IJDAR (2005) 7: 241–259
Digital Object Identifier (DOI) 10.1007/s10032-004-0140-5

International Journal on Document Analysis and Recognition IJDAR

A corpus for OCR research on mathematical expressions

Utpal Garain, B.B. Chaudhuri

Computer Vision & Pattern Recognition Unit, Indian Statistical Institute, 203 B. T. Road, Calcutta-700 035, India
(e-mail: {utpal,bbc}@isical.ac.in)

Received: July 10, 2003 / Accepted: November 22, 2004
Published online: March 18, 2005 – © Springer-Verlag 2005

Abstract. This paper is concerned with research on OCR (optical character recognition) of printed mathematical expressions. Construction of a representative corpus of technical and scientific documents containing expressions is discussed. A statistical investigation of the corpus is presented, and usefulness of this analysis is demonstrated in the related research problems, namely, (i) identification and segmentation of expression zones from the rest of the document, (ii) recognition of expression symbols, (iii) interpretation of expression structures, and (iv) performance evaluation of a mathematical expression recognition system. Moreover, a groundtruthing format has been proposed to facilitate automatic evaluation of expression recognition techniques.

Keywords: OCR – Mathematical expressions – Database – Groundtruthing – Statistical learning – Performance evaluation

1 Introduction

Automatic transcription of printed scientific and technical documents into electronic format largely depends on the success in recognizing the typeset mathematics. Several studies dealing with recognition of printed mathematics have been reported in the literature. These research efforts have been surveyed in [1–3]. From these reports it is understood that unavailability of a suitable corpus of expressions has prompted the researchers to define their own data set for testing their algorithms. As a result, replication of experiments and comparison of performance among different methods has become a difficult task.

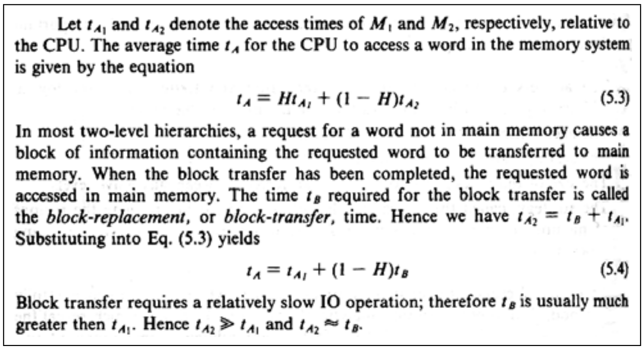
In this correspondence, we present a database of mathematical expression images that would facilitate research in automatic understanding of expressions. The only relevant database available so far is the University of Washington English/Technical Document Database III (UW-III) [4]. However, the database is mainly constructed for general OCR (optical character recognition) research and contains 25 groundtruthed (into TEX) document pages containing about 100 expressions. Therefore, it does not seem to be a representative corpus for the respective research. Another drawback of this data set is that groundtruthing of expressions into TEX only does not support an in-depth analysis of recognition performance [5–7]. Another freely available source of expressions is the set used by Raman for his Ph.D. work [8]. However, the expressions available here are synthetic (generated by TEX) and isolated (i.e., not a part of any document) in nature.

In this paper, we discuss the contents of a database of printed scientific documents collected at the Computer Vision and Pattern Recognition Unit of the Indian Statistical Institute, Kolkata, India and describe how this database addresses various research considerations related to recognition of printed mathematical expressions. This work is an extension of our earlier effort presented in [9].

In printed documents, expressions appear in two modes, namely, *embedded* (mixed with text and also referred to as inline expression) and *displayed* (typed on a separate line). Figure 1 shows a typical example of such a document. The content of the database, therefore, is arranged into a two-level hierarchy. At the top level are 400 scientific and technical document images containing mathematical expressions. For each document, its embedded and displayed expressions are collected into two different files. Correspondence between a top-level document with its lower-level files storing embedded and displayed expressions is maintained through the naming convention for files. Expressions are groundtruthed following a specified format explained later.

However, this paper not only discusses the content of the database and its organization, it is also focused on other issues such as measuring the comprehensiveness of the corpus, analyzing it for several research considerations, and designing an automated tool for testing and evaluating the performance of an expression recognition system. The rest of the paper is organized as follows.

Correspondence to: U. Garain (e-mail: utpal@isical.ac.in)



Let t_{A_1} and t_{A_2} denote the access times of M_1 and M_2, respectively, relative to the CPU. The average time t_A for the CPU to access a word in the memory system is given by the equation

$$t_A = Ht_{A_1} + (1 - H)t_{A_2} \tag{5.3}$$

In most two-level hierarchies, a request for a word not in main memory causes a block of information containing the requested word to be transferred to main memory. When the block transfer has been completed, the requested word is accessed in main memory. The time t_B required for the block transfer is called the *block-replacement*, or *block-transfer*, time. Hence we have $t_{A_2} = t_B + t_{A_1}$. Substituting into Eq. (5.3) yields

$$t_A = t_{A_1} + (1 - H)t_B \tag{5.4}$$

Block transfer requires a relatively slow IO operation; therefore t_B is usually much greater then t_{A_1}. Hence $t_{A_2} \gg t_{A_1}$ and $t_{A_2} \approx t_B$.

Fig. 1. A sample page containing embedded as well as displayed expressions.

The structure of the proposed database, its content, sampling procedure, and method of groundtruth generation are outlined in Sect. 2. Section 3 presents a statistical study of the database to measure the comprehensiveness of the proposed corpus. This section also contains a linguistic analysis to assist in the identification of embedded expressions in scientific documents. Next, Sect. 4 demonstrates how the proposed corpus contributes toward testing of an expression recognition system and presents an automatic way to compute a single figure of merit for measuring overall recognition performance of a system. Section 5 concludes the paper.

2 Organization of the database

The proposed database contains 400 document images containing 2,459 displayed and 3,101 embedded expression fragments. Both real (297 pages) and synthetic (generated by TEX or Microsoft Word) documents (103 pages) are present in the data set. Real documents are collected from (i) books of various branches of science, (ii) science journals, (iii) proceedings of technical conferences, (iv) question papers (College/University level examination), etc.

Synthetic documents are selected from sources that are available in Microsoft Word or TEX format. Several electronically available journals, conference proceedings, and Internet sites related to various science subjects were considered for this purpose. A few pages were selected from the technical articles published by the members of our research unit. Documents in the database were grouped into three groups depending on the abundance of expressions in the documents. Group I refers to those documents where the number of expressions is relatively small compared to the other two categories. Similarly, group II points to those pages where the density of expressions is higher than that of group I pages, and documents under group III show the highest density of expressions. A summary of the collected samples is given in Table 1.

Several factors influence the choice of materials, some of which are described below:

- *Relative frequency of expressions*: The documents show variation in the number of expressions contained in them. Sample documents are divided into three groups based on the number of displayed expressions and percentage of sentences containing embedded expressions (Table 1).
- *Nature of expressions*: Documents are selected from various branches of science to cover a wide range of expressions that may appear in the literature. The details of the specific topics covered in the database are discussed later. Pages containing expressions having varying geometric structures and layouts are considered to make the data set a representative one.
- *Variations in typeset*: The data set contains documents published using old mechanical typeset as well as those printed by offset and other modern machines.
- *Page layout*: Documents may be printed in single-column or multicolumn format. Apart from text and expressions, they may contain graphs, charts, illustrations, half-tone pictures, etc.
- *Aging effect*: Many important scientific/technical materials that are not available in electronic form are over 100 years old. OCR conversion of these documents into electronic form remains a big challenge. The data set contains samples from older as well as recently published materials to reflect this aging effect.
- *Other degradations*: Photocopied versions, pages from bound journals, damaged documents, etc. are also present in the data set to represent different levels of degradation and distortion.

2.1 Digitization of samples

HP flatbed scanners (Scanjet 5470C and Scanjet ADF) are used to digitize real samples into `TIFF` files. Documents are scanned in gray scale, and resolution varies from 75 dpi to 600 dpi. Old materials and documents printed in smaller font sizes are scanned with higher resolutions. Scanning at different dpi values helps to study the effect of resolution on recognition performance. On the other hand, no scanning is involved for synthetic documents, and therefore such scanning is free from common digitization errors. These noise-free binary images are generated either by TEX or by Microsoft Word editor.

2.2 Generation of truthed data

In the current database, each sample page (say, `docxxx.tif`) generates three files with extensions `.atr`, `.emb`, and `.dis`, as shown in Fig. 2. The `docxxx.atr` file contains a set of attributes defined for the page `docxxx.tif`. Some of the attributes are the `page ID`, `publication information`, `page type`, etc. Attributes such as `degradation type` (*original/photocopy*), `salt/pepper noise` (*yes/no*), `blurred` (*yes/no*), etc. refer to the page condition and describe the visual condition of a given sample page. Many of these attributes are identical to those used in the UW Document Image Database [4].

Table 1. Coverage of the document database

Group label	Source	#Samples (pages)	Avg. no. of displayed exps per page	% of sentences containing embedded exps
Group I	Real	116	2.16	7.61%
	Synthetic	44	3.07	8.82%
Group II	Real	101	6.24	19.23%
	Synthetic	32	7.11	17.05%
Group III	Real	80	11.45	40.57%
	Synthetic	27	10.61	38.12%
Total		400	6.11	20.04%

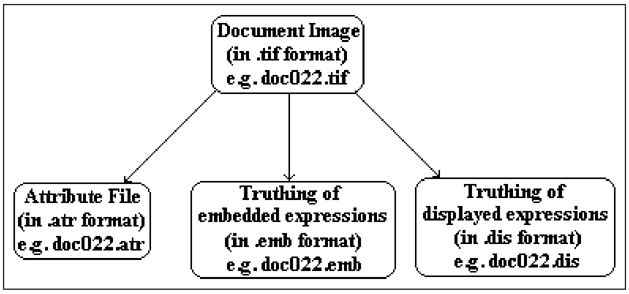


Fig. 2. Groundtruthing of scientific document images

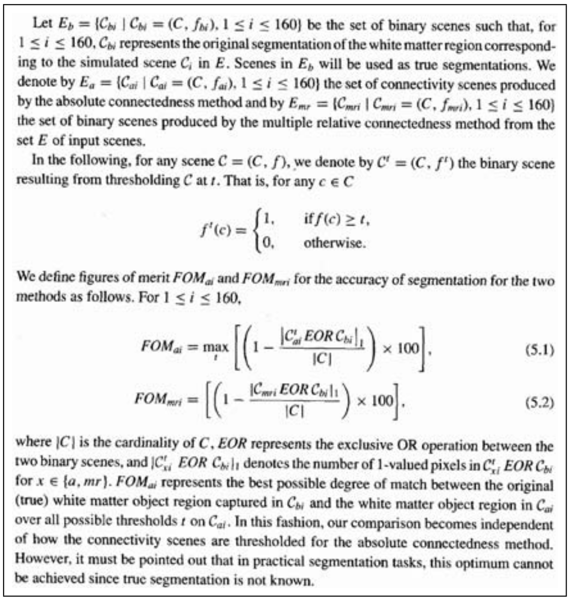
Documents in the database show variety in their page layout. Some of the documents are in single-column format, while others are in multicolumn format. Several documents contain graphs, charts, illustrations, halftones, etc. Figure 3 shows a few documents present in the database. However, an entire page is not considered for generation of groundtruth, and only the blocks containing text and mathematical expressions are considered. Several techniques (a summary of them can be found in [10]) have been proposed for automatic analysis of page layout, and a modified version of the technique proposed by Pavlidis and Zhou [11] has been employed in our system to locate the blocks containing text and expressions.

Groundtruthing of embedded expressions. The file `docxxx.emb` describes truth for the embedded expressions contained in the page `docxxx.tif`. Embedded expressions are recorded along with the sentence containing it. A sentence is said to have one or more embedded expressions if it needs the use of `math mode` in case the sentence was prepared using TEX. Approaches presented in [12–18] are studied to develop an automatic way of identifying zones containing embedded expressions in the documents. But it is observed that frequent manual intervention is needed to make the results error-free. However, the addition of some N-gram-based linguistic properties (explained later) does help considerably to improve the identification process.

The truthed data for a page is contained within a <page> and </page> tag pairs. The image file name is stored within tags, namely, <imagefile> and </imagefile>. A sentence containing embedded expressions is recorded within the tag pairs <sentence> and </sentence>. Since multiple sentences can contain embedded expressions, there are multiple instances of <sentence> and </sentence> tag pairs. On the other hand, a sentence may consist of multiple text lines, and therefore a line content is enclosed by a <line> and </line> tag pair. An upper-level tag structure for groundtruthing of embedded expressions is presented below:

```
<page>
 <imagefile> ... </imagefile>
 <sentence>
   <line>
    THe bounding box of the line is recorded.
    Next, the text and math portions, if any,
    are recorded separately.
   </line>
   <line>
    ...
   </line>
   .
   .
   .
 </sentence>
 <sentence>
   ...
 </sentence>
 .
 .
 .
</page>
```

Each line of a sentence containing an expression is separately marked with a pair of (x, y) coordinates for the top-left and bottom-right corners of the minimum upright rectangular box (called the *bounding box* and recorded within <bbox> </bbox> tag pairs) enclosing that text line. The expression zones within a text line are highlighted by the bounding box coordinates of the expression zone. The text and the expression portions of a text line are separately truthed within <text> </text> and <math> </math> tag pairs, respectively. The tag structure for a line is shown below:



Let $E_b = \{\mathcal{C}_{bi} \mid \mathcal{C}_{bi} = (C, f_{bi}), 1 \le i \le 160\}$ be the set of binary scenes such that, for $1 \le i \le 160$, $\mathcal{C}_{bi}$ represents the original segmentation of the white matter region corresponding to the simulated scene $\mathcal{C}_i$ in E. Scenes in E_b will be used as true segmentations. We denote by $E_a = \{\mathcal{C}_{ai} \mid \mathcal{C}_{ai} = (C, f_{ai}), 1 \le i \le 160\}$ the set of connectivity scenes produced by the absolute connectedness method and by $E_{mr} = \{\mathcal{C}_{mri} \mid \mathcal{C}_{mri} = (C, f_{mri}), 1 \le i \le 160\}$ the set of binary scenes produced by the multiple relative connectedness method from the set E of input scenes.

In the following, for any scene $\mathcal{C} = (C, f)$, we denote by $\mathcal{C}^t = (C, f^t)$ the binary scene resulting from thresholding $\mathcal{C}$ at t. That is, for any $c \in C$

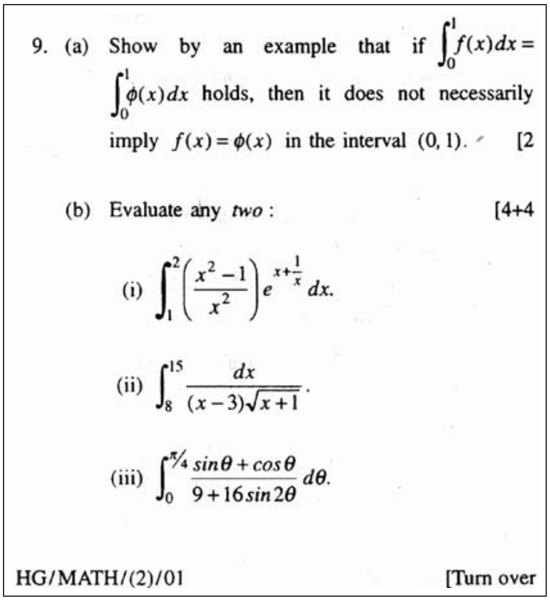
$$f^t(c) = \begin{cases} 1, & \text{if } f(c) \ge t, \\ 0, & \text{otherwise.} \end{cases}$$

We define figures of merit FOM_{ai} and FOM_{mri} for the accuracy of segmentation for the two methods as follows. For $1 \le i \le 160$,

$$FOM_{ai} = \max_t \left[\left(1 - \frac{|\mathcal{C}^t_{ai}\, EOR\, \mathcal{C}_{bi}|_1}{|C|} \right) \times 100 \right], \tag{5.1}$$

$$FOM_{mri} = \left[\left(1 - \frac{|\mathcal{C}_{mri}\, EOR\, \mathcal{C}_{bi}|_1}{|C|} \right) \times 100 \right], \tag{5.2}$$

where $|C|$ is the cardinality of C, EOR represents the exclusive OR operation between the two binary scenes, and $|\mathcal{C}^t_{xi}\ EOR\ \mathcal{C}_{bi}|_1$ denotes the number of 1-valued pixels in $\mathcal{C}^t_{xi}\ EOR\ \mathcal{C}_{bi}$ for $x \in \{a, mr\}$. FOM_{ai} represents the best possible degree of match between the original (true) white matter object region captured in $\mathcal{C}_{bi}$ and the white matter object region in $\mathcal{C}_{ai}$ over all possible thresholds t on $\mathcal{C}_{ai}$. In this fashion, our comparison becomes independent of how the connectivity scenes are thresholded for the absolute connectedness method. However, it must be pointed out that in practical segmentation tasks, this optimum cannot be achieved since true segmentation is not known.



9. (a) Show by an example that if $\int_0^1 f(x)dx = \int_0^1 \phi(x)dx$ holds, then it does not necessarily imply $f(x) = \phi(x)$ in the interval $(0, 1)$. [2

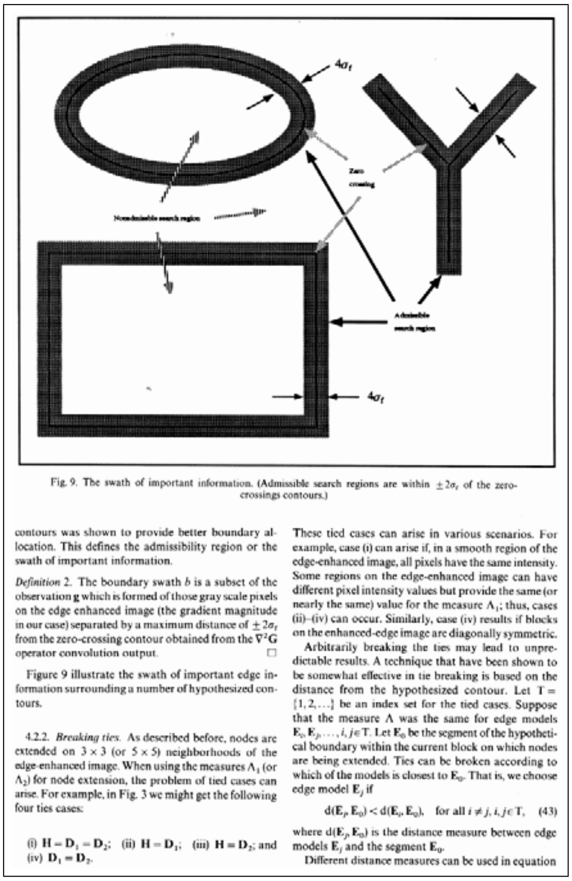
(b) Evaluate any *two* : [4+4

(i) $\int_1^2 \left(\frac{x^2 - 1}{x^2} \right) e^{x + \frac{1}{x}} dx.$

(ii) $\int_8^{15} \frac{dx}{(x-3)\sqrt{x+1}}.$

(iii) $\int_0^{\pi/4} \frac{sin\theta + cos\theta}{9 + 16 sin 2\theta} d\theta.$

HG/MATH/(2)/01 [Turn over



$4\sigma_t$
Zero crossing
Non-desirable search region
Admissible search region
$4\sigma_t$

Fig. 9. The swath of important information. (Admissible search regions are within $\pm 2\sigma_t$ of the zero-crossings contours.)

contours was shown to provide better boundary allocation. This defines the admissibility region or the swath of important information.

Definition 2. The boundary swath b is a subset of the observation **g** which is formed of those gray scale pixels on the edge enhanced image (the gradient magnitude in our case) separated by a maximum distance of $\pm 2\sigma_t$ from the zero-crossing contour obtained from the ∇^2G operator convolution output. □

Figure 9 illustrate the swath of important edge information surrounding a number of hypothesized contours.

4.2.2. *Breaking ties.* As described before, nodes are extended on 3 × 3 (or 5 × 5) neighborhoods of the edge-enhanced image. When using the measures Λ_1 (or Λ_2) for node extension, the problem of tied cases can arise. For example, in Fig. 3 we might get the following four ties cases:

(i) $\mathbf{H} = \mathbf{D}_1 = \mathbf{D}_2$; (ii) $\mathbf{H} = \mathbf{D}_1$; (iii) $\mathbf{H} = \mathbf{D}_2$; and (iv) $\mathbf{D}_1 = \mathbf{D}_2$.

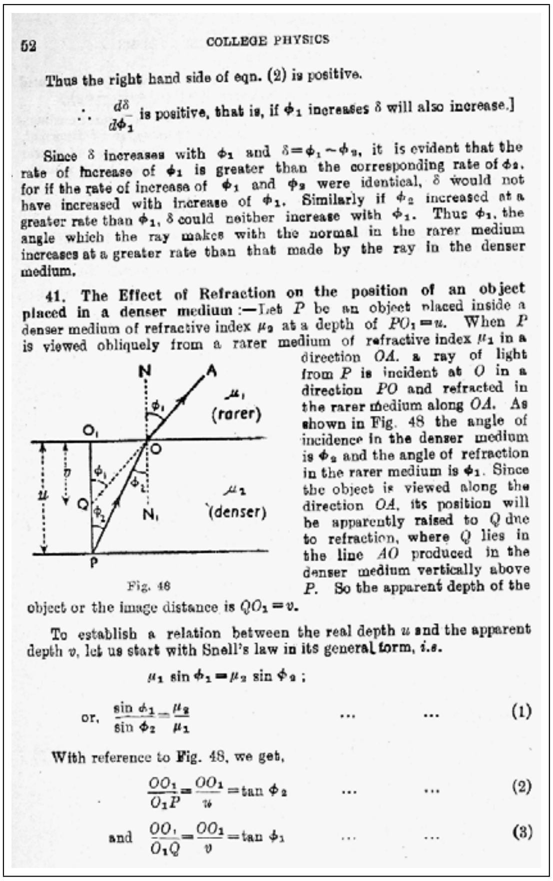
These tied cases can arise in various scenarios. For example, case (i) can arise if, in a smooth region of the edge-enhanced image, all pixels have the same intensity. Some regions on the edge-enhanced image can have different pixel intensity values but provide the same (or nearly the same) value for the measure Λ_1; thus, cases (ii)–(iv) can occur. Similarly, case (iv) results if blocks on the enhanced-edge image are diagonally symmetric.

Arbitrarily breaking the ties may lead to unpredictable results. A technique that have been shown to be somewhat effective in tie breaking is based on the distance from the hypothesized contour. Let T = {1, 2, ...} be an index set for the tied cases. Suppose that the measure Λ was the same for edge models $\mathbf{E}_i, \mathbf{E}_j, \ldots, i, j \in$ T. Let $\mathbf{E}_0$ be the segment of the hypothetical boundary within the current block on which nodes are being extended. Ties can be broken according to which of the models is closest to $\mathbf{E}_0$. That is, we choose edge model $\mathbf{E}_j$ if

$$d(\mathbf{E}_j, \mathbf{E}_0) < d(\mathbf{E}_i, \mathbf{E}_0), \quad \text{for all } i \ne j,\ i, j \in \mathrm{T}, \tag{43}$$

where $d(\mathbf{E}_j, \mathbf{E}_0)$ is the distance measure between edge models $\mathbf{E}_j$ and the segment $\mathbf{E}_0$.

Different distance measures can be used in equation



52 COLLEGE PHYSICS

Thus the right hand side of eqn. (2) is positive.

$\therefore \frac{d\delta}{d\phi_1}$ is positive, that is, if ϕ_1 increases δ will also increase.]

Since δ increases with ϕ_1 and $\delta = \phi_1 \sim \phi_2$, it is evident that the rate of increase of ϕ_1 is greater than the corresponding rate of ϕ_2. for if the rate of increase of ϕ_1 and ϕ_2 were identical, δ would not have increased with increase of ϕ_1. Similarly if ϕ_2 increased at a greater rate than ϕ_1, δ could neither increase with ϕ_1. Thus ϕ_1, the angle which the ray makes with the normal in the rarer medium increases at a greater rate than that made by the ray in the denser medium.

41. The Effect of Refraction on the position of an object placed in a denser medium :—Let P be an object placed inside a denser medium of refractive index μ_2 at a depth of $PO_1 = u$. When P is viewed obliquely from a rarer medium of refractive index μ_1 in a direction OA. a ray of light from P is incident at O in a direction PO and refracted in the rarer medium along OA. As shown in Fig. 48 the angle of incidence in the denser medium is ϕ_2 and the angle of refraction in the rarer medium is ϕ_1. Since the object is viewed along the direction OA, its position will be apparently raised to Q due to refraction, where Q lies in the line AO produced in the denser medium vertically above P. So the apparent depth of the object or the image distance is $QO_1 = v$.

Fig. 48

To establish a relation between the real depth u and the apparent depth v, let us start with Snell's law in its general form, *i.e.*

$$\mu_1 \sin \phi_1 = \mu_2 \sin \phi_2 ;$$

$$\text{or,} \quad \frac{\sin \phi_1}{\sin \phi_2} = \frac{\mu_2}{\mu_1} \quad \ldots \quad \ldots \tag{1}$$

With reference to Fig. 48, we get,

$$\frac{OO_1}{O_1P} = \frac{OO_1}{u} = \tan \phi_2 \quad \ldots \quad \ldots \tag{2}$$

$$\text{and} \quad \frac{OO_1}{O_1Q} = \frac{OO_1}{v} = \tan \phi_1 \quad \ldots \quad \ldots \tag{3}$$

Fig. 3. A few document images present in the database

```
<line>
 <bbox> ... </bbox>
 <text>
  ...
 </text>
 <math>
   ...
 </math>
   .
   .
   .
</line>
```

For each embedded expression, its enclosing bounding box is recorded within <bbox> </bbox> tag pairs. Next, a <GC> </GC> tag pair is used to indicate the geometric complexity (GC) of the expression. GC is mea-

Table 2. Statistics on type styles

Style	Expression-level statistics	
	Embedded	Displayed
Regular	186 (6%)	271 (11%)
Italic	2,078 (67%)	1,254 (51%)
Bold	558 (18%)	516 (21%)
Bold-italic	279 (9%)	418 (17%)

sured by the number of horizontal lines (on which expression symbols are arranged) found in that expression. For example, symbols of expression shown in Fig. 4a are arranged on `9` horizontal lines as explained in Fig. 4b. Therefore, `GC` becomes `9` for this expression. The dominant baseline [7] of an expression is treated as `level 0`, and the `level` number increases above and decreases below the baseline.

Next, the content of the embedded expression is presented. For this purpose, MathML [1] presentation tags are used. However, for a symbol (where it is operator, identifier, or numeral), three additional tag pairs, namely, <level> </level>, <style> </style>, and <truth> </truth>, are used. For example, if t is an identifier used in the expression, its MathML representation (i.e., <mi> t </mi>) is extended as follows:

```
<mi>
 <level> ... </level>
 <style> ... </style>
 <truth> ... </truth>
</mi>
```

The `level` indicates the horizontal level number at which the symbol appears. For example, symbols of the expression in Fig. 4a appear in one of the nine horizontal levels numbered `--3` to `5` including `0`. The `style` indicates the type style (`n`: normal, `b`: bold, `i`: italic, `bi`: bold-italic) of the symbol. The identity of the symbol is given within the <truth> </truth> tag pair.

An algorithm can be designed for the automatic computation of `GC`, symbol `levels`. One such algorithm can be found in [22]. On the other hand, detection of symbol `style` is done by following the algorithms outlined in [19,20]. It is noted that the expression symbols are often typed with a font style, which is different from the dominant style of the document text. In several cases, variations in font faces are also observed, i.e., the expressions are printed in a font different from the one used for the remaining text of the document. Statistics regarding different type styles used in expressions are presented in Table 2. For computing the type style of an expression, it is found that, in many cases, the style of all expression symbols may not be unique, and therefore the results presented in Table 2 show the dominant (determined by the majority of symbols) style in the expressions.

Appendix A shows the groundtruth data for the embedded expressions contained in the page in Fig. 1. If a page has a multicolumn layout, expressions are truthed following the normal reading sequence. In the database, 1084 (out of 5402) sentences are found to have embedded expressions, and the groundtruth for each of these sentences is available in the corpus.

Groundtruthing of displayed expressions. The file `docxxx.dis` truths the displayed expressions contained in the image `docxxx.tif`. Automatic identification of displayed expressions in document images has been achieved by the algorithm proposed in [21]. For a sample page, all displayed expressions are truthed within <page> </page> tag pairs. Each expression is truthed under <math> </math> tags. For a multiline expression, two consecutive expression lines are linked together with a *Thread* pointer. So, if the *Thread* pointer (which can have only binary values) for any expression is `1`, then the current expression continues to the next line. If a page has a multicolumn layout, expressions are truthed following the normal reading sequence.

Within a pair of <math> </math> tags, truthing starts with <bbox> </bbox> tags giving the bounding box coordinates for the expression. Next, the `GC` of the expression is recorded using the tag pairs <GC> </GC>. The expression sequence number, if any exists, is given under the <expno> </expno> tags. Expression content is truthed in a way that is similar to that used for truthing embedded expressions; the only difference is the additional presence of <bbox> </bbox> tag pairs for each symbol of a displayed expression.

Appendix B presents the truthed data for the displayed expressions contained in the page in Fig. 1. In the corpus, there are 2399 displayed expressions, and groundtruth data for each expression are available in 400 `docxxx.dis` files. In addition, 60 synthetic expressions are selected from the data set used in AsTeR research [8]. The reasons for this choice are: (i) it contains examples from various branches of mathematics. The expressions show wide variations in their structural complexity, making the data set a representative one; (ii) for each expression its corresponding correct $\TeX$ string is available, and these strings are used to generate groundtruth; (iii) the data set is free and globally available through the Internet, thus allowing comparative study. Therefore, the total number of expressions present in Part II is 2459. Table 3 gives an idea of how the expressions are divided among different topics and purposes.

2.3 Validation of the truthed data

While generating the truthed data, manual intervention was involved in the following aspects to check possible errors: (i) bounding box for each line containing embedded expressions, bounding box for each embedded and displayed expressions; (ii) `GC` information for each expression; (iii) `level` information associated with each expression symbol; and (iv) symbol identity. Moreover, for a displayed expression the expression sequence number, if any is entered manually. Bounding box information of expression symbols (of displayed expressions) has not been checked as once the bounding box for an ex-

[1] See W3C Math Home at `http://www.w3.org/Math/`.

Table 3. Coverage of the data set of displayed expressions

Source	Area	Expressions	Remarks
Expressions found in 390 technical and scientific documents collected under Part I of the proposed database.	Algebra	439	This data set shows various image level noise like degradation due to aging, digitization errors, etc. Expressions are scanned with resolution varying from 75 to 600 dpi.
	Calculus	205	
	Differential equations	313	
	Integrals	351	
	Logic and set theory	186	
	Statistics and probability	176	
	Trigonometry and geometry	171	
	Vector	254	
	Miscellaneous	304	
AsTeR data set [8]	Series	5	This data set is freely available at `http://www.cs.cornell.edu/info/people/raman/aster/`. These TeXgenerated expressions are free from any digitization errors. The resolution is 300 dpi.
	Logarithms	4	
	Fractions	9	
	Roots	4	
	Sums	3	
	Superscripts and Subscripts	9	
	Limits	2	
	Matrix	1	
	Trigonometry	8	
	Integrals	6	
	Miscellaneous	9	
Total		2,459	

Table 4. Occurrence rate of symbol categories

Symbol (# Classes)	# Occurrences in		Total Count
	embedded	displayed	
Roman letter (52)	9,687	20,361	30,048 (0.36)
Arabic numeral (10)	3,544	6,882	10,426 (0.13)
Greek symbol (41)	1,654	3,493	5,147 (0.06)
Calligraphic letter (16)	71	309	380 (0.01)
Mathematical operator (39)	1,541	10,907	12,448 (0.15)
Relational operator (39)	954	4,693	5,647 (0.07)
Arrow symbol (32)	612	2,123	2,735 (0.03)
Bracket symbol (06)	2,127	4,314	6,441 (0.08)
Misc. symbol (33)	945	3,511	4,456 (0.05)
Punctuation mark (06)	2,268	2,695	4,963 (0.06)
Total (274)	23,403	59,288	82,691 (1.00)

pression is fixed, there is hardly any error in finding a bounding box for individual symbols.

The groundtruth of the text portions for the lines containing embedded expressions is generated by a commercial OCR system. As the contents of embedded and displayed expressions are difficult to check by looking at the MathML presentation tags, an additional utility has been designed to generate a global description of a page from its truthed data that provides expressions in TeX format. For a page, there are two upper-level descriptions, one generated for the embedded expressions (using the associated `.emb` file) and the other one for displayed expressions (using the `.dis` file).

Using these description files (when viewed after compiling as a TeX document), it is more convenient to check the validity of the truthed data for an expression content. This is done by visually comparing the appearance of the expression in the image as well as in TeX format. Moreover, such an upper-level description provides ways to further check the `GC` value of an expression and

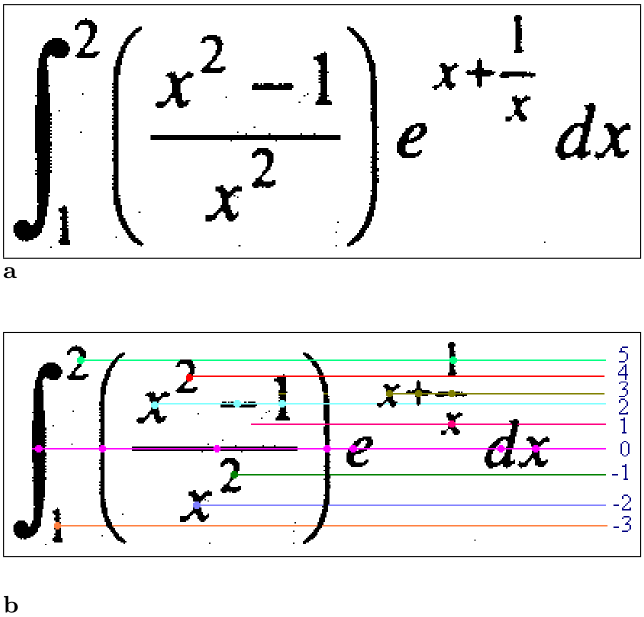


Fig. 4. Geometric complexity of expressions: an example

the `style`, `level`, and identity (`truth`) data for expression symbols. OCR accuracy for the text portion is also checked at this level.

3 Statistical investigation of the corpus

Initially, the data set is investigated to measure its comprehensiveness for expression recognition research. Next, some additional studies are presented for identification of embedded expressions and issues related to testing an expression recognition system.

3.1 Study of comprehensiveness

Representativeness of the proposed corpus is measured against some basic aspects that are important in any expression recognition research. The presence of different symbols, their occurrence frequencies, and the appearance of different two-dimensional (2D) structures (e.g., *superscript/subscript*, *root*, *fraction*, etc.) are studied to understand the corpus coverage. For each 2D structure, an idea of degree of nestedness (`DoN`) is introduced. For example, consider three structures (i) e^x, (ii) e^{x^x}, and (iii) $\sqrt{1+\frac{1}{x}}$. The first two *superscript* structures show values for `DoN` equal to `1` and `2`, respectively. On the other hand, the third structure is a mixed one and its `DoN` equals `2` since a *fraction* structure is nested inside the *root*. For a 2D structure, its `DoN` value is automatically computed from the operator tag pairs associated with that structure. For example, e^x uses only one pair of MathML presentation tags, namely, <msup> </msup>, making its `DoN` equal to 1, whereas $\sqrt{1+\frac{1}{x}}$ uses one pair of <mfrac> </mfrac> tags within <msqrt> </msqrt> tags, making `DoN` equal to 2.

Table 5. List of some frequently occurring operator symbols

No.	Symbol	Abundance (% of expressions in which the symbol appears)
1	=	94%
2	+	93%
3	– (Minus)	
4	/	
5	(	60%
6	)	
7	Fraction line	51%
8	[	35%
9	]	
10	{	20%
11	}	
12	<	16%
13	>	

The investigation results are outlined below.

Observation 1. *Expression symbols.* Symbols occurring in expressions are quite different from those occurring in normal text. Apart from the Roman letters, symbols like Arabic digits, Greek symbols, mathematical operators, function words, etc. are frequently used to write expressions. Analysis shows that the symbols present in the corpus can be partitioned into ten categories: (i) Arabic numerals (*AN*), (ii) Roman letters (*RL*), (iii) Greek symbols (*GS*), (iv) calligraphic letters (*CL*), (v) mathematical operators (*MO*), (vi) relational operators (*RO*), (vii) arrow symbols (*AS*), (viii) bracket symbols (*BS*), (ix) miscellaneous symbols (*MS*), e.g., prime("*′*"), for all ("∀"), there exists ("∃"), etc., (x) punctuation marks (*PM*).

The frequencies at which these symbol categories occur in embedded as well as displayed expressions are given in Table 4. The distinct number of symbol classes under each category is shown in braces in the first column. Table 4 gives an idea about the extra symbols an OCR has to classify for recognizing expressions. Availability of this large number of samples helps one to design suitable classification schemes for expression symbols. For this purpose, the training and test symbol sets are marked separately in the corpus.

Next, occurrence frequencies for symbols are computed. The list of symbol frequencies gives an idea about the relative abundance of different symbols in expressions. Based on these statistics, one may design a prototype library where reference symbols are arranged according to their occurrence frequencies to speed up the symbol recognition process. Moreover, occurrence statistics concerning mathematical operators help to identify a small group of symbols that show high occurrence frequencies. Table 5 presents a few of these operators and their abundance (not the occurrence frequencies) in the set of displayed expressions. The high occurrence and shape simplicity of these operators may help one to quickly identify zones containing displayed expressions.

Table 6. Occurrence of operator structures

Structure	# Occurrences	DoN values
Superscript	4,267	1, 2, 3, 4
Subscript	3,986	1, 2, 3
Fraction	2,063	1, 2, 3, 4, 6
Root	227	1, 2, 3, 5
Overline	60	1, 2, 3
Underline	13	1, 2
Overbrace	47	1, 2, 4
Underbrace	19	1, 3
Ellipses	828	1
Accent	341	1, 2, 3
Matrix	73	1, 2, 3
Stacking of symbols	154	1, 2

Observation 2. *Operator structures.* Symbols in expressions are arranged around different operators and form different geometric layouts, some of which are 1D in nature while others are 2D. For example, operators like "+", "−", and "=" and function words like *sin*, *cos*, *log*, etc. induce 1D structures, while superscripts, subscripts, operators (like "$\sum$," "$\prod$," "$\int$," etc.) with limit expressions, matrixes, etc. form the 2D layout.

Properties of such structures along with their occurrence statistics have been studied in the proposed corpus. Twelve elementary structures have been detected that are 2D in nature. They are: (i) *superscript*, (ii) *subscript*, (iii) *fraction*, (iv) *root*, (v) *overline*, (vi) *underline*, (vii) *overbrace*, (viii) *underbrace*, (ix) *ellipse*, (x) *accent*, (xi) *matrix*, and (xii) *stacking of symbols*.

The number of such structures observed in the data set are reported in Table 6. However, sometimes it is observed that structures appear in *nested* mode. A structure is called *nested* if it contains another structure within it. With each class of structures, the observed DoN (degree of nestedness) is listed in column 3 of Table 6.

Observation 3. *Structural complexity of expressions.* As discussed earlier, the geometric (or structural) complexity (GC) of an expression is defined by the number of horizontal lines on which the constituent symbols are arranged. Expressions in the database are investigated to check their complexity levels, and they are grouped into different classes based on their GC value. Frequencies of these classes in the database are reported in Table 7. Results show that embedded expressions are less complex in their structure than displayed expressions. The highest complexity observed in embedded expressions is 5, whereas for displayed expressions the value of GC can be as high as 15.

3.2 Linguistic properties of sentences containing embedded expressions

A linguistic analysis of the sentences present in scientific documents indicates that a word-level N-gram model could be of great help to categorize a sentence in a technical document into one of two categories, namely, with or without expressions. This analysis is motivated by Zipf's law [23], which is formally defined as $P_r \sim 1/r^a$, where P_r is the frequency of occurrence of the rth ranked item and a is close to 1.

Table 7. Geometric complexity of expressions

Geometric	Number of expressions	
complexity (GC)	Embedded	Displayed
1	695	347
2	1,498	489
3	481	628
4	374	427
5	53	109
6	–	202
7	–	93
8	–	70
9	–	31
10	–	16
11	–	23
12	–	12
13	–	7
14	–	3
15	–	2

In other words, the law says that the nth most common word in a human language text occurs with a frequency inversely proportional to n. The implication of this law is that there is always a set of words that are specific to a particular context. In our approach, the use of embedded expression is treated as the context and based on this, sentences with and without expressions are labeled with one of the two categories, namely, (i) category I (C_1) in which sentences do not contain any expression and (ii) category II (C_2) where each sentence contains one or more expression fragments. To generate N-gram profiles of a category, the following steps are performed.

- Step 1. Digits, punctuation marks, and expression portions (in the case of sentences containing expressions) are discarded so that sentences contain only words.
- Step 2. Each sentence is scanned to generate all distinct word-level N-grams (for $N = 1$ to 3) and their frequencies are recorded. To compute bigram and trigram frequencies, each sentence is padded with sufficient blanks on both sides. Stop words are discarded while generating the N-grams.
- Step 3. Relative frequency of each N-gram ($w_{1,N}$) is computed as the ratio of its associated count (i.e., the number of its occurrences) to the total number of similar (based on the value of N) N-grams. For example, let $c^i_{1,N}$ be the number of occurrences of the ith N-gram $w^i_{1,N}$ and T_N be the total number of distinct N-grams; then relative frequency $f^i_{1,N}$ of

$w^i_{1,N}$ is computed as

$$f^i_{1,N} = \frac{c^i_{1,N}}{\sum_{j=1}^{T_N} c^j_{1,N}} \text{ for } N = 1 \text{ to } 3\,. \qquad (1)$$

- Step 4. When steps 1–3 are completed, a list of all N-grams along with their frequencies is produced.
- Step 5. Sort the list in descending order of frequencies. The resulting list then represents an N-gram frequency for a category.

In our study, a profile for category I (C_1) is built using 2,655 sentences having nearly 35,000 words. On the other hand, 870 sentences containing about 12,000 words have been used to generate the profile for category II (C_2). A number of observations follow from an inspection of the N-gram profiles of the categories.

- Many of the stop words show high occurrence frequency in both category profiles. However, note that a list of stop words in this application is different from the standard stop word list used in text retrieval applications. This is so because several words like "if," "then," "where," etc. are important for the purposes of the present study, though in general they are treated as stop words. Therefore, the list of stop words mentioned in step 2 above is specially computed for the current application.
- The top 150 or so N-grams are highly representative of a particular category. Of course, there is nothing special about rank 150 itself, and it is chosen mostly by inspection. One could always carry out more elaborate statistics and choose an optimal cutoff rank.
- The top ranking N-grams in category I are mostly unigrams that reflect the distribution of the words in the sentences of that category, whereas for category II, the presence of frequent bigrams (e.g., *such that*, *the following*, etc.) and trigrams (e.g., *let us consider*, *is given by*, etc.) is also observed. In both categories, there is, of course, a long tail to the distribution of N-grams that goes well beyond 150. These N-grams represent terms that do not show much distinguishing power as far as classification is concerned.

A given sentence is classified by a similarity measure implemented as follows:

- Step 1. All N-grams are generated for a target sentence (S). Let L be the number of such N-grams.
- Step 2. Each N-gram found ($w^k_{1,N}$, for $k = 1$ to L) in S is searched in the category profile of C_1 and C_2. Let p_k and q_k be the relative frequency of $w^k_{1,N}$ in C_1 and C_2, respectively. If any $w^k_{1,N}$ is not found in the profile of C_1 (or in C_2), then p_k (or q_k) is set to zero.
- Step 3. S is classified as follows:

$$\begin{array}{rl} \text{If } \sum_{k=1}^{L} p_k > \sum_{k=1}^{L} q_k, & \text{then } S \in C_1 \\ \text{Elseif } \sum_{k=1}^{L} p_k < \sum_{k=1}^{L} q_k, & \text{then } S \in C_2 \\ \text{Otherwise,} & \text{Indeterminate} \end{array} \qquad (2)$$

This approach for sentence categorization has been tested with 877 sentences (which have not been used to generate category profile of either C_1 or C_2), and the classification results are reported in Table 8. These results show some interesting patterns:

- The classification method performs slightly better for longer sentences.
- The accuracy improves with the increased length (in terms of the number of N-grams) of the profiles. Overall, the method yields its best performance at a profile length of 150. At this stage, the system misclassified only 13 sentences out of 877, producing an overall classification accuracy of 98.52%.
- The result shows that the N-gram model provides important indications to label sentences containing embedded expressions. This feature, therefore, could be integrated with other aspects for robust identification of embedded expressions.

4 Issues related to system testing and performance evaluation

The database provides utilities to choose the right dataset to test an expression recognition system and to evaluate the test results in an automatic manner. The utilities are as follows:

4.1 Selection of test data

Choosing the right set of data is always an important aspect of testing any system performance. In the case of expression recognition, test data must represent the variability in terms of expression symbols, structures, geometric complexity, etc. Selecting an adequate number of test samples is another important criterion. The proposed corpus provides necessary support in this direction. Four types of queries can be made to choose the proper test samples. Queries are of the following nature:

- Type I: *Query for symbols*: A query can be made to search for documents and, in particular, embedded and displayed expressions that contain a particular query *symbol*. The word *symbol* refers to identifiers, constants, operators, etc. that occur in expressions. A query made on the document level is answered with a set of document IDs that contain the query symbol. Similarly, a set of expressions tagged with document IDs are returned when the query is made on the expression level. A query processor is implemented by maintaining two different forms of indexing – one at the individual-expression level and another at the document level. Symbol name and its relative occurrence frequency are used as index keys. The relative frequency of a symbol ($\mathcal{S}$) refers to an expression-level feature and is computed as

$$\frac{\text{No. of occurrences of } \mathcal{S} \text{ in an expression } \mathcal{E}}{\text{Total no. of symbols in } \mathcal{E}} \qquad (3)$$

A document-level indexing is just an aggregation of the results obtained for its constituent expressions.

Table 8. Correct classification of sentences using N-gram model

Sentence length (#Words)	< 10	< 10	< 10	< 10	≥ 10	≥ 10	≥ 10	≥ 10
Profile length (#N-grams)	50	100	125	150	50	100	125	150
Sentences without exps.	217/239	225/239	232/239	234/239	398/424	407/424	416/424	419/424
Sentences with exps.	60/68	63/68	65/68	66/68	139/146	142/146	144/146	145/146
Accuracy	90.2%	93.8%	96.7%	97.7%	94.2%	96.3%	98.3%	98.9%

The resultant expressions (or documents) are ranked (in descending order) by occurrence frequency of the query symbol in them.

- Type II: *Query for function words*: Expressions (or documents) containing a particular function word (e.g., sin, cos, log, max, etc.) can also be queried. This facility is the same as in type I explained above. Necessary index structures are maintained for this purpose.
- Type III: *Query for structures*: The presence of structures (e.g., *fraction*, *root*, *script*, *sum*, *product*, etc.) can also be queried. The query result for a particular structure returns the number of expressions (or documents if the query is made at the document level) or document IDs containing the expressions along with a value representing the degree of nestedness DoN for each occurrence of the structure. Through this query facility one may be assured that the test set contains samples for the structure one is looking for. The geometric complexity of the structures is also reflected in the associated DoN values.
- Type IV: *Query for geometric complexity*: Each expression is tagged with its geometric complexity (GC), as explained before. Therefore, one may pose a query to find expressions (and the documents containing them) having a particular complexity level. A support for such types of queries helps one to understand the geometric variability of expression structures. Moreover, based on the geometric complexity of expressions, one may choose the right data set to test one's expression recognition algorithm.

4.2 Performance evaluation of an expression recognition system

Several techniques have been proposed for evaluating graphics recognition systems [24], but they mostly consider images of engineering drawings containing lines, arcs, circles, etc. For expression recognition, no method has emerged as a standard. The quantitative evaluation of expression recognition results is, in fact, a difficult task since recognition involves two major stages: symbol recognition and structural analysis. The stages are tightly coupled, and therefore, if evaluation in one stage is done independently of the other, it may not reflect the true performance of the system. Errors in the symbol recognition stage affect the structure analysis results. This calls for an integrated evaluation mechanism for judging the performance of an expression recognition system.

To the best of our knowledge, three different techniques have so far been proposed to evaluate expression recognition results. Chan and Yeung [5] presented an integrated performance measure consisting of two independent measures: one for recognition of *symbol*s and another for recognition of *operator*s. These two measures are combined with equal weights. Here, the term *operator* includes 2D (e.g., script, limit, fraction, etc.) as well as 1D (+, −, etc.) structures. However, the actual performance is measured manually. Later on, Okamoto et. al. [6] have presented an automatic approach to evaluating their structure analysis method. They attempted to evaluate performance by checking whether each typical structure such as scripts, limits, fractions, roots, and matrices is recognized correctly. In their approach, expressions against which a system is evaluated are groundtruthed into MathML format. More recently, Zanibbi et. al. [7] presented another automatic way of evaluating performance. In that approach, an expression is visualized as a set of symbols appearing on different baselines. The performance is assessed by separately counting the number of (i) correctly recognized baselines and (ii) properly placed (w.r.t. the corresponding baseline) symbols.

As the methods proposed in [6,5] count only the number of properly recognized structures, errors in recognizing simple structures get the same weighting as errors that occur in the processing of complex nested structures. On the other hand, the technique proposed in [7] presents more in-depth analysis of the recognition results but does not provide any single figure of merit for overall performance evaluation.

The present corpus provides an automatic tool for testing system performance. It evaluates an expression recognition system by computing a performance index (γ). For a given expression, the proposed index considers the geometric (or structural) complexity of the expression to judiciously analyze the recognition result produced by a system. As explained earlier, the structural complexity of an expression is defined by GC, i.e., the number of horizontal lines on which constituent symbols are arranged. Moreover, we consider that an error in recognizing a base level (that is, dominant baseline [7]) structure would be more severe than an error in recognizing structures at higher levels. This is because symbols placed in horizontal lines other than the baseline are structurally dependent on the base-level symbols. Therefore, errors in placement of a base-level symbol (say, s_0) affect the placement of other symbols structurally related to s_0. In general, placement errors for base-level symbols affect the regeneration of an expression more than errors for other symbols. This situation will become more clear when the following examples are considered:

$$a+b+c=2+\alpha\,, \tag{4}$$

$$a^2+b+c=2+\alpha_1\,, \tag{5}$$

$$a^2+b^2+c^{2^{n_7}}=2+\alpha_1\,, \tag{6}$$

$$a^2+b^2+c^{2^{n_7}}=\frac{2+\alpha_1}{\beta^2}\,. \tag{7}$$

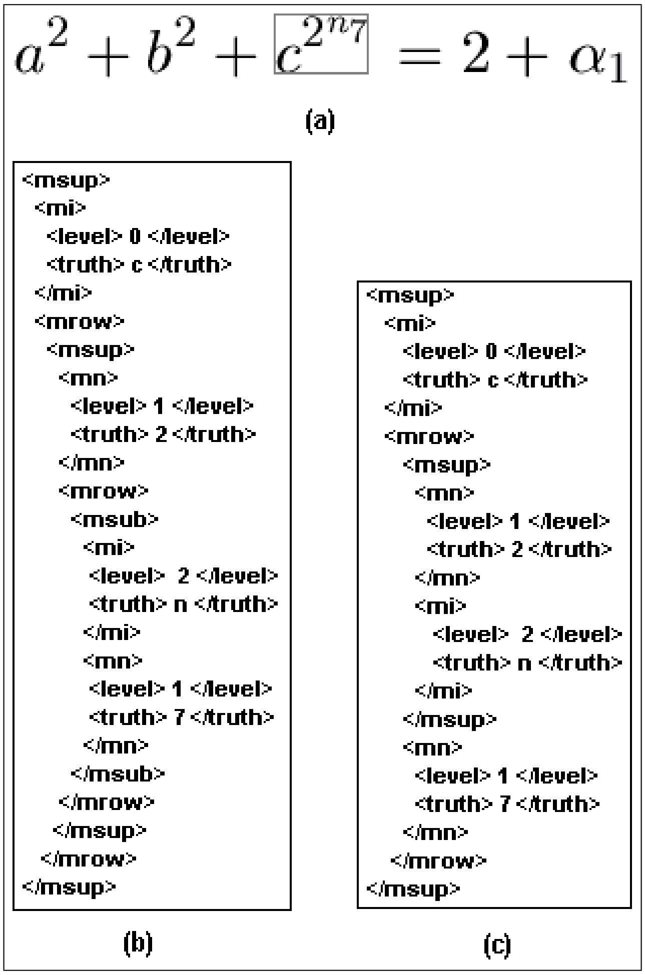
All the structures in Eq. 4 are in one level, and hence $\texttt{GC} = 1$. On the other hand, the expression in Eq. 5 shows a complexity `GC = 3` because a has a one-level superscript and α has a one-level subscript. Following this logic, the GC values of Eqs. 6 and 7 are 5 and 8, respectively. One can visualize that an error in interpreting the position of any base-level symbol (for example, "a" in Eq. 5 or "c" in eq. 6) would affect the layout of other immediate symbols nested in it. Moreover, because the use of non-base-level symbols (e.g., script or limit expressions) increases the structural complexity of an expression, any systematic evaluation strategy would be expected to consider how a system can recognize simple expressions and then to check the system's response as the complexity increases. Therefore, to evaluate the efficiency of an approach that deals with the recognition of expressions, one has to take the geometric complexity of the expressions into account. The four expressions in Eqs. 4–7 show a gradual increment in the structural complexity of the expressions.

Integrated performance measure: In our approach, an automatic performance evaluation can be done using the truthed data stored in the corpus. As explained earlier, the expressions in the dataset are groundtruthed using MathML presentation tags. The recognition result for an expression is compared with the groundtruth corresponding to that expression. If they do not match, then the result is not correct. Errors originate from two sources: (i) symbol recognition errors and (ii) errors in structure interpretation. Symbol recognition errors are easily computed as

$$\frac{\text{No. of wrongly recognized symbols}}{\text{Total no. of symbols}}\,. \tag{8}$$

However, the computation of structure recognition errors is not trivial. This is so because the parsing of an expression may not be fully correct, but some of its symbol arrangements may be interpreted properly, and the system should be given partial credit for it. In our method, the erroneous arrangement of a symbol (s) is penalized by a factor $\frac{1}{\mid i\mid+1}$, where i is the `level` of the symbol, s.

It may be noted that in the case of computing a structure recognition error, only spatial arrangements are important. For example, no structure recognition error is reported if X^m is recognized as X^{rn}. This is so because such symbol classification errors are accounted for by computing the symbol recognition accuracy. In the foregoing example, a structure recognition error is detected only if identification of the *superscript* structure fails.



```
<msup>
 <mi>
  <level> 0 </level>
  <truth> c </truth>
 </mi>
 <mrow>
  <msup>
   <mn>
    <level> 1 </level>
    <truth> 2 </truth>
   </mn>
   <mrow>
    <msub>
     <mi>
      <level>  2 </level>
      <truth> n </truth>
     </mi>
     <mn>
      <level> 1 </level>
      <truth> 7 </truth>
     </mn>
    </msub>
   </mrow>
  </msup>
 </mrow>
</msup>
```

```
<msup>
  <mi>
    <level> 0 </level>
    <truth> c </truth>
  </mi>
  <mrow>
    <msup>
      <mn>
        <level> 1 </level>
        <truth> 2 </truth>
      </mn>
      <mi>
        <level>  2 </level>
        <truth> n </truth>
      </mi>
    </msup>
    <mn>
      <level> 1 </level>
      <truth> 7 </truth>
    </mn>
  </mrow>
</msup>
```

Fig. 5a–c. Performance evaluation. **a** Image of an expression. **b** Groundtruthed data for the subexpression marked in **a**. **c** Recognition results obtained on recognition of the subexpression

For any test expression, let S_t be the total number of symbols, S_e be number of symbols recognized incorrectly, R_i be the number of symbols in the ith level, and O_i be the number of ith-level symbols for which incorrect arrangement analysis is encountered. Now, the performance index (γ) is defined as

$$\gamma = 1-\frac{S_e+\sum_i O_i\times\frac{1}{\mid i\mid+1}}{S_t+\sum_i R_i\times\frac{1}{\mid i\mid+1}}\,. \tag{9}$$

Assuming a test set ($\mathcal{T}_f$) contains Z expressions, γ_k is computed for all $k=1,2,\cdots,Z$, and to rate the overall system performance, an average γ_{avg} is computed as

$$\gamma_{avg}=\frac{1}{Z}\sum_k \gamma_i\,. \tag{10}$$

The performance of any expression recognition system can be judged following Eq. 9 and an average performance can be computed according to Eq. 10. Let us consider a small expression fragment to understand how evaluation is done using the groundtruth data. Figure 5b shows the groundtruth for a nested structure of the expression shown in Fig. 5a. The bounding box and style information is ignored for better convenience. However, one should consider bounding box information for evaluating segmentation results (e.g., whether symbols are properly extracted or not). Let us assume that the symbols are properly segmented in the present case.

Figure 5c shows the recognition output when a system incorrectly recognizes the subexpression $c^{2^{n_7}}$ in Fig. 5a as $c^{2^n 7}$, which encounters an incorrect arrangement of "7." Such incorrect arrangements can be automatically detected by comparing the recognition results against the groundtruth data. The comparison algorithm works iteratively; it starts by locating the innermost operator tags (like <msub>, <msup>, <mfrac>, etc.) and locates the symbols for which arrangements are incorrect. For example, absence of the innermost <msub> tag of Fig. 5b in Fig. 5c indicates an improper arrangement of "n" and "7." Next, comparison of the second <msup> tags in Fig. 5b and c pinpoints the incorrect arrangement of the symbol "7," whereas the symbols "2" and "n" imply the correct arrangement.

Next, to evaluate this recognition result, γ is computed following Eq. 9. In this case, $S_t = 4, S_e = 0, R_0 = 1, R_1 = 2, R_2 = 1, O_0 = 0, O_1 = 1, O_2 = 0$. Placement for only one symbol is incorrect at level 2. Therefore, $\gamma = 1 - \frac{1/2}{4+\sum 1+2/2+1/3} = 0.921$.

5 Conclusion

In this paper, the development of a corpus of printed scientific and technical documents is discussed. This study is amied at facilitating research on the automatic recognition of expressions. A considerable number of samples is gathered from various branches of science to make the data set a representative one. Groundtruthing of expression images follows a new format useful for several research considerations. A statistical analysis of the database is presented to show the comprehensiveness of the proposed corpus. Several utilities are provided to assist in designing and testing an expression recognition system. A new performance index is proposed to produce a single figure of merit after evaluation of such a system.

Immediate future work includes plans to make this corpus available on the Internet. This will take some time as final correction of the truth data remains incomplete. To provide guarantees concerning the truth data, we plan to check the generated data at two levels. Initially, the truthed data (T) is to be manually checked and corrected by two different groups in isolation to produce two versions of the same truthed data (T), namely, T_1 and T_2. Design of a user-friendly interface is in progress to assist this manual correction. Later on, T_1 and T_2 are to be compared by automatic file comparison utilities like `cmp`, `diff`, etc. Differences are to be marked and corrected to produce the final version of the truthed data.

Acknowledgements. The authors would like to thank Prof. Dorothea Blostein of the School of Computing, Queen's University, Kingston, Ontario, Canada, Dr. A. Ray Chaudhuri and Dr. Joydip Mitra of the Indian Statistical Institute, Kolkata, India for many useful discussions on this work. The authors would also like to thank the anonymous referees for their valuable comments and suggestions to improve the work presented in this manuscript. One of the authors (B. B. Chaudhuri) sincerely thanks Jawaharlal Nehru memorial Fund for partial support for this work. Finally, the authors express their gratitude to Pradip Dey, Udayan Kar, R.P. Ghosh, and Md. S. Chowdhury for their help regarding the manual intervention needed to generate the truthed data.

References

1. Fateman RJ, Tokuyasu T (1996) Progress in recognizing typeset mathematics. In: Proc. SPIE, San Jose, CA, 2660:37–50
2. Blostein D, Grbavec A (1997) Recognition of mathematical notation. In: Bunke H, Wang PSP (eds) Handbook of character recognition and document image analysis. World Scientific, Singapore, pp 557–582
3. Chan K-F, Yeung D-Y (2000) Mathematical expression recognition: a survey. Int J Doc Anal Recog 3(1):3–15
4. Phillips I (1998) Methodologies for using UW databases for OCR and image understanding systems. Proc. SPIE, Document Recognition V, 3305:112–127
5. Chan K-F, Yeung D-Y (2001) Error detection, error correction and performance evaluation in on-line mathematical expression recognition. Pattern Recog 34:1671–1684
6. Okamoto M, Imai H, Takagi K (2001) Performance evaluation of a robust method for mathematical expression recognition. In: Proc. 6th international conference on document analysis and recognition, Seattle, pp 121–128
7. Zanibbi R, Blostein D, Cordy JR (2002) Recognizing mathematical expressions using tree transformation. IEEE Trans Pattern Anal Mach Intell 24(11):1455–1467
8. Raman TV (1994) Audio system for technical readings. Doctoral dissertation, Cornell University, Ithaca, NY
9. Garain U, Chaudhuri BB (2001) On development and statistical analysis of a corpus for printed and handwritten mathematical expressions. In: Proc. 4th IAPR international workshop on graphics recognition (GREC2001), Canada, 2001, pp 429–439
10. Jain AK, Yu B (1998) Document representation and its application to page decomposition. IEEE Trans Pattern Anal Mach Intell 20(3):294–308
11. Pavlidis T, Zhou J (1992) Page segmentation and classification. Comput Vis Graph Image Process 54:484–496
12. Lee HJ, Wang J-S (1995) Design of a mathematical expression recognition system. In: Proc. 3rd international conference on document analysis and recognition, Montreal, pp 1084—1087
13. Toumit J-Y, Garcia-Salicetti S, Emptoz H (1999) A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents. In: Proc. 5th international conference on document analysis and recognition, Bangalore, India, pp 119–122
14. Fateman RJ (2000) How to find mathematics on a scanned page. In: Proc. Document Recognition and Retrieval VII, January 2000, San Jose, CA
15. Kacem A, Belaid A, Ben Ahmed M (2001) Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context. Int J Doc Anal Recog 4(2):97–108

16. Chowdhury SP, Mandal S, Das AK, Chanda B (2003) Automated segmentation of math-zones from document images. In: Proc. 7th international conference on document analysis and recognition, Edinburgh, UK, pp 755–759
17. Jin J, Han X, Wang Q (2003) Mathematical formulas extraction. In: Proc. 7th international conference on document analysis and recognition, Edinburgh, UK, pp 1138–1141
18. Suzuki M, Tamari F, Fukuda R, Uchida S, Kanahori T (2003) INFTY – An integrated OCR system for mathematical documents. In: Proc. ACM symposium on document engineering (DocEng), Grenoble, France, pp 95–104
19. Chaudhuri BB, Garain U (1998) Automatic detection of italic, bold and all-capital words in document. In: Proc. 14th international conference on pattern recognition (ICPR), Brisbane, Australia, pp 610–612
20. Chaudhuri BB, Garain U (2001) Extraction of type style based meta-information from imaged documents. Int J Doc Anal Recog 3(3):138–149
21. Chaudhuri BB, Garain U (2000) An approach for recognition and interpretation of mathematical expressions in printed documents. Pattern Anal Appl 3:120–131
22. Mitra J, Garain U, Chaudhuri BB, Swamy HVK, Pal T (2003) Automatic understanding of structures in printed mathematical expressions. In: Proc. 7th international conference on document analysis and recognition, Edinburgh, UK, pp 540–544
23. Zipf KG (1949) Human behavior and the principle of least effort: an introduction to human ecology. Addison-Wesley, Reading, MA
24. Phillips I, Chhabra A (1999) Empirical performance evaluation of graphics recognition systems. IEEE Trans Pattern Anal Mach Intell 21(9):849–870

Appendix A

Groundtruthing of the sentences containing embedded expressions for the image shown in Fig. 1.

```
<page>
<imagefile> doc004.tif </imagefile>
<sentence>
<line>
 <bbox> (67,12) (908,43) </bbox>
 <text>
  Let
 </text>
 <math>
  <bbox> (101,20) (137,42) </bbox>
  <GC> 3 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 1 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  and
 </text>
 <math>
  <bbox> (195,20) (224,43) </bbox>
  <GC> 3 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 2 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  denote the access times of
 </text>
 <math>
  <bbox> (526,15) (558,38) </bbox>
  <GC> 2 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
```

```
      <level> 0 </level>
      <style> i </style>
      <truth> M </truth>
     </mi>
    </mrow>
    <mrow>
     <mn>
      <level> -1 </level>
      <style> n </style>
      <truth> 1 </truth>
     </mn>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  and
 </text>
 <math>
  <bbox> (616,14) (650,38) </bbox>
  <GC> 2 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> M </truth>
     </mi>
    </mrow>
    <mrow>
     <mn>
      <level> -1 </level>
      <style> n </style>
      <truth> 2 </truth>
     </mn>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  , respectively, relative to
 </text>
</line>
<line>
 <bbox> (3,47) (106,68) </bbox>
 <text>
  the CPU.
 </text>
</line>
</sentence>
<sentence>
<line>
 <bbox> (115,45) (909,70) </bbox>
 <text>
  The average time
 </text>
 <math>
  <bbox> (313,50) (333,70) </bbox>
  <GC> 2 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <mn>
      <level> -1 </level>
      <style> i </style>
      <truth> A </truth>
     </mn>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  for the CPU to access a word
  in the memory system
 </text>
</line>
<line>
 <bbox> (4,78) (268,105) </bbox>
 <text>
  is given by the equation
 </text>
</line>
</sentence>
<sentence>
<line>
 <bbox> (318,265) (906,293) </bbox>
 <text>
  The time
 </text>
 <math>
  <bbox> (433,271) (454,291) </bbox>
  <GC> 2 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <mn>
      <level> -1 </level>
      <style> n </style>
      <truth> B </truth>
     </mn>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  required for the block transfer
  is called
 </text>
</line>
<line>
 <bbox> (4,298) (452,325) </bbox>
 <text>
  the block-replacement, or
  block-transfer, time.
 </text>
</line>
</sentence>
<sentence>
<line>
 <bbox> (557,296) (906,324) </bbox>
 <text>
  Hence we have
 </text>
 <math>
  <bbox> (744,296) (900,324) </bbox>
  <GC> 3 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 2 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
   <mo>
    <level> 0 </level>
    <style> n </style>
    <truth> = </truth>
   </mo>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <mi>
      <level> -1 </level>
      <style> i </style>
      <truth> B </truth>
     </mi>
    </mrow>
   </msub>
   <mo>
    <level> 0 </level>
    <style> n </style>
    <truth> + </truth>
   </mo>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 1 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  .
 </text>
</line>
</sentence>
<sentence>
<line>
 <bbox> (6,423) (908,452) </bbox>
 <text>
  Block transfer requires a relatively
  slow IO operation; therefore
 </text>
 <math>
  <bbox> (720,428) (741,447) </bbox>
  <GC> 2 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <mi>
      <level> -1 </level>
      <style> i </style>
      <truth> B </truth>
     </mi>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  is usually much
 </text>
</line>
<line>
 <bbox> (5,457) (182,483) </bbox>
 <text>
  greater then
 </text>
 <math>
  <bbox> (146,461) (173,483) </bbox>
  <GC> 3 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 1 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  .
 </text>
</line>
</sentence>
<sentence>
<line>
 <bbox> (193,457) (515,484) </bbox>
 <text>
  Hence
 </text>
 <math>
  <bbox> (270,457) (362,484) </bbox>
  <GC> 3 </GC>
  <mrow>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 2 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
  </mrow>
  <mo>
   <level> 0 </level>
   <style> n </style>
   <truth> \gg </truth>
  </mo>
  <msub>
   <mrow>
    <mi>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <msub>
     <mrow>
      <mi>
       <level> -1 </level>
       <style> i </style>
       <truth> A </truth>
      </mi>
     </mrow>
     <mrow>
      <mn>
       <level> -2 </level>
       <style> n </style>
       <truth> 1 </truth>
      </mn>
     </mrow>
    </msub>
   </mrow>
  </msub>
 </mrow>
</math>
 <text>
  and
 </text>
 <math>
  <bbox> (423,461) (510,483) </bbox>
  <GC> 3 </GC>
  <mrow>
```

```
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <msub>
      <mrow>
       <mi>
        <level> -1 </level>
        <style> i </style>
        <truth> A </truth>
       </mi>
      </mrow>
      <mrow>
       <mn>
        <level> -2 </level>
        <style> n </style>
        <truth> 2 </truth>
       </mn>
      </mrow>
     </msub>
    </mrow>
   </msub>
   <mo>
    <level> 0 </level>
    <style> n </style>
    <truth> \approx </truth>
   </mo>
   <msub>
    <mrow>
     <mi>
      <level> 0 </level>
      <style> i </style>
      <truth> t </truth>
     </mi>
    </mrow>
    <mrow>
     <mi>
      <level> -1 </level>
      <style> i </style>
      <truth> B </truth>
     </mi>
    </mrow>
   </msub>
  </mrow>
 </math>
 <text>
  .
 </text>
</line>
</sentence>
</page>
```

Appendix B

Groundtruthing of the displayed expressions for the image shown in Fig. 1.

```
<page>
<imagefile> doc004.tif </imagefile>
<math>
 <bbox> (331,126) (908,151) </bbox>
 <GC> 3 </GC>
 <expno> (5.3) </expno>
 <mrow>
  <msub>
   <mrow>
    <mi>
     <bbox> (331,130) (337,144) </bbox>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <mi>
     <bbox> (340,137) (349,147) </bbox>
     <level> -1 </level>
     <style> i </style>
     <truth> A </truth>
    </mi>
   </mrow>
  </msub>
  <mo>
   <bbox> (361,131) (379,138) </bbox>
   <level> 0 </level>
   <style> n </style>
   <truth> = </truth>
  </mo>
  <mi>
   <bbox> (391,126) (411,144) </bbox>
   <level> 0 </level>
   <style> i </style>
   <truth> H </truth>
  </mi>
  <msub>
   <mrow>
    <mi>
     <bbox> (412,130) (418,144) </bbox>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <msub>
     <mrow>
      <mi>
       <bbox> (421,137) (430,148) </bbox>
       <level> -1 </level>
       <style> i </style>
       <truth> A </truth>
      </mi>
     </mrow>
     <mrow>
      <mn>
       <bbox> (435,141) (438,150) </bbox>
       <level> -2 </level>
       <style> i </style>
       <truth> 1 </truth>
      </mn>
     </mrow>
  </msub>
 </mrow>
</msub>
<mo>
 <bbox> (450,126) (467,144) </bbox>
 <level> 0 </level>
 <style> n </style>
 <truth> + </truth>
</mo>
<mo>
 <bbox> (478,126) (485,150) </bbox>
 <level> 0 </level>
 <style> n </style>
 <truth> ( </truth>
</mo>
<mrow>
 <mn>
  <bbox> (490,126) (495,144) </bbox>
  <level> 0 </level>
  <style> n </style>
  <truth> 1 </truth>
 </mn>
 <mo>
  <bbox> (508,135) (525,136) </bbox>
  <level> 0 </level>
  <style> n </style>
  <truth> - </truth>
 </mo>
 <mi>
  <bbox> (538,126) (558,145) </bbox>
  <level> 0 </level>
  <style> i </style>
  <truth> H </truth>
 </mi>
</mrow>
<mo>
 <bbox> (560,126) (568,150) </bbox>
 <level> 0 </level>
 <style> n </style>
 <truth> ) </truth>
</mo>
<msub>
 <mrow>
  <mi>
   <bbox> (570,131) (576,145) </bbox>
   <level> 0 </level>
   <style> i </style>
   <truth> t </truth>
  </mi>
 </mrow>
 <mrow>
  <msub>
   <mrow>
    <mi>
     <bbox> (579,138) (588,148) </bbox>
     <level> -1 </level>
     <style> i </style>
     <truth> A </truth>
    </mi>
   </mrow>
   <mrow>
    <mn>
     <bbox> (591,141) (597,151) </bbox>
     <level> -2 </level>
```

```
       <style> i </style>
       <truth> 2 </truth>
      </mn>
     </mrow>
    </msub>
   </mrow>
  </msub>
 </mrow>
</math>
<math>
 <bbox> (346,377) (907,403) </bbox>
 <GC> 3 </GC>
 <expno> (5.4) </expno>
 <mrow>
  <msub>
   <mrow>
    <mi>
     <bbox> (346,383) (352,397) </bbox>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <mi>
     <bbox> (355,391) (364,401) </bbox>
     <level> -1 </level>
     <style> i </style>
     <truth> A </truth>
    </mi>
   </mrow>
  </msub>
  <mo>
   <bbox> (376,384) (393,391) </bbox>
   <level> 0 </level>
   <style> n </style>
   <truth> = </truth>
  </mo>
  <msub>
   <mrow>
    <mi>
     <bbox> (405,383) (411,397) </bbox>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <msub>
     <mrow>
      <mi>
       <bbox> (414,390) (423,400) </bbox>
       <level> -1 </level>
       <style> i </style>
       <truth> A </truth>
      </mi>
     </mrow>
     <mrow>
      <mn>
       <bbox> (428,393) (432,403) </bbox>
       <level> -2 </level>
       <style> i </style>
       <truth> 1 </truth>
      </mn>
     </mrow>
    </msub>
   </mrow>
  </msub>
  <mo>
   <bbox> (443,379) (460,396) </bbox>
   <level> 0 </level>
   <style> n </style>
   <truth> + </truth>
  </mo>
  <mo>
   <bbox> (471,378) (478,403) </bbox>
   <level> 0 </level>
   <style> n </style>
   <truth> ( </truth>
  </mo>
  <mrow>
   <mn>
    <bbox> (483,378) (488,396) </bbox>
    <level> 0 </level>
    <style> n </style>
    <truth> 1 </truth>
   </mn>
   <mo>
    <bbox> (501,386) (519,388) </bbox>
    <level> 0 </level>
    <style> n </style>
    <truth> - </truth>
   </mo>
   <mi>
    <bbox> (531,378) (551,396) </bbox>
    <level> 0 </level>
    <style> i </style>
    <truth> H </truth>
   </mi>
  </mrow>
  <mo>
   <bbox> (553,377) (561,403) </bbox>
   <level> 0 </level>
   <style> n </style>
   <truth> ) </truth>
  </mo>
  <msub>
   <mrow>
    <mi>
     <bbox> (563,382) (569,396) </bbox>
     <level> 0 </level>
     <style> i </style>
     <truth> t </truth>
    </mi>
   </mrow>
   <mrow>
    <mi>
     <bbox> (573,388) (582,400) </bbox>
     <level> -1 </level>
     <style> i </style>
     <truth> B </truth>
    </mi>
   </mrow>
  </msub>
 </mrow>
</math>
</page>
```

Utpal Garain received both his B.E. and M.E. in computer science and engineering from Jadavpur University, Kolkata in 1994 and 1997, respectively. Mr. Garain started his career as a software professional in industry and later on joined as a research personnel at the Indian Statistical Institute, Kolkata, where he is currently a full-time faculty member. He is one of the key scientists involved in the development of a bilingual (Devanagri & Bangla) OCR system, the first of its kind in India. Mr. Garain has published several technical papers in reputable international journals and conferences. His areas of interest are in digital document processing including optical character recognition for Indian language scripts, online character recognition, document data compression, etc.

B.B. Chaudhuri received his B.Tech. and M.Tech. from Calcutta University, India in 1972 and 1974, respectively, and his Ph.D. from the Indian Institute of Technology, Kanpur, in 1980. Currently he is the head of the Computer Vision and Pattern Recognition Unit of Indian Statistical Institute, India. His research interests include pattern recognition, image processing, computer vision, natural language processing, and digital document processing. Professor Chaudhuri has published more than 250 research papers in reputed journal/conferences and has authored three books. Professor Chaudhuri has received many awards and prizes including Sir J.C. Bose Memorial Award (1986), M.N. Saha Memorial Award (1989 and 1991), UGC Homi Bhabha Fellowship award (1992), Dr. Vikram Sarabhai Research Award (1995), and C. Achuta Menon Prize (1996), Homi Bhabha award (2003), and Jawaharlal Nehru Fellowship (2004) for his contribution in the field of Engineering sciences, Indian language processing, computer applications, etc. Professor Chaudhuri is a fellow of IEEE, IAPR, IETE (India), National Academy of Sciences, and Indian National Academy of Engineering (India). He currently serves as associate editor of Pattern Recognition, Pattern Recognition Letters, Int. J. Pattern Recognition and Artificial Intelligence (IJPRAI), Int. J. of Computer Vision, and VIVEK.