

Assessing Engineering Drawings through Automated Assessment: Discussing Mechanism to Award Marks

Kim Nee Goh¹ and Rofans Beleam Hilisebua Manao²

*Computer and Information Sciences Department
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia
gohkimnee@petronas.com.my¹, inbox@rofans.net²*

Abstract

Engineering Drawing (ED) is a widely taught subject in the universities offering engineering degrees. The rise of the ratio between the ED course instructors and participants has driven a need towards automatic assessment of ED assignments. Our initial research discussed the technicalities of how automated marking can be achieved. As part of the whole assessment software, a comprehensive automated award system should be developed. This award system should imitate the methods of manual awarding implemented by actual ED tutors or lecturers. This should encompass checking a student's work based on its correctness and creativity against the sample answer scheme provided. The mark has to be unbiased and inherent to the distribution of the overall marks. This paper will discuss case scenarios of ED in comparison to the answer scheme and how the awarding system should work in different scenarios. In order to develop these rules, we have created a simple module capable on reading simple circle or group of circle drawings. We will have several test cases of different drawings and conduct our study from these samples. The paper concludes with results of the experimentation and recommendations driven from the results.

Keywords: *engineering drawing, automatic, assessment, marking, marks*

1. Introduction

Engineering Drawing (ED) is a type of technical drawing that is used to clearly define engineering objects such as buildings, valves, or even a tiny screw. Students enrolled for engineering courses in an educational institution will most likely need to enroll in this course during their study. In some institutions, the number of students surpasses the number of instructors by many fold, thus there are not enough instructors to assess assignments from these students. This raises the need for automatic ED assessment software.

AutoDesk AutoCAD is the most widely used Computer Aided Design (CAD) tool for ED. AutoCAD is widely used especially in education and research. AutoCAD has many file formats; the most popularly used is Drawing (DWG). Another file format, in which we used, is Drawing Exchange Format (DXF). These formats are professionally documented it will allow us to learn and build necessary Application Programming Interface (API) to extract and manipulate the codes.

Because DXF is a file type is composed of ASCII based codes, it makes it easier to process as compared to other file formats. DWG is built on bit codes, thus manipulating the codes would be more complicated. As we convert from DXF to Scalable Vector Graphics (SVG), our program enables easy access to obtain the objects values in an ED. SVG is Extensible Markup Language (XML) graphic vector file format, recommended by the World Wide Web Consortium. Because it is an XML based file, SVG is easier to handle and process as compared to DXF.

From these authors [1], a prototype was developed that relies on the flexibility of AutoCAD DXF and W3C SVG. Our paper describes how the software reads DXF code and

converts the code into equivalent XML based SVG format. The SVG produced will be used to compare with the answer scheme provided by the instructor. Currently, our prototype proves this concept by working with a circle or groups of circles drawing. We are currently working on extending the capabilities of the software.

In our prototype, it works by comparing the SVG files attributes. For example, an SVG circle drawing has three attributes. First is the “r” which defines the radius of that circle. The others are “cx” and “cy” which is the x and y coordinate of the circle’s midpoint. Figure 1 shows a sample of SVG code.

```
<svg
xmlns="http://www.w3.org/200
0/svg" version="1.1">
<circle
cx="9.667172840374078"
cy="13.48236319755094"
r="2.702626290815926"
stroke="black" stroke-
width="1"/>
</svg>
```

Figure 1. Sample SVG Code for a Circle

Our prototype works by recording these attributes, values and subtracting them with the answer scheme’s attributes to get the differences, which could be inferred as an error. The result of this subtraction is a number that we call “Error Value” (EV). This means that if $EV=0$, the student did not have any error in their ED and will receive full mark. However if $EV>0$, then the student’s marks shall be deducted. The bigger the EV value, the larger the marks deduction would. The deduction is based on the value of EV and it’s called “Deduction Factor” (DF). For a group of circles drawing, our prototype will treat each circle separately which will result in a group of EVs and DFs. Each circle will contribute a certain weight (W) towards the whole mark and the drawing “Overall Deduction Factor” (ODF) will be based on the cumulative result of $(DF*W)$ for each circle.

While it is common sense to understand that a bigger EV means a bigger DF, which is not the case for every time. In a manual grading system, instructors sometimes deduct marks based on other factors. And our prototype is yet to fully imitate this mechanism. We believe that if our automatic assessment software is able to imitate the manual grading system, the system will be more likely to be accepted and used. Driven by these ideals, we are set to find means on how to enable our prototype to imitate the manual grading system by discussing several case studies. Since our prototype can only understand circle drawings so far, we will base our research on this type of drawings only.

In this paper, we will first outline several findings from researchers that have worked on automatic assessment for programming courses. Then we will present viewpoints on automatic assessment. We will also discuss some ideas on how to read raw shapes and drawings. Several cases of ED assignments will be discussed to show different case scenarios. At the end of this paper, we will provide some recommendations that will enable our system reach its objectives.

2. Previous Work

2.1. Automatic Assessment

A lot of work has been done on automatic assessment on programming courses. Authors in [2] and [3] serves as pioneer in this area, authors [4] and [5] combines algorithm visualization into automatic assessment and authors [6] and [7] provide more details on additional features of automatic assessment software.

The idea of automatic assessment in programming courses was first derived because of two factors. Firstly, manual assessment requires help from software to compile and display the answers to some programming exercises. Secondly, as programming courses become more popular, there were more students than the instructors could handle.

As elaborated in [2], automatic assessment could not be made entirely without a human's intervention as it was yet impossible for a computer program to examine several aspects of a programming assignment. For example, it is almost improbable for a program to interpret students' comments of the code they wrote. However, the author of [3] argues that by reengineering the assignments, only a few possibilities of answers can be produced. By limiting the number of possible outcome, a precise algorithm can be written to automatically assess a particular programming assignment.

In [6, 7] and [8], details on how to develop more reliable assessment software are outlined. Several important factors that were discussed are the usage of graphical interface, client-server architecture, email protocol and web based quiz system. The application of these features will not only enhance the automatic assessment but also provide the ability to perform virtual mass courses as shown in [9].

A slightly different concept is implemented in [10] by the creation of mathematic automatic assessment. This is made possible because mathematical equations can be translated into a series of codes. The same idea is implemented in [11] where codes are used to represent digital logic design.

2.2. ED Representation

As mentioned, automatic assessment only works on assignments that can be translated into code. Some ED representations are either available in handwritten drawings or bit-coded files. These representations are impossible to be automatically assessed.

DXF is one of the few file formats that are able to represent Engineering Drawing in ASCII based codes. While it comes in handy and provides a great opportunity to allow automatic assessment, it is still hard to write a computer program to read DXF as its design is complicated and its proprietary license limits our efforts to conduct research on it. SVG comes as a viable alternative as the works done in [12, 13] and [14] proves that a bidirectional conversion can be performed between DXF and SVG without losing any ED data. It was also proven that SVG format is lesser in size and therefore faster to be processed. The conversion is basically done by converting only critical part of the file that directly represents the drawing while ignoring other parts of the file.

In order to resolve other factors that the current computer program is not yet able to interpret in ED, the researchers of [15] have determined that there are three cases where considerations need to be made possible. The first is intersection between two lines, the type of the intersection and the angle between the two lines will determine how the drawings are best recognized. The second and third is parallelism and perpendicularity where starting and end points of two different lines will determine how parallel and perpendicular those two lines are. The last case is circles and arcs that require their own sets of understanding. The authors of [15] have proposed formulas to calculate the factors of these four cases. In [16] several algorithms have also been proposed to address features from CAD representation.

3. Method

3.1. Prototype

Based on our previous work [1], our prototype is only able to work with circles or groups of circles. A reading of a circle is straightforward, while a group of circles are treated separately as a single object before the DF of each circle is accumulated to produce the ODF. Figure 2 and Figure 3 represents the different types of drawings that our prototype can

understand. As long as a drawing is composed fully of AutoCAD Circle objects, our prototype is able to mark it.

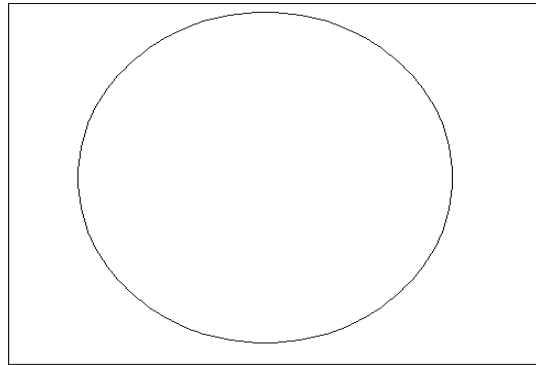


Figure 2. A CAD Representation of Circle

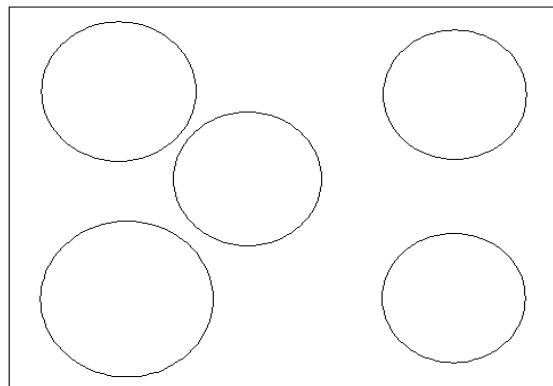


Figure 3. A CAD Representation of a Group of Circles

3.2. Marking Process

After the circle drawn and saved into DXF file has been converted into SVG as shown in Figure 1, a circle is represented by tag <circle> that has three compulsory attributes: “r”, “cx” & “cy”. Table 1 summarizes each of the attributes’ definition.

Table 1. Meanings of Circle’s SVG Attributes

Attributes	Significant
r	the radius of the circle
cx	x-coordinate of the circle’s midpoint
cy	y-coordinate of the circle’s midpoint

To find the “Error Value” (EV), the system will first determine the values of r_m , cx_m , and cy_m (circle’s attributes of the answer scheme) and also the values of r_s , cx_s and cy_s (attributes taken from student’s file). The value of EV is as shown in Formula 1 to calculate error value and DF is shown in Formula 2.

$$EV = |r_m - r_s| + |cx_m - cx_s| + |cy_m - cy_s| \quad \text{_____} \quad (1)$$

$$DF = EV * W \quad \text{_____} \quad (2)$$

The concept is DF is based on several set of acceptable EV values. For a certain EV value then there is a set of certain DF values. Table 2 presents simple representation of this concept.

Table 2. A Sample of Relationship between EV and DF

EV	DF (as percentage of the full mark)	Possible mark (out of 100)
0	0	100
1 – 2	1 – 15	85 – 99
3 – 6	16 – 35	65 – 84
7 – 12	36 – 70	30 – 64
13 – 18	71 – 100	0 – 29

The example in Table 2 however is just a sample case representation. We are yet to determine the relationship between EV and DF by conducting more sample ED case scenarios to determine the best option.

In the case of multiple circles drawing, we will utilize the value of ODF instead of individual DF of each circle. Suppose the example of Figure 3 where there are five circles, assuming names of Circle 1 to 5. Each circle has its own EV and DF as represented in Table 3. Each circle also has a Weight (W) in regards as a component towards the overall drawing.

Table 3. Values of EV and DF of each circle in Figure 3

Circles	EV	DF	W
1	ev ₁	df ₁	w ₁
2	ev ₂	df ₂	w ₂
3	ev ₃	df ₃	w ₃
4	ev ₄	df ₄	w ₄
5	ev ₅	df ₅	w ₅

Therefore, to derive the value of ODF we need to accumulate the DF of the individual circle. The formula to calculate the overall DF (ODF) is thereby presented in Formula 3.

$$ODF = \sum_{i=1-5} (df_i * w_i) \quad \text{—————} \quad (3)$$

4. Results and Discussion

There are at least three possible cases that our prototype has to consider in order to mark circle drawings properly. The cases derived from the variations of the attributes. The first case comes from variation of radius, “r”, the second comes from variation of midpoint coordinate, “cx” and “cy”, and the last comes from a combination of both. In this section, we will discuss the possibility of performing automatic assessment.

In order to understand whether each case needs to be treated differently, we will first alter the properties of one circle and check for its final ODF. Once the ODF of that particular state is recorded, we will modify another circle and check whether the final ODF is equivalent to the previous state. Based on the calculations that we have performed, this section will discuss and present the results.

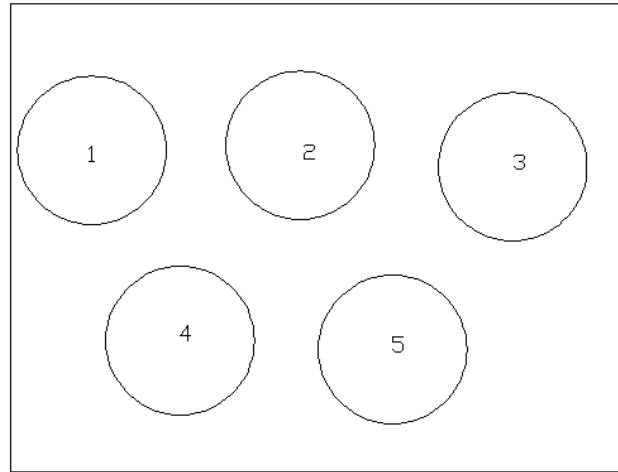


Figure 4. Group of Circles

Figure 4 represents five circles drawn in AutoCAD, each numbered accordingly. They all have the same radius ($r = 1.0143$). The “cx” and “cy” properties for each circle are defined in Table 4. Since all circles are equivalent, each circle is then given a 20% DF value. This means, if a student missed drawing one of the five circles, he or she only gets 80% of the total mark.

Table 4. Values of cx and cy of each Circle in Figure 4

Circles	cx	cy
1	2.16	5.19
2	5.00	5.26
3	7.90	4.97
4	3.36	2.59
5	6.26	2.47

4.1. Radius “r” Variation

A possible case where one of the circle’s radius, “r”, property could be invalid is illustrated in Figure 5. In this example, the “r” value of one of the circle has been resized while its remaining properties remain intact. While Circle 5 in Figure 5 assembles a smaller version of Circle 5 in Figure 4, we have experimented with a larger and smaller versions of other circles in the drawing as well.

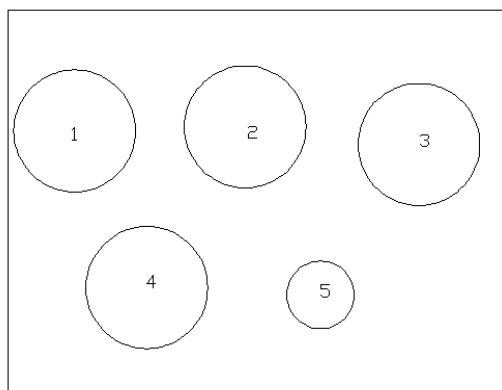


Figure 5. Example of “r” Variation

In Figure 5, the value of r_5 is 0.562. There is a slight difference with the other circles' "r". In order to calculate the EV of this particular circle, we just need to implement Formula 1, followed by Formula 2 to determine the DF. Through this, we will get the value of $EV = 0.4522$.

We tried to resize the value of r_5 a few times and found that the marking of this case is quite straightforward. We also found this to be consistent while we tried with other circles. Therefore, we conclude that the formulas that we have presented are sufficient. In the case of "r" variation, the current EV formula is sufficient.

4.2. "Cx" and "Cy" Variation

In order to understand this case, we have translated the midpoint of Circle 4 by a factor of (-0.5, -0.5) as presented in Figure 6.

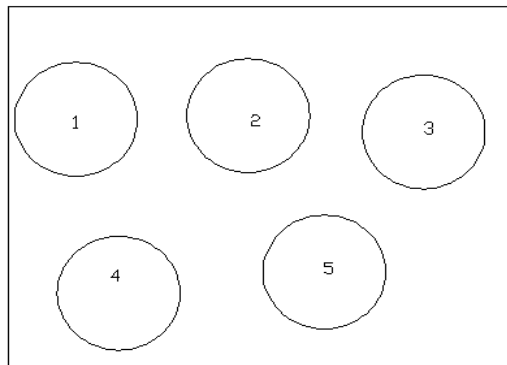


Figure 6. Example of "cx" & "cy" Variation

According to this state, the EV value should be 1. We found this value to be consistent when we translated the midpoint of other circles by the same factor. It seems that the variation of these attributes will not derive a need to alter the current EV formula.

4.3. Combined Variation

The last possible case of variation is the combination of "r" together with "cx" and "cy" variation. As shown in Figure 7, we have resized the "r" property of Circle 5 into $r_5 = 0.0621$ and translated the cx_5 and cy_5 by a factor of (-0.5, -0.5). This will produce an EV of 1.4522.

However, when we modified Circle 5 and assign $r_5 = 0.0621$ and move cx_5 and cy_5 by a factor of (-0.25, -0.25), we found out that the EV was also 1.4522. The question now is whether the two cases should produce the same DF. To observe the cases, we have created scenarios based on Figure 7 and Figure 8.

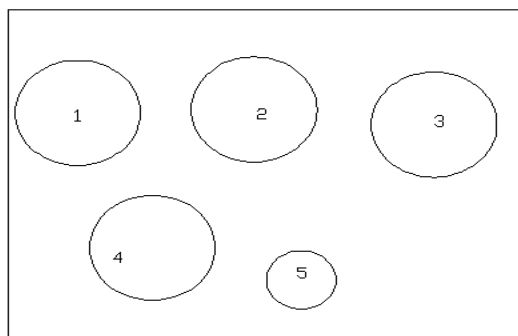


Figure 7. Circle 5 is Translated by a Factor of (-0.5, -0.5) and Slightly Reduced in Size

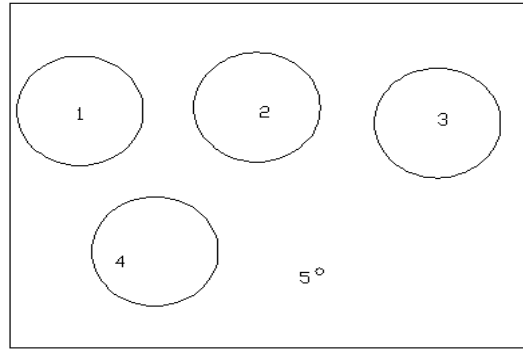


Figure 8. Circle 5 is Translated by a Factor of (-0.25, -0.25), size is Greatly Reduced

Majority would argue that students producing the second drawing (Figure 8) should be penalized with more reductions to the marks awarded. It seems that a straightforward solution is unfair for this problem. However, we must not modify or vary the current as it has worked well the ordinary radius variation and midpoint coordinate variation.

4.4. Possible and Proposed Solution

From the above case scenarios, we believe that the combination of more than one variation will cause the variables in the formula to no longer carry the same weight. We proposed the adjusted version of “Error Value”, which is expressed in Formula 4.

$$EV = @_1|r_m-r_s| + @_2|cx_m- cx_s| + @_3|cy_m- cy_s| \quad \text{————— (4)}$$

While the rest of the formula contains exactly the same information as the previous formula, Table 5 shows the meanings of the new variables.

Table 5. A Sample of Relationship between EV and DF

Variable	Significance
@ ₁	Weight of the result of $ r_m-r_s $
@ ₂	Weight of the result of $ cx_m- cx_s $
@ ₃	Weight of the result of $ cy_m- cy_s $

The exact value of the new variables can only be determined through further experimentation using different case scenarios.

4. Conclusion

Although the current prototype only works with circles, we would need to find the best solution for assessing objects in an ED. Marking should be based not only on correctness but also creativity, the variations of attributes for each object. Our paper showed that it is possible to award marks for creativity based on the proposed formulas. Further experimentation with different objects and case scenarios is necessary to further improve the formulas.

References

- [1] K. N. Goh and R. B. H. Manao, “Automatic Assessment for Engineering Drawing”, *J. Soft Computing and Software Engineering*, (2013).
- [2] D. Jackson, “A Semi-Automated Approach to Online Assessment”, *Proceedings of the 5th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000)*, Helsinki, Finland, (2000) June 11-13.
- [3] R. Saikkonen, L. Malmi and A. Korhonen, “Fully automatic assessment of programming exercises”, *SIGCSE Bull*, vol. 33, no. 3, (2001).

- [4] A. Korhonen and L. Malmi, "Algorithm Simulation with Automatic Assessment", Proceedings of the 5th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000), Helsinki, Finland, (2000) June 11-13.
- [5] L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä and P. Silvasti, "Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2", 3rd J. Informatics in Education, vol. 3, no. 2, (2004).
- [6] C. Douce, D. Livingstone and J. Orwell, "Automatic Test-Based Assessment of Programming – A Review", ACM 5th J. Educational Resources in Computing, vol. 2, no. 4, (2005).
- [7] M. Joy, N. Griffiths and R. Boyatt, "The BOSS online submission and assessment system", J. Educational Resources in Computing, vol. 5, no. 3, (2005).
- [8] J. English and P. Siviter, "Experience with an Automatically Assessed Course", Proceedings of the 5th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000), Helsinki, Finland, (2000) June 11-13.
- [9] L. Malmi, A. Korhonen and R. Saikkonen, "Experiences in Automatic Assessment on Mass Courses and Issues for Designing Virtual Courses", Proceedings of The 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, (ITiCSE'02), Aarhus, Denmark, (2002) June 24-28.
- [10] A. Rasila, M. Harjula and K. Zenger, "Automatic assessment of mathematics exercises: Experiences and future prospects", ReflekTori2007 Symposium of Engineering Education, Helsinki, Finland, (2007).
- [11] Z. Stanisavljevic, B. Nikolic and J. Djordjevic, "A Module for Automatic Assessment and Verification of Students' Work in Digital Logic Design", Proceedings of the 19th International Conference and Workshops on Engineering of Computer-Based Systems, Novi Sad, Serbia, (2012) April 11-13.
- [12] N. Ahmad and A. F. M Anwarul Haque, "Manufacturing Feature Recognition of Parts Using DXF File", Proceedings of the 4th International Conference on Mechanical Engineering, Dhaka, Bangladesh, (2001) December 26-28.
- [13] M. A. Fahiem and S. Farhan, "Representation of Engineering Drawings in SVG and DXF for Information Interexchange", Proceedings of the 6th WSEAS International Conference on Circuits, Systems, Electronics, Control and Signal Processing, Cairo, Egypt, (2007) December 29-31.
- [14] H. Kuchynkova, "Compatibility of Data Transfer between CAD Applications", 12th J. Radioengineering, vol. 12, no. 4, (2003).
- [15] L. Yan and L. Wenyin, "Engineering drawings recognition using a case-based approach", Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, (2003) August 3-6.
- [16] E. S. A. Nasr and A. K. Kamrani, "A New Methodology for Extracting Manufacturing Features from CAD System", J. Computers and Industrial Engineering, vol. 51, no. 3, (2006).

Authors



Kim Nee Goh obtained her MSc in Human-Computer Interaction from University of Michigan, USA in 2007. She is currently working as a Lecturer in the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia. Her research interest includes persuasive technologies and interface and interaction design especially in the health domain. She has published technical papers in international, national journals and conferences.



Rofans Beleam Hilisebua Manao obtained his degree in Information and Communication Technology from Universiti Teknologi PETRONAS in 2012. He joined Accenture Technology Solutions Sdn Bhd in April 2013. His responsibility is providing technical support on Oracle Unified Inventory Management. He is currently pursuing his masters by research degree at his alma mater under a part time study mode. His research is related with Software Pattern, Cloud Computing, and Model Checking.

