Automarking of CAD Tutorials

Kim Seng LEE\textsuperscript{a*} and Yi Han HO\textsuperscript{b}

\textsuperscript{a} Department of Mechanical Engineering, Engineering Drive 1, National University of Singapore, Singapore 117576

\textsuperscript{b} Graduate, Department of Mechanical Engineering,

*Corresponding Author’s E-mail: mpeleeks@nus.edu.sg

**Keywords:** CAD Tutorial; Automarking; Technology; Software Programme, CAD Modelling, DrawCompare

**Extended Abstract**

As part of the education process, Computer Aided Design (CAD) software such as SolidWorks are used to educate students on the process of CAD modelling. Tutorials are designed for students to practice what they have learnt, and hence there is also a need to grade the students' work so that they can learn from their mistakes. Manually grading such tutorials is a tedious and error-prone process, as the teaching staff doing the grading (markers) have to manually compare the students' submissions to the answer key, which can take an extremely long time, and the marker might miss out certain mistakes that the students have made. Therefore there is a need to write a programme to automate this process in order to save time and improve the accuracy of the marking.

Two features were identified in SolidWorks that were useful for this purpose, namely the "Compare Geometry" and "DrawCompare" functions. These features were designed in SolidWorks for the purpose of comparing the differences between earlier and later versions of the same model, when users want to know the changes made between the different versions. While these functions are useful for the automarking process, there were also challenges that had to be overcome because the comparison to be made here is not between different editions of a model, but between the student's answer and the model answer.

For the “Compare Geometry” function, the main problem was that SolidWorks makes use of the coordinate system within the part document as a baseline for comparison. In other words, the programme places the origin of each document together in order to make its comparison. Problems arise because the student may not have begun his modelling from the same point as the model answer, and the “Compare Geometry” function will produce a comparison based on different origins with respect to the modelled part. This comparison, based on different origins within the document, would not be useful to the grader at all as the differences highlighted by the programme would be erroneous. Several solutions to this problem were explored, but the most suitable solution we found was to force...
students to begin modelling their parts from the same reference point, as other solutions did not completely fit into the purposes of this project, which is to highlight the actual differences between the student and model answers.

As for the “DrawCompare” function, the problem arises from the function assuming that the drawing views are in the exact same positions on the drawing sheet, as normally changes made to the models are automatically updated by SolidWorks onto the drawing sheet, and there is little need for users to shift the positions of the drawing views. However, when comparing drawings done by different people (in this case the student and marker), these drawing views may be in the same generic order (i.e. isometric, left/right, plan view, etc.) but their exact x- and y-positions on the drawing sheet are going to be different from that of the model answer. When “DrawCompare” makes this comparison, it simply superimposes the two drawings on top of each other in order to highlight the differences in pixels. In order to solve this, a programme was written in the SolidWorks API to shift these views to the appropriate positions in order for “DrawCompare” to function the way that would be helpful for this project. With this code, however, there are some sources of error that would cause the programme to not work properly. For example, hidden views located within the drawing sheet parameters would cause the programme to grind to a halt with a run-time error. The only way to solve this is to manually move these hidden views out of the drawing sheet and to run the programme again.

It was later found that tutorials involving assemblies would be submitted with a mixture of its constituent parts, drawing and assembly files. In order for the programme to function efficiently and not compare different parts with each other, there was a need to have a programme to sort out these files into various folders before the “Compare Geometry” function could be automated effectively.

One of the key emphasis during CAD modelling is the different projection angles of drawing documents (i.e. third-angle or first-angle projection). Depending on the requirements of the CAD drawing requested, users have to make use of the appropriate projections. Hence, during the marking process, there is a heavy penalty imposed for students who make use of the wrong projection symbol. In order to reinforce this, a code was written to identify the projection symbols used in the drawing document and these documents are sorted based on their projection symbols, or the lack of it, so that the penalty can be appropriately meted out, and these mistakes would not be overlooked by the marker.

With the work done during the course of this project and the implementation of suggestions raised during the tutorial sessions, the programme has been proven to be useful based on tests conducted using sample files from student tutorials. Although it is not completely perfect, the programme is useful as it reduces the input and amount of time a tutor would require to grade a batch of up to 400 tutorials, while at the same time improving the tutor's efficiency in not overlooking certain mistakes that students may make. As a result, this project would undoubtedly enhance the learning for students who are being introduced to SolidWorks as a CAD modelling software.