IMEGE: Image-based Mathematical Expression Global Error

Francisco Álvaro, Joan-Andreu Sánchez, José-Miguel Benedí

{falvaro, jandreu, jbenedi}@dsic.upv.es

Abstract

Mathematical expression recognition is an active research field that is related to document image analysis and typesetting. Several approaches have been proposed to tackle this problem, and automatic methods for performance evaluation are required. Mathematical expressions are usually represented as a coded string like LaTeX or MathML for evaluation purpose. This representation has ambiguity problems given that the same expression can be coded in several ways. For that reason, the proposed approaches in the past either manually analyzed recognition results or they reported partial errors as symbol error rate. In this study, we present a novel global performance evaluation measure for mathematical expression based on image matching. In this way, using an image representation solves the representation ambiguity as well as human beings do. The proposed evaluation method is a global error measure that also provides local information about the recognition result.

1 Introduction

Automatic recognition of Mathematical Expressions (ME) is an important problem for scientific document analysis and scientific document typesetting [3]. ME recognition techniques have been studied for both handwritten [21] and printed ME [5]. For recognition of handwritten ME, most of the works have concentrated on online recognition [6]. Online recognition of ME makes use of stroke information that is not present in offline recognition. Offline techniques [21, 16] must be considered for handwritten and printed ME recognition.

Online ME recognition is used to process ME that are given through tactile or pen-based interfaces. It makes possible the development of applications that can work with this kind of input, which have recently become very common. Recognition of printed ME can be used for the automatic transcription of scientific documents and for document information retrieval [22]. Handwritten documents can also be transcribed using offline ME recognition techniques [21].

ME recognition comprises mainly two problems, that is, the recognition of mathematical symbols of the ME, and the recognition of the structural relation between these mathematical symbols [21, 18]. As a pattern recognition problem, a fundamental issue in ME recognition is the definition of automatic evaluation techniques. Since the recognition of mathematical symbols can be stated as a classical classification problem, the classification error rate of individual symbols is usually provided as a performance measure [12, 1]. However the recognition of the structural relation between mathematical symbols, which can be seen a parsing problem, requires more sophisticated evaluation methods [10].

A range of different scenarios can be considered for the structural evaluation of ME recognition techniques. On one end of the range, the full ground-truth structural information is explicitly available [15, 18]. On the other end, the ground-truth structural information is not available, and only confidence measures or the input ME can be considered for evaluation.

When the ground-truth structural information is fully available, a representation (for example LaTeX format or MathML format) that allows automatic evaluation is needed. Evaluation techniques are usually based on tree-matching [17], but these techniques could report non-existent recognition errors due to the representation ambiguity of the coded ME as we describe later.

When no ground-truth information is available, the evaluation performance must rely on confidence measures computed as posterior probabilities that are dependent on the model [8], or the recognition output can be compared with the input data through some process.

In this paper, we present an automatic performance evaluation measure of ME recognition systems when the ground-truth information is available as a coded string in LaTeX. Given a recognition result and its ground-truth, this approach does not compare the structure of the coded representations directly. From each ME string representation, we generated the image that it described, and then we compared both images. This way we avoid the ambiguity representation problem by comparing ME as human beings do, but the comparison between the images should be tackled in order to obtain a normalized error value.

The remainder of the paper is organized as follows. In Section 2, we review some proposals for the evaluation of ME. Section 3 describes the proposed measure, and conclusions are presented in Section 4.

2 Evaluation of ME recognition systems

The automatic evaluation of a ME is not an easy task and this fact has made the definition of widely accepted evaluation measures difficult. Several research studies have introduced different recognition techniques for ME and most of the times each study has used a different method for evaluation [10]. Since it is difficult to compare different techniques and even to evaluate the goodness of the performance measure, an automatic and objective performance evaluation metric would be of major interest.

In the past, several metrics have been proposed to report performance of mathematical expression recognition systems. The authors usually used a set of different measures to present the results of a certain experiment. There are metrics such as symbol recognition rate [2, 19], operator recognition rate [4], structure recognition rate [13], or baseline recognition rate [21] that can be computed if the ground-truth is available. However, these values only take into account the evaluation of a specific part of the ME recognition problem. Another measure that is often used is the expression recognition rate [4, 13, 21]. However, it does not provide any information about errors; it only determines whether or not an expression is perfectly recognized, and sometimes it is manually calculated.

Given that the previous methods only report partial errors, several global measures have also been presented. Chan and Yeung [4] proposed an integrated performance measure, which was a simple combination of symbol recognition and operator recognition rates. Garain and Chaudhuri [7] presented a global performance index that combined symbol and structural errors according to the complexity of the ME. Recently, Sain *et al.* [17] have presented EMERS, a tree matching-based performance evaluation measure.

If the ME output of a recognition system is represented as a tree structure, then it is possible to define a set of edit operations and to compare two expressions by computing an edit distance between these trees. The EMERS metric defines a performance evaluation of ME recognition systems using this idea. This method is based on the ME representation as a MathML string. Since this format is an application of XML, it explicitly describes both the structure and content of the expression. Hence, with this information, the difference between two mathematical expressions can be computed as the edit distance between their trees.

Given two trees A and B, the EMERS metric computes the edit distance between them by using three operations: insertion, deletion, and substitution. Moreover, each operation has a cost function that decreases the deeper the involved node is in the expression tree. The time complexity of the algorithm is $O(|A|^2|B|^2)$ or $O(n^4)$ because A and B should be of the same size. It should be noted that the EMERS metric is not a normalized distance. This method calculates the set of edit operations to transform A into B in order to obtain a minimum total cost, which is computed as the sum of the cost of these operations. If both expressions are identical, the EMERS value should be equal to zero.

The main problem with these metrics is the representation ambiguity of the ME ground-truth. Given a ME, it is usually coded as a string in LaTeX or MathML, and then the automatic performance evaluation is done using this representation. However, the same ME can be represented in several correct ways using these codifications. Therefore, an automatic performance evaluation measure that can tackle the representation ambiguity problem is required.

3 Image-based ME global error

Given a recognition result of a certain expression (usually as a coded string like LaTeX or MathML), we wanted to evaluate the performance of this result. Since there can be several string representations of the same ME, and the image obtained should be unique, we propose comparing the images directly instead of their string representation.

As an image can be generated from a ME given as a coded string, the idea was to compute a matching between the recognized expression image (test image) and the ground-truth label (reference image). Once we had an image-based model to perform that task, we proposed a novel method to obtain an error value as a result of comparing two images. With this method, we were able to compute a global error value of a ME recognition avoiding the representation ambiguity problem.

In the following subsections we explain how by using an image-matching model (3.1), we can define the evaluation algorithm (3.2) that is used to finally compute the recognition error (3.3).

3.1 Image-matching model (IDM)

In order to obtain a matching between two images, the initial idea was to compute a 2-dimensional warping between them. Levin and Pieraccini [11] extended the 1-dimensional dynamic time warping algorithm to two dimensions and noted that it had exponential complexity. Keysers *et al.* [9] presented several deformation models that were less constrained and, consequently, their complexity was lower. These models were introduced for image classification, and the Image Distortion Model (IDM) represented the best compromise between computational complexity and evaluation accuracy. For this reason, we chose the IDM to perform a matching between two images.

The IDM is a zero-order model of image variability [9]. This model uses a mapping function with absolute constraints; hence, it is computationally much simpler than a 2-dimensional warping. Its lack of constraints is compensated using a local gradient image context window. This model obtains a dissimilitude measure from one image to another such that if two images are identical, their distance is equal to zero.

The IDM has two parameters: warp range (w) and context window size (c). The algorithm requires each pixel in the test image to be mapped to a pixel within the reference image not more than w pixels from the place it would take in a linear matching. Over all these possible mappings, the best matching pixel is determined using the $c \times c$ local gradient context window by minimizing the difference with the test image pixel. Fig. 1 illustrates how the IDM works and the contribution of both parameters, where the warp range w constrains the set of possible mappings and the $c \times c$ context window computes the difference between the horizontal and vertical derivatives for each mapping. It should be noted that these parameters need to be tuned.

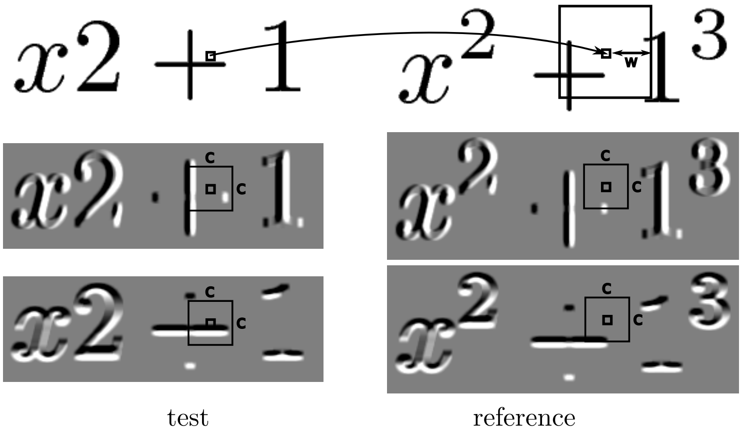


Figure 1: Image Distortion Model (IDM) visual representation.

3.2 The evaluation algorithm (BIDM)

Once we had a model that was able to detect similar regions of two images, we wanted to use this information to compute an error measure between them. Starting from the IDM-distance algorithm presented in [9], we proposed the Binary IDM (BIDM) evaluation algorithm shown in Fig. 2. First, instead of calculating the vertical and horizontal derivatives using Sobel filters, these derivatives are computed using the method described in [20] because, in this work, we can dealt with binary images. Next, the double loop computes the IDM distance for each pixel, where these values are stored individually. After that, the difference between each pixel of the test image and the most similar pixel found in the reference image can be represented as a gray-scale image (Fig. 3a). At this point, we have a dissimilitude value for each pixel of the test image. However, rather than knowing how different a pixels is, we want to know whether or not a pixel is correct. This is achieved by normalizing the distance values in the range $[0, 255]$ and then performing a binarization process using Otsu's method [14] (Fig. 3b). Finally, we intersect the foreground pixels of the test image with the binarized mapping values (like an error mask), and, as a result, we know which pixels are properly recognized and which are incorrectly recognized (Fig. 3c). Therefore, since the background pixels do not provide information, the number of correct pixels is normalized by the foreground pixels.

The time complexity of the algorithm is $O(IJw^2c^2)$, where $I \times J$ are the test image dimensions, w is the warp range parameter, and c is the local gradient context window size. It is important to note that in practice both w and c take low values compared to the image sizes.

3.3 Recognition error (IMEGE)

The BIDM algorithm computes the number of pixels of the test image that are correctly allocated in the reference image according to the IDM model. However, it should be noted that this process is from one image to another one. It is possible for a recognition result to omit a symbol of the reference; however in this case, the rest of the pixels of the test image could be correct and then no error would be reported. For this reason, the algorithm that we use follows the concepts of precision and recall to compute the Image-based Mathematical Expression Global Error (IMEGE). First, we compute the BIDM value from the test image to the reference, and the result obtained represents the number of pixels that are properly recognized from all the pixels that are

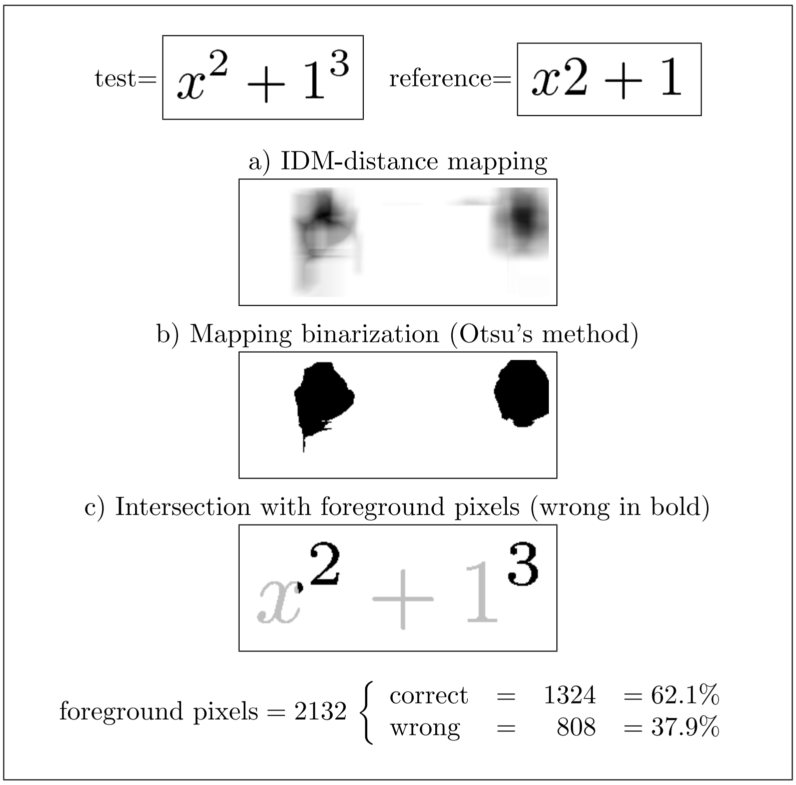
```
Input: test image $A$ $(I \times J)$, warp range $w$
reference image $B$ $(X \times Y)$, context window size $c$
Output: BIDM$(w, c)$ from $A$ to $B$

  $A^v$ = vertical_der($A$); $A^h$ = horizontal_der($A$)
  $B^v$ = vertical_der($B$); $B^h$ = horizontal_der($B$)
  for $i = 1$ to $I$ do {
      for $j = 1$ to $J$ do {
          $i' = \left\lfloor i\frac{X}{I} \right\rfloor$ ,  $j' = \left\lfloor j\frac{Y}{J} \right\rfloor$ ,  $z = \left\lfloor \frac{c}{2} \right\rfloor$
          $S_1 = \{1, \ldots, X\} \cap \{i' - w, \ldots, i' + w\}$
          $S_2 = \{1, \ldots, Y\} \cap \{j' - w, \ldots, j' + w\}$
          $s = \min_{\substack{x \in S_1 \\ y \in S_2}} \sum_{m=-z}^{z} \sum_{n=-z}^{z} (A^v_{i+n,j+m} - B^v_{x+n,y+m})^2 + (A^h_{i+n,j+m} - B^h_{x+n,y+m})^2$
          $map(i, j) = s$
      }
  }
  normalize_depth(map, 255)
  binarize(map)  //Otsu's method

  fg = $\{(x, y) \mid A(x, y) < 255\}$     //Foreground pixels
  cp = fg $\cap \{(x, y) \mid map(x, y) = 0\}$  //Correct pixels

  return $\frac{|cp|}{|fg|}$    //Correct pixels ratio
```

Figure 2: Binary IDM (BIDM) evaluation algorithm.



a) IDM-distance mapping

b) Mapping binarization (Otsu's method)

c) Intersection with foreground pixels (wrong in bold)

$$\text{foreground pixels} = 2132 \begin{cases} \text{correct} & = & 1324 & = 62.1\% \\ \text{wrong} & = & 808 & = 37.9\% \end{cases}$$

Figure 3: Example of the BIDM algorithm process given two mathematical expression images.

proposed as a solution for the recognition problem. This represents the precision (p). Second, we compute the same value from the reference image to the test image, and the value obtained represents the number of pixels that are properly allocated in the test image, which represents the recall (r). Finally, both values are combined using the harmonic mean $f_1 = 2(p \cdot r)/(p + r)$, and we obtain the final error value. Fig. 4 illustrates an example of this process.

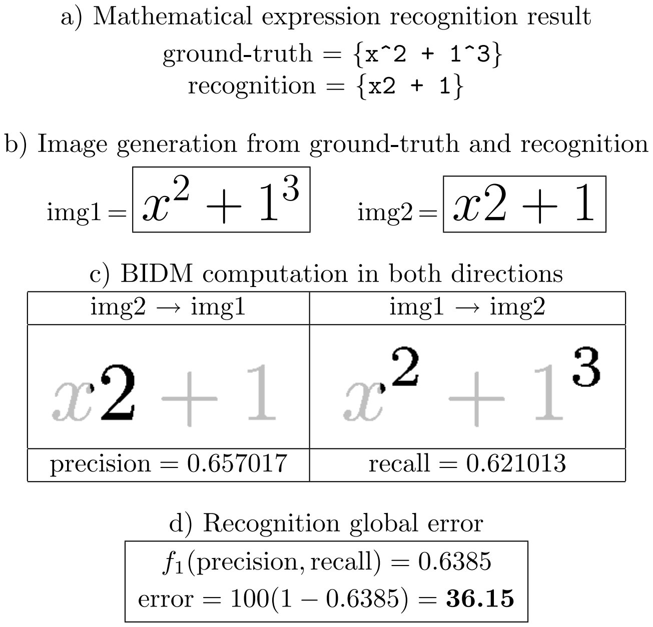

a) Mathematical expression recognition result
ground-truth = {x^2 + 1^3}
recognition = {x2 + 1}
b) Image generation from ground-truth and recognition
img1 =
img2 =
c) BIDM computation in both directions
img2 → img1
img1 → img2
precision = 0.657017
recall = 0.621013
d) Recognition global error
f_1(precision, recall) = 0.6385
error = 100(1 − 0.6385) = **36.15**

Figure 4: Example of procedure for computing the IMEGE measure given a ME recognition and its ground-truth.

4 Conclusions

In this work, we have presented IMEGE, which is a novel performance evaluation measure of ME based on image matching. It solves the representation ambiguity problem in a natural way, using the image generation to compare ME as human beings do. We proposed the BIDM algorithm which matches correct recognized regions and detects misrecognized zones. Both symbol and structural errors are detected and the algorithm also provides local information about which zones are misrecognized. The IMEGE metric provides a normalized symmetric global error value given a ME recognition and its ground-truth. Since it only requires being able to generate the image representation of the coded information, it can be used in any ME recognition case (online/offline or printed/handwritten).

Acknowledgements

Work supported by the EC (FEDER/ FSE) and the Spanish MEC/MICINN under the MIPRCV “Consolider Ingenio 2010” program (CSD2007-00018), the MITTRAL (TIN2009-14633-C03-01) project, the FPU grant (AP2009-4363), and by the Generalitat Valenciana under the grant Prometeo/2009/014.

References

[1] F. Álvaro and J.A. Sánchez. Comparing several techniques for offline recognition of printed mathematical symbols. *International Conference on Pattern Recognition*, 0:1953–1956, 2010.

[2] K. Ashida, M. Okamoto, H. Imai, and T. Nakatsuka. Performance evaluation of a mathematical formula recognition system with a large scale of printed formula images. In *Proceedings of the Second International Conference on Document Image Analysis for Libraries*, pages 320–331, Washington, DC, USA, 2006. IEEE Computer Society.

[3] K.F. Chan and D.Y. Yeung. Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3:3–15, 2000.

[4] K.F. Chan and D.Y. Yeung. Error detection, error correction and performance evaluation in on-line mathematical expression recognition. *Pattern Recognition*, 34(8):1671 – 1684, 2001.

[5] P. A. Chou. Recognition of equations using a two-dimensional stochastic context-free grammar. In W. A. Pearlman, editor, *Visual Communications and Image Processing IV*, volume 1199 of *SPIE Proceedings Series*, pages 852–863, 1989.

[6] U. Garain and B.B. Chaudhuri. Recognition of online handwritten mathematical expressions. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 34(6):2366–2376, 2004.

[7] U. Garain and B.B. Chaudhuri. A corpus for OCR research on mathematical expressions. *International Journal on Document Analysis and Recognition*, 7:241–259, 2005.

[8] O. Golubitsky and S.M. Watt. Confidence measures in recognizing handwritten mathematical symbols. In *Proceedings of the 16th Symposium, 8th International Conference. Held as Part of CICM '09 on Intelligent Computer Mathematics*, pages 460–466, 2009.

[9] D. Keysers, T. Deselaers, C. Gollan, and H. Ney. Deformation models for image recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(8):1422–1435, 2007.

[10] A. Lapointe and D. Blostein. Issues in performance evaluation: A case study of math recognition. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 1355–1359, Washington, DC, USA, 2009. IEEE Computer Society.

[11] E. Levin and R. Pieraccini. Dynamic planar warping for optical character recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 3:149–152, 1992.

[12] C. Malon, S. Uchida, and M. Suzuki. Mathematical symbol recognition with support vector machines. *Pattern Recognition Letters*, 29(9):1326–1332, 2008.

[13] M. Okamoto, H. Imai, and K. Takagi. Performance evaluation of a robust method for mathematical expression recognition. In *Proc. 6th International Conference on Document Analysis and Recognition (ICDAR'01)*, pages 121–128, September 2001.

[14] N. Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[15] I. Phillips. Methodologies for using UW databases for OCR and image understanding systems. In *Proceedings of SPIE, Document Recognition V*, volume 3305, pages 112–127, 1998.

[16] D. Průša and V. Hlaváč. Mathematical formulae recognition using 2d grammars. *International Conference on Document Analysis and Recognition*, 2:849–853, 2007.

[17] K. Sain, A. Dasgupta, and U. Garain. EMERS: a tree matching-based performance evaluation of mathematical expression recognition system. *International Journal of Document Analysis and Recognition*, 2010.

[18] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori. Infty- an integrated OCR system for mathematical documents. In C. Vanoirbeek, C. Roisin, and E.Munson, editors, *Proc. of ACM Symposium on Document Engineering*, pages 95–104, Grenoble, 2003.

[19] Y. Takiguchi, M. Okada, and Y. Miyake. A fundamental study of output translation from layout recognition and semantic understanding system for mathematical formulae. *Document Analysis and Recognition, International Conference on*, 0:745–749, 2005.

[20] A.H. Toselli, A. Juan, and E. Vidal. Spontaneous Handwriting Recognition and Classification. In *Proc. of the 17th International Conference on Pattern Recognition*, pages 433–436, Cambridge, UK, August 2004.

[21] R. Zanibbi, D. Blostein, and J.R. Cordy. Recognizing mathematical expressions using tree transformation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(11):1–13, 2002.

[22] J. Zhao, M.Y. Kan, and Y.L. Theng. Math information retrieval: user requirements and prototype implementation. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 187–196, New York, NY, USA, 2008. ACM.