

## 트리 구조를 이용한 수식 인식 연구

\*, \*\*, \*\*\*

## A Study on Equation Recognition Using Tree Structure

Byung-Joon Park\*, Hyun-Sik Kim\*\*, Wan-Tae Kim\*\*\*

**요약** 수식은 일반 문장에 비해 복잡한 구조와 다양한 문자와 기호가 사용되어 단순한 키보드 입력만으로는 모든 문자 집합을 입력할 수 없어 한글이나 워드 같은 문서편집기 내에서도 자체적으로 구현된 수식 편집기를 사용하고 있다. 수식을 올바르게 표현하기 위해 구문을 해석할 수 있는 의미가 될 수 있는 사전 학습 정보가 필요하다. 문자가 입력되더라도 크기와 위치 서로간의 관계에 따라 다른 수식으로 표현될 수 있기 때문이다. 즉 표현될 위치와 크기 등 문자와 기호들 간의 상호관계를 고려하여 수식의 형태를 트리 모델로 표현한다. 문자인식 응용의 한 분야로 문자나 기호(부호)를 인식하는 기술을 이미 널리 알려졌지만, 수식을 입력과 해석하는 방법은 일반적인 텍스트에 비해 복잡한 분석 과정이 필요하다. 본 논문에서는 수식의 문자를 인식하고 표현되는 위치와 크기의 결정을 빠르게 해석하는 수식 인식을 구현하였다.

**Abstract** The Compared to general sentences, the Equation uses a complex structure and various characters and symbols, so that it is not possible to input all the character sets by simply inputting a keyboard. Therefore, the editor is implemented in a text editor such as Hangul or Word. In order to express the Equation properly, it is necessary to have the learner information which can be meaningful to interpret the syntax. Even if a character is input, it can be represented by another expression depending on the relationship between the size and the position. In other words, the form of the expression is expressed as a tree model considering the relationship between characters and symbols such as the position and size to be expressed. As a field of character recognition application, a technique of recognizing characters or symbols(code) has been widely known, but a method of inputting and interpreting a Equation requires a more complicated analysis process than a general text. In this paper, we have implemented a Equation recognizer that recognizes characters in expressions and quickly analyzes the position and size of expressions.

**Key Words** : Equation recognition, Pattern matching, Thinning, OCR, Image Scaling

## 1.

컴퓨터 비전은 관찰 대상의 모양, 크기, 조명, 회전과 같은 여러 가지 변화에 따라 똑같은 영상이라도 다르게 보이는 상황이 될 수 있다. 특히 모양은 똑같지만, 각각 다른 크기를 가지고 있는 두 개의 객체를 검사할 경우 다른 객체로 인식할 문제를 가지고 있다[1, 2]. 영상은 수식을 포함하고 있는 수화책과 한글을 이용하였다. 문서에 있는 수식을 인식하기 위해 기존에는 문자를 학습하여 문

자와 자연 언어 연구의 이론적 배경이 되는 언어학이나 논리학에서는 언어 자체를 독립된 현상으로 생각하여 언어에 의해 표현된 문장의 문법적 구조를 중시하였고, 이에 영향을 받은 구문 중심적 문장 분석 방법이 주로 이용되었다[3].

구문 중심적 문장 분석 방법은 음운 분석, 구문 분석, 의미 분석의 여러 단계로 분리되어 있다 각각의 방법들은 각 단계에서 분석에 필요한 지식 정보들이 별개의 지식 베이스에 분산되어 저장되기 때문에 지식의 상호 참조가

\* Department of Software Engineering, Seoil University

\*\* Corresponding Author : Department of Software Engineering, Seoil University (hskim@seoil.ac.kr)

Received June 14, 2018

Revised June 26, 2018

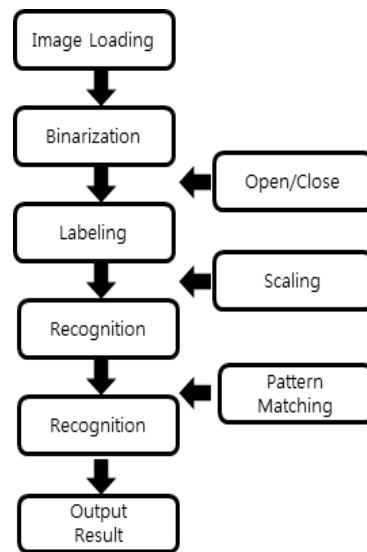
Accepted July 26, 2018

불가능하며 구조와 의미의 모호함을 해결하기 어렵다. 특히 수식의 범위를 결정하기가 모호한 구문을 해석해야 할 경우 일반적인 정보만을 이용한다면 단순 나열의 트리가 생성될 수 있기에 문자열 마다 의미를 주어 다른 지식 베이스와 연동을 할 수 있는 문법적 규칙이 있는 방법이 요구된다.

따라서 본 논문에서는 수식의 구조 분석과 의미 분석들을 하나의 제어 구조 아래 통합시키고 지식 정보의 연동을 실현할 수 있는 모델의 구조를 만들고자 한다. 수식의 각 단어나 기호마다 문법적인 의미를 주어 모든 정보가 언제든지 필요에 따라 적용 할 수 있도록 한다[4].

수식 인식은 크게 몇 가지 과정을 거친다. 첫 번째로 수학에서 표현하는 수식의 인식은 여러 다양한 유형을 가지고 있는 연산자 및 기호를 기반으로 각각의 수식 요소들에 어떤 의미를 부여할지를 결정하여 문자와 문장의 형태로 구성한다. 모든 노드는 각각의 문자열 방식으로 변환된다. 두 번째로 수식의 처리는 입력된 문자에 대한 해당 수식의 유형을 분석한다. 각각의 유형을 분석한 후 유형에 맞는 해당하는 절차의 함수를 호출한다. 세 번째 계산의 내부형식으로 표현된 수식을 알아보기 쉬운 형태의 벡터 이미지로 재구성한다. 벡터 이미지는 크기에 상관없이 이미지화할 때 다시 계산되므로 계단 현상 없는 깨끗한 이미지를 볼 수 있게 한다.

수식 인식 모듈은 입력 이미지를 가공하여 문자정보를 추출한다. 그 정보를 적절히 분석하여 문자형으로 생성하고, 추출된 문자 및 계산이 가능한 문자를 트리 형태의 수식 모듈로 재구성하였다. 1장에서는 공간상에 쓰인 수식으로부터 기호 단위 인식이 완벽하게 되었다는 가정 하에 수식의 구조를 해석하는 방법을 제안한다. 2장에서는 시스템을 구현하는 방법을 소개하고 3장에서는 실험결과에 관해 서술하였다. 그리고 마지막으로 4장에서는 결론 및 향후 과제를 기술한다. [그림 1]은 수식인식 과정의 순서도이다.



1.

Fig. 1 Equation Recognition Module

## 2.

본 논문에서는 수식을 기호를 중심으로 순차적으로 읽어 들어 구문론적으로 분석 후 기호(심벌)를 중심으로 위치 관계를 문자형 수식으로 표현한다. 입력된 수식은 심벌 간의 위치 관계의 조합으로 인식되며, 기호는 순차적으로 추가할 때마다 기호와 이미 추가된 기호 간의 연관 관계를 분석하여 전체 수식의 구조를 쉽게 문자형으로 만들 수 있다. 제안 모델을 이용하여 기호별로 분할 인식된 수식으로부터 기호를 하나씩 트리로 생성하면서 수식 인식이 이루어진다[5].

본 논문의 모델은 다음과 같이 세 가지 방식으로 요약할 수 있다. 첫째, 입력된 수식의 기호를 하나씩 순차적으로 추가하며 수식을 완성하는 방법을 사용한다. 두 개 이상의 단위로 병합하는 분석 기법에 비해 더 빠른 트리 탐색을 함으로써 효율적으로 인식이 가능하다. 둘째, 제안 모델을 통해 미완성된 수식에 기호를 새로 추가하여 기호의 위치를 지정한다. 셋째, 입력된 모든 수식을 다시 탐색하여 지수를 안정적으로 인식을 할 수 있다. [표 1]은 수식 데이터의 타입이다.

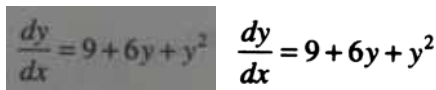
1.  
Table 1. Equation Data Type

Attribute	Input Data
Arithmetic	$3 \times 5 + 2$
Equation	$x^2 - 5x + 6 = 0$
Inequality	$6x - 9 \geq 3(x + 2)$
Summation	$\sum_{k=1}^{10} (2k + 1)$
Integral	$\int_1^3 9x^2 + 2x + 6 dx$
Differential	$\frac{d}{dx} = 9 + 6y + y^2$
Lim	$\lim_{x \rightarrow 0} \frac{A}{3}$
Matrix	$A = \begin{bmatrix} 5 & 7 \\ 3 & -4 \end{bmatrix}$

2.1

문자인식에 필요한 이미지 정보를 메모리에 저장 후 수식 영역만 따로 학습한다. 특히 템플릿 영상은 정적 템 버로 선언되어 있어, 처음 1회만 읽고 이후 과정은 생략 된다. 학습 수식 데이터는 30개의 수식 종류 10개 세트로 약 300개의 학습 데이터를 이용하였다.

영상을 이진화한 후 라벨 링을 하여 텍스트를 뽑아낸다. 그 후 문자를 인식하여 텍스트를 분리한 후 시그마나 분수를 기준으로 인접한 픽셀을 동일한 영역에 포함한다. 인접한 픽셀은 상하좌우 영역으로 재검색한다[6]. [그림 2]는 입력된 수식의 전 처리 전과 후의 이미지이다.



2. (a) , (b)  
Fig. 2 (a) Before Preprocessing  
(b) After Preprocessing

문자인식에 필요한 수식을 메모리에 로드한 후 필요한 영역만을 잘라내 학습된 이미지와 비교한다. [그림 3]은 수식의 영역을 지정한 것이다.

3.  
Fig. 3 Equation Region

프로그램의 실행 시 최초 1회만 학습 데이터를 로딩하고, 이후부터는 학습 데이터 로딩 과정을 생략한다. 학습 데이터는 날짜를 기준으로 만들어져서 최신 데이터를 기준으로 등록된다. [그림 4]는 수식인식과정에서 전 처리 과정인 수식 영역 지정 과정과 문자인식 과정이다.

4.  
Fig. 4 Character Recognition

2.2

수식의 이진 영상에서 문자 영역을 추려내기 위해 라벨링 과정을 수행한다. 영상을 탐색하여 인접한 255레벨의 픽셀을 같은 영역에 포함한다. 수식의 입력 데이터는 분석하기 전 적당한 기호 간의 그룹화가 필요하다. 즉 sin, cos, tan, 같은 함수는 하나의 단위로 취급을 해야 한다. 본 논문에서는 미리 학습된 함수 명을 기준으로 각각의 단위로 인식된 영문을 하나의 함수의 기준에 맞춰 그룹화 한다.

각 영역의 위치 및 크기 정보를 클래스 화하여 트리에 저장한 후 분석을 수행한다. 여러 개의 문자가 하나의 블록 영역을 형성한다. 즉 sin, cos, log 등에 대한 인식이 가능하고, s, l, n, x 같은 경우는 sin, x 로 각 영역별 하나의 블록으로 인식한다. 즉 여러 개의 문자형 각각이 하나의 기호를 형성하는 경우를 탐색한다. 현재 lim, log, sin, cos, tan 삼각함수 등의 인식이 가능하다[7]. 다음은 인수의 인식이다. 수식에서 인수는 문장의 오른쪽에 오게 된다. 인수가 되는 조건은 우측에 근접해 있는 문자가 왼쪽 문자의 60 ~ 70% 정도의 높이가 약 50% ~ 75% 정

도의 경우이다. 문자의 크기나 높이는 문서마다 다를 수 있으므로 조절할 수 있게 하여야 할 것이다. 수식은 클래스로 저장한 후 이를 수식유형에 대해 분석을 한다. 이때 연산자를 나타내는 영역으로 되어있는 문자열을 분석한 후 연산자에 맞는 계산 과정을 호출한다. [그림 5]는 수식의 문자열 그룹화를 수행한 이미지이다.

$$\tan x = \lim_{m \rightarrow a} \sum_{x=n}^{x=m} \frac{A^c}{Q}$$

5. Fig. 5 Equation Grouping

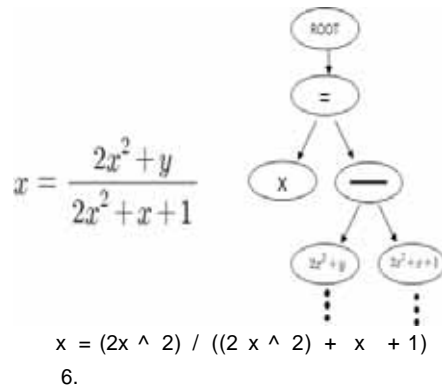
2.3

수학 함수는 개개의 기호를 하나의 기호로 그룹화한 후 그래프를 생성한다. 각 노드의 속성은 인식된 기호의 결과와 좌표를 가지고 있다. 수식을 인식한 후 문자열로 받아와 구문 분석을 한 후 수식의 유형과 연산자에 맞는 함수를 호출하여 수식의 계산된 결과를 생성한다[8].

이 과정은 단순 문자가 아닌 수식이므로 문자처럼 읽을 수 있는 방식을 위한 것이다. 수식 계산은 그림 3과 같이 처리된다. 트리 구조로 저장된 수식은 터미널 노드를 리스트에 저장한 후 수식 유형의 분석에 활용한다[8]. 각 연산자를 나타내는 영역별로 그룹화 된 기호와 스트링을 분석하여 그래프를 생성한다. 그래프의 각 노드는 하나의 기호(예 : 시그마, 분수)를 중심으로 노드의 속성을 인식된 결과에 따라 좌표를 설정한다. 그 후 각 노드 사이의 관계를 그래프로 표현한다. [그림 6]은 [그림 5]에서 그룹화 된 기호를 수식계산 트리를 그래프로 그린 것이다.

3.

개발 프로그램은 Visual Studio 2015의 C#을 이용하였다.



6. Fig. 6 Equation Calculation Tree

데이터는 수학책을 이용하여 300개의 수학식을 구성하였다. 하나의 수식 당 30개의 서로 다른 학습 데이터를 구성하여 실험을 하였다[9].

트리 구조 방식으로 수식을 변화하여 계산한 결과 전체적으로 약 96% 이상의 인식률을 보였다. 식이 잘 안 되는 부분은 지수 부분으로 일반 텍스트와 지수의 인식이 제대로 안 되는 부분이 있었다. 이 부분은 서로 간의 간격과 높이 문제를 조절하여 해결해야 할 것이다. 또한 수식의 기호를 사용하고 있는 임계치의 값이 적당치 않을 경우와 분모 분자의 기호들의 분수나 시그마 등의 기호를 벗어나는 경우에 해당한다. 이는 학습 데이터의 조정이나 학습 데이터의 수정으로 해결을 할 수 있을 것이다[10, 11].

또한 복잡한 변수를 갖는 수식은 같은 책에서 추출한 수식 영상이라도 크기가 다르게 되는 경우도 있어 스케일 조절을 하여 인식률을 높일 필요가 있다. [표 2]는 XML을 응용하여 수학 수식을 표현한 마크업 언어인 MathML과 본 논문의 수식인식 결과를 비교한 것이다.

2.

Table 2. Equation Data Type

input	MathML	Recognition
3×5+2	$\langle mrow \rangle$ $\langle mn \rangle 3 \langle /mn \rangle$ $\langle mo \rangle * \langle /mo \rangle$ $\langle mn \rangle 5 \langle /mn \rangle$ $\langle mo \rangle + \langle /mo \rangle$ $\langle mn \rangle 2 \langle /mn \rangle$ $\langle /mrow \rangle$	3 + 5 + 2
x <sup>2</sup> -5x+6=0	$\langle mrow \rangle$ $\langle msup \rangle$	(x <sup>2</sup> - 5x + 6) = 0

	$x^2$ $2 < \frac{2}{mn}$ $5 < \frac{5}{mn}$ $x < \frac{x}{mi}$ $6 < \frac{6}{mn}$ $= < \frac{=}{mo}$ $0 < \frac{0}{mn}$	
$6x - 9 \geq 3(x + 2)$	recognition not	$(6x - 9) \geq (3 * (x + 2))$
$\sum_{k=1}^{10} (2k + 1)$	$\sum_{k=1}^{10} (2k + 1)$	$\sum_{k=1}^{10} (2k + 1)$
$\int_1^3 (9x^2 + 2x + 6) dx$	$\int_1^3 (9x^2 + 2x + 6) dx$	integral (3, 1) $(9x^2 + 2x + 6) dx$
$\frac{d}{dx} = 9 + 6y + y^2$	$\frac{d}{dx} = 9 + 6y + y^2$	$(d/dx) = (9 + 6y + y^2)$

$\lim_{x \rightarrow 0} \frac{A}{3}$	recognition not	$\lim_{x \rightarrow 0} (x^0)$ (A / 3)
$A = \begin{bmatrix} 5 & 7 \\ 3 & -4 \end{bmatrix}$	recognition not	(A) = ({5, 7}, {3, -45})

4.

본 연구에서는 수식을 트리 블록모델로 분석하여 인식하는 방법을 제안하였다. 제안 방법은 실용적으로 사용이 가능한 인식 속도를 내어주고 있다. 하지만, 탈 초점 상태로 되어 시야 심도가 낮거나 변이가 심한 블러(Blur)된 이미지와 수식의 기호 종류 및 입력 범위가 넓은 경우와 하나의 연산 안에 또 다른 연산이 행하여지는 연산의 경우 기호를 먼저 찾는 학습의 문제로 인식이 제대로 되지 않는 것을 볼 수 있다. 이를 해결하기 위해 기호 간의 특징이나 수식마다 다를 수 있는 기호들 간의 규칙을 개선하고 개인이 프로그램 상에서 특징을 조정할 수 있도록 해야 할 것이다.

연구에서 제안된 모델의 실험을 효율적으로 진행하기 위해 수식 입력에 대해 기호 분할 및 인식 과정이 수행된다고 가정을 하였으나, 다량의 기호를 인식하거나 인식할 기호의 후보를 생성하는 경우에도, 작은 변형만으로 충분히 인식할 수 있으므로, 실용적인 수식 입력 시스템의 개발이 용이할 것이라고 본다.

향후 연구로서, 많은 자료를 수집하여, 정량적인 성능 평가를 수행하고, 더 복잡한 구조의 수식에 대해서도 빠른 탐색이 가능하도록 하여야 할 것이다. 또한 수식 내의 알파벳, 숫자, 괄호의 순서 등의 문법적인 요소를 도입하여 위치 관계만으로는 비교하기 힘든 지수에 대한 좀 더 효율적인 알고리즘이 필요할 것이다. 또한, 코드가 많은 수식의 분류는 과도한 재귀연산이 초래되어 컴퓨터 자원이 낭비되어 알고리즘의 최적화와 인식 속도의 개선이 필요하다. 또한 필기체 인식을 추가하여 직접 사람이 입력한 필기체를 인식할 수 있게 하여야 할 것이다.

REFERENCES

[1] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing Analysis and Machine Vision," Cengage Learning, 2007.

[2] R. T. Chin, and C. A. Harlow, "Automated Visual Inspection: A Survey," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.PAMI-4, No.6, pp.557-573, 1982.

[3] R .C. Gonzalez, and R.E. Woods, Digital Image Processing, 2nd Ed, Prentice-Hall, 2002.

[4] A. Grvavec and D. Blosten, "Mathematics Reconition Using Graph Rewriting," Proc. Third Int. Conf. on Document Analysis and Recognition, pp. 417-421, Montreal, Canada, Au g. 1995.

[5] M Okamoto and B Miao , "Recognition of mathematical expressions by using the layout structures of symbols," In Proc. Int. Conference on Document Analysis and Recognition, Saint M alo, France, pp. 242-250, 1991.

[6] Y. Weixin, et al. "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.

[7] H. Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] S. Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[9] C. Razafimandimby, V. Loscri, A. M. Vegni, "A neural network and IoT based scheme for performance assessment in internet of robotic things," IEEE Intern. Conf. on IoT Design and Implement. pp. 241 - 246, April, 2016.

[10] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886-893, 2005.

[11] S. Dhalwar, Y. Chinta. and B. Karthikeyan. "Image Processing Based Vehicle Parking Management Using Mobile Application", Indian Journal of Science and Technology 9, pp. 44-50, 2016.

**(Byung-Joon Park)** [ ]



- 2000 2 : ( )
- 2002 8 : ( )
- 2010 2 : ( )
- 2018 7 ~ :

< > , , ,

**(Hyun-Sik Kim)** [ ]



- 2004 2 : ( )
- 2010 8 : ( )
- 2018 7 ~ :

< > , , ,

**(Wan-Tae Kim)** [ ]



- 2005 2 : ( )
- 2011 2 : ( )
- 2018 7 ~ :

< > IoT, Device, AI System , Smart