

제한된 글자 디자인에 의한 한글 조합형 글꼴의 자동생성

강 상수, 조 환규
부산대학교 전자계산학과

Automatic generation of Hangul Johap typeface
using small character set

Sang-soo Kang, Hwan-gue Cho
Dept. of Computer Science, Pusan National University

요약

한글 글꼴을 새롭게 만들려면 지금까지는 기본 글자인 자소를 디자인하든지 아니면 완성된 글자 전체를 디자인해야 했다. 조합형의 글자디자인의 경우, 전체 글자가 아니라 부분적인 자소를 디자인하기 때문에 전체 글자의 모양이 좋지 않다는 단점이 있었고 완성형의 경우 완성된 글자 전체를 모두 디자인해야 하는 단점이 있었다. 본 논문에서는 한글 글꼴 개발의 한 방법으로 제한된 글자의 디자인에 의한 전체 글꼴 생성에 관한 한 방법을 제시한다. 이 방법은 표준으로 설정된 몇 글자를 디자인하면 그 글자를 분석하여 자소들을 위한 글꼴 화일이 만들어지고 자소 글꼴 화일로부터 다른 모든 글자를 만들어 낸다.

I. 서론

한글을 컴퓨터에 사용할 경우 영어 등의 다른 글에 비해 많은 어려움이 있다. 이 중 글꼴의 개발은 특히 많은 문제를 안고 있다. 가장 큰 문제는 글자 자체가 복잡하고 많은 글자가 필요하기 때문에 발생하는 글꼴 설계상의 어려움이다. 글자가 많은 만큼 많은 기억공간을 유지할 필요가 있다는 것도 문제점 중 하나이다. 글꼴 자체가 크게 중요하지 않은 시스템에서는 속도도 빠르고 처리하기 쉬운 비트맵 방식을 사용하고 있지만 글꼴이 중요시되는 문서편집기나 탁상출판시스템에서는 항상 문제점으로 등장하는 것이 다양한 글꼴을 많이 확보하고 지원하는 문제이다.

한글은 여러 가지 원인으로 글꼴을 만들기가 어렵기 때문에 글꼴이 다양하지 못하다. 영어의 경우 천 여종, 일본의 경우 수백 여종에 이르는 글꼴이 한글의 경우 수십 종에 불과한 것이 한글 글꼴의 현실이다. 한글은 영어와는 달리 모아쓰기 방식을 사용하기 때문에 글꼴을 만들기가 매우 까다롭다는 것이 가장 큰 이유 중 하나이다. 또 영어의 경우 대소문자와 몇몇의 특수 문자를 합쳐서 많아야 200여종의 글자만 설계하면 쉽게 새로운 글꼴이 만들어진다. 이에 비해 한글의 경우 표현 가능한 글자 수는 총 11571자이다. 이들이 모두 사용되는 것은

아니지만 완성형만을 고려하더라도 KSC5601 완성형의 경우 2350자라는 많은 글자를 설계하여야 한다. 그리고 이들 각 글자들은 영어의 각 알파벳글자들에 비해 설계하기가 매우 어렵다. 이런 이유로 다양한 방법을 통해서 많은 글꼴을 빨리 만들어 낼 수 있는 방법이 연구되고 있다. [2-5]

본 논문에서는 한글의 모아쓰기 특징을 충분히 반영하면서 빠르게 글꼴을 생성할 수 있는 한 방법을 제시한다.

조합형으로 한글의 글꼴을 디자인할 경우 각 자소의 모양만을 기준으로 설계하는 것이 아니고 이들이 글자를 이룰 때의 모양에 대한 고려가 반드시 필요하다. 이 때문에 각 자소를 디자인하는 일을 완성형의 경우보다 어렵다. 완성형의 경우 글자 전체를 디자인하기 때문에 각 글자는 미려하지만 많은 글자를 디자인해야 하는 어려움이 따른다. 글자들 사이의 상대 크기도 충분히 고려되어 디자인해야 하는 어려움도 있다.

본 논문에서 제안하는 방법은 조합형을 위한 자소 글꼴을 만들어 내는데 목표를 두면서 글자의 디자인에서는 완성형을 사용한다. 조합형을 처리할 수 있게 자소들을 추출할 수 있는 글자집합을 구하여 자소가 아니라 그 글자집합에 해당하는 완성된 글자를 디자인하도록 하고 받아들인 글자집합으로부터 자소들을 자동으로 추

출한다.

본 논문에서는 먼저 2장에서 이 논문의 전체적인 접근 방향에 대해 몇 가지 과정으로 나누어 설명하고 3장에서는 실제적인 글자생성과정과 그 구체적 방식을 제시한다. 그리고 마지막으로 4장에서는 결론과 향후 연구 과제에 대해 설명한다. 본 논문에서 사용하는 용어는 주로 [8]에서 정의된 용어이다.

II. 전체 처리과정

2.1. 전체 처리의 개괄

글꼴을 개발할 경우 조합형을 기준으로 한다면 자소를 위한 글꼴을 만들어야 한다. 자소를 위한 글꼴을 만들기 위해서는 각 자소가 실제로 글자로 조합될 경우 얼마나 다른 형태를 띄느냐에 대해 먼저 알아야 한다. 이 경우 이미 여러 가지 연구가 되어 있고 자주 사용되는 비트맵 한글의 경우 첫소리 글자 8벌, 가운데소리 글자 4벌, 끝소리 글자 4벌이 가장 많이 사용된다. 기타 다른 시스템의 경우 6x2x1벌, 2x2x1벌, 10x8x4벌, 어떤 경우에는 21x3x14등 다양한 벌 수가 존재한다[6,7].

이렇게 벌 수가 달라지는 원인은 비트맵 글꼴의 경우 각 자소의 위치와 크기가 고정되기 때문이다. 같은 모양이라도 위치나 크기가 달라지면 다른 벌을 유지해야 하는 것이다. 그러나 크기와 위치는 글자의 종류에 따라 고정되는 것이 보통이므로 글자의 위치와 크기가 가변적이라면 실제적으로 필요한 벌 수는 매우 줄어든다.

이 논문에서 제안하는 방식은 조합형처럼 자소글꼴의 조합에 의해 글자를 만드는 방식을 이용하지만 디자인은 완성된 글씨를 디자인한다. 이를 위하여 가장 먼저 해야 하는 작업은 실제적으로 모양이 달라지는 자소를 고려해서 모든 자소를 포함하는 글자집합을 선정하는 작업이다. 글자집합의 선정에는 이 이외에도 글자의 타입분류 등 여러 가지 작업이 고려되어야 한다.

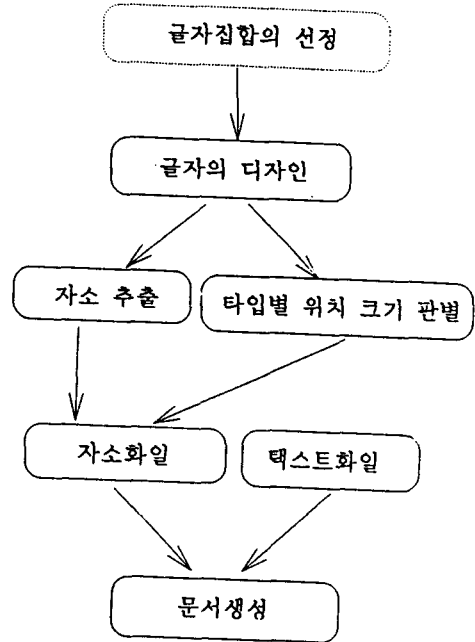
글자집합의 선정이 끝나면 그 글자집합내의 글자들은 기존의 방법과 마찬가지로 디자인되어야 한다. 디자인 작업이 끝나면 디자인한 글자에서 자소들을 선정하고 글자의 타입별로 각 자소의 위치와 크기가 추출된다. 마지막으로 분리된 자소와 그들의 타입별 위치 및 크기가 하나의 자소 글꼴 화일에 저장된다.

실제적인 글꼴을 생성하는 시스템은 여러 가지 방법이 있겠지만 이 논문에서는 문서를 만들 일반 아스키문서화일을 유지하고 글씨체를 지정하면 주어진 일반 아스키문서화일을 지정된 글씨체로 문서화시키는 방법을 택했다.

이 과정을 다시 정리해 보면 [그림 2-1]과 같다. 각 과정에서 하는 일을 정리하면 [표 2-1]과 같다.

글자 집합의 개수는 글자의 질 혹은 응용분야가 무엇

인지에 대해 달라질 수 있다. 본 논문에서는 되도록 작은 글자 집합에서 모든 글자를 만들어 낼 수 있는 자소를 위한 글꼴을 만들기 위한 한 예로 20글자에 대한 디자인 방식을 제안한다.



[그림 2-1] 전체 시스템의 개요

[표 2-1] 전체 시스템의 각 과정에서 하는 일

과 정	하는 일
글자집합의 선정	모든 자소를 추출할 수 있고 모든 글자형에 대해 그 위치와 상대 크기를 알 수 있는 글자집합을 선정
글자의 디자인	정해진 글자집합내의 글자를 디자인.
자소 추출	디자인된 글자들에서 자소를 분리해 냄
타입별 위치 크기판별	미리 나누어진 타입별로 각 자소의 글자 위치와 크기를 판별
자소화일	추출된 자소는 100x100의 표준 격자로 저장하고 글자의 타입별로 자소의 위치와 크기를 저장하는 자소글꼴화일 만들
문서 생성	자소화일과 문서를 만들 일반 아스키문서 화일을 받아들여 그 글씨체를 통해 문서를 생성

2.2. 글자집합의 선정

글자집합의 선정에서 가장 먼저 할 작업은 모든 글자를 만들 수 있는 자소의 추출이다. 한글의 자소는 종류만 생각한다면 닿소리글자 14자와 홀소리 글자 10자의 24자이다. 그러나 이것만으로는 부족하다. 먼저 고려해야 하는 점은 변형되는 자소이다. 글꼴을 생성할 경우 쓸 수 있는 모든 글자를 고려하여야 한다. 겹글자나 결합되는 자소에 따라 모양이 변하는 자소에 대해서도 고려해 주어야 한다.

결합되는 자소에 따라 심하게 변경되는 대표적인 자소는 첫 소리글자 중 'ㄱ'자이다. 이 글자는 결합하는 가운데 소리글자가 가운데소리 세로글자나 가운데 소리 가로글자나에 따라 변화가 심하다.

가운데 소리글자의 경우 'ㅣ'가운데소리 글자와 결합하는 가운데소리 글자의 경우는 'ㅣ'가운데소리 글자가 없는 가운데소리 글자에서 'ㅣ'를 삽입하여 합쳐 줌으로써 구할 수 있으므로 이 들을 모두 받아들이지는 않는다(예를 들면 'ㅁ'는 'ㅣ'에서 'ㅍ'는 'ㅇ'에서 각각 'ㅣ' 가운데소리 글자를 합성하여 구할 수 있다). 이런 점들을 모두 고려해서 실제로 추출해야할 첫소리 글자, 가운데소리 글자, 끝소리 글자 별로 필요한 자소는 [표 2-2]와 같이 정리할 수 있다.

아래 표에서 첫소리 글자는 가운데소리 글자에 따라 모양이 바뀌지 않는 경우가 타입 1, 모양이 바뀌는 경우가 타입 2이다.

[표 2-2] 입력할 글자집합에 포함되어야할 자소

첫소리글자	타입1	ㄴ, ㄷ, ㄱ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅊ, ㅋ
	타입2	ㄱ, ㅋ, ㅅ, ㅈ, ㅊ, ㅋ
가운데소리 글자	ㅣ, ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅜ, ㅠ, ㅡ, ㅣ	
끝소리 글자	기본적으로 첫소리글자를 이용. 겹글자는 홀글자를 합성하여 구함	

이런 점들을 만족하는 글자집합은 여러 가지가 있을 수 있다. 여러 번의 실험과 검사를 반복해서 3가지의 글자집합을 선정하였다.

글자집합1 = {ㅈ, ㅊ, ㅋ, ㅌ, ㄴ, ㄷ, ㄱ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅊ, ㅋ, ㅌ, ㅍ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅜ, ㅠ, ㅡ, ㅣ}

글자집합2 = {삼, 카, 낚, 체, 며, 례, 쯤, 좌, 좌, 트, 득, 취, 세, 웅, 뽕, 꿈, 프, 회, 빛, 지}

글자집합3 = {카, 샤, 뽕, 려, 지, 코, 보, 초, 수, 듀, 와, 뒤, 찰, 현, 낚, 땅, 쯤, 득, 늘, 썹}

이들 글자집합마다 실제적으로 글자를 뽑는데 적용된 방식은 세부적인 부분이 약간씩 다르다. 같은 기준을 가지면서도 디자인하기 좋은 쉬운 글자를 선택해야 한다. 전반적으로 글자집합 1 보다 글자집합 3 이 글자가 비교적 쉬워졌다는 것을 알 수 있다. 다음 장 부터는 글자집합 3에 적용된 방식을 중심으로 설명한다.

III. 글자의 생성

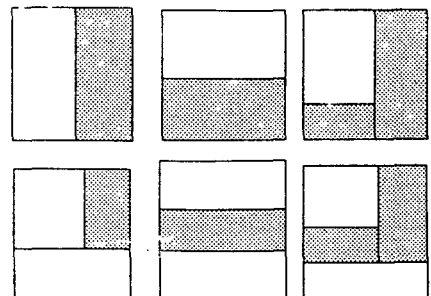
3.1. 글자의 디자인

글자의 디자인은 매우 어려운 단계이다. 이는 개인적인 미적 감각이 요구되는 부분이기 때문에 일반 사람들이 쉽게 할 수 있는 부분은 아니다. 그리고 글꼴의 디자인 자체는 본 논문의 주제를 벗어나는 부분이므로 생략하기로 한다.

본 논문에서는 글자를 디자인할 때 단순히 타블렛을 통해 글자를 입력하는 방식을 사용했다. 실제적인 응용 프로그램에 적용하기 위해서는 정확한 위치에 글자를 입력해야 하겠지만 일반 사람들이 멧을 내어 꾸밈체로 쓴 글자들도 글자의 디자인에 도움이 될 것이다. 이러한 꾸밈체를 일반사람이 부담없이 쓸 수 있는 타블렛으로 입력하여 글꼴을 디자인하는 것이다.

3.2. 자소의 분류

글씨를 쓸 때 자소의 상대적인 크기와 위치 등에 따라 자소를 분리할 필요가 있다. 가장 많은 자소의 분류는 [그림 3-1]과 같은 6가지 분류이다.



* 그림에서 음영부분은 가운데소리 글자 부분

[그림 3-1] 한글의 글자별 타입분류

그러나 위와 같은 분류는 단순한 형태상의 분류이고 사실상은 보다 세부적인 분류가 필요하다. 이번 논문에서는 가운데소리 글자를 중심으로 8개, 받침의 유무를 따지면 총 16개의 타입분류를 하였다. [표 3-1]은 분류와 그에 해당하는 글자 집합내의 실제 글자를 보여준다.

[표 3-1] 가운데 소리 글자 분류와 글자 집합에서의 분류

가운데소리 글자의 분류	글자 집합 중의 글자	
	받침이 없는 글자	받침이 있는 글자
ㄱ, ㅋ, ㆁ, ㆁ	카, 샤	칼
ㄷ, ㅌ, ㄷ, ㅌ	빠, 려	현
ㅣ	지	깁
ㄴ, ㄴ	코, 보	폼
ㅡ	츠	늬
ㅍ, ㅍ	수, 듀	뚝
나, ㄴ, 나, ㄴ	와	망
ㄹ, ㄹ, ㄹ	뒤	췌

위의 분류는 가운데소리 글자의 글자 내에서의 무게 중심을 따져서 비슷한 글자끼리 묶은 것이다. 일반적으로 글자를 디자인할 경우 무게중심이 안정되어야 한다 [9]. 바꾸어 말하면 가운데소리 글자의 중심이 비슷하면 첫소리 글자의 모양, 위치, 크기 등이 모두 비슷하다. 필요한 타입은 16가지이지만 닿소리 글자와 한글의 빈도수를 고려해서 가장 많은 빈도수를 보이는 4가지 타입에 대해서는 글자를 두 가지씩 선정했다.

3.3. 자소글꼴화일의 생성

만들어진 글자집합을 이용하여 자소를 추출하고 각 타입별로 글자들의 위치와 크기를 설정하여 디자인한 글자의 자소 글꼴 화일을 만들어야 한다.

글자들은 자소로 분리되어 각각 100x100의 표준으로 정규화 되어 자소화일에 저장된다. 이 때 실제로 문서를 만들 경우 작업을 줄이기 위해서 가운데소리 글자에 따라 타입의 변화가 없는 것도 완전히 빌을 갖추어 저장하였다.

자소의 상대 위치는 디자인된 글자를 기준으로 전체 글자에서 각 자소의 위치를 상대위치로 위에서 나누어진 타입에 따라 분류하여 저장한다.

n번째 글자에 대한 상대좌표를 구하는 식은 아래와 같다.

$$Rsize_n(JASO, POS) = \frac{Maxsize_n(JASO, POS) - Minsize_n(JASO, POS)}{MaxChar_n(POS) - MinChar_n(POS)}$$

(JASO ∈ {첫소리 글자, 가운데소리 글자, 끝소리 글자}
 POS ∈ {x좌표, y좌표})

3.4. 글자의 생성

실제적인 글자를 생성하기 위해서는 위에서 만들어진 자소글꼴화일과 문서를 만들기 위한 일반 아스키 문서 화일이 필요하다. 본 논문에서는 2 바이트 조합형 형태의 한글을 사용하였다.

한 글자를 만드는 알고리즘은 다음과 같다.

입력 : 자소글꼴(FONT), 자소 타입 (Type) 및 비율 (Ratio) 표

출력 : 글자를 위한 제어점들의 집합(Cp)

과정 :

a. SplittoJaso(ch, &fir, &mid, &las)

b. $Type_{JASO} = TypeTable_{JASO}(index)$
 $Ratio_{JASO} = RatioTable_{JASO}(index)$
 ($index \in \{fir, mid, las\}$)

c. $Cp_{JASO} = FONT_{JASO}(Type_{JASO}) * Ratio_{JASO}$

d. $Cp_{CHAR} = \sum Cp_{JASO}$

3.5. 생성결과

아래의 결과는 타블렛 입력에 의하여 디자인한 글자와 실제로 생성된 문서들이다.

가 사 ㅍ ㄹ ㄹ ㄹ
 코 보 츠 수 듀
 외 뒤 칠 흰 깁
 망 췌 뚝 늬 폼

(a) 입력받은 글자

먼 길 떠나기 위해
 단정어서 깎다
 아직 어둠이 머뭇거리는
 새벽 하늘이 아침이 온다
 희끗희끗 날리며 앉으며
 순식간에 천지를 휘감아
 환상지는 눈발
 서늘 부딪히는 눈발
 서늘 부딪히며 떨어리며
 지는 안 하늘 폭풍이인다.
 나뭇가지 위의 새둥지기
 툭 떨어지고 새들이
 끈뭉끈뭉 황급히 떠난다.

(b) 생성된 글자
 [그림 3-2] 생성예 1

카 사 버 려 지
 코 보 츠 수 두
 외 튀 찰 편 김
 망 찢 뚝 늘 름

(a) 입력받은 글자

카 사 버 려 지
 코 보 츠 수 두
 외 튀 찰 편 김
 망 찢 뚝 늘 름

(a) 입력받은 글자

그 집 앞에 있는 커다란 나무 밑에는
 식탁이 마련되어 있었고 (3일의 쏘끼)와
 (모자칼이 해터)가 차를 들고 있었다.
 그들은 자기를 사이에 끼여 앉은 도어
 마우스를 우선으로하는지 그 위에 팔꿈
 치를 얹고 머리위로 이야기를 나누고 있
 었다.

"도어 마우스가 몹시 불편하겠구나."
 엘리스는 측은한 생각이 들어 이렇게 흥
 얼거렸다.

"하지만 차가 들어 모르고 있을 터니
 까 다행이야."
 식탁은 저법 넉넉했는데 인일인지 그들
 식은 한쪽어 복잡하게 몰려 앉아 있었다.

[그림 3-4] 생성예 3

여전언 미쳐 몰랐어요
 기 수 울
 뭉가을없이 밤마다 뜰드 달드
 '여전언 미쳐 몰랐어요'
 이렇게 사무치게 그리울 켜도
 '여전언 미쳐 몰랐어요'
 달이 암만 밝아도 쳐다볼 켜
 '여전언 미쳐 몰랐어요'
 이제는 저 달이 서름인 켜
 '여전언 미쳐 몰랐어요'

(b) 생성된 글자
 [그림 3-3] 생성예 2

IV. 결론 및 향후 연구 방향

본 논문에서는 한글의 글꼴을 다량으로 만들기 위한
 방법을 제시하였다. 이 방법은 조합형의 글자 디자인과
 완성형의 글자 디자인 사이에 나타나는 문제점을 해결
 하기 위해 두 가지 방식의 장점을 결합하였다. 논문에서
 제안한 방식과 조합형, 완성형의 글자 디자인에 대한 비
 교는 [표 4-1]과 같다.

[표 4-1] 제안된 방식과 조합형, 완성형 글자 디자인
 사이의 비교

비교 \ 방식	조합형	완성형	제안된 방식
글꼴의 디자인 난이도	어려움	어려움	비교적 쉬움
디자인에 걸리는 시간	많이 소요	매우 많이 소요	단시간에 디자인
글꼴을 위한 공간	적음	많음	적음
다른 글꼴형식으로 변환	쉬움	어려움	쉬움

본 논문에서는 글꼴표현을 위해 스트록 방식을 사용하였다. 비트맵 방식과 윤곽선 방식 그리고 사용된 방식의 비교가 [표 4-2]에 제시되어 있다.

[표 4-2] 제안된 방식과 비트맵, 윤곽선 방식의 비교

비교 방식	비트맵 방식	윤곽선 방식	제안된 방식
해상도 관계	해상도에 의존적	해상도에 관계없음	해상도에 관계없음
출력속도	매우 빠름	느림	비교적 빠름
기억장소 소모	매우 심함	약간 심함	적음
변형가능성	거의 변형 불가능	다양한 변화가능	다양한 변화 가능
글자의 디자인	어려움	매우 어려움	쉬움
다른 화일로 변환	매우 어려움	어려움	비교적 쉬움

현재 글꼴 생성 시스템에서 글자꼴을 디자인하는 것은 단순한 스트록의 형태이다. 비록 스트록에서 그 굵기를 줄 수 있다고 하더라도 조그마한 글씨에서는 상관없지만 큰 글씨에는 영향을 미친다. 글자를 디자인하는 부분이 실용될 수 있게 보장되고 기본 스트록 폼에서 돌기를 주는 방법 등의 글자가 실용적으로 사용될 수 있을 것으로 생각된다.

현재의 생성되는 시스템에서는 문서작성용 아스키화일을 따로 입력받는다. Tex이나 troff등의 시스템에서도 이러한 방식을 따르고 있지만 WYSWYG을 따를 수 있는 프로그램의 개발도 필요하다고 생각된다.

마지막으로 생각해 볼 문제는 글자의 표준에 관한 문제이다. TYPE i이나 true type등은 한글의 조합형 방식에는 그다지 적합하지 못하다. 조합형 한글을 저장하는 표준안과 실제로 글자가 생성되었을 때의 표준안이 정립되어 여러 시스템에서 개발한 다양한 글씨체들이 별다른 노력없이 서로 호환성 있게 교환되어야 전체적인 한글 소프트웨어가 더욱 빨리 발전할 수 있을 것으로 생각된다.

참고 문헌

- [1] 임 순범, "전산글꼴 처리기술과 글꼴 자료교환," 개방시스템 단기강좌 자료집, 1994, pp153-172
- [2] 안 은영, 조 형제, "기본 획 합성에 의한 한글 글꼴 생성," 정보과학회논문지, 21권, 4호, 1994년, 4

- 월, pp648-658
- [3] 안 상수, "한글꼴의 원형태 연구2," 제5회 한글 및 한국어정보처리 학술대회 학술 발표 논문집, 1993, pp161-171
- [4] 최 혜옥, 박 동인, "한글꼴을 근간으로 한 서체설계지원시스템," 제 5회 한글 및 한국어정보처리 학술대회 학술 발표 논문집, 1993, pp173-184
- [5] 안 상수, "한글꼴의 원형태 연구," 제4회 한글 및 한국어정보처리 학술대회 논문집, 1992, pp601-611
- [6] 문화부, 문화부 개발 서체 견본(한글날 546돌 기념), 서울시스템주식회사, 1992.
- [7] 임 현모, "컴퓨터와 한글의 만남", 정보문화사, 1992.
- [8] 김 경석, "한글 기계화와 전산화 분야의 잘못된 용어 몇 가지와 그 대안," 한글 세소식210, 1990, 2, pp 12-23
- [9] 송 현, "한글 자형학," 디자인, 1988