Math Spotting in Technical Documents Using Handwritten Queries

Li Yu and Richard Zanibbi
Deptpart of Computer Science
Rochester Institute of Technology

March 28, 2009

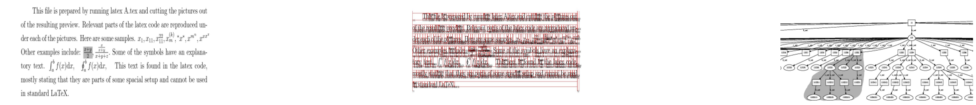
Content-based image retrieval (CBIR) for documents has been studied for a long time [3]. It focuses on indexing and retrieval capabilities in a large set of document images and provides a convenient way for people to access the information in these images, which means retrieving document images by their visual features (color,shape,texture,etc) so converting them to electronic formats can be avoided. As a sub-problem, word spotting means spotting a specific word or a set of words in a large document images database. In our research, we are concentrating on spotting mathematical symbols instead of words so its called math spotting. A new method will be proposed based on several current technologies and applied for the spotting problem.

Over the years, different features have been extracted from images and various similarity measures have been tried. A hierarchical representation of the page layout called X Y tree has been usually used to retrieve query image by comparing tree structure [1]. Based on pixel projection, the X-Y tree can be gotten by X-Y cutting (Figure 1b), which cuts the page in vertical and horizontal directions alternatively. Figure 1 shows the process of getting a X-Y tree of a document segment. As shown in Figure 1b, X-Y tree cutting divides the page into a lot of red component boxes, each of which corresponds to one node in X-Y tree (Figure 1c). At the same time, the minimum component boxes that don't contain any other component boxes represent the leaves in the tree. Fro example, Figure 2a shows a simple query which has been first drawn by pen in blank papers then scanned, the X-Y cutting and X-Y tree for it are shown in 2b and 2c respectively. The minimum component boxes that contain "x","+","y","-","2" response to the five leaves in the X-Y tree exactly. Considering that different symbols in the documents may have different X-Y tree representations, [2] sheds light on a possible way to retrieve math symbols in a document images database, although it hasn't been tried on spotting problem yet. After getting X-Y tree, the problem of spotting math notations reduces to the problem of subtree matching. Figure 2c shows a X-Y tree for the query image, and if we can find a sub-tree in Figure 1c which is the same as or similar to the query X-Y tree, it may provide an effective evidence that the query are included in the page. Many different sub-tree isomorphism algorithm have been studied recently. Considering the running time requirement and the fact that the symbols are located in the leaves of X-Y tree, a button-up algorithm [5] with liner time in the size of the trees is preferred. Three steps are included:

- Given two X-Y tree named T1 and T2, get a compacted directed acyclic graph representation G of the for- est F consisting of the disjoint union of T1 and T2. A linear time algorithm which scans each node only once is to be implemented a to find the largest common forest between T1 and T2.
- Building up a mapping M from T1 to T2 based on result from last step.
- Evaluate the cost of mapping (edit distance [4]) based on the visual feature of each leaf in the tree.

Figure 3a shows a bottom-up mapping from Ta to Tb. The edit distance from Ta to Tb equals to the cost of mapping between them. In other words, the edit distance between Ta and Tb is the sum of three parts: 1,cost of deleting a_1,a_2,a_3,a_4 in Ta. 2,cost of adding b_1,b_4 in Tb. 3,cost of converting (a_5 to b_2),(a_6 to b_3)...,(a_{10} to b_8) respectively. Typically, the edit distance between two isomorphic trees (figure 3c) is zero, indicating there is no cost for converting (a_i to b_j) in an exact matching.

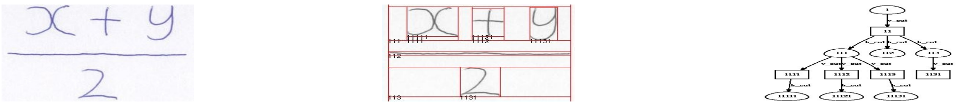
However, in some cases, only X-Y tree representation cannot provide accurate spotting results since other contents other than math notations may have same or similar X-Y tree when compared with query. In order to solve this problem, some visual features from the page may be helpful. Our initial idea is to use the ratio of black pixel region to component box area to identify the irrelevent notations with same layout



This file is prepared by running latex A.tex and cutting the pictures out of the resulting preview. Relevant parts of the latex code are reproduced under each of the pictures. Here are some samples. $x_1, x_{11}, x_{11}^{22}, x_m^{(k)}, {}^*x^*, x^{m^n}, x^{x^{x^x}}$ Other examples include: $\frac{x+y}{2}$, $\frac{\frac{x}{x+y}}{x+y+z}$. Some of the symbols have an explanatory text. $\int_a^b f(x)\mathrm{d}x$, $\oint_a^b f(x)\mathrm{d}x$, This text is found in the latex code, mostly stating that they are parts of some spacial setup and cannot be used in standard LaTeX.

(a) a document segment (b) X-Y tree cutting (c) X-Y tree

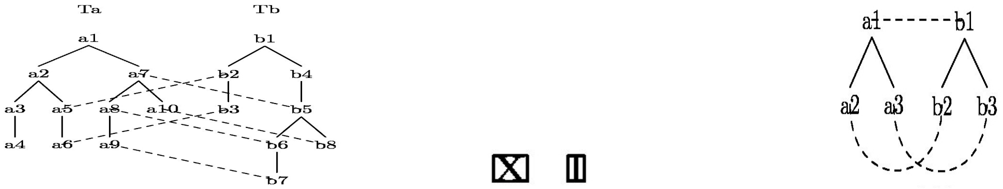
Figure 1: page in image database



(a) a query image (b) X-Y tree cutting (c) X-Y tree

Figure 2: query image

structure as query, which leads to a weighted mapping cost. For example, figure 3b shows two letters X and I in their component boxes. Instinctively, I gets a large ratio to its component box. And then if we look at the exact matching shown in figure 3c again, the cost of mapping may not be zero because the cost of converting (a_i to b_j) will be based on the visual feature mentioned above. Alternatively, such visual feature may only be used in a higher level in the X-Y tree. For example, only the roots of two trees will be computed to avoid excessive time costing. Other kinds of visual features may have more accurate result but higher computation burden.



(a) a mapping form Ta to Tb (b) component boxes (c) an exact mapping

Figure 3: mapping and distance

Math spotting is essentially an information retrieval problem. Therefore, evaluation metrics for our system have been adopted from information retrieval research. Two of most popular evaluation measures are described as: T_p(Precision) $= T_1/T_2$ and T_r(Recall) $= T_1/T_3$, where T_1 is the number of correct ones in the results and T_2 is the total number of results returned by the system, T_3 equals to the total number of correct pages stored in database. T_p only relates to the results returned by the spotting system while T_r associates the results with the ones in the database. Usually, there exists an inverse relationship between the two metrics. For example, if we increase the number of spotting, Tr can often increase since more correct pages in the database be retrieved, decreasing Tp due to increasing number of incorrect returned pages.

References

[1] S. Marinai, E. Marino, and G. Soda. Layout based document image retrieval by means of xy tree reduction. 1:432–436, 2005.

[2] Nasayuki Okamoto and Bin Miao. Recognition of mathematical expressions by using the layout structures of symbols. 1:242–250, 1991.

[3] Stevan Rudinac, Goran Zajic, Maja Rudinac, and Branimir Reljin. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.

[4] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM*, Volume 26(422-433), 1979.

[5] G. Valiente. An efficient bottom-up distance between trees. *Eighth International Symposium on String Processing and Information Retrieval*, (212-219), 2001.