Algorithmic Grading Strategies for Computerized Drawing Assessments

Dr. Mariana Silva, University of Illinois, Urbana-Champaign

Mariana Silva is an Adjunct Assistant Professor and Curriculum Development Coordinator in the Mechanical Science and Engineering Department at the University of Illinois at Urbana-Champaign. She received her BSME and MSME from the Federal University of Rio de Janeiro, Brazil and earned her Ph.D. in Theoretical and Applied Mechanics from the University of Illinois at Urbana-Champaign in 2009. Besides her teaching activities, Mariana serves as an academic advisor in the Mechanical Science and Engineering department.

Prof. Matthew West, University of Illinois, Urbana-Champaign

Matthew West is an Associate Professor in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign. Prior to joining Illinois he was on the faculties of the Department of Aeronautics and Astronautics at Stanford University and the Department of Mathematics at the University of California, Davis. Prof. West holds a Ph.D. in Control and Dynamical Systems from the California Institute of Technology and a B.Sc. in Pure and Applied Mathematics from the University of Western Australia. His research is in the field of scientific computing and numerical analysis, where he works on computational algorithms for simulating complex stochastic systems such as atmospheric aerosols and feedback control. Prof. West is the recipient of the NSF CAREER award and is a University of Illinois Distinguished Teacher-Scholar and College of Engineering Education Innovation Fellow.

Algorithmic grading strategies for computerized drawing assessments

1. Introduction

Introductory mechanics courses have important learning objectives focusing on students' ability to accurately draw or sketch particular types of diagrams, such as free body diagrams and graphs of shear forces and bending moments in beams. To achieve mastery of these drawing skills it is essential that students have many opportunities to practice and that they receive rapid and accurate feedback on whether they are drawing the correct diagram for a given mechanical problem. With the growing student enrollment in many engineering programs, however, it becomes increasingly difficult to provide prompt and accurate grading using the traditional approach of having students submit hand-drawn diagrams which are graded by a teaching assistant or grader. One way to circumvent this overwhelming grading process in large classes is to adopt multiple-choice questions. Unfortunately, assessing drawing skills using a multiple-choice instrument is mostly limited to testing students' interpretation of given drawings, rather than testing their ability to construct new drawings themselves. For example, research was conducted to investigate the validity of the use of multiple-choice questions to assess graphing abilities[2]. A group of students were asked to select the graph that best represented a situation, while a second group of students had to construct the graph that best represented the same situation. The results indicated that students obtained lower performance when responding to multiple-choice questions and concluded that the use of multiple-choice instruments to assess graphing abilities may be invalid.

Another appealing alternative to hand grading is the use of automated computer-based systems, where students need to draw a diagram or graph on the computer and this can then be immediately graded algorithmically and feedback returned. Computer-based homework systems have been widely adopted in large introductory STEM courses in recent years[5], due to benefits to both students and instructors, such as immediate feedback, integration with online content, and reduced grading workloads. The development of computer-based questions with automatic grading allows instructors to implement more frequent testing in their classes. Educational research indicates that frequent testing leads to better retention than rehearsal strategies such as rereading noted or previously solved problems[9;10], especially when immediate feedback is provided[4;7] In addition to better retention, students require repeated practice in order to achieve mastery of a given skill[1]. Specifically, the construction of free-body diagrams that are helpful and accurate takes time and practice, and for that reason the need for computer-based drawing tools is of utmost importance.

Roselli et al.[11] developed an online free body diagram assistant that allows students to construct these drawings by inserting forces and moments using the mouse, and the ability to receive immediate feedback. Commercial online systems such as McGraw-Hill Connect[6] and Pearson MasteringEngineering[8] have also developed graphing questions in which students need to draw graphs, such as a free-body diagram, using the mouse to insert objects. Unfortunately, these systems do not provide much feedback on the drawing features. Moreover, they mostly have one answer implemented as the correct one, and therefore other possible variations of the correct answer are often marked as incorrect. Another computer-based system that allows drawing of Statics problems is Mechanix [12], a hand-sketch homework system that uses sketch-recognition to grade the drawings and gives instant feedback to the users.

The central question addressed by this paper is as follows. When receiving a computer-drawn mechanical diagram or sketch from a student, what algorithmic procedure should be used to grade the submission? In Section 2 we present a list of general requirements that should be satisfied by any algorithm for grading online drawing problems, and then in Section 3 we present a specific algorithm that we show in Section 4 satisfies the requirements. Results from implementation in a 180-student introductory mechanics course at the University of Illinois at Urbana-Champaign in Fall 2016 as part of a wider reform [13] are presented in Section 5, and Section 6 gives conclusions.

2. Functionality requirements for grading algorithms

We identify five key functionality requirements for a grading algorithm for engineering drawings. The algorithm should:

R1. be able to provide students with meaningful feedback about errors in their diagram,

R2. be easy to understand for problem authors, and require only data which is readily available to authors,

R3. be adaptable to different types of drawings or sketches,

R4. be fast to execute,

R5. be robust to unexpected or unusual inputs.

Requirement R1 stems from the need to support the learning goals articulated in Section 1, R2 is needed to be practical in higher-education settings where the course content is often highly customized by individual instructors and cannot be centrally generated, R3 permits a single platfrom to support different engineering courses and thus leverages expertise, and R4 and R5 are needed for practical deployment in large-enrollment university courses (200 or more students).

3. A specific grading algorithm implementation

The drawing tool developed for this study allows users to insert and manipulate objects in the drawing space (canvas), in order to construct the solution to a given problem. Figure 1 illustrates a typical mechanics problem, where students need to complete a free-body diagram, and draw the corresponding shear and bending moment diagrams. For the free-body diagram, we created objects to represent concentrated forces, distributed forces and moments; for the shear and bending moment diagrams, we created two different sketching lines to represent linear and higher order curves. Figure 2 depicts these available drawing objects, and the names associated with each one of them, which are used by the grading algorithm.

The grading algorithm compares the position of objects that are placed in the canvas with respect

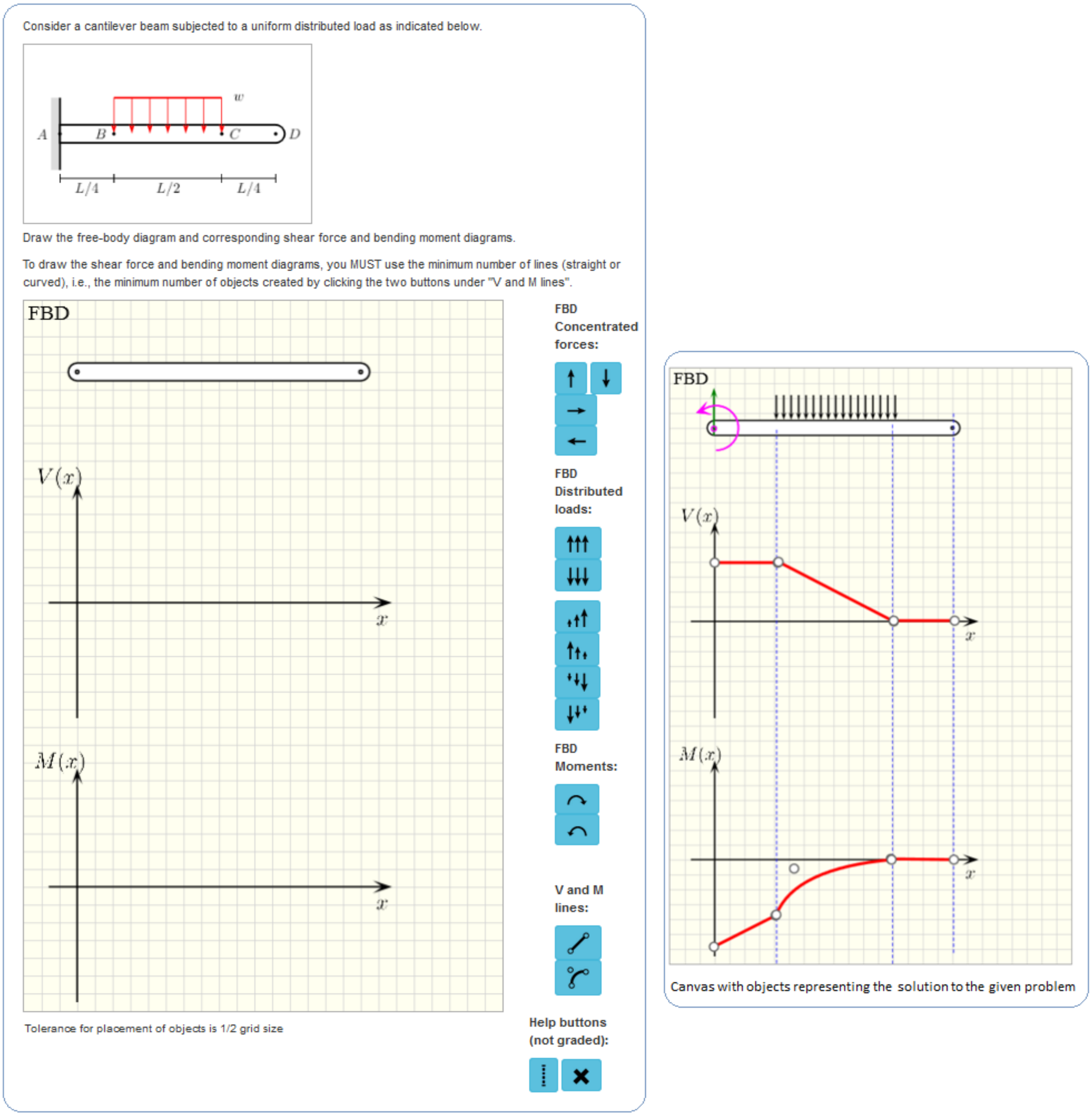

Figure 1: Left: example question that uses the drawing tool. Users need to complete a free-body diagram and shear and bending moment diagrams. Right: sketch of the solution using the drawing tool.

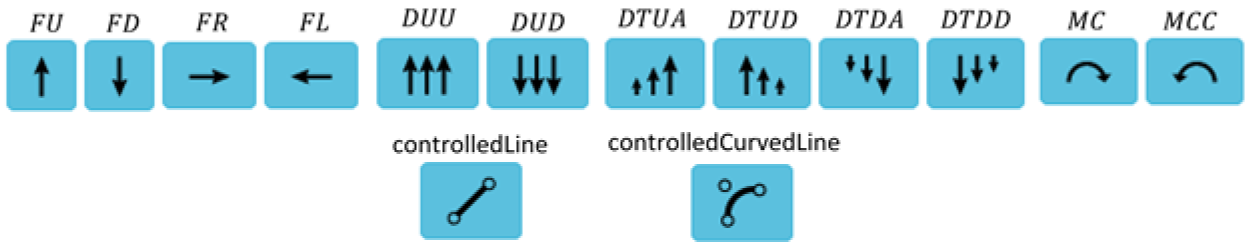

Figure 2: List of all possible objects and their corresponding names used internally by the algorithm.

to the expected position of each object. The list of objects inserted in the canvas by the user are denoted as *submitted objects* and the list of objects corresponding to the correct solution are denoted as *required objects*. The algorithm generates the drawing space from a set of pre-defined variables, initialized by the problem author, as illustrated in Fig. 3. These variables are used to determine the position of the required objects.

The following steps are necessary to write the grading function for each question:

1. Create a list of required and optional objects. For the example illustrated in Fig. 1, we have:
   (a) requiredObjects $= [R_{Ay}, M_A, w_{BC}, V_{AB}, V_{BC}, V_{CD}, M_{AB}, M_{BC}, M_{CD}]$
   (b) optionalObjects $= [R_{Ax}]$
2. Determine the following properties for all required and optional objects:
   - Upper bound (yUp), lower bound (yLower), right bound (xUp) and left bound (xLower). Note that these four properties together define the bounding boxes where the object should lie in order to be marked as correct (purple dashed lines in Fig. 4).
   - Name: identifies the name of the object. Figure 2 indicates the objects used in this study, and their corresponding names. Sometimes a required or optional object can be represented by more than one configuration, and therefore they should have more than one name associated to it. For example, the vertical force at A, R_{Ay}, can be represented by a right or left arrow in a free-body diagram (the direction is determined only after performing numerical calculations). Therefore, R_{Ay} has names `FD` and `FU` (or name = `["FD","FU"]`).
   - Feedback name: this is the string name that is used for the feedback messages
   - Id: unique identifier for each object
   - Found: all objects are initialized with found = false

   The shear and bending moment diagram sketching tools (controlledLine and controlledCurvedLine) have one additional property denoted *slope* that can be set to “zero”, “positive” or “negative”. For example, the shear force $V(x)$ from A to B corresponds to a straight line with zero slope.

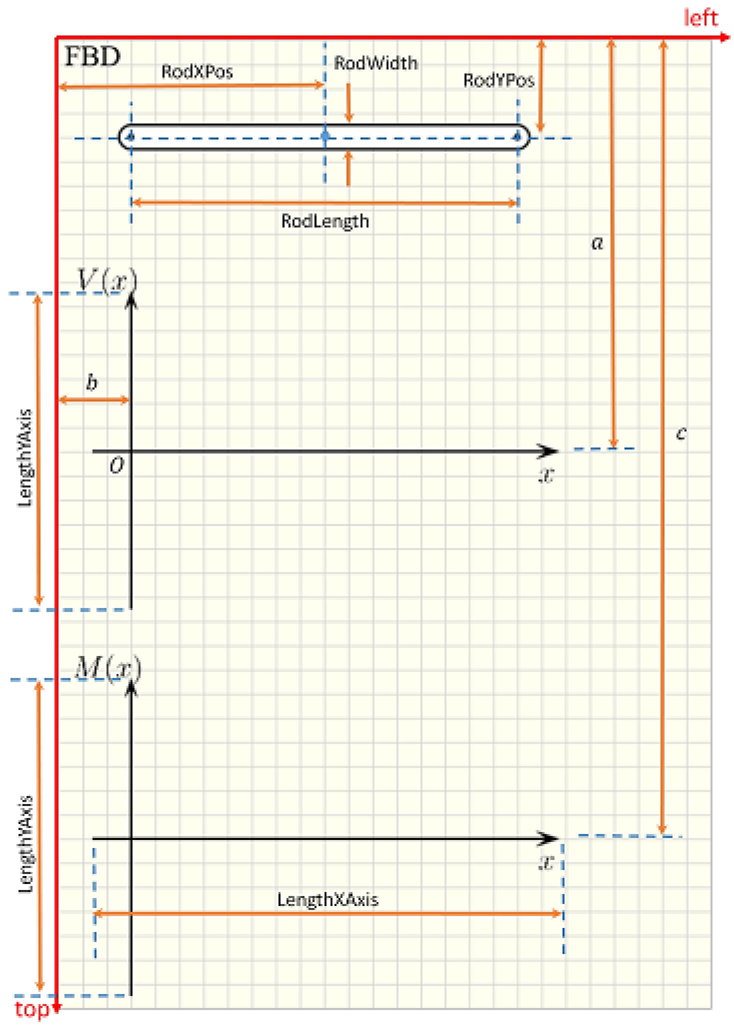

Figure 3: Canvas listing the pre-defined variables used to determine the position of the required objects.

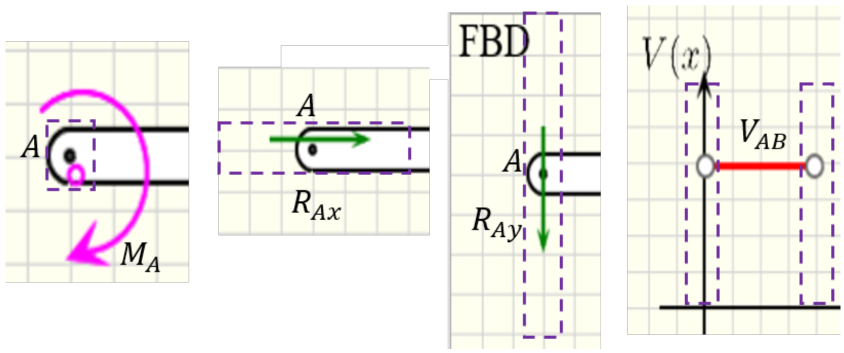

Figure 4: Example of bounding boxes that define the location of the required and optional objects. The center of the moment M_A should be located inside a square with center at A, located at position (RodXPos,RodYPos), and side equal to grid-size. The reaction forces R_{Ay} and R_{Ax} should be located inside the larger dashed purple lines, allowing the arrows to start or end at point A. For the sketch of the shear force, the controlledLine object should have zero slope and its end points must lie inside the dashed box.

3. Run function to process grading: Loop over all submitted objects trying to find matching required or optional objects within the tolerance described by the bounding boxes. The pseudo-code for this function is presented in Algorithm 1. The functions to check the tolerance will depend on the type of object. For illustration purposes, we also present the pseudo-code for the function that checks the tolerance of free-body diagrams (Algorithm 2).

4. Evaluation of algorithm functionality requirements

To evaluate the algorithm described in Section 3, we now evaluate it against the five key functionality requirements described in Section 2.

Requirement R1: Instant feedback. The grading algorithm provides feedback to students, indicating when they have missing parts of the diagrams, define incorrect slopes, or use the incorrect sketching tools. As shown in Section 5, better feedback message and algorithm could be implemented in the future. Note from Algorithm 1 that a question is marked as correct if all the required objects have property "found == true" and there is no extra object inserted by the user. Optional objects are not considered extra objects. If the algorithm finds extra objects, it gives the feedback message "Found extra object". If the algorithm cannot find a required object, it will give the feedback message "Object [feedbackName] was not found".

Requirement R2: Simple to understand for authors. The process grading function 1 was implemented in a generic form and does not depend on the given problem. Problem authors need to provide only data corresponding to the expected position of the required and optional objects, which will be different for each question, and is simple to understand for authors within the context of a given question.

Requirement R3: Adaptability to different drawing types. Because of its general framework,

Algorithm 1 Process grading function

```
m = number of submittedObjects
n = number of requiredObjects
for i = 1 to m do
  for j = 1 to n do
    if submittedObjects[i].name == requiredObjects[j].name then
      if requiredObjects[j].name == "controlledLine" then
        call checkToleranceControlledLines(submittedObjects[i], requiredObjects[j])
        if submittedObjects[i].found = true then
          break loop j
        end if
      end if
      if requiredObjects[j].name == "controlledCurvedLine" then
        call checkToleranceControlledCurvedLines(submittedObjects[i], requiredObjects[j])
        if SubmittedObjects[i].found == true then
          break loop j
        end if
      end if
      if requiredObjects[j].name is one of the FBD object names then
        call checkToleranceFBD(submittedObjects[i], requiredObjects[j])
        if submittedObjects[i].found == true then
          break loop j
        end if
      end if
    end if
  end for
end for
if requiredObjects[j].found == true for all j and submittedObjects[i].found == true for all i then
  question is marked as correct
end if
for each i where submittedObjects[i].found == false do
  question is marked as incorrect
  print feedback message "Found extra object."
end for
for each j where requiredObjects[j].found == false do
  question is marked as incorrect
  print feedback message "The object [feedbackName] was not found."
end for
```

Algorithm 2 Check tolerance for FBD objects

```
function checkToleranceFBD(submittedObject, requiredObject)
if requiredObject.xLower < submittedObject.xpos < requiredObject.xUp then
  if requiredObject.yLower < submittedObject.ypos < requiredObject.yUp then
    submittedObject.found = true
    requiredObject.found = true
  end if
end if
```

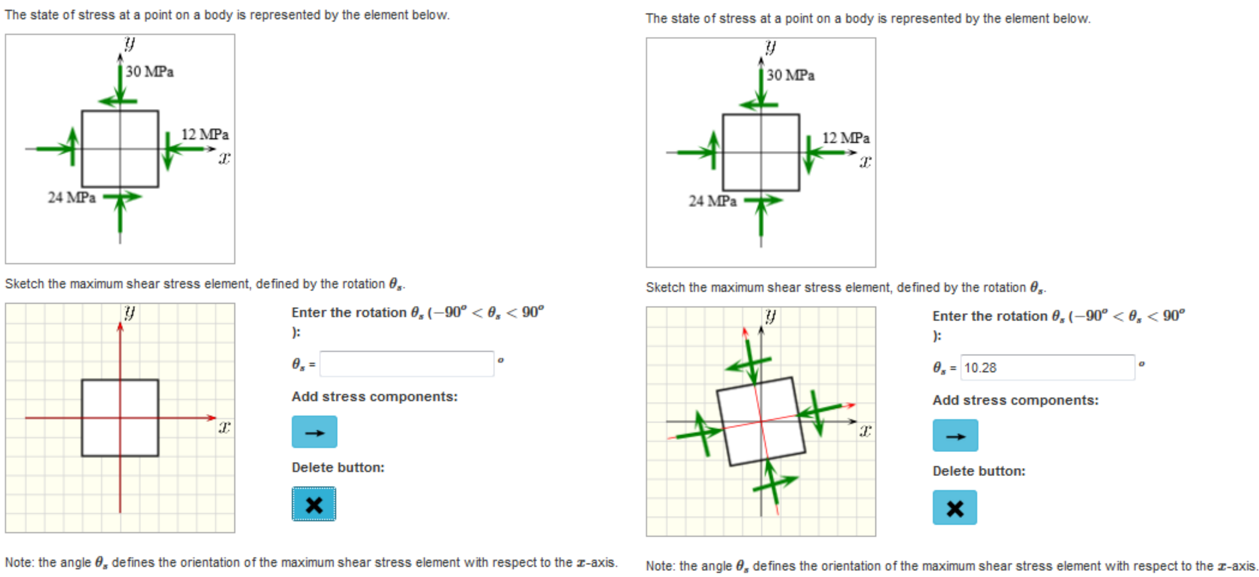

Figure 5: Drawing tool to solve stress transformation problems. Left: given problem. Right: solution including square rotation and arrow objects. The implementation of the grading function for this problem used the same tolerance check used for free-body diagrams.

the grading algorithm can be easily adapted to work on different types of drawing problems. For example, Figure 5 shows another type of sketching problem that typically appears in introductory solid mechanics classes, where students needed to complete the stress element (represented by the square), by drawing arrows representing stress components.

Requirements R4 and R5: Grading processing time and robustness. The presented grading algorithm framework is simple, fast to execute and does not require an iterative procedure. It has complexity of order equal to $O(nm)$, where n is the number of required objects and m is the number of submitted objects, which could be improved to $O(n) + O(m)$ using a spatial-search data structure such as a quad-tree or simple grid binning. Since we are not dealing with a large number of objects, the grading is finalized almost instantly even with $O(nm)$. In addition, the algorithm is robust and easily handles unexpected inputs such as the insertion of multiple objects.

5. Results from implementation

The system described in Section 3 was implemented within the PrairieLearn[14] online system and used for both homeworks and testing-centere[3;15] quizzes (frequent exams) in an introductory mechanics course at the University of Illinois at Urbana-Champaign in Fall 2016 with 180 students. The PrairieLearn system was introduced in the same class in Fall 2015, however, the drawing tool was not yet available and questions assessing shear and bending moment diagrams were created using a multiple-choice format. We present results in the following sections from three evaluation sources: (1) student interaction data with the system, (2) student affect data reported via survey and anonymous written feedback, and (3) instructor feedback.

5.1. Student interaction data with the system

The online PrairieLearn system[14] that delivered the problems to students also recorded all student interactions with the system. To analyze these interactions, we selected a pair of homework-quiz questions addressing the two distinct problem types of sketching shear force and bending moment diagrams (Figs. 6-7), and drawing stress elements. Before comparing quiz and homework performance, it is worthwhile to understand the difference in the grading scheme of these two different assessments. When taking a homework, students are allowed to have unlimited attempts to get a question marked as correct. Moreover, making a mistake in a given question does not prevent them from achieving a perfect score (grading scheme discussed in West et al.[14]). In addition, students can obtain a perfect score in a homework assessment without answering all questions. However, when taking a quiz, students have their score penalized each time they get the question marked as incorrect. The examples presented in Fig. 7 had a starting value of 10 points, and students were penalized by one point each time they answered a question incorrectly.

Summary data from student interactions with the PrairieLearn system is presented in Figure 8. We first compared the success rate for the two problem types in homeworks, i.e., the percent of students that had a question marked as correct (regardless of the number of attempts). The results are indicated in Fig. 8a and indicate that students were largely successful at solving the problems (70% and 90% success rates for the two problem types in homeworks), and the success rates were somewhat lower on quizzes as would be expected.

The system also tracks the number of attempts that each student needs to get a question marked as correct. Figure 8b shows the average number of attempts needed for a correct attempt, where the average is taken only over the students that got the question marked as correct. While doing problems on homeworks, around 5 to 6 attempts were needed on average for a student to correctly solve the problem, indicating that they were repeating the question to learn the concept, while on a quiz the average number of attempts for successful solutions was less than two, showing that students had achieved mastery or near mastery.

To understand the types of errors made by students, Figs. 8c-8d show how frequently the "object was not found" and "extra object found" errors were reported to students. The results depicted in Fig. 8c were obtained by first calculating the rate that each student received the message "object was not found", which excludes all the students that never made a mistake, and then averaging these rates over this set of students. In a similar way, the results that appear in Fig. 8d were

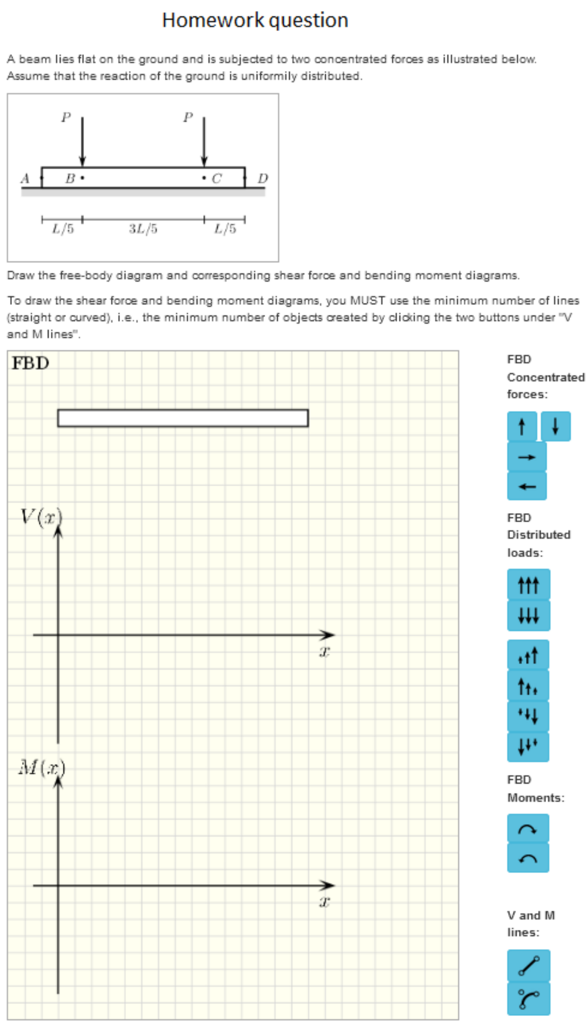

Figure 6: Homework question corresponding to the data collected in Fig. 8

obtained by first calculating the rate that each student received the message "extra object found", which again excludes all the students that never made a mistake, and then averaging these rates over this set of students. Overall, the error rates were higher for V-M diagrams, which is consistent with the lower success rates on these questions. The rates for the two error types were similar and high, indicating that it might be useful to report more detailed breakdown of the specific error cause (see also Section 5.2 for more discussion on this point).

5.2. Student perceptions survey

A survey addressing student perceptions of the drawing tools was administered after the end of the semester using an online software where 50 students responded to the survey (28% of the class). The results are summarized in Fig. 9.

Students agree that it is important to draw quality engineering drawings (Fig. 9-a), and indicated

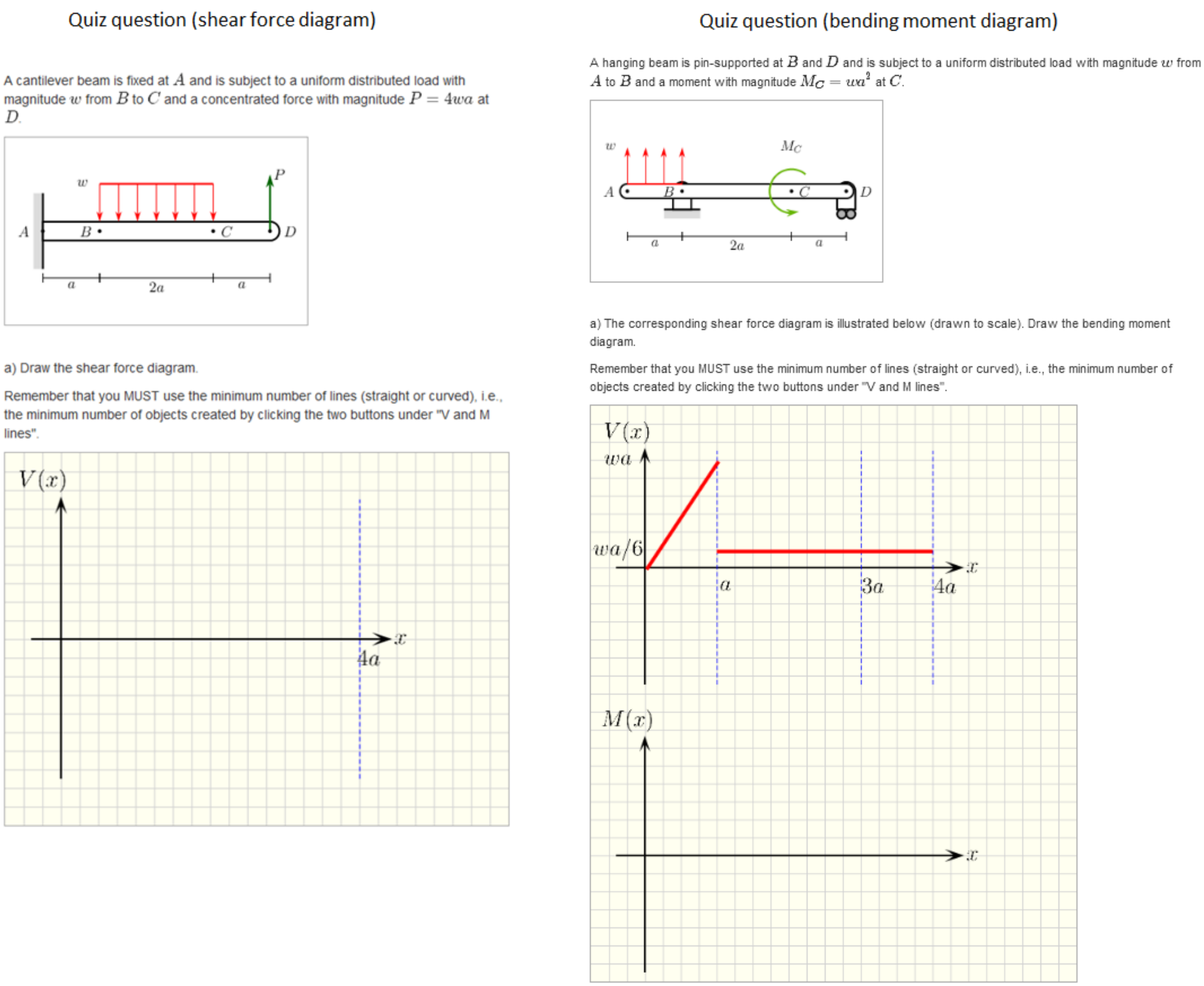

Figure 7: Quiz questions corresponding to the data collected in Fig. 8

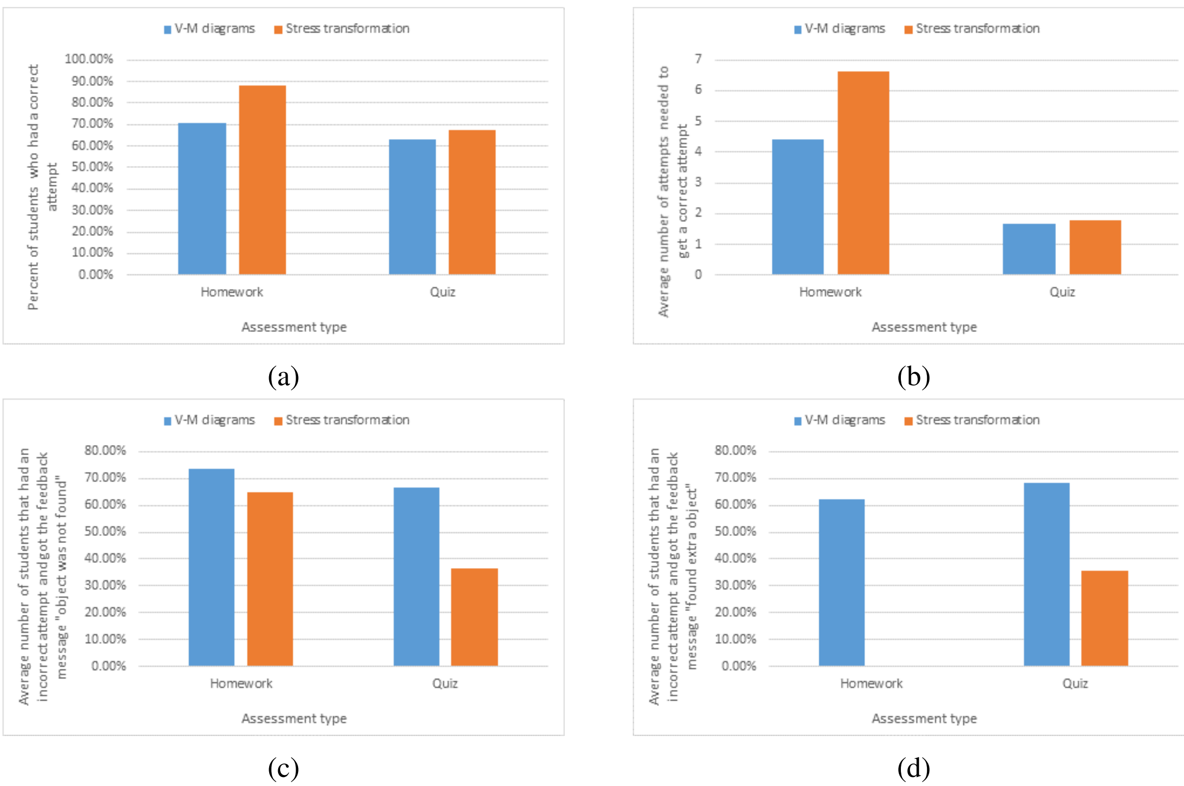

Figure 8: Data from student interactions with the online drawing problems. See Section 5.1 for discussion.

that the drawing tools helped them understanding important concepts covered in class (Fig. 9-b). Moreover, students felt that the drawing tools helped them in preparing for the quizzes (Fig. 9-c).

The survey also indicated that students were able to understand the drawing instructions (Fig. 9-d) and that the placement of vectors (objects) in the diagrams was not difficult (Fig. 9-e). However, many students felt that manipulating the objects inside the canvas (for example, rotating or moving the object to the correct location) was complicated (Fig. 9-f). On another hand, the majority of the students believed that if they knew what the correct answer should be, they could draw the diagrams using the drawing tool (Fig. 9-g). It is not clear to the authors if the "difficulty" in object manipulation was associated with the fact that when students didn't know the solution of the problem, they would just try random attempts using the drawing tool, and felt as if the drawing itself was the frustrating part of the problem solution. Based on observation of students during office hours, we noticed that students were spending a lot of time trying to place the objects precisely on the canvas, even tough it was announced that the tolerances were very large. To remediate this, we plan to include more messages about tolerances in the problem statement, to ease the manipulation of the objects.

One of the questions asked the students if they felt the feedback message helped them figure out the drawing mistakes on their own. The results show the class was evenly divided about the efficacy of the feedback message (Fig. 9-h). In this first implementation of the grading algorithm, the feedback was very simple and only pointed out missing or extra objects, or if a slope was incorrect or not. This simplistic approach caused confusion to many students, since some of their drawing errors would generate two messages (instead of one). Note that according to Algorithm 1, when a student inserts a controlledLine that matches the bounding box of the required object but has the incorrect slope, the algorithm will send a message "Found object but slope is incorrect", but also give another message "Found extra object", since the submitted object property found was still marked as false. The authors plan to work on improving the feedback algorithm, including correcting this inconsistency, but keeping a balance of how much scaffolding should be provided to students while maintaining a desirable difficulty level.

5.3. Instructor feedback

One of the authors was the instructor of the class when the drawing tool was first implemented in a introductory solid mechanics course. In previous semesters, students had very high performance when responding to questions about shear and bending moment diagrams or stress transformation in paper-based multiple-choice exams. The performance significantly decreased when the questions were presented during the computer-based quizzes using the drawing tools and there were no pre-defined answers to select from. However, this same trend was not observed for other types of questions, indicating that something was different about drawing questions. We believe that students were able to attain high performance in the past because they were able to guess and use partial knowledge to get the correct answer when presented with multiple-choice graphical questions. However, when asked to actually draw the answer to the questions, it was evident that the knowledge was lacking and students were not prepared to answer them successfully. More effort should be placed on drawing tools to help students develop good engineering graphing skills, and this new system does this.

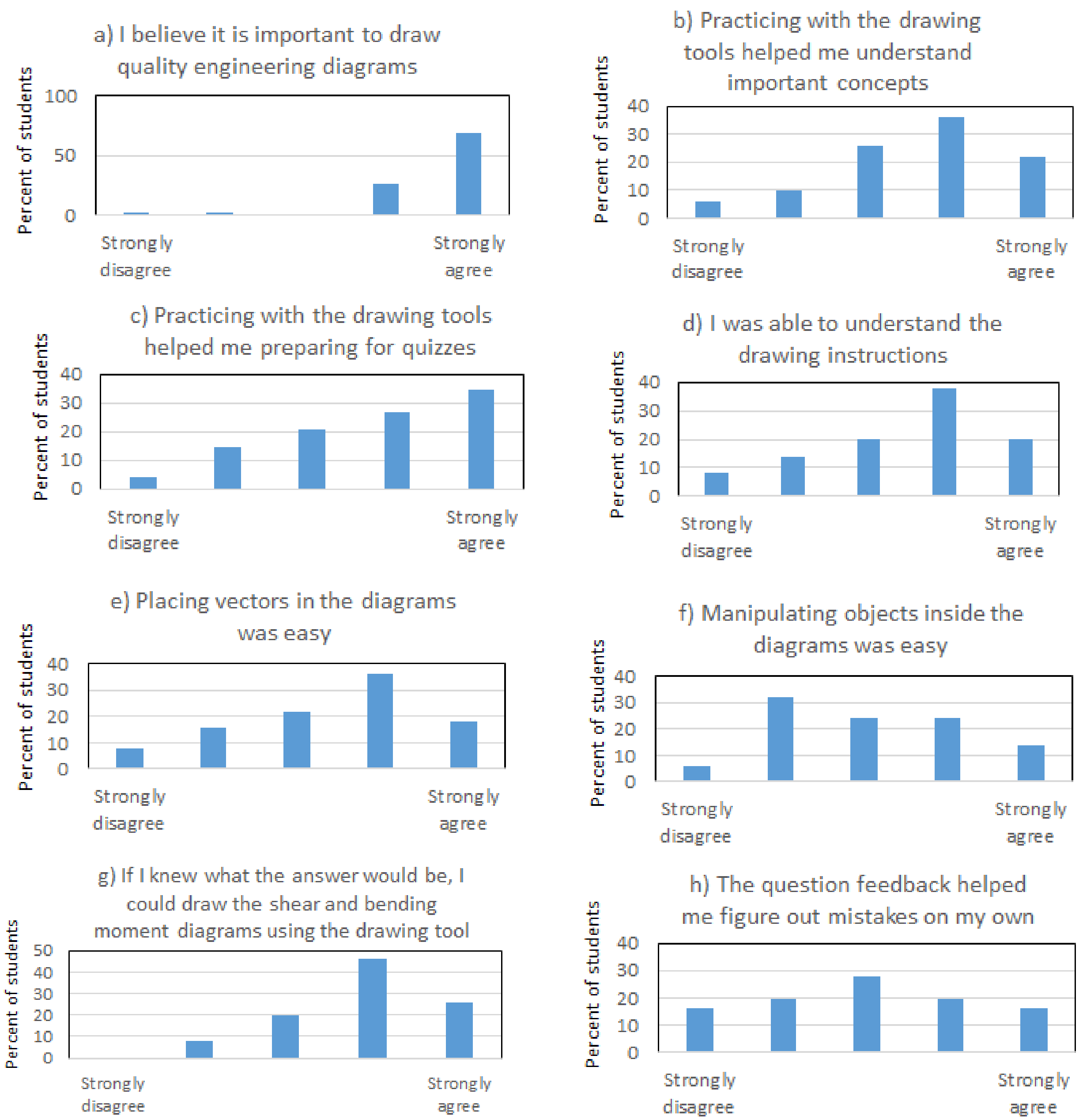

Figure 9: Anonymous survey results from $N = 50$ students. See Section 5.2 for discussion.

6. Conclusions

In this paper we presented a set of five key requirements for grading algorithms for drawing problems, we described and implemented a specific grading algorithm that satisfies the five requirements, and we collected and analyzed multiple types of evaluation data from the use of the system in an introductory mechanics course with approximately 180 students. The evaluation results indicate that the system was able to efficiently and robustly grade student diagrams and provide formative feedback. Future avenues of work include implementing more types of drawing questions (e.g., state-machine diagrams, circuit-drawing problems) and making it easy to optionally specify more precise feedback for scaffolding when students are first learning a topic.

Acknowledgments. This work was supported by NSF DUE-1347722, NSF CMMI-1150490, and the College of Engineering and the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign as part of the Strategic Instructional Initiatives Program (SIIP).

References

[1] J. R. Anderson. *Learning and memory: An integrated approach*. John Wiley and Sons, second edition, 2000.

[2] C. Berg and P. Smith. Assessing students abilities to construct and interpret line graphs: Disparities between multiple-choice and free-response instruments. *Science Education*, 78:527–554, 1994.

[3] B. Chen, M. West, and C. Zilles. Do performance trends suggest wide-spread collaborative cheating on asynchronous exams? In *Proceedings of the Fourth (2017) ACM Conference on Learning at Scale*, 2017. doi: 10.1145/3051457.3051465.

[4] J. A. Kulik and C.-L. C. Kulik. Timing of feedback and verbal learning. *Review of Educational Research*, 58:79–97, 1988.

[5] K. A. Lack. Current status of research on online learning in postsecondary education. Ithaka S+R, 2013.

[6] McGraw-Hill. McGraw-Hill Connect. URL `http://connect.mheducation.com/`.

[7] H. Pashler, N. J. Cepeda, J. T. Wixted, and D. Rohrer. When does feedback facilitate learning of words? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31:3–8, 2005.

[8] Pearson. MasteringEngineering. URL `https://www.masteringengineering.com/`.

[9] H. L. Roediger and A. C. Butler. The critical role of retrieval practice in long-term retention. *Trends in Cognitive Sciences*, 15:20–27, 2011.

[10] D. Rohrer, K. Taylor, and B. Sholar. Tests enhance the transfer of learning. *Journal of Experimental Psychology Learning Memory and Cognition*, 36:233–239, 2010.

[11] R. Roselli, L. Howard, and S. Brophy. A computer-based free body diagram assistant. *Computer Applications in Engineering Education*, 14:281–290, 2006. doi: 10.1002/cae.20088.

[12] S. Valentine, R. Lara-Garduno, J. Linsey, and T. Hammond. Mechanix: A sketch-based tutoring system that automatically corrects hand-sketched statics homework. In *The Impact of Pen and Touch Technology on Education*, pages 91–103. Springer, 2015.

[13] M. West and G. L. Herman. Sustainable reform of "Introductory Dynamics" driven by a community of practice. In *ASEE 2014: Proceedings of the American Society for Engineering Education 121st Annual Conference and Exposition*, 2014. Paper ID #10519.

[14] M. West, C. Zilles, and G. Herman. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *Proceedings of the American Society for Engineering Education (ASEE) 2015 Annual Conference*, 2015. doi: 10.18260/p.24575.

[15] C. Zilles, R. T. Deloatch, J. Bailey, B. B. Khattar, W. Fagen, C. Heeren, D. Mussulman, and M. West. Computerized testing: A vision and initial experiences. In *Proceedings of the American Society for Engineering Education (ASEE) 2015 Annual Conference*, 2015. doi: 10.18260/p.23726.