2018 ASEE ANNUAL CONFERENCE & EXPOSITION
JUNE 24–27, 2018
SALT PALACE CONVENTION CENTER
SALT LAKE CITY, UT

Paper ID #21082

CADcompare™: A Web-based Application that Compares PDF CAD Drawings

Mr. Lukas W. DiBeneditto, Purdue University

Lukas W. DiBeneditto is an Undergraduate Research Assistant of Mechanical Engineering Technology at Purdue University, Purdue Polytechnic New Albany. He received his Associates of Arts in Communications from Jefferson Community and Technical College and is a Certified SOLIDWORKS Associate (CSWA). He is the lead software developer for CADcompare, a web-based application designed to decrease grading times and increase the accuracy of engineering CAD drawings. CADcompare can compare multiple student files to a grading key automatically, it highlights easily missed differences, and utilizes PDF CAD drawing files which most CAD software can generate. He aspires to help people with biomedical engineering in the development and research of orthopedic and robotic prostheses.

Dr. Rustin Webster, Purdue University, New Albany

Dr. Rustin Webster is an assistant professor of Mechanical Engineering Technology at Purdue University. Prior to joining Purdue, Dr. Webster worked in the Department of Defense field and specialized in mechanical design, research and development, and business development. He studied at Murray State University and the University of Alabama at Birmingham where his research was on immersive virtual learning environments for educational training purposes. Furthermore, Dr. Webster has received various professional certifications from the American Society of Mechanical Engineers, SOLIDWORKS, the Project Management Institute, and NACE International.

CADcompare™: A Web-based Application that Compares PDF CAD Drawings

Abstract

This work in progress describes the development of a web application titled CADcompare™, which automatically compares, displays, and highlights differences in Portable Document Format (PDF) files of computer-aided design (CAD) drawings and is specifically designed to compare multiple student files to an instructor's grading key. CADcompare augments the grading process of technical and engineering CAD drawings by highlighting differences that can be easily missed by a human grader, such as incorrect line type(s), color(s), or double lines (i.e., lines on top of each other). Some CAD software has built-in comparison tools, however, none of the comparison tools accept PDF files to compare, are web-based applications, or can compare multiple student files at once like CADcompare can. Grading engineering CAD drawings with accuracy and fairness can take a lot of time, the intended use of CADcompare is to act as a grading tool to help instructors grade faster, more accurately, and without unintended bias. Spring 2017, a Windows® based proof of concept version of CADcompare installed on a personal computer showcased the strengths of the software. CADcompare was able to compare multiple student drawings to the grading key much faster than previously used methods. Outcomes from user testing prompted the current development of a web-based version. This paper offers general details on how CADcompare compares PDF files, market analysis, the work in progress, and a planned research study comparing grading times with and without CADcompare in an introductory engineering graphics course.

Introduction

Often in an introductory engineering graphics course, students will learn about or review freehand sketching, geometric entities, solid primitives, line types, line precedence, projection types, dimensioning and tolerancing, engineering drawings, computer-aided design (CAD), etc. At some point, the instructor may ask students to use CAD software, such as AutoCAD® to replicate a dimensioned two-dimensional (2D) engineering drawing. If a student creates the drawing correctly and precisely follows all instructor given setup directions on how to generate (i.e., plot or save) the drawing as a Portable Document Format (PDF) file, then the student and instructor PDF files should be nearly identical. If aligned and compared using software-based image processing techniques, any differences between the student and instructor PDF files should be immediately apparent and thus should decrease grading times.

Image processing techniques and consequently image differencing can be a very complex and challenging topic to explain and understand, however; a general synopsis is needed concerning this work in progress. Generally, computers display images as thousands of tiny dots that refresh

on a screen many times a second. These dots are called pixels (admittedly an oversimplification), and many thousands of pixels make up an image (Smith, n.d.). A pixel on a screen can be in any number of states, such as black, white, or a combination of different values to make different colors. An image is usually a binary file stored in computer memory and rendered on the screen via a video graphics card. The larger the memory and faster the processor(s) are on the video graphics card, the faster the computer can draw pixels on the screen and the smoother video is. If the video card can draw images (i.e., video) faster than the computer can download the image data then the user will notice a stoppage in the video playback, this is why video is preloaded (i.e., buffered) into computer memory prior to starting video playback online.

The left portion of Figure 1 shows 1 pixel having the dimensions of 1 pixel (height) by 1 pixel (width) and having the color value of white. The right portion of Figure 1 shows 3 pixels with the dimensions of 1 pixel (height) and 3 pixels (width) and having the ordered (i.e., sequence) color values of white, grey, and black.

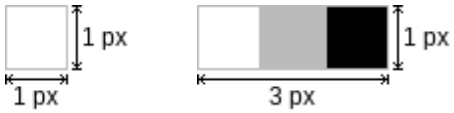

Figure 1. Defining pixels for use in this paper.

When an image is stored as a file, it contains the encoding format for the file, size of the image, the precise order of the pixels to be displayed, every color value for every pixel, and any other associated meta-information such as date and time of creation. If two images are aligned and have what should be the same pixels in the same location but actually have different color values then those pixels are different. Pixel sequences are what humans see as coherent images, and in this context, we can define color values of white as 0 and black as 1. The left portion in Figure 2 contains three pixels with the ordered sequence values of (0, 1, 0), and the middle portion contains the sequence values of (1, 0, 0). The right portion of Figure 2 contains the symmetric difference (i.e., disjunctive union, the difference of pixels) of the left and middle portions combined with the sequence values of (1, 1, 0).

Figure 2. Three separate images referred to as left, middle, and right portions.

In general, CADcompare uses image differencing to compare pixels and then displays the differences (i.e., the right portion of Figure 2) by highlighting them. There are essentially only two states when comparing the same pixel location in two different images, either the pixel value is exactly the same or it is different. The elegance in image differencing is being able to manage ranges of color values so that images can be the aligned by best fit when pixels have slight differences in color values. It is important to understand that CADcompare does not assign a grade

(e.g., letter or percentage) to the drawing, it only highlights differences between PDF file(s). CADcompare augments the grading process of technical and engineering CAD drawings by highlighting differences (i.e., differences in pixel color values) that can be easily missed.

Challenges Related to Comparing CAD Drawings

Universities often setup computer systems with software that allows system administrators to protect the core operating system and configuration files on a computer. In other words, each time the computer is restarted the computer goes back to a known configuration. If every university-owned computer is basically the same and students are instructed to reproduce an instructor given 2D CAD drawing(s), any differences found are probably incorrect or at least need to be reviewed by a human grader. The opinion of the authors is that an instructor can benefit by not having to grade the entire drawing since it is only the differences that may be incorrect. CADcompare highlights those differences so that the instructor is analyzing only what may be incorrect and can ignore the irrelevant parts of drawings. Thus, this should decrease grading times.

However, pixel-based comparisons of images have limitations, specifically when used as a grading tool in this context. The current web-based prototype will find any differences between a grading key and a student drawing. This solution works but is not robust enough for general release. For example, if a student has correctly reproduced a 2D engineering drawing, but scales the file incorrectly when creating the PDF, CADcompare will ignore the background of the drawing and will highlight all the other pixel differences without respect to scale. Additional actions that can produce false positives (i.e., incorrectly indicating that a particular condition or attribute is present) are when the PDF CAD drawing is rendered by choosing a software specific built-in PDF rendering engine compared to the Adobe® PDF rendering engine, or if a student chooses a different paper orientation (i.e., landscape versus portrait). In both instances, CADcompare would highlight any pixels that are different but do not accurately represent the correctness of a student drawing. These challenges often deal with geometric spatial transformations and classifications such as cropping, resizing (i.e., scaling), rotation, translation, and perspective distortion. Most can be easily accounted for by relying on the student to follow instructor directions, such as to rotate the drawing to the correct orientation and use the correct PDF rendering engine.

There are also inherent intended alterations related to 2D CAD drawings such as changes to geometric shapes (e.g., shading) and lines (e.g., type, width, and color) which introduce further image differencing challenges. When a CAD drawing is saved at a lower print quality to decrease file size (i.e., file compression), the pixel-based comparison can be complicated by unintended image artifacts which are essentially image noise near lines. To overcome the geometric spatial transformations and slight variances in pixel colors caused by compression advanced image processing and feature recognition often become needed.

The process of feature matching (i.e., finding unique features) in images to then perform the necessary geometric spatial transformations to allow for pixel-based comparison is known as image registration (see Figure 3). The authors believe that the challenges related to pixel-based comparisons of PDF files (i.e., images) specifically related to geometric spatial transformations and slight variances in pixel colors can be overcome with the use of an image processing application programming interface (API) such as MATLAB® software (The MathWorks Incorporated, n.d.). CAD drawings usually have different title blocks related to the student's unique name, date, and drawing title, these variances should be able to be overcome by allowing the instructor to highlight certain areas to be ignored such as the results generated in the Appendix. Another possibility is to allow the instructor to select areas (i.e., portions of a drawing) to then be geometrically transformed and then best fit, such as ignoring line thickness to account for incorrect line scaling.

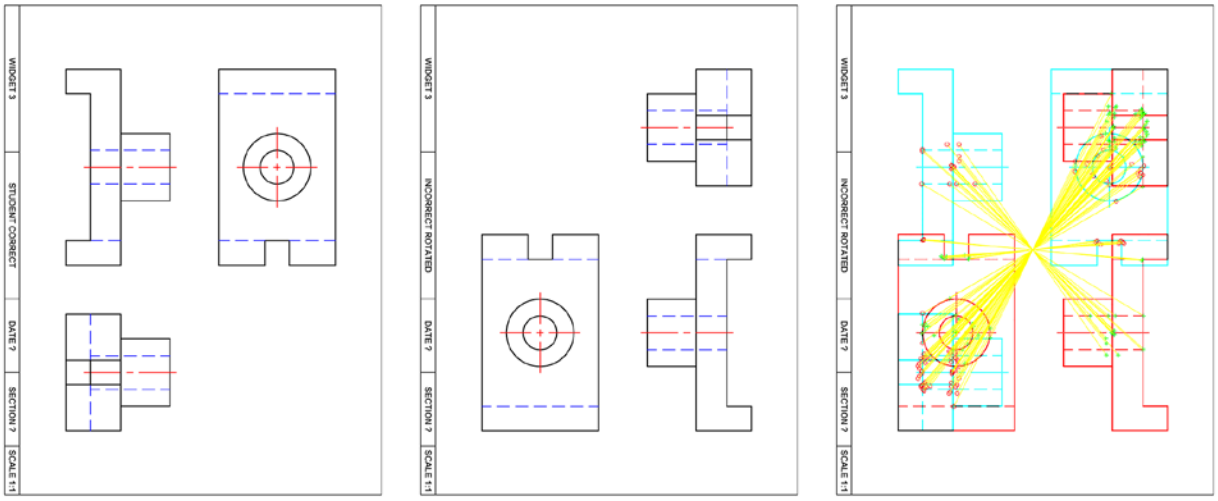

Figure 3. Grading key (left), Student Submission (middle), and the Results of MATLAB Matched Features from Image Registration (right).

Market Analysis

Based on revenue position rankings amongst the global 100 software leaders, the authors consider Dassault Systèmes, Siemens, Autodesk, and PTC as the global leaders in CAD software (PwC, & International Data Corp, 2014). Based on application and field of work, all four companies have various CAD software packages used in industry and taught in secondary and postsecondary education. Each also sells software that can create 2D engineering drawings and render those drawings as PDF files. Some CAD software has built-in comparison tools, however, none of the comparison tools accept PDF files to compare, are web-based applications, or can compare multiple student files at once like CADcompare can.

SOLIDWORKS®, which is sold by Dassault Systèmes® has the Compare Utility™ (Dassault Systèmes, n.d.). The tool is built into SOLIDWORKS and allows a user to compare the properties of two SOLIDWORKS documents or two configurations of the same SOLIDWORKS document. The tool has the capability to compare features and parameters in solid part files, volume in assembly files, and bill of material (BOM) tables in assembly and drawing files. However, the main limitations to the SOLIDWORKS Compare Utility are that it can only compare two files at once, which equates to one grading key and one student submission.

NX®, which is sold by Siemens® has the model compare and drawing compare tools (Siemens Product Lifecycle Management Software Incorporated, 2017a, 2017b). The model compare tool is similar to that of the SOLIDWORKS Compare Utility, in that the geometries of bodies in two related or unrelated parts can be compared to the features and expressions. The drawing compare tool allows a user to track, compare, and analyze the current state of a drawing with a previously saved state, another revision, or another drawing. Again the main limitation of NX compare tools is that it does not accept PDF files and cannot compare one grading key file to multiple student files simultaneously.

AutoCAD®, which is sold by Autodesk® has a plug-in titled DWG Compare™ (Autodesk Incorporated, n.d.). The tool is based on the drawing compare feature in AutoCAD Architecture® and compares two versions of a drawing. The comparison differences are temporarily displayed on the original version drawing. During the comparison process, the original drawing is opened and is overlaid with representations of geometry and differences from the comparison drawing. Any changes a user makes, while DWG Compare is active, are made only to the original version of the drawing (i.e., cannot merge the drawing versions). DWG Compare is also not an integration tool and if necessary the user must manually integrate the differences between the original and compare versions of the drawing.

Creo®, which is sold by PTC® has a tool to compare two part files or two different versions of a part file and obtain a graphical report on the differences between the two parts or versions (PTC, n.d.). A user can compare features and geometry of the part files and measure the geometric deviation between each. Two components in the assembly, cabling, piping, or welding modes can also be compared. The main limitations of Creo compare tool are that it does not accept PDF files and cannot compare one grading key file to multiple student files simultaneously.

Acrobat®, which is sold by Adobe® has the compare files tool, which is designed to find changes between two versions of a PDF file (Adobe Systems Incorporated, 2017). Acrobat analyzes the files and presents a report detailing differences between the documents. The information in PDF files is encoded in a data encapsulation method known as objects which define a state and behavior to represent components of the document such as fonts, pages, and sampled images (Adobe Systems Incorporated, 2008). However, the main limitation of the Acrobat compare files tool is

that any small change in the image object is represented as a change to the entire image object and does not highlight the small change but the entire image object.

CADcompare

CADcompare does a pixel-based comparison of PDF files, which nearly every CAD software program can generate. Thus far in this work in progress, the authors have created a Windows-based proof of concept (Spring 2017), a web-based prototype (under-development), and branding elements (see Figure 4, and Figure 5).

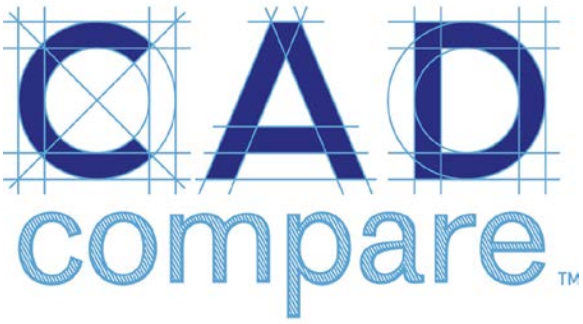

Figure 4. CADcompare Logo.

Figure 5. CADcompare Icon.

Proof of Concept

To use the proof of concept version of CADcompare, an instructor needs to install the software on a Windows computer by downloading a ZIP file and extracting the contents. Figure 6 shows the folder directory that is created after folder extraction. The next step requires the instructor to place a student PDF file(s) and the grading key PDF file into their respective folders (i.e., "studentpdfs" and "teacherkey") (see Figure 6). To run CADcompare the instructor would need to double-click the "CADCompare.cmd" batch file. Figure 6 shows the text-based window that is shown to the instructor during the multiple file comparison. The instructor will then find new CADcompare generated PDF file(s) (see Figure 7) in the "results" folder which highlights any differences found between the students' and instructor PDF file(s). Admittedly the proof of concept is not very user-friendly, has minor bugs, and is not aesthetically appealing, however; it demonstrated that pixel-based comparison of PDF CAD drawings was possible and fast at less than one second per student file.

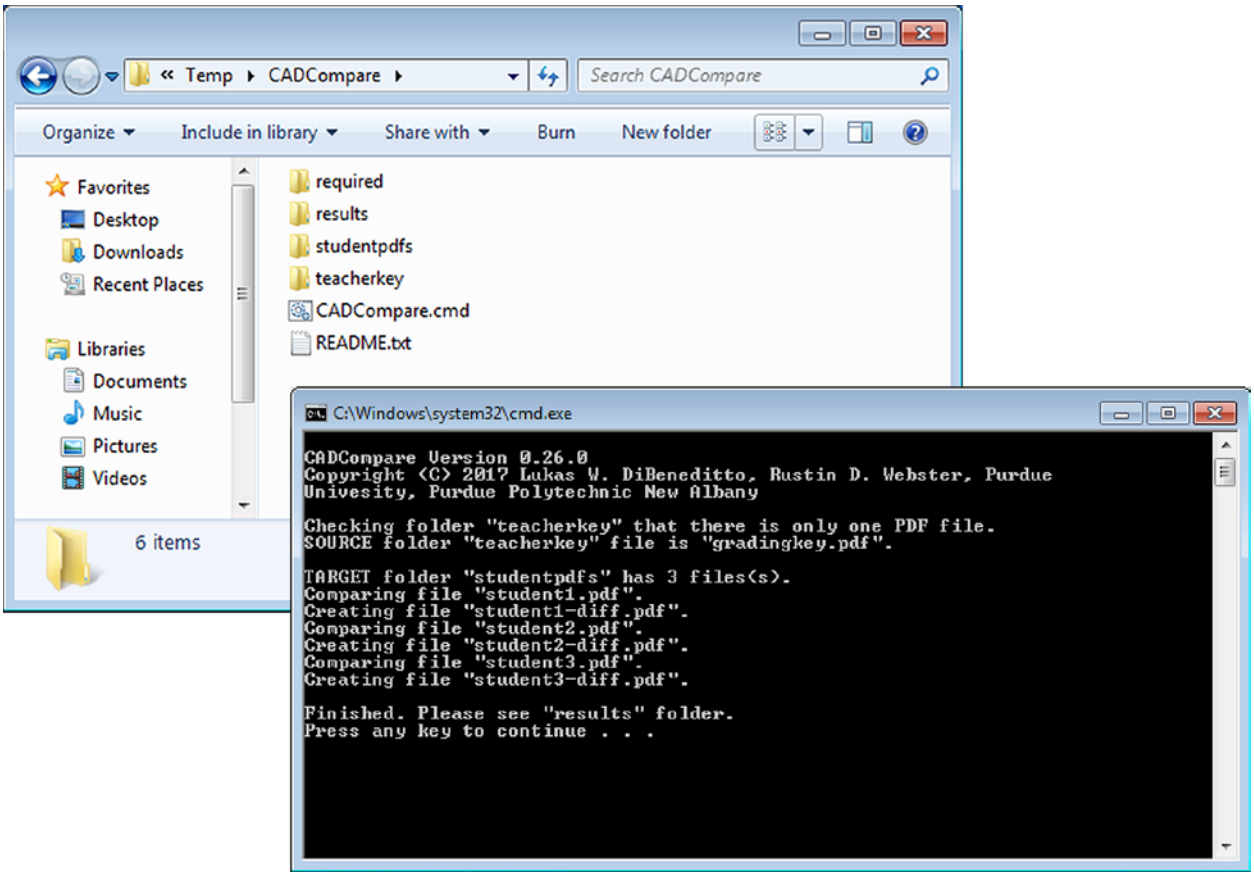

Figure 6. Extracted ZIP file contents (left) and Proof of Concept Running (right).

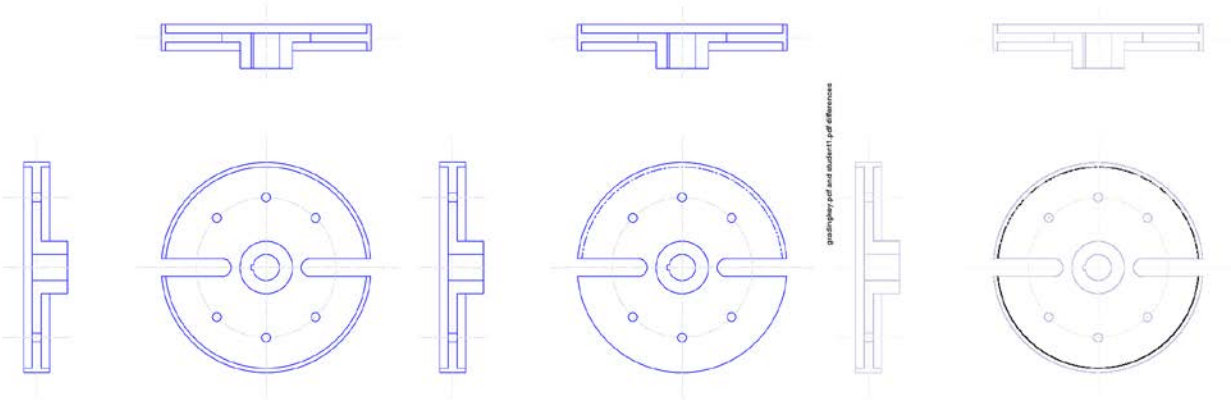

Figure 7. Grading key (left), Student Submission (middle), and the Results of the Windows CADcompare Proof of Concept (right).

The proof of concept works by running a Windows batch file program, which is a Windows-based scripting language. The batch file program checks for a grading key and student files then iterates over each student file by running an ImageMagick convert and compare program. ImageMagick is an open source image processing library. ImageMagick compares the PDF file(s) at 300 dots per inch (dpi) resolution and overlays the highlighted differences on top of the original student version before making a new CADcompare PDF and placing it in the results folder.

The limitations of this proof of concept were that an instructor would need to be using a Windows operating system, a ZIP file has to be downloaded and extracted for the "install", it is not relatively user-friendly (i.e., text-based), the PDF files need to be the same orientation and same document size, does not provide for any image registration, nor detailed instructor interaction. Detailed testing and debugging did not take place since in the opinion of the authors the Windows-based version showed enough promise to initiate development of the web-based version.

Web-based Prototype

To use the current web-based version an instructor uses a web browser to visit the webpage https://www.purdue.edu/apps/cadcompare/ and logs into access CADcompare. The instructor is prompted with an option to upload the grading key PDF file and a student PDF file(s). The software compares the PDF files at 300 dpi and displays the highlighted differences on the web page, and provides the option to toggle between the grading key, the student file, and the CADcompare PDF (see Appendix for test cases).

The web application is currently being hosted on a Linux® based Apache® web server at Purdue University and it uses custom code written in HTML®, CSS®, JavaScript®, and PHP®, which are programming languages for web development, and ImageMagick for the primary image processing library.

CADcompare has numerous security measures, such as all work sessions are managed, uploaded files are locked and deleted when the instructor leaves the webpage, and file upload security to prevent client-side code injection attacks, wherein an attacker may execute malicious scripts on the web server. Sandboxing and virus scanning was requested by the authors however it was declined by Information Technology at Purdue (ITaP) since in its current form only authorized users can access the web application. It is also anticipated that a thorough security review will need to be done before it is released to the public.

Future iterations of CADcompare should eliminate the following known limitations:

- The user must log in to a website.
- Any characters that are not letters, numbers, a hyphen-minus, underscore, or space that are in the PDF filenames are replaced with underscores (i.e., for security).
- Comments and annotations are stripped from results PDF file(s).
- Student files need to be the same size and orientation as the grading key.
- Geometric spatial transformations resulting in false positives.

Discussion

Throughout the development process, there were a number of challenging issues specifically related to decreasing the number of test cases resulting in false positives (i.e., pixels highlighted even though the student has a correct drawing). To overcome the challenges related to the pixel-based comparison as described in this paper, the next iteration of CADcompare will integrate advanced image processing tools and capabilities. The authors believe that the challenges of rotation, scaling, and translation should also be overcome with feature recognition, which should significantly increase the value of CADcompare.

A more user-friendly and intuitive web interface will also be developed. Built-in selection and manipulation tools will hopefully allow an instructor to highlight particular areas of the grading key, which will prompt CADcompare to ignore non-highlighted areas when running (i.e., comparing PDF files). The title block could be an important area to ignore because each could have a different student name, date, etc. The instructor could also assign a standard three view drawing but only want to grade the front view. Thus, there is no need for CADcompare to include the top or side view when comparing PDF files (i.e., images).

Significant coordination between the authors and multiple departments at Purdue University West Lafayette Main Campus was needed to develop the current web-based version of CADcompare, to gain access to, and safeguard the security of the web server. During development, the web server and the development environment were both upgraded unannounced, which caused unintended delays. There are multiple advantages in using Purdue University servers for prototype development, such as lower cost to operate and safety measures built-in. However, the authors intend to move CADcompare to an external server (https://www.cadcompare.com/) when ready for public release. In anticipation, the developers applied for and won a small Purdue University Statewide Commercialization & Research Grant in the fall of 2018. The funds can be used for continued marketing and publicity efforts, external web hosting, domain acquisition, and legal fees, such as trademark filing.

Finally, live testing of CADcompare will begin in the fall of 2018. Instructor(s) of record for CGT 11000 Technical Graphics Communications at Purdue University, Purdue Polytechnic New

Albany will use CADcompare at four assessment points. Three are homework assignments, which consist of a total of 16 dimensioned single view drawings and two dimensioned multiview drawings (i.e., front and right side). The fourth assessment point is an exam that consists of two dimensioned single view drawings and one dimensioned multiview drawing (i.e., standard three views). Students will use AutoCAD to recreate each 2D drawing with the correct shapes, size, line types, line colors, and line precedence. Students will not be required to annotate the drawings (i.e., dimension and tolerance). The student will then render the AutoCAD drawing as a PDF file using instructor-given settings and uploaded to a Learning Management System (LMS).

The instructor(s) will grade each assessment point in one sitting (i.e., four total grading sessions) and record elapsed time for each. Timing will begin when the instructor has logged into the LMS and end when the last student submission is graded. To explore how CADcompare can augment instructor grading to decrease grading times of 2D PDF CAD drawings, the grading times for each session will be compared to grading times taken in fall 2017, which did not include CADcompare (i.e., human-grader only). Table 1 shows data from fall 2017.

The research study results will be disseminated along with a detailed literature review on the use of grading tools for engineering drawings. At the time of this writing, the authors found limited published literature on the topic (Goh, Mohd, & Mano, 2013, Sanna, Lamberti, Paravati, & Demartini, 2012).

Table 1. Fall 2017 Grading Session Times

Assessments	***n***	**Time (hr:min:sec)**	**M (hr:min:sec)**	**SD (hr:min:sec)**
Assignment 7. Widgets 1-6				
1. Instructor A	25	0:30:07	0:24:48	0:07:32
2. Instructor B	13	0:19:28		
Assignment 8. Widgets 7-15				
1. Instructor A	25	0:37:50	0:33:44	0:05:49
2. Instructor B	14	0:29:37		
Assignment 9. Widgets 16-18				
1. Instructor A	24	0:58:25	0:43:38	0:20:54
2. Instructor B	14	0:28:52		
Exam 2. Widgets 1-3				
1. Instructor A	26	0:57:50	0:50:17	0:10:41
2. Instructor B	14	0:42:43		

Conclusion

The web-based prototype, while still in development, shows that pixel-based comparison of a PDF grading key to student PDF CAD drawing(s) can produce a desirable outcome and automate the grading process of multiple student drawings simultaneously. CADcompare augments the grading process of technical and engineering CAD drawings by highlighting differences that can be easily missed by a human grader, such as incorrect line type(s), color(s), or double lines (i.e., lines on top of each other). The intended use of CADcompare is to act as a tool to help instructors grade faster, more accurately, and without unintended bias. Once development is complete a research study will take place to explore these intended advantages of CADcompare. The reader should be reminded that CADcompare does not assign a grade (e.g., letter or percentage) to the student drawing, it only highlights differences between PDF file(s). Thus, the instructor is still required to view and analyze the CADcompare generated PDF differences that have been highlighted and assign the external grade.

References

Adobe Systems Incorporated. (2008). *Document management — Portable document format — Part 1: PDF 1.7*. Retrieved from https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf

Adobe Systems Incorporated. (2017). *Compare two versions of a PDF file (Acrobat Pro).* Retrieved from https://helpx.adobe.com/acrobat/using/compare-documents.html

Autodesk Incorporated. (n.d.). *DWG Compare.* Retrieved from https://apps.autodesk.com/ACD/en/Detail/HelpDoc?appId=1050435430275763742

Dassault Systèmes. (n.d.). *Running the compare utility*. Retrieved from http://help.solidworks.com/2018/english/SolidWorks/sldworks/t_running_the_compare_utility.htm

Goh, K. N., Mohd, S. R., & Mano, R. B. (2013). Automatic Assessment for Engineering Drawing. In: Zaman H.B., Robinson P., Olivier P., Shih T.K., Velastin S. (eds) Advances in Visual Informatics. *Lecture Notes in Computer Science, 8237*(IVIC 2013). doi:10.1007/978-3-319-02958-0_45

PTC. (n.d.). *About Part Comparison*. Retrieved from http://support.ptc.com/help/creo/creo_pma/usascii/#page/fundamentals%2Ffundamentals%2Ffund_five_sub%2FAbout_Part_Comparison.html

PwC, & International Data Corp. (2014). *Global 100 Software Leaders by Revenue*. Retrieved from http://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/explore-the-data.html

Sanna, A., Lamberti, F., Paravati, G., & Demartini, C. (2012). Automatic assessment of 3D modeling exams. *IEEE Transactions on Learning Technologies, 5*(1), 2-10. doi:10.1109/TLT.2011.4

Siemens Product Lifecycle Management Software Incorporated. (2017a). *Drawing Compare*. Retrieved from https://docs.plm.automation.siemens.com/tdoc/nx/12/nx_help/#uid:xid1096083

Siemens Product Lifecycle Management Software Incorporated. (2017b). *Model Compare*. Retrieved from https://docs.plm.automation.siemens.com/tdoc/nx/12/nx_help/#uid:genid_model_compare_424_12589

Smith, A. (n.d.). *A Pixel Is Not A Little Square, A Pixel Is Not A Little Square, A Pixel Is Not A Little Square! (And a Voxel is Not a Little Cube). Technical Memo 6.* (Microsoft) Retrieved from http://alvyray.com/Memos/CG/Microsoft/6_pixel.pdf

The MathWorks Incorporated. (n.d.). *Find image rotation and scale using automated feature matching*. Retrieved from https://www.mathworks.com/help/vision/examples/find-image-rotation-and-scale-using-automated-feature-matching.html

Appendix

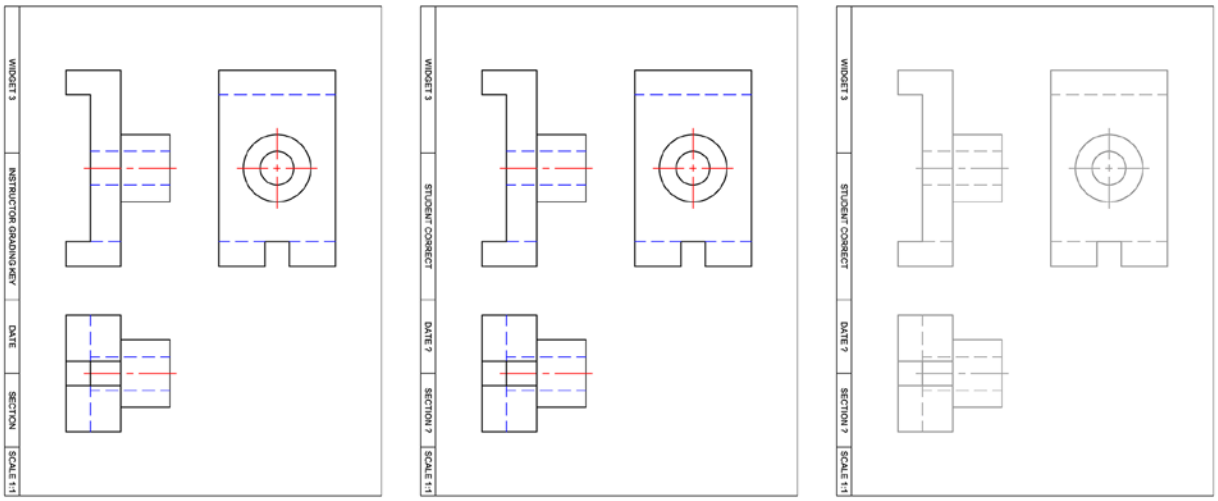

Figure 8. Instructor Grading Key (left), Student Submission (middle), and Results: Correct – No Errors (right).

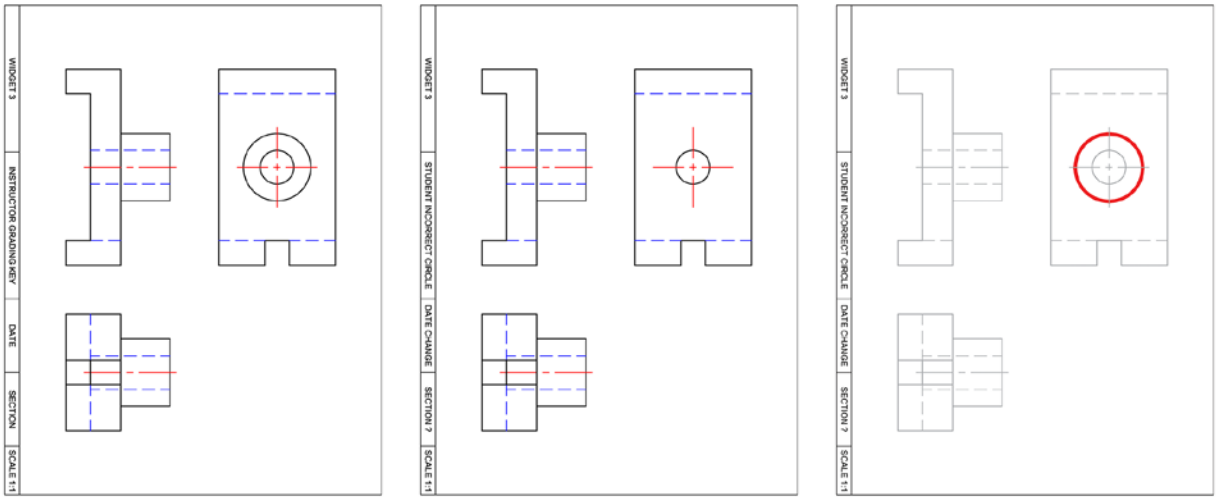

Figure 9. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Geometry (right).

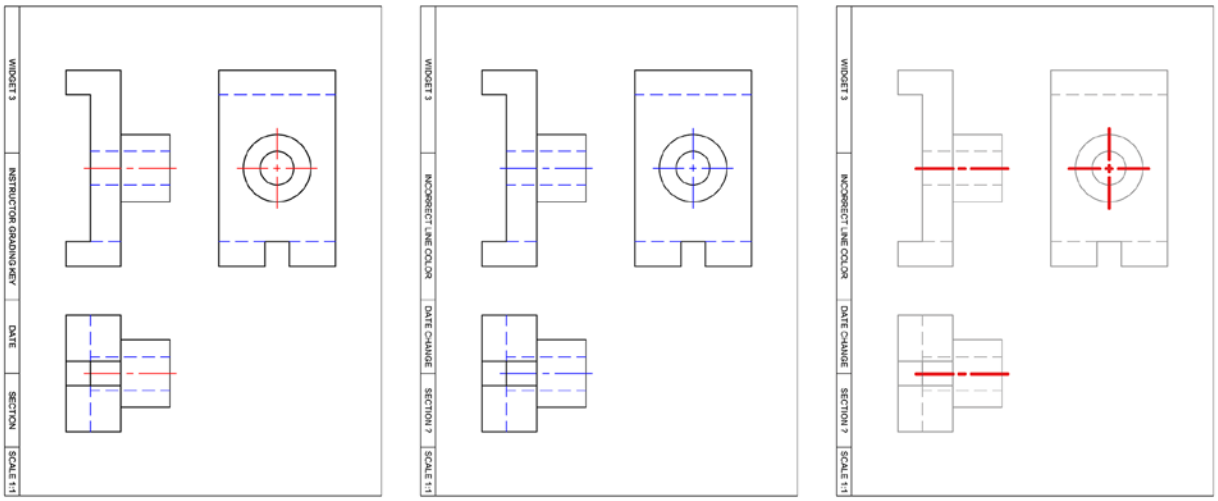

Figure 10. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Color (right).

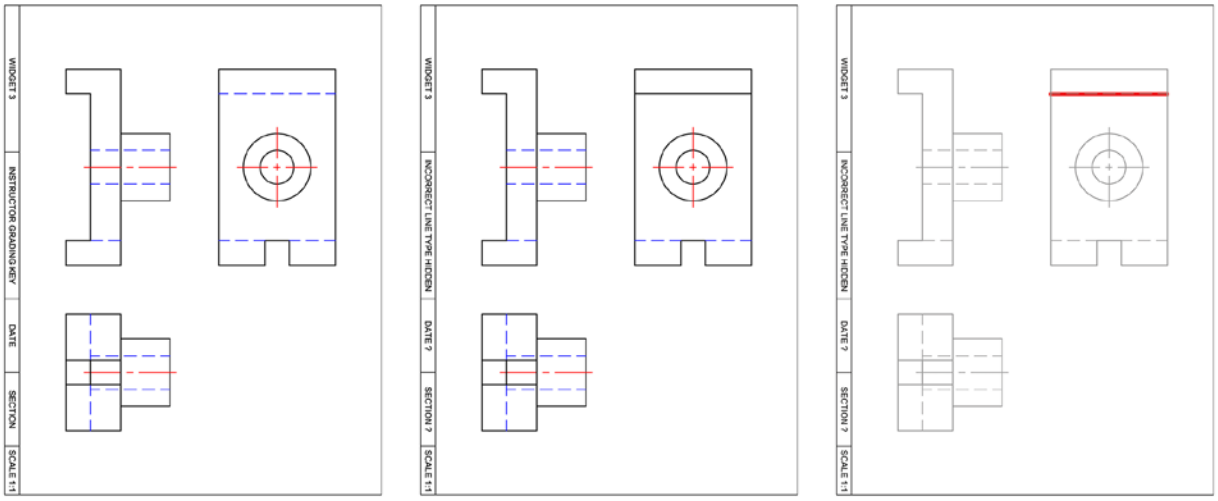

Figure 11. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Line Type (right).

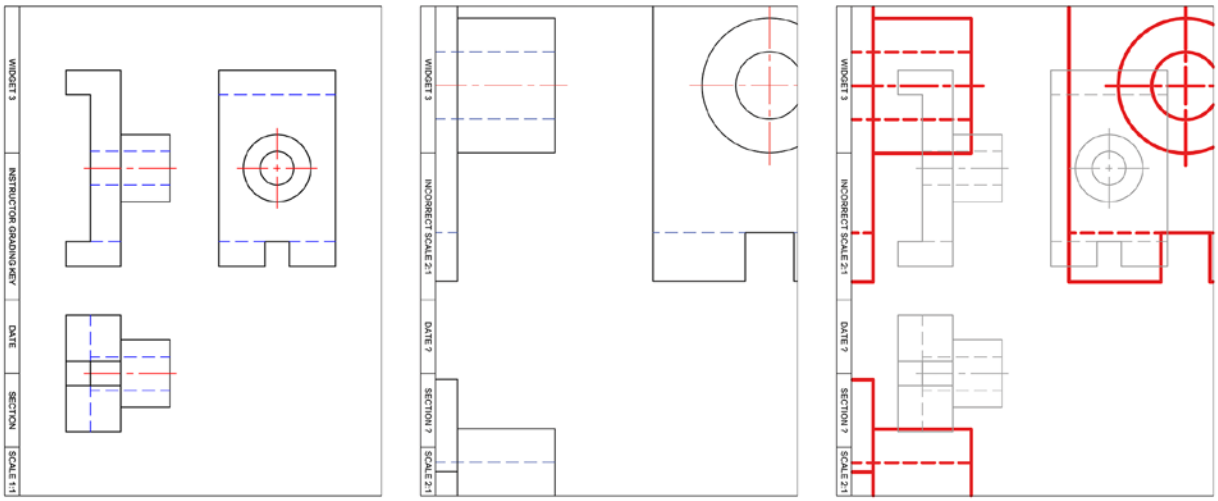

Figure 12. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Scale (right).

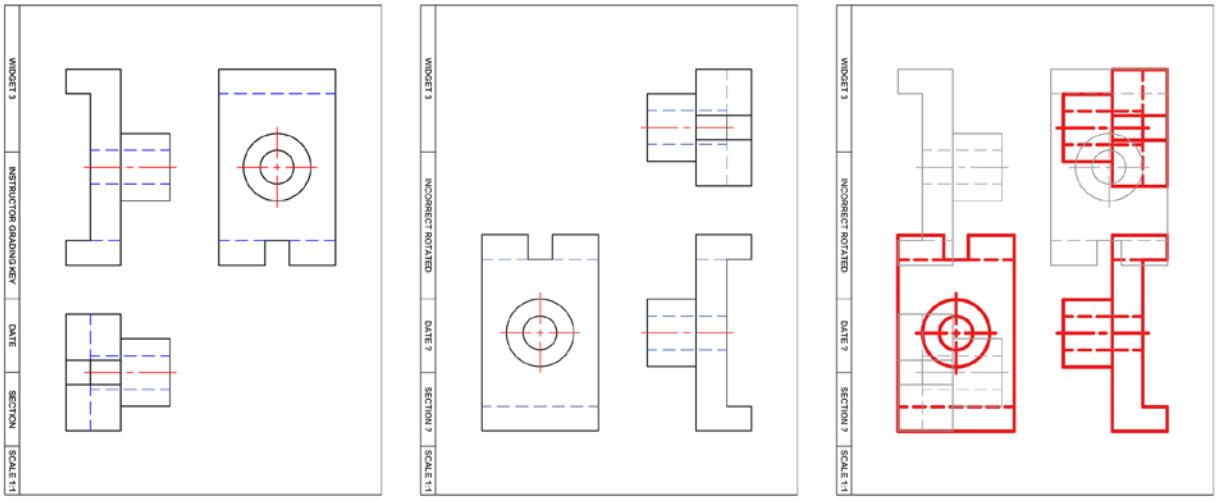

Figure 13. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Rotation (right).

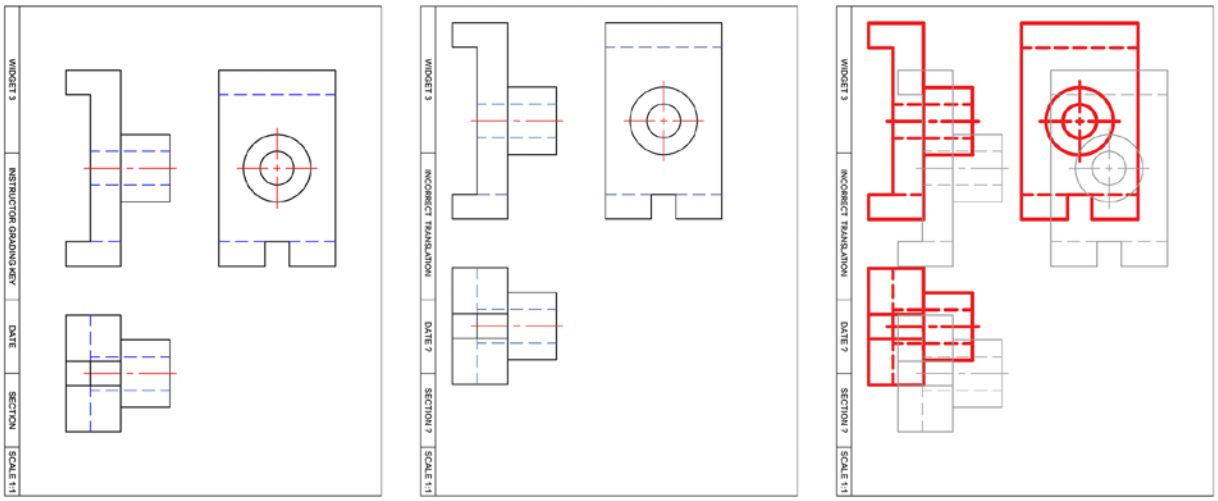

Figure 14. Instructor Grading Key (left), Student Submission (middle), and Results: Incorrect – Translation (right).