SYMBOL-LEVEL GROUNDTRUTHING ENVIRONMENT FOR OMR

Michael Droettboom
Digital Knowledge Center, Sheridan Libraries
The Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218, USA

Ichiro Fujinaga
CIRMMT, Faculty of Music
McGill University
555 Sherbrooke Street West
Montréal, QC H3A 1E3, Canada

ABSTRACT

A simple framework for evaluating OMR at the symbol level is presented. While a true evaluation of an OMR system requires a high-level analysis, the automation of which is a largely unsolved problem, many high-level errors are correlated to these more tractably-analyzed lower-level errors.

1. INTRODUCTION

Complex document understanding systems are difficult to evaluate [5]. On the other hand, pixel-level and symbol-level analysis is much more feasible to implement and still quite useful for improving the accuracy of document understanding systems [11].

We demonstrate that, at least at the level of symbol recognition, the test driven software development technique [1] can be used to develop robust optical music recognition (OMR) systems. We argue that developing a performance evaluation metric for a pattern recognition software system is similar to the test-driven software development methodology. The goal in both cases is to minimize errors.

In many classification problems the evaluation metric is fairly straightforward. For example at the character level of OCR, it is simply a matter of finding the ratio between correctly identified characters and the total number of characters. In other classification domains, this is not so simple, for example document segmentation [7], recognition of maps [4], mathematical equations [9], graphical drawings, and music scores. In these domains, there are often multiple correct output representations, which makes the problem of comparing a given output to high-level groundtruth very difficult [5]. In fact, it could be argued that a complete and robust system to evaluate OMR output would be almost as complex and error-prone as an OMR system itself.

Symbol-level analysis may not be directly suitable for comparing commercial OMR products [2, 8], because such systems are usually "black boxes" that take an image as input and produce a score-level representation as output. Nevertheless, higher-level evaluations such as [8] are often correlated to lower-level errors, thus studying the effects of lower-level errors to higher-level errors with open-source systems should provide insights for evaluating commercial products.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2004 Universitat Pompeu Fabra.

2. EVALUATIONS

Testing is useful for developing new approaches to recognition, comparing the relative performances (error rate, efficiency) of alternative approaches and validating incremental modifications to algorithms (e.g. regression testing). In general, there are three types of evaluation strategies [3]: 1) symbol-level evaluation, 2) combination of symbol-level evaluations, and 3) an edit cost function. [1]

The symbol-level evaluation involves calculating the recognition rate of basic graphic symbols, which may be line segments, arcs, or characters. Sophisticated metrics will refine the recognition rate by counting separately the correctly identified symbols, missed symbols, and false positive (added) symbols.

In many document-understanding systems, including OMR, metrics are required at higher-levels. This is often calculated as some linear combination of weighted lower-level errors. The assignments of weights are non-trivial, however [10]. Ng, et al., 2003 [8] provides a first attempt at developing evaluation weights for OMR. However, as music is perceptual, some errors are "more important than others" and can rely on levels of complex context. Therefore any fully complete evaluation metric must rely, at least on some level, on human studies.

Another popular metric for the evaluation of complex document understanding systems is to calculate the cost of manually correcting and redrawing graphical objects. This is achieved by calculating the editing time required by empirically finding the times for operations such as mouse clicks [12]. Once the relationship between the symbol-level errors and the type of operations required to correct them is established, this metric can be calculated automatically.

[1] The term "low-level" used in [3] has been changed to "symbol-level" for better clarity.

To sum up, in all cases, symbol-level error analysis is required.

3. SYMBOL-LEVEL ANALYSIS

The goal of our symbol-level analysis system is to evaluate the performance of our OMR system, GAMUT (Gamera-based Adaptive Music Understanding Tool) [2] on very small score features, before the symbolic representation of the score has been assembled. Some of these are:

- assignment of accidentals to notes
- combination of clefs and accidentals into key signatures
- construction of time signatures from individual digits
- number of beams in each beam complex
- exact location of noteheads relative to the staff and ledger lines
- identification and disambiguation of dots (augmentation, *staccato*, bass clef, *fermata*, etc.)
- interpreting repeat, *Da capo* and other "flow-control" markings
- recognition of tuplets (triplets, quintuplets etc.)

It is clear from each of these cases that each example can be evaluated cleanly, either strictly correct or incorrect, rather than having multiple correct representations or a graded scale of correctness.

The system works by generating result tables, one for each of the evaluation categories, with three columns:

1. the subject of the test, highlighted with some surrounding context
2. GAMUT's interpretation of the excerpt
3. the groundtruth

If the groundtruth is incorrect, it may be edited and submitted back to the system for re-evaluation.

Figures 1 and 2 are examples of these evaluation tables.

This approach, while clearly very primitive, has already been very useful for improving the accuracy of GAMUT. For instance, when the test system was first implemented, the number of correctly pitched noteheads was around 79%. The test system allowed us to improve the algorithm to the point where it now operates at 96%. It is also very easy to add new test categories as the need arises.

[2] In previous incarnations, this system was known as AOMR (Adaptive Optical Music Recognition), which has since become a generic term.

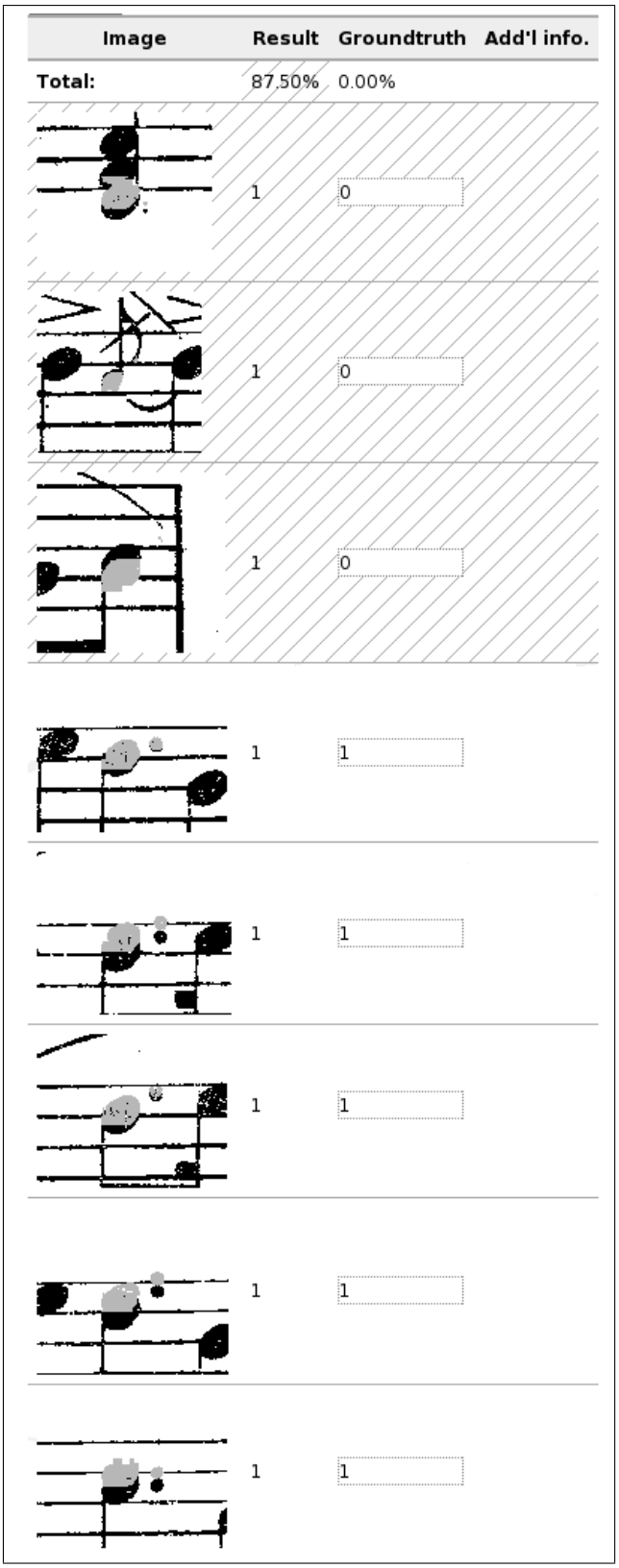


Figure 1. Excerpt of the table showing the performance of augmentation dots. The first three examples do not match the groundtruth, since they are not genuine augmentation dots.

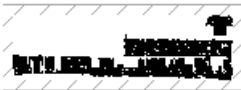
Image	Result	Groundtruth
Total:	95.65%	0.00%
	1, 2	1, 2, 3
	1, 0, 1	1
	2, 1	2, 3, 1
	2, 1, 2, 3	2, 3, 2, 3
	2	2
	2, 3, 1	2, 3, 1
	2	2
	1, 2, 3	1, 2, 3
	1	1
	1	1
	1	1
	1	1
	1	1
	1	1

Figure 2. Excerpt of the table showing beam interpretation. The first four elements do not match the groundtruth, which is a list list of numbers representing the number of beams at each point along the set of beams.

4. CONCLUSIONS

As research domains mature, evaluation metrics become more important. As the MIR community grows, efforts should be made to create metrics and groundtruth datasets. We presented here a simple but fundamental model for evaluating OMR systems and different algorithms within one OMR system. Automatically comparing the final symbolic outputs of OMR systems at the high-level is still a largely unsolved problem, and one which we hope the MIR community will consider as an application of recent developments in the searching and analysis of symbolic music data.

5. REFERENCES

[1] Beck, K., 2002. *Test driven development.* Boston: Addison-Wesley.

[2] Bruno, I., P. Bellini, and P. Nesi. 2003. Assessing optical music recognition tools. Paper presented at *Second MUSICNETWORK Open Workshop*.

[3] Chhabra, A., and I. Phillips. 1998. A benchmark for graphics recognition systems. *Proceedings of IEEE Workshop on Empirical Evaluation Methods in Computer Vision*.

[4] Hinz, S., C. Wiedemann, and A. Baumgartner. 2000. A scheme for read extraction in rural areas and its evaluation. *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 134–9.

[5] Hu, J., R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong. 2001. Why table ground-truthing is hard. *International Conference on Document Analysis and Recognition (ICDAR)*, 129–33.

[6] Kanai, J., S. Rice, T. Nartker, and G. Nagy. 1995. Automated evaluation of OCR zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17: 86–90.

[7] Mao, S., and T. Kanungo. 2001. Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 23(3): 242–56.

[8] Ng, K., and A. Jones. 2003. A quick-test for optical music recognition systems. Paper presented at *Second MUSICNETWORK Open Workshop*.

[9] Okamoto, M., H. Imai, and K. Takagi. 2001. Performance evaluation of a robust method for mathematical expression recognition. *Proceedings of the International Conference on Document Analysis and Recognition*, 121–8.

[10] Randriamasy, S. and L. Vincent. 1994. Benchmarking and page segmentation algorithms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 411–6.

[11] Wenyin, L., and D. Dori. 1997. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications,* 9: 240–50.

[12] Wenyin, L., Z. Liang, T. Long, and D. Dori. 1999. Cost evaluation of interactively correcting recognized engineering drawings. *Proceedings of IAPR Workshop on Graphics Recognition,* 335–40.