

# 혼합형 제어 전략을 사용한 인쇄체 한글 문자의 인식

심 원 태, 김 진 형

한국과학기술원, 전산학과

## Recognition of Printed Korean Characters using a Mixed Control Strategy

Wontae Sim and Jin H Kim

KAIST, Department of Computer Science

### 요 약

패턴 인식 알고리즘의 제어 전략에는 인식하고자 하는 패턴의 모델을 그 모델의 계층적 규칙에 따라 여러 개의 부분으로 분리한 다음, 각 부분을 인식하여 하나의 패턴을 인식하는 하향식과 적당히 전처리를 하고 분할 한 후에 패턴을 인식하는 상향식이 많이 쓰이고 있다. 그러나 이 두 방법은 각각의 장 단점이 존재하므로 혼합하여 사용하는 것이 보다 효과적이다. 본 논문에서는 패턴 인식의 제어 전략 중 상향식(bottom-up)과 하향식(top-down)이 혼합된 혼합형(mixed) 제어 전략을 사용한 인쇄체 한글 문자 인식 알고리즘을 제안한다. 이 제어 전략 하에서 인식될 글자는 상향식 방법으로 한글 문자의 기본적인 6가지 유형 중의 하나로 먼저 분류함으로써 글자의 탐색 공간을 축소시켰으며, 그 다음에는 분류된 글자의 유형을 가지고 하향식 방법으로 음소를 인식하였다. 인식하는 과정에서 잘못이 발견될 때에는 잘못이 발생한 지점으로 피드백함으로써 인식 알고리즘에 융통성을 부여하였다.

### I. 서 론

패턴을 인식하고 분석하려는 대상이 점점 복잡하고 다양해짐에 따라, 효율적으로 패턴을 분석하고 인식하기 위한 제어 전략이 필요하게 되는데, 한글 문자 인식에 대한 연구에 있어서도 초기에는 기본 자모만을 인식 대상으로 하였기 때문에 제어 전략이 필요치 않았지만, 인식 대상이 기본 자모에서 자모로 구성된 글자의 인식에 대한 연구로, 그리고 잠음이 내포된 글자의 인식에 대한 연구로 범포함에 따라 복잡한 제어 전략이 필요하게 되었다.

이러한 제어 전략은 관점에 따라 여러가지로 분류할 수 있지만, 보통은 상향식과 하향식 그리고 이 두 가지 방법을 혼용한 혼합형 제어 전략으로 구분된다. 상향식 제어 전략은 입력 패턴에 대하여 적당히 전처리를 한 후, 패턴으로 부터 여러 개의 특징을 추출하고 이러한 특징들을 가지고 패턴을 분류하게 된다. 이러한 방법은 패턴을 인식하는 자체보다도 세그멘테이션과 패턴의 특징을 추출하는데 어려움이 있다. 하향식 제어 전략은 인식하고자 하는 패턴의 모델을 그 모델의 계층적 규칙에 따라 여러 개의 부분으로 분리한 다음, 각 부분을 인식하여 하나의 패턴을 인식하게 된다. 이러한 제어 전략은 인식하고자 하는 물체에 대한 가정이 적절하다면 패턴을 빠르게 인식하게 되지만, 적절하지 못하다면 탐색하는데 시간 낭비를

를 초래한다. 하지만 이 제어 전략은 패턴의 특징을 추출하는데 있어 필요한 지역적인 부분에서만 행할 수 있고, 물체에 대한 가정을 사용하여 각 부분에 어떠한 특징이 존재할 것이라고 미리 예측하고 패턴의 특징을 추출할 수 있다는 장점이 있다. 혼합형 제어 전략은 저 단계 처리를 한 후, 기본적인 특징을 추출하고 이러한 특징을 가지고 하나의 모델을 설정한 다음, 이 모델에 하향식 제어 전략을 적용하여 패턴을 인식하게 된다[1].

일반적으로 패턴 인식은 인식하고자 하는 패턴의 특징을 그대로 유지하면서 처리해야 할 자료의 양을 줄이고자 전처리 과정에서 세신화를 하게 되지만, 변형이 심한 패턴이 세신화된 경우에는 인식하는데 어려움을 가중시킨다.

따라서 본 논문에서는 인식할 문자 패턴에 대하여 세신화 처리를 하지 않았으며, 인식할 패턴의 변형 정도, 그리고 한글 문자의 유형상의 특성을 고려하여 패턴 인식의 제어 전략으로 혼합형 제어 전략을 사용했다. 이 제어 전략 하에서 인식될 글자는 상향식 방법에 의하여 한글 문자의 기본적인 6가지 유형 중의 하나로 분류(제3장 2절)함으로써 글자의 탐색 공간을 축소 시켰으며, 그 다음에는 분류된 글자의 유형을 가지고 하향식 방법으로 음소를 인식하였다(제3장 3절). 음소를 인식함에 있어 모음을 자음보다 먼저 인식하였으며, 자음을 인식하는 때에는 수직적인 특징, 즉 crossing number와 run의 크기를 주로 사용 하였다. 인식하는 과정에서 바람직한 못한 결과가 발

생될 때에는 잘못이 발생한 지점으로 피이드백 함으로써 인식 알고리즘에 융통성을 부여 하였다(제3장 4절).

## II. 한글 문자의 구조 및 인쇄체 문자의 특징

24개의 기본 자모(14개의 자음과 10개의 모음)로 구성되는 한글은 자음과 모음을 포함하여 최소 2자, 최대 6자의 기본 자소가 규칙적으로 결합되어 하나의 글자가 생성된다. 생성 가능한 현대어 글자는 12,369자나 되고, 이러한 글자 중 일상 생활에 사용되고 있는 글자만 해도 약 1,500 자에 이른다 [2]. 그러나 한글은 모음의 형태와 중심의 존재 유무에 따라 그림 1과 같이 6가지 형태의 유형으로 분류할 수 있기 때문에 이러한 구조상의 특성을 이용하면 한글 인식하는 데 있어 탐색 공간을 축소시켜준다.

인쇄체는 기계로 인쇄된 글자이기 때문에 제한된 활자체로 되어 있으며 알아보기가 쉽고 불변적인 글자의 특징을 가지게 된다. 이러한 성질은 같은 글자라도 글씨를 쓰는 사람의 습관, 펜의 종류 등에 따라 다른 필기체에서는 찾아 볼 수 없는 특성으로 필기체를 인식하는 것보다 인쇄체를 인식하는 문제를 쉽게 만들어 준다. 이런 관점 하에서 인쇄체 한글 문자의 특징을 열거하면 다음과 같다.

- 첫째, 수직모음의 상단 끝점은 글자의 상단점과 같다.
- 둘째, 수평모음의 좌단 끝점은 글자의 좌단점과 같다.
- 셋째, 복모음의 구성요소 중 수직모음의 상단 끝점은 글자의 상단점과 같고, 수평모음의 좌단 끝점은 글자의 좌단점과 같다.
- 네째, 명조체는 고딕체에 비하여 음소 간에 연결이 많이 일어난다.

이러한 인쇄체 문자의 특성은 인식할 글자를 상향식 분석에 의한 6가지 형태로 분류하는데 사용된다.

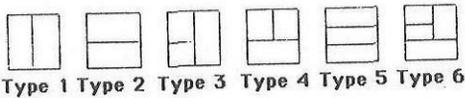


그림 1. 한글의 유형

## III. 인식 시스템

### 1. 시스템 개관

일반적으로 패턴 인식은 인식하고자 하는 패턴의 특징을 그대로 유지하면서 처리하는 자료의 양을 줄이기 위해서 전처리 과정에서 세선화 처리를 하게 된다[3]. 세선화 처리가 잘 된다면 세선화된 패턴으로부터 인식할 패턴의 기본적인 특성을 쉽게 추출할 수 있어 인식 과정이 비교적 간단해진다. 하지만 변형이 심한 패턴에 대해서 세선화 처리를 하면 변형된 패턴의 특성대로 세선화를 하기 때문에 변형이 더욱 심각해져 세선화된 패턴으로부터 패턴을 인식한다는 것은 매우 어려운 일이며 시간 낭비만 초래한다. 이런 경우에는 세선화 처리를 하

식 시스템의 구조는 그림 2와 같이, 크게 전처리 과정 단계, 상향식 단계, 하향식 단계, 피이드백 단계로 구성되어 있다. 우선 전처리 과정 단계에서는 광학 스캐너를 통해서 입력된 문자 영상에 대하여 일정한 크기의 문자로 표준화하여 실제로 인식할 문자를 만들어 준다[4]. 상향식 단계에서는 인식할 문자의 탐색 공간을 줄이고자 인식할 문자를 그림 1과 같은 글자의 유형으로 분류한다. 하향식 단계에서는 상향식 단계에서 분류된 글자의 유형을 가지고 글자를 인식하게 된다. 피이드백 단계에서는 하향식 단계를 처리하는 도중 에러가 발생한 경우에 에러의 발생 지점을 찾아내어 그 지점으로 다시 돌아가 앞서 실행한 과정을 반복하게 된다.

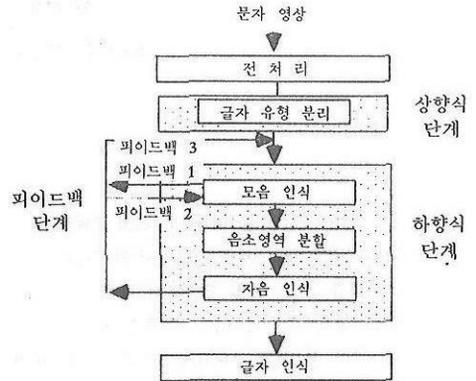


그림 2. 전체적인 시스템 구조

### 2. 상향식 단계

앞의 II장에서 열거한 인쇄체 한글 문자의 특성을 사용하여 한글 문자를 6가지의 유형으로 분류한다. 이 단계에서 사용되는 중요한 특징으로는 글자의 전체적인 경계를 나타내는 네 개의 끝점, 즉 상단점, 하단점, 좌단점, 우단점과 글자에서의 행의 합, 열의 합이다. 네 끝점을 찾기 위해서 상단점은 글자의 오른쪽 상단을 기준으로 하여 우측에서 좌측으로, 하단점은 글자의 왼쪽 하단을 기준으로 하여 좌측에서 우측으로, 좌단점은 글자의 왼쪽 상단을 기준으로 하여 윗쪽에서 아래쪽으로, 그리고 우단점은 글자의 오른쪽 하단을 기준으로 하여 아래쪽에서 윗쪽으로 스캔한다. 그리고 스캔하다 만난 1인 점의 이웃의 합이 임의의 임계치보다 크면 찾고자 하는 점으로 결정한다. 글자의 유형은 모음의 형태와 받침의 유무에 따라 결정되므로 모음의 형태와 받침이 존재할 조건을 간략히 서술하면 다음과 같다.

#### 가) 수직 모음 존재할 조건

- a) 상단점이 글자 너비의 1/2 우측에 존재하여야 한다.
- b) 기다란 수평 모음이 존재하지 않아야 한다.
- c) 상단점 및 상단점의 좌우 2 인접점 중에서 열의 합이 글자 높이의 1/2 보다 큰 점이 존재하여야 한다.

#### 나) 기다란 수평 모음 존재할 조건

- a) 좌단점이 글자 높이의 1/3 하단에 존재하여야 한다.
- b) 좌단점 및 좌단점의 상하 2 인접점 중에서 행의 합이 (글자 너비 -  $\alpha$ ) 보다 큰 점이 존재하여야 한다.

여기서  $\alpha$ 는 잡음을 고려한 상수이다

- 다) 짧은 수평 모음 존재할 조건
  - 가) 좌단점이 글자 높이의 1/3 하단에 존재하여야 한다
  - 나) 좌단점 및 좌단점의 상하 1 인접점 중에서 좌단점으로 부터 1인 화소가 연속적으로 10개 나타나는 점이 존재하여야 한다
- 라) 받침 존재할 조건
  - 가) 하단점이 글자 너비의 1/2 좌측에 존재하여야 한다
  - 나) 최소한 하단점은 좌단점보다 글자 높이의 1/4 만큼 아래에 존재하여야 한다
- 또는
  - 가) 하단점이 글자 너비의 1/2 우측에 존재하여야 한다
  - 나) 좌단점으로부터 획의 두께 만큼 떨어진 하단에서 획의 합이 글자 너비의 1/3 보다 큰 행이 존재하여야 한다

이러한 조건하에서 글자의 유형 분리 알고리즘은 다음과 같다

1. 만일 수직 모음이 존재하면 type1, type3, type4, type6의 certainty factor(CF)를 1씩 증가시킨다.
  - 1-1. 만일 수평 모음이 존재하면 type3, type6의 CF를 1씩, type4의 CF는 0.5 증가시킨다
  - 1-1-1. 만일 받침이 존재하면 type6의 CF를 받침이 존재치 않으면 type3의 CF를 1씩 증가시킨다
  - 1-2. 수평 모음이 존재치 않으면 type1, type4의 CF를 1씩 증가시킨다.
    - 1-2-1. 만일 받침이 존재하면 type4의 CF를 받침이 존재치 않으면 type1의 CF를 1씩 증가시킨다
2. 수직 모음이 존재치 않으면 type2와 type5의 CF를 1씩 증가 시킨다
  - 2-1. 만일 받침이 존재하면 type5의 CF를 받침이 존재치 않으면 type2의 CF를 1씩 증가시킨다

이 알고리즘을 수행 했을때, 가장 큰 값을 갖는 type이 글자의 유형으로 결정되며 나머지는 candidate type이 된다

### 3 하향식 단계

이 단계는 실제로 음소를 인식하는 단계인데, 앞의 상향식 단계에서 글자의 type을 결정해 주면 그 type에 해당하는 유형이 하나의 모델이 된다 모델이 결정되면 그 모델은 그림 1에 나타나 있는 바와 같이 다시 여러 개의 영역으로 분리되어 각 영역을 처리하게 된다 각 영역에서는 어떤 부류의 음소가 인식되기를 기대하면서 음소와 관련된 특징들을 추출하면서 해당 영역의 처리를 끝내게 된다 모든 영역의 처리가 끝나면 하나의 모델, 즉 하나의 글자를 인식하게 된다 이 단계는 모음 인식 단계, 음소 분할 단계, 초성 인식 단계, 종성 인식 단계로 구성된다

#### 1) 모음 인식

음소의 영역을 분리하기 위한 편의성 때문에 처음보다 모음을 먼저 인식하게 된다

에 각각 상하, 좌우로 가치를 첨가함에 따라 다른 형태의 모음이 되기때문에 가치의 갯수는 모음을 분리하는 중요한 특징이 된다 이외에도 모음 인식에 사용된 주요한 특징으로는 네개의 끝점과, 부분적인 행의 합과 열의 합 등이 있다. 모음은 구조가 단순하여 인식하기가 비교적 쉽지만, 자음과 붙어버린 수평모음, 복모음의 경우는 혼동의 소지를 많이 내포하고 있다. 모음의 수평, 수직 부분은 쉽게 찾을 수 있으므로 이에 대한 설명은 생략하고, 여기서는 모음의 또 다른 일부본인 상하, 좌우 가치의 선정 기준을 설명하겠다 가치 부분은 변형을 많이 받은 부분이면서 모음을 인식하는데 중요한 부분이기 때문에 가치 여부를 명확히 선정하여야 한다 수평 모음의 윗쪽과 아래쪽에 존재하는 가치는 각각 좌단점의 윗쪽과 좌단점으로부터 획의 두께 만큼 떨어진 아래쪽에서, 왼쪽에서 오른쪽으로 임의의 범위까지 스캔하여 0에서 1로 바뀌는 점에 대하여 그림 3의 (a)의 빗금친 부분에 해당하는 화소의 합을 구해 그 값이 5보다 크면 가치로 채택하게 된다 수직 모음의 우측에 존재하는 가치는 글자의 우단점의 왼쪽에서, 그리고 수직 모음의 좌측과 중간에 존재하는 가치는 모음의 수직 부분의 왼쪽에서, 위에서 아래로 임의의 범위까지 스캔하여 0에서 1로 바뀌는 점에 대하여 각각 그림 3의 (b), 그림 3의 (c)의 빗금친 부분에 해당하는 화소의 합을 구해 그 합이 각각 2와 3보다 크면 가치로 선정하게 된다 이 단계에서는 자음과 모음을 명확히 구분하여 인식할 수 없을 때에는 candidate를 함께 내 주도록 되어 있다

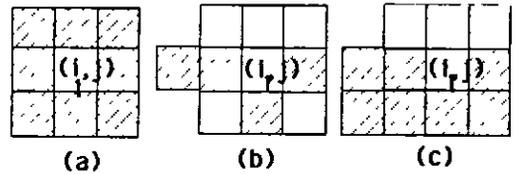


그림 3 가치 판정에 사용되는 틀의 배열

#### 2) 음소 분할

인식할 자음의 영역을 대략적으로 분리해 준다. 물론 자음의 영역 내에 모음의 부분이 포함될 수도 있지만, 포함된 부분은 자음 인식에 있어서 중요한 역할을 하지 않기 때문에 그리고 포함되는 부분은 모음의 가치 부분이기 때문에 별 문제가 되지 않는다

#### 3) 초성 인식

초성 인식에 사용되는 특징으로는 초성 자음 영역 내에서 수평, 수직 방향에 대한 crossing number, 1의 run의 크기 그리고 run의 상대적 위치를 사용하였다 run이란, 0 혹은 1이 연속적으로 나타나는 것을 말하는데, 크기에 따라 small, medium, large run으로 분류할 수 있다 초성 영역 너비(혹은 높이)의 1/3 보다 작으면 행(혹은 열)에 대한 small run이 되며, 초성 영역의 너비(혹은 높이)의 2/3 보다 크면 행(혹은 열)에 대한 large run이 되며, 그 사이의 크기를 가지면 행(혹은 열)에 대한 medium run이 된다 crossing number란, 행 혹은 열에서 1인 run의 빈도수를 말한다 그런데 처리할 글자의 경계 부분에서는 변형을 많이 받기 때문에 crossing

