

# 한글 음절의 유형 분석 정보에 의한 낱자 폰트의 설계

## (Design of Phoneme Fonts using an Analyzed Information of Hangul Syllable Forms)

李 桂 英\*, 金 奎 植\*\*, 李 相 範\*\*\*

(Gye Young Lee, Kyu Sik Kim, and Sang Burm Rhee)

### 要 約

본 논문에서는 한글 음절을 구성하고 있는 낱자의 유형을 분석하는 방법을 제안하였으며, 분석된 유형 정보를 이용하여 초성, 중성, 중성별 낱자 폰트를 설계하였다. 그리고, 설계된 473개의 낱자 폰트로써, 조합 가능한 모든 한글 음절의 출력이 가능한 알고리즘을 제시하였다. 실험 결과, 본 연구의 낱자 유형 정보 분석 방법과 한글 출력 알고리즘은 경계성 있는 한글 폰트의 설계에 유용하게 사용될 수 있음을 보였다.

### Abstract

This paper proposed an analysis method of common form for phonemes which is able to construct a Hangul syllable and designed the Choseong, Jungseong, and Jongseong phoneme fonts based on analyzed information. Also, It presents the algorithm which is able to construct the output of all Hangul syllables using 473 phoneme fonts. Through the experiment, an analyzed information and output algorithm could be applied to the design of Hangul fonts, effectively.

### I. 서 론

한글은 우리의 말인 한국어를 표기하여 주는 언어로서, 초성, 중성, 중성이 규칙적으로 조합되는 기본 원리를 갖고 있으며, 음성 기호로서의 역할과 소리마디의 구분이 가능하게 구성되어 있다<sup>[1],[2]</sup>. 그리고, 현재 사용되고 있는 한글의 글자꼴은 대부분 하나의 음절을 단위로 하여 도안되고 있으며, 하나의 음절이 하나의

코드에 대응된 형태를 취하고 있다.

따라서, 한글 패턴의 인식시<sup>[3],[6]</sup>에는 음절에 대한 자소의 분리 과정이 항상 필요하며, 신경 회로망을 이용한 패턴 인식시<sup>[10],[13]</sup>에도 음절이 처리 단위가 된다. 그리고, 한글 패턴의 설계시에는 원하는 모든 한글 음절을 대상으로 하여야 하기 때문에 한글의 소리 마디를 셋으로 나누는 삼분법의 과학성이 무시되고 있는 실정이다<sup>[7],[9]</sup>.

한글의 삼분법 특성은 소리 마디를 성모와 운모로 나누는 중국어의 이분법(반절법)에 대비되는 것으로, 음성 기호의 능력을 가지면서도 우리의 말이라는 자연 언어에 바탕을 두고 낱자의 수를 크게 줄여 주는 데 있다. 현대 국어의 낱자에 대해서 삼분법을 사용하는 경우에는 67개의 낱자가 필요하지만 이분법을 사용하게 되면 567 낱자가 필요하며, 현행의 표준 코드에 따르게 되면 11,172자가 필요하게 된다.

\*正會員, 東國大學校 電子計算學科  
(Dept. of Computer Sci., Dongguk Univ.)

\*\*正會員, 東國大學校 情報管理學科  
(Dept. of Information Management, Dongguk Univ.)

\*\*\*正會員, 檀國大學校 電子工學科  
(Dept. of Elec. Eng., Dankook Univ.)

接受日字: 1992年 1月 10日

따라서, 한글 글자꼴의 설계에 있어서 한글의 조합 특성을 최대한 이용하게 되면, 적은 낱자의 생성으로 원하는 모든 한글 음절을 출력할 수 있게 된다. 본 연구는 한글 낱자의 음절내 위치, 면적 정보를 바탕으로 각 낱자의 극점 유형 밀집도를 분석하여 모든 낱자를 유사 유형으로 그룹화한 후, 가능한 적은 낱자 설계를 하고자 하였다.

그리고, 설계된 낱자를 하나의 음절로 형성하여 출력할 수 있는 알고리즘을 구현함으로써, 낱자 유형 분류 정보에 의한 한글 출력 알고리즘의 유용성을 보이 고자 하였다.

II. 한글의 음절 구조 분석

C를 초성, J를 중성, V1을 수직 모음, V2를 수평 모 음이라 할 때, 한글 낱자의 구성에 의한 음절내 초성과 중성의 모양, 위치, 크기는 중성의 모양에 따라 크게 변화된다. 즉, 받침이 없는 유형과 받침이 있는 유형에 대해서 수직 모음이 존재하는가, 수평 모음이 존재하 는가, 아니면 복모음이 있는 가에 따라서 변화하게 된 다. 중성꼴 역시 초성과 중성의 영향을 받아 음절내 위 치가 변하게 되며, 그 크기와 모양 역시 다양하게 나타 난다.

그리고, 중성은 초성과 중성의 영향을 받아 음절내 위치와 모양이 달라지게 된다. 따라서, 똑 같은 초성이 라 하더라도 결합된 다른 낱자의 유형에 따라 그 모양 과 위치, 크기는 각각 달라지며, 중성과 중성 낱자의 경우에도 마찬가지로 현상이 나타난다.

본 연구에서는 한글 음절을 구성하는 각 낱자의 구 조 분석을 위하여, 이미지 스캐너를 통하여 입력된 한 글 패턴에 대해서, 그림 1과 같은 음절의 낱자별 분리 처리 및 분리된 낱자의 위치 정보 추출을 행하였다.

- 단계 1 : 음절별 화일 생성
- 단계 2 : 잡음 제거 과정
- 단계 3 : 최적 음절의 분리 과정
- 단계 4 : 초, 중, 중성의 낱자 분리 과정
- 단계 5 : 분리된 낱자의 위치 정보 추출 과정

그림 1. 음절의 낱자별 분리 처리  
Fig. 1. The phonemes separation of syllable.

이미지 스캐너로 입력된 자료는 이미지 화일로 저장 되며(단계1), 입력 오류나 데이터의 파손, 잡음 등은 포인트 블러쉬를 이용하여 제거, 정정하였다(단계2). 그리고, 입력 자료의 양을 최적화하기 위하여 최적 음

절의 분리 과정(단계3)을 두었다.

즉, 이미지 리더로 입력된 한글 음절을 화면 전체 단 위로 저장시킨 후, 이미지 화일 \*.pcx를 프로그램 조 작이 가능한 \*.pcu 화일로 생성하여, 최적 음절의 극 좌점(x1), 극우점(x2), 극상점(y1), 극하점(y2)을 찾 아 내어, \*.pic 화일을 생성하였다.

단계 4와 단계 5는 \*.pic로 구성된 이미지 화일에 대 해 낱자 분리를 행하고, 분리된 각 낱자의 위치 정보 (x1, x2, y1, y2)를 추출해 내는 과정이다. 이상의 과 정을 통하여 추출된 음절내 초성 낱자의 위치 정보와 면적 중에서 초성 'ㄱ'에 대한 정보를 표 1에 보였다.

표 1. 초성 'ㄱ'의 정보표  
Table 1. The table of 'Choseong ㄱ'.

결합 중성	초성 위치				초성 면적	결합 중성	초성 위치				초성 면적
	x1	x2	x3	x4			x1	x2	x3	x4	
ㄱ	1	11	27	25	364	ㄲ	9	1	55	20	874
내	1	11	27	26	390	ㄳ	9	1	55	20	874
ㄴ	1	11	35	25	476	ㅋ	1	12	37	43	1116
ㄷ	1	11	36	25	490	ㆁ	1	12	37	43	1116
ㄹ	1	14	38	28	518	ㅏ	1	12	39	43	1178
ㅑ	1	11	37	29	648	ㅓ	1	11	41	41	1200
ㅕ	1	11	38	29	666	ㅗ	1	10	41	41	1240
ㅗ	1	12	29	40	784	ㅛ	9	1	60	26	1275
ㅛ	1	13	29	41	784	ㅜ	9	1	61	26	1300
ㅜ	1	14	33	39	800	ㅡ	9	1	59	29	1400
ㅡ	1	13	31	40	810						

표 1은 음절내 초성 위치 정보에 의해 산출된 초성 면적에 의해서 분류된 표로서, 초성의 크기가 가장 작 은 것은 중성 'ㄱ'과 결합될 때이며, 가장 큰 경우는 중 성 'ㅡ'와 결합될 때이다. 그리고, 중성 'ㄲ, ㄳ, ㄴ, ㆁ, ㅏ, ㅓ'와 같은 수평 모음(V2)이 결합되는 경우는 다 른 중성의 결합시와는 달리 해당 초성의 첫번째 유효 화소의 위치가 확연히 달라짐을 알 수 있다.

따라서, 표 1의 위치 정보를 이용한 초성 유형의 분 류는 크게 'ㄱ~ㅗ', 'ㄲ~ㄳ', 'ㅋ~ㅣ', 'ㅏ~ㅡ'의 4 개 집단으로 나눌 수 있다. 그러나, 'ㄱ~ㅗ' 그룹은 수 직 복모음인가, 수직/수평 복모음인가에 따라, 결합 되는 초성의 유형을 더욱 세분화할 필요가 있기 때문 에, 초성의 면적과 극점 좌표의 밀집도를 이용하여 그 유사 그룹을 분류하였다.

그림 2는 표 1에 보인 초성의 위치 정보를 극점의 분 포 그래프로 보인 것이다. 본 연구에서는 그림 2의 결 과로서 초성의 유형을 초성 면적의 분류(sort) 정보와

극점의 밀집 유형에 따라서 분류하였다.

즉, “ㄱ, ㅋ, ㆁ, ㆁ, ㆁ”, “ㅋ, ㅋ, ㆁ, ㆁ”, “ㄱ, ㆁ, ㆁ”, “ㆁ, ㆁ”, “ㆁ, ㆁ, ㆁ”, “ㆁ, ㆁ”, “ㆁ, ㆁ”에 대해서 각각 1에서 7의 초성 유형 코드를 부여하였다. 나머지 초성의 유형 역시 그림 2와 유사한 그래프가 구해졌으며 따라서, 본 연구에서는 모든 초성은 중성의 유무에 따라 각각 7개씩 모두 14가지로 분류하여 그 모양꼴을 설계하였다.

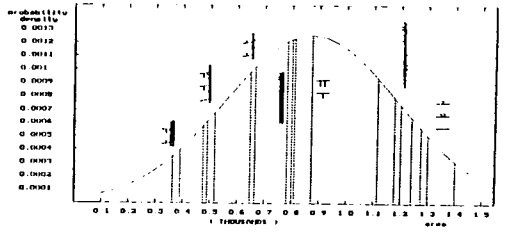


그림 3. 초성 날자(ㄱ)의 가우스 확률 밀도 함수(PDF)  
Fig. 3. The PDF of 'choseng ㄱ'.

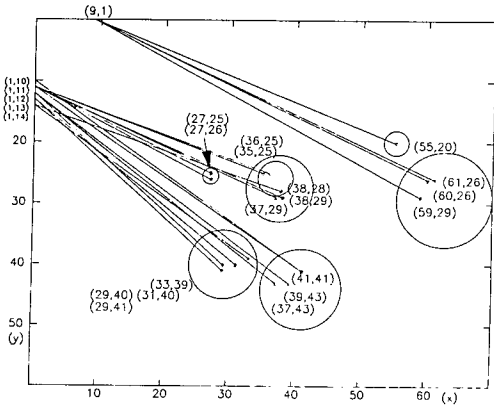


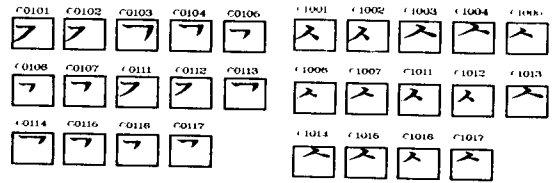
그림 2. 초성 날자(ㄱ)의 극점 분포 그래프  
Fig. 2. The Graph of 'choseng ㄱ'.

그림 2에서 “ㄱ~ㄴ” 그룹과 “ㄱ~ㆁ” 그룹 역시 더욱 세분화할 수 있으나, 본 연구에서는 유형 분석의 한 방안과 유형 결합 알고리즘의 프로토타입(prototype)만을 제시하기로 한다. 따라서, 이상의 결과로서 “ㄱ, ㅋ, ㆁ, ㆁ, ㆁ”, “ㅋ, ㅋ, ㆁ, ㆁ”, “ㄱ, ㆁ, ㆁ”, “ㆁ, ㆁ”, “ㆁ, ㆁ, ㆁ”, “ㆁ, ㆁ”, “ㆁ, ㆁ”와 결합되는 모든 초성에 대해서, 중성의 유무에 따라 각각 7개씩 모두 14가지로 분류하여 그 모양꼴을 설계하였다.

이와 같은 분류의 타당성을 확인하기 위한 한 방안으로, 본 연구에서는 날자의 면적을 random varivale로 하여 PDF를 구하였으며, 그림 3에 나타난 바와 같이 초성 면적에 대한 각 날자의 PDF 길이는 앞서 분류된 것 처럼 그룹화가 가능하다.

그림 4는 초성 ‘ㄱ’과 ‘ㅅ’의 모양꼴을 보인 것이다.

표 2는 중성 ‘ㄱ’(중성 그룹코드 1)’와 ‘ㅅ’(중성 그룹코드 2)’의 위치 정보와 면적 정보를 보인 것이다. 그림 5는 표 2에 보인 중성의 위치 정보를 극점의 분포 그래프로 보인 것이다. 본 연구에서는 그림 5의 결과로서 중성의 유형을 극점의 밀집 유형에 따라 6개의 유사 그룹으로 분류하였다.



(a) 초성 ‘ㄱ’ (b) 초성 ‘ㅅ’

그림 4. 초성 모양꼴의 예  
Fig. 4. Example of choseng font.

표 2. 중성 ‘ㄱ, ㅅ’의 정보표  
Table 2. The table of 'Jungseong ㄱ, ㅅ'.

중성	쌍초성	음절꼴	중성 위치				면적 정보	음절꼴	중성 위치				면적 정보
			x1	y1	x2	y2			x1	y1	x2	y2	
무	유	따	43	1	67	53	1248	매	33	1	58	53	1300
무	무	가	41	1	67	53	1352	매	30	1	58	53	1456
ㄴㅅ	유	뎨	43	1	67	40	936	뎨	33	1	58	40	975
		뎨	43	1	67	40	936	뎨	33	1	58	40	975
ㄴㅅ	무	간	41	1	67	40	1014	맨	30	1	58	40	1092
		갓	41	1	67	40	1014	맨	30	1	58	40	1092
ㄴㅅ	유	따	43	1	67	30	696	뎨	33	1	58	30	725
		따	43	1	67	30	696	뎨	33	1	58	30	725
ㄴㅅ	무	갓	41	1	67	30	754	맨	30	1	58	30	812
		갓	41	1	67	30	754	맨	30	1	58	30	812

\* 결합되는 중성(ㄴ, ㅅ) 형태와 (ㄴ) 형태로 구분됨

중성 ‘ㄱ, ㅋ, ㆁ, ㆁ, ㆁ’는 그림 5의 결과로서 중성 유무, 쌍초성 유무, 중성이 있을 때 ‘ㄴ, ㅅ’의 결합 유무에 영향을 받음을 인지할 수 있다. 즉, 중성이 없을

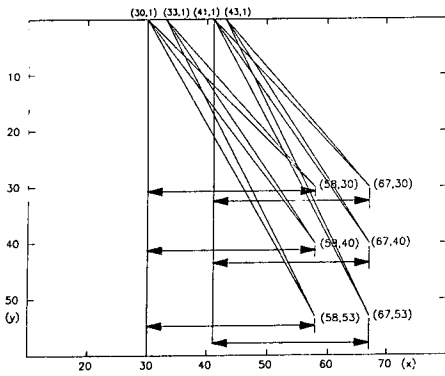


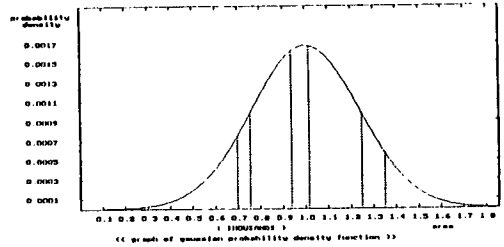
그림 5. 중성 날자(ㅏ, ㅑ)의 극점 분포 그래프  
Fig. 5. The Graph of 'Jungseong ㅏ, ㅑ'.

때는 쌍초성(ㅓ, ㅕ, ㅖ, ㅗ, ㅛ)의 유무에 따라 중성의 위치와 크기가 변하게 되며, 중성이 있을 때는 중성 'ㄴ, ㄷ'의 유무에 따라 수직 모음의 길이가 달라지며, 이 경우에도 쌍초성이 결합되면 모음의 위치가 우측으로 이동되어 그 면적 정보가 작아짐을 알 수 있다. 따라서, 중성 'ㅏ, ㅑ, ㅓ, ㅕ, ㅖ' 그룹은 다음과 같이 그 유형이 분류된다.

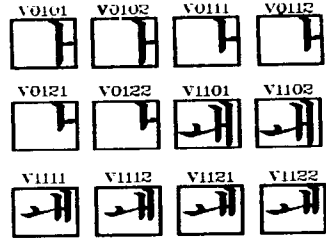
- ① 중성이 없고, 쌍초성이 결합된 경우  
(예: 까, 따, 빠 등)
- ② 중성이 없고, 쌍초성이 아닌 경우  
(예: 가, 나, 다 등)
- ③ 중성이 'ㄴ, ㄷ'이고, 쌍초성인 경우  
(예: 간, 단, 뺏, 짖 등)
- ④ 중성이 'ㄴ, ㄷ'이고, 쌍초성이 아닌 경우  
(예: 간, 난, 단 등)
- ⑤ 중성이 'ㄴ, ㄷ'이 아니고, 쌍초성인 경우  
(예: 각, 땀, 빨 등)
- ⑥ 중성이 'ㄴ, ㄷ'이 아니고, 쌍초성이 아닌 경우  
(예: 각, 낙, 닥 등)

이 이외의 중성 그룹에 대한 분류 결과는 제 3장의 표 5에서 정리하였다. 그림 6의 (a)는 중성 그룹 코드 1에 대한 PDF를 보인 것이며, (b)는 설계된 중성 'ㅏ'와 'ㅑ'의 모양꼴을 보인 것이다.

그림 7은 중성(ㅏ, ㅑ)에 대한 위치 및 면적 정보와 극점 그래프이다. 그림 7과 같은 중성의 정보 추출 실험을 통하여, 중성의 위치와 면적은 중성의 영향을 가장 많이 받을 수 있으며, 중성의 유형은 'ㅓ, ㅕ, ㅖ, ㅏ'와 같은 수평 모음, 'ㅓ, ㅑ, ㅓ' 등의 'ㅏ'형 모음,



(a) 중성 그룹 코드 1의 PDF 그래프

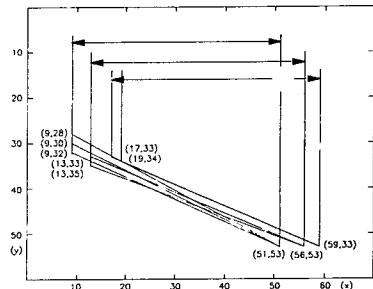


(b) 중성 모양꼴의 예

그림 6. 중성 모양꼴의 예  
Fig. 6. Example of jungseong font.

중성꼴	음절꼴	x1	y1	x2	y2	면적	음절꼴	x1	y1	x2	y2	면적
'ㅡ'형	극	9	30	51	53	996	곤	9	30	55	48	828
	국	9	32	51	53	882	군	9	40	55	48	828
	극	9	28	51	53	1050	근	9	28	55	48	920
'ㅏ'형	각	13	33	56	53	860	간	15	35	58	48	559
	곽	13	35	56	53	774	관	15	37	58	48	473
'ㅑ'형	격	17	33	59	53	840	견	17	35	60	48	559
	기	17	33	59	53	840	긴	17	35	60	48	559
	력	19	34	59	53	740	퀸	17	36	60	48	504

(a) 위치 및 면적 정보표



(b) 극점 분포 그래프

그림 7. 중성 날자의 정보와 극점 분포 그래프  
Fig. 7. Graph & information table of Jongseong 'ㅏ, ㅑ'.

‘ㄱ, ㅋ, ㆁ’ 등의 ‘ㄱ’형 모음과 결합될 때의 세가지로 분류하였다.

이상의 실험을 통하여 음절내 각 낱자의 위치 정보를 분석한 결과, 거의 모든 초성, 중성, 중성끝이 똑같은 위치에 놓이는 경우가 없었다. 그러나 본 연구에서는 이들 위치 정보와 면적 정보, 극점의 밀집 유형 정보를 최대한 이용하여 초성, 중성, 중성별 유형의 분류를 시도하였다.

그림 8은 중성 ㄱ, ㄴ, ㅂ, ㅅ의 모양꼴을 보인 것이다.

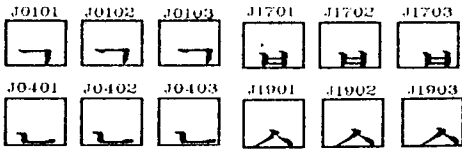


그림 8. 중성 모양꼴의 예  
Fig. 8. Example of Jongseong font.

### Ⅲ. 낱자 유형의 분류와 유형 정보를 이용한 모아쓰기 알고리즘

#### 1. 글자꼴 유형 정보의 분류

#### 1) 초성의 유형 정보 분류

한글 음절의 초성꼴은 표1, 그림 2의 결과로서 그림 9와 같이 분류되었다. 그림 9의 노드 A는 받침의 유무를 판단하며, 노드 B와 C는 받침 유무에 따른 중성꼴 정보를 판단하게 된다. 노드 B와 노드 C는 실험에 의해서 얻어진 표 3의 7개의 중성 정보 그룹 코드에 의해서 각각 7개의 초성 유형 정보로 분류된다.

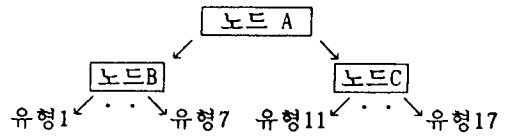


그림 9. 초성 유형 분류를 위한 판별 트리  
Fig. 9. Decision tree for assortment of Choseong.

즉, 초성은 받침의 유무에 따라서 그리고, 중성의 결합 형태에 따라서 분류될 수 있으며, 본 연구에서는 표 1과 그림 2의 결과에 따라 표 3과 같이 14개의 유형으로 분류하였다.

#### 2) 중성의 유형 정보 분류

중성꼴은 초성과 중성의 영향을 받아 그 유형이 분류된다. 그림 10은 중성꼴 결정을 위한 판별 트리이다.

표 3. 초성의 유형 분류표  
Table 3. Assortment table of Choseong.

중성 유무	초성유형 코드	결합되는 중성꼴	중성꼴 특징	중성그룹 코드	중성 순서 코드
없음	01	ㅏ, ㅑ, ㅓ, ㅕ, ㅗ	(V1), 단모음1	1	V01, V03, V05, V07, V21
	02	ㅓ, ㅕ, ㅗ, ㅛ	(V1), 복모음1	2	V02, V04, V06, V08
	03	ㅓ, ㅕ, ㅗ	(V2), 단모음2	3	V09, V13, V19
	04	ㅓ, ㅕ	(V2), 단모음3	4	V14, V18
	05	ㅓ, ㅕ	(V1, V2), 복모음2	5	V10, V12
	06	ㅓ, ㅕ	(V1, V2), 복모음3	6	V11, 16
	07	ㅓ, ㅕ, ㅗ	(V1, V2), 복모음4	7	V15, V17, V20
있음	11	ㅏ, ㅑ, ㅓ, ㅕ, ㅗ	단모음1	1	V01, V03, V05, V07, V21
	12	ㅓ, ㅕ, ㅗ, ㅛ	복모음1	2	V02, V04, V06, V08
	13	ㅓ, ㅕ, ㅗ	단모음2	3	V09, V13, V19
	14	ㅓ, ㅕ	단모음3	4	V14, V18
	15	ㅓ, ㅕ	복모음2	5	V10, V12
	16	ㅓ, ㅕ	복모음3	6	V11, 16
	17	ㅓ, ㅕ, ㅗ	복모음4	7	V15, V17, V20

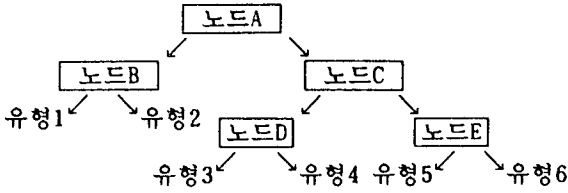


그림 10. 중성꼴 결정을 위한 판별 트리  
Fig 10. Decision tree for assortment of Jungseong.

노드 A는 중성의 유무를 판단한다. 노드 B는 중성이 없을때 초성의 유형을 판별하는 것으로, 표2에서 분류된 중성 그룹 코드 1과 2는 쌍초성(ㅍ, ㅑ, ㅓ, ㅕ)의 유무에 따라서 2개의 유형으로 분류된다. 노드 C는 중성이 있을때의 중성의 유형 조건으로 중성 그룹 코드 1, 3, 5, 7이면 노드 D로, 중성 그룹 코드 2, 4, 6이면 노드 E로 이동된다. 표 4는 노드 C 처리를 위하여, 중성 유형 코드와 비교되는 중성 집단을 보인 것이다.

표 4. 노드 C 처리를 위한 테이블  
Table 4. Processing table for node C.

중성 그룹 코드	중 성 순 서 코드	중 성 비교유형	비 고
1	V01, V03, V05, V21	b, d	중성 ㄴ, ㄷ의 유무에 영향을 받음 (b: 중성 ㄴ, ㄷ) (d: ㄴ, ㄷ이 제외된 중성)
3	V09, V13, V19		
5	V10, V12		
7	V15, V17, V20		
2	V02, V04, V06, V08	a, c	중성 ㄴ의 유무에 영향을 받음 (a: 중성 ㄴ) (c: ㄴ이 제외된 중성)
4	V14, V18		
6	V11, V16		

표 5. 초성의 유형 분류 정보  
Table 5. Assortment information of Choseong forms.

초 성 유형	초성 유형 분류 코드	초성 유형 대표 코드	관련된 중성의 집단 분류 코드
쌍초성	1	1	1, 2
쌍초성 제외	2	2	1, 2
ㄱ, ㅋ, ㅅ, ㅆ	3	1	3, 5, 6
ㄱ, ㅋ, ㅅ, ㅆ 제외	4	2	3, 5, 6
ㅎ	5	1	4, 7
ㅎ 제외	6	2	4, 7

표 6. 중성꼴 결정 테이블  
Table 6. Decision table of Jungseong forms.

중성 그룹 코드	중 성 순 서 코드	받침 유무 코드	초성 유형 분류 코드	중성 코드 유형(대표)
1	V01, V03, V05 V07, V21 (ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ)	0	1	o(0)
		0	2	o(0)
		1	1	b(1)
		1	2	b(1)
		1	1	d(2)
		1	2	d(2)
2	V02, V04, V06, V08 (ㅑ, ㅓ, ㅕ, ㅗ, ㅛ)	0	1	o(0)
		0	2	o(0)
		1	1	a(1)
		1	2	a(1)
		1	1	c(2)
		1	2	c(2)
3	V09, V13, V19 (ㅏ, ㅑ, ㅓ)	0	3	o(0)
		0	4	o(0)
		1	3	b(1)
		1	4	b(1)
		1	3, 4	d(2)
		1	4	d(2)
4	V14, V18 (ㅏ, ㅑ)	0	5	o(0)
		0	6	o(0)
		1	5	a(1)
		1	6	a(1)
		1	5	c(2)
		1	6	c(2)
5	V10, V12 (ㅑ, ㅓ)	0	3	o(0)
		0	4	o(0)
		1	3	b(1)
		1	4	b(1)
		1	3	d(2)
		1	4	d(2)
6	V11, V16 (ㅑ, ㅓ)	0	3	o(0)
		0	4	o(0)
		1	3	a(1)
		1	4	a(1)
		1	3	c(2)
		1	4	c(2)
7	V15, V17, V20 (ㅑ, ㅓ, ㅓ)	0	5	o(0)
		0	6	o(0)
		1	5	b(1)
		1	6	b(1)
		1	5	d(2)
		1	6	d(2)

그림 10의 노드 D는 중성의 유형에 따라서 b, d의 두 유형으로 분류한다. 즉, 중성이 'ㄴ, ㄷ'이면 b형, 'ㄴ, ㄷ'이 아니면 d형으로 분류된다. 노드 E는 중성 'ㄴ'의 영향을 받으며, 중성 'ㄴ'이 존재하면 a형, 존재하지 않으면 c형으로 분류된다. 표 4의 중성 그룹 코드는 초성꼴 결정시에도 정보를 제공하며, 중성꼴 결정의 경우에는 검색 길이를 짧게 해주는 잇점이 있다.

그리고, 중성꼴 결정시에도 그 유형 정보가 유효하게 사용된다. 중성꼴의 유형 결정은 초성도 영향을 끼

치게 되며, 그림 5에 의하여 6가지 관련 유형을 산출하였다. 표 5는 중성꼴 결정에 영향을 미치는 초성의 유형 정보이며, 그림 10의 중성꼴 결정 판별 트리와 표 4, 5의 결과로서 모든 중성꼴을 결정하는 표 6이 작성되었다.

표 6의 초성 유형 분류 코드 항목과 중성 유형 코드 항목에 대하여 각각 대표 코드를 부여하고, 중성 유형 코드 항목을 별도의 정보표로 갖게 하면, 모든 중성 집단 코드에 대해서 표 7과 같이 요약된 중성 집단 비교 테이블의 작성이 가능하다.

표 7. 중성 집단 비교 테이블

Table 7. Decision table of Jungseong group.

중성유무	초성유형 대표코드	중성유형 대표코드	중성유형 코드
0	1	0	01
0	2	0	02
1	1	1	11
1	2	1	12
1	1	2	21
1	2	2	22

3) 중성의 유형 분류 정보

초성꼴이 중성과 중성 유형 코드에 의해서 결정된 후, 중성꼴이 초성과 중성 유형에 따라 결정되면, 중성꼴의 결정은 중성이 어떠한 형태로 결정되었는가에 가장 큰 영향을 받는다. 그림 7을 통하여 중성의 유형은 크게 세가지로 분류하였으며, 표 8은 중성꼴 결정을 위한 테이블을 보인 것이다. 더욱 세분하면 중앙, 우측, 극우측의 위치에 대하여 상, 하의 이동을 고려한 6가지 이상으로 분류할 수도 있다.

표 8. 중성꼴 결정 테이블

Table 8. Decision table of Jongseong form.

중성코드 집단	중성유형 코드	위치	예
V09, V13, V14, V18, V19	01	중앙	곡, 교, 국, 규, 극
V01, V03, V10,	02	우측	각, 약, 광
기타	03	극우측	객, 썩, 깃

2. 날자 유형 폰트를 이용한 모아쓰기 알고리즘

표 1~8을 이용하여 조합 가능한 모든 한글 음절을

출력할 수 있는 알고리즘을 제시한다. 그림 11은 날자 유형 폰트를 이용한 모아쓰기 알고리즘의 전체 모듈 구성을 보인 것이다.

단계 1 : 전처리 과정

단계 2 : 초, 중, 중성의 유형 정보 인지 과정

단계 3 : 모아쓰기 출력과정

그림 11. 모아쓰기 알고리즘의 모듈 구성

Fig. 11. Module constructions of output algorithm.

날자 단위로 설계된 글자꼴을 모아쓰기 출력하기 위해서는 입력된 음절내 날자 코드의 결합 가능 여부의 확인이 필요하다. 따라서, 단계 1의 전처리 과정은 입력된 코드의 결합 방법에 대한 오류를 검사하고, 음절이 형성되지 않을 경우에는 오류 메시지를 출력한다.

음절이 형성되는 경우에는 받침 유무를 판단하여 초, 중, 중성의 최종 정보 코드를 산출하는 다음 단계를 수행한다. 단계 2는 초, 중, 중성의 결합 형태에 따른 각 날자의 유형 정보를 결정해 주는 과정으로 초성은 중성과 중성의 유형에 따라서, 중성은 초성과 중성에 따라서 그리고, 중성은 중성의 유형에 따라서 그 모양꼴 정보가 추출된다.

날자별 모양꼴 정보를 위해 형성되는 코드 구조는 Pxxyy이며, xx는 날자 순서 코드로서, 초성은 01~19, 중성은 01~21, 중성은 00~27(00::무중성)을 갖는다. yy는 날자 유형 코드로서 초성은 01~07(중성이 없을 때)과 11~17(중성이 있을 때), 중성은 01, 02, 11, 12, 21, 22를 가지며, 중성은 01~03을 갖는다. P는 날자 분류 코드(초성: C, 중성: V, 중성: J)이다. 그림 12는 초성꼴의 유형을 결정하기 위한 알고리즘이다.

단계1: IF 중성 그룹 코드 = /\* 표 3의 적용 부분 \*/

1 {t, f, j, k, l}: 초성 유형 코드 = 1;

2 {h, h, k, k}: 초성 유형 코드 = 2;

3 {g, gg, -}: 초성 유형 코드 = 3;

4 {r, rr}: 초성 유형 코드 = 4;

5 {n, nn}: 초성 유형 코드 = 5;

6 {ae, ae}: 초성 유형 코드 = 6;

7 {c, c, -}: 초성 유형 코드 = 7;

단계2: IF 중성 = 있음 /\* 중성 유무의 적용 부분 \*/

THEN 초성 유형 코드 = 초성 유형 코드 + 10;

단계3: RETURN [초성 유형 코드];

그림 12. 초성꼴 결정 루틴

Fig. 12. Decision routine of Choseong code.

그림 12와 같이 초성의 유형 코드는 중성 그룹의 유형에 따라서 1차적으로 분류되며, 최종적으로는 중성 유무에 따라서 유형 코드가 추출된다. 중성풀 결정을 위한 코드는 6가지(01, 02, 11, 12, 21, 22)의 유형 정보를 갖는다.

그림 13은 중성풀 결정을 위한 알고리즘으로, V-INDEX는 표 7에 해당하는 테이블을 갖는다. 중성 유형 코드는 초성의 유형(단계 1)에 따라서 분류되며, 중성 유무(단계 2) 및 중성이 있을 때 중성 ㄴ, ㅅ 또는 ㄴ과의 결합 유무(단계 3)에 따라 결정된다.

단계1: IF 중성 그룹 코드 = /\* 표 5의 적용 부분 \*/

1, 2: IF 초성 코드 = 쌍초성(ㄱ, ㄷ, ㅃ, ㅆ, ㅈ)

THEN 초성유형 대표코드 = 1:

ELSE 초성유형 대표코드 = 2:

3, 5, 6: IF 초성 코드 = “ㄱ, ㅍ, ㅅ, ㅈ”

THEN 초성유형 대표코드 = 1:

ELSE 초성유형 대표코드 = 2:

4, 7: IF 초성 코드 = “ㅎ”

THEN 초성유형 대표코드 = 1:

ELSE 초성유형 대표코드 = 2:

단계2: IF 중성 = 있음 THEN GOTO 단계3:

중성 유무 코드, 중성분류 대표코드 = 0: GOTO 단계 4:

단계3: 중성 유무 코드 = 1: /\* 표 4의 적용 부분 \*/

IF 중성 그룹 코드 =

1, 3, 5, 7: IF 중성 코드 = “ㄴ, ㅅ”

THEN 중성분류 대표코드 = 1:

ELSE 중성분류 대표코드 = 2:

2, 4, 6: IF 중성 코드 = “ㄴ”

THEN 중성분류 대표코드 = 1:

ELSE 중성분류 대표코드 = 2:

단계4: 중성 유형 코드 = /\* 표 7의 적용 부분 \*/

V-INDEX[중성 유무 코드, 초성유형 대표코드, 중성분류 대표코드]:

단계5: RETURN[중성 유형 코드]:

그림 13. 중성풀 결정 루틴

Fig. 13. Decision routine of Jungseong code.

중성풀의 결정은 표 8이 사용되며, 그림 14는 중성풀 결정 알고리즘이다. 그림 11의 단계 4는 그림 12~14에서 결정된 초, 중, 중성의 유형 코드를 이용하여, 입력된 단어나 어절을 합성한 후 출력시키게 된다. 따라서, 본 연구의 유형 분석 정보를 이용하여 설계된 낱자를 위한 한글 모아쓰기 오토마타는 그림 12, 13, 14에 해당된다.

단계1: IF 중성 = 있음 THEN GOTO 단계 2:

중성 유형 코드 = 00: GOTO 단계 3:

단계2: IF 중성 순서 코드 = /\* 표 8의 적용 부분 \*/

9, 13, 14, 18, 19(ㄴ, ㄷ, ㅍ, ㅈ, ㅊ): 중성 유형 코드 = 01:

1, 3, 10(ㄱ, ㅋ, ㆁ): 중성 유형 코드 = 02:

2, 4, 5, 6, 7, 8, 11, 12, 15, 16, 17, 20, 21: 중성 유형 코드 = 03:

단계3: RETURN[중성 유형 코드]:

그림 14. 중성풀 결정 루틴

Fig. 14. Decision routine of Jongseong code.

#### IV. 실험 및 검토

본 연구에서 제시된 음절 구조의 분석, 낱자 유형의 분류와 유형별로 설계된 폰트의 모아쓰기 알고리즘은 IBM PC /386 상에서 C언어로 구현되었다. 실험용 문자로는 익숙해져 있는 종래의 글자꼴을 벗어나지 않기 위하여, 인쇄소에서 사용되는 서적용 활자체를 사용하여 얻어진 문자 단위의 영상에 대해서 음절 구조의 분석과 음절내 낱자의 유형 분류를 행하였다.

실험에 사용된 한글 음절수는 한글 빈도순 상위 700자와 실험에 필요한 346자를 합하여 1046자이며, 실험용 문자 크기는 화면 출력 화소 비율을 1:1.3으로 하여 65×50으로 하였다.

음절의 낱자별 분리 처리 과정(그림1)에서, 낱자 분리를 위한 \*·pic 이미지 화일은 크게 두가지 형태로 나타날 수 있다. 즉, 각각의 낱자가 접촉되지 않은 형태와 낱자의 접촉이 이루어진 형태이다. 낱자가 서로 접촉되지 않은 경우에는 그림 1의 단계 3과 동일한 처리가 되어 각 낱자의 좌표가 추출되며, 낱자와 낱자가 연결된 경우에는 낱자 분리 알고리즘이 사용되어야 한다. 낱자 분리에 관한 많은 알고리즘이 연구되고 있으나, 이 과정은 본 연구의 목적에 크게 영향을 미치지 않기 때문에 단계 2에서 최소 1화소를 제거하는 방식을 채택하였다.

표 1의 실험에서 각각의 초성은 결합되는 중성의 유형과 초성의 좌표 x, y에 따른 면적별 분류표에 의하여 몇개의 그룹으로 나눌 수 있음을 알 수 있었다. 그리고, 각 그룹의 구분은 그림 2의 극점 분포 그래프를 이용하여, 초성의 경우는 표 3에서 구분한 것과 같이 초성은 14가지로, 중성은 6가지(표 7), 중성은 3가지(표 8)로 하였다.

‘ㄱ, ㅋ’, ‘ㄴ, ㄴ’ , ‘ㅍ, ㅍ’ 등과 결합되는 초성은 거



의 동일한 정보를 갖고 있음을 실험 과정에서 인지하였으며, 분류의 타당성을 확인하기 위하여 작성된 PDF 값의 길이에 따른 7개 그룹의 그룹내 PDF의 차이는 초성 유형 코드 1의 경우  $-9.507e-05$ 에서  $-1.457e-04$  사이, 유형 코드 2의 경우  $8.124e-06$ 에서  $1.909e-04$  사이 등으로 그룹내 모양꼴에 대한 유사성이 있음을 확인하였다.

중성의 경우는 더욱 세분하여, 상하의 위치 변동까지 고려한 3가지 정도의 유형 추가가 가능하나, 실험에 사용된 인쇄용 활자체의 경우에는 그 위치 변동의 정도가 미미하였으며, 아름다운 한글 글자꼴에서의 중성 위치에 대한 연구가 앞으로 있어야 할 것이다.

실험을 통하여, 생성된 낱자의 갯수는 모두 473개로서 초, 중, 중성별 각 유형의 크기는 실제 화소의 극점 인식에 의한 표준 크기로 하였으며, 초, 중, 중성별 유형의 갯수는 다음과 같다.

- ① 초성 19자 X 14 유형 = 266 유형
- ② 중성 21자 X 6 유형 = 126 유형
- ③ 중성 27자 X 3 유형 = 81 유형

이들의 유형 결정은 모양꼴 결정 루틴(그림 12, 13, 14)에 의해 추출된 유형 코드로 결정되며, 조합 과정(그림 11의 단계 3)에서 초, 중, 중성의 3 유형을 버퍼에 결합시킨 후 출력시켰다. 유형의 크기를 다양화하는 경우에도 본 연구의 유형 분류표가 그대로 적용될 수 있으나, 473개의 낱자 설계는 별도로 작업되어야 한다.

음절의 크기를 축소하는 경우( $33 \times 25$  등)에는 낱자 유형이 약 50% 정도 감소시킬 수 있으며, 총 279개(초성 6유형, 중성 4유형, 중성 3유형) 정도의 낱자 설계만으로 원하는 한글 음절을 출력시킬 수 있다. 그러나, 낱자의 유형 수를 축소 또는 확장 하는 경우에는 본 연구에서 제시된 정보표와 출력 알고리즘의 일부가 수정되어야 한다. 그림 15는 그림 11의 유형 결정 과정(단계2)을 거친 후, 출력된 결과를 보인것이다.

#### IV. 결 론

앞으로 기대되는 전자 사전의 구축과 전자 출판을 위한 한글 텍스트의 입력 작업에 그 효율성을 기하기 위해서는, 인쇄 및 컴퓨터용 한글의 정확한 유형 정보를 그 크기별로 정한 후, 한글 폰트를 표준화하여 한글 인식의 각종 문제를 해결할 수 있어야 한다.

본 연구에서는 한글 폰트 제작의 효율성을 기하기

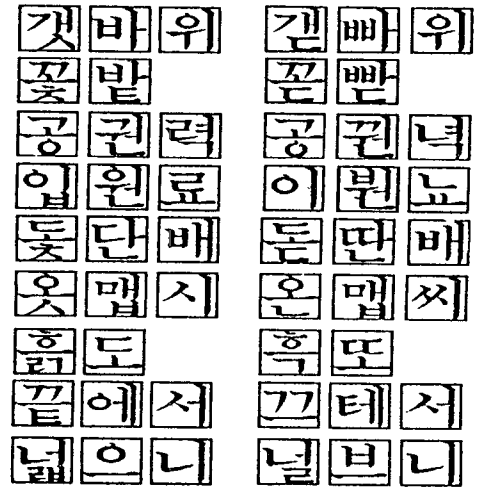


그림 15. 형 결정에 의해 출력된 결과

Fig 15. Output result by decision code.

위하여, 한글의 삼분법 특성을 고려한 음절내 각 낱자의 위치, 면적 정보와 낱자별 극점 위치의 밀집 유형 등을 분석하였고, 그에 따른 낱자 폰트 설계의 프로토타입을 제시하였다.

이러한 한글 음절의 낱자 유형 정보는 경제성 있는 한글 폰트의 개발에 있어서 유용하게 이용될 수 있으며, 제시된 낱자꼴 결정 루틴을 이용한 모아쓰기 알고리즘은 한국어 음운 변동 처리 시스템의 음가 정보 출력부로 구현함으로써, 그 실용성을 확인하였다.

한글 폰트 개발의 경제성을 위한 한글 낱자의 모아쓰기 처리는 한글 고유의 특성이며, 반드시 해결하여야 할 과제이다. 따라서, 모아쓰기 글자의 고품질을 위해서는 다양한 낱자 설계 방법이 앞으로 연구되어야 하며, 한글의 여러가지 서체에 맞는 음절 생성 알고리즘의 기술 개발과 한글의 구조에 적합한 표현 방식의 연구도 병행되어야 할 것이다.

물론 어떠한 글자꼴이 가장 아름답고, 과학적이고 합리적인가에 관한 연구는 단시일 내에 이루어질 수 없다. 따라서, 본 연구에서는 종래의 글자꼴에 크게 벗어나지 않으면서 조형 변화와 기계화가 용이한 한글 글자꼴의 설계를 전제로 하여, 기존 글자꼴의 낱자 결합에 관한 정보를 분석하였다.

그리고, 이에 따라 최소한의 초성(14가지), 중성(6가지), 중성(3가지)의 낱자꼴을 설계하여, 이들을 합성하여 한글 음절을 출력할 수 있는 방식을 연구하였다.

參 考 文 獻

[1] 이계영, 이상범, “한국어 음운 변동 처리를 위한 효율적인 Rule Base System의 구성, 전자공학회 논문집-B, 제28권 제12호, pp. 9-18, 1991년 12월.

[2] 김성락, 남시병, 이상범, “음절 단위를 이용한 한국어 연속 단어 인식”, 전자공학회 논문집-B, 제28권 제1호, pp. 9-14, 1991년 1월.

[3] 남궁 재찬, “Index Window 알고리즘을 이용한 한글 패턴의 부분 분리와 인식에 관한 연구”, 인하대학교 대학원 박사학위 논문, 1982년 2월.

[4] 박규태, 강현철, “원소 변환을 이용한 한글 패턴의 구조 분석”, 전자공학회 논문집, 제26권 제12호, pp. 61-69, 1989년 12월.

[5] 김정옥, 박규태 외, “계층구조 획 추출에 의한 한글 인식”, 전자공학회 논문집-B, 제28권 제8호, pp. 1-9, 1991년 8월.

[6] 이현표, 이균하 외, “한글 문서에서의 낱자 분리 알고리즘”, 한국 인지 과학회-한국 정보 과학회,

한글 및 한국어 정보처리 학술 발표 논문집, pp. 203-208, 1989년 10월.

[7] 송현, “한글 자형학”, 디지털사, 1985.

[8] 김정수, “한글의 역사와 미래”, 열화당, 1990.

[9] 김진평, “한글의 글자 표현”, 미진사, 1989.

[10] 전운호, 최종호, “가중치를 공유하는 신경회로망을 이용한 필기체 한글 문자 인식”, 전자공학회 논문집-B, 제28권 제8호, pp. 85-93, 1991년 8월.

[11] 이일병, “신경회로망 방법에 의한 필기체 한글 문자인식”, 신경회로망 응용 워크샵 논문집, 한국전자통신연구소, pp. 80-85, 1989년.

[12] K. Fukushima, “Neocognition: A hierarchical neural network capable of visual pattern recognition,” *Neural Networks*, vol. 1, pp. 119-130, 1988.

[13] T. Ito and K. Fukushima, “Recognition of Spatio-temporal Patterns with a Hierarchical Neural Network,” *International Joint Conference on Neural Networks(IJCNN)*, vol. 1, pp. 273-276, Jan. 1990.

著 者 紹 介

李 桂 英(正會員) 第28卷 B編 第12號 參照  
 현재 동국대학교 자연과학대학  
 전자계산학과 부교수

李 相 範(正會員) 第28卷 B編 第12號 參照  
 현재 단국대학교 전자공학과  
 교수

◆

金 奎 植(正會員)  
 1953年 6月 1日生. 1990年 8月 단  
 국대학교 대학원 전자공학과 졸업  
 (공학석사). 현재 동국대학교 정보  
 관리학과 강사. 주관심분야는 패턴  
 인식, 자연어 처리 등임.

