

적응적 패턴 정합을 이용한 필기체 한글의 오프라인 인식

박정선, 이성환
충북대학교 컴퓨터과학과

Off-line recognition of handwritten Hangeul using adaptive pattern matching

Jeong-Seon Park and Seong-Whan Lee

Department of Computer Science, Chungbuk National University

요약

본 논문에서는 다양한 양상으로 발생하는 필기체 한글에서의 변형을 흡수하기 위한 적응적 패턴 정합 방법을 제안한다. 제안된 방법은 정합의 정확도를 개선하기 위하여 구조적인 정보를 사용하고, 단일 정합 과정에서의 처리 속도를 개선하기 위하여 각각의 점은 점에서의 이웃들만을 고려하여 처리한다. 또한, 반복 적용시 수렴 속도를 빠르게 하기 위하여 입력 패턴을 부분 영역으로 나누어 각 부분 영역에서의 정합이 성공적인지 여부에 따라 적응적으로 다음 반복 정합 과정을 수행한다.

제안된 방법의 효용성을 입증하기 위하여 다양한 필기체 한글 데이터에 대하여 실험한 결과, 제안된 적응적 패턴 정합 방법이 기존의 패턴 정합 방법보다 처리 속도 및 정합의 성능 측면에서 우수함을 알 수 있었다.

1. 서론

한글 문자 인식에 대한 연구는 지난 20여년 동안 꾸준히 진행되었으나 대부분 오프라인 인쇄체 한글 인식 또는 온라인 필기체 한글 인식에 중점을 두고 있었기 때문에, 오프라인 필기체 한글 인식에 관한 연구는 초보적인 수준에 있다. 이는 필기체 한글의 경우 입력 문자에 포함된 많은 잡영과 왜곡, 그리고 필기자에 따라 다양하게 나타나는 필체의 변형 등을 해결하기가 매우 어렵기 때문이다[이성환93a]. 따라서, 필기체 한글의 오프라인 인식 기법을 개발하기 위해서는 다양한 잡영, 왜곡, 필체의 변형 등의 문제를 해결하는 것이 중요한 관건이다.

원형 정합 방법은 입력 문자 영상을 인식 대상이 되는 모든 문자 모델과의 정합을 통하여 유사도나 거리를 구하여 인식하는 방법으로 인식 방법이 단순하고, 하드웨어로 구현하기가 쉬우며, 단일 활자체와 같이 변형이 심하지 않은 문자 영상은 잘 처리하지만, 필기체 문자와 같이 변형이 심한 데이터에는 적합하지 않은 것으로 알려져 왔다. 그러나, 일본의 Kanji, 중국의 한자, 우리나라

의 한글 등과 같이 문자의 형태가 복잡하고 인식해야 할 문자 부류의 수가 많은 대용량의 필기체 문자 인식에 대한 연구가 활발히 진행되면서 필기체 문자에서 발생하는 변형을 효과적으로 흡수할 수 있는 비선형 형태 정규화 또는 비선형 패턴 정합 방법 등이 개발됨에 따라 원형 정합 방법의 성능이 개선되었고 최근들어서는 그 사용이 증가되고 있는 추세이다[Mori92].

패턴 정합은 원형 정합 방법의 하나로서 두 패턴 사이에 최적의 정합을 찾는 것이 중요하다. 두 패턴 사이에 최적의 정합을 찾는 가장 확실한 방법은 두 패턴 사이에 조합 가능한 모든 정합에 대하여 정합 척도를 계산하여 그 중에서 가장 가까운 정합을 최적의 정합으로 선택하는 것이다. 그러나, 이것은 일반적으로 문자 패턴이 매우 많은 수의 검은 점으로 구성되는 것을 고려해 볼 때 무수한 수의 조합 가능한 정합이 존재하고 그 중에서 최적을 선택한다는 것은 NP-complete 문제이다.

두 패턴을 보다 빠르고 효율적으로 정합하기 위해서 가능한 조합에 제약을 가하는 기법들이 제안되고 있는데, 대표적으로 이완 정합(relaxation matching) 방법[Davis76, Ranad80]이 있다. 이 방법은 특징 쌍에 관계된 확률을 반복적으로 갱신함으로써 최적의 정합을 얻는다. 다른 방법으로 동적 프로그래밍 정합을 이용한 기법[Amini90, Tsuku84]이 있으나 이러한 방법들은 문자 패턴이 갖는 구조적인 정보와 결합하기 위해서는 그에 따른 복잡한 절차가 추가되어야 하는 한계성이 있다.

보다 수학적이며 문자 패턴이 갖는 구조적인 특징과 쉽게 결합할 수 있는 방법으로 국부적 선형 변환을 이용하여 패턴을 정합하는 방법이 최근들어 Wakahara[Wakah90]에 의하여 제안되었다. 이 방법은 비선형 패턴 정합 방법의 하나이며, 검은 점을 기준으로 이 점과 이웃한 검은 점들의 분포를 고려하여 최적의 선형 변환을 찾음으로써 다양하게 발생하는 문자 패턴의 형태 변형을 흡수하고자 한다.

본 논문에서는 Wakahara가 제안한 국부적 선형 변환을 이용한 패턴 정합에서의 문제점을 제시하고, 그 문제점을 해결할 수 있는 새로운 패턴 정합 방법을 제안한다. 제안된 방법은 정합의 정확도를 개선하기 위하여 구조적인 정보를 사용하며, 단일 정합 과정에서의 처리 속도를 개선하기 위하여 각각의 검은 점에서의 이웃만을 고려하여 최적의 선형 변환을 찾는다. 또한, 반복 적용시 수렴 속도를 빠르게 하기 위하여 입력 패턴을 부분 영역으로 나누어 각 부분 영역에서의 정합이 성공적인지 여부에 따라 다음 반복 정합 과정을 수행하는 적응적 패턴 정합(adaptive pattern matching)을 도입한다. 이러한 적응적 패턴 정합은 Wakahara가 지적했듯이 대부분이 첫번째 반복 단계에서 입력 패턴의 70% 이상의 검은 점이 표준 패턴의 검은 점과 일치하고 반복을 거듭하여도 이미 일치된 부분은 거의 변하지 않는 현상에 바탕을 둔다. 이상과 같은 적응적 기법을 적용함으로써 각 반복 단계에서 이미 최적의 변환이 이루어진 영역의 검은 점에 대해 또 다시 계산해야 하는 불필요한 시간의 낭비를 줄일 수 있다.

본 논문의 구성은 다음과 같다. II 장에서는 Wakahara가 제안한 국부적 선형 변환을 이용한 패턴 정합의 기본 개념을 소개하고, III 장에서는 II 장에서 소개한 패턴 정합 방법의 문제점을 개선하기 위한 적응적 패턴 정합 방법을 제안한다. IV 장에서는 제안된 방법의 효용성을 입증하기 위하여 다양한 필기체 한글 데이터에 대하여 실험한 결과를 제시하고 분석한다. 마지막으로 V 장에서는 결론 및 앞으로의 연구 방향에 대하여 논의한다.

II. 국부적 선형 변환을 이용한 기존의 패턴 정합 방법

본 장에서는 Wakahara가 제안한 국부적 선형 변환을 이용한 패턴 정합 방법[Wakah90]의 기본 개념을 소개한다.

먼저, M개의 검은 점으로 구성된 입력 패턴을 S라고 하고 이를 다음과 같이 나타내자.

$$S = (S_1, S_2, \dots, S_i, \dots, S_M) \quad (1)$$

여기서, S_i 는 입력 패턴의 i번째 검은 점의 x좌표와 y좌표로 구성된 위치 벡터이다.

같은 방법으로 N개의 검은 점으로 구성된 표준 패턴을 R이라고 하고 다음과 같이 표현한다.

$$R = (R_1, R_2, \dots, R_k, \dots, R_N) \quad (2)$$

여기서, R_k 는 표준 패턴의 k번째 검은 점의 위치 벡터이다.

다음에 전개될 국부적 선형 변환을 이용한 패턴 정합은 입력 패턴의 각 부분 영역에서의 왜곡을 선형 변환 연산으로 표현할 수 있다는 가정에 기초한다. 즉, 입력 패턴 S와 표준 패턴 R에서의 검은 점의 분포를 각 부분 영역에서의 선형 변환으로 표현할 수 있다고 가정한다. 만약 이러한 가정이 타당하다면 각각의 부분 영역에서 최적으로 정합하는 국부적 선형 변환을 결정함으로써 두 패턴의 정합을 성공적으로 수행할 수 있으며, 또한 두 패턴에 대해서 적당한 윈도우 함수를 적용함으로써 가능한 선형 변환의 범위를 제한할 수도 있다. 이것이 국부적 선형 변환을 이용한 패턴 정합의 기본 개념이다.

2.1 단일 정합 과정

입력 패턴 S에 대한 단일 정합 과정은 입력 패턴의 모든 검은 점에 대해서 다음의 5 단계를 수행함으로써 이루어진다.

■ 단계 1

입력 패턴의 i번째 검은 점 S_i 에 대한 선형 변환을 다음과 같이 정의한다.

$$S_i^* = A_i S_i + B_i \quad (3)$$

여기서, A_i 는 회전, 신축, 밀림 등의 선형 변환을 표현하는 2 x 2 행렬이며, B_i 는 위치 변환을 나타내는 벡터로 x축과 y축의 위치 변환 매개변수로 구성된다.

$$A_i = (a_{pq}), \quad p, q = 1, 2$$

$$B_i = (b_x, b_y)$$

■ 단계 2

변형의 연속성을 보장하기 위해 S_i 에 이웃한 S_j 에도 같은 A_i 와 B_i 를 적용한다.

$$S_j^* = A_i S_j + B_i, \quad j \neq i \quad (4)$$

■ 단계 3

입력 패턴의 i 번째 검은 점 S_i 에 대해서 최적의 A_i 와 B_i 를 결정할 수 있는 목적 함수를 구성하기 위해서 다음을 가정하자. 즉, S_i 에 대해서 A_i 와 B_i 를 적용하여 얻어진 검은 점 S_i^* 와 이에 대응하는 표준 패턴의 검은 점 R_i 와의 차이를 다음과 같이 나타내자.

$$E_i = R_i - S_i^* = R_i - (A_i S_i + B_i) \quad (3')$$

같은 개념으로 R_j 와 S_j^* 에 대해서 E_j 를 다음과 같이 정의한다.

$$E_j = R_j - S_j^* = R_j - (A_i S_j + B_i), \quad j \neq i \quad (4')$$

그러면, 최소 제곱 데이터 적합화(least-squares data fitting) 기준에 의해 최적의 A_i 와 B_i 는 E_i 를 최소화시키는 값이어야 한다. 그러나, 입력 패턴과 표준 패턴의 검은 점 사이의 대응 관계를 알 수 없으므로 이웃하는 검은 점들의 E_j 에 가중치를 적용하여 더한 다음의 목적 함수를 최소화시키는 값이 최적의 A_i 와 B_i 라고 하자.

$$E_i = \sum_{j=1}^M \omega_{ij} \|E_j\|^2 \quad (5)$$

여기서, ω_{ij} 는 가중치로서 입력 패턴의 검은 점 S_i 에 이웃한 검은 점 S_j 가 목적 함수 E_i 에 미치는 영향을 조절하기 위한 가중치이다. 이 가중치는 S_i 와 거리가 먼 점일수록 작아지는 특성을 갖는 다음과 같은 가우스 윈도우 함수(Gaussian window function)에 의해 계산된다.

$$\omega_{ij} = \text{EXP}(- \|S_i - S_j\|^2 / \theta_s^2)$$

여기서, θ_s 는 가우스 윈도우 함수의 분포를 표현하는 윈도우 매개변수이다.

그러나, E_i 와 마찬가지로 E_j 를 구하기 위해서는 입력 패턴과 표준 패턴의 검은 점 사이의 대

응 관계를 알고 있어야 하므로 다음과 같이 S_j 에 대해서 R_k 의 거리에 따른 가중치를 사용해서 계산한다.

$$E_i = \sum_{j=1}^M \omega_{ij} \sum_{k=1}^N \rho_{jk} \|S_j^* - R_k\|^2 = \sum_{j=1}^M \omega_{ij} \sum_{k=1}^N \rho_{jk} \|A_i S_j + B_i - R_k\|^2 \quad (6)$$

여기서, ρ_{jk} 도 S_j 와 R_k 에 대해서 ω_{ij} 와 같은 의미를 갖는 가중치이며 다음 식에 의해 계산된다.

$$\rho_{jk} = \text{EXP}(-\|S_j - R_k\|^2 / \theta_d^2)$$

이 때, θ_d 는 가우스 윈도우 함수의 분포를 표현하는 윈도우 매개변수이며, 다음 식에 의해 정의된다.

$$\theta_d = \alpha \cdot \theta_s, \quad 0.0 < \alpha < 1.0 \quad (7)$$

여기서, α 는 상수이다.

■ 단계 4

최소 제곱 데이터 적합화의 특성에 의해 식 (6)에 정의된 목적 함수는 A_i 와 B_i 의 각각의 원소에 대해 편미분한 값이 0이 될 때 최소값을 갖는다.

$$\partial E_i / \partial a_{pq} = 0, \quad p, q = 1, 2 \quad (8)$$

$$\partial E_i / \partial b_r = 0, \quad r = x, y$$

위 식은 6 개의 선형 방정식이 되며 이 방정식의 해를 찾음으로써 최적의 A_i 와 B_i 를 구할 수 있다.

■ 단계 5

단계 4에서 구해진 선형 변환을 입력 패턴에 적용함으로써 변환된 입력 패턴을 얻는다.

$$S_i^* = A_i S_i + B_i$$

이상의 단일 정합 과정은 입력 패턴의 검은 점의 수가 M이고 표준 패턴의 검은 점의 수가 N

일 때, 총 $M \times M \times N$ 의 계산 복잡도가 요구된다. 입력 패턴의 모든 검은 점은 점에 대한 병렬 처리를 가 정한다고 하더라도 최소한 $M \times N$ 의 계산 복잡도가 필요하다.

입력 패턴의 검은 점 S_i 에 대해서 위의 5 단계를 거쳐 정합된 예가 그림 1에 있다. 그림에서 빗금친 원은 입력 패턴의 검은 점들을 나타내고 흰 원은 표준 패턴의 검은 점들을 나타내며, 검은 원은 입력 패턴을 최적의 A_i 와 B_i 에 의해 변환하였을 때 입력 패턴의 검은 점과 표준 패턴의 검은 점이 일치된 것을 나타낸다. 만약 이웃한 검은 점들의 분포를 고려하지 않고 최적의 변환을 찾는다 면 입력 패턴의 검은 점 S_i 는 바로 인접한 표준 패턴의 검은 점 R_r 과 대응하게 될 것이다. 그러나, 이 윗한 검은 점들의 분포를 고려하여 최적의 변환 함수를 찾는다면, 입력 패턴과 표준 패턴의 전체적 인 차이를 최소화시키는 R_k 와 대응하도록 변환이 이루어짐을 볼 수 있다.

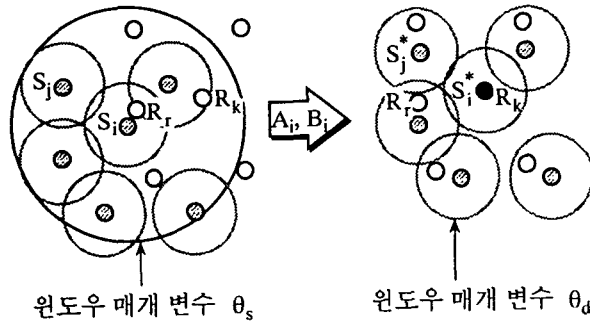


그림 1. 국부적 선형 변환의 예

2.2 반복 정합 과정

어떤 알고리즘을 반복 적용할 때 유의해야 할 검은 점은 검은 점의 반복 횟수가 증가함에 따라 알고리즘이 수렴하는 성질을 가져야 한다는 것이다. 국부적 선형 변환을 이용한 패턴 정합에서는 다음과 같이 윈도우 매개변수의 크기를 점차 줄임으로써 알고리즘이 수렴하도록 유도한다.

$$\begin{aligned} \theta_s(t) &= \theta_s(0) / \sqrt{t} \\ \theta_d(t) &= \alpha \cdot \theta_s(t) \end{aligned} \tag{9}$$

여기서, t 는 반복 횟수를 나타내며, $\theta_s(0)$ 는 초기 윈도우 매개변수의 값으로 실험에 의하여 경험적으로 결정한다.

1번째 반복 단계 이후의 변환된 입력 패턴 $S(t)$ 와 표준 패턴 R 사이의 거리 $D(t)$ 는 다음과 같이 계산된다.

$$D(t) = \frac{1}{2} \left[\frac{1}{M} \sum_{i=1}^M \min_k \|S_i(t) - R_k\|^2 + \frac{1}{N} \sum_{k=1}^N \min_i \|R_k - S_i(t)\|^2 \right] \quad (10)$$

이 때, 입력 패턴과 표준 패턴의 검은 점의 수로 나누기 때문에 검은 점의 수에 무관한 거리를 얻을 수 있다.

이상과 같은 반복 정합 과정은 반복 횟수 t 가 증가할수록 입력 패턴과 표준 패턴과의 거리 $D(t)$ 는 점점 감소하게 된다. 또한, 윈도우 매개변수의 값이 점점 작아지기 때문에 결국은 점-대-점 변환이 이루어지며, 결과적으로 입력 패턴에는 더 이상의 변환이 이루어지지 않고 수렴하게 된다.

III. 제안된 적응적 패턴 정합 방법

본 장에서는 II 장에서 소개한 국부적 선형 변환을 이용한 패턴 정합 방법의 문제점을 지적하고 이것을 해결하기 위한 새로운 적응적 패턴 정합 방법을 제안한다. 먼저, 국부적 선형 변환의 정확성을 높이기 위해 구조적인 정보를 사용하는 개념과 단일 정합 과정에서 불필요한 계산 낭비를 줄이기 위하여 각각의 검은 점에 대한 이웃만을 고려하여 처리하는 방법과 국부적 선형 변환의 반복 적용시 수렴 속도를 가속화하기 위한 적응적 패턴 정합 방법을 소개한다.

3.1 구조적인 정보

앞 절에서 소개한 국부적 선형 변환을 이용한 패턴 정합은 이웃하는 검은 점들의 분포 정보를 고려하여 선형 변환을 찾기 때문에 이렇게 찾아진 변환이 전역적으로 볼 때는 국부적 최소(local minima)에 빠질 우려가 있다. 이러한 현상을 방지하고 정합의 정확도를 개선시키기 위해 입력 패턴과 표준 패턴의 각각의 검은 점에서 구조적인 정보를 추출한 다음, 이 정보를 목적 함수에 추가함으로써 구조적인 정보가 서로 유사한 방향으로 패턴 정합이 이루어지도록 한다[Wakah93].

본 논문에서는 구조적인 정보로서 각각의 검은 점을 중심으로 문자 패턴을 8 방향으로 주사 하면서 만나는 획의 수를 사용한다. 이 때 각각의 검은 점을 기준으로 0° 방향부터 시작하여 45° 씩 회전하면서 교차 횟수(crossing count) 특징을 추출한다. 문자 패턴의 검은 점에 대해서 이와 같은 8 방향 교차 횟수 특징을 추출한 예가 그림 2에 있다.

다음으로, 추출된 구조적인 정보를 앞 장에서 정의된 목적 함수에 추가하여 식 (11)과 같이 개선된 목적 함수를 유도한다.

$$\Phi_i = \sum_{j=1}^M \omega_{ij} \sum_{k=1}^N \rho_{jk} \xi_{jk} \|A_i S_j + B_i - R_k\|^2 \quad (11)$$

여기서, ξ_{jk} 는 S_j 와 R_k 의 8 방향 교차 횟수 특징인 벡터 C 의 유사도이며, 0부터 1까지의 실수 값을 가져야 한다. 이러한 성질을 만족하는 유사도 함수로서 C 값의 차에 대한 가우스 함수를 사용할 수

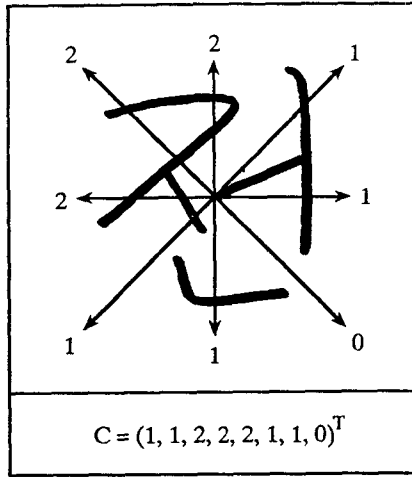


그림 2.8 방향 교차 횟수 특징

있다.

$$\xi_{jk} = \text{EXP}(-\|C(S_j) - C(R_k)\|^2)$$

여기서, $C(S_j)$ 는 입력 패턴의 i 번째 검은 점 S_j 의 8 방향 교차 횟수 특징 벡터이며, $C(R_k)$ 는 표준 패턴의 k 번째 검은 점 R_k 의 8 방향 교차 횟수 특징 벡터이다.

구조적인 정보를 사용하여 개선된 목적 함수는 ξ_{jk} 가 추가된 것만을 제외하면 앞장에서 유도된 식 (6)과 유사하므로 목적 함수를 최소화시키는 A_i 와 B_i 는 앞장에서와 유사한 방법으로 계산할 수 있다.

3.2 단일 정합 과정의 복잡도 개선

앞장에서 소개한 Wakahara가 제안한 국부적 선형 변환을 이용한 패턴 정합 방법[Wakah90, Wakah93]을 살펴 보면, 단일 정합 과정의 3 단계에서 목적 함수를 계산할 때, 입력 패턴과 표준 패턴의 모든 검은 점에 대해서 점 사이의 거리를 계산하고 가우스 윈도우 함수를 적용한다. ω_{ij} 를 계산하기 위한 가우스 윈도우 함수의 특성을 살펴보면 S_i 와 S_j 의 거리가 클수록 적용되는 가중치는 거의 0에 가깝게 된다(그림 3).

이와 같이 S_i 를 기준으로 거리가 매우 먼 점일수록 목적 함수에 끼치는 영향이 매우 적다는 것을 감안할 때, 패턴의 모든 검은 점을 고려하는 것은 불필요한 계산의 낭비를 초래한다는 문제점이 있다. 이러한 문제를 해결하기 위하여 거리가 일정 영역 이하인 검은 점들만을 이웃으로 정의하고, 이 이웃을 이용해서 목적 함수를 식 (12)와 같이 다시 나타내 보자.

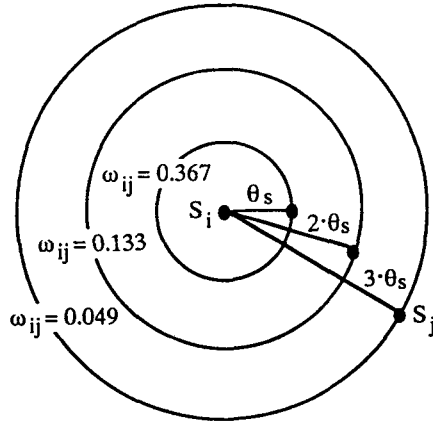


그림 3. 거리에 따라 다르게 적용되는 가중치의 예

$$\begin{aligned} \Phi_i = & \sum_{j \in W_i} \omega_{ij} \sum_{k \in W_j} \rho_{jk} \xi_{jk} \|A_i S_j + B_i - R_k\|^2 \\ & + \sum_{j \in W_i} \omega_{ij} \sum_{k \notin W_j} \rho_{jk} \xi_{jk} \|A_i S_j + B_i - R_k\|^2 + \sum_{j \notin W_i} \omega_{ij} \sum_{k=1}^N \rho_{jk} \xi_{jk} \|A_i S_j + B_i - R_k\|^2 \end{aligned} \quad (12)$$

여기서, W_i 는 입력 패턴의 검은 점 S_i 와의 거리가 임계값 이하인 입력 패턴의 검은 점만을 포함하는 집합이며, W_j 는 입력 패턴의 검은 점 S_j 와의 거리가 임계값 이하인 표준 패턴의 검은 점만을 포함하는 집합이다.

위 식에서 두번째 항과 세번째 항은 가우스 윈도우 함수의 성질에 의해 0에 근사하는 것을 알 수 있다. 따라서, 식 (11)에 정의된 목적 함수에 근사하는 다음과 같은 함수를 유도할 수 있다.

$$\hat{\Phi}_i = \sum_{j \in W_i} \omega_{ij} \sum_{k \in W_j} \rho_{jk} \xi_{jk} \|A_i S_j + B_i - R_k\|^2 \quad (13)$$

입력 패턴의 모든 검은 점에 대해서 S_i 의 이웃 W_i 에 속하는 평균 검은 점의 수를 $|W_i|$ 라고 하고 S_j 의 이웃 W_j 에 속하는 평균 검은 점의 수를 $|W_j|$ 라고 할 때, 위 식에 의해 정의된 근사 함수를 사용하여 입력 패턴 S 에 대한 단일 정합 과정을 수행할 경우 계산 복잡도는 $M \times |W_i| \times |W_j|$ 가 된다.

3.3 수렴 속도를 가속화하기 위한 적응적 패턴 정합

II 장에서 소개된 패턴 정합은 매 반복 단계마다 모든 검은 점에 대하여 국부적 선형 변환을 구하게 된다. 그러나, 윈도우 매개변수만 적절히 정의한다면 국부적 선형 변환을 이용할 때 패턴 정합을 한번 반복하였을 경우 입력 패턴의 70% 이상의 검은 점이 표준 패턴의 검은 점과 일치하고 반

복을 거듭하여도 이미 일치된 부분은 거의 변하지 않는다는 사실을 고려해 볼 때, 모든 검은 점에 대한 변환은 불필요한 시간의 낭비를 초래하며, 수렴 속도도 그만큼 느려진다. 이러한 문제를 해결하기 위하여 입력 패턴을 부분 영역으로 나누어 각 부분 영역에서의 정합이 성공적인지 여부에 따라 적응적으로 다음 반복 정합 과정을 적용하는 것이 본 논문에서 제안된 적응적 패턴 정합의 기본 개념이다.

그림 4는 Wakahara의 논문[Wakah90]중에 예시되어 있는 "5"에 대한 두 패턴의 정합을 바탕으로 적응적 패턴 정합의 예를 보여준다. 그림에서 "I"는 입력 패턴의 검은 점을 표시하고, "O"는 표준 패턴의 검은 점을 나타내며, "●"은 입력 패턴의 검은 점과 표준 패턴의 검은 점이 서로 일치한 것을 나타낸다. 이 그림을 바탕으로 적응적 패턴 정합의 수행 과정을 설명하면 다음과 같다.

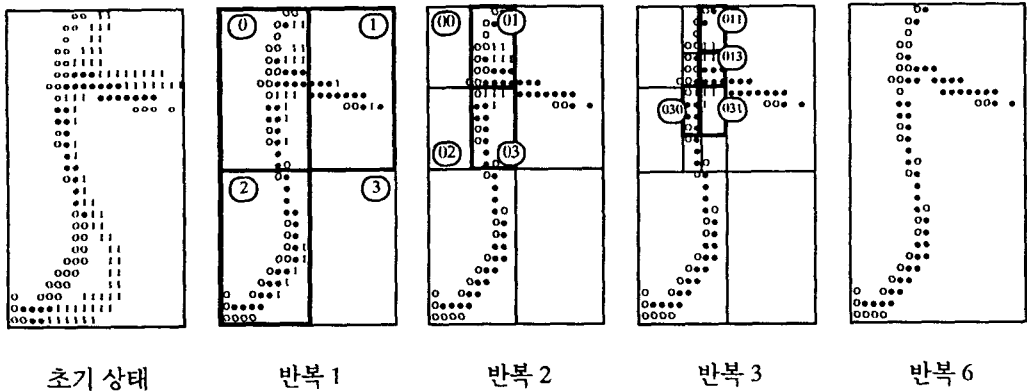


그림 4. 적응적 패턴 정합의 예

■ 반복 1

1. 먼저, 입력 패턴에 대하여 표준 패턴과의 국부적 선형 변환을 수행한다.
2. 변환된 입력 패턴과 표준 패턴과의 거리를 계산한다.
3. 거리가 임계치 이하이면 정합을 끝낸다.
4. 그렇지 않으면, 패턴을 4 개의 부분 영역(0, 1, 2, 3)으로 나누어 각각의 부분 영역별로 거리를 계산하여 다음 반복 정합 과정을 적용할 것인가를 결정한다. (그림에서는 부분 영역 0, 1, 2가 다음 반복 정합 과정에 적용될 영역이다.)

■ 반복 2

1. 앞 단계에서 결정된 반복 과정 적용 대상 영역(0, 1, 2)에 대하여 국부적 선형 변환을 수행한다.
2. 변환된 각각의 부분 영역에서 대해서 표준 패턴과의 거리를 계산한다.
3. 모든 부분 영역의 거리가 임계치보다 작으면 정합을 끝낸다.

4. 그렇지 않으면 거리가 임계치보다 큰 부분 영역을 다시 4 개의 부분 영역으로 나누어 다음 반복 정합 과정을 적용할 것인가를 결정한다. (그림에서는 부분 영역 01과 03이 다음 반복 정합 과정에 적용될 영역이다.)

■ 반복 3

반복 2와 같은 과정을 거친다.

(그림에서는 부분 영역 011, 013, 030, 031이 다음 반복 정합 과정에 적용될 영역이다.)

⋮

■ 반복 6

표준 패턴과의 거리가 임계치보다 작으므로 정합을 끝낸다. 이 반복 과정에서 입력 패턴의 모든 검은 점이 표준 패턴의 검은 점과 일치한다.

이상에 기술된 적응적 패턴 정합 알고리즘은 입력 패턴에 대해서 국부적 선형 변환을 계산하는 과정과 패턴을 간단한 형태의 부분 영역들로 나누는 과정으로 구성된다. 이 때 부분 영역의 갯수, 크기 및 모양은 사용자가 임의로 결정할 수 있다.

IV. 실험 및 결과 분석

제안된 방법은 PC 486(33MHz) 상에서 C 언어를 사용하여 구현되었으며, 7mm x 8mm 사각형 내에 필기된 문자를 Microtek MSF-300C 영상 스캐너를 통하여 200 DPI의 해상도로 입력받은 필기체 한글 데이터를 테스트에 사용하였다. 실험 대상 문자로는 한글 기계화 연구소에서 조사한 한글 사용 빈도순 상위 520자[한글75]를 선택하였으며, 10명의 필기자가 한 문자에 대하여 각각 10번씩 필기한, 총 52,000자의 필기체 한글 데이터를 사용하였다. 이 때, 8명이 필기한 문자 데이터를 학습 데이터로 사용하였으며, 나머지는 테스트 데이터로 사용하였다.

4.1 인식 시스템의 구조

앞서 소개된 국부적 선형 변환을 이용한 패턴 정합을 사용하여 오프라인 필기체 한글을 인식하는 시스템의 구조는 다음과 같다.

먼저, 필기체 문자에서 발생하는 변형, 즉 불균일한 획의 두께, 획 사이의 간격, 그리고 필기자의 습관에 의해 발생하는 비정상적으로 긴 획 등의 변형을 전역적으로 흡수하기 위하여 대용량

의 필기체 문자 인식에 효과적인 것으로 알려진 비선형 형태 정규화[이성환93b]를 수행한다. 본 논문에서는 여러 가지 비선형 형태 정규화 방법 중 처리 속도의 측면에서는 다소 느리지만 인식을 및 변형량의 측면에서 성능이 우수한 것으로 보고된 Yamada 등이 제안한 내접원에 바탕을 둔 획 밀도를 이용한 방법[Yamad90]을 사용하여 필기체 한글에서 발생하는 전역적인 변형을 흡수하고자 하였다.

다음은 대 분류 단계로서 식 (10)에 정의된 거리 계산 방법을 적용하여 모든 표준 패턴과의 거리를 계산하여 거리가 가까운 소수의 후보 문자 집합으로 대 분류한다.

마지막으로 대 분류된 소수의 후보 문자 집합에 대하여 상세 분류를 수행하는데 반복적인 패턴 정합을 적용하여 입력 패턴을 변환시킨 다음 거리를 계산하여 가장 가까운 표준 패턴의 부류를 최종 인식 문자 부류로 결정한다.

4.2 실험 결과

그림 5는 "문"의 입력 패턴에 대해서 비선형 형태 정규화를 통하여 문자 패턴의 전역적인 변형을 흡수한 다음 모든 표준 패턴과의 거리를 계산하여 대 분류된 후보 문자 집합 중 상위 5 등까지의 표준 패턴과 반복적으로 정합하는 과정을 보여준다. 대 분류에서는 "문"의 표준 패턴과의 거리가 더 가까워 "문"으로 오인식되었지만, 반복적으로 패턴 정합을 적용한 결과 입력 패턴이 "문"의

정규화	대 분류	상세 분류						
	표준	반복 1	반복 2	반복 3	반복 4	반복 5	반복 6	반복 7
후보 "문"	 D=744	 D=585	 D=328	 D=247	 D=239	 D=229	 D=224	 D=224
후보 "문"	 D=872	 D=527	 D=312	 D=264	 D=244	 D=209	 D=205	 D=205
후보 "료"	 D=1252	 D=662	 D=509	 D=454	 D=360	 D=309	 D=307	 D=307
후보 "은"	 D=1883	 D=947	 D=750	 D=723	 D=687	 D=680	 D=673	 D=666
후보 "은"	 D=1937	 D=1083	 D=842	 D=650	 D=568	 D=532	 D=529	 D=529

그림 5. 반복적인 패턴 정합의 예

표준 패턴과 점점 더 유사해지면서 거리가 작아져 "문"으로 정확히 인식된다. 또한 일정 반복 후에는 입력 패턴에 더 이상의 변환이 발생하지 않고 수렴한다. 이 때, 후보 "은"에 대하여 입력 패턴을 변환하였을 경우 "ㄱ"의 수직획 부분은 "은"의 어떠한 부분과도 정합될 수 없으므로 일정 반복 후에도 정합되지 않고 남아 있는 것을 볼 수 있다.

본 논문에서 제안된 방법의 특징은 크게 다음의 세 가지로 요약할 수 있다. 첫째, 정합의 정확도를 개선하기 위하여 구조적인 정보를 사용하고, 둘째, 단일 정합 과정의 처리 속도를 개선하기 위하여 각 점에 대하여 이웃을 고려하여 처리하며, 셋째, 반복 정합시 수렴 속도를 빠르게 개선하기 위하여 적응적으로 반복 정합을 수행한다.

본 절에서는 이상과 같은 특성을 갖는 제안된 패턴 정합 방법의 성능을 검증하기 위하여 대분류된 소수의 후보 문자 집합에 대하여 다음 5 종류의 패턴 정합 방법을 적용하여 상세 분류하였다. 이 때, 상위 10 등까지의 대분류율은 99.31%이다.

- 방법 1 : 기존의 방법
- 방법 2 : 구조적인 정보의 사용
- 방법 3 : 일정 거리 이내에 이웃하는 검은 점들만 고려
- 방법 4 : 적응적으로 패턴 정합을 수행함
- 방법 5 : 방법 2, 3, 4를 모두 적용

상기된 각 방법의 성능을 인식률과 처리 속도의 측면에서 비교하였다. 인식률은 누적 인식률을 비교하였으며, 처리 속도는 하나의 문자 패턴을 상세 분류하는데 소요되는 평균 CPU 시간을 측정하였다. 그림 6은 상세 분류에 적용된 방법에 대하여 누적 인식률을 보여 주며, 그림 7은 각 방

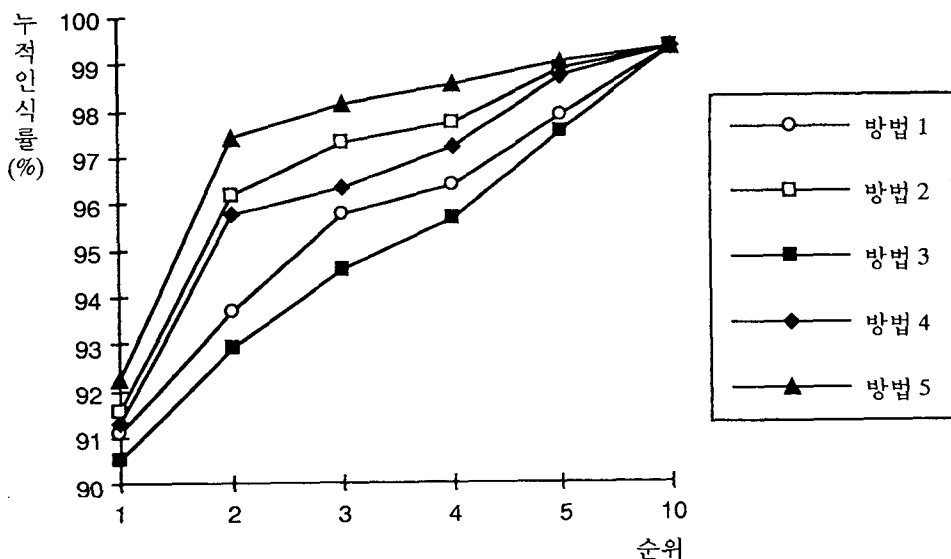


그림 6. 각 방법의 누적 인식률

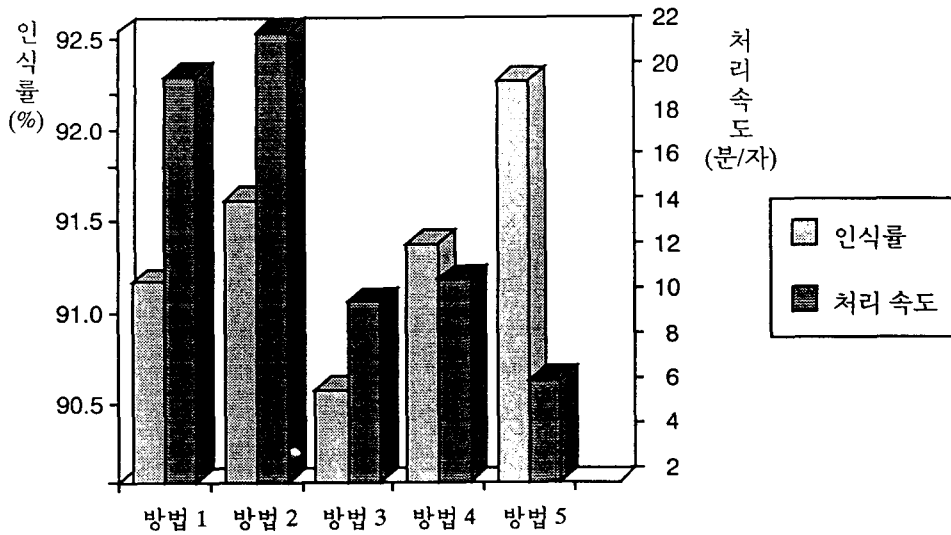


그림 7. 각 방법에 대한 처리 속도와 인식률

법의 인식률과 처리 속도를 같이 나타냄으로써, 인식률과 처리 속도의 관계를 한 눈에 비교할 수 있도록 하였다.

그림 6과 그림 7로부터 각 방법의 성능을 다음과 같이 파악할 수 있다. 첫째, 구조적인 정보를 사용하였을 경우 기존의 방법보다 처리 속도는 다소 느리지만 정합의 정확도가 개선되어 인식률이 향상되었다. 둘째, 단일 정합 과정에서 가우스 윈도우 함수의 특성을 조사하여 이웃만을 고려함으로써 인식 성능에는 큰 차이없이 처리 속도가 향상되었다. 셋째, 적응적 패턴 정합을 도입함으로써 수렴 속도 및 정합의 성능을 대폭 개선하였다. 마지막으로, 제안된 방법을 모두 적용하였을 경우, 기존의 방법에 비해 정합의 성능 및 처리 속도 측면에서 기존의 방법보다 매우 우수함을 알 수 있었다.

또한, 제안된 적응적 패턴 정합 기법의 수렴 속도가 빨라짐을 보이기 위해 그림 4의 데이터에 대해 표준 패턴과의 거리가 감소하는 예를 그림 8에 나타내었다. 이 때, 입력 패턴과 표준 패턴 사이의 초기 거리를 100으로 환산하여 반복 횟수에 따라 거리가 감소하는 비율을 나타내었다. 그래프로부터 제안된 적응적 패턴 정합 기법이 기존의 패턴 정합보다 빠르게 최적의 정합에 수렴함을 알 수 있다.

V. 결론

본 논문에서는 다양한 양상으로 발생하는 필기체 한글에서의 변형을 흡수하기 위한 적응적 패턴 정합 방법을 제안하였다. 제안된 방법은 정합의 정확도를 개선하기 위하여 구조적인 정보를 사용하며, 단일 정합 과정에서의 처리 속도를 개선하기 위하여 각각의 점은 점에서의 이웃만을 고

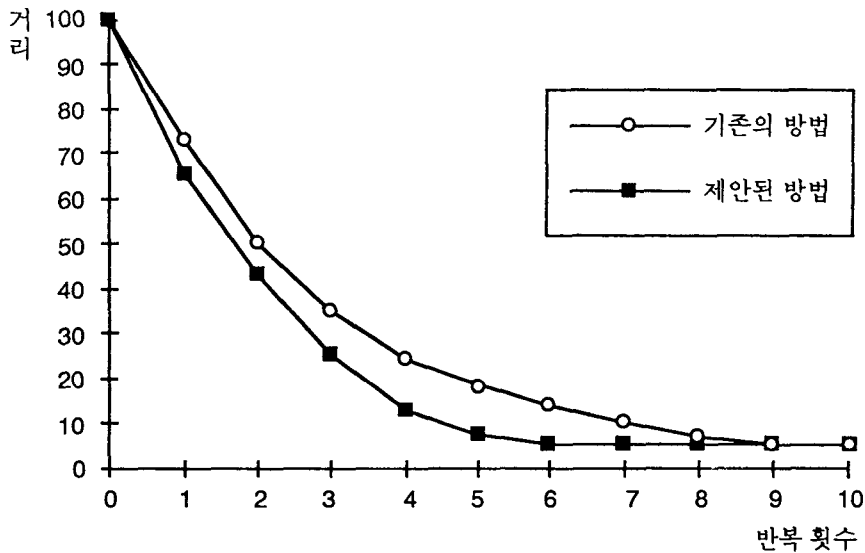


그림 8. 수렴 속도의 비교

려하여 처리한다. 또한, 패턴 정합의 반복 적용시 수렴 속도를 빠르게 하기 위하여 입력 패턴을 부분 영역으로 나누어 각 부분 영역에서의 정합이 성공적인지 여부에 따라 다음 반복 정합 과정을 적응적으로 수행한다. 실험을 통하여 제안된 적응적 패턴 정합 방법이 기존의 패턴 정합 방법보다 수렴 속도 및 정합의 정확도 측면에서 우수함을 알 수 있었다.

본 연구 결과를 바탕으로 앞으로 연구해야 할 과제로는 다양한 실험에 의해 경험적으로 초기 가중치 매개변수를 결정하는 단점을 극복하기 위하여 패턴의 특성에 따라 자동으로 초기 가중치 매개변수를 결정할 수 있는 기법이 개발되어야 하며, 제안된 방법에 적합한 표준 패턴의 구축 방법에 대한 연구도 필요하다.

참고 문헌

- [Amini90] A. A. Amini, T. E. Weymouth and R. C. Jain, "Using Dynamic Programming for Solving Variational Problems in Vision," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 9, 1990, pp. 855-867.
- [Davis76] L. S. Davis and A. Rosenfeld, "An Application of Relaxation Labeling to Spring-Loaded Template Matching," Proc. 3rd Int. Conf. on Pattern Recognition, 1976, pp. 591-597.
- [Mori92] S. Mori, C. Y. Suen and K. Yamamoto, "Historical Review of OCR Research and Development," Proceedings of the IEEE, Vol. 80, No. 7, July 1992, pp. 1029-1058.
- [Ranad80] S. Ranade and A. Rosenfeld, "Point Pattern Matching by Relaxation," Pattern Recognition, Vol. 12, 1980, pp. 269-275.
- [Tsuku84] J. Tsukumo and K. Asai, "Non-Linear Matching Method for Handwritten Character

- Recognition," Proc. 7th Int. Conf. on Pattern Recognition, Montreal, Canada, Aug. 1984, pp. 770-773.
- [Wakah90] T. Wakahara, "Dot Image Matching Using Local Affine Transformation," Proc. 10th Int. Conf. on Pattern Recognition, Atlantic City, USA, June 1990, pp. 837-841.
- [Wakah93] T. Wakahara, "Handwritten Numeral Recognition Using LAT with Structured Information," Proc. 3rd Int. Workshop on Frontiers in Handwriting Recognition, Buffalo, USA, May 1993, pp. 164-174.
- [Yamad90] H. Yamada, K. Yamamoto and T. Saito, "A Nonlinear Normalization for Handprinted Kanji Character Recognition - Line Density Equalization," Pattern Recognition, Vol. 23, No. 9, 1990, pp. 1023-1029.
- [이성환93a] 이 성환, 문자 인식: 이론과 실제, 홍릉과학출판사, 1993년 8월.
- [이성환93b] 이 성환, 박 정선, "대용량 필기체 문자 인식을 위한 비선형 형태 정규화 방법의 정량적 평가," 대한전자공학회 논문지, 제 30권 B편 제 9호, 1993년 9월, pp. 896-905.
- [한글75] 한글 기계화 연구소, 한글 기계화 연구, 1975년.