

## 트루타입폰트 기반 자동 획 분할

장현규<sup>o</sup>, 구상옥, 정순기

경북대학교 컴퓨터 공학과

{seirion<sup>o</sup>, sokoo}@vr.knu.ac.kr, skjung@knu.ac.kr

### Automatic Stroke Extraction Based on TrueTypeFont

Hyun Gyu Jang<sup>o</sup>, Sang Ok Koo, Soon Ki Jung

Department of Computer Engineering, Kyungpook National University

#### 요 약

이 논문에서는 트루타입 폰트(TrueTypeFont: TTF)로부터 한자의 외곽선 정보를 얻어 자동으로 획을 분리해 내는 방법을 제안한다. 트루타입 폰트는 각 글자에 대한 외곽선 정보를 벡터 형식으로 저장하고 있다. 저장된 벡터의 방향과 위치 정보를 사용하여 한자의 획을 하나씩 분리해 낸 후, 그것을 조합하면 효율적으로 한자 학습 콘텐츠를 완성할 수 있다. 제안한 방법으로 완성된 콘텐츠는 트루타입 폰트가 갖고 있는 벡터 이미지로서의 특성을 그대로 취할 수 있다는 장점이 있다. 제안한 방법으로 획을 분리하고 분리된 획을 순서대로 조합하여 보다 양질의 학습 콘텐츠를 제작할 수 있을 것으로 기대한다.

#### 1. 서론

디지털 미디어의 발달로 수많은 양의 지식들이 새롭게 뉴미디어로 옮겨가고 있다. 많은 사람들이 이런 멀티미디어의 장점을 인식하고 최대한 많은 지식들을 디지털화 하는 데 많은 노력을 기울이고 있다. 다양한 미디어를 이용한 학습 콘텐츠(contents)는 기존의 미디어를 이용하는 것에 비해 훨씬 효율적이고 뛰어난 학습 효과를 얻을 수 있기 때문에 그것들을 인터넷이나 CD-ROM 과 같은 미디어를 통해 사용자에게 널리 서비스하고 있다.

한자 학습의 경우, 글자 하나하나에 대한 소리와 뜻 외에도 애니메이션을 통해 획이 쓰여지는 순서나 방향을 익히는 데 새로운 미디어가 매우 효율적이라는 사실을 알고 있다. 인터넷이나 모바일 단말기를 이용하여 사용자가 원한다면 언제든지 자유롭게 이러한 학습 도구들을 이용할 수 있으며, 이미 많은 콘텐츠 사업자들이 많은 학습 물들을 생산해 내고 있다. 그러나 한자 학습 콘텐츠를 생산하는 데 있어서 가장 걸림돌이 되는 것은 그 자체의 방대한 양이다. 글자 하나하나가 의미가 있는 한자는 일일이 모든 글자에 대하여 학습 내용을 편집하고 변형하여야 한다. 특히, 한자의 획을 순서대로 배치하는 것과 획 방향대로 글자가 쓰여지는 효과를 주기 위해서는 사람의 손으로 일일이 내용을 지정해 주어야 한다.

이 논문에서는 트루타입 폰트 내의 글자 정보를 이용하여 번거로우면서도, 방대한 양의 작업들을 자동화하는 방법에 대하여 소개한다. 트루타입 폰트의 벡터 정보는 획을 이루기 위한 외곽선 정보를 가지고 있으므로, 그것들의 규칙을 찾아내어 효과적으로 획을 분리해 낼 수 있다. 2절에서 관련 연구에 대하여, 3장은 트루타입 폰트로부터 획을 분리해 내는 방법을 자세히 다루고, 4장에 그 결과 분석한다. 끝으로 5장에서 결론 및 향후 연구 계획에 대해 기술한다.

#### 2. 관련 연구

이번 절에서는 기존에 소개된 다른 한자 학습 콘텐츠에 대해 소개하고 그것들의 문제점을 살펴보고, 이 논문에서 제안한 방법이 기존 한자 학습 콘텐츠의 제작의 문제점을 어떠한 방법으로 해결할 지에 대해 서술한다.

#### 2-1 기존 한자 학습 콘텐츠 소개와 문제점

기존의 많은 한자 학습 콘텐츠들은 gif 애니메이션을 사용한 것이 대부분이다[1][2]. 이것은 간단히 이미지 편집만을 이용하는 것이므로, 가장 많이 사용되는 방법이다. 하지만 모든 글자에 대해 획 수만큼 이미지를 손수 다루는 것은 굉장히 노력이 많이 드는 작업이며, 효과적으로 획의 진행방향을 보여 줄 수 없는 단점이 있다. [4]는 획이 진행되는 과정을 보다 부드러운 애니메이션으로 재생한다. 이것은 획의 방향을 고려하여 부드러운 애니메이션으로 만든 것으로 가시적인 효과가 있으므로 학습효과는 훨씬 높다. 그러나 이 콘텐츠 또한 사용자가 직접 글자의 획에 대해 편집해야만 하는 번거로움이 여전히 존재한다. [3]은 효율적인 콘텐츠 개발을 위한 개발 툴을 제안하였다. 사용자가 글자에 대한 획 순서와 방향에 대한 정보를 입력하며, 그것을 이용하여 획 애니메이션을 생성해 낸다. 그러나 이 방법은 기존의 방법 보다는 유용하지만 여전히 사람의 입력이 필요하므로 콘텐츠 제작에 많은 시간을 소비하게 된다.

#### 2-2 제안한 방법

이 논문에서는 한자 학습 콘텐츠를 제작하는 데 매우 효율적인 방법을 제시한다. 트루타입 폰트의 글자 코드를 입력으로 받으면, 글자에 대한 외곽선 정보를 추출하고, 외곽선을 이루는 벡터들의 규칙을 이용하여 자동으로 획을 추출한다. 추출된 획은 트루타입 폰트내의 원래 글자의 부분집합이며, 따라서 모든 획들을 조합하면 완성된 글자를 만들어 낼 수 있다. 이는 한자의 획 순서와 획 방향 정보를 생성하기 위해 일일이 이미지를 편집하는 수고로움을 덜어줄 것이다.

#### 3. 방법

본 절은 트루타입 폰트로부터 얻은 한자의 컨투어(contour) 정보를 이용하여 글자의 모든 획을 분리해 내는 방법을 소개한다. 이 방법은 모두 네 단계를 통해 수행된다. 먼저 컨투어의 그룹을 나눈다. 이것은 획을 나눔에 있어 서로 영향을 받지 않는 그룹을 나누어 획 비교 대상을 줄이는 효과가 있다. 다음으로 그룹별로 모든 컨투어에 대해 커브 성분을 제거 하고 직선

선분만을 추출한다. 그 후 각 그룹별로 획을 이룰 수 있는 벡터 집합을 찾아내고, 마지막으로 획 조각들이 합쳐져야 하는 경우를 찾아 그것을 하나의 획으로 묶어준다.

### 3.1 직선 성분 뽑아내기

획을 이루는 벡터들의 쌍들을 찾아내기 전에 우선 그러한 작업들을 용이하게 하는 사전 작업이 필요하다. 그래서 베지어 커브(vezier curve)로 이루어져 있는 컨투어들에서 곡선 성분은 제거 하고, 직선 성분만 남긴다. 그림 1과 같이 커브를 생성하는 컨트롤 포인트를 무시하고 끝점만을 연결하여 직선을 생성한다. 글자의 외곽 정보에서 커브 성분을 뽑아내어도 전체 글자의 모양은 유지되며, 직선만을 이용하면 기울기 계산이나 벡터들 간의 거리계산이 매우 간단해 지는 장점이 있다.

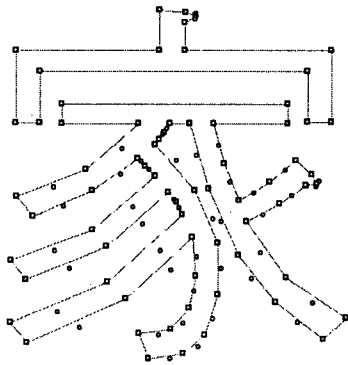


그림 1. 직선 성분만 추출한 결과

### 3.2 컨투어 그룹 나누기(contour grouping)

컨투어는 글자 외곽선을 이루는 하나의 폐곡선을 말한다. 글자에 따라서 하나 또는 2개 이상의 폐곡선으로 글자 모양이 표현된다. 그림 2는 '한(韓)'자에 대한 그룹 나누기 결과이다. 컨투어의 진행 방향이 반시계 방향이면 하나의 그룹이 된다. 그리고 (b)와 같이, 컨투어가 다른 컨투어에 포함 되는 경우는 내부의 컨투어는 시계방향으로 진행이 된다. 그러므로 두 개의 컨투어가 포함관계에 있고, 벡터 진행 방향이 반대이므로 같은 그룹에 속하게 된다. 따라서 결과는 (a)와 같이 3개의 그룹으로 나누어진다.

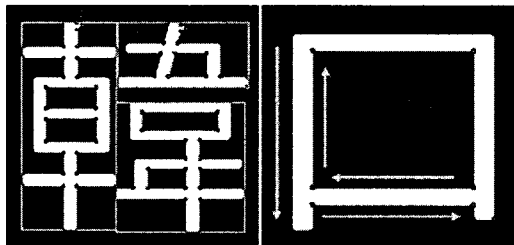


그림 2. 컨투어 그룹 나누기

### 3.3 라인 쌍 찾기

획을 구성하기 위해서는 평행한 두 축의 벡터 그룹이 필요하다. 하나의 벡터 그룹에 대해 상대되는 벡터 그룹을 찾아내는

것이 획을 구성하기 위한 기본적인 방법이다. 벡터 쌍을 찾아내는 기준은 거리와 벡터의 기울기, 그리고 벡터의 방향이다. 두 개의 벡터가 정해진 값 Td보다 작은 거리에 있고, 두 벡터 간의 각도가 Ta 보다 작은 경우 두 벡터는 하나의 획을 이루는 벡터 그룹에 포함된다. 두 벡터 간의 거리는 그림 3과 같이 직선과 점간의 거리를 이용하여 구할 수 있다.

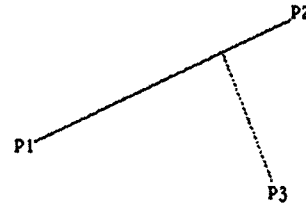


그림 3. 직선과 점 간의 거리

2개의 벡터는 그림 4와 같이 세 가지 경우로 배치 되는데, 1과 2의 경우만 벡터간의 거리 계산이 가능하며, 3의 경우는 고려 대상에서 제외되며 같은 벡터 그룹이 되지 않는다.

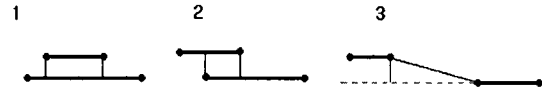


그림 4. 두 벡터의 배치

벡터의 기울기는 간단히 벡터 값에 의해 결정되고, 두 벡터 간의 각도는 식 (1)과 같이 두 벡터의 외적으로 구할 수 있다.

$$\theta = \sin^{-1} \frac{\|u \times v\|}{\|u\| \cdot \|v\|} \quad \text{식 (1)}$$

그림 5는 획을 구성하기 위한 벡터 그룹을 찾아내는 알고리즘을 코드로 표현 한 것이다.

```

MakeStroke(vector1, vector 2)
{
    If ( Distance(vector, vector 2) < Td
        && angle(vector1, vector2) < Ta
        && CrossProduct(line1, line2) > 0 )
    {
        If(SameDirection(vector1, vector2)) AddSameVectorSet()
        else AddOppositeVectorSet()
    }
}
    
```

그림 5. 벡터 그룹 찾기 알고리즘

### 3.4 획 합치기 (stroke merging)

앞서 완성된 획들은 아직 완전하지 않다. 여기서는 '十'자 형태의 획 모양인 경우, 왼쪽과 오른쪽 획 또는 위와 아래 획을

서로 연결시켜주어 완전한 획을 완성시켜야 한다. 그림 6은 '十'자 형태의 획의 끝부분에서 벡터의 진행 방향을 나타낸다. 왼쪽 획의 경우 오른쪽 끝부분의 벡터 진행방향이 모두 시계방향이다. 마찬가지로 오른쪽 획의 경우도 왼쪽 끝 부분의 벡터 진행방향이 시계방향이다. 이와 같이 '十'자 형태의 획이 나타나는 모든 경우, 획의 끝 부분에서는 벡터의 진행방향이 시계 방향임을 알 수 있다. 따라서 이러한 획을 찾아 수평한 위치에 있는 다른 획과 같은 벡터 그룹으로 포함시킨다.

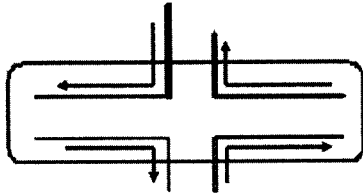


그림 6. '十'자 형태의 획

#### 4. 결과 및 분석

그림 7은 한(韓), 전(田), 중(中), 우(雨) 글자를 각각 논문에서 제안한 방법으로 획을 분할한 결과 그림이다. 폰트는 '굴림체'를 사용하였고, 글자의 크기는 가로, 세로 각각 500 픽셀, 그리고 획을 찾기 위한 벡터 그룹을 검색하는 데 필요한 임계 값 Td와 Ta는 각각 35와 30을 사용하였다. 굴림체는 획의 간격이 거의 일정하고, 획 굵기의 변화가 적어서 획을 분리해 내는 데 규칙이 매우 잘 적용된다.

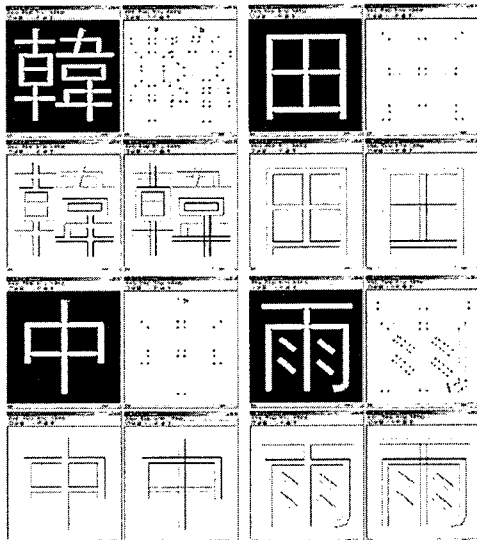


그림 7. 굴림체를 적용한 결과

그림 8은 "한양해서체"를 사용한 결과이다. 그림 6과 비교해서 성공적으로 획 분할이 되지 않는 경우가 발견된다. 이것은 획의 굵기의 변화가 상대적으로 크고 곡선의 변화가 더 심하기 때문이다.

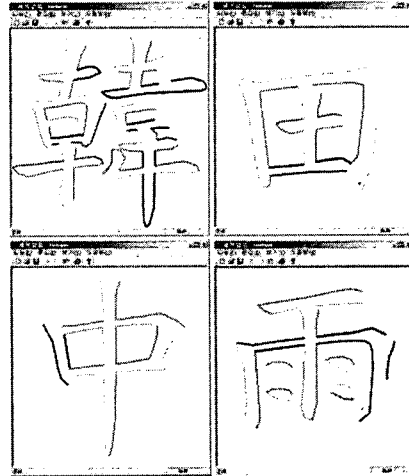


그림 8. 한양해서체를 적용시킨 결과

#### 5. 결론 및 향후 연구 계획

본 논문에서는 비교적 규칙이 잘 적용되는 특정 폰트에 대해서 성공적으로 획이 분리될 수 있다는 사실을 보였다. 제안한 방법을 이용하여 많은 양의 콘텐츠를 제작하여 웹을 통해 공유할 수 있을 것이다. 그러나 완전한 한자 학습 콘텐츠 제작을 위해서는 획 분리뿐만 아니라 획의 순서와 방향에 대한 정보도 함께 포함되어야 한다. 향후 연구로 자동으로 획의 순서를 결정할 수 있는 방법에 대해 연구할 예정이다.

#### 감사의 글

이 논문은 2005년도 BK21 과제에 의해 지원되었습니다.

#### 참고 문헌

- [1] 마법천자문 <http://www.magichanja.com/>
- [2] 아이한자 <http://www.ihanja.com/>
- [3] Sang Ok Koo, Hyun Gyu Jang and Soon Ki Jung, Efficient Stroke Order Animation of the Chinese Character, KCJC, 2005
- [4] USC chinese Department Homepage, <http://www.usc.edu/dept/ealc/chinese/newweb/home.htm>