

날씨에 따른 한글 폰트 윤곽선 벡터 변형 알고리즘

박동연[†], 조세란^{**}, 김남희^{***}, 임순범^{****}

Hangul Font Outline Vector Modification Algorithm According to Weather Information

Dong-Yeon Park[†], Se-Ran Jo^{**}, Nam-Hee Kim^{***}, Soon-Bum Lim^{****}

ABSTRACT

Recently, research on various font designs has been actively conducted to deliver effective emotional information in a digital environment. In this study, we propose a Hangul font outline vector modification algorithm that effectively conveys sensitivity according to weather information and can be transformed immediately. The algorithm performs a series of transformations: sets outlines according to design pattern templates, calculates the glyph's position to reflect physical rules, splits outline segments into smaller sizes and deforms the outlines. Through this, we could create several vector font designs such as humidity, cloud, wind, and snow. The usability evaluation was close to good, so it can be used in diverse ways if we improve readability and effective design expression.

Key words: Vector Font, Outline Modification Algorithm, Weather Font, Font Design

1. 서 론

디지털 기술이 발전함에 따라 우리는 다양한 디자인의 디지털 폰트를 사용하여 정보를 더욱 간결하고 감성적으로 전달할 수 있게 되었다[1]. 문자를 단순히 언어의 표현 수단만이 아닌 시각적 유희의 수단, 개성 표현의 수단, 더 나아가 비언어적 의미를 내포한 정보 전달 매체로 사용할 수 있게 된 것이다[2,3]. 이처럼 여러 용도와 폰트 디자인에 대해 증가하는 사용자의 수요를 충족시키기 위하여 폰트 디자인을 주제로 한 다양한 연구들이 진행되고 있다[4,5].

디지털 폰트는 디자인 방식에 따라서 크게 비트맵 방식과 벡터 방식으로 나눌 수 있다. 비트맵 방식은 래스터 형식의 이미지와 같이 픽셀로 글자의 형태를 나타내는 방식으로, 글자를 확대하거나 기울이는 등의 조작용을 가하면 외곽선이 거칠어지고 계단 현상 등이 나타난다는 단점이 있다. 하지만 비교적 조작용이 쉽다는 장점이 있기 때문에 래스터 방식을 이용하여 폰트 디자인과 관련된 다양한 연구들이 진행되었다. 특히 최근에 각광받고 있는 딥러닝 기술을 이용하여 폰트 생성 관련 연구도 활발히 진행되었다. 대표적으로 손글씨 이미지를 통해 글자의 디자인적

※ Corresponding Author : Se-Ran Jo, Address: (04310) Cheongpa-ro 47-gil 100, Yongsan-gu, Seoul, Korea, TEL : *** - **** - ***** FAX : +82-504-299-5779, E-mail : shooshoo329@sookmyung.ac.kr

Receipt date : Jun. 13, 2022, Revision date : Jul. 25, 2022
Approval date : Aug. 25, 2022

[†] Dept. of IT Engineering., Graduate School, Sookmyung Women's University
(E-mail : yeon0729@sookmyung.ac.kr)

^{**} Dept. of IT Engineering., Graduate School, Sookmyung Women's University

^{***} Dept. of IT Engineering., Graduate School, Sookmyung Women's University
(E-mail : kelly9455@sookmyung.ac.kr)

^{****} SookMyung Women's University
(E-mail : sblim@sm.ac.kr)

※ This research was supported by Basic Science Research Program (2021R111A4A01059550) through the National Research Foundation of Korea(NRF) funded by the Ministry of Education.

특징을 분석하고, 이를 기반으로 새로운 글자를 만들어내는 연구가 진행된 바 있다[5]. 이 외에도 스타일 변형 기반으로 폰트를 생성하거나 폰트별로 특징을 추출하여 이를 혼합해 새로운 폰트 디자인을 생성하는 연구도 진행되었다[6]. 벡터 방식은 글자의 윤곽선 정보를 숫자의 배열로써 저장하는 방식으로, 글자에 조각을 가하더라도 윤곽선의 위치 정보를 즉시 계산하여 조정하기 때문에 윤곽선이 부드럽게 유지된다는 장점이 있다[3,7,8]. 이는 폰트를 디자인할 때 윤곽선의 위치 정보를 계산하는 것이 어렵다는 단점이 있지만, 상황 정보에 따라서 자유롭게 디자인을 수정해서 생동감 있게 감성을 전달할 수 있다는 가능성도 잠재하고 있다.

본 연구에서는 한글 벡터 폰트의 윤곽선을 상황 정보에 따라 즉시 변형하여 정보를 더욱 직관적으로 전달하고, 각 정보가 대표하는 감성을 효과적으로 전달할 수 있는 한글 폰트 윤곽선 벡터 변형 알고리즘을 제안하고자 한다. 이 때, 상황 정보는 각 감성을 뚜렷하게 구분할 수 있고, 수치적으로도 범위의 차이가 분명한 날씨 정보를 이용하여 설계하고자 한다.

2. 관련 연구

2.1 폰트의 윤곽선 벡터 표현

일반적인 벡터 폰트는 글자의 윤곽선을 여러 조각, 즉 글리프(Glyph)로 나누어 각각을 직선, 원호, 곡선 등으로 표현하는 폰트를 의미한다. 이러한 각 글리프들은 정보의 나열인 벡터 형식으로 위치 정보를 저장한다. 이 때 각 곡선, 즉 글리프들은 베지어 곡선(Bezier Curve)으로 표현한다. 베지어 곡선이란 곡선의 시작점과 끝점, 그리고 그 사이에 위치하는 내부 제어점으로 곡선을 표현하는 방식이다. 제어점의 수가 늘어날수록 더욱 섬세한 곡선 표현이 가능하지만, 그만큼 계산이 복잡해지기 때문에 너무 많은 수의 제어점을 사용하지는 않는다[9]. 근래에는 내부 제어점이 1개인 2차 베지어 곡선과 내부 제어점이 2개인 3차 베지어 곡선으로 대부분의 곡선 및 디자인을 표현할 수 있기 때문에, 3차 베지어 곡선까지만 이용하는 경향이 있다.

주로 사용되는 벡터 폰트로는 TTF(TrueType Font)와 OTF(OpenType Font) 포맷이 있다. TTF는 애플과 마이크로소프트사가 개발한 폰트로, 2차

베지어 곡선 방식을 사용한 폰트이며 Windows 운영체제에서 주로 사용된다. OTF는 어도비와 마이크로소프트사가 개발한 폰트로, 3차 베지어 곡선 방식을 사용하며 모든 환경에서 동일한 모양을 만들기 위해 개발되었다. OTF 포맷은 TTF 포맷보다 섬세한 곡선 표현이 가능하지만 계산 과정이 2차원보다 복잡하다는 단점이 있다.

2.2 폰트 스타일 자동 변형 연구

새롭게 디자인한 폰트 스타일을 모든 글자에 다 적용하기 위해선 많은 시간과 인력이 필요하다. 이러한 문제점을 해결하기 위해 폰트의 스타일 자동 변형 연구가 많이 진행되고 있다. 폰트 스타일 자동 변형 연구에는 폰트 내부 스타일 변형, 스켈레톤 기반 변형, 윤곽선 변형 등이 있다.

우선 폰트 내부 스타일 변형의 경우 글자의 윤곽선 내부 디자인을 변형하는 방법으로, 딥러닝 신경망을 이용한 래스터 이미지 기반 스타일 변형 연구가 주를 이룬다[5,6,10]. 하지만 이는 학습 모델을 만들기 위한 다량의 데이터와 시간이 필요하며, 확대 등의 글자 변형 시 계단 현상 등이 나타나 깔끔한 디자인 표현이 어렵다는 한계점이 있다. 이러한 문제를 해결할 수 있는 벡터 기반 글리프 스타일 변형에 관한 연구도 진행되고 있다. 벡터 기반 스타일 변형 방법에는 스켈레톤 기반 변형과 윤곽선 변형 방법 등이 포함되며, 주로 글리프를 개별 세그먼트 또는 영역으로 구분하여 각각 매핑된 스타일을 적용하는 방식으로 진행된다[11].

스켈레톤 기반 변형 연구의 경우, 주로 글리프의 중앙 골격을 추출하여 이를 기반으로 스타일을 변형하는 방식으로 진행된다. 이는 읽기 쉽고 일관성 있는 서체를 생성할 수 있다는 점과 폰트에 더욱 다양한 변형을 적용할 수 있는 기본 서체를 만들 수 있다는 점에서 이점을 가진다[12]. 윤곽선 변형 연구는 글리프의 윤곽선을 추출한 후 윤곽선의 좌표 정보를 수정하여 디자인을 변형하는 방식으로 진행된다. 윤곽선 변형 연구는 주로 알고리즘을 이용하여 윤곽선의 일부에 기하학적 변형을 주는 방식으로 진행되는 데, 최근에는 물리 법칙을 수식으로 변환하여 일부 윤곽선의 좌표를 변형하여 디자인을 적용하는 연구도 진행된 바가 있다[13]. 본 연구에서는 TTF 포맷의 한글 벡터 폰트를 대상으로 윤곽선 변형을 통해

다양한 폰트 디자인을 생성하는 알고리즘을 제안하고자 한다.

3. 폰트 벡터화를 통한 디자인 알고리즘 및 구현 결과[14]

3.1 연구 목표

본 연구에서는 픽셀 이미지처럼 계단 현상이 발생하는 기존의 래스터 이미지 디자인 방식보다, 자유롭고 동적으로 디자인을 변형할 수 있는 방식을 적용하고자 한다. 이를 위해 폰트의 윤곽선 데이터를 벡터 형태로 추출하고, 다양한 디자인 변형 알고리즘을 이용해 윤곽선의 모양을 조정하는 방법을 사용하였다. 이러한 방법은 디자인을 수치에 따라서 유동적으로 변화시킬 수 있을 때 더욱 효과적으로 그 감성을 전달할 수 있고, 날씨 정보는 수치에 따라서 그 감성과 정보를 효과적으로 보여줄 수 있는 데이터 중 하나이다. 본 연구에서는 한글의 폰트 정보를 벡터 데이터 형태로 추출하여 기상 정보의 감성과 정보를 잘 나타낼 수 있도록 하는 벡터 변형 알고리즘을 제안한다.

3.2 알고리즘 개요

한글 폰트의 윤곽선 정보를 벡터 데이터로 추출하고 변형하는 프로세스는 위 Fig. 1과 같이 나타낼 수 있다. 본격적인 디자인 변형을 수행하기에 앞서, 디자인하고자 하는 글자와 기본 폰트를 선정한다. 이때 기본 폰트는 재현 실험의 용이성과 접근성을 보장할 수 있도록 네이버의 나눔스퀘어 폰트(Nanum Square)와 같이 무료로 사용이 가능한 폰트들을 사용하였다.

우선 글자의 고유한 유니코드 번호와 폰트 변형을

위한 다양한 오픈 소스 라이브러리를 함께 사용하여, 디자인하고자 하는 글자의 글리프셋을 불러온다. 유니코드는 다양한 문자를 컴퓨터에서 표현할 수 있도록 하는 표준 코드번호이다. 따라서 디자인하고자 하는 문자가 어떤 것이든 상관없이 본 프로세스에 적용 가능하다는 장점이 있다. 오픈 소스 라이브러리로는 Flat 라이브러리[15]와 Bezmerizing 라이브러리[16]를 사용하였다. 이를 통해 불러온 글리프셋은 하나의 글자를 표현하기 위한 다양한 글리프의 묶음을 의미한다. 글리프셋을 이루는 각 글리프의 방향과 위치 등 물리적 특징을 분석하여, 직접 설계한 다양한 날씨 디자인 패턴, 즉 윤곽선 디자인 변형 알고리즘을 적용함으로써 새로운 날씨 디자인 폰트를 만들 수 있다.

이러한 윤곽선 디자인 변형 알고리즘은 크게 세 가지 단계로 이루어져 있다. 첫 번째는 길이에 따라서 윤곽선 글리프들을 잘라 재구성하는 것이다. 일반적인 폰트에서는 외곽선이 꺾이는 부분을 기점으로 글리프들이 나누어져 있다. 이 상태에서 디자인을 바로 적용하게 된다면 패턴이 길이에 따라 적용되어 디자인이 균일하게 보이지 않을 수 있으므로, 글리프의 길이와 표현하고자 하는 디자인에 따라서 글리프들을 적절한 길이로 잘라 재구성하여 더욱 촘촘하고 균일한 디자인을 표현하도록 한다.

두 번째는 물리 규칙 반영을 위한 글리프 위치 확인이다. 날씨에 따른 감성을 사실적이고 생생하게 표현하기 위해서는 물리 규칙에 맞게 글리프를 변형시켜야 한다. 예를 들어, 눈이 소복하게 쌓인 모습을 보여주기 위해서는 글자의 가장 윗부분에만 쌓이게 해야 한다. 또는, 빗물이 떨어져 물방울이 맺히는 디자인을 표현하기 위해서는 글자의 가장 아랫부분에 물방울이 맺히게 해야한다. 이 외에도 바람이 부는

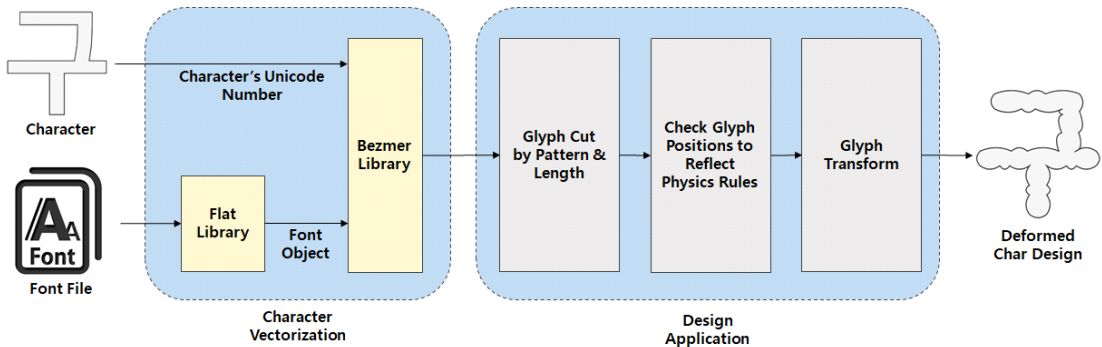


Fig. 1. Process of Hangul font vector modification algorithm.

표현을 하기 위해서는, 바람이 부는 방향의 반대편 글리프에 선이 훑날리는 디자인을 적용해야 한다. 이렇게 글리프의 위치를 파악하기 위해서는 글리프의 방향 정보를 사용할 수 있다. 일반적인 폰트 외곽선 글리프들은 시계 방향을 따라 연결되어 있다. 즉 오른쪽 방향의 글리프는 항상 글자의 위편에 있고, 왼쪽 방향의 글리프는 항상 아래편에 있다. 동일한 맥락으로 아래 방향의 글리프는 항상 글자의 오른쪽에, 위 방향의 글리프는 항상 글자의 왼쪽에 있다. 이를 고려하여 각 글리프의 시작 좌표와 끝 좌표를 계산하면 글리프의 방향, 즉 글리프의 위치를 알 수 있다.

마지막 세 번째 단계는 글리프 변형이다. 이 단계가 윤곽선 디자인 변형 알고리즘에서 가장 핵심적인 역할을 수행하는데, 바로 글리프의 모양을 직접적으로 바꾸는 것이다. 표현하고자 하는 디자인 감성 및 패턴에 따라서 글리프의 유형과 좌표를 변경한다. 직선을 볼록한 곡선으로 변형할 경우, 직선 글리프를 2차 베지어 곡선으로 변경하고, 베지어 곡선의 중간에 있는 제어점의 좌표를 글자 바깥 방향으로 이동시킴으로써 직선을 곡선으로 변형할 수 있다. 이를 표현하고자 하는 디자인 패턴에 따라서 다양한 방법으로 응용 가능하다. 표현하고자 하는 기상 정보의 수치값에 따라 디자인을 극대화하여, 날씨의 감성을 생동감 있게 표현하는 다양한 글자 디자인을 만들 수 있다.

3.3 다양한 디자인 패턴 및 결과

윤곽선 벡터 변형 알고리즘으로 인해 생성되는 모든 디자인은 위 수식으로 표현이 가능하다. 우선 식(1)은 한 글자의 글리프셋을 이루는 글리프의 나열을 나타낸다. 하나의 글리프는 하나의 벡터와 같다. 식(2)부터 식(4)까지는 각 글리프를 이루는 선의 종류와 좌표를 나타내는 식이다. 식(2)는 직선 벡터일 경우이며, V_i^0 는 직선의 시작점의 좌표이고, V_i^1 는 직선의 끝점의 좌표이다. 하나의 좌표에는 x축 기준의 좌표 값과 y축 기준의 좌표 값이 튜플 형태로 함께 저장되어 있다. 식(3)은 2차 베지어 곡선을 표현하는 식으로써, V_i^0 은 곡선의 시작, V_i^1 은 곡선의 제어점, V_i^2 은 곡선의 종료점을 의미한다. 식(4)는 3차 베지어 곡선을 표현하는 것이고, 2차 베지어 곡선과 동일하게 V_i^0 은 곡선의 시작, V_i^1 은 곡선의 첫 번째 제어점, V_i^2 은

곡선의 두 번째 제어점, V_i^3 은 곡선의 종료점을 의미한다.

$$G = \{V_0, V_1, V_2 \dots, V_n\} \tag{1}$$

$$V_i = \{V_i^0, V_i^1\} = \{(x_i^0, y_i^0), (x_i^1, y_i^1)\} \tag{2}$$

$$V_i = \{V_i^0, V_i^1, V_i^2\} = \{(x_i^0, y_i^0), (x_i^1, y_i^1), (x_i^2, y_i^2)\} \tag{3}$$

$$V_i = \{V_i^0, V_i^1, V_i^2, V_i^3\} = \{(x_i^0, y_i^0), (x_i^1, y_i^1), (x_i^2, y_i^2), (x_i^3, y_i^3)\} \tag{4}$$

3.3.1 구름 디자인

구름 디자인은 하늘에 구름이 떠있는 것과 같이 글자의 외곽선이 불규칙적으로 울퉁불퉁한 형태를 띠도록 설계하였다. 우선 섬세한 디자인 적용을 위해서 모든 방향의 글리프들을 길이에 따라서 잘라 재구성하였고, 자른 모든 글리프들을 2차 베지어 곡선으로 변경하였다. 각 글리프의 방향에 따라서 글리프 중간에 위치한 제어점을 바깥 방향으로 이동시킴으로써 울퉁불퉁한 모양을 만들 수 있도록 하였다(식(5) 참조). 아래의 Fig. 2에서는 제어점의 조정값이 달라짐에 따라 디자인 변형 정도도 달라지는 모습을 확인할 수 있다.

$$V_i^1 = \begin{cases} (x_i^1, y_i^1 - r) & (right) \\ (x_i^1, y_i^1 + r) & (left) \\ (x_i^1 - r, y_i^1) & (up) \\ (x_i^1 + r, y_i^1) & (down) \end{cases} \tag{5}$$

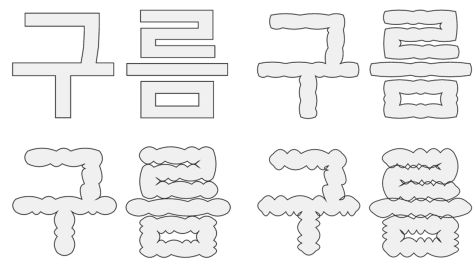


Fig. 2. Variations of cloud design.

3.3.2 눈 디자인

눈 디자인은 눈이 사물 위로 동그랗게 쌓인 모습에서 착안하여 디자인을 설계하였다. 위쪽 글리프에 만 디자인을 적용하기 위해 오른쪽 방향의 직선 글리프만 일정한 길이로 자른 뒤, 각 글리프들을 2차 베지어 곡선으로 변경한다. 적설량 수치에 따라서 각 글리프의 중간 제어점을 위로 올리도록 한다. 이 때 겨

울 느낌을 더해주기 위해서 글자의 오른쪽 아래에 눈사람 모양의 글리프를 임의적으로 추가해주었다.

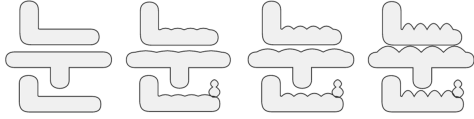


Fig. 3. Variations of snow design.

3.3.3 자외선 디자인

자외선 디자인은 강한 햇살의 느낌을 살릴 수 있도록 조정값이 커짐에 따라 글자가 뾰족해질 수 있도록 디자인하였다. 앞서 설명한 구름 디자인과는 반대로 글리프의 제어점을 안쪽으로 이동시키도록 하였다(식 (6) 참조). 이 때 조정값은 자외선 지수에 따라서 바뀌도록 하였다.

$$V_i^1 = \begin{cases} (x_i^1, y_i^1 + r) & (right) \\ (x_i^1, y_i^1 - r) & (left) \\ (x_i^1 + r, y_i^1) & (up) \\ (x_i^1 - r, y_i^1) & (down) \end{cases} \quad (6)$$

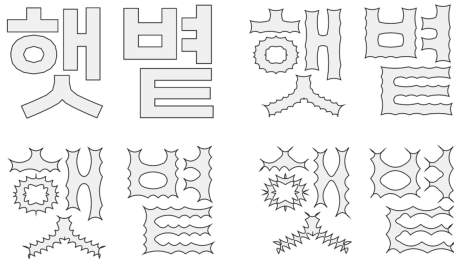


Fig. 4. Variations of ultraviolet ray design.

3.3.4 물방울 디자인

물방울 디자인은 강수량을 나타내기 위한 디자인이다. 비가 오는 감성을 나타내기 위해서 물방울이 글자에 맺힌 느낌으로 설계하였다. 우선 아래쪽 글리프를 의미하는 왼쪽 방향의 직선 글리프들만 짧은 길이로 잘라 재구성한다. 해당 글리프들을 3차 베지어 곡선으로 변경하고, 베지어 곡선의 두 제어점을 서로 교차시켜 베지어 곡선이 고리 모양을 그릴 수 있도록 한다(식 (7) 참조). 이 과정을 다양한 크기로 반복해서 수행하면 물방울이 맺힌 느낌을 줄 수 있다. 강수량 값의 변화에 따른 글자 디자인 변화는 Fig. 4과 같다.

$$\begin{cases} V_i^1 = (x_i^1 - r, y_i^1 + r) \\ V_i^2 = (x_i^2 + r, y_i^2 + r) \end{cases} \quad (left) \quad (7)$$

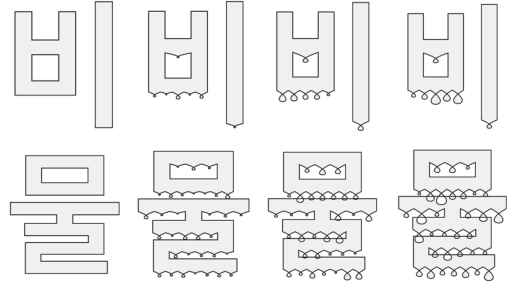


Fig. 5. Variations of waterdrop design.

3.3.5 습도 디자인

습도 디자인은 기상 정보 중 습도 수치를 나타내기 위한 디자인이다. 습도가 높을 때 축축하고 불쾌한 느낌을 살릴 수 있도록 디자인하였다. 이를 위해서 각 글리프 뒤편의 아래쪽에 위치해있는 글리프를 구불구불하게 흘러내리는 모양으로 나타낼 수 있도록 하였다. 왼쪽 방향으로 이동하는 모든 글리프들을 자갈하게 자른 뒤, 제어점을 차례로 변경함으로써 울룩불룩한 디자인을 만들었다(식 (8) 참조).

$$V_i^1 = (x_i^1, y_i^1 + r) \quad (left) \quad (8)$$

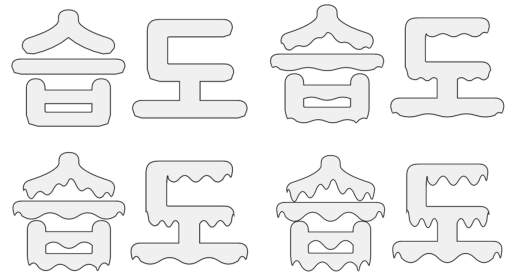


Fig. 6. Variations of humidity design.

3.3.6 바람 디자인

바람 디자인은 일기예보에서 풍속 수치에 따라 디자인을 변경 및 시각화하기 위한 디자인이다. 바람이 부는 것에 따라서 물체가 한쪽으로 훑날리는 것을 본따 디자인을 설계하였다. 단, 풍향은 고려하지 않고 풍속만 고려하였다. 이를 위해서 위로 올라가는 직선을 아주 짧은 길이로 자갈하게 자르고 뾰족한 모양으로 튀어나오게 할 수 있도록 각 글리프의 좌표

값을 조정하였다(식 (9) 참조). 풍속 수치에 따라 변화하는 디자인은 다음 Fig. 7과 같다.

$$\begin{cases} \begin{cases} V_i^0 = (x_i^0, y_i^0) \\ V_i^1 = (x_i^1 - r, y_i^1) \end{cases} & (i \bmod 2 = 0) \\ \begin{cases} V_i^0 = (x_i^0 - r, y_i^0) \\ V_i^1 = (x_i^1, y_i^1) \end{cases} & (i \bmod 2 = 1) \end{cases} \quad (up) \quad (9)$$

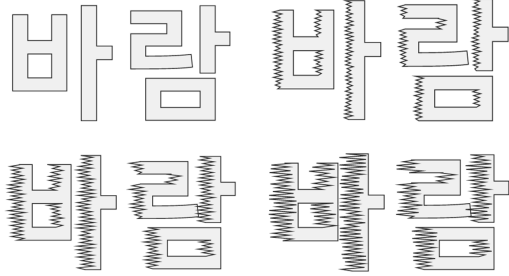


Fig. 7. Variations of wind design.

3.3.7 먼지 디자인

최근의 기상 정보에는 통상적인 날씨 정보와 함께 구체적인 미세먼지 수치도 함께 제공된다. 먼지 디자인은 이러한 미세먼지 수치에 따라서 시야가 뿌옇거나 흐릿하게 보이는 느낌을 살리고자 디자인하였다. 앞서 설명한 구름 디자인처럼 선의 모양을 변경하여 디자인한 방법과는 달리, 먼지 디자인은 외곽선 글리프 길이에 따라서 새롭게 반투명한 원을 그리도록 하였다(식 (10) 참조).

$$\begin{cases} V_i^0 = (x + r, y - dy, x + dx, y - r, x, y - r) \\ V_i^1 = (x - dx, y - r, x - r, y - dy, x - r, y) \\ V_i^2 = (x - r, y + dy, x - dx, y + r, x, y + r) \\ V_i^3 = (x + dx, y + r, x + r, y + dy, x + r, y) \end{cases} \quad (10)$$

$$(k = 4.0 \div 3.0 \times (\sqrt{2.0} - 1.0))$$

$$(dx = dy = r * k)$$

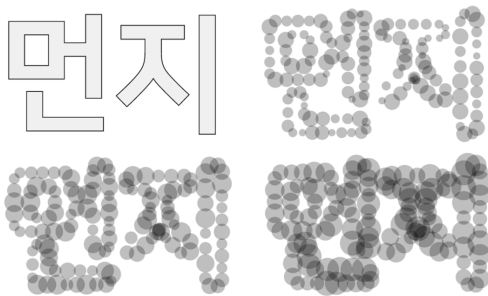


Fig. 8. Variations of dust design.

3.3.8 아지랑이 디자인

아지랑이는 햇빛이 강하게 쬐어 지면이 뜨겁게 달구어졌을 때, 공기가 공중에서 아른아른 움직여 먼 풍경이 지면 근처에서 아른거리며 보이는 현상을 의미한다. 이는 자외선처럼 구체적인 수치는 아니지만 더위로 인해 지표면 시야가 아른거리는 디자인을 효과적으로 보여줄 수 있을 것으로 예상하여 디자인하였다. 각 세로축과 가로축의 변형에 따른 디자인 변화는 아래 Fig. 9와 같다. 본 디자인에서는 각 글리프를 이루는 정점의 좌표값 중 x좌표는 sin함수를, y좌표는 cos함수를 적용한 후, 조정값을 곱하는 과정을 정점마다 일관적으로 적용하여 디자인을 변경하였다. 이는 다음 식(11)과 같이 정의할 수 있다.

$$V_i = (x_i^0 + \sin(x_i^1 \times 0.01) \times n, y_i^1 + \cos(y_i^0 \times 0.01) \times n) \quad (11)$$

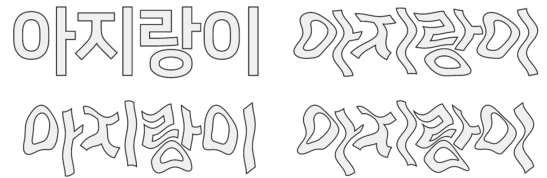


Fig. 9. Variations of haze design.

4. 사용성 평가

4.1 사용성 평가 설계

폰트 벡터화를 통한 다양한 글자 디자인 변형 알고리즘이 목적에 적합하게 설계 및 구현되었는지 파악하기 위해서 사용성 평가를 진행하였다. 사용성 평가에서는 각 폰트 디자인에 대한 효과성, 효율성, 만족도를 파악하고자 하였다. 이를 위해서 각각 ‘디자인적 의미가 잘 전달이 되는가?’, ‘폰트로써 사용할 수 있는가’, ‘폰트에 대한 만족도는 어떻게 되는가?’라는 질문에 대해서 5점 리커트 척도로 응답하도록 하였다. 전반적인 윤곽선 변형 알고리즘을 통한 디자인 만족도에 대해서도 5점 리커트 척도를 사용하여 응답하도록 하였다. 이 외에도 평가 참여자들의 질적인 경험을 분석하기 위해 가장 만족한 폰트 디자인과 그 이유, 본 알고리즘 및 연구에서 개선이 필요한 점을 서술할 수 있도록 하였다.

Table 1. Usability evaluation items and contents.

Evaluation Item	Question Content	Foramt
Effectiveness	Is the design meaning well communicated?	5 Likert
Efficiency	Is it suitable for use as a design font?	5 Likert
Satisfaction	How satisfied are you with the font?	5 Likert

4.2 평가 결과

사용성 평가에는 20대부터 50대 이상까지 다양한 연령대로 이루어진 평가 참여자 100명이 참여하였다. 우선 평가 참여자들의 평균적인 폰트 관심도를 5점 리커트 척도를 사용하여 조사한 결과, 보통(3)인 것으로 확인하였다. 단 평가 참여자의 연령대와 성비 중 20대 여성이 약 75%가량 차지하고 있는 것이 데이터 비율의 동일하게 유지하지 못한 것에 대해서 아쉬움을 찾을 수 있다.



Fig. 10. Usability evaluation results for each design of the contour design modification algorithm. (a) Effectiveness, (b) Efficiency, and (c) Satisfaction.

전반적인 사용성 평가는 Fig. 10과 같다. 각 폰트 디자인에 대해서 효과성, 효율성, 만족도를 5점 리커트 척도를 사용하여 조사한 결과, 습도 디자인이 모

든 측면에서 각각 3.94점, 3.78점, 3.68점으로 가장 높은 점수가 나온 것을 확인하였다. 다음으로 높은 점수를 받은 디자인은 구름 디자인이다. 습도 디자인과 같이 전반적인 항목에서 모두 높은 점수를 보이거나, 습도 디자인에 비해서 약간 점수가 낮은 것을 볼 수 있다. 이에 대한 구체적인 사용자의 의견으로는, 구름의 몽글몽글한 디자인을 잘 살려주었으나, 글자의 크기가 작아질수록 표현력이 떨어질 것 같아 아쉽다는 의견이 있었다. 그 다음으로는 바람 디자인과 눈 디자인이 비슷한 점수를 보여준다. 다만 바람 디자인은 효과성 측면에서 눈 디자인보다 점수가 높고, 눈 디자인은 바람 디자인보다 효율성과 만족도 측면에서 약간 점수가 더 높은 것을 확인할 수 있다. 다만 눈 디자인에서는 구름 디자인과 비슷하다는 의견이 있었고, 바람 디자인은 전달력이 좋으나 뾰족한 모양이 너무 강조되어 번개나, 태풍 등에 가까워 보인다는 의견이 있었다.

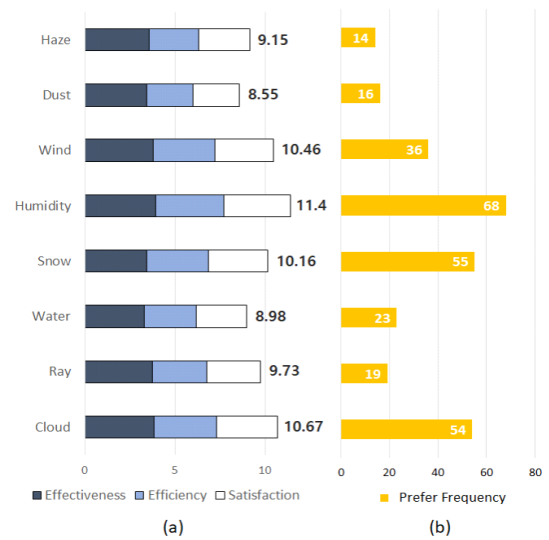


Fig. 11. Comparison between usability evaluation results by design and preferred font frequency results. (a) accumulate sum of usability evaluation and (b) prefer frequency.

Fig. 11은 평가 참여자별로 가장 만족한 폰트 3가지를 선정하였을 때의 도수 결과와 앞서 설명한 폰트별 효과성, 효율성, 만족도 결과를 합산했을 때를 비교한 그래프이다. 가장 만족한 폰트에서는 습도, 눈, 구름, 바람 디자인 순서로 가장 높은 수치를 보여준다. 이는 앞서 평가한 개별 항목들의 합한값과도 유

사한 추이를 보이는 것을 확인할 수 있다. 비교적 낮은 선호도를 보이는 4가지 디자인 항목도 동일한 것을 볼 수 있다. 이를 통해서 개별 수치 및 선호 폰트 조사가 서로 동일한 결과를 보여주어 사용성 평가가 일관된 결과를 제시하고 있음을 알 수 있다. 선호하는 폰트 디자인을 선정한 근거도 함께 조사하였는데, 주로 날씨의 특징이 잘 드러나거나, 실제로 사용하기 좋을 것 같아서 혹은 디자인이 마음에 들어서라는 의견이 대다수였다. 이 외에도 디자인이 독특하고 참신하거나, 일관적으로 잘 적용이 되었기 때문이라는 의견도 존재하였다.

윤곽선 벡터 알고리즘의 개선점을 조사하면서 공통적으로 언급된 사항을 확인하였다. 이는 가독성에 관련된 것이었는데, 본 논문에서 제안한 디자인 알고리즘이 날씨의 감성과 디자인 변형 특징을 중점적으로 강조하다 보니 가독성을 충분히 고려하지 못한 것으로 인한 결과라고 추정한다. 이는 아지랑이, 먼지, 물방울, 자외선 디자인이 낮은 점수를 받은 것에 대해서도 관련성이 있는 것을 확인하였다. 이에 대해서 구체적인 평가 참여자들의 의견을 확인해보았을 때, 특히 먼지는 외곽선이 정확하게 드러나지 않아서, 아지랑이는 글자가 너무 구부러져 있어서 가독성이 떨어질 것 같다는 의견이 있었다. 특히 단어가 아니라 문장 단위로 디자인을 평가한다면 더 가독성이 떨어진다는 의견이 공통적으로 등장하는 것을 확인하였다.

5. 결 론

본 논문에서는 한글 벡터 폰트의 윤곽선을 날씨 정보에 따라 변형하여 정보를 더욱 직관적으로 전달하고, 각 정보가 내포하는 감성 또한 효과적으로 전달할 수 있는 한글 폰트 윤곽선 벡터 변형 알고리즘을 제안하였다. 디자인 변형 과정은 다음과 같다. 먼저, 기본이 될 폰트를 선정하고 글자 글리프셋을 불러온 후 설계한 날씨 디자인 패턴, 즉 윤곽선 디자인 변형 알고리즘을 적용한다. 알고리즘은 크게 세 단계를 거쳐 적용된다. 먼저, 디자인이 균일해 보일 수 있도록 글리프의 길이에 따라 적절한 길이로 분할하여 재구성한다. 다음으로 물리 규칙을 반영할 수 있도록 글리프의 위치를 찾아낸다. 이는 벡터 폰트의 글리프가 시계방향으로 연결되어 있다는 특징을 이용한다. 마지막으로 글자의 윤곽선에 각 날씨의 특징

을 나타낼 수 있게 설계한 디자인을 적용한다. 이는 표현하고자 하는 디자인 감성 및 패턴에 따라서 글리프의 유형과 좌표를 변형하는 것으로 반영한다. 날씨 디자인 패턴은 구름, 눈, 자외선, 물방울, 습도, 바람, 먼지, 아지랑이로 총 8가지의 패턴을 설계하였다.

본 논문에서 제시한 디자인 변형 알고리즘이 목적에 맞게 설계되었는지 확인하기 위해 사용성 평가를 진행하였다. 사용성 평가는 각 디자인에 대해 효과성, 효율성, 만족도 파악을 목적으로 하였고, 각각 ‘디자인적 의미가 잘 전달이 되는가?’, ‘폰트로써 사용할 수 있는가’, ‘폰트에 대한 만족도는 어떻게 되는가?’라는 질문에 대해 5점 리커트 척도로 조사하였다. 평가 결과 습도, 구름, 바람, 눈이 전반적으로 높은 점수를 받았고, 가장 만족한 폰트 선정 결과와 비교하였을 때 유사한 결과를 보여 평가 결과의 일관성을 확인할 수 있었다. 전체 디자인의 평균 만족도가 3.58점으로 디자인이 전반적으로 잘 적용되었다는 확인할 수 있었고, 특히 그중 상위 네 가지 디자인인 습도, 구름, 바람, 눈은 점수가 높아 바로 사용이 가능할 것으로 보인다. 만족스러운 폰트 선정 이유에 대한 답변을 통해 만족도 하위 네 가지 디자인의 경우 해당 날씨의 특징이 잘 드러나지 않았거나 가독성이 낮음을 이유로 낮은 점수를 받았음을 알 수 있다. 정보의 특징이 잘 드러나고 가독성이 보장되는 것이 디자인 폰트 선정에 있어 가장 중요한 요인이기 때문이다.

후속 연구에서는 만족도 점수 하위 디자인의 가독성을 높이기 위해 변형 정도를 줄이거나, 윤곽선을 뚜렷하게 하고, 폰트 주변에 음영을 넣는 방식 등으로 개선을 진행할 수 있을 것이다. 물방울과 자외선 디자인의 경우는 날씨의 특징 및 감성이 직관적으로 전달되지 않는다는 의견이 있었기 때문에 전반적인 디자인 수정이 함께 이루어져야 할 것으로 보인다. 이렇게 날씨의 특징을 더 직관적으로 나타내고 가독성을 보장할 수 있는 폰트 디자인을 다양하게 설계할 수 있다면, 디자인 폰트를 더욱 효과적이고 효율적으로 생성하는 데 도움이 될 것으로 보인다. 또한 이러한 디지털 벡터 폰트는 유튜브(Youtube)와 같은 영상 콘텐츠 플랫폼에서도 효과적으로 사용될 수 있을 것이다. 디지털 벡터 폰트는 상황과 감성에 따라 자유로운 변형이 가능하기 때문에, 콘텐츠 산업 속 다양한 목적과 의도에 맞게 활발히 적용이 가능하다는

강점이 있다. 특히 벡터 데이터를 기반으로 비교적 쉽게 동적인 표현이 가능하다는 점에서, 콘텐츠 소비자들에게 다양한 감성 및 감정을 생동감 있게 전달할 수 있을 것이다.

REFERENCE

- [1] Y.D. Moon, "Effective Multimedia Contents Design by New Font Development," *The Journal of Korean Institute of Information Technology*, Vol. 11, No. 7, pp. 199-204, 2013.
- [2] M.J. Son and H.J. Lee, "Emotion Communication through Motion Typography Based on Movement Analysis," *Journal of Digital Contents Society*, Vol. 12, No. 4 pp. 541-550, 2011.
- [3] Y.D. Moon, "Digital Font Design for Effective Web Contents Description," *The Journal of Korean Institute of Information Technology*, Vol. 8, No. 1, pp. 133-138, 2010.
- [4] D.Y. Park, J.Y. Jeon, S.Y. Lim, and S.B. Lim, "A Study on Influence of Stroke Element Properties to Find Hangul Typeface Similarity," *Journal of Korea Multimedia Society*, Vol. 23, No. 12, pp. 1552-1564, 2020.
- [5] J.Y. Jeon, J.Y. Yang, and S.B. Lim, "Implementation of Automatic Hangul Handwriting Production Service Using Deep Learning," *Proceedings of HCI Korea 2019*, pp. 1155-1157, 2019.
- [6] J.S. Lee and H.C. Choi, "Font Generation using Style Transfer," *Proceeding of Institute of Electronics and Information Engineers*, pp. 1934-1937, 2020.
- [7] S.B. Lim, "Processing Techniques and Utilization Trends for Contour Font," *The Journal of Korea Institute of Electronics Engineers*, Vol. 18, No. 11, pp. 69-76, 1991.
- [8] Y.C. Kim and U.C. Kim, "Improvement of the Readability for Text using Graphic Software - Laying Stress on Anti-Aliasing in Digital Media," *Journal of the Korea Contents Association*, Vol. 8, No. 12, pp. 141-150, 2008.
- [9] King Sejong Memorial Society and Hangul Font Development Institute, *Korean Font Dictionary*, King Sejong Memorial Society, Seoul, 2011.
- [10] W. Wang, J. Liu, S. Yang, and Z. Guo, "Typography with Decor: Intelligent Text Style Transfer," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5889-5897, 2019.
- [11] P.K. Dhanuka, N. Kumawat, and N. Jindal, "Vector based Glyph Style Transfer," *Special Interest Group on Computer Graphics and Interactive Techniques Conference*, No. 21, pp. 1-2, 2019.
- [12] F.A. Pereira, T. Martins, S. Rebelo, and J. Bicker, "Generative Type Design: Creating Glyphs from Typographical Skeleton," *Proceeding of 9th International Conference on Digital and Interactive Arts*, pp. 1-8, 2019.
- [13] N. Kumawat, "Automated Physics Based Animation of Fonts," *Special Interest Group on Computer Graphics and Interactive Techniques Conference*, No. 6, pp. 1-2, 2020.
- [14] DyeonPark/WeatherFont-with-Vector: Modification Algorithm of Hangul Font Contour Vector according to Weather, <https://github.com/DyeonPark/WeatherFont-with-Vector> (accessed June 8, 2022).
- [15] Flat - Xxyxyz, <https://xxyxyz.org/flat> (accessed June 8, 2022).
- [16] Aparrish/bezmerizing: a tiny quirky library with some bezier curve functions, <https://github.com/aparrish/bezmerizing> (accessed June 8, 2022).



박 동 연

2021년 숙명여자대학교 학사
2021년~현재 숙명여자대학교 대학원(공학석사)
관심분야: 컴퓨터 비전, 인공지능, UI/UX, HCI, 장애인 접근성 등



김 남 희

2022년 숙명여자대학교 학사
2022년~현재 숙명여자대학교 대학원(공학석사)
관심분야: 게임, 게임 보안, XR, 메타버스 등



조 세 란

2021년 숙명여자대학교 학사
2021년~현재 숙명여자대학교 대학원(공학석사)
관심분야: UI/UX, 웹 접근성, 장애인 접근성, HCI, 웹/모바일 멀티미디어 응용 등



임 순 범

1982년 서울대학교 학사
1983년 한국과학기술원 석사
1992년 한국과학기술원 박사
1989년~1992년 ㈜휴먼컴퓨터 창업(연구소장)
1992년~1997년 ㈜삼보컴퓨터 프린터개발부 부장
1997년~2001년 건국대학교 컴퓨터과학과 교수
2001년~현재 숙명여자대학교 IT공학과 교수
2015년~현재 숙명여자대학교 ICT융합연구소 연구소장
2006년 University of Colorado 방문교수
2012년~현재 한국전자출판학회 회장, 명예회장
2014년~현재 한국멀티미디어학회 회장, 명예회장
관심분야: 컴퓨터 그래픽스, 웹/모바일 멀티미디어 응용, 디지털 포용, 전자출판(폰트, 전자책, XML 문서), User Interface