process

## 4. 실험 및 결과

본 논문의 실험에서는 Intel Pentium(r) III Processor 700MHz, 128MB RAM, Windows Me, Microsoft Visual C++ 언어를 사용하였다. 실험에 사용된 샘플은 KAIST에서 제작한 Korea National Language Information Base(대한민국 국어 정보베이스) CD-ROM에 수록된 한국어 필기체 데이터 베이스 샘플을 사용하였다. 한국어 필기체 데이터 베이스 샘플은 100×100의 명도 영상으로 정규화 되어 있다.

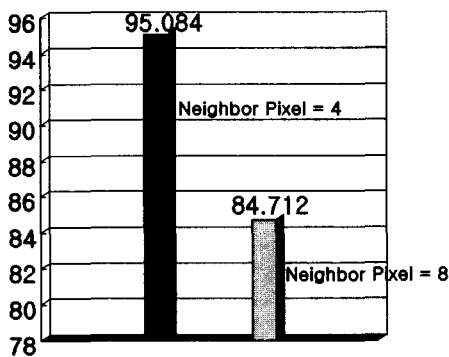
### 4.1 Watershed 알고리즘 각 단계 실험 결과

기울기 연산자 적용 방법에 따라 [그림 5]에서와 같이 100개의 상위 필기체 샘플을 적용한 결과이다. 수행 시간에 있어 로버츠 연산자가 기울기 연산 수행 속도는 빠르나 Minima 추출에 있어 과도한 추출로 과분할의 결과를 가짐으로 프리윗 연산자가 상대적으로 좋은 결과를 보인다.



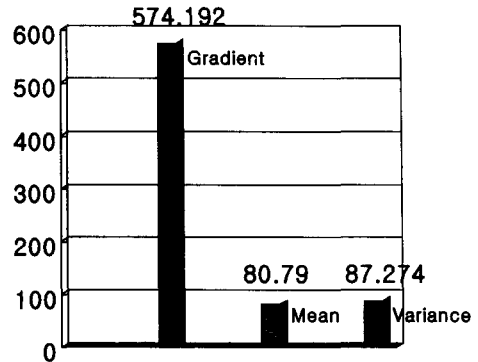
[그림 5] 기울기 연산자에 따른 평균 수행 시간  
 [Fig. 5] The average running time by gradient operator

Local minima detection 단계에 있어서는 [그림 6] 에서와 같이 100개의 상위 필기체 샘플을 적용하였다. 8방향의 주변 픽셀을 고려한 경우 수행 시간에 있어 많은 이득을 보인다.



[그림 6] Neighbor pixel에 따른 평균 수행 시간  
 [Fig. 6] The average running time by neighbor pixel

영역 병합단계에서의 유사도 측정 방법에 따라 [그림 7]과 같이 상위 100개의 샘플을 적용하였다. 평균값의 차를 이용하였을 경우 수행 시간에 있어 빠른 수행을 보인다.



[그림 7] 유사도 측정 방법에 따른 평균 수행 시간  
 [Fig. 7] The average running time by similarity measurement method

#### 4.2 실험 결과 및 검토

<표 4>는 영상 향상 수행 시간을 측정한 것이다. 일반적인 영상 향상 방법인 BM 필터링이나 BM 필터링에 UM 연산자를 이용한 방법은 평균값 필터의 연산 속도가 오래 걸리는 단점을 가지고 있다.

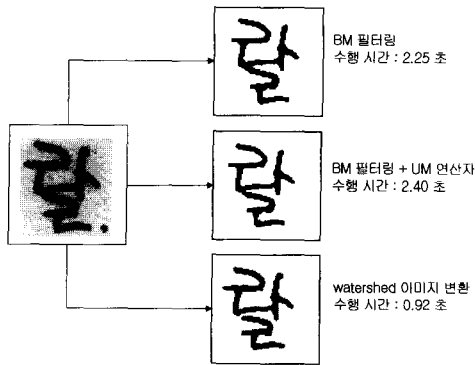
<표 4> 영상 향상 수행 시간

<Table 4> Image enhancement running time

| 영상향상방법      | BM 필터링 | BM 필터링+ UM 연산자 | Watershed 문자영상변환 |
|-------------|--------|----------------|------------------|
| 평균 수행 시간(초) | 2.03   | 2.13           | 0.86             |

이러한 일반적인 평균값 필터의 수행 시간에 대한 단점을 Watershed 알고리즘 각 단계의 수행 시간의 단축으로 영상 향상의 과정에 걸리는 수행 시간을 크게 줄일 수 있었다. 영상 향상의 결과 이미지에 있어서도 일반적인 평균값 필터를 사용하는 방법보다 좋은 결과를 보인다. [그림 8]는 영상 향상 방법에 따른 결과 영상이다. BM 필터링을 이용하였을 경우 자소의 접촉 부분에서의 미세한 잡영을 처리하지 못하고 불게 되는 현상이 발생한다. 또한 UM 연산자를 적용 시 자소의 접촉 부분에서의 처리가 좋아졌지만 여전히 완전히 처리를 하지 못한다. Watershed 문자 영상 변환의 경우 접촉 부분에 있어서의 미세한 잡영을 처리하여 영상이 향상된다.





[그림 8] 영상 향상 방법에 따른 결과 영상

[Fig. 8] The result image by image enhancement method

### 5. 결 론

오프라인 한글에 있어 동적 정보의 손실과 다양한 필체와 자소의 겹침 부위에서 미세한 잡영의 빈번한 발생으로 불완전한 전처리를 수행하여야 한다. 이러한 단점을 보완하기 위해 전처리 과정에 대한 다양한 연구가 필요하다. 이러한 불완전한 전처리 과정 중 영상 향상 과정에서의 수행 시간과 자소의 접촉 부분에서의 잡영을 효과적으로 처리하기 위해 Watershed 알고리즘을 적용하였다.

본 논문에서는 Watershed 알고리즘의 각 단계에서의 적용 방법에 따른 수행 시간을 평가하여 Watershed 알고리즘의 수행 시간을 단축하였다. Watershed 알고리즘을 이용한 Watershed 문자 영상 변환을 통해 효과적으로 자소의 겹침 부분에 있어서의 미세한 잡영을 효과적으로 처리하여 영상을 향상시켰다. 수행 시간 또한 기존의 평균값 필터를 이용한 방법과 비교하여 상당히 빠른 결과를 얻었다. 기울기 영상 변환에 있어서 속도와 결과 영상을 고려해 프리윗 연산자를 적용하였고, 이 연산자는 수평, 수직에 대해 잘 기울기를 강조하여 한글의 특성과 잘 부합되었다. 지역적 낮은 고도 추출에 있어서는 8방향 주변 픽셀을 고려하여 적절한 Minima를 추출하였다. 그래서 수행 시간에 있어서도 이득을 보았고, 4방향 고려 시 잦은 과분할의 단점도 보완하였

다. 영역 병합 과정에서는 두 인접 영역간의 평균값의 차를 이용하여 적절한 영역 병합을 통해 수행 시간의 이득을 얻었다. 따라서 전체적인 Watershed 알고리즘의 수행 시간을 줄였다. 하지만 Watershed 알고리즘의 수행 과정에서 Minima의 적절한 추출을 위해 더욱더 연구가 필요하고 Minima에 있어 영역간의 관련성을 가진 마커의 구성은 많은 연구가 필요 되어진다. 이와 같이 Watershed 문자 영상 변환을 통해 향상된 이미지는 이미지를 처리하기 좋게 구성되어져 다음 영상 분할 과정과 세선화에 있어서도 좋은 결과를 얻을 수 있다. 또한 Watershed 알고리즘의 다양성은 이미지 처리 분야에 다양하게 응용 되어질 수 있다.

### ※ 참고문헌

- [1] 이형구, “은닉 마르코프 모델에 기반한 필기체 한글 인식 방법,” 서울대학교 학위논문, pp.5-13, 1996.
- [2] 이성환, 박희선, “한글 인식의 사례 연구 : 최근 5년 동안의 연구 결과를 중심으로”, 제1회 문자 인식 워크샵 발표 논문집, 충북대학교, pp.3-46, 1993.
- [3] 신충호, 이종은, 김단환, 김형균, 김재석, 오무송, “문자 인식의 전처리 과정에서 영상 향상”, 한국정보처리학회 춘계학술발표논문집, 제8권, 제1호, pp.139-142, 2001.
- [4] Jos B.T.M Roerdink and Arnold Meijster, “The Watershed Transform : Definitions, Algorithms and Parallelization Strategies,” Institutur for mathematics and Computing Science, Report IWI 99-9-06, 1999.
- [5] E. R. Dougherty, “An introduction to morphological image processing,” Optical Engineering, Volume TT9, 1992.
- [6] Michel Couprie and Gilles Bertrand, “Topological Grayscale Watershed Transformation,” Groupe ESIEE Cite Descartes.
- [7] J. Serra and L. Vincent. “Lecture notes in mathematical morphology,” Ecole Nationale Supérieure des Mines de Paris, France. 1989.

- [8] Soille, P. and Vincent, L., "Determining watersheds in digital pictures via flooding simulations," In Visual Communications and Image Processing'90, Volume, SPIE-1360, 1990.
- [9] L. Vincent and P. Soille, "Watersheds in digital spaces : An efficient algorithm based on immersion simulations." PAMI, 13, 1991.
- [10] Gonzalez & Woods, 'Digital Image Processing', 도서출판 그린, pp.165-253, 1998.
- [11] Randy Crane, 'Simplified Approach to Image Processing (H/C)', Prentice-Hall, pp.107-110, 1994.
- [12] 이성환, '문자인식-이론과 실제', 2권, 홍릉과학출판사, pp.217-230, 1993.
- [13] 최경주, 변혜란, 이일병, "효과적인 이진화를 위한 영상 개선 기법의 정의 및 구현," 정보과학회논문지, 제26권, 제2호, pp.284-296, 1999.
- [14] Stefan Thurnhofer, Sanjit K. Mitra, "Quadratic Volterra Filters with Mean-Weighted Highpass Characteristics", IEEE Workshop on Nonlinear Signal and Image Processing, June, 1995.
- [15] J. N. Kapur, P. K. Sanhoo and A. K. Wong, "A New Method for Graylevel Picture Thresholding Using the Entropy of the Histogram," Computer Vision, Graphics and Image Processing, Vol. 29, 1985.
- [16] J. R. Parker, 'Algorithms for Image Processing and Computer Vision', Wiley Computer Publishing, pp. 68-115, 1997.

최영규



1994년 단국대학교 전자공학과 졸업(공학사)  
 1997년 단국대학교 대학원 전자공학과 졸업(공학석사)  
 2001년~ 현재 단국대학교 대학원 전자공학과(공학박사 수료)  
 2000년~2001년 (주)패스싸인 관심분야 : 패턴인식, 인공지능, 멀티미디어응용

이상범



1974년 연세대학교 전자공학과(공학사)  
 1978년 서울대학교 대학원 전자공학과(공학석사)  
 1986년 연세대학교 대학원 전자공학과(공학박사)  
 1984년 미국 IOWA대학교 컴퓨터공학과 객원교수  
 1979년~1999년 단국대학교 전자·컴퓨터공학과 교수  
 1997년~1999년 단국대학교 교무·연구처장  
 1997년~현재 단국대학교 멀티미디어산업기술연구소장  
 2000년~현재 단국대학교 공학부 컴퓨터공학전공 교수  
 관심분야 : 컴퓨터구조, 패턴인식, 디지털 신호처리