메타폰트를 이용한 UFO 기반의 한글 외곽선 폰트 시스템

권경재O, 손민주, 정근호, 최재영
숭실대학교 컴퓨터학부
gwon@ssu.ac.kr, sonmibz@ssu.ac.kr, metaopus@gmail.com, choi@ssu.ac.kr

Korean Outline Font Editing System based on UFO Using METAFONT

Gyeong-Jae GwonO, Min-Ju Son, Geun-Ho Jeong, Jae-Young Choi
School of Computer Science & Engineering, Soongsil University

요 약

오늘날 폰트를 디자인하는데 주로 사용되는 외곽선 방식은 글자의 크기를 손쉽게 변경할 수 있으나 글자의 굵기나 스타일을 변화시키려면 다시 수작업을 통해서 디자인을 변경해야 한다. 이를 보완하기 위한 프로그래머블 폰트인 메타폰트는 매개변수를 사용하여 글자의 변화가 매우 용이하다. 하지만 메타폰트는 프로그래밍 언어이므로 메타폰트에 대한 선행학습이 필요하여 폰트 디자이너에게 사용되지 않았다. 따라서 본 논문에서는 폰트 디자이너에게 익숙한 외곽선 방식에 편집기를 제공하면서 글자의 스타일에 대한 변화를 메타폰트에서 처리하여 다양한 폰트를 파생할 수 있는 한글 외곽선 폰트 시스템을 제안한다. 이를 위하여 본 시스템에서는 외곽선 방식의 폰트를 제작할 수 있도록 하는 웹 외곽선 폰트 편집기를 구현하였으며 외곽선 방식의 폰트를 메타폰트로 변환하는 UFO2mf를 구현하였다. 본 논문에서 제안하는 폰트 시스템은 기존 외곽선 방식의 스타일 변화에 대한 수고를 덜어줄 수 있을 것으로 기대한다.

주제어: 메타폰트, 외곽선 폰트, 폰트 시스템, 프로그래머블 폰트, UFO 폰트

1. 서론

한글 폰트는 사용하는 글자 수가 로마자에 비해 매우 많고 형태가 복잡하다. 한글 폰트를 디자인하기 위해 11,172자를 디자인해야 하며 한글은 조합형 글자이므로 똑같은 자소라고 하더라도 같이 조합되는 자소들에 따라서 크기와 위치, 모양이 약간씩 다르다[1]. 이렇게 같은 자소라도 상황에 따라 다양한 모양을 가질 수 있는 것을 '벌수'라고 한다. 따라서 폰트 디자이너가 일반적인 외곽선 방식의 폰트 편집기를 이용하여 한글 폰트를 제작하기 위해서는 필요한 여러 모양의 벌수를 일일이 디자인해야 하므로 일 년 이상의 시간이 소요된다.

오늘날 폰트를 디자인하기 위해 사용되는 외곽선 방식은 베지어 곡선(Bézier Curve)을 사용하여 문자의 외곽선을 그리는 방식이다. 외곽선 방식으로 제작된 폰트는 외곽선 정보를 이용해 폰트를 확대 및 축소하기 때문에, 크기에 상관없이 사용할 수 있으며 고품질의 출력이 가능하다. 하지만 외곽선 방식은 완성된 폰트의 획의 굵기, 모양과 같은 스타일의 변화를 주기 위해서는 많은 비용과 시간을 들여 일일이 별도의 폰트를 제작해야 한다는 단점이 있다[2].

이러한 외곽선 방식의 단점을 해결하기 위하여 새로운 폰트 제작 방식인 프로그래머블 방식이 연구되었다. 프로그래머블 방식은 폰트를 제작할 때 사용하는 프로그래밍 언어를 이용하여 프로그램을 제작하고 실행하여 폰트를 생성한다. 프로그램 코드를 작성할 때, 폰트 스타일에 영향을 줄 수 있는 요소를 매개변수화할 수 있으며 매개변수에 값을 변화시킴에 따라 다양한 모양의 글자를 제작할 수 있다[3]. 따라서 프로그래머블 폰트는 크기와 위치가 다른 벌수를 매개변수를 이용하여 쉽게 디자인할 수 있어 한글 폰트를 디자인하기에 적합한 기능을 제공한다[4]. 하지만 프로그래머블 방식은 프로그래밍 언어를 사용하여 폰트를 제작해야 하므로 폰트 디자이너가 사용하기가 어렵다.

우리는 이와 같은 단점을 개선하고자 프로그래머블 폰트 중 하나인 메타폰트를 기반의 폰트 템플릿을 사용자에게 제공하고 사용자는 웹을 이용하여 자신이 원하는 스타일에 맞춰 매개변수의 값을 수정하여 템플릿에 적용할 수 있는 시스템을 선행 연구를 진행하였다[5, 6]. 하지만 선행 연구에서 제안한 시스템은 제공하는 메타폰트 템플릿에서 종속되어 크게 벗어나지 않는 스타일 변화만 줄 수 있다. 따라서 본 논문에서는 메타폰트에서 매개변수를 이용하여 다양한 폰트를 파생할 수 있다는 장점을 반영하고 선행 연구에서 템플릿에 종속되어 변화에 한계가 있다는 단점을 개선하여 메타폰트 이용한 UFO 기반의 한글 외곽선 폰트 시스템을 제안한다. 본 시스템은 메타폰트에 대한 지식이 없는 폰트디자이너가 익숙한 외곽선 방식을 사용하여 폰트를 디자인하면 외곽선 방식의 데이터를 메타폰트 템플릿으로 자동으로 변화시켜 매개변수로 다양한 모양의 폰트를 파생할 수 있다.

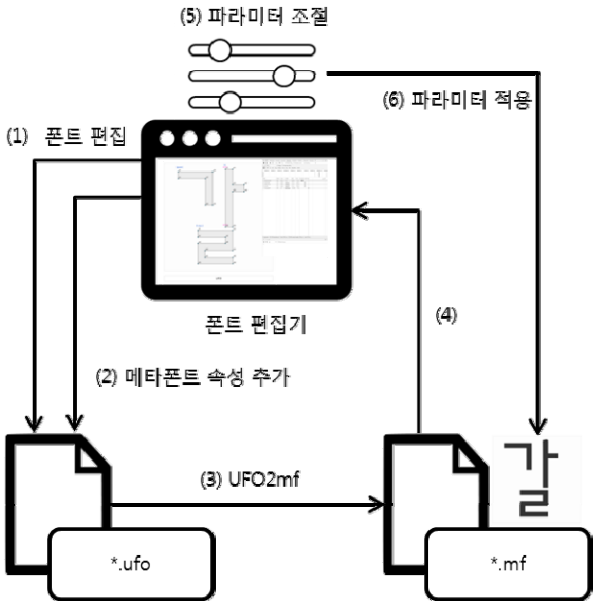


그림 1. 폰트 시스템 구조

2. 관련연구

2.1 메타폰트

대표적인 프로그래머블 폰트인 메타폰트는 TeX 조판 시스템의 질을 높이기 위해 개발되었다. 메타폰트는 1986년 스탠포드대학교의 도널드 커누스(Donald Knuth) 교수가 개발한 폰트 설계시스템으로, 메타폰트 프로그래밍 언어를 사용하여 폰트를 생성한다[7]. 메타폰트는 프로그래밍 언어이기 때문에 사용자는 변화가 가능한 스타일을 매개변수로 만들어 매개변수의 값을 수정하는 작업만으로 다양한 폰트를 제작할 수 있다. 하지만 메타폰트는 외곽선 방식과는 다르게 프로그래밍 언어이므로, 일반 폰트 디자이너가 사용하기에는 어려워 잘 사용되지 않았다.

외곽선 방식은 GUI 폰트 편집기를 통해 글자 디자인의 변화를 실시간으로 확인 할 수 있고 마우스를 이용하여 다루기 쉽기 때문에 오늘날 폰트를 디자인하기 위해 자주 사용된다. 하지만 메타폰트는 프로그래밍 코드이므로 코드를 컴파일하고 실행하기 전까지는 디자인을 확인할 수 없다. 또한 메타폰트는 논리적인 수학적 방식을 사용하여 코드를 작성하여 문자를 디자인하기 때문에 사용하기가 어렵다[8]. 그러므로 메타폰트가 디자인의 변화를 실시간으로 확인할 수 있다고 하더라도 폰트 디자이너들은 메타폰트에 대한 선행학습이 필요하다. 따라서 본 시스템에서는 폰트 디자이너가 메타폰트를 사용하지 않고 익숙한 외곽선 방식을 사용하여 직관적으로 폰트를 디자인할 수 있도록 외곽선 폰트 편집기를 제공한다. 그리고서 메타폰트의 매개변수를 활용하기 위해 디자이너가 완성한 외곽선 방식의 폰트를 본 시스템은 자동으로 메타폰트 코드로 변환하고 매개변수를 적용하여 다양한 폰트를 파생할 수 있도록 한다.

2.2 UFO (Unified Font Object)

본 외곽선 폰트 편집기를 이용하여 외곽선 방식의 폰트를 완성하고 메타폰트 코드로 변환하기 위해서 외곽선 폰트에 메타폰트에 필요한 데이터를 일부 포함해야 한다. 본 시스템은 이를 위하여 외곽선 방식을 사용하는 UFO 폰트 형식을 사용한다. UFO는 cross-platform, cross-application의 특징을 가지고 있고, 특히 폰트 데이터가 XML로 기술되어 사람과 기계 모두가 읽을 수 있는 형식을 갖도록 규정되었다[9]. 또한 정의되어 있지 않은 XML의 태그나 속성을 사용자가 정의할 수 있다. 이를 이용하여 시스템에 필요한 데이터를 UFO에 추가하고 파싱하여 사용할 수 있다. 본 시스템에서는 이러한 UFO의 특징을 이용하여 사용자가 완성한 폰트를 UFO 폰트 형식으로 저장하고, 메타폰트에 필요한 정보를 UFO에 포함하여 메타폰트로 변환한다.

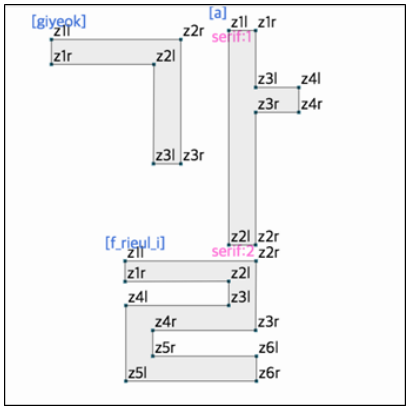
3. UFO 기반의 한글 외곽선 폰트 시스템

본 시스템의 구조는 그림 1과 같다. 웹 기반의 폰트 편집기를 제공하여 폰트 디자이너가 친숙한 외곽선 방식을 이용하여 폰트를 편집한다. 또한 사용자는 메타폰트에서 제공하는 속성들을 폰트에 부여할 수 있으며 이때 필요한 데이터를 UFO 폰트 형식에 저장한다. 그러고 나서 UFO2mf를 이용하여 메타폰트 코드로 변환하여 메타폰트로 만들어진 결과를 폰트 편집기를 통해 나타낸다. 사용자는 스타일을 변경할 수 있는 매개변수를 이용하여 다양한 스타일의 폰트를 파생할 수 있다.

3.1 웹 기반의 폰트 편집기

본 시스템에서 제공하는 웹 폰트 편집기는 외곽선 방식을 사용하는 일반 폰트 편집기의 기능을 제공한다.

사용자는 문자를 디자인하는 것뿐만 아니라 본 시스템에서 제공하는 메타폰트의 기능을 UFO 형식에 추가할 수 있다. 예를 들어, 그림 2(a)에서 외곽선 방식에서 'ㅏ'의 세로획 양 끝에 메타폰트에서 제공하는 2개의 세리프 기능을 폰트 편집기를 이용하여 추가하면 UFO 폰트 형식에 해당하는 데이터가 저장된다. 이후에 UFO2mf를 실행시키면 UFO에 저장된 데이터를 읽어 메타폰트로 알맞게 변환하게 되는데 그림 2(b)와 같이 해당하는 위치에 사용자가 추가한 각각의 세리프가 표현된 것을 확인할 수 있다.



(a) 외곽선 방식

(b) 메타폰트

그림 2. 세리프 속성 추가 및 변환 결과

3.2 UFO 속성

UFO를 메타폰트로 변환하기 위해 필수적으로 포함해야 하는 데이터가 존재한다. 메타폰트는 펜 기능을 이용하여 글자를 그리는 방식이기 때문에 펜의 양 끝점을 알아야 한다. 따라서 UFO 폰트 형식에서 펜의 양 끝점에 맞게 점들을 쌍으로 묶어주는 작업이 필요하다. 따라서 UFO 형식에 기존에 존재하는 태그인 'point'에 'name'이라는 속성을 추가하여 상응하는 두 개의 점이 한 쌍이라는 정보를 얻는다. 또한 웹 폰트 편집기에서 사용자가 추가하는 속성들을 위하여 'point' 태그에 필요한 속성들을 추가하였다. 예를 들어, 그림 3은 속성이 추가된 UFO 폰트 형식이다. 쌍을 위하여 'name' 속성이 추가된 것을 확인할 수 있고, 또한 'serif', 'dependX', 그리고 'dependY'와 같은 속성이 추가된 것을 볼 수 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<glyph name="a" format="1">
  <advance width="1024"/>
  <outline>
    <contour>
      <point name="z1l" serif="1" x="576" y="963" type="line"/>
      <point name="z2l" serif="2" x="576" y="413" type="line"/>
      <point name="z2r" x="644" y="413" type="line"/>
      <point name="z3r" dependX="x2r" x="644" y="753" type="line"/>
      <point name="z4r" x="753" y="753" type="line"/>
      <point name="z4l" x="753" y="817" type="line"/>
      <point name="z3l" dependX="x1r" x="644" y="817" type="line"/>
      <point name="z1r" x="644" y="963" type="line"/>
    </contour>
  </outline>
</glyph>
```

그림 3. 속성이 추가된 UFO 폰트 형식

3.3 UFO2mf

본 논문에서는 완성된 UFO 형식의 폰트를 메타폰트로 변환하기 위하여 파이썬을 이용하여 UFO2mf를 구현하였다. XML 파서를 이용하여 UFO를 파싱하여 메타폰트에서 필요한 데이터로 계산하여 코드를 작성한다. 또한 사용자가 추가한 속성을 읽어 해당하는 함수를 호출하여 추가 처리한다. 그리고 사용자에게 글자의 스타일 변화를 위하여 제공하는 14개의 매개변수를 적용하여 메타폰트 코드를 생성하였다.

3.4 한글 매개변수 추출

본 시스템에서는 폰트의 스타일을 변경할 수 있는 매개변수를 제공한다. 이를 위하여 한글의 특징을 분석하여 그림 4와 같이 14개의 매개변수를 선정하였다[6]. 글자 전체의 넓이와 높이를 조절할 수 있으며, 초성, 중성, 그리고 종성 각각의 위치를 수평, 수직으로 이동시킬 수 있다. 또한 곡선의 정도를 매개변수로 주어 곡선을 점점 직선에 가깝게 변화시킬 수 있다. 뿐만 아니라, 세리프의 크기와 글자의 굵기, 그리고 기울기를 매개변수로 구현하였다. 이처럼 14개의 매개변수로 사용자는 폰트의 전반적인 스타일을 변화시킬 수 있다.

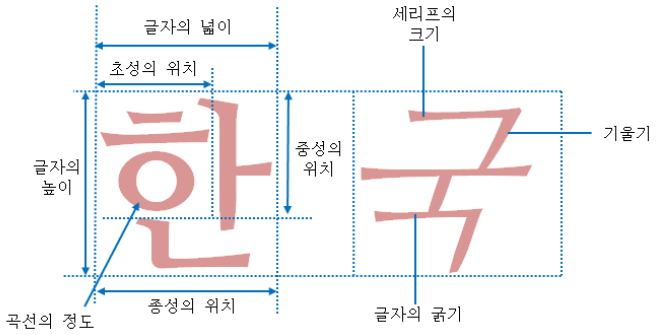
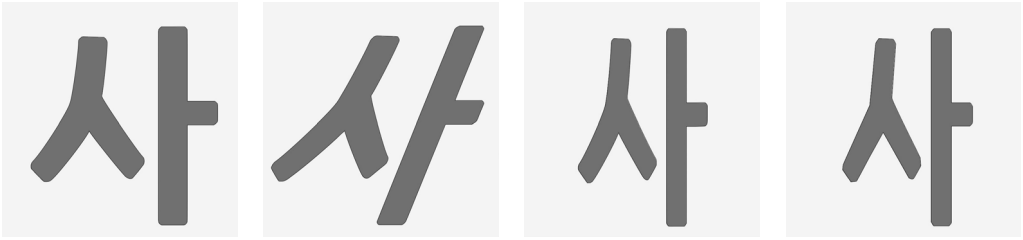


그림 4. 한글 매개변수 종류

(a) 기본 샘물체

(b) 점을 이동하여 외곽선 변형

그림 5. 웹 폰트 편집기를 이용한 폰트 편집



(a) 기본형 (b) 기울임 (c) 넓이 축소 (d) 곡선의 정도

그림 6. 매개변수에 따른 글자의 변화

4. 실험 및 분석

HTML5로 제작된 웹 폰트 편집기의 UI는 그림 5(a)와 같다. 그림 5에서 나타내는 폰트는 '샘물체'를 기반으로 하였다. 왼쪽 화면은 사용자가 마우스를 이용하여 일반적인 폰트 편집기처럼 편집할 수 있다. 편집을 완료하고 변환 버튼을 누르면 메타폰트 코드로 변환되고, 오른쪽 화면에서 메타폰트로 생성된 폰트를 보여준다. 그림 5(b)는 그림 5(a)에서 점을 이동하여 문자의 스타일을 변화시킨 예이다.

사용자는 매개변수를 조절할 수 있는 슬라이더를 이용하여 자신이 원하는 스타일의 폰트를 생성할 수 있다. 예를 들어 그림 6(a)은 UFO 형식의 나눔고딕 '사'를 UFO2mf를 사용하여 메타폰트로 변환한 결과이다. 그림 6(b)은 기울기의 값을 변화를 주어 글자를 기울였고, 그림 6(c)은 글자의 전체 넓이를 줄여 표현하였다. 마지막으로 그림 6(d)은 글자의 전체 넓이를 줄이고, 곡선의 정도의 값을 최소값으로 주어서 획의 모서리를 곡선에서 직선으로 표현한 모습이다. 이처럼 다양한 변수를 적용하여 새로운 스타일의 폰트를 생성할 수 있다.

5. 결론

본 논문에서 제시하는 한글 외곽선 폰트 시스템은 사용자가 편집한 외곽선 폰트에 메타폰트를 활용하여 다양한 스타일의 폰트를 생성할 수 있는 효율적인 시스템이다. 기존의 외곽선 방식을 사용할 경우, 글자를 제작하는데 많은 시간이 요구될 뿐만 아니라 완성된 글자의 스타일을 변경하기 위해서는 일일이 새롭게 작업해야 하는 불편함이 있다. 이와 달리 본 시스템은 메타폰트에서 제공하는 매개변수 값을 수정하여 다양한 폰트를 파생할 수 있다. 또한 사용자는 메타폰트에 대한 지식과 프로그래밍 기술이 없어도 제공하는 웹 폰트 편집기를 이용하여 편리하게 매개변수 값을 조절할 수 있다.

본 시스템에서 현재 제공하는 매개변수는 14개뿐이지만, 한글의 스타일을 결정하는 요소는 더욱 다양하므로 다양한 매개변수를 추출하고 적용시키기 위한 연구를 진행 중이다. 또한 웹 폰트 편집기는 기본적인 기능만 제공하지만, 지속해서 필요한 기능들을 추가 개발할 예정이다. 본 시스템을 이용하여 기존의 외곽선 방식보다 적은 비용으로 폰트를 제작할 수 있을 것이며, 폰트의 시장성을 더욱 높일 수 있을 것으로 기대한다.

참고문헌

[1] 이준희, 정내권, 컴퓨터속의 한글, 정보시대사, 1991.

[2] 임순범, "윤곽선 글자꼴의 처리 기술 및 활용 추세", 전자공학회지, 제18권, 제11호, pp.858-865, 1991.

[3] A. Shamir and R. Ari, "Feature-based design of fonts using constraints", Electronic Publishing, Artistic Imaging, and Digital Typography. Springer Berlin Heidelberg, pp.93-108, 1998.

[4] J. R. Laguna, "Hong-Zi: A Chinese METAFONT", Proceeding of TUG, TUGboat, Volume 26, No. 2, pp.125-128, 2005.

[5] 권경재, 손민주, 최재영, 정근호, "METAFONT를 이용한 구조적 한글 폰트 생성기", 정보과학회 컴퓨팅의 실제 논문지, 제22권, 제9호, pp. 449-454, 2016.

[6] 손민주, 권경재, 최재영, 정근호, "사용자 인터렉션을 위한 메타폰트 기반 한글 글꼴 웹 편집기", 한국정보처리학회 춘계학술발표대회 논문집, 제23권, 제1호, pp.861-864, 2016.

[7] D. E. Knuth, Computers & Typesetting Volume C: The METAFONT book, Addison-Wesley Longman Publishing, 1986.

[8] D. Crossland, "Why didn't mEtaFoNt catch on?", Proceeding of TUG, TUGboat, Volume 29, No. 3, pp.418-420, 2008.

[9] T. Leming, J. Rossum, and E. Bloklant, Unified Font Object(UFO), Available: www.unifiedfontobject.org/ (Retrived 2016, Aug. 1).