2009 10th International Conference on Document Analysis and Recognition

Identification of Mathematical Expressions in Document Images

Utpal Garain
Indian Statistical Institute,
203, B.T. Road, Kolkata 700108, India.
utpal@isical.ac.in

Abstract

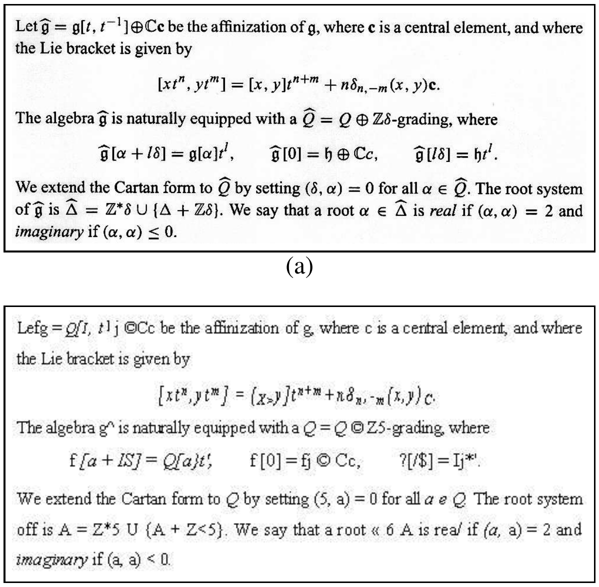
Identification of mathematical expressions in document images is attempted. Several features starting from low level image features, shape features, linguistics information, etc. are extracted and combined to pinpoint the expressions. A performance index has been formulated to evaluate the identification results. Experiment uses a dataset of 200 publicly available images containing 1163 embedded and 1039 displayed expressions. Test results show accuracies of 88.3% *and* 97.2%*, respectively for extracting embedded and displayed expressions perfectly.*

1 Introduction

The existing optical character recognition (OCR) systems show severe limitations for converting scientific papers into corresponding electronic form. Figure 1 demonstrates one such example obtained from one the popular OCR systems [1] . This is because current systems fail to properly recognize mathematical expressions that often appear in scientific documents. This makes the conversion of scientific papers from printed to electronic form a difficult task.

Though several studies have been reported on recognition of mathematical expressions [1, 2, 3] but most of them assume that expressions are available in isolated form. Therefore, in order to use the existing expression recognition methods, sufficient effort must be given to isolate the expressions first from the rest of a document. This could be a major reason behind the poor success rate of recognizing scientific documents containing expressions.

Among the existing techniques dealing with identification of expression zones, methods proposed by Lee and Wang [4], Fateman [5], Inoue *et. al.* [6], Toumit *et. al.* [7], etc. significant. These approaches are well explained with examples but experimental details like dataset, results, etc. are not reported. The method by Kacem *et. al.* [8] is tested on a private dataset consisting of 300 expressions and a success rate of about 93% is acheved. A similar technique is used in [9] to locate mathematical expressions in printed documents but its performance are not reported. More recently, Chowdhury *et. al.* [10] showed that their method works well for segmenting displayed expression (with a success rate of 97.69%) but gives only 68.08% accuracy for extraction of embedded expression. The authors of [10, 11] concluded that the extraction of embedded expressions is quite difficult without doing character recognition.



Let $\widehat{\mathfrak{g}} = \mathfrak{g}[t, t^{-1}] \oplus \mathbb{C}\mathbf{c}$ be the affinization of $\mathfrak{g}$, where $\mathbf{c}$ is a central element, and where the Lie bracket is given by

$$[xt^n, yt^m] = [x, y]t^{n+m} + n\delta_{n,-m}(x, y)\mathbf{c}.$$

The algebra $\widehat{\mathfrak{g}}$ is naturally equipped with a $\widehat{Q} = Q \oplus \mathbb{Z}\delta$-grading, where

$$\widehat{\mathfrak{g}}[\alpha + l\delta] = \mathfrak{g}[\alpha]t^l, \qquad \widehat{\mathfrak{g}}[0] = \mathfrak{h} \oplus \mathbb{C}c, \qquad \widehat{\mathfrak{g}}[l\delta] = \mathfrak{h}t^l.$$

We extend the Cartan form to $\widehat{Q}$ by setting $(\delta, \alpha) = 0$ for all $\alpha \in \widehat{Q}$. The root system of $\widehat{\mathfrak{g}}$ is $\widehat{\Delta} = \mathbb{Z}^*\delta \cup \{\Delta + \mathbb{Z}\delta\}$. We say that a root $\alpha \in \widehat{\Delta}$ is *real* if $(\alpha, \alpha) = 2$ and *imaginary* if $(\alpha, \alpha) \leq 0$.

(a)

Lefg = *Q[I, t*¹ j ©Cc be the affinization of g, where c is a central element, and where the Lie bracket is given by

*[xt*ⁿ*,yt*ᵐ*] = {x>y]t*ⁿ⁺ᵐ*+n8*ₙ*,-*ₘ*(x,y)c·*

The algebra g^ is naturally equipped with a *Q* = *Q* © Z5-grading, where

f *[a + IS]* = *Q[a}t',*   f [0] = fj © Cc,   ?[/$] = Ij*'.

We extend the Cartan form to *Q* by setting (5, a) = 0 for all *a e Q* The root system off is A = Z*5 U {A + Z<5}. We say that a root « 6 A is rea/ if *(a,* a) = 2 and *imaginary* if (a, a) < 0.

(b)

Figure 1. OCR of scientific documents: an example (a) image (b) recognition results.

Review of the above approaches reveals their strengths and weaknesses. Weaknesses are mostly observed in the following aspects that need further research. (i) *Sufficient experiment*: Most the previous studies lack in enough experiment (or did not report test results) to check the efficiency of the proposed methods. Therefore, potential of such methods are not readily evident. (ii) *Scope of the problem*: Some techniques failed to address the entire scope and difficulty of the extraction problem. In many of these studies, only displayed expressions are considered for extraction. (iii) *Nature of test dataset*: Unavailability of any standard dataset prompted previous researchers to define their own dataset and as a result, replication of experiments and comparison of performance among different methods has become a difficult task. (iv) *Evaluation strategy*: Unavail-

[1] ABBYY FineReader OCR Professional 7.0

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.203

ability of any evaluation technique makes performance evaluation difficult for many of the previous methods.

The study embodied in this paper is motivated by these research needs. The proposed approach considers extraction of both embedded and displayed expressions. Finding out appropriate set of features for the present problem is a significant contribution of this study. Different feature averaging schemes have been explored in this paper and their efficiencies in extracting expressions have been verified experimentally. Use of datasets available in public domain would make the present study comparable with other techniques. Formulation of a performance evaluation strategy facilitates an easy way for judging the performance of the proposed approach.

2 Features used for extraction of expressions

Next two sub-sections explain the features used for identification of displayed and embedded expressions.

2.1 Displayed expressions

Displayed expressions appear as isolated text lines and they exhibit some image level features that distinguish them from a normal text line. For instance, the statistical investigation on the corpus of 400 pages, as described in [12] reveals that the mean value of the white spacing between two normal text lines is about 0.4 times the text height whereas the mean value of the white spacing above and below the displayed expressions is nearly 1.8 times the text height. By text height we mean the average height of the text lines of a document. In general, normal text of point-size 10 and 12 are found in technical documents. Therefore, as far as text height is concerned displayed expressions are often taller (along vertical direction) than any normal text line.

Another important characteristic of displayed expressions is attributed to the spatial arrangement of the constituent symbols. The y-coordinates of the lower-most black pixels (or the center pixels of the symbols' bounding boxes) of expression symbols are generally scattered over the expression zone, whereas such pixels of the symbols of a normal text line predominantly lie on one or two straight lines. So, if the standard deviation (σ_y) among the y-coordinates of the lower-most black pixels of the expression symbols is calculated, a value much larger than the similar σ_y-value calculated for a normal text line is obtained.

Based on these observations, the following four features (f_{ws}, f_{ms}, f_{mh} and f_{mo}) are defined and integrated through one of the averaging schemes discussed in the next section. The feature, f_{ws} captures the property related to the white space surrounding (above and below) a text line and is measured as,$f_{ws} = 1 - e^{(-\frac{r}{r_\mu})}$, where r denotes the average of the white space (measured in number pixel rows) above and below a text line and r_μ denotes the mean of the white space between two consecutive text lines. In case of the first line, only the line below it is considered to measure r (similarly, for the last text line its preceding line is considered).

The second feature (f_{ms}) is designed to measure the scatteredness of the constituent symbols in a text line and is defined as $f_{ms} = 1 - e^{-\sigma_y}$, where σ_y denotes the standard deviation among the y-coordinates of the lower-most pixels of the symbols (i.e. connected components) of a text line. The feature, f_{mh} measures the height (h) of a text line in terms of pixel rows and compares it to the mean (h_μ) of all h-values. The value of f_{mh} is computed as, $f_{mh} = 1 - e^{(-\frac{h}{h_\mu})}$.

The fourth feature (f_{mo}) keeps track of the occurrence of a few mathematical operator symbols that often appear in expressions [12]. Only 12 operators namely, '=', '+' '-', '/', '(', ')', '[', ']', '{', '}', '<', and '>' (here, 'minus', 'fraction line', 'bar' all are treated as the same symbol '-' because adjacent symbols or context are to be considered to resolve their meaning which is not necessary for the present purpose) are considered for computation of f_{mo}. If k_1 ($k_1 \leq 12$) denotes the number of operators identified in a text line and q_i denotes the probability of occurrence of the i-th symbol then f_{mo} is defined as, $f_{mo} = 1 - e^{-k_1 \sum_{i=1}^{k_1} q_i}$.

Note that each of the features are normalized and their values (real) are bounded in the interval [0, 1]. It is also to be noted that the features are computed in such a way that the higher values of a feature indicate the higher chances for occurring of a particular characteristic of displayed expressions. The rationale behind the usage of the exponential functions is to have a significant change in the corresponding feature value for a slight change in the quantity being measured. For example, the value f_{ws} increases rapidly with a slow increase in the quantity r/r_μ.

2.2 Embedded expressions

Design of an extraction technique started studying the corpus [12] of 400 scanned pages of scientific documents containing 3101 embedded expressions. Two existing commercial OCR systems (that are often used commercially for converting papers into electronic form) were engaged to recognize these pages. Recognition results for sentences with and without expressions are separated for further investigation. Important to note that similar types of observations are resulted in from output of both the OCRs. The set of observations that are relevant in the present context is as follows: (i) Sentences without expressions are recognized with almost no error. (ii) Also, high recognition accuracy is obtained for normal text words in sentences with expressions. (iii) Many expression symbols (mostly Greek letters, majority of mathematical operators, special symbols, etc.) are either rejected (signaled by some special symbol) or misrecognized with a suspicion mark. (iv) In the context of writing embedded expressions, frequent use of certain words or phrases (e.g. *let*, *assume*, *such that*, *the following*, *let us consider*, *is given by*, etc.) motivates us to incorporate linguistics knowledge for extraction of embedded expressions [12]. It is observed that N-gram (N =1, 2, and 3) based category profile of one class (e.g. sentences with expressions) markedly differ from the other. Let C_E denotes the category of sentences containing embedded expressions and (p_i) is the probability that the i-th sentence belongs to the category C_E is computed. In our approach, these observations are captured through five word-level features such as f_{mc}, f_{ce}, f_{ts}, f_{ms}, and f_{cd} are described next.

The feature, f_{mc} measures the confidence of OCR in recognition of a word, w_i and defined as, $f_{mc} = 1-e^{-\frac{c_{off}}{c}}$, where c denotes the confidence of the OCR for recognition of the word, w_i and c_{off} is the offset confidence level based on which the OCR either accepts a recognition result or suspects it to be wrong. For many function words like '*cosh*', '*argmin*', '*lim*', etc. high character level similarity measures are obtained but still the OCR may suspect their recognition since they do not appear in a common dictionary. However, in our approach, a finite automata is maintained to spot occurrence of any such function word. For a function word, f_{mc} is assigned to 1. Similarly, most of the OCR systems properly recognize mathematical operators that can be typed directly from the keyboard (e.g. '=', '+', etc.). These symbols mostly segmented as individual words and f_{mc} values for such words are also assigned to 1.

The second feature, f_{ce} measures the inclination of the sentence containing the word, w_i, towards belonging to the category C_E and it is set to p_i as discussed before. The feature, f_{ts} records the type style of a word. Type style (i.e. regular, italic, and bold) is checked at the character level following the approach presented in [13]. The value of f_{ts} for a word is measured as, $f_{ts} = 1 - e^{-k_2}$, where k_2 is the number of characters in the word for which italic or bold styles are detected.

The feature, f_{ms} measures the scatteredness of the constituent symbols in a word and is computed following the method explained above in section 2.1. Another factor, f_{cd} measures the inter character distance within the word. Let g denote the average of the inter-character gaps for the word under evaluation and g_μ denotes the mean inter-character gap for normal text words then f_{cd} is computed as, $f_{cd} = 1 - e^{(-\frac{g}{g_\mu})}$. The value of g_μ is computed considering a few normal text words for which the OCR shows high recognition confidence.

3 Extraction of expressions

Extraction of displayed expressions: Since a displayed expression appear as a separate text line in a document, extraction of these expressions concentrate on text lines only to decide which text lines correspond to displayed expressions. The features described in section-2.1 are evaluated for each text line. Let m_1 be the total number of text lines in a document image. For all the m_1-lines the four features (f_{ws}, f_{ms}, f_{mh}, and f_{mo}) are evaluated and a $4 \times m_1$ evaluation matrix is formed as follows:

$$V_s = \begin{bmatrix} f_{ws}^1 & f_{ws}^2 & \cdots & f_{ws}^{m_1} \\ f_{ms}^1 & f_{ms}^2 & \cdots & f_{ms}^{m_1} \\ f_{mh}^1 & f_{mh}^2 & \cdots & f_{mh}^{m_1} \\ f_{mo}^1 & f_{mo}^2 & \cdots & f_{mo}^{m_1} \end{bmatrix} \quad (1)$$

Each column in matrix V_s represents each text line in the image and consists of four values corresponding to four features that reflects four different aspects for that line to be a displayed expression. Next, these feature values get combined and mapped into a scalar by using one of the averaging methods namely, (i) arithmetic mean, (ii) geometric mean, (iii) harmonic mean, (iv) weighted mean, etc. The function M_n transforms the $4 \times m_1$ matrix V_s into a $1 \times m_1$ matrix V_s' as follows:

$$V_s' = (f_s^1, f_s^2, \cdots, f_s^{m_1}) \quad (2)$$

where $f_s^i = M_n(f_{ws}^i, f_{ms}^i, f_{mh}^i, f_{mo}^i)$; i varies from 1 to m_1. M_n defining one of the averaging methods gives each text line a degree of qualification (or figure of merit) to be a displayed expression. Higher f_s-value of a line indicates better prospect for that line to be a displayed expression.

Looking at the f_s^i values in V_s', the text lines representing displayed expressions are selected against a threshold (τ_1) determined empirically. The i-th text line is assumed to be a displayed expression if $f_s^i > \tau_1$. A portion of the dataset is used to determine the value of τ_1. At present, a simple supervised technique is involved to find the value of τ_1. Note that the value of τ_1 will vary with the use of different averaging method as M_n. For instance, if the *arithmetic mean* is used to define M_n^a then for a set of 1500 labeled lines among which 263 lines correspond to displayed expression τ_1 is computed to be 0.73.

Extraction of embedded expressions: Features discussed in section 2.2 are used at two steps to extract embedded expression fragments. At first, f_{mc} and f_{ce} are computed for each word and next, based on their values the word is tagged as *mathematical* or not. A large number of normal text words are ignored by looking at the values of f_{mc} and f_{ce} at the first stage. The identifications of *mathematical* words are further verified for their final acceptance. For this purpose, other three features (i.e. f_{ts}, f_{ms}, and f_{cd}) are evaluated for each of these initially tagged *mathematical* words.

Mathematically, f_{mc} and f_{ce} are evaluated for each word and then averaged by using M_n as follows:

$$f_{sus} = M_n(f_{mc}, f_{ce})$$

Basically, M_n assigns a confidence score to each word to be *mathematical* i.e. part of an embedded expression. If this measure is above a threshold (τ_2), then the word is deemed as *mathematical*. Let m_2 be the number of such words which are further verified to pinpoint the zones correspond to embedded expressions. For this purpose, other three features (i.e. f_{ts}, f_{ms}, and f_{cd}) are evaluated for each of the m_2 words and a $3 \times m_2$ matrix is formed as follows:

$$V_w = \begin{bmatrix} f_{ts}^1 & f_{ts}^2 & \cdots & f_{ts}^{m_2} \\ f_{ms}^1 & f_{ms}^2 & \cdots & f_{ms}^{m_2} \\ f_{cd}^1 & f_{cd}^2 & \cdots & f_{cd}^{m_2} \end{bmatrix} \quad (3)$$

Next, feature values under each column get combined by using some form of M_n and V_w is transformed into a $1 \times m_2$ matrix i.e. V_w' as follows:

$$V_w' = (f_w^1, f_w^2, \cdots, f_w^{m_2}) \quad (4)$$

where $f_w^i = M_n(f_{ts}^i, f_{ms}^i, f_{cd}^i)$, i varies from 1 to m_2.

Looking at the f_w value, the identification of a word representing embedded expression is accepted against a threshold (τ_3) determined empirically. A word is suspected as *mathematical* if $f_{sus} > \tau_2$ and this identification is accepted if $f_w > \tau_3$. If adjacent words in a sentence are labeled as mathematical then they are grouped together according to their positional proximity to form embedded expressions. Computation of τ_2 and τ_3 follows the same approach used for computing τ_1.

4 Experimental results

The experiment has been carried out on 200 scanned pages among which 150 pages are taken from the dataset described in [12] and the remaining 50 pages are considered from the INFTY database [14]. INFTY images are considered only for extraction of displayed expressions because these images do not represent the true presence of embedded expressions in real documents. Expressions are labeled following the approach proposed in [12]. Altogether, 1163 embedded and 1039 displayed expressions are spotted in these pages. The dataset is divided into two parts. The first part (say, Part-I) is used to compute the threshold values (τ_1, τ_2 and τ_3) as described in section-3 and to learn the weights of the function M_n when weighted average is used to average the feature values. This part consists 50 pages, 35 from [12] and another 15 from [14]. These pages, altogether, contains 263 displayed and 271 embedded expressions. The rest of the pages forms Part-II that is consisting of 150 pages. The Part-II is used in testing the efficiency of the proposed method. In total, it contains 776 displayed and 892 embedded expressions.

Computation of Extraction Efficiency: One of the following four cases can occur during extraction of expressions: (i) *Perfect*: An extraction is correct i.e. the bounding box corresponding to the extracted zone finds a match in the groundtruth. (ii) *Partial*: An extraction is partially correct i.e. the extracted zone shows a bounding box that partially matches the groundtruth. (iii) *Missed*: An expression is missed i.e. is not extracted at all. (iv) *False*: False identification i.e. an extracted zone does not actually contain any mathematics at all. Extraction of displayed expressions do not exhibit *partial* extraction as an entire text line is extracted. For a multi-line expression, each line is treated as separate displayed expression.

To evaluate the extraction results, let T_1 be the total number of expressions (embedded or displayed or both) properly extracted, T_2 be the number for which partial extraction is done, T_3 be the number of expressions, which are missed (i.e. not extracted) and T_4 be the number of zones that do not contain any expression (false identification). The extraction efficiency (ϵ) for an input page is computed as

$$\epsilon = \frac{T_1 + \sum_{i=1}^{T_2} \alpha_i - (\beta T_3 + \gamma T_4)}{T} \tag{5}$$

where, $T = T_1 + T_2 + T_3$. α_i ($0 < \alpha_i < 1$) determines the amount of reward we give to each partial extractions and β and γ determine the amount of penalty we give to missed and false identifications. α_i is computed as $\alpha_i = \frac{S_e}{S_a}$, where S_e is the number of symbols found within the i-th extracted zone and S_a is the number of symbols present in the zone. The value of S_a is obtained from the groundtruth where MathML presentation tags are available for each expression. The experiment set β and γ to 1. However, an empirically chosen value of γ may penalize false identifications is a more judicious manner. If a document image, D_i shows an extraction accuracy of ϵ_i, then an average efficiency ϵ_{av} is computed in a dataset of N documents as follows, $\epsilon_{\text{av}} = \frac{1}{N} \sum_{i=1}^{N} \epsilon_i$.

We conducted five different experiments varying the averaging methods for feature integration. All these five experiments use the Part-I portion of the dataset to compute the thresholds, i.e. τ_1, τ_2 and τ_3. Let E-I, E-II, E-III, E-IVA and E-IVB denote these five experiments. The first three experiments use arithmetic, geometric and harmonic means, respectively for averaging the feature values. Hence, the forms of M_n as defined in section 3 are used in E-I, E-II and E-III, respectively. The results produced by these three experiments are reported in Table-1. It is seen that use of arithmetic mean gives better extraction results than using geometric or harmonic mean. This is mainly due to the values of τ_1, τ_2 and τ_3. In case of using geometric and harmonic means these values are calculated from narrow margins between text and expression zones. However, in case of using arithmetic mean a wider margin is obtained. Because of these threshold values, it is noted that E-II and E-III do miss identification of many expressions although they give less number of false identifications.

The experiments E-IVA and E-IVB use weighted average. In E-IVA the weights are computed from the variances observed in feature values. E-IVB employs an optimization technique to find an optimal set of weights. A gradient descent algorithm is implemented for this purpose. A set of training data (i.e. Part-I) is iteratively used to find a local optima and the weight values returned by the algorithm are then used. This shows that using weighted mean the extraction results are significantly improved. The experiment E-IVB gives the best results among the five. In this case, an overall efficiency, ϵ = 0.89 is achieved and this is the best result what we could achieve in this study.

Error Analysis: Analysis of results shows that the false identifications of text lines as displayed expressions are sometimes due to presence of chapter/section titles, table/figure captions, etc. that are generally separated by large white space giving high value of the factor, f_{ws}. Sometimes, false recognition of a few letters as mathematical operator (e.g. 'C' as '(', *hyphen* as *minus*, etc.) increases the value of the factor, f_{mo}. Identification of displayed expression is missed in some cases (2.8% at most as in E-IVB) where the expressions do not exhibit the features used for their extraction. The expressions that consist of only a few symbols and are densely typed with successive text lines create some problem during extraction.

In a few cases, text lines consisting of embedded expressions are identified as displayed expressions. However, in

Table 1. Summary of Expression Extraction Results.

Nature of Extraction	Accuracy									
	Embedded (# Expressions: 892)					Displayed (# Expressions: 776)				
	E-I	E-II	E-III	E-IVA	E-IVB	E-I	E-II	E-III	E-IVA	E-IVB
Perfect	761 (85.3%)	692 (77.6%)	603 (67.6%)	788 (88.3%)	740 (95.4%)	728 (93.8%)	669 (86.2%)	633 (81.6%)	740 (95.4%)	754 (97.2%)
Partial	64 (7.2%)	84 (9.4%)	117 (13.1%)	70 (7.8%)	72 (8.1%)	NIL	NIL	NIL	NIL	NIL
Missed	67 (7.5%)	116 (13%)	172 (19.3%)	45 (5.1%)	32 (3.6%)	48 (6.2%)	107 (13.8%)	143 (18.4%)	36 (4.6%)	22 (2.8%)
False	51	43	29	39	27	35	18	9	23	18
Efficiency (ϵ)	0.76	0.65	0.53	0.82	0.87	0.83	0.7	0.62	0.88	0.92

these cases text line contains more mathematics than normal text. However, identification of the mathematics intensive text lines as displayed expressions is not a severe error. In case of embedded expressions, *partial* and *missed* extractions sometimes occur due to short expression fragments containing only 2 or 3 symbols. False identifications (i.e. a text portion is wrongly identified as embedded expression) are encountered for documents with excessive degradation due to aging, improper digitization, etc. where large number of broken and merged characters appear and disturb the extraction process.

5 Conclusions

A method for the extraction of mathematical expressions contained in scientific documents is presented. Extraction of both types of expressions, i.e. embedded as well as displayed expressions are considered. The method is based on a set of easily computable features. A publicly available dataset is used in this study. A performance evaluation method is presented to measure the efficiency of the proposed method. The present study points to several new research avenues that can be explored in future. The overall performance itself demands further research in this area. Newer features and classification methods may help in designing a better system. The present method does use precomputed threshold values (e.g. τ_1, τ_2, and τ_3). Instead, use of learning algorithms (e.g. neural net, support vector machines, etc.) for text/maths discrimination is planned as a future extension of the current study. Moreover, integration of the proposed approach with one of the existing OCR systems and evaluation of its performance in recognizing scientific documents would be experimented in future.

Acknowledgement: The author would like to express his appreciation to Prof. M. Suzuki for his kind favor towards freely sharing the INFTY [14] database.

References

[1] D. Blostein and A. Grbavec, "Recognition of Mathematical Notation," *Handbook of Character Recognition and Document Image Analysis*, Eds. H. Bunke and P.S.P. Wang, World Scientific Publishing Company, pp. 557-582, 1997.

[2] K-F. Chan and D-Y. Yeung, "Mathematical Expression Recognition: A Survey," *IJDAR*, vol. 3, no. 1, pp. 3-15, 2000.

[3] U. Garain and B.B. Chaudhuri, "On OCR of Printed Mathematical Expressions," *Digital Document Processing*, Ed. B. B. Chaudhuri, Advances in Pattern Recognition, Springer-Verlag, London Ltd. pp. 235-259, 2007.

[4] H.J. Lee and J.-S. Wang, "Design of a Mathematical Expression Understanding System," *Pattern Recognition Letters*, vol. 18, no. 3, pp. 289-298, 1997.

[5] R.J. Fateman "How to find mathematics on a scanned page," *Proc. of the SPIE*, vol.3967, pp. 98-109, San Jose, California, USA, 1999.

[6] K. Inoue, R. Miyazaki, and M. Suzuki, "Optical Recognition of Printed Mathematical Documents," *Proc. of Asian Technology Conference in Mathematics* (ATCM), Springer-Verlag, pp. 280-289, 1998.

[7] J.-Y. Toumit, S. Garcia-Salicetti, and H. Emptoz, "A Hierarchical and Recursive Model of Mathematical Expressions for Automatic Reading of Mathematical Documents," *Proc. of the 5th ICDAR*, pp. 119-122, Bangalore, India, 1999.

[8] A. Kacem, A. Belaid and M. Ben Ahmed, "Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context," *IJDAR*, vol. 4, no. 2, pp. 97-108, 2001.

[9] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "INFTY – An Integrated OCR System for Mathematical Documents," *Proc. of ACM Symposium on Document Engineering* (DocEng), pp. 95-104, France, 2003.

[10] S.P. Chowdhury, S. Mandal, A.K. Das and B. Chanda, "Automated Segmentation of Math-Zones from Document Images," *Proc. of the 7th ICDAR*, pp. 755-759, Edinburgh, Scotland, 2003.

[11] J. Jin, X. Han and Q. Wang, "Mathematical Formulas Extraction," *Proc. of the 7th ICDAR*, Edinburgh, Scotland, pp. 1138-1141, 2003.

[12] U. Garain, "Recognition of Printed and Handwritten Mathematical Expressions," *Ph.D. Thesis,* Indian Statistical Institute, Kolkata, India, 2005.

[13] B.B. Chaudhuri and U. Garain, "Extraction of type style based meta-information from imaged documents," *IJDAR*, vol. 3, no. 3, pp.138-149, March, 2001.

[14] S. Uchida, A. Nomura, and M. Suzuki, "Quantitative analysis of mathematical documents," *IJDAR*, Vol. 7, No. 4, pp. 211-218, 2005.