A Benchmark for Raster to Vector Conversion Systems

Ihsin T. Phillips[1] and Atul K. Chhabra[2]

[1] Department of Computer Science/Software Engineering,
Seattle University, Seattle, Washington 98122, USA
yun@seattleu.edu

[2] Bell Atlantic Network Systems, Advanced Technologies,
500 Westchester Avenue, White Plains, NY 10604, USA
atul@Basit.COM

Abstract. This paper presents a benchmark for evaluating the Raster to vector conversion systems. The benchmark is designed for evaluating the performance of graphics recognition systems on images that contain straight lines (solid or dashed), circles (solid or dashed), partial arcs of circles (solid or dashed), as well as, bounding boxes of text blocks within the images. This benchmark gives a scientific comparison of vectorization software and uses practical performance evaluation methods that can be applied to complete vectorization systems. Three systems were evaluated under this benchmark and their performance results are presented in this paper. We hope that this benchmark will help assess the state of the art in graphics recognition and highlight the strengths and weaknesses of current vectorization technology and evaluation methods.

Keywords: Engineering-drawing, Benchmark, Performance Evaluation, Raster ot Vector Conversion.

1 Introduction

Driven by the need to convert a large number of hard copy engineering drawings into CAD files, raster to vector conversion has been a field of intense research for the last three decades. In addition to research prototypes in several academic and industrial research centers, several commercial software products are currently available to assist users in converting raster images to vector (CAD) files. However, the process of selecting the right software for a given vectorization task is still a difficult one. Although trade magazines have published surveys of the functionality and ease of use of vectorization products [1], a scientific, well designed, comparison of the auto-vectorization capability of the products is not available. Responding to this need, two graphics recognition competitions were held recently[2, 3].

The benchmark we present here is designed for evaluating the performance of graphics recognition systems on images that contain straight lines (solid or dashed), circles (solid or dashed), partial arcs of circles (solid or dashed), as well as, bounding boxes of text blocks within the images. (The preliminary version of the benchmark we present in this paper was used in [4] competition.) Although the evaluator [5] we adopted and used in this benchmark is limited to the above seven entity types, nevertheless, it is useful, since all engineering drawings use only a combination of these geometric elements. Upgrading the benchmark is straight forward. We just need to provide the evaluator the new entity parameter information and the performance evaluation criteria.

This benchmark gives a scientific comparison of vectorization software and uses practical performance evaluation methods that can be applied to complete vectorization systems. Three systems were evaluated under this benchmark and their performance results are presented in this paper. We hope that this benchmark will help assess the state of the art in graphics recognition and highlight the strengths and weaknesses of current vectorization technology and evaluation methods.

This paper is organized as follows. In section 3, the benchmark specifications are presented. The performance evaluation and performance measurements are described in section 4. The performance evaluation results of the three systems are given in section 5. Our discussion is given in section 6.

2 Benchmark Specifications

2.1 Operating Platforms

The operating platforms for this benchmark are PC's running MicroSoft Windows 95, Sun SPARCstations running Solaris 2.5.1, and Silicon Graphics Indy running Irix 6.2. However, all participants chose to use PC's and SGI machines at this benchmark.

2.2 Data Set

The images used in this benchmark (both for training and testing) are selected from the UW-III document image database [6]. The methodology [7, 8] used in groundtruthing images in the UW document database series has been proven to be very reliable, therefore images in any of the series are suitable for benchmarks. We select only CAD images from the UW-III database. Our intention of the synthetic image selection were: to keep the benchmark simple in order to encourage participation, and to satisfy our domain constraint: images can only contain text, lines, arcs, and circles. However, the selected images are complex, "real life" archived drawings. Each of these images has in them over 500 object entities of lines, arcs, circles, and text. (We have removed entities other than these four types from the original images.) Some artificial distortions were added, randomly, to these selected images, to help make them resemble real images. The distortion were simple, such as changing the thickness of lines, the length of dashes and gaps in dashed lines, and the orientation and size of text.

Within the selected images, there are four kinds of drawings– Mechanical, architectural, and two distinct types of utility drawings. The images are carefully partitioned into two sets so that the images in the training set and the testing set have similar characteristics. Figure 1 shown a training image.

2.3 Input and Output Specifications: File Format

Only bi-level images were used in this benchmark. The images were in TIFF 6.0 CCITT Group 4 format. The software for generating synthetic images was built using several publicly available components [9, 10, 11]. The software, several sample images and the associated ground truth (VEC) files were made available through the benchmark web site [3].

2.4 Output Specification: Vector File Format

In order to make the evaluation simple, we specified a simpler vector file format (the VEC format) looks like below:

```
%VEC-1.0 xsize ysize [dpi]
L C|D x1 y1 x2 y2 width
A C|D xcenter ycenter radius start_angle end_angle width
C C|D xcenter ycenter radius width
T x1 y1 x2 y2 orientation fontHeight fontWidthFactor fontStrokeWidth %TEXT
```

The first line is the VEC file indicator, follows by a list of entity descriptions. The first letter of each entity description stands for the entity type: L for line, A for arc, C for circle, and T for text. The second letter indicated wether a solid (continue) or a dashed entity. The remaining are the attributes of each of the entities. The $x - y$ coordinate system must have its origin at the top left-hand corner of the image with the y axis pointing downward and the x axis pointing to the right. The units of the x and y coordinates should be in pixels.

3 Performance Evaluation

Before the benchmarking, we provided the participants a set of training images for the determination of the optimal parameters of their systems on each of the test image categories. The training images resemble those test images. The predetermined parameters of the systems were used in this benchmark. In turn, the participants provided the benchmarking committee the executables of their recognition software and the predetermined (trained) parameters of their systems for each image category. Each of the participating systems were tested on the same set of test images.

The recognition result of a participating system on a test image is matched, by the performance evaluator against the corresponding groundtruth of the test image. The matching results are the numbers of one-to-one matches, one-to-many matches, many-to-one matches, as well as the numbers of false-alarms and misses. (A complete description of the evaluator is given in [12].)

3.1 Performance Measurements: Metrics

Performance measurements for a recognition system can be formulated, using a linear combination of some or all of the matching results: the counts of the matches, the false-alarms, and the misses. Let *one2one* be the count of the one-to-one matches, *one2many* be the count of the one-to-many matches, *many2one* be the count of the many-to-one matches, $false\_alarm$ be the count of the false-alarms, $misses$ be the count of the misses, N be the count of the entities in the groundtruth file, and M be the count of the entities in the result file.

We define the following system performance measurements:

- The Detection Rate, $DetectionRate = w_1 \cdot \frac{one2one}{N} + w_2 \cdot \frac{one2many}{N} + w_3 \cdot \frac{many2one}{N}$. *DetectionRate* is, roughly, the percentage of the groundtruth entities being detected. Here, w_1 should weight more than that of w_2 and w_3 since one should favor a one-to-one match over the other two types of matches. For this benchmark, w_1 and w_2 were set to 1.
- The Misses' Rate, $MissRate = \frac{misses}{N}$. $MissRate$ is the percentage of the groundtruth entities which were not detected by the recognition system. Note that *DetectionRate* and $MissRate$ may not necessary add up to one, because of the factors involve in the computation of $Detection_Rate$.
- The False-alarm Rate, $FalseAlarmRate = \frac{false\_alarm}{M}$. $FalseAlarmRate$ is the percentage of the detected entities produced by the system but do not have their correspondences in the groundtruth.
- The Recognition Accuracy Rate, $AccuracyRate = w_4 \cdot \frac{one2one}{M} + w_5 \cdot \frac{one2many}{M} + w_6 \cdot \frac{many2one}{M}$. *AccuracyRate* indicates, roughly, the percentage of the detected entities within the result file have their matches in the groundtruth entities. Thus, one can consider *AccuracyRate* as a measurement of the overall accuracy rate of a recognition system. Again, one should have more weight on w_4 than that of w_5 and w_6 to favor the one-to-one matches. For this benchmark, w_4 and w_5 were set to 1.
- The Post-editing Cost, $EditingCost = w_7 \cdot false\_alarms + w_8 \cdot misses + w_9 \cdot one2many + w_1 0 \cdot many2one$. *EditingCost* is an estimated cost for a human post-editing effort to clean-up the recognition result. It should be clear that a higher *EditingCost* requires a higher post-editing effort. Entities missing from the result file need to be added and those false-alarms need to be removed. Moreover, for each one-to-many match, one need to remove the many (those partial matches) from the result file and add the real one to it. And for each many-to-one match, it requires one removal and many additions. Note that, w_7 is the factor for one deletion effort and w_8 is the factor for one insertion effort, the two factors should be weighted according to the post-editing tool one use for a deletion and an insertion during the post cleaning. The weights assign to w_9 and $w_1 0$ are more complex; they are depended on the method one used in the counting of these two types of matches. For example, the evaluator we used in this benchmark gives one count for a one-to-many and one count for a many-to-one. For this benchmark, we set w_7 and w_8 to one, and assigned zero to both w_9 and w_{10}.

4 Benchmark Results and Performance Analysis

Three participating systems were tested in this benchmark; two were commercial products and one came from an university. The test images used in this benchmark consists of four mechanical drawings, one architectural drawing, two utility drawings and one structural drawing, a total of eight test images. Each of three participating systems were tested on all these eight images. Their recognition results were evaluated and the system performance measurements were computed. Recall that the performance evaluator uses an acceptance threshold to determine whether a pair is a match. (A match is when the match score of the pair is equal or higher than this acceptance threshold.) The matching criteria for a pair of entities defined in [5] is roughly a similarity measurement. When the acceptance threshold is set high, the evaluator accepts only those pairs that are very similar (having high matching scores). Lowering the acceptance threshold, the evaluator lower its match requirement. We expect that with a high acceptance threshold, only those systems with high recognition precision can score high in their performance measurements, and for those not-too-good systems, the performance measurements would be low. However, we are interested to know the trends of the system performance with respect to the changes in the acceptance threshold. Our theory is that for those high recognition precision systems, lowing the acceptance threshold value may increase their performance a little, not drastically. On the other hand, for those not-too-good systems, their performance measurements may increase greatly when the evaluator's acceptance threshold is set lower. Thus, using a various acceptance thresholds in the evaluation may reveal the stability of a recognition system.

With the above concepts in mind, nine acceptance thresholds were used in the performance evaluation – from .5 to .9, in the steps of .05. That is, for each recognition result file produced by a system, we obtained nine sets of matching counts using these nine acceptance thresholds. This in term, per system, per test image, we computed nine sets of performance measurements. There are total of eight test images used in this benchmark. The results of the performance measurements *DetectionRate, MissRate, FalseAlarmRate, Accuracyrate, EditCost* with respect to the nine thresholds, for the three participating are available upon requests.

4.1 Analysis of Performance Characteristics

We are interested in learning whether the performance characteristics of recognition systems can be observed through the changes of the evaluator's acceptance threshold. We are happy to report that we did indeed observe some performance characteristics of these participating systems.

To illustrate the trend of change in the performance of the three participating systems with respect to the nine acceptance thresholds, we plot of the counts of the false-alarms vs. the misses. Due to the limited space, a sample of the plot is given in Figure 2-4. Figure 2 (3 and 3 also) contains three nine-point curves (one curve per system) where the nine points correspond to the nine thresholds used.

The first point on each curve corresponds to a threshold of 0.5 and the last point on each curve corresponds to a threshold of 0.9. We observed the followings.

- In general, all three curves in each of the plots show upward trends. That is, as the acceptance threshold is increased, all three systems produce more misses and more false-alarms.
- In general, the first three or four points on most of the curves (they correspond to the threshold values 0.5, 0.55, 0.6, and 0.65) either form a tight cluster, or have equal or higher counts of misses and false-alarms than the counts for the points corresponding to 0.65, 0.7 or 0.75 thresholds. The interpretation for this trend may be that using an acceptance threshold below .65 does not yield a better evaluation for a given system. Or, it may be that the performance measurements produced by the evaluator using thresholds below 0.65 are not reliable (we suspect that with the acceptance threshold set too low, the evaluator may be making matching errors consequently resulting in more misses and false-alarms.) We are currently investigating this.
- In most of the cases, all systems produce more false-alarms than misses. This may be partly due to one of the following reasons. (1) At present, the evaluator does not match any dashed entity to any solid entity. So, if a dashed-line in a test image is detected by a vectorization system as several little straight line segments, the evaluator produces counts of one miss (dashed-line) and several false-alarms (little line segments.) (2) When a text string in a test image is not correctly detected as a text region, it is often 'vectorized' into several small lines, arcs, etc. In this case, the evaluator currently produces counts of one miss (the missing text string) and several false-alarms (the little 'vectors').

We also observed some performance characteristics for each of the three systems. For example, we observed one system has the smallest increases in the counts of misses and in the counts of false-alarms. If we take the amount of increases, with respect to the increase in the acceptance threshold, as an indicator of the stability of a system, this system win over the other tow by a significant margin. The same system also produces much fewer misses than the other two. For the four mechanical drawings, the system which was designed specific for mechanical drawings produces much fewer false-alarms and fewer misses than the other two systems. It is apparent that customizing a system for a specific type of drawings can lead to a significant improvement in performance. The System-C has been designed specifically for mechanical drawing recognition. It is clear from the above that System-C performs better on mechanical drawings than the other two systems.

5 Discussion

The benchmark limited itself to a quantitative evaluation of the automatic vectorization capability of the participating systems. Several other constraints were imposed either due to lack of time and resources or in order to keep the evaluation protocol simple. The primary constraints were as follows. (1) Only synthetic bi-level images were used for both training and testing. (2) The only 'noise' in the images was in the form of thickness of lines, length of dashes and gaps in dashed lines, and the orientation and size of text. No 'image noise' was added. (3) We only tested at the image resolution of 200 dots per inch. (4) We only tested for detection of straight lines, arcs, circles, and text. Detection of polylines, dimensioning, objects, symbols, etc. was not tested. (5) Only one kind of dashed line was used. This was the simple dash-dash line. (6) No match was attempted between dashed entities and solid entities.

There are some known shortcomings in our evaluation process which we will address in the near future. If a vectorization system erroneously recognizes a dashed line as a sequence of short continuous lines, then our evaluation method assigns a single miss but a large number of false-alarms (because we do not attempt to match a dashed line with continuous line segments). We need to allow matching of dashed lines with several small line segments, but this should be penalized somewhat due to the fragmentation introduced.

If a text region in not correctly identified, then we assign a single miss accompanied with a large number of false-alarms. This happens because if a text region is not correctly identified, then the vectorization software will invariably try to 'vectorize' the region. The resulting short lines ('vectors') will count as false-alarms because we do not attempt to match a text area with any other type of entity. In order to correct the misinterpretation, one only needs to box a text region and mark it as text (we are not talking about OCR here. OCR is outside the scope of this benchmark). This is a very simple post-processing operation. Therefore, this kind of error should not be penalized so heavily.

Gathering data to test and compare graphics recognition systems is very time consuming. This benchmark only used synthetic images with associated ground truth. Future benchmarks should include synthetic images with image degradation and real images with manually created ground truth. The graphics recognition community needs to collaborate in building a database of images and ground truth files.

The real strengths and weaknesses of a system are revealed by stress testing the system. We can accomplish this by testing the performance of a vectorization system with increasing image degradation and increasing image complexity. This should be attempted in a future benchmark.

Future benchmarks will hopefully attract participation from many more vectorization software companies. All the systems that we tested in this benchmark are among the best products or research prototypes available for vectorization. A larger number of systems in the benchmark will provide us broader trends and will give us a real assessment of the state of the technology.

References

1. David Byrnes. Raster-to-vector comes of age with AutoCAD Release 14. *CADALYST*, pages 48–70, December 1997.

2. R. Kasturi and I. Phillips. The first international graphics recognition contest – dashed-line recognition competition. Graphics Recognition: Methods and Applications, First International Workshop, University Park, PA, USA, August 1995.
3. A. Chhabra and I. Phillips. Web page for the Second International Graphics Recognition Contest – Raster to Vector Conversion. http://graphics.nynexst.com/iapr-tc10/contest.html.
4. *Proceedings of Second IAPR Workshop on Graphics Recognition*, Nancy, France, August 1997.
5. I. Phillips, J. Liang, and R. Haralick. A performance evaluation protocol for engineering-drawing recognition systems. In *Proceedings of Second IAPR Workshop on Graphics Recognition*, pages 333–346, Nancy, France, August 1997.
6. I. Phillips. Users' reference manual. CD-ROM, UW-III Document Image Database-III.
7. I. Phillips, J. Ha, R. Haralick, and D. Dori. The implementation methodology for the cd-rom english deocument database. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 484–487, Tsukuba, Japan,, October 1993.
8. I. Phillips, S. Chen, and R. Haralick. Cd-rom document database standard. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 478–483, Tsukuba, Japan,, October 1993.
9. VOGLE, a public domain device portable graphics library. ftp://munnari.oz.au/ pub/graphics/vogle.tar.gz. Used for the Hershey fonts and the software for rendering the fonts.
10. Sam Leffler and Silicon Graphics, Inc. TIFF software distribution. ftp://ftp.sgi.com /graphics/tiff/tiff-v3.4beta036-tar.gz. Used for rendering the training and test images in TIFF CCITT Group 4 format.
11. D. Knuth. The portable random number generator. http://www-cs-faculty. stanford.edu/ knuth/programs.html. Also published in *The Art of Computer Programming, Volume 2/Seminumerical Algorithms,* 3rd edition, section 3.6. Addison-Wesley, Reading, MA, USA, 1997.
12. I. Phillips, J. Liang, R. Haralick, and A. Chhabra. A performance evaluation protocol for graphics recognition systems. In K. Tombre and A. Chhabra, editors, *Graphics Recognition: Methods and Applications, Second International Workshop, Nancy, France, August 1997, Selected Papers*, Lecture Notes in Computer Science. Springer, Berlin, 1998. to appear.

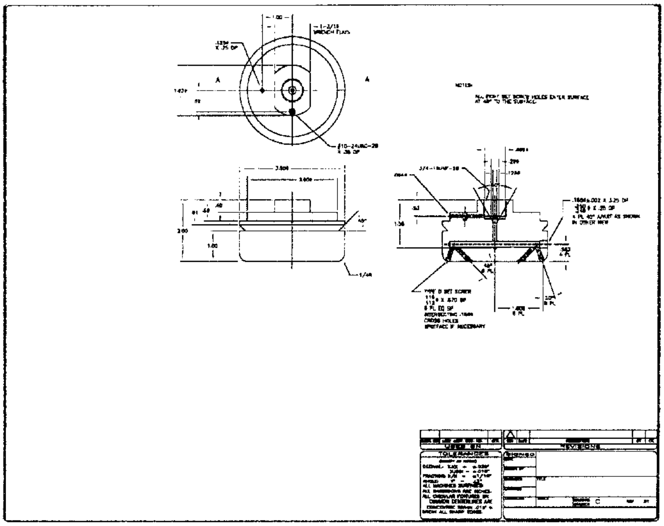

Fig. 1. Training image mech.tif (mechanical drawing)

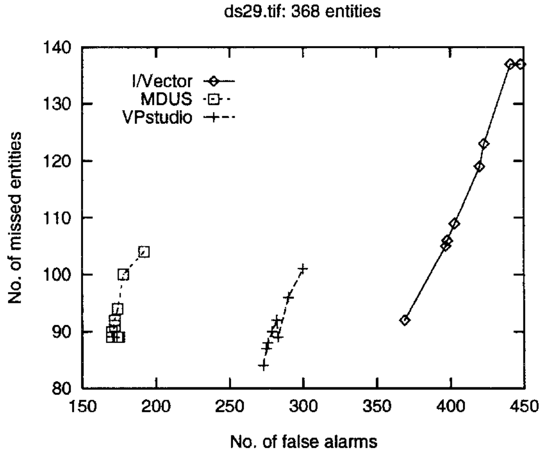

Fig. 2. Performance curves of the systems for the image ds29.tif (image of a mechanical drawing)

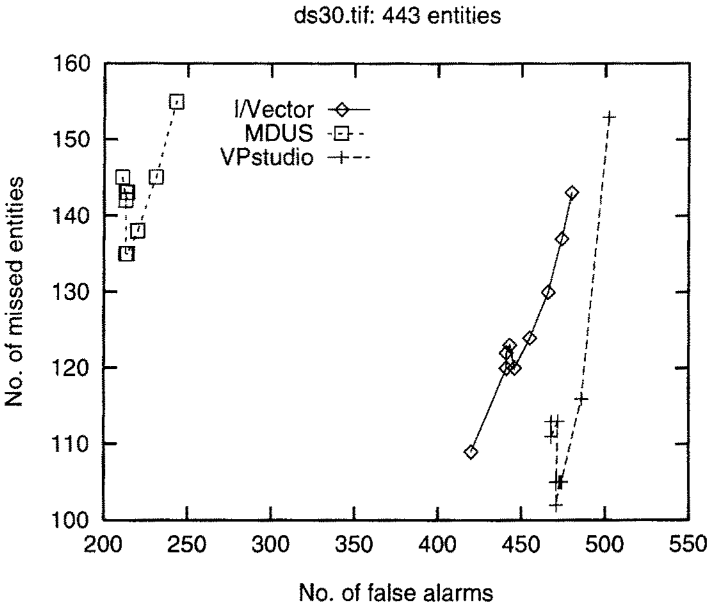

Fig. 3. Performance curves of the systems for the image ds30.tif (image of a mechanical drawing)

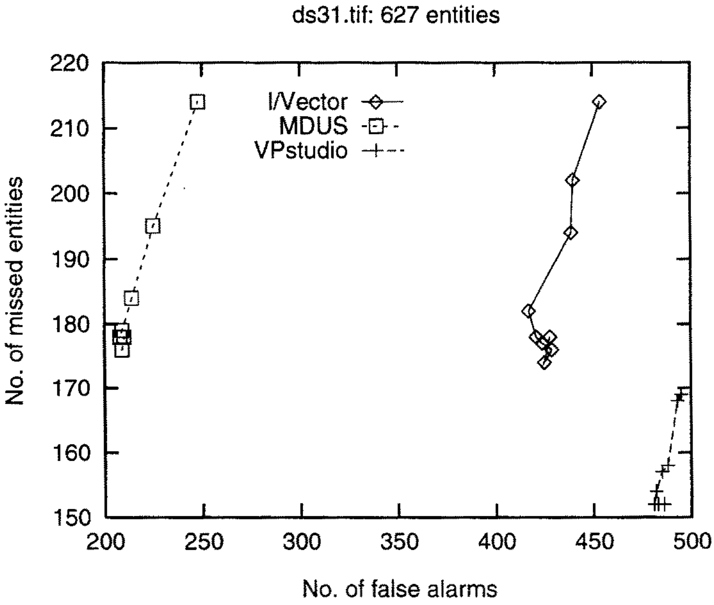

Fig. 4. Performance curves of the systems for the image ds31.tif (image of a mechanical drawing)