폰트 밀도함수를 이용한 폰트 타입의 인식

o 진성아, 주문원
성결대학교 멀티미디어학부

Fontface Recognition Using the Font Density Function

Seong-Ah CHIN, Moon-Won CHOO
Division of Multimedia, Sungkyul University

요 약

폰트는 텍스트 정보를 기술하는 기본 요소로서 다양한 타입에 따른 독특한 감성정보를 내재하고 있다. 본 연구는 문서에 나타나 있는 영문폰트의 분포에 따른 감성정보 자동추출 시스템의 전처리 단계로서 문서상에서 특정의 폰트를 인식하는 모듈을 소개하고자 한다. 폰트 디자이너에 생성된 대부분의 폰트는 glyph data 라고 하는 2D boundary 좌표값에 의해 그 모양(Shape)이 결정된다. 이 데이터로부터 정의된 폰트밀도함수와 각 문자가 등장하는 보편적 확률 값의 linear combination으로부터 각 폰트를 식별 할 수 있다.

1. 서론

폰트는 텍스트 정보를 표현하는 기본 매체로서 자동적으로 타입과 스타일, 그리고 크기가 생성할 수 있도록 설계되어 있다. 설계자들은 각 폰트타입이 갖고 있는 독특한 감성정보를 염두에 두고 설계하므로 특정 폰트 타입으로 구성된 문장은 폰트가 갖고 있는 내재적인 감성정보를 암묵적으로 전달하게 된다. 역으로, 특정 감성적 요소를 강조하고자 하는 문장을 생성할 때에 적절한 폰트타입을 선택하면 정보의 전달과 표현에 시너지 효과를 발생시킬 수도 있을 것이다. 이러한 배경으로 특정 문장을 구성하는 폰트타입의 분포를 이용하여 문장이 내포하고 있는 감성정보를 측정하고자 하였다. 이를 위하여 특정 감성타입을 각 폰트별로 조사하고, 감성 퍼지값으로 처리한 후, 문장 내의 폰트 타입의 분포와 이 감성 퍼지값을 이용하여 문장의 감성정보를 추출하고자 하였다. 여기서는 이 처리과정을 위한 전단계로서 트루타입 폰트의 생성과정[1],[2],[3]을 살펴보고, 각 폰트를 식별할 수 있는 방법을 제시하는데 있다.

현재로 문서인식 연구에 비하여 폰트인식에 관한 연구는 대단히 미비한 상태에 있다. 폰트인식에 관한 관련 대표적 연구로는 Zramdini가 시도한 타이포그래픽을 사용하는 통계적 방법[4]으로, 텍스트의 weight, slope, size를 특징으로 하여 multivariate Bayesian classifier를 이용하여 특정 폰트를 식별한다.

본 연구는 폰트 인식모듈을 개발하는 것으로, 먼저 각 폰트가 갖는 segment width density(WDH) function을 정의한다. WDH 함수는 각 영문 폰트가 가지는 고유한 segment width과 그 높이를 이용하여 정의된다. 이 함수와 각 폰트의 출현하는 빈도수를 이용하여 각 폰트가 갖는 고유한 특징을 추출할 수 있다. 그림.1은 인식모듈의 전처리 흐름도를 보여주고 있다. 본 글에서는 이 전처리 단계에서 처리된 문서의 텍스트 블록에서 추출된 폰트의 식별 모듈에 초점을 맞추고 있다.

2. 폰트식별모듈

2.1 WDH 함수

segment width은 각 폰트에 해당하는 한 문자에서 높이가 고정되어 있을 때의 내부의 width이다. 그림 1에서 보듯이 특정 폰트의 각 문자마다 고유의 segment width이 존재한다. 따라서 폰트 디자

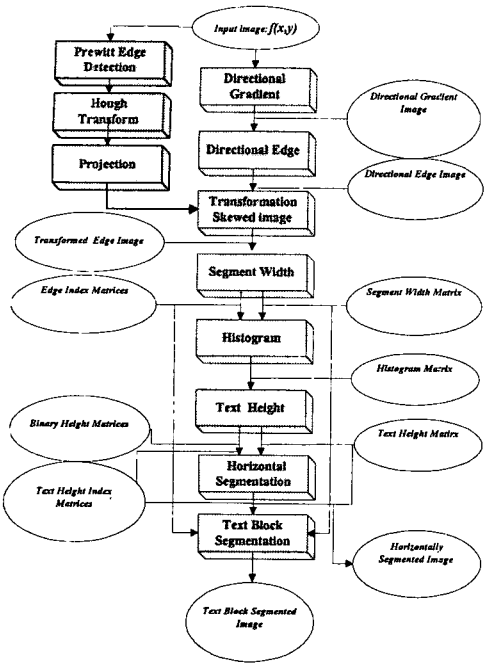

그림.1 전처리 흐름도

인 시 고려되는 glyph table정보를 이용하여 정확한 segment width 밀도함수식을 세우는 것이 가능하다. 트루타입 폰트의 각 영문자는 glyph table에 있는 boundary glyph data로 구성되어진다. 각 glyph data는 각 문자의 가장자리의 x y좌표값을 나타낸다.

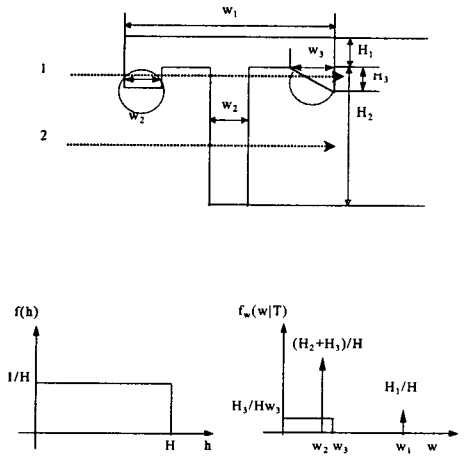

그림.2 폰트 T의 glyph data

각 glyph data는 Bezier Curve(그림.2의 w_3인 경우)나 straight line(w_2인 경우)을 그리고 각각이 on curve인지 off curve인지에 대한 정보를 갖는다. Bezier Curve와 straight line은 parametric 형태의 수식으로 표현가능하며, 이를 이용하여 segment width 밀도함수를 정의할 수 있다. 그림. 2. 에서 보는 바와 같이, 각 문자마다 고정된 segment width(Fixed Segment Width)이 존재하는 경우도 있고, 연속 segment width(Continuous Segment Width)을 갖는 경우가 존재한다. 그림.2의 w_3와 같은 부분은 연속 segment width에 해당된다. 각 문자는 straight line 또는 Bezier Curve로 구성되어짐을 알 수 있고, 각 glyph data로 부터 고정된 높이에서 segment width을 산출하는 식을 새울 수 있으므로, 이를 이용하여 특정 폰트에서 각 문자가 갖는 segment width 밀도함수를 구할 수 있다. 단 고정된 높이는 일양분포(uniformly distributed) density function을 갖게된다.

결론적으로 parametric 방정식을 사용하여 특정 폰트에서 각 문자에 해당하는 segment 밀도함수를 식 2.1a,b와 같이 정의할 수 있다. 특정 폰트의 모든 문자에 대해 각 문자가 출현하는 확률을 이용하여 특정 폰트에 대한 함수를 구할 수 있다.

$$Class1 : f_i(w_i|L) = \begin{cases} \dfrac{f_i(h)}{\left|\dfrac{d}{dh}g(h)\right|} & ; w_i \in \text{Continuous Segment Width} \\ 0 & ; \text{otherwise} \end{cases}$$

(식 2.1a)

$$Class2 : f_i(w_i|L) = \begin{cases} \dfrac{h_i}{H}\delta(w_i - c) & ; w_i \in \textit{Fixed Segment Width} \\ 0 & ; \text{otherwise} \end{cases}$$

(식 2.1b)

식2.1a, 식2.1b는 segment width 밀도함수를 정의한다. 즉, 입력으로 일양분포를 갖는 고정된 높이가 주어지면, segment width이 연속인 경우는 식 2.1a에 해당되고 고정된 경우는 식 2.1b에 해당된다. 단 H는 특정 문자에 해당되는 전체 높이이고 $f_w(w_i|L)$는 특정 폰트에서 문자 L이 주어진 경우의 segment w_i의 밀도함수이다. 각 문자에 대한 segment width 밀도함수를 이용하여 특정 폰트에 대한 밀도함수를 구할 수 있다. 그림 3은 각 문자에 대한 segment width 밀도함수를 구하는 절차

를 보여준다. glyph data를 입력 값으로 받은 뒤 각 segment가 straight line혹은 Bezier Curve인지를 조사한다. 고정된 높이를 알고 있을 때 segment width 길이를 알 수 있다.

2.2 폰트 식별 모듈

조사된 segment를 histogram을 이용하여 각 segment의 빈도수를 계산한다(그림.3 참조). 일반적으로 영어문장에서 등장하는 알파벳의 빈도수를 조사하여 이를 확률값 $P(L=j)$로 하기로 한다(각 빈도수를 알아내기 위하여 sci.crypt뉴스 그룹에 의뢰하여 도움을 얻었다). 각각의 빈도수는 Table 2.1과 같다.

L	P(L)	L	P(L)	L	P(L)	L	(P)
E	12.4%	H	6.7%	M	2.5%	P	1.6%
T	8.9%	S	6.2%	W	2.3%	B	1.3%
A	8.0%	R	6.1%	C	2.2%	V	0.8%
O	7.6%	D	4.6%	F	2.2%	K	0.7%
N	7.0%	L	3.6%	G	2.0%	Q	0.1%
I	6.7%	U	2.7%	Y	2.0%	X	0.1%

Table 2.1 알파벳 확률값

$f(w)$를 폰트 밀도함수라고 한다면, 식 2.2에서 $f_i(w|L)$는 특정 문자에 대한 segment 밀도함수가 된다.

$$f(w) = \sum_{L=A,B,\ldots Z} P(L=j)\, f_j(w|L=j) \qquad \text{(식 2.2)}$$

따라서 식 2.2로부터 각 폰트에 해당하는 고유한 영역을 추출할 수 있다.

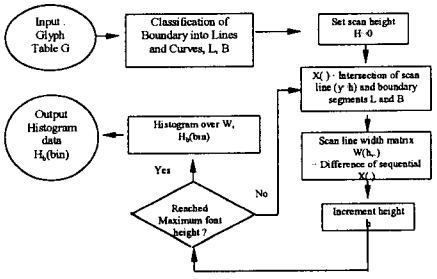

그림 3
Histogram Over Segment With

그림4는 Times Roman에 대한 폰트 식별함수를 이용하여 생성된 clustering 영역이다. Segment width 193에서 가장 높은 빈도수 254.76를 보이고, 다음으로 89에서 69.65를 기록하고 있다(segment width은 마이크로소프트사에서 제공하는 glyph table의 데이터를 그대로 사용하였다). 점선으로 표시된 부분이 어떤 segment에 대해서도 식별 가능한 영역이다. 즉, 폰트의 사이즈가 작아질지라도 segment width에 대한 높이는 스케일링된 값으로 얻을 수 있다.

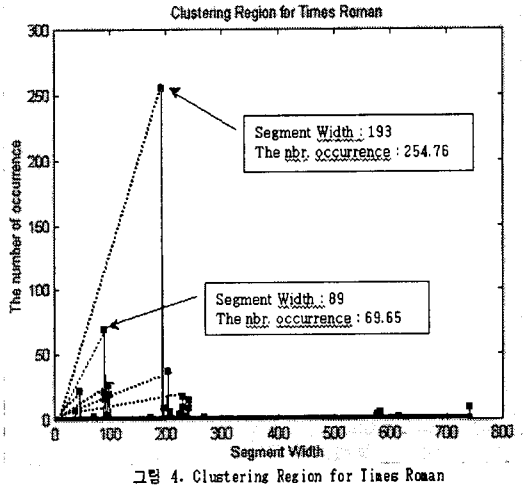

그림 4. Clustering Region for Times Roman

3. 결론

본 논문에서는 폰트를 식별할 수 있는 모듈을 제시했다. 각 폰트만이 갖는 고유한 segment width 밀도함수를 이용하여 정의하고 실험하였다. 이렇게 하여 추출된 폰트 타입은 독특하지만 보편적인 감성정보를 수반한다고 할 수 있으므로, 특정 문장에서 자동추출된 폰트타입의 분포와 각 폰트가 갖고 있는 감성정보의 퍼지모델을 사용하여, 그 문장의 감성정보를 가늠해 볼 수 있을 것이다. 또한 보편화된 감성정보의 특성을 활용하여, 폰트와 같은 정적인 환경과 더불어 문장이 갖고 있는 동적인 의미론적 감성을 추출할 수 있다면, 웹 환경에서 다양한 용도로 활용할 수 있을 것이다.

[참고문헌]

[1] Donald E. Knuth, "Digital Typography, Center for the Study of language and information Staford Junior University, 1999

[2] Laurence Penny, "A history of TrueType", TrueType Typography Technical Report, TrueType Development Team at Apple, 1999

[3] D. Herman and the Apple TrueType team, "The TrueType Reference Manual", Apple Computer Inc. Feb. 1998

[4] A. Zramdini, R. Ingold, "Optical Font Recognition Using Typographical Features", IEEE Transactions on Pattern Analysis And Machine Intelligence. Vol. 20. No. 8, pp 877-882, August 1999