It's Not Just about Accuracy: Metrics That Matter When Modeling Expert Sketching Ability

TRACY HAMMOND, SHALINI PRIYA ASHOK KUMAR, MATTHEW RUNYON, JOSH CHERIAN, BLAKE WILLIFORD, SWARNA KESHAVABHOTLA, and STEPHANIE VALENTINE, Sketch Recognition Lab, Texas A&M University
WAYNE LI, College of Design, Georgia Institute of Technology
JULIE LINSEY, iDREEM Lab, Georgia Institute of Technology

19

Design sketching is an important skill for designers, engineers, and creative professionals, as it allows them to express their ideas and concepts in a visual medium. Being a critical and versatile skill for many different disciplines, courses on design sketching are often taught in universities. Courses today predominately rely on pen and paper; however, this traditional pedagogy is limited by the availability of human instructors, who can provide personalized feedback. Using a stylus-based intelligent tutoring system called *SketchTivity*, we aim to eventually mimic the feedback given by an instructor and assess student-drawn sketches to give students insight into areas for improvement. To provide effective feedback to users, it is important to identify what aspects of their sketches they should work on to improve their sketching ability. After consulting with several domain experts in sketching, we came up with several classes of features that could potentially differentiate expert and novice sketches. Because improvement on one metric, such as speed, may result in a decrease in another metric, such as accuracy, the creation of a single score may not mean much to the user. We attempted to create a single internal score that represents overall drawing skill so that the system can track improvement over time and found that this score correlates highly with expert rankings. We gathered over 2,000 sketches from 20 novices and four experts for analysis. We identified key metrics for quality assessment that were shown to significantly correlate with the quality of expert sketches and provide insight into providing intelligent user feedback in the future.

CCS Concepts: • **Applied computing → Intelligent learning environments**; **Computer-managed instruction**;

This material is based in part upon work supported by the National Science Foundation under Grant Numbers 1441331 and 1441291 and by Microsoft (Surface Hub Grant). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or Microsoft.
Authors' addresses: T. Hammond, Sketch Recognition Lab, Texas A&M University; email: hammond@cse.tamu.edu; S. P. A. Kumar, Sketch Recognition Lab, Texas A&M University; email: ashalinipriya90@gmail.com; M. Runyon, Sketch Recognition Lab, Texas A&M University; email: mattrunyon@tamu.edu; J. Cherian, Sketch Recognition Lab, Texas A&M University; email: jcherian92@gmail.com; B. Williford, ESketch Recognition Lab, Texas A&M University; email: bwilliford@gmail.com; S. Keshavabhotla, Sketch Recognition Lab, Texas A&M University; email: kswarna93@gmail.com; S. Valentine, Sketch Recognition Lab, Texas A&M University: email: steph.lynn.valentine@gmail.com; W. Li, College of Design, Georgia Institute of Technology; email: wayne.li@design.gatech.edu; J. Linsey, iDREEM Lab, Georgia Institute of Technology; email: julie.linsey@me.gatech.edu.

https://doi.org/10.1145/3181673

Additional Key Words and Phrases: Design sketching, design education, intelligent tutoring system, sketch recognition, intelligent feedback

ACM Reference format:
Tracy Hammond, Shalini Priya Ashok Kumar, Matthew Runyon, Josh Cherian, Blake Williford, Swarna Keshavabhotla, Stephanie Valentine, Wayne Li, and Julie Linsey. 2018. It's Not Just about Accuracy: Metrics That Matter When Modeling Expert Sketching Ability. *ACM Trans. Interact. Intell. Syst.* 8, 3, Article 19 (July 2018), 47 pages.
https://doi.org/10.1145/3181673

1 INTRODUCTION

Drawing is one of the oldest forms of human expression, predating even written communication. Throughout history, it has been used in a wide range of fields, including art, design, engineering, science, and education. Learning to draw is considered vital in learning to produce other forms of visual art. Not only are the skills acquired through learning drawing helpful in drawing a single design, but it is often the first step in producing new artwork. Furthermore, an artist's ability and intelligence—his or her understanding of form, artistic vision, and conceptual framework—can be revealed through their drawing. As such, drawing can be a powerful form of communication.

A sketch is a rapidly executed freehand drawing that is not usually intended to be a finished work [75]. Sketches are commonly preliminary drawings created prior to more sophisticated artwork or concepts. They are usually drawn quickly and with minimum detail to express and document high-level ideas for brainstorming and quick communication. Designers and other creative professionals use design sketches to quickly generate and explore ideas and communicate these ideas to others [31]. Even with the growing popularity of modern computing devices and software, design sketching is still used in various stages of the design process by experts in the field of art, engineering, and design because of its many benefits: It can be used to convey preliminary design ideas [61], identify and solve hidden challenges using visual sketches [61], assist in the ideation process during the exploration of ideas with loose sketches [12], and engage and attract the audience with designs [67]. The imperfection of sketches implies and encourages changeability, as opposed to CAD designs that demand the addition of details. The more detailed and complete a design, and the more time invested in it, the harder it becomes to abandon the design in exchange for a better one. Design sketching not only allow designers to communicate their ideas but also lets the learners improve their general academic performance and problem-solving capability [61], hone analytical skills [18], stimulate both halves of the brain [63], and improve writing and critical thinking skills when sketching is integrated into the thought process [15]. Design sketching also assists in other academic areas by boosting self-confidence from successful artistic pursuits [39] and improving three-dimensional (3D) spatial recognition skills [68].

Courses on design sketching are crucial to the development of future engineers. Traditionally, these courses are taught in classroom environments where students are taught by instructors and students practice on their own at home. The students' sketches are later presented to their peers and instructor to obtain feedback. This paradigm makes it difficult for the instructor to give valuable and detailed feedback as the class size increases.

To eliminate the problems that exist in the regular classroom setup, we have built an intelligent tutoring system called *SketchTivity*, a web-based platform where students can practice their design sketching fundamentals and get real-time humanlike feedback and assessment. *SketchTivity* has various lessons and challenges that allow students to practice the basics of design sketching. One major challenge of such an application is to mimic or possibly even improve on the immediate and useful feedback given by a dedicated human instructor.

Feedback is critical to steady progress in learning a new skill. Students are not only more motivated when they obtain immediate personalized feedback, but they also master the skill more quickly. Furthermore, feedback keeps students engaged in the activity of learning. Feedback is also critical to online activities. Chen et al. [13] identified immediate feedback as one the most important factor behind an optimal experience for web activities. Thus our system places an emphasis on providing users with feedback throughout the process of learning design sketching. While this article does not provide specific feedback techniques, it provides underlying information about what differentiates experts and novices so that a computer may effectively provide feedback and judge the progress of a user.

Artificial intelligence not only can learn from given data but also has the capability of discovering new facts about the data [66]. In this study, we apply artificial intelligence on sketch data not only to grade the sketches automatically without human intervention but also *to discover what about a sketch determines its quality*. Feature types were influenced by the rubrics of experts and then analyzed through the statistical significance of feature values that distinguish experts from novices.

The major contributions of this research are features to identify the quality of a sketch as well as the underlying meaning of the metrics that can eventually lead to humanlike intelligent personalized feedback. We have also designed a holistic score based on the features of a sketch. Such a score of *overall goodness of a sketch* is not intended to be shown to the user, as we expect that personalized, directed feedback that tells the user what to focus on will be more effective for them. Rather, such a metric provides value to the computer and to the overall analysis of the learning platform. When learning how to sketch, one metric, such as speed, may suffer when focusing on accuracy and vice versa. An overall score helps the computer to put in perspective a decrease in accuracy after suggesting to users that they try to increase their speed.

2 RELATED WORK

Web-based educational applications and intelligent-tutoring systems are gaining popularity in the domain of education and have proven to be an effective way of teaching and learning. Before diving into the details of our study, this section describes the related and relevant prior work to illustrate how *SketchTivity* differs from existing systems.

2.1 Sketch-based Intelligent Tutoring Systems

Sketch-based intelligent tutoring systems are applications that enable users to draw on a sketch-based user interface and to be provided feedback and assessment on their solutions. Mechanix [55, 73, 74], a sketch-based tutoring system for students enrolled in statics courses, allows students to enter planar trusses and free body diagrams into the system. These are then checked against the solution entered by the instructor to give detailed, real-time feedback. Mechanix helps students reach the correct answer by giving beneficial feedback such as "You are missing an input force at Node B" and "You have not drawn an axis." However, it does not provide feedback on the quality of the actual drawing, just feedback on forces and their locations [24, 72]. Mechanix has been shown to improve student learning and homework motivation [2–6].

Maestoso [8, 70] is another tutoring system that uses a sketch-based interface in the field of music for novice learners to learn music theory. It enables students to progress through lessons covering the fundamentals of music theory. This research tool also gives automated instructor-emulated feedback. It provides details on corrections to their solution and also reviews where students were unsuccessful in their solution. Other representative disciplines where sketch-based tutoring systems have been incorporated include East Asian languages [69, 71], math [40], and electrical engineering [16].

Tutoring systems that are more closely related to drawing and sketching instruction have focused solely on fine art drawing or other specific forms. Applications like iCanDraw [17] and EyeSeeYou [14] provide direction and feedback to the users while drawing faces and eyes, respectively, from an image to learn drawing technique. The Drawing Assistant [37], PortraitSketch [81], and Painting with Bob [9] are extensions of iCanDraw and EyeSeeYou that either incorporate a wider set of figures and detailed features or provide a digital painting platform for novices. These systems tend to focus on drawing specific forms rather than a general approach to drawing any form from one's imagination.

2.2 Creativity Support Tools

Some systems are not strictly educational but support users in creative activities. ShadowDraw [46] helps users draw certain forms like bicycles more accurately by utilizing a database of sketches and providing underlays (shadows) for the user to trace over as they sketch. Similarly, Ma et al. [51] built a sketch-based video annotation tool that assists users with their sketches by utilizing beautification and a database of user submitted sketches. While our system also helps users draw specific forms more accurately, it does so in a way that does not involve tracing or beautification. Instead, a faded scaffolding approach [38] is used that mirrors design sketching pedagogy.

2.3 Mobile Drawing Applications

Mobile-based pen and stylus interfaces have been developed in recent years and used widely in fields of drawing and sketching. Current educational drawing applications that teach design sketching can be divided into two categories: drawing instruction applications and drawing gaming applications. While drawing instruction applications have adapted aspects of drawing instructions on mobile devices, drawing gaming applications employ drawing as a game mechanic and do not necessarily teach drawing skills.

2.3.1 Drawing Instruction Applications. The Learn to Draw Sketchbook by Walter Foster[1] is a drawing instruction application that focuses on a fine art style of drawing and relies on a step-by-step tracing approach to teach drawing. The step-by-step approach is effective; however, it relies purely on tracing and does not teach drawing from imagination or perspective sketching. Furthermore, there is no way to gauge the performance of the users' sketch in the application.

Draw This from Peter Hamilton[2] is another application that uses a step-by-step guidance approach. Another valuable feature is that it incorporates accuracy metrics on tracing shapes. The downside of this application is that it relies purely on the tracing of shapes and forces users to draw shapes in a certain way, which is not the standard technique taught in design sketching. It does not help users understand three-dimensionality and perspective. The assessment capability provided by this application is limited and does not help students understand the areas to focus on.

2.3.2 Drawing Gaming Applications. Circled[3] developed by Underbeak[4] is an application in which the sole objective is to draw circles of various sizes accurately. It measures the accuracy of user's circles and, depending on the performance, more levels and modes are unlocked. Although this application provides feedback, it is limited to one feature—accuracy of the circle. It does not

[1] https://goo.gl/LdnV4n.
[2] http://goo.gl/RX57sn.
[3] https://goo.gl/cyMvdi.
[4] http://goo.gl/EfVXDN.

consider the speed and the technique used while drawing circles that are important factors in labeling them as good sketches.

Draw Something[5] is a multiplayer game that allows people to sketch and have others guess what they sketched. This game can be a good tool for improving visual communication skills. That said, it does not teach users any sketching techniques and is not professionally oriented. The application also has no way of automatically assessing the sketches drawn.

2.4 Sketch Recognition

Sketch recognition is the automated recognition of hand-drawn sketches and diagrams. Sketch recognition algorithms can be classified primarily into three categories: gesture-based recognition [62, 79], geometric recognition [25–27] and vision recognition [23, 54]. Gesture-based sketch recognition uses the inherent properties of the sketch path and timing to identify shapes. In these recognition algorithms either the system learns the user's style of drawing or the user has to learn in a style required by the system. Rubine used features like initial angle, sharpness, speed, and total angle traversed to recognize shapes. These are some of the most popular features cited in sketch recognition research. Sezgin, Stahovich, and Davis [65] took advantage of gesture-based features like speed and curvature to distinguish different shapes.

Geometric recognizers usually utilize a bottom-up approach where the basic shapes like lines, arcs, or circles are recognized first. A higher level recognizer is built on top of this low-level recognizer that uses geometric constraints to check if the primitive shapes when put together form a more complex shape. They allow users to draw a shape in a natural way. PaleoSketch [58] is a powerful low-level geometric recognizer capable of recognizing more than 10 basic shapes (up to 18 depending on the version [28, 29]) with high accuracy.

Vision-based algorithms use concepts from computer vision similar to those used on images after preprocessing of the sketches, focusing on the pixels left on the screen after the sketch is drawn. The screen coordinates are used by Kara and Stahovich [41] to apply template matching algorithms. Hausdorff, Modified Hausdorff, the Tanimoto coefficient, and the Yule coefficient are used in this article.

Some algorithms combine multiple approaches. KnitSketch [52] segments sketches into structural forms using both gesture-based and geometric sketch recognition. Still more state-of-the-art approaches like statistic and graph-based methods [48, 49] for sketch recognition continue to push the field further.

All the existing algorithms enable us to only identify different shapes, whereas *SketchTivity* assesses the sketches as well. *SketchTivity* uses geometric recognition in its first phase to recognize shapes and gesture-based features, which help to distinguish users for the assessment phase. We take advantage of the fact that gesture-based recognizers are user dependent in the second phase of our system.

3 DESIGN SKETCHING

Before discussing the software and our methods of evaluation, it is important to discuss traditional pedagogy in design sketching education. This is helpful in understanding the motivation of *SketchTivity* and the evaluation system presented in this study.

3.1 Traditional Pedagogy

Students in many disciplines, including industrial design, architecture, concept design, transportation design, and mechanical engineering, are often required to enroll in courses that

[5] https://goo.gl/GiS3hl.

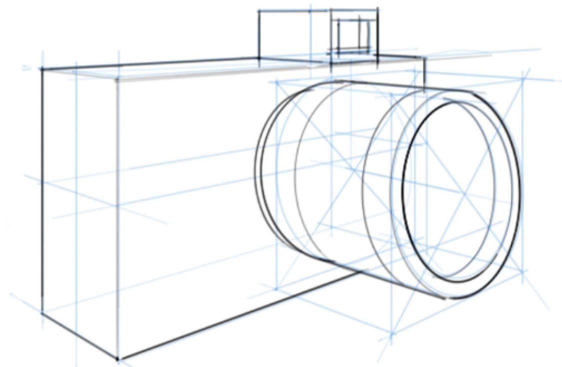

Fig. 1. Design sketch of camera.

teach fundamentals of design sketching because of the benefits design sketching and its related applications provide. These courses train students to learn sketching and draw quality sketches that approach the level of professionals in their respective fields. Students practice basic shapes and perspective drawing to eventually be able to combine primitives quickly and accurately to create complex three-dimensional sketches, such as that shown in Figure 1.

Students have several resources at their disposal to learn design sketching and master it. Sketching is typically taught in studio environments for industrial design and architecture. In studios, students are taught by the instructors during class time and are expected to practice on their own outside of class. Later, these sketches are shared for feedback from both the instructor and peers. This studio setting is the predominant means by which students learn sketching in universities. As the availability of instructors for feedback is typically limited to classroom and office hours, as the classroom size grows, it becomes increasingly difficult for the instructors to provide valuable individualized feedback to all of the students

Sketchbooks have proven to be an effective tool for learning sketching, as they provide both mobility and accessibility for students to practice their design sketching regularly and continuously [57]. An advantage of using sketchbooks is that one can keep track of the history of his or her practice and progress, which is important in regulating learning.

3.2 What Are Sketches?

In the previous sections, we discussed where sketches are used and how they are traditionally taught. This sections explains what sketches are and what the qualities of a good sketch are. Buxton provides in-depth knowledge of the characteristics of a design sketch [12]. He defines sketches as having the following qualities:

- Quick—A sketch is quick to make or at least gives that impression.
- Timely—A sketch can be provided when needed.
- Inexpensive—A sketch is cheap. Cost must not inhibit the ability to explore a concept, especially early in the design process.
- Disposable—If you cannot afford to throw it away when done, then it is probably not a sketch.
- Plentiful—Sketches tend not to exist in isolation.
- Clear Vocabulary—The style in which a sketch is rendered follows certain conventions that distinguish it from other types of renderings.

- Distinct Gesture—There is a fluidity to sketches that give them a sense of openness and freedom.
- Minimal Detail—Include only what is required to render the intended purpose or concept.
- Appropriate Degree of Refinement—By its resolution or style, a sketch should not suggest a level of refinement beyond that of a project being depicted.
- Suggest and Explore Rather than Confirm—Sketches don't "tell," they "suggest."
- Ambiguity—Sketches are intentionally ambiguous, and much of their value derives from their being able to be interpreted in different ways [12].

4 OVERVIEW OF *SKETCHTIVITY*

SketchTivity is a web-based stylus and touch capable intelligent tutoring system that allows students to learn and practice design sketching fundamentals [36, 47, 76, 78]. It incorporates the practices of traditional instructional pedagogy with artificial intelligence and entertainment [77], allowing students to practice a series of shapes with increasing complexity from simple, basic geometric shapes like lines and circles to more complex three-dimensional primitives, such as cuboids and cylinders. It also recognizes, analyzes, and evaluates the sketches using sketch recognition techniques.

SketchTivity has been integrated into the mechanical engineering design curriculum at the Georgia Institute of Technology as part of a pilot section. At Georgia Tech, they think it is crucial for a graduate to be able to sit down at a potential dinner and quickly sketch out several potential solutions with a client. As such, the students are taught sketching fundamentals [33]. Twenty students took part in a pilot using *SketchTivity*. Comparing the first half of their sketches versus the second half of their sketches, we show a significant increase in both accuracy (different in mean = 1.33832, $p < 0.00001$) and speed (difference in mean = 1.19862, $p < 0.00001$) [42]. It has also been shown to improve student sketching ability on a pre- and post-test graded by human experts [32, 34, 35].

4.1 User Interface

4.1.1 Lessons and Challenges. The system includes three main parts: lessons, challenges, and a sketchbook. The lessons section has been organized into basic, perspective, and primitive shapes. Figures 2, 3, and 4 show various lessons in the software. Currently, there are eight lessons overall, each having their own sub-lessons that differ by variations in angles, sub-types, and visibility of vanishing points in perspective within the shape. For example, the Lines lesson has three sub-lessons—horizontal lines, vertical lines, and diagonal lines. The following are the lessons and sub-lessons that are currently active: lines—horizontal, vertical, and diagonal; curves—horizontal arcs, vertical arcs, diagonal arcs, s-curves, and accelerated curves; boxes—squares and rectangles; circles—three different levels of scaffolding; planes—planes in two-point perspective with closer and farther vanishing points; ellipses—circles inscribed within squares in two-point perspective, with three different levels of scaffolding; three-dimensional boxes—cubes and cuboids; and cylinders.

Challenges (Figure 5) are designed to give the students an opportunity to enhance learning and improve their creativity, imagination, and perspective skills. These skills are much needed for design sketching. The sketchbook (Figure 6) has a plain sketching area with different tools, including pens and markers in various colors and thicknesses to choose from for free-form drawing. Both challenges and sketchbook sketches can be saved by the user.

Each of the lessons has eight exercises that are generated variations of the same problem by varying parameters such as angle, length, size, and location in space (perspective). These variants help students improve their muscle memory and hone their skill in drawing a particular shape. The

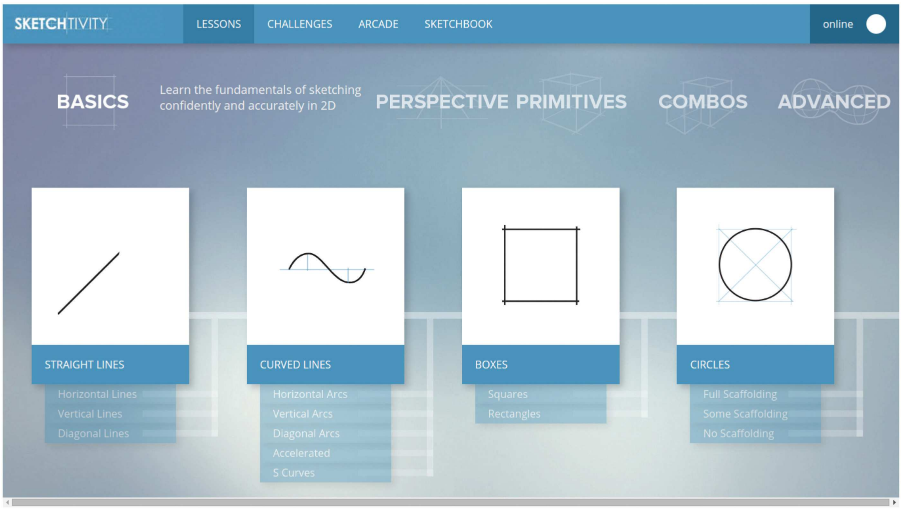

Fig. 2. Basic shapes.

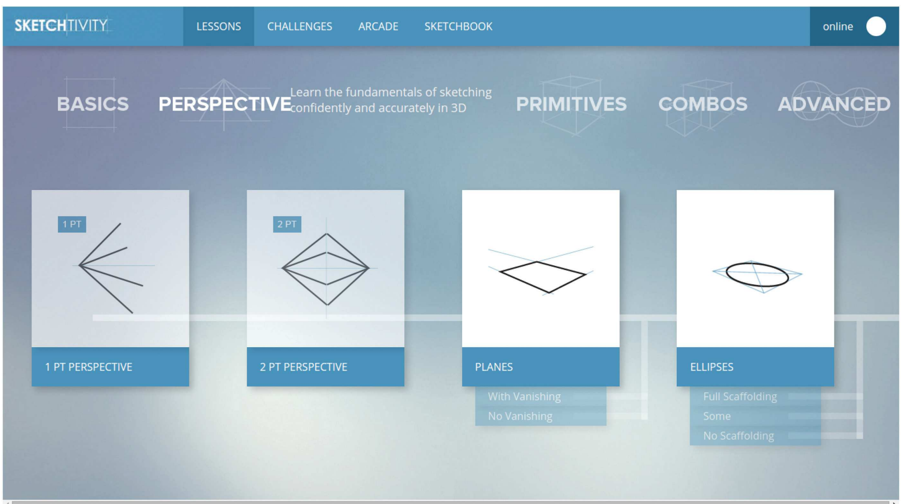

Fig. 3. Perspective shapes.

screen displays visual cues to indicate what a student is expected to draw. For example, the lines lesson has two dots that a student has to connect using a straight line. Also, students are given guidelines in the form of scaffolding to teach them proportions in drawing better. These scaffolding are removed slowly as one progresses to next sub-lessons. After the shape is recognized in an exercise, the system takes the student automatically to the next exercise. Students are evaluated and the results combining all of the exercises are displayed at the end of the lesson.

4.1.2 Drawing Tools. The students can view any of the completed exercises in the current lesson by clicking on the review panel on the left-hand side of the screen. This is a valuable tool for students to see how they are progressing through a lesson. Also, to help students while drawing, there are three buttons provided on the screen—*Undo*, *Skip*, and *Next* (see Figure 7). The *Undo*

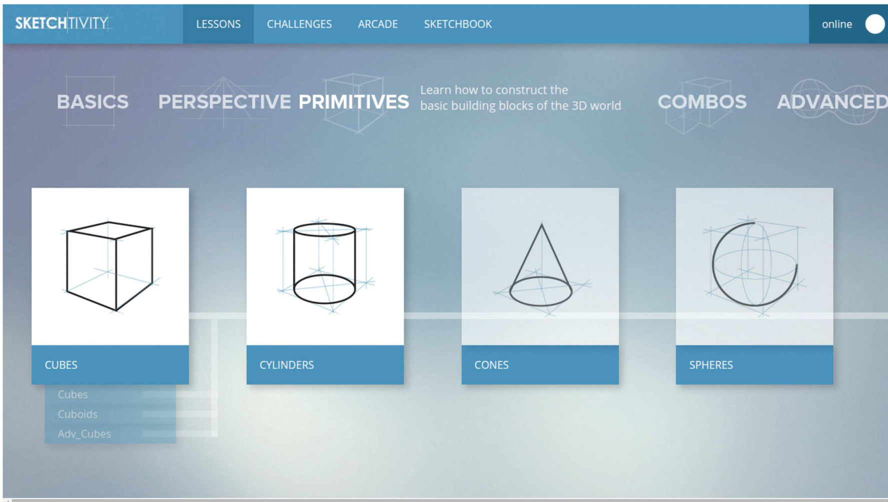

Fig. 4. Primitive shapes.

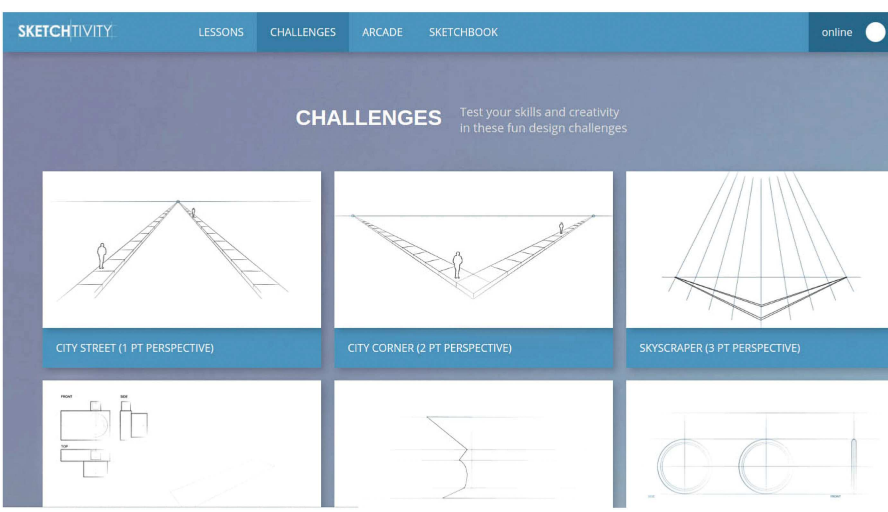

Fig. 5. Challenges.

button can be used to erase unintended strokes on the screen. Usage of this button is limited to once per exercise to restrict students from overusing the *Undo* button to submit better sketches. The *Skip* and *Next* buttons have similar functionality to each other. While the *Skip* button is designed to bypass the current exercise in a lesson, the *Next* button progresses to next exercise when the students are done with drawing, but the system failed to recognize it.

4.1.3 Architecture. The software is developed to work over the Internet on mobile devices. The front-end is developed using HTML5, CSS3, and JavaScript. All the recognition and evaluation algorithms are executed by the browser using JavaScript. This significantly reduces the load on the server and greatly speeds up responses in real time. The system runs on an Apache server. The

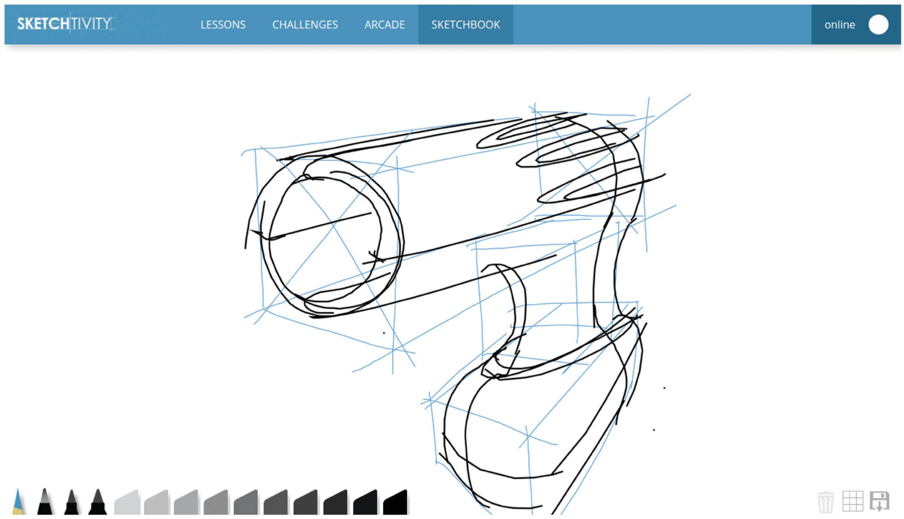

Fig. 6. Sketchbook.

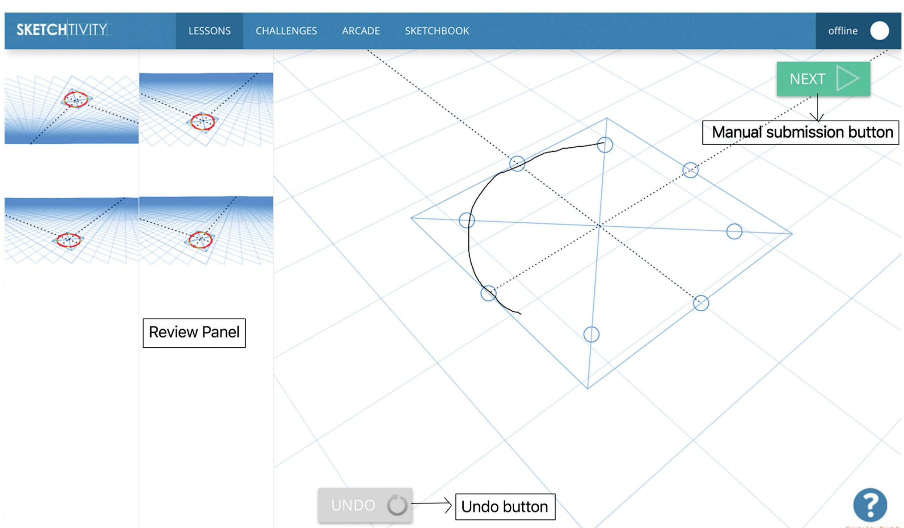

Fig. 7. Sample lesson in progress.

server-side has been developed using PHP where all the functions relating to storing sketch data and user sessions are executed. The sketches and user details are stored in a MySQL database.

The software stores each shape drawn by the user as a JSON document encoded as a *BLOb*[6] in the database. The system also stores the undone strokes to know the activity of users that led to final submission. Unfinished and unrecognized submissions that have been submitted by users are stored as well. The skipped lessons are not stored in the database.

[6]**B**inary **L**arge **Ob**ject.

4.2 Recognition System

4.2.1 Sketch Representation. Modern pen-based interfaces provide positional information along with current time in two-dimensional coordinate system that is usually the window coordinates. The software generates a point as the pen or stylus moves over the input device. We record each point as its x-y coordinate along with its timestamp. The timestamp recorded is the Epoch time, which is the time that has elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, January 1, 1970, in milliseconds. A stroke is a collection of time-ordered points that are between the pen-up and pen-down events. A sketch is composed of one or more strokes and is defined as a shape when it satisfies geometric constraints and is recognized by our software.

4.2.2 Preprocessing. Before passing strokes to our recognition system, it is necessary to preprocess the strokes. Sezgin and Davis [56, 64] state that the input strokes will contain noisy and inaccurate samples, which is caused by spatial and temporal quantization of the input by the hardware. Spatial digital noise comes from conversion of ink positions to screen coordinates. The difference in the sampling rates of the operating system and the tablet causes temporal quantization errors. This motivates us to eliminate and decrease the problems caused by such noise, and we resample points and time to do so. This also helps make subsequent steps in the process easier.

4.2.3 Resample Points. The number of points sampled by the system depends on the speed of the pen movement and the sampling rate of the hardware. There will be fewer points as speed increases. We then use the resampling algorithm described in the ShortStraw corner finding recognizer [80]. This resampling algorithm does not target a fixed number of points but rather resamples in such a way that the points are equidistant from each other. Resampling this way helps control for devices with different sampling rates or screen sizes.

4.2.4 Resample Time. It became apparent that the timestamps recorded for points of strokes that were drawn very quickly were not accurate. The timestamp remains the same for many consecutive points in such cases, causing problems when calculating speed, as mentioned by Sezgin and Davis [64]. To solve this issue, we resample time as well. We linearly interpolate the timestamp for a group of points with the same timestamp based on distance between the first point in the group and the next point in the stroke with a different timestamp.

4.2.5 Merging and Segmenting. To make recognition easier, we merge and segment the comprising strokes based on the geometric constraints of our recognition algorithm. For squares, rectangles, cubes, and cuboids, we segment the strokes to obtain the individual edges of the shape. For circles and ellipses, we merge the strokes that satisfy the following conditions: (i) The strokes are consecutive. The difference in the timestamps between the last point of the first stroke and the first point of the next stroke are less than a threshold. (ii) The strokes are close to each other. The distance between one of the end points of one stroke is close to one of the end points of the other stroke.

4.2.6 Overtracing and Overdrawing. Overtracing is the technique of drawing over an already-drawn stroke. This is a natural way of drawing closed shapes like circles and ellipses. In these specific cases of circles, it is usually observed that the end of circle overlaps with the beginning of the stroke. To facilitate this while keeping in mind our recognition and evaluation components of the system, we allow users to overtrace until it is within some threshold and ignore these extra data for our future processing of the strokes.

For straight lines connecting two given end points, the correct technique of drawing in design sketching is to draw beyond the end points ensure high line quality throughout the stroke. This

Table 1. Recognition Results

Shapes	#Completed sketches	#Recognized	Efficiency
Rectangles	314	311	0.99044586
Circles	919	914	0.994559304
Ellipses	567	566	0.998236332
Planes	64	63	0.984375
Cubes/Cuboids	251	224	0.892430279
Cylinders	615	608	0.988617886

technique is called overdrawing. In our system, overdrawn strokes are accepted and will be considered in further steps of recognition and evaluation.

4.2.7 Recognition. In our system, while we have an expectation of what shape users will draw each time, we still need to perform recognition to know when the user is finished drawing the intended shape. We have a *Next* button that allows users to click that they are done, but we have found that clicking that button every time interrupts the flow of their drawing practice. Thus, we use recognition to determine when the user is done drawing. While we could use a simple *timeout*, we found that fast, fluid sketches drawn quickly one after another are the best, so any timeout would have to be as short as possible. However, novices tend to pause between strokes while learning, so any timeout would need to be long enough to accommodate novices. Thus, we perform recognition to be able to intelligently know when the user is finished so that the system can provide immediate feedback as soon as the user is done drawing and advance to the next sketch after a short time for them to digest their feedback.

To achieve recognition, we rely on a unique combination of established sketch recognition algorithms. We use geometric recognition to identify the shape of the sketches drawn by students. The user interface lets a student draw strokes until the shape is recognized. Each time a stroke is drawn the recognition system takes that stroke, combines it with the previous drawn strokes and checks to see if the geometric constraints for the shape are satisfied. Once recognition is completed, the strokes are sent to the next stage for evaluation. Recognition results, as defined by the system correctly predicting that the student has finished his or her sketch before he or she has pressed *Next*, are shown in Table 1. Shapes that are not properly recognized are often due to tight system tolerances or sloppy user drawings. Additionally, when working with the system, students often became curious about the boundaries of what could be accepted, and they would experiment with the interface. So, recognition also serves as data validation. Because an unrecognized shape could mean that the shape was a completely invalid shape, the feature values were not included as part of this analysis.

4.2.8 Recognition Accuracy. In the spring 2016 semester in which 20 students participated in testing the software, our system recognized more than 97% of the sketched forms. Table 1 shows the efficiency of recognizing these basic forms. Each form has its own recognition algorithm and was iteratively tested and improved throughout the development process to be as robust as possible. The system recognizes the form after certain criteria have been fulfilled (e.g., a cube has 8 corners and 12 edges connecting them). The system checks for this criteria at the end of every stroke drawn. The user can also demand that a partially completed diagram be evaluated by clicking a button.

The unrecognized sketches were either drawn deliberately to break the system, or they were incomplete or drawn in a way that was unexpected. For example, several students drew the hidden lines (back side) of the cube with a dashed line. We plan to support this in the future.

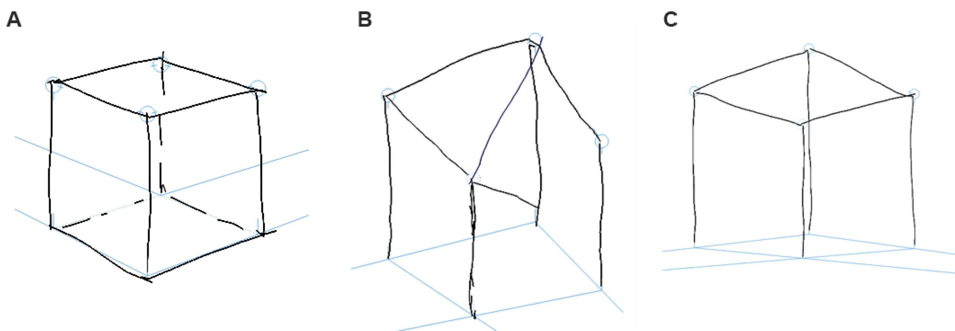

Fig. 8. Examples of unrecognized cubes. Cube A used dotted lines. Cube B has out-of-place lines and is missing the bottom side. Cube C is missing the bottom side, as the user assumed the construction lines could serve as the bottom side.

For all the shapes except cubes, students got their first exercise of a particular lesson recognized, implying an intuitive user interface. In the case of cubes, the students sometimes took a few attempts to understand what the exercise was expecting (i.e., all the sides were to be drawn irrespective of whether visible or not), and sometimes the recognizer failed them when what they drew was unexpected. The students sometimes joined the wrong corners or omitted important edges on the lower or invisible faces of the cube. Figure 8 shows examples of cube drawings that were not recognized by the system.

We have sketch data from experts as well. The recognition rates for those sketches were not included in the dataset because it was 100%.

5 GRADING RUBRIC

One of the essential elements of an intelligent tutoring system is to provide immediate feedback to the students to help in identifying the errors and guide them towards better understanding of the concepts [7]. Thus, assessing sketches and giving feedback is a significant component of *SketchTivity*. In this section, we explain different rubrics we use to assess the student sketches.

5.1 Pre-evaluation Processing

Before sending the sketches to the evaluation system, we preprocess the segmented and merged strokes to ignore the hooks at the start and end of the stroke that are common when drawing fast. When removing hooks, we limit the removal to no more than 5% of the stroke at either end for further analysis. In an effort not to remove points from the stroke such that we lose significant information about it, we also restrict the number of points removed to be a maximum of 5 points.

5.2 Categories for Grading Rubric

To design an assessment system for the sketches, the first step is to get useful features from the sketch that determine the quality of sketch. We have divided our evaluation features into three categories as given below. They are in order of hierarchy from more traditional to less traditional methods of sketching evaluation.

- Visual—The features in this category are dependent on how the final sketch looks. We consider only the x-y coordinates of the points on the sketch for all features in this category.
- Technique—These are a set of features that are dictated by the finesse with which the user executes the sketches. The features in this category include the timestamps of the points as

Fig. 9. Left: good smoothness, poor accuracy; right: good accuracy, poor smoothness.

well as the x-y vector coordinates of the sketches. From these data, we can measure features like speed and speed fluidity.

- Planning—These are a set of features that demonstrate high level planning in sketching the whole shape. These features tell us the way the strokes of the shape are drawn as well as how sketchers think about the shape they are drawing and what they are focusing on.

The set of features and/or their inspiration that have been used in this study for evaluating sketches have come from the following listed sources:

- Characteristics of sketches (mentioned in Section 3.2)
- Interviews of experts (instructors of design sketching)
- Features used in sketch recognition that we believed could capture the features and techniques described by the expert
- Features from motor-control studies that we believed could capture the fluidity and consistency mentioned by the experts [1, 21, 53]
- Planning features that we believe are important to identify the expertise level of the user based on watching experts drawing in action.

5.3 Motivations for Grading Rubric

This section briefly explains the motivation for choosing the initial set of features. Detailed explanation of the features is given in the following section.

5.3.1 Visual. Design sketches are drawn to represent ideas in visual form and to be shared with others. An important aspect of this is to be able to visually depict the idea that is conceived in one's mind and express it on paper. Visual accuracy, realism, and neatness of the sketches are important. One of the important reasons why computer-aided design (CAD) became popular was because one could draw accurate and precise designs using it, and drawings could be easily modified later without redrawing. We consider two different measures to evaluate sketches in *SketchTivity* based on this property of design sketches as explained below.

- Accuracy—Accuracy refers to the degree of conformity and correctness of the shape compared to the expected shape. In other words, this tells how close the sketch is to the expected shape.
- Smoothness—Smoothness refers to a state of consistency in the sketch. A wavy and jittery sketch is not neat and legible. This is another way of seeing the precision of the sketch.

Both accuracy and smoothness are needed in a sketch for it to clearly communicate the ideas of the person drawing it. A sketch with shaky or poor line quality can become too ambiguous and may be misinterpreted. This is generally not ideal unless the sketch is intentionally abstract. Figure 9 illustrates the difference between accuracy and smoothness. While the left one is a smooth fluid stroke, it is not accurate with respect to the desired shape, which in this case is a simple line. Conversely, the right one is accurate but not smooth.

5.3.2 Motion-based Technique. The motivation behind this category of features is to see how an user moves his hand during the process of sketching. Sketching is an acquired skill and the experts who have a lot of experience in sketching will have noticeable differences in their sketches and sketch quality.

- Speed—A designer draws multiple sketches in the early stages of the process to be able to decide what works and what does not. Hence, it becomes important for him/her to be quick enough in generating the sketches and putting his/her ideas on paper in visual form. This is also a feature that has been derived from design sketching practices.
- Speed fluidity—In the past several decades, there has been an abundance of research on the model of human movement towards a goal in human–computer interaction [21, 22]. Our motivation for this particular feature is from the research on the minimum jerk's law and trajectory-based Fitts' law. The speed of the motion is maximum in the middle of the path of the movement and decreases as you move away from this.
- Accuracy vs. speed—There has also been research in applying Fitts' law to find many other interesting relationships in hand movement. Fitts' law [53] provides details of how speed and accuracy have a tradeoff in aimed movements. Similar tradeoffs can be applicable to trajectory-based movements like using a stylus on tablets as well. This motivates us to further analyze the relationship between these features and see how they differ between experts and novice users.

5.3.3 Planning. An experienced designer who has more knowledge about the concepts of design sketching and its techniques will plan initially before starting to draw and will have an idea of what they are going to sketch. The expert has had enough experience to know which strokes should be drawn next and in the order that will make the sketch look good. This set of features has been developed from consulting with domain experts and referring to design sketching textbooks.

- Number of strokes—A user may drawn a square using a single stroke or multiple strokes. The number of strokes is an important feature for shapes such as rectangles and cubes, as novices tend to draw rectangles in one single stroke, while experts tend to draw rectangles using four lines.
- Stroke order coupling—When drawing a complex shape, one tends to use multiple strokes. There are exponential permutations of drawing the shape using multiple strokes. Experts, however, will group the drawing of parallel lines, whereas novices may group the drawing of connected lines. This is because novices tend to think of rectangles as four 90° angles in a row, whereas experts think of rectangles as two sets of parallel lines that meet at a vanishing point. In 2D perspective sketching, a drawn rectangle does not actually have 90° angles, even if the real-life representation does. This is a significant difference between how novices and experts perceive the world.
- Stroke direction coupling—A stroke can be drawn from left to right or right to left, bottom to top or top to bottom for a line, and clockwise or anti-clockwise for circles and ellipses. Because novices think of the rectangle as a closed shape, this can also be a big difference. However, this feature can be very dependent on handedness. Thus, for rectangles and cuboids, this feature instead measures if parallel lines are in the same direction.
- Overdrawing—A recommended practice is for the lines to be drawn beyond the end points to have a higher quality line between the end points. An inexperienced person may not know this or know the importance of this technique and may fail to follow it.

5.4 Features Used

5.4.1 Accuracy. To measure the accuracy of a shape with respect to the ideal expected shape based on the visual guidelines given, called the reference shape, we use motivation from template

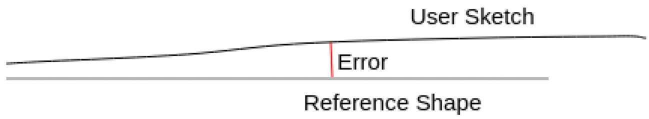

Fig. 10. Deviation of horizontal line at one point.

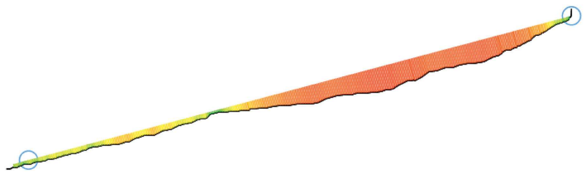

Fig. 11. Deviation of a diagonal Line.

based sketch recognition algorithms [41, 79]. For a given sketch, we find the distance between each of the points on the sketch and its corresponding point on the reference shape to calculate the deviation at that point. The measures used for measuring accuracy are as follows:

- Maximum deviation—The maximum of all deviations in a stroke(s).
- Average deviation—Calculated by adding the deviations at every point and dividing it by the total number of points.
- Hausdorff similarity—Hausdorff distance is a popular method used to compare how similar two sketches are [41]. Hausdorff distance between two sets of points A and B is given by the equation

$$H(A, B) = \max(h(A, B), H(B, A)), \tag{1}$$

where

$$h(A, B) = \max_{a \in A}(\min_{b \in B \|a-b\|}). \tag{2}$$

We use this distance and get the similarity measure for the sketch using the equation given below.[7] When the Hausdorff distance is zero, the similarity is 100%,

$$similarity(A, B) = 1 - (H(A, B).\frac{\sqrt{2}}{300})^{\frac{1}{1.4}}. \tag{3}$$

For lines (Figure 11), deviation at each point is calculated by finding a point on reference shape that is closest to it. This is same as the drawing a perpendicular line to the reference shape from the point on the sketch. Figure 10 illustrates this.

In case of circles (Figure 12) and ellipses (Figure 13), the distance between a point on the sketch and the point on the reference shape that is at the same radial angle to the center is the deviation of a point on the circle or ellipse. We first connect the point to the center and then find the point of intersection of that line on the reference shape to get the deviation of a point.

Squares and rectangles (Figure 14) are made of four sides. The evaluation system is given four strokes that make up the sketch of the square the user drew. They are merged and segmented before recognition phase. We find the corresponding line on the reference shape that is closest the user line and implement the same algorithm that we used for lines to get deviation at each point on these lines.

[7] http://goo.gl/52OsBi.

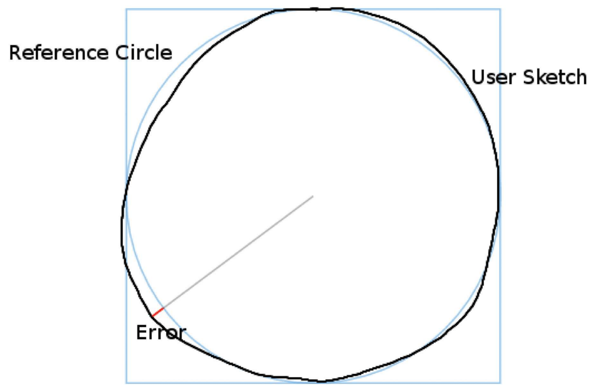

Fig. 12. Deviation of circle at one point.

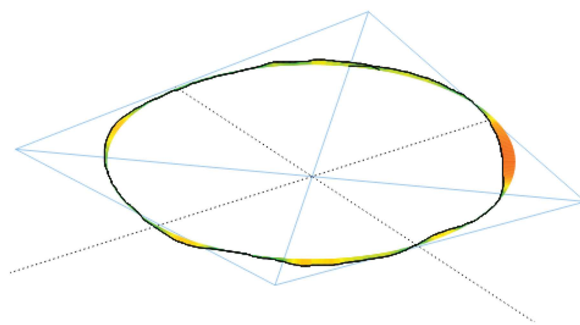

Fig. 13. Deviation of an ellipse.

Fig. 14. Deviation of a rectangle.

Cubes and cuboids (Figure 15) are made of 12 edges. For this also, like squares and rectangles, the user strokes are merged/segmented to form 12 separate lines representing the edges in the preprocessing stage, and then the corresponding edges of the reference shape that is closest to the user stroke is identified. After this, a similar method used for rectangles and squares is used to get the deviation for each point on the drawn cuboid.

5.4.2 Speed. Speed quantifies how fast the strokes in the sketch were drawn by the user. It measures the rate at which the user moves the pen over the screen of the tablet. It is calculated by

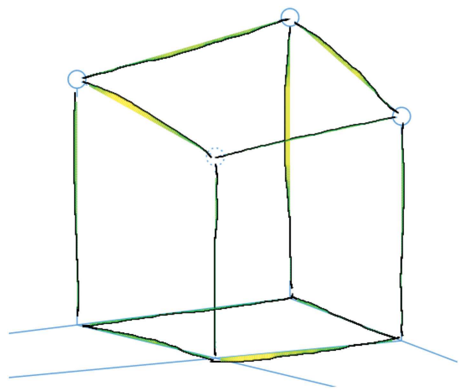

Fig. 15. Deviation of cubes.

using the following formula for speed:

$$speed = \frac{distance}{time}. \tag{4}$$

At each point the distance covered from the previous point is divided by the difference in their timestamps to get the speed. The features that have been used to measure speed are as follows:

- Maximum speed
- Minimum speed
- Average speed given by speed between consecutive points:

$$avgSpeed = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}}. \tag{5}$$

For all the shapes, these features are calculated in the same way. For complex shapes that have multiple strokes, the speed of each stroke is calculated to give a value for the whole shape together. Average speed is calculated by using the total path length traversed by each of the strokes divided by the time taken to finish the whole shape; that is, start time of the first stroke and the end time for the last stroke.

5.4.3 Smoothness or Accuracy Consistency. The texture of the sketch defines the smoothness. It is a way to represent the precision of the sketch. Smoothness helps in measuring the closeness of the sketch to the desired shape. It is defined as the measure of deviation of the stroke from the ideal shape that the user was intending to draw. A smooth stroke is drawn steadily without jolts during the action of sketching. Mathematically, the strokes that are differentiable at least once are smooth. A good quality sketch should have no unwanted kinks or cusps.

For calculating smoothness, we take motivation from three of the Rubine [62] features representing the curvature and sharpness. While the first two features denote the curvature of the stroke, the third feature distinguishes smooth strokes and those with sharp angles. The absolute angle of a "V" curve and "U" curve can be the same, but the sharpness feature help in distinguishing between them. The angle at a point p is calculated as shown in Figure 16 using Equation (6),

$$\theta_p = \arctan \frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} - \Delta y_p \Delta y_{p-1}}. \tag{6}$$

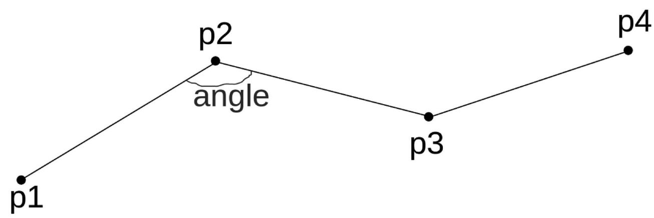

Fig. 16. Smoothness calculation of lines.

The following features are measured to know the smoothness of the sketch:

- Maximum absolute angle—maximum of absolute angle at every point
- Average angle—average of angle at every point at every point

$$averageangle = \frac{\sum_{p=1}^{p-2} \theta}{p-2} \tag{7}$$

- Average absolute angle—average of absolute angle at every point

$$averageabsoluteangle = \frac{\sum_{p=1}^{p-2} |\theta|}{p-2} \tag{8}$$

- Average squared angle—average of square of angle at every point

$$averagesquareangle = \frac{\sum_{p=1}^{p-2} \theta^2}{p-2} \tag{9}$$

- Total angle—sum of angle at every point at every point

$$totalangle = \sum_{p=1}^{p-2} \theta \tag{10}$$

- Total absolute angle—sum of absolute angle at every point

$$totalabsoluteangle = \sum_{p=1}^{p-2} |\theta| \tag{11}$$

- Total squared angle—sum of square of angle at every point

$$totalsquaredangle = \sum_{p=1}^{p-2} \theta^2. \tag{12}$$

For lines, at each point the angle between two lines—one joining the point and previous point and another joining the point and the next point—is calculated. Figure 16 demonstrates how the angle is calculated among points p_1, p_2, and p_3.

To get the smoothness of circles, it is not possible to use what has been used for lines because of the curvature of the circle. The angle between two adjacent lines joining consecutive points in a circle will always be at an angle, and this is dependent on the radius of the circle. To simplify the calculations of the circle sketch assessment, the reference circle is stretched in such a way that it forms a straight line. The points on the user's stroke are also stretched such that the distance between points on the original user sketch and the corresponding point on the original reference shape (the reference point at the same radial angle as the user point) is the same after stretching the shape. Figure 17 shows this concept.

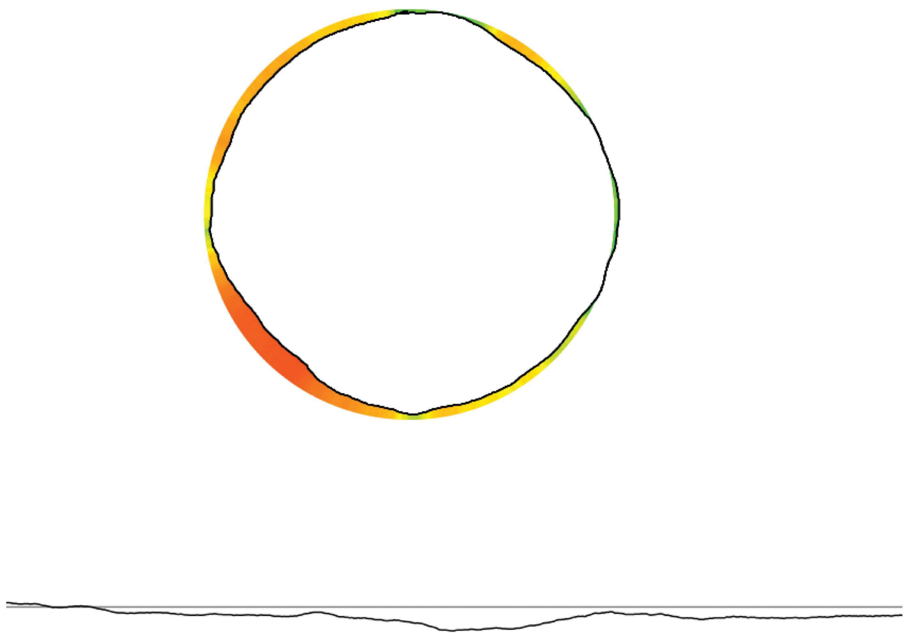

Fig. 17. Smoothness calculation of circles.

This is done by making a line the same length of the already resampled circle stroke. For each new point, the x-value is simply increased consistently by the resampled distance between the points (see Equation (13)). The y-value is simply the distance of the current point from the center point c minus the radius (see Equation (14)),

$$\begin{aligned} x_{new_i} &= distance(p_{old_i}, p_{old)i-1}) \\ &= \sqrt{(x_{old_i} - x_{old_{i-1}})^2 + (y_{old_i} - y_{old_{i-1}})^2}, \end{aligned} \tag{13}$$

$$\begin{aligned} y_{new_i} &= distance(c_i, p_{old_i}) \\ &= \sqrt{(x_{c_i} - x_{old_i})^2 + (y_{c_i} - y_{old_i})^2} - r. \end{aligned} \tag{14}$$

We assess ellipses in much the same way as circles. The ellipse reference shape is stretched into a straight line.

The smoothness of each of the four edges of squares and rectangles is measured separately to come up with a single value for each of the features mentioned in the list above. Each of the 12 edges of the cubes and cuboids are taken individually, and their smoothness is measured to get the smoothness features of the whole shape.

5.4.4 Speed Fluidity. Speed fluidity calculates how the speed changes during the drawing of a stroke. When drawing in a straight line, the speed profile generalizes to a regular parabola, in which the pen speeds up and then slows down as shown in Figure 18. This has been used in the past to recognize gestures before they are completed [50].

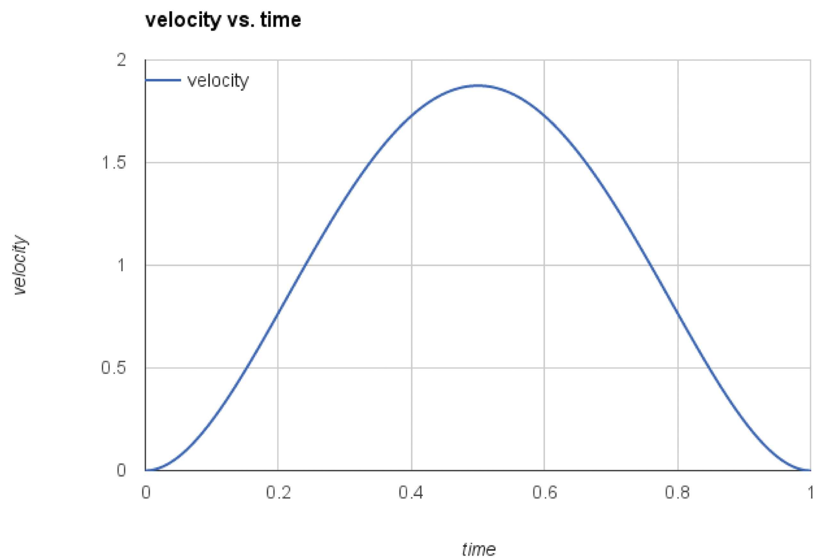

Fig. 18. Velocity vs. time graph while moving hand in space towards a target.

The following features have been used to measure it.

- Speed fluidity middle 50 to last 25—This is the ratio of speed between the points at 1/4th distance and 3/4th distance of the stroke to the speed between the points at 3/4th distance and the last point.
- Speed fluidity first 75 to last 25—This is the ratio of speed between the first point and the point at 3/4th of the distance of the stroke and the speed between the point at 3/4th distance of the stroke and the last point. Since the speed decreases in the end, the ratio is expected to be more than 1.

5.4.5 Planning.

- Stroke coupling and breaking—This feature helps us measure the way the strokes are coupled and broken to get the final recognizable sketches. It is speculated that the quality sketches will break the shape into components and try to draw them individually to get higher realism and better quality. We number the strokes in the final shape to determine this. We do not consider the stray strokes and the strokes that were removed using the *Undo* button.
- Stroke direction coupling—This feature helps us measure the direction of each of the strokes in the shape and gives the sketch an overall value. The possible directions for a straight line are top-to-bottom, bottom-to-top, right-to-left, and left-to-right. Two possible directions for circles and ellipses are clockwise and counterclockwise
  For horizontal lines, strokes drawn from left to right are given a score of 1 while strokes drawn from right to left are given a score of −1. Similarly, for vertical lines, we give a score of 1 and −1 for lines being drawn from top-to-bottom and bottom-to-top, respectively. In the case of diagonal likes, we treat them like horizontal or vertical lines depending on the angle they make with the *x*-axis. We use the grading score similar to the closer of the two possibilities—horizontal and vertical.
  For circles and ellipses, we assign a score of +1 and −1 for clockwise and anticlockwise respectively. To find the direction, we consider four points on the stroke that are equidistant

from each other, which will be the stroke's start point, the stroke's end point, and two points in between. Depending on how close the start and end points are, these four points will more-or-less form a shape similar to a triangle, with three lines connecting the four end points. Then we use Equation (15) to get the direction. This assumes that there is not a lot of overtracing. If there is a lot of overtracing, then more points are needed (or, rather, they should be closer together). One way to estimate the number of points needed is to multiply the length of the stroke by $3/2\pi r$. Equation (15) essentially just computes the angles between the points and checks if they are continually increasing or decreasing. The last line is only if the number of points was poorly estimated,

$$
\begin{aligned}
\theta_1 = angle(p_1 \Rightarrow p_2) = atan2(y_2 - y_1, x_2 - x_1) \\
\theta_2 = angle(p_2 \Rightarrow p_3) = atan2(y_3 - y_2, x_3 - x_2) \\
\theta_3 = angle(p_3 \Rightarrow p_4) = atan2(y_4 - y_3, x_4 - x_3) \\
if(\theta_2 - \theta_1) > \pi\{\theta_2 = \theta_2 - 2\pi\} \\
if(\theta_1 - \theta_2) > \pi\{\theta_2 = \theta_2 + 2\pi\} \\
if(\theta_3 - \theta_2) > \pi\{\theta_3 = \theta_3 - 2\pi\} \\
if(\theta_2 - \theta_3) > \pi\{\theta_3 = \theta_3 + 2\pi\} \\
if(\theta_3 < \theta_2 < \theta_1)\{return\ clockwise\} \\
if(\theta_1 < \theta_2 < \theta_3)\{return\ counterclockwise\} \\
else\{repeat\ process\ with\ one\ more\ equidistant\ point\},
\end{aligned}
\tag{15}
$$

We give a score ranging from 0 to 2 to squares and rectangles depending on the direction of each of the component strokes. When the strokes that are parallel are drawn in the same direction, the score is incremented by 1. Scores between 0 and 6 are given to cubes and cuboids based on the direction of the component strokes in the sketch. Drawing parallel sides of the cuboid with strokes in same direction moves the stroke towards 6.

- Stroke order coupling—This feature lets us check the order in which the strokes of a complex shape that can have multiple strokes are drawn. This feature does not mean anything for lines, circles, and ellipses. For the other two shapes, we describe the algorithm used briefly below.
  For squares and rectangles, we assign a binary score of 0 or 1 to the shapes based on the drawing of parallel lines together. We give the sketches in which the parallel lines have been drawn one after the other (i.e., drawing the top and the bottom horizontal side of the shape together and left and right vertical side of the square together) a score of 1 and the rest of the sketches a score of 0.
  A binary score of 0 or 1 is given to the cubes and cuboids depending on the order of drawing the top face, bottom face, and edges. We give a score of 1 when all the sides of one face or the vertical edges are drawn together one after the other before proceeding to another face or the vertical edges.
- Overdrawing—Designers draw the lines beyond the desired end points to get a better quality of line between the end points. To check this feature, we added a feature that is the ratio of length of the stroke to the total length of the whole shape. For complex shapes, the length of the strokes is added to get the path length of the sketch.

5.4.6 Ratio Metrics. There has been research on the tradeoff between accuracy and speed in the motor skills while moving the hand to an aimed target. To see if this kind of relationship exists in the act of sketching as well, we add the following features to our set:

- Ratio of average absolute angle to average speed
- Ratio of average square angle to maximum speed
- Ratio of average deviation to average speed
- Ratio of maximum deviation to maximum speed
- Ratio of average deviation to size of shape

Table 2 shows the final list of features used in analysis. While all of them are novel in this context, the table lists which of them are based on new algorithms created specifically for this article.

6 DATA COLLECTION

Our first goal was to determine which of these features were significantly different between novices and experts. This section describes the experiments conducted.

All the users were given separate credentials to login and draw sketches in each of the lessons using our software *SketchTivity*. To maintain consistency, they were all asked to use Wacom Cintiq tablets. All users were told to go through five of the lessons—Lines, Circles, Rectangles, Ellipses, and Cuboids.

Novice data—We used data collected at Georgia Tech from 20 students enrolled in *ME 1770, Introduction to Engineering Graphics and Visualization*. The class predominantly consists of freshmen Mechanical Engineering students ranging from ages 18 to 19. They had no formal education in design sketching, but some had taken K–12 art classes.

Expert data—We collected data from four different experts with expertise in the area of design sketching. They all had a high level of comfort in using Wacom Cintiq tablets. One of the experts is an instructor for the course, two other experts are lecturers and professors in the Industrial Design Department at Georgia Tech, and the final expert is a Master's graduate student who has been the TA for the course.

For this study, we use only the sketches that have been recognized by our recognition system, i.e., incomplete sketches that the system failed to recognize were not included. A total of 2,627 sketches collected from both experts and students were used for our analysis.

Table 3 shows the number of shapes for each of the shapes for both experts and novices.

Once the data were collected, we used our evaluation system to get all the feature values that were discussed in the previous section and other features that we believed are important and stored them in the database.

While differences between novices and experts can be more visibly seen in more complex sketches, sketch recognition can allow for differences to be seen even in the fundamental basic shapes and primitives comprising complex sketches. This is part of the motivation of our study, as mastering these basic fundamentals is crucial to producing more expert-level sketches, and it is important to assess improvement in them.

6.1 Analysis and Results

6.1.1 Correlation between Features. To see the relationships between different parameters while drawing, we perform a correlation between the different features discussed above. The analysis showed that there is moderate correlation between some of the features. The pair of features that has this kind of relationship were (i) speed and smoothness—positive correlation and (ii) accuracy and size of shape—negative correlation. So we added some more features to check for these relationships in the data. Table 4 shows the correlation for all the shapes for those features. Most of the other features had a correlation <0.05.

6.1.2 Experts vs. Students. The first step in developing a grading system is to check which of the features are helpful in making a quality sketch. For this analysis, we checked to see which of

Table 2. Overall List of Features

Feature	Category	Novelty (Inspired by)
Accuracy Metrics	**Accuracy**	**No**
Average deviation	Accuracy	No [60, 65]
Maximum deviation	Accuracy	No [20, 65]
Hausdorff similarity	Accuracy	No [20, 41]
Speed Metrics	**Speed**	**No**
Maximum speed	Speed	No [62]
Minimum speed	Speed	No [65]
Average speed	Speed	No [65]
Accuracy Consistency Metrics	**Smoothness**	**Yes for curved shapes**
Maximum angle	(Accuracy Consistency)	Yes for curved, No [62]
Average angle	(Accuracy Consistency)	Yes for curved, No [62]
Average absolute angle	(Accuracy Consistency)	Yes for curved, No [62]
Average square angle	(Accuracy Consistency)	Yes for curved, No [62]
Total angle	(Accuracy Consistency)	Yes for curved, No [62]
Total absolute angle	(Accuracy Consistency)	Yes for curved, No [62]
Total squared angle	(Accuracy Consistency)	Yes for curved, No [62]
Speed Consistency Metrics	**Fluidity**	**Yes**
Ratio of middle 50% to last 25% of stroke	Fluidity	Yes
Ratio of first 75% to last 25% of stroke	Fluidity	Yes
Planning Metrics	**Planning**	**Yes**
Number of strokes	Planning	No [30]
Stroke order coupling	Planning	Yes
Stroke direction coupling	Planning	Yes
Overdrawing—Ratio of path length to reference length	Planning	No [59, 60]
Combined Metrics	**Combined**	**Yes**
Ratio of average absolute angle to average speed	Combined	Yes
Ratio of average square angle to average speed	Combined	Yes
Ratio of average deviation to average speed	Combined	Yes
Ratio of maximum deviation to maximum speed	Combined	Yes
Ratio of average deviation to size of shape	Combined	Yes

Table 3. Number of Sketches

Shapes	#Student sketches	#Expert sketches
Lines	343	62
Circles	952	103
Rectangles	231	61
Ellipses	548	61
Cuboids	224	42

Table 4. Correlation between Features

Pair of features	Lines	Circles	Rect.	Ellipses	Cuboids
Average absolute angle and **average speed**	−0.25	−0.75	−0.65	−0.64	−0.58
Average square angle and **average speed**	−0.41	−0.67	−0.47	−0.56	−0.23
Average deviation and **size of shape**	0.10	0.544	0.43	0.10	0.47

the features work well in distinguishing novices from experts in the hope that we can use these features to eventually provide them targeted helpful feedback to improve their drawing. There are multiple ways of doing this using univariate analysis. We use Welch's t-test and subset selection to achieve this. Welch's t-test is a statistical test used to determine if two sets are significantly different from each other.

The next step is to see how well this set of features aids in distinguishing the two classes of users. To identify this, we use machine-learning classification algorithms.

6.1.3 Statistical Analysis. - We performed a t-test for each of the features on expert and student data. Welch's t-test, which controls for unequal variances, is used to see whether means of two sets of data are significantly different from each other. We use a t-test to understand if these features can be used to distinguish between the users and which of these features might be more useful in distinguishing novices from experts. The results of running a t-test for lines, circles, rectangles, and cuboids is given in Tables 5, 6, 7, 8, and 9, respectively. Any feature having a p-value below $\alpha = 0.05$ is considered significant and has been marked in gray in the tables.

For lines, most features were significantly different between novices and users. Lines are a fundamental building block for sketching. Enhanced speed, accuracy, and fluidity/consistency of lines benefit sketching many more advanced shapes, making it clear why so much time is spent on sketching lines in design sketching classes. An interesting feature to note is overtracing (ratio of path length to reference length). Experts draw through the marked end points, whereas novices tend to start and end at the marked end points.

For circles, we see an interesting effect in that accuracy does not contribute significantly towards distinguishing novices from experts. However, the smoothness of the lines does play an effect. We note that the minimum and average speeds are more significant (which are higher in experts), but not the maximum speed, showing that consistency is much more important than just drawing quickly. Notice that the ratio of path length to reference length is not significant, since experts do not do as much over-tracing on circles as they do on lines. Additionally, novices are less worried about stopping at the end points, so they do more over-tracing than they are do for other shapes.

Table 5. *t*-Test Results for Lines

Features	*t*-value	*p*-value
Accuracy Metrics		
Average deviation	4.64	<0.000005
Maximum deviation	4.73	<0.000005
Hausdorff similarity	−5.25	<0.000001
Speed Metrics		
Maximum speed	−6.12	<0.000001
Minimum speed	−3.11	<0.005
Average speed	−6.29	<0.000001
Accuracy Consistency (Smoothness) Metrics		
Maximum angle	3.99	<0.0001
Average angle	0.61	<0.54
Average absolute angle	12.60	<0.000001
Average square angle	3.56	<0.0005
Total angle	−1.46*	<0.15
Total absolute angle	7.33*	<0.000001
Total squared angle	5.57*	<0.000001
Speed Consistency (Fluidity) Metrics		
Ratio of middle 50% to last 25% of stroke	−4.11	<0.0001
Ratio of first 75% to last 25% of stroke	−4.78	<0.00001
Planning Metrics		
Number of strokes	N/A	N/A
Stroke order coupling	N/A	N/A
Stroke direction coupling	3.07	<0.005
Overdrawing—Ratio of path length to reference length	−2.59	<0.05
Combined Metrics		
Ratio of average absolute angle to average speed	4.37	<0.00005
Ratio of average square angle to average speed	1.65	<0.1
Ratio of average deviation to average speed	2.84	<0.005
Ratio of maximum deviation to maximum speed	3.58	<0.0005
Ratio of average deviation to size of shape	2.73	<0.01

Rectangles, like lines, have accuracy as a distinguishing feature, along with speed and many other of the same features as lines. Again, like circles, we see for rectangles that maximum speed is not a distinguishing feature and that rather average and minimum speeds that encourage consistency are higher differentiators. Planning features such as number of strokes and stroke order are extremely important, because experts draw parallel lines together, whereas novices do not.

In terms of significant features, ellipses look similar to circles, which is not surprising. Ellipses are harder to draw than circles, so we do get one accuracy feature with significance (Hausdorff), but in general it is the smoothness, fluidity, and speed features that make the most difference.

For cuboids, we get the surprising result that accuracy is not significantly different, just speed, smoothness, and fluidity. This may be because as more lines go on the screen, there is more confusion, and our eyes are better at averaging to get the right answer. In terms of planning features, while the number of strokes is significant—since we still get many novices drawing rectangles in

Table 6. t-Test Results for Circles

Features	t-value	p-value
Accuracy Metrics		
Average deviation	1.14	<0.26
Maximum deviation	0.30	<0.77
Hausdorff similarity	0.21	<0.84
Speed Metrics		
Maximum speed	−1.95	<0.053
Minimum speed	−5.72	<0.000001
Average speed	−3.69	<0.0005
Accuracy Consistency (Smoothness) Metrics		
Maximum angle	−3.35	<0.005
Average angle	−2.61	<0.05
Average absolute angle	6.16	<0.000001
Average square angle	5.72	<0.000001
Total angle	−3.51	<0.001
Total absolute angle	6.48	<0.000001
Total squared angle	6.32	<0.000001
Speed Consistency (Fluidity) Metrics		
Ratio of middle 50% to last 25% of stroke	0.88	<0.39
Ratio of first 75% to last 25% of stroke	0.78	<0.44
Planning Metrics		
Number of strokes	−0.17	<0.87
Stroke order	N/A	N/A
Stroke direction	1.37	<0.18
Overdrawing—Ratio of path length to reference length	−0.10	<0.92
Combined Metrics		
Ratio of average absolute angle to average speed	9.24	<0.000001
Ratio of average square angle to average speed	9.29	<0.000001
Ratio of average deviation to average speed	9.10	<0.000001
Ratio of maximum deviation to maximum speed	7.75	<0.000001
Ratio of average deviation to size of shape	3.14	<0.005

a single stroke—stroke order is not. This may be because there are many parallel lines, and experts may group them in very different ways.

Table 10 highlights the features that were statistically significantly different for each shape. Again, this shows that speed, smoothness, and fluidity were important across all shapes. Note that the number of strokes is not considered for lines as the shape is immediately recognized after the first stroke. Note that stroke order is not considered for lines for the same reason. Likewise, it is not considered for ellipses and circles, since most experts draw those shapes with one stroke, and number of strokes will capture that feature.

Classification—To test how well the set of features can distinguish the experts and students, we also performed classification on the data. We tested multiple classifiers (e.g., Linear, REPTree, SVM, Random Forest, etc.). Each classifier performed similarly (yet slightly worse) than the random forest classifier, and thus we report only the results of the Random Forest classifier here. Random Forest is an ensemble learning method that outputs the most commonly produced result of many

Table 7. *t*-Test Results for Rectangles

Features	*t*-value	*p*-value
Accuracy Metrics		
Average deviation	5.72	<0.000001
Maximum deviation	5.13	<0.000005
Hausdorff similarity	−3.36	<0.005
Speed Metrics		
Maximum speed	−1.10	<0.28
Minimum speed	−8.32	<0.000001
Average speed	−6.54	<0.000001
Accuracy Consistency (Smoothness) Metrics		
Maximum angle	6.91	<0.000001
Average angle	0.73	<0.47
Average absolute angle	14.18	<0.000001
Average square angle	11.86	<0.000001
Total angle	0.29	<0.77
Total absolute angle	11.00	<0.000001
Total squared angle	10.47	<0.000001
Speed Consistency (Fluidity) Metrics		
Ratio of middle 50% to last 25% of stroke	−2.39	<0.05
Ratio of first 75% to last 25% of stroke	−2.41	<0.05
Planning Metrics		
Number of strokes	−11.42	<0.000001
Stroke order coupling	−13.06	<0.000001
Stroke direction coupling	−1.00	<0.33
Overdrawing—Ratio of path length to reference length	−4.48	<0.00005
Combined Metrics		
Ratio of average absolute angle to average speed	8.93	<0.000001
Ratio of average square angle to average speed	7.85	<0.000001
Ratio of average deviation to average speed	11.19	<0.000001
Ratio of maximum deviation to maximum speed	8.74	<0.00001
Ratio of average deviation to size of shape	10.78	<0.000001

individual trees and has been shown to achieve high accuracy across a wide range of problems [11]. All results use 10-fold cross-validation to classify the experts and students. Since the number of experts in our study is much less than the number of students, the number of samples in both classes is imbalanced, making it not suitable to be used for classification directly. We use a method called undersampling that randomly removes instances from the majority class to reduce its effect on the machine-learning algorithm. To do this, we use Weka, an open source package that contains a number of machine-learning algorithms. The results of the classification are given in Table 11.

We notice that novices and experts are more strongly differentiated for those shapes that include perspective points. We think this is because many novices may already have lots of practice drawing lines and circles. But once they have to start thinking about perspective points, their sketches start to show evidence of hesitation and uncertainty, including a slower speed, wavering lines, stops and starts, and so on.

Table 8. *t*-Test Results for Ellipses

Features	*t*-value	*p*-value
Accuracy Metrics		
Average deviation	−1.22	<0.23
Maximum deviation	−0.26	<0.80
Hausdorff similarity	2.67	<0.01
Speed Metrics		
Maximum speed	−6.00	<0.000001
Minimum speed	−8.42	<0.000001
Average speed	−14.53	<0.000001
Accuracy Consistency (Smoothness) Metrics		
Maximum angle	−0.84	<0.41
Average angle	0.24	<0.82
Average absolute angle	27.95	<0.000001
Average square angle	19.62	<0.000001
Total angle	−0.67	<0.51
Total absolute angle	24.42	<0.000001
Total squared angle	17.54	<0.000001
Speed Consistency (Fluidity) Metrics		
Ratio of middle 50% to last 25% of stroke	3.50	<0.001
Ratio of first 75% to last 25% of stroke	6.09	<0.000001
Planning Metrics		
Number of strokes	2.85	<0.005
Stroke order coupling	N/A	N/A
Stroke direction coupling	2.20	<0.05
Overdrawing—Ratio of path length to reference length	−6.87	<0.000001
Combined Metrics		
Ratio of average absolute angle to average speed	22.09	<0.000001
Ratio of average square angle to average speed	18.88	<0.000001
Ratio of average deviation to average speed	7.98	<0.000001
Ratio of maximum deviation to maximum speed	10.04	<0.000001
Ratio of average deviation to size of shape	1.63	<0.11

6.2 Holistic Score

For the computer to keep track of a students' overall progress, we designed a holistic grading metric to evaluate the sketches. To design a metric that increased as sketches increased, we needed not only data that were labeled isExpert and notIsExpert but also continuous values that represent the goodness of a sketch to train and test against.

We collected 40 sketches of each shape type[8] and divided them into 2 groups of 20.[9] We then asked two experts to rank all of the shape groups from best to worst (1 being the best).

Ranked data give much more fine-grained information about how a sketch improves over time. This also helps address the problem that our current data were only divided between experts and novices, and both expert and novice data will vary over time. Thus, it is possible to have examples where individual novice sketches are better than individual expert sketches. These ranked sketches

[8]Except cuboids, where we only had 37 linked metrics to drawings.
[9]Twenty and 17 for the cuboids.

Table 9. t-Test for Cuboids

Features	t-value	p-value
Accuracy Metrics		
Average deviation	−0.47	<0.64
Maximum deviation	−0.88	<0.39
Hausdorff similarity	0.066	<0.95
Speed Metrics		
Maximum speed	−6.38	<0.000005
Minimum speed	−3.68	<0.001
Average speed	−18.015	<0.000005
Accuracy Consistency (Smoothness) Metrics		
Maximum angle	−2.99	<0.005
Average angle	−1.85	<0.071
Average absolute angle	17.72	<0.000005
Average square angle	9.53	<0.000005
Total angle	−0.57	<0.57
Total absolute angle	−0.03	<0.99
Total squared angle	−0.02	<0.99
Speed Consistency (Fluidity) Metrics		
Ratio of middle 50% to last 25% of stroke	−3.50	<0.005
Ratio of first 75% to last 25% of stroke	−1.13	<0.27
Planning Metrics		
Number of strokes	−3.74	<0.0005
Stroke order	−0.91	<0.37
Stroke direction	−5.44	<0.000005
Overdrawing—Ratio of path length to reference length	−5.16	<0.000005
Combined Metrics		
Ratio of average absolute angle to average speed	13.035	<0.000005
Ratio of average square angle to average speed	10.25	<0.000005
Ratio of average deviation to average speed	15.57	<0.000005
Ratio of maximum deviation to maximum speed	2.64	<0.05
Ratio of average deviation to size of shape	0.98	<0.34

will serve as both our initial training data (only half is used in training) as well as our gold standard for comparing our numerical improvement.

The experts ranked the sketches based on perceivable differences between accuracy and line quality (smoothness of line work) of the various forms. These two metrics form the basis for grading in design sketching courses and are also among the only features that can be observed by humans. Sketches are ideally accurate and the line work is clean and smooth as opposed to choppy and inconsistent [45].

6.2.1 Similarity Metric. Before we could use our rankings as training or a gold standard, we need to make certain that the rankings were reliable. As such, we analyzed the similarity of the ranked lists from the two experts: E_1 and E_2.

To determine the similarity of the two experts, we first computed the Euclidean distance (see Equation (16)) for each group of 20 sketches between reviewers E_1 and $E2$.

Note that the similarity metric would be the negative of this value (see Equation (17)).

Table 10. Summary of Feature p-values ($p < 0.05$ Is Shaded)

	Lines	Circles	Rectangles	Ellipses	Cuboids
Accuracy Metrics					
Average deviation	<0.000005	<0.26	<0.000001	<0.23	<0.64
Maximum deviation	<0.000005	<0.77	<0.000005	<0.80	<0.39
Hausdorff similarity	<0.000001	<0.84	<0.005	<0.01	<0.95
Speed Metrics					
Maximum speed	<0.000001	<0.053	<0.28	<0.000001	<0.000005
Minimum speed	<0.005	<0.000001	<0.000001	<0.000001	<0.001
Average speed	<0.000001	<0.0005	<0.000001	<0.000001	<0.000005
Accuracy Consistency (Smoothness) Metrics					
Maximum angle	<0.0001	<0.005	<0.000001	<0.41	<0.005
Average angle	<0.54	<0.05	<0.47	<0.82	<.071
Average absolute angle	<0.000001	<0.000001	<0.000001	<0.000001	<0.000005
Average square angle	<0.0005	<0.000001	<0.000001	<0.000001	<0.000005
Total angle	<0.15∗	<0.001	<0.77	<0.51	<0.57
Total absolute angle	<0.000001∗	<0.000001	<0.000001	<0.000001	<0.99
Total squared angle	<0.000001∗	<0.000001	<0.000001	<0.000001	<0.99
Speed Consistency (Fluidity) Metrics					
Ratio of middle 50% to last 25% of stroke	<0.0001	<0.39	<0.05	<0.001	<0.005
Ratio of first 75% to last 25% of stroke	<0.00001	<0.44	<0.05	<0.000001	<0.27
Planning Metrics					
Number of strokes	N/A	<0.87	<0.000001	<0.005	<0.0005
Stroke order coupling	N/A	N/A	<0.000001	N/A	<0.37
Stroke direction coupling	<0.005	<0.18	<0.33	<0.05	<0.000005
Overdrawing—Ratio of path length to reference length	<0.05	<0.92	<0.00005	<0.000001	<0.000005
Combined Metrics					
Ratio of average absolute angle to average speed	<0.00005	<0.000001	<0.000001	<0.000001	<0.000005
Ratio of average square angle to average speed	<0.1	<0.000001	<0.000001	<0.000001	<0.000005
Ratio of average deviation to average speed	<0.005	<0.000001	<0.000001	<0.000001	<0.000005
Ratio of maximum deviation to maximum speed	<0.0005	<0.000001	<0.00001	<0.000001	<0.05
Ratio of average deviation to size of shape	<0.01	<0.005	<0.000001	<0.11	<0.34

Table 11. Results of Random Forest Classification

	Recall	Precision	F-measure
Lines	0.799	0.799	0.799
Circles	0.631	0.633	0.630
Rectangles	0.840	0.840	0.839
Ellipses	0.910	0.911	0.910
Cuboids	0.982	0.982	0.982

We arrived at this number by first randomly selecting 20 shapes from each of our shape groups and sending them to two experts to rank independently from 1 to 20, where 1 was the best. The average rank then served as the gold standard,

$$distance(x, y) = \sqrt{\sum_{i=1}^{20}(E1_i - E2_i)^2}, \tag{16}$$

$$similarity(x, y) = \sqrt{\sum_{i=1}^{20}(E1_i - E2_i)^2}. \tag{17}$$

We used a Monte Carlo method to determine the standard distribution for a random set of rankings. We generated 1 million random rankings from 1 to 20. After doing this several times, we noted that 95% of the data always produced a Euclidean distance of greater than 28.77 or a similarity less than −28.77. For all shapes, the two reviewers had a similarity greater than −28.77, or $p < .05$. Looking at their data, it was interesting to note that the two experts were almost identical towards the beginning and end of the spectrum but had more disagreement in the middle. Table 12 shows the details for the two experts for their first set of 20 ranked lines.

6.2.2 Training the Classifier. We trained a classifier on the average rank produced by our two experts. Because we do not have a lot of data, we first reduced the number of features to help avoid over-fitting. For each shape, we removed all features with $p > 0.001$. We then trained a linear regression classifier on this data using 20-fold leave-one-out cross-validation. Tables 12 and 13 shows the similarity of the two expert raters as well as the computer versus the average of the expert raters.

We chose linear regression, because we were looking for a human-understandable model that could guide our future thinking and provide insight into how we can provide effective and intelligent feedback to the user. The linear regression model produces not only a ranking but also an equation that can be easily integrated into our system to provide internal rankings. We were looking for more than just a number; we also wanted our model to help guide our HCI, UX, and future development so we could also provide helpful feedback based on the values for the features in the algorithm (such as "try drawing faster, and with more purpose next time"). An additional benefit is that this model can easily run in real time in javascript on an HTML browser without having to do any processing on the server, which could cause a delay. Since this simplest model does perform reasonably well, this implies that a more complex model, such as one that uses Random Forest or another classifier, and/or one that retrains itself as it gets more data, would do even better.

Table 12. Similarity of Rankings for Experts and the Computer

Lines S1																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−12.57	<0.000001
E_2	1	2	4	6	3	7	9	12	10	11	14	13	5	15	8	17	16	18	19	20		
E_{Avg}	1	2	3.5	5	4	6.5	8	10	9.5	10.5	12.5	12.5	9	14.5	11.5	16.5	16.5	18	19	20	−24.34	<0.01
C	1	7	13	11	3	5	8	10	15	4	17	12	14	16	2	20	6	19	9	18		

Lines S2																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−21.77	<0.005
E_2	1	2	3	5	15	16	7	9	8	4	12	14	11	17	19	6	18	10	13	20		
E_{Avg}	1	2	3	4.5	10	11	7	8.5	8.5	7	11.5	13	12	15.5	17	11	17.5	14	16	20	−24.53	<0.01
C	10	5	4	3	7	6	2	8	16	9	18	15	13	19	11	20	12	1	14	17		

Circles S1																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−14.42	<0.000005
E_2	1	3	2	11	4	10	8	7	6	16	9	5	13	12	15	20	17	14	19	18		
E_{Avg}	1	2.5	2.5	7.5	4.5	8	7.5	7.5	7.5	13	10	8.5	13	13	15	18	17	16	19	19	−20.25	<0.0005
C	4	3	1	7	6	14	17	10	12	5	8	2	13	16	20	15	9	11	19	18		

Circles S2																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−18.76	<0.0005
E_2	1	7	3	2	4	15	5	8	13	6	18	12	19	10	14	9	16	11	17	20		
E_{Avg}	1	4.5	3	3	4.5	10.5	6	8	11	8	14.5	12	16	12	14.5	12.5	16.5	14.5	18	20	−22.07	<0.005
C	6	16	15	3	13	4	8	5	7	1	20	17	10	11	2	12	9	18	19	14		

Rects S1																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−28.18	<0.05
E_2	2	9	13	14	4	5	6	12	1	16	10	11	15	17	7	19	8	18	3	20		
E_{Avg}	1.5	5.5	8	9	4.5	5.5	6.5	10	5	13	10.5	11.5	14	15.5	11	17.5	12.5	18	11	20	−14.30	<0.000005
C	1	8	3	4	7	6	9	5	2	16	12	13	19	15	11	14	10	20	17	18		

Computer values for Set 1 (S1) were produced by 20-fold cross-validation. Computer values for Set 2 (S2) used the equations from S1.

6.2.3 Equations Created. Using the data from rankings, we trained a linear classifier for each of the shapes and produced the equations listed in Equation (18),

$$\begin{aligned}
LineRank &= 0.5965 * avgDev \\
&\quad - 7.3061 * minSpeed + 8.0342 \\
CircleRank &= 0.3339 * ratioAvgDevToAvgSpeed \\
&\quad + 1.2463 \\
RectangleRank &= -48.1763 * minSpeed \\
&\quad + 14.4651 * avgSpeed + 0.3612 * avgSquareAngle \\
&\quad + 0.0137 * totalAbsAngle - 0.0008 * totalSquareAngle \\
&\quad - 11.1426 \\
EllipseRank &= -6.28 * avgSpeed \\
&\quad + 0.4381 * avgSquareAngle - 0.0742 * avgSqAngleToAvgSpeed \\
&\quad + 12.885
\end{aligned} \tag{18}$$

Table 13. Similarity of Rankings for Experts and the Computer

Rects S2																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−24.25	<0.01
E_2	1	12	13	5	3	7	6	8	10	14	15	9	11	4	18	16	2	17	19	20		
E_{Avg}	1	7	8	4.5	4	6.5	6.5	8	9.5	12	13	10.5	12	9	16.5	16	9.5	17.5	19	20	−27.29	<0.05
C	6	16	15	3	13	4	8	5	7	1	20	17	10	11	2	12	9	18	19	14		

Ellipses S1																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−16.37	<0.00005
E_2	1	8	6	4	2	11	7	9	3	10	5	17	14	20	12	13	18	19	16	15		
E_{Avg}	1	5	4.5	4	3.5	8.5	7	8.5	6	10	8	14.5	13.5	17	13.5	14.5	17.5	18.5	17.5	17.5	−26.91	<0.05
C	1	8	16	2	15	4	19	9	6	5	3	18	17	7	10	14	11	20	13	12		

Ellipses S2																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−13.56	<0.000001
E_2	3	1	4	8	6	5	11	9	2	10	17	7	16	13	14	12	15	18	20	19		
E_{Avg}	2	1.5	3.5	6	5.5	5.5	9	8.5	5.5	10	14	9.5	14.5	13.5	14.5	14	16	18	19.5	19.5	−27.46	<0.05
C	15	1	6	2	3	17	8	9	16	14	11	13	4	7	5	10	12	18	20	19		

Cuboids S1																					Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	−28.32	<0.05
E_2	4	1	7	5	19	11	14	12	9	15	2	8	6	18	17	10	16	3	13	20		
E_{Avg}	2.5	1.5	5	4.5	12	8.5	10.5	10	9	12.5	6.5	10	9.5	16	16	13	16.5	10.5	16	20	−11.20	<0.000001
C	3	1	2	4	15	12	11	8	7	14	6	9	5	13	17	19	20	10	16	18		

Cuboids S2																		—	—	—	Simil.	P
E_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	—	—	—	−20.59	<0.05
E_2	6	2	1	17	10	8	3	9	7	4	16	14	12	5	11	13	15	—	—	—		
E_{Avg}	3.5	2	2	10.5	7.5	7	5	8.5	8	7	13.5	13	12.5	9.5	13	14.5	16	—	—	—	−21.52	<0.05
C	9	5	7	2	3	6	10	4	14	8	16	12	17	15	1	11	13	—	—	—		

Computer values for Set 1 (S1) were produced by 20-fold cross-validation. Computer values for Set 2 (S2) used the equations from S1.

$$
\begin{aligned}
CubeRank = {} & 33.7661 * avgSpeed \\
& + 0.0848 * maxAngle + 10.5368 * avgAbsAngle \\
& - 0.436 * avgSquareAngle + -77.9688 * overdrawing \\
& - 2.7829 * strokeDirection + -23.2111 * ratioMiddle50ToLast25 \\
& + 0.4516 * ratioMaxDevToMaxSpeed + 91.1589.
\end{aligned}
$$

Because the data input to the system rank between 1 and 20, where 20 is worst, the generated linear regression equation maps onto a scale that is predominately between 1 and 20 for most feature values. However, with extreme feature values, the actual ends of the scale can be anywhere between $-\infty$ to ∞ depending on the feature values As we would ideally have a scale between 0 and 1, where 1 is best, we altered the equation such that the value v was mapped to $1 - v/20$. We also applied a max and min so that the value remained between 0 and 1 even in extreme cases, e.g., $1 - max(0, min(1, v/20))$. Equation (19) shows the normalized equations. While this may collapse values at the top and the bottom, this should still be okay for generalized improvement tracking,

$$\begin{aligned} LineRank &= 1 - max(1, min(0, (0.5965 * avgDev \\ &\quad - 7.3061 * minSpeed + 8.0342)/20)) \\ CircleRank &= 1 - max(1, min(0, (0.3339 * ratioAvgDevToAvgSpeed \\ &\quad + 1.2463)/20)) \\ RectangleRank &= 1 - max(1, min(0, (-48.1763 * minSpeed \\ &\quad + 14.4651 * avgSpeed + 0.3612 * avgSquareAngle \\ &\quad + 0.0137 * totalAbsAngle - 0.0008 * totalSquareAngle \\ &\quad - 11.1426)/20)) \\ EllipseRank &= 1 - max(1, min(0, (-6.28 * avgSpeed \\ &\quad + 0.4381 * avgSquareAngle - 0.0742 * avgSqAngleToAvgSpeed \\ &\quad + 12.885)/20)) \\ CubeRank &= 1 - max(1, min(0, (33.7661 * avgSpeed \\ &\quad + 0.0848 * maxAngle + 10.5368 * avgAbsAngle \\ &\quad - 0.436 * avgSquareAngle + -77.9688 * overdrawing \\ &\quad - 2.7829 * strokeDirection + -23.2111 * ratioMiddle50ToLast25 \\ &\quad + 0.4516 * ratioMaxDevToMaxSpeed + 91.1589)/20)). \end{aligned} \tag{19}$$

6.2.4 Testing Our Equations. Using our second set of completely new sketches ranked by the experts, we tested our equations to produce another computer ranking. We then computed a similarity metric between the computers rankings and the average rankings between the two experts.

Figures 19, 20, 21, 22, and 23 show some of the high- and low-scoring sketches for each of the shapes.

6.3 Discussion

The statistical analysis we performed using t-tests shows that most of the features for lines and rectangles are significant. Table 14 shows the features and how significant they were in differentiating experts and novices for the various forms. One interesting observation from the results is that the accuracy of the experts and novices for shapes including circles, ellipses, and cuboids does not differ significantly. This is congruent with the theory of design sketching. Design sketches are used to quickly get as many ideas as possible on the paper to share them with others, generating further ideas. Hence, experts do not always give attention to perfect accuracy and concentrate on other aspects of the sketch like the concept itself and line quality. Below are a few metrics that do matter when modeling expert sketching ability.

6.3.1 Accuracy. Accuracy may be dependent on the size of the shape. As the shapes increase in size, accuracy decreases because of the need to change the hand gesture to draw bigger strokes. The pen can be moved over screen by moving the wrist, elbow, or shoulder. Accuracy metrics are surprisingly poor differentiators for circles, ellipses, and cuboids. For circles and ellipses, this is because the simple shapes to not affect the overall perception of the three-dimensional shape. Rather, it is the surrounding lines that affect the shape more. Thus a visually appealing circle or ellipse only needs to be smooth, quickly drawn, and of circular shape. In terms of cuboids, we believe that viewers are accustomed to averaging/reconciling lines that go to the same vanishing point. But more importantly, we think that experts focus more on having lines going in the right general direction than exactly fitting the right end points.

Table 14. Summary of Features Used in the Generated Equations

	Lines	Circles	Rect.	Ellipses	Cuboids
Accuracy Metrics					
Average deviation					
Maximum deviation					
Hausdorff similarity					
Speed Metrics					
Maximum speed					
Minimum speed					
Average speed					
Accuracy Consistency (Smoothness) Metrics					
Maximum angle					
Average angle					
Average absolute angle					
Average square angle					
Total angle					
Total absolute angle					
Total squared angle					
Speed Consistency (Fluidity) Metrics					
Ratio of middle 50% to last 25% of stroke					
Ratio of first 75% to last 25% of stroke					
Planning Metrics					
Number of strokes	N/A				
Stroke order coupling	N/A	N/A		N/A	
Stroke direction coupling					
Overdrawing—Ratio of path length to ref. length					
Combined Metrics					
Ratio of average absolute angle to average speed					
Ratio of average square angle to average speed					
Ratio of average deviation to average speed					
Ratio of maximum deviation to maximum speed					
Ratio of average deviation to size of shape					

6.3.2 *Speed.* Speed was a strong feature across all shapes. The only anomaly is that the maximum speed was not significant in circles. However, the minimum speed was, implying that consistency is even more important than just simply going fast. When drawing straight lines there is a larger increase in speed, so higher maximum speed may be as a result from some straight line segments within the circle, leading to jerkier strokes. Experts tend to be much faster than novices, and this may be due to higher confidence and an ability to plan farther ahead.

6.3.3 *Average Square Angle.* The features that represent smoothness of strokes are better in experts than students. The average angle does not contribute to making the sketches of better quality; however, the average squared angle does. This is because average angle can be in both positive and negative quadrants, hence cancelling off when the stroke is wavy, whereas the average absolute angle measures overall waviness independent of the quadrant.

6.3.4 *Speed Fluidity.* Both features to measure speed fluidity perform fairly well in case of lines, rectangles, ellipses, and cubes. The t-values of these shapes for the features show that the speed decreases as a pen approaches the end points in cases of straight lines but increases for ellipses.

6.3.5 *Planning Metrics.* The features under planning perform moderately well for all shapes except for circles, although they are most valuable for cuboids.

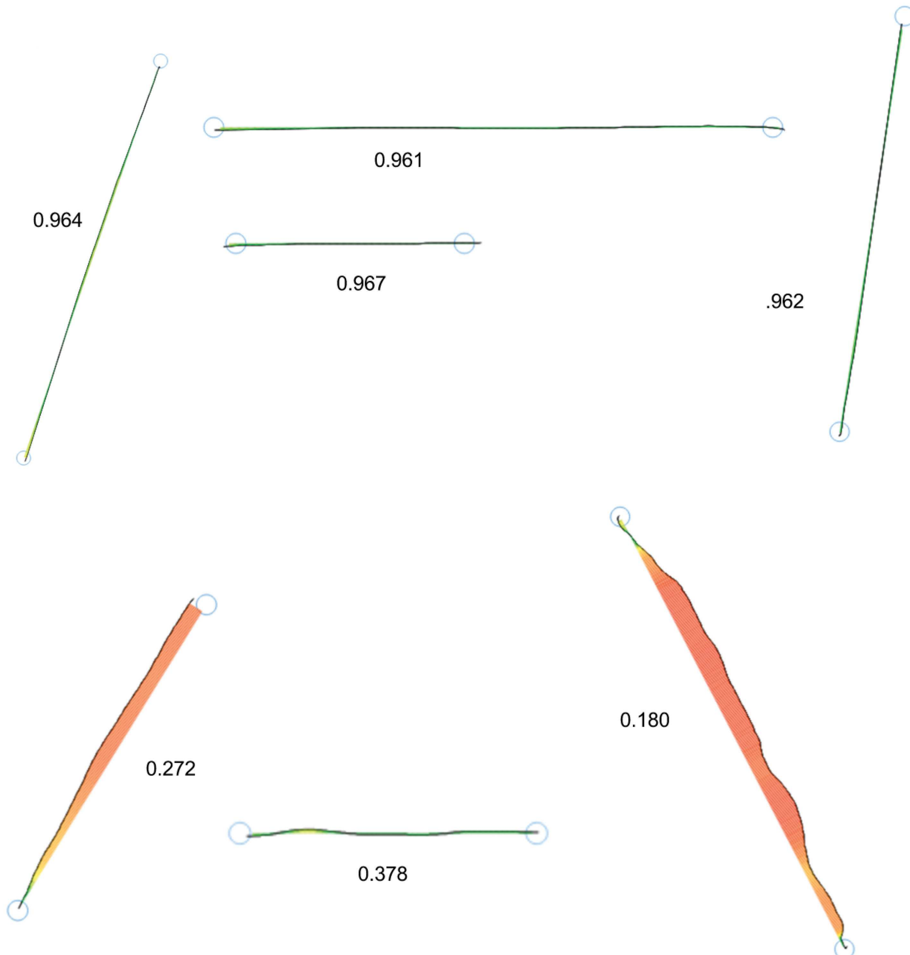

Fig. 19. High-score (top) and low-scoring lines (bottom). Note the importance of both accuracy and speed/fluidity. The top lines appear stronger and more confident, whereas the bottom lines show hesitation.

The number of strokes performs relatively well for rectangles, ellipses, and cuboids. The experts drew each of a rectangle or a cuboid with one stroke, whereas students drew several sides with a single stroke. For ellipses the opposite occurs, experts try to draw the whole shape with one stroke but the novice uses multiple strokes to finish the shape.

Stroke direction coupling was significant for all of the shapes with lines. Experts tended to draw all parallel lines in the same direction, whereas novices would draw them in different directions. This is again tied to the fact that novices would draw multiple lines with the same stroke.

Overdrawing, or the ratio between the stroke length and the reference length, also performs well for our dataset. Stroke order performs well in case of rectangles, confirming that the experts draw the parallel lines together.

6.3.6 Speed Ratios. Speed and smoothness or deviation together contribute to the visual clarity of sketches. To see how they perform together, we included several ratio features. Average square

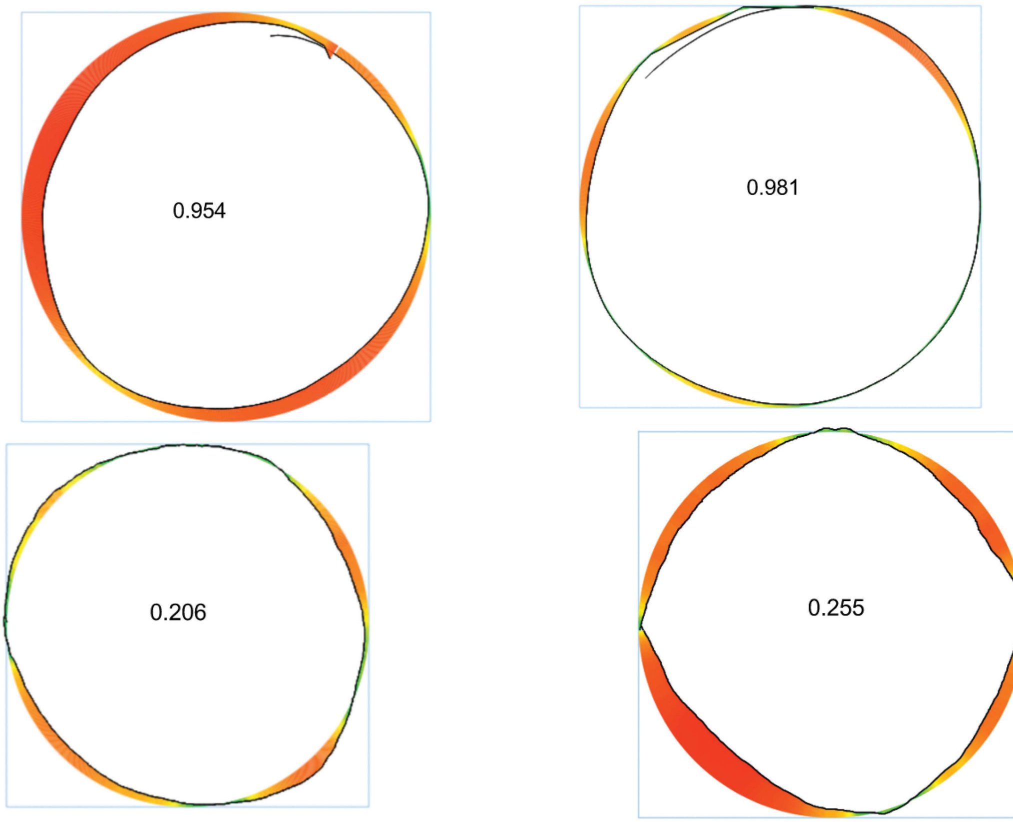

Fig. 20. The top two circles show high scores produced by drawn circles, whereas the bottom two show low scores. Note that for circles, both good and bad circles may have accuracy errors. You can see confidence and speed in the top two circles, whereas you can see hesitation in the bottom two circles.

angle to average speed, average deviation to average speed, and maximum deviation to maximum speed all perform particularly well. Speed, as mentioned above, is a particularly powerful distinguishing feature. Accuracy is affected by the speed with which one draws. On one hand, the faster one draws, the lower the accuracy. On the other hand, as one draws faster, fluidity and smoothness should improve. Ideally, one will improve both in accuracy and in speed, and these represent overall metrics that represent those improvements.

It is interesting that for circles, the single feature average deviation to average speed was enough to predict the ranking of sketches. And this feature was important for all shapes. Though accuracy of experts and students is not significantly different, the ratio of average deviation and average speed is significantly different. This proves that when students draw at a speed equal to the experts their accuracy will be worse than the experts and vice versa.

We also checked the ratio of other visual features (deviation and square angle) to speed. All of these features performed relatively well.

7 TAKE-AWAY POINTS

This set of features/groups of features helps in distinguishing experts from novices in sketching. One other finding that was not surprising was that as shape complexities increase, the experts

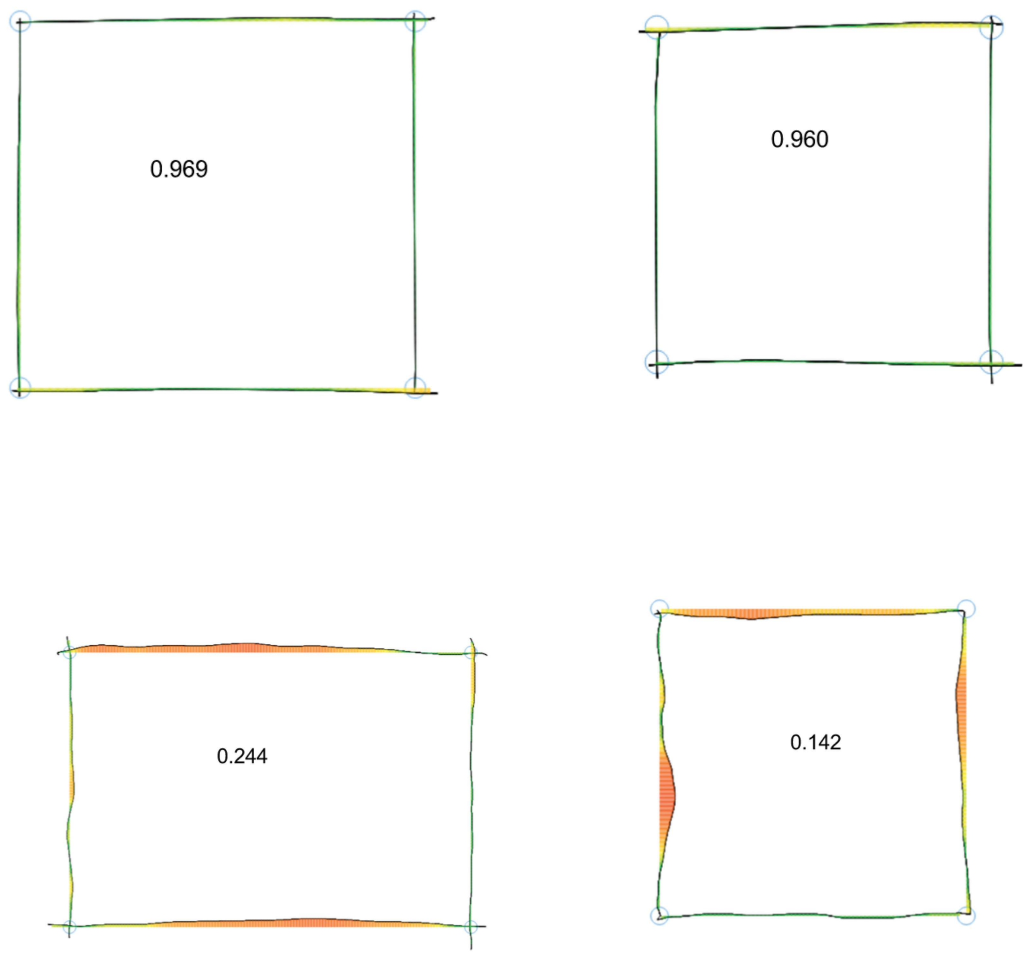

Fig. 21. The top two rectangles show high scores produced by drawn rectangles, whereas the bottom two show low scores. Again, notice the hesitation and disfluency in the bottom sketches.

perform much better than the novices, and there is significant difference in the sketches. The novices had difficulty drawing complex shapes like cuboids and ellipses, which require more practice.

From this information, we can begin modeling expert sketching ability and can develop an empirically based evaluation system.

Through the design of this system as well as the analysis of experts versus novices, we have learned several things, which we list here:

- UI Design—For effective practice, users should keep their hand in approximately the same place, this means that users should not have to click a submit button each time.
- UI Design—For effective practice, users should continuously sketch one shape after the next, with little to no wait time after each shape.
- UI Design—Any timeout would be too long for experts and too short for novices.
- UI Design—By automatically recognizing when the user has completed the shape, we can provide immediate personalized feedback the instant the user lifts up their pen.

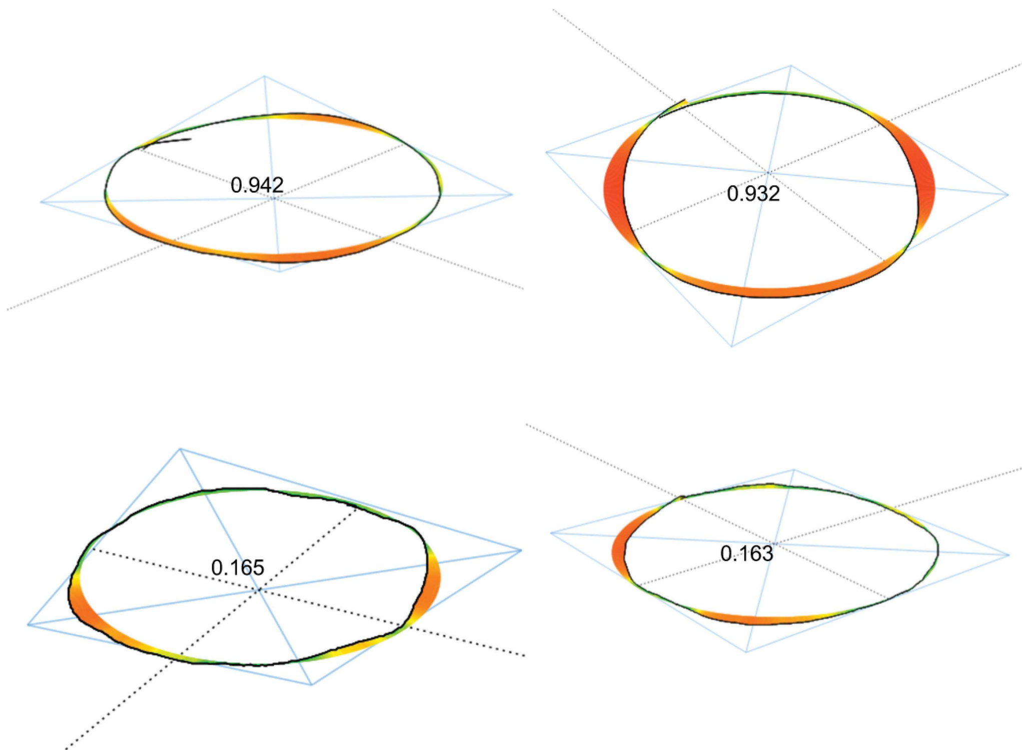

Fig. 22. High and low scores for ellipses. This figure is a good example of how much more fludity and speed matter more for ellipses than perfect accuracy.

- UI Design—For the same reasons, after displaying the feedback, the sketchbook should automatically progress to the next sketch after a minimal time for them to review the feedback (while also providing them away to go back and review that feedback as long as they want later in the lesson).
- Accuracy—While accuracy is certainly important, it is not the only measurement that distinguishes novices from experts. Speed and consistency features are sometimes even more important.
- Accuracy—Accuracy is not a distinguishing feature for circles. We think this is because when looking at a drawing in two-point perspective, lines play a much greater perceptual part in harmonious viewing of the image, as it is harder to use circles to gain perspective. Speed and consistency continues to play an important part.
- Planning—Experts tend to draw rectangles in four strokes. Novices tend to draw rectangles in a continuous stroke, as they think of rectangles as four 90° angles. Novices draw the shape as they perceive it and how it exists in real life. Experts draw the shape as they actually visualize it, in which a rectangle is drawn as two sets of parallel lines that extend to a vanishing point. Similar things happen with cubes to an extent, with novices using fewer strokes and a more haphazard manner and experts grouping their strokes to parallel lines. However, due to the large number of lines in a cube, experts may switch between groups of parallel lines, and this is not captured in stroke order; rather, another feature would need to be invented to properly capture that.

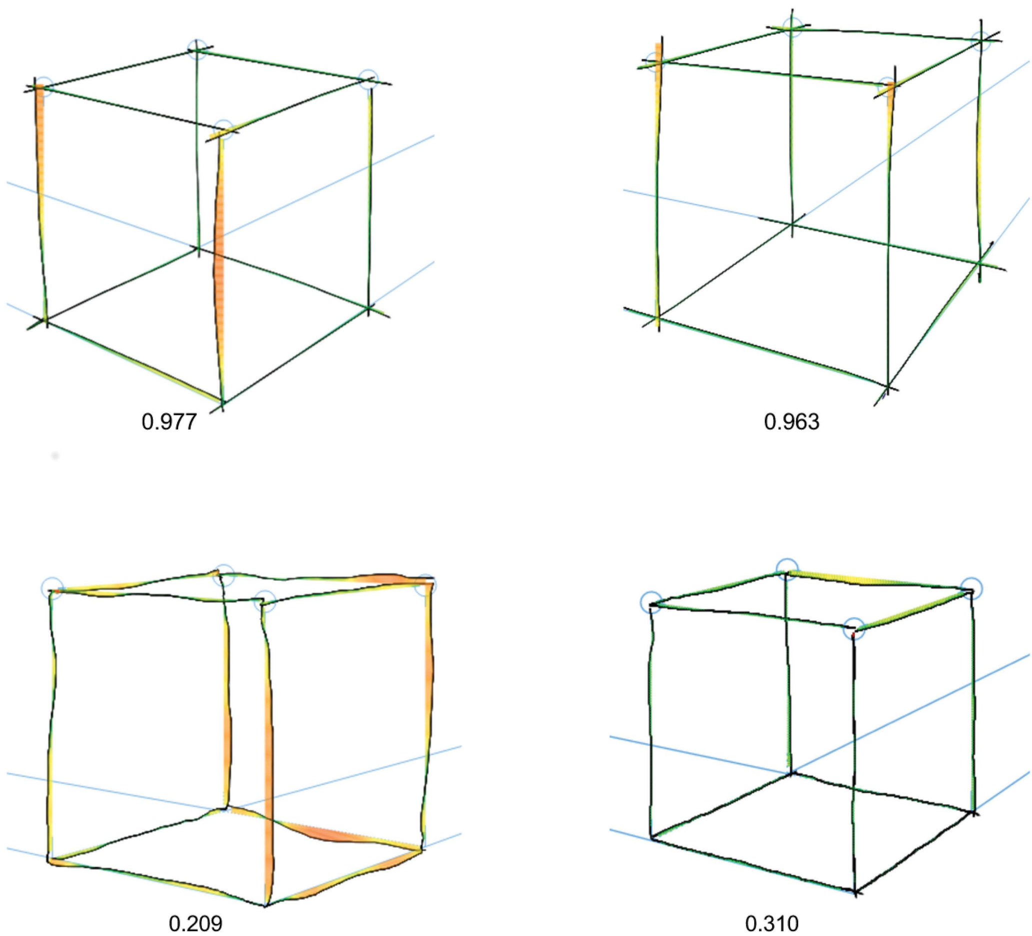

Fig. 23. High and low scores for cuboids. Even though the top and bottom cubes have a similar amount of *red*, drawing confidence, speed, and fluidity come across strongly in the top two figures.

- Planning—Stroke direction is very dependent on handedness, so any model that would include that feature needs to take that into account. That said, experts tended to be consistent. Their direction related to the vanishing points, and this was seen in the cuboid section.
- Recognition—Just as sketch recognition systems benefit from visual, gestural, and perception/geometric techniques, evaluating goodness of drawing also benefits from the same three categories of features.
- Single Score—While a single score is not necessarily helpful for a student who needs targeted feedback, it is very useful for the system to be able to weigh tradeoffs as the user starts to pick up speed and loses accuracy or vice versa. The single score helps quantify improvement even as some metrics are decreasing and others are increasing.

8 CONCLUSION

As design sketching is widely used to quickly and effectively communicate ideas in a wide range of disciplines, it is commonly taught in universities. Traditionally, students are taught in a typical classroom setting, where they may not receive significant amounts of personalized feedback due to the size of the class and the limited availability of the instructor. To explore the possibility of using *SketchTivity*, a web-based intelligent tutoring system, to provide users with personalized feedback we develop a single internal score representative of drawing ability. To do this, we extracted a set of features from samples belonging to five shape types, *lines*, *circles*, *ellipses*, *squares*, and *cuboids*, and processed the significant features through Linear Regression to produce an equation designed to evaluate the quality of the sketch. When the equation results are compared against analyses completed by expert human sketchers, we found that our algorithm can statistically significantly evaluate the quality of a sketch.

9 FUTURE WORK

This work provides insight into how novices' sketches differ from experts' as well as a possible metric for a holistic internal computer score to represent overall progress. By noting which metrics distinguish experts and novices, we can identify certain occurrences and provide helpful feedback phrases to use when appropriate, such as, "We noticed that you are stopping at the end point for each stroke. Attempting to sketch through the end point may make your sketches more fluid." The internal holistic score will help measure overall progress, as encouraging sketchers to increase speed may initially decrease accuracy, and vice versa. By combining those two things together, we hope to use this information to provide personalized informative feedback to help improve progress. This work also forms an empirically based starting point to introduce fun gaming mechanics to the software that may improve student motivation and self-efficacy.

Pressure is also an important criteria in sketching to produce quality sketches. In the future, a pressure-sensitive device can be used to get pressure and line-weight-related features as well. Signature verification is a field that is similar to sketch recognition and user identification in *SketchTivity*. Extensive research has been done in this field, not only on recognizing the distinct identity [19] of the sketcher but also on recognizing the age and gender of the sketcher [43, 44]. Pressure and other features, such as entropy [10], should be tested to see if they can be used to improve our results. Furthermore, such algorithms can be used in implementing a sketch-based biometric system.

9.1 Converting Quality Measurements to Instructive Feedback

While our current work does not fully convert quality measurements to intelligent instructive feedback, it is a necessary first step towards that goal. Some future ways in which this can be achieved include the following:

- Being transparent to users about their performance in the metrics most directly relevant to expert sketching ability like accuracy, smoothness, and speed and displaying this information in scales after exercises. These data can also be mapped over time to show changes in performance in the individual metrics.
- Utilizing additional metrics like stroke order, pressure, and so on, to provide humanized feedback in the form of dialogue and a "virtual" instructor.
- Taking advantage of metrics that cannot be obtained without sketch recognition to provide feedback that goes beyond what human instructors are capable of.
- Utilizing relationships between metrics to construct higher-level aggregate metrics that can more directly lead to instructive feedback, e.g., the positive correlation between speed and smoothness can lead to a single metric called line quality and dialogue, such as "your line quality is poor, try sketching a little faster!"

- Testing the various feedback based on the feature values and evaluating how it effects student performance.
- Obtaining more qualitative and quantitative grading data from experts, both in the form of ranking, but more importantly in terms of textual feedback that could help in designing the feedback of our system so that the words used by real human instructors could be used to guide the feedback. We could also codify this real feedback using machine learning and build classifiers for automated feedback in the system.

ACKNOWLEDGMENTS

The authors thank Ethan Hilton for providing his insights and helping in data collection for the experiments. The authors also thank Paul Taele, Dr. Jeffrey Liew, and Dr. Erin McTigue for giving valuable feedback that assisted the project. We also thank the many sketching and drawing instructors who have provided valuable guidance on the design of this system as well as members of the Sketch Recognition Lab who provided feedback on this article.

REFERENCES

[1] Johnny Accot and Shumin Zhai. 1997. Beyond fitts' law: Models for trajectory-based HCI tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*. ACM, New York, NY, 295–302. DOI: http://dx.doi.org/10.1145/258549.258760

[2] Olufunmilola Atilola, Martin Field, Erin McTigue, Tracy Hammond, and Julie Linsey. 2011. Evaluation of a natural sketch interface for truss FBDs and analysis. In *Proceedings of the 2011 Frontiers in Education Conference (FIE'11)*. IEEE Computer Society, Washington, DC, S2E–1–1–S2E–6. DOI: http://dx.doi.org/10.1109/FIE.2011.6142959

[3] Olufunmilola Atilola, Martin Field, Erin McTigue, Tracy Hammond, and Julie Linsey. 2011. Mechanix: A sketch recognition truss tutoring system. In *American Society of Mechanical Engineers (ASME) 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 7: 5th International Conference on Micro- and Nanosystems; 8th International Conference on Design and Design Education; 21st Reliability, Stress Analysis, and Failure Prevention Conference*, Vol. 7. ASME, Washington, DC, 645–654.

[4] Olufumilola Atilola, Erin M. McTigue, Tracy Hammond, and Julie Linsey. 2013. Mechanix: Evaluating the effectiveness of a sketch recognition truss tutoring program against other truss programs. In *Proceedings of the 120th American Society for Engineering Education Annual Conference & Exposition (ASEE'13)*. ASEE, 169–192.

[5] Olufunmilola Atilola, Stephanie Valentine, Hong-Hoe Kim, David Turner, Erin McTigue, Tracy Hammond, and Julie Linsey. 2014. Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. *Artif. Intell. Eng. Des. Anal. Manufact.* 5, 2 (2014), 169–192. DOI: http://dx.doi.org/10.1017/S0890060414000079

[6] Olufunmilola Atilola, Francisco Vides, Erin M. Mctigue, Julie S. Linsey, and Tracy Anne Hammond. 2012. Automatic identification of student misconceptions and errors for truss analysis. In *Proceedings of the 119th American Society for Engineering Education Annual Conference & Exposition (ASEE'12)*. ASEE. https://peer.asee.org/21004.

[7] Robert L. Bangert-Drowns, Chen-Lin C. Kulik, James A. Kulik, and MaryTeresa Morgan. 1991. The instructional effect of feedback in testlike events. *Rev. Educ. Res.* 61, 2 (1991), 213–238. DOI: http://dx.doi.org/10.3102/00346543061002213 arXiv:http://rer.sagepub.com/content/61/2/213.full.pdf+html

[8] Laura Barreto, Paul Taele, and Tracy Hammond. 2016. A stylus-driven intelligent tutoring system for music education instruction. In *Revolutionizing Education with Digital Ink: The Impact of Pen and Touch Technology on Education*, Tracy Hammond, Stephanie Valentine, and Aaron Adler (Eds.). Springer International Publishing, Cham, 141–161. DOI: http://dx.doi.org/10.1007/978-3-319-31193-7_10

[9] Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. 2014. Painting with bob: Assisted creativity for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST'14)*. ACM, New York, NY, 419–428. DOI: http://dx.doi.org/10.1145/2642918.2647415

[10] Akshay Bhat and Tracy Hammond. 2009. Using entropy to distinguish shape versus text in hand-drawn diagrams. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1395–1400. http://dl.acm.org/citation.cfm?id=1661445.1661669.

[11] Gerard Biau. 2012. Analysis of a random forests model. *Journal of Machine Learning Research* 13 (2012), 1063–1095.

[12] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

[13] Hsiang Chen, R. T. Wigand, and M. S. Nilan. 1999. Optimal experience of web activities. *Comput. Hum. Behav.* 15, 5 (1999), 585–608. DOI: http://dx.doi.org/10.1016/S0747-5632(99)00038-2

[14] Danielle Cummings, Francisco Vides, and Tracy Hammond. 2012. I don't believe my eyes!: Geometric sketch recognition for a computer art tutorial. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (SBIM'12)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 97–106.

[15] Karen Ernst daSilva. 2001. Drawing on experience: Connecting art and language. *Primary Voices K-6* 10, 2 (2001), 2–9.

[16] Ruwanee de Silva, David Tyler Bischel, WeeSan Lee, Eric J. Peterson, Robert C. Calfee, and Thomas F. Stahovich. 2007. Kirchhoff's pen: A pen-based circuit analysis tutor. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling (SBIM'07)*. ACM, New York, NY, 75–82. DOI: http://dx.doi.org/10.1145/1384429.1384447

[17] Daniel Dixon, Manoj Prasad, and Tracy Hammond. 2010. iCanDraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, New York, New York, 897–906.

[18] Koos Eissen and Roselien Steur. 2009. *Sketching: Drawing Techniques for Product Designers* (12th ed.). BIS Publishers, Amsterdam, The Netherlands.

[19] Brian David Eoff and Tracy Hammond. 2009. Who dotted that'I'?: Context free user differentiation through pressure and tilt pen data. In *Proceedings of the Graphics Interface 2009 Conference (GI'09)*. Canadian Information Processing Society, Toronto, Ontario, Canada, 149–156. http://dl.acm.org/citation.cfm?id=1555880.1555916.

[20] Martin Field, Stephanie Valentine, Julie Linsey, and Tracy Hammond. 2011. Sketch recognition algorithms for comparing complex and unpredictable shapes. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Vol. 3. AAAI Press, 2436–2441. DOI: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-406

[21] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psychol.* 47, 6 (1954), 381.

[22] Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.* 5, 7 (1985), 1688–1703.

[23] Leslie Gennari, Levent Burak Kara, Thomas F Stahovich, and Kenji Shimada. 2005. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Comput. Graph.* 29, 4 (2005), 547–562.

[24] Matthew G. Green, Benjamin W. Caldwell, Michael Helms, Julie S. Linsey, and Tracy Anne Hammond. 2015. Using natural sketch recognition software to provide instant feedback on statics homework (truss free body diagrams): Assessment of a classroom pilot. In *Proceedings of the 2015 ASEE Annual Conference and Exposition*. ASEE, 186–198.

[25] Tracy Hammond and Randall Davis. 2003. LADDER: A language to describe drawing, display, and editing in sketch recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*. Morgan Kaufmann Publishers, 461–467. http://dl.acm.org/citation.cfm?id=1630659.1630728.

[26] Tracy Hammond and Randall Davis. 2004. Automatically transforming symbolic shape descriptions for use in sketch recognition. In *Proceedings of the 19th National Conference on Artifical Intelligence (AAAI'04)*. AAAI Press, San Jose, CA, 450–456. http://dl.acm.org/citation.cfm?id=1597148.1597222.

[27] Tracy Hammond and Randall Davis. 2005. LADDER, a sketching language for user interface developers. *Comput. Graph.* 29, 4 (2005), 518–532. https://doi.org/10.1016/j.cag.2005.05.005.

[28] Tracy Hammond, Drew Logsdon, Brandon Paulson, Joshua Johnston, Joshua Peschel, Aaron Wolin, and Paul Taele. 2010. A sketch recognition system for recognizing free-hand course of action diagrams. In *Innovative Applications of Artificial Intelligence*. AAAI, 1781–1786. http://www.aaai.org/ocs/index.php/IAAI/IAAI10/paper/view/1581.

[29] Tracy Hammond, Drew Logsdon, Joshua Peschel, Joshua Johnston, Paul Taele, Aaron Wolin, and Brandon Paulson. 2010. A sketch recognition interface that recognizes hundreds of shapes in course-of-action diagrams. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA'10)*. ACM, New York, NY, 4213–4218. http://dx.doi.org/10.1145/1753846.1754128.

[30] Tracy Hammond and Brandon Paulson. 2011. Recognizing sketched multistroke primitives. *ACM Trans. Interact. Intell. Syst.* 1, 1, Article 4 (Oct. 2011), 34 pages. http://dx.doi.org/10.1145/2030365.2030369.

[31] Jay D. Helsel. 1979. *Reading Engineering Drawings Through Conceptual Sketching* (1st ed.). Glencoe/Mcgraw–Hill, Columbus, OH, USA.

[32] Ethan Hilton, Wayne Li, Tracy Hammond, and Julie Linsey. 2018. Effectively teaching sketching in engineering curricula. (unpublished).

[33] Ethan Hilton, Wayne Li, Sunni H. Newton, Meltem Alemdar, Raghuram Pucha, and Julie Linsey. 2016. The development and effects of teaching perspective free-hand sketching in engineering design. In *Proceedings of the American Society of Mechanical Engineers International Design & Engineering Technical Conferences and Computers & Information in Engineering Conference (ASME IDETC/CIE'16)*. ASME. DOI: http://dx.doi.org/10.1115/DETC2016-60250

[34] Ethan Hilton, Myela Paige, Blake Williford, Wayne Li, Tracy Hammond, and Julie Linsey. 2017. Engineering drawing for the next generation: Students gaining additional skills in the same timeframe. In *American Society for Engineering Education Annual Conference (ASEE)*. ASEE, 1–12. https://peer.asee.org/27874.

[35] Ethan Hilton, Myela Paige, Blake Williford, Wayne Li, Tracy Hammond, and Julie Linsey. 2017. Improving the sketching ability of engineering design students. In *Proceedings of the International Conference on Engineering Design (ICED'17)*, Vol. 9: Design Education. The Design Society, 217–224.

[36] Ethan Hilton, Blake Williford, Wayne Li, Erin McTigue, and Tracy Hammond. 2016. Developing a consistent method for evaluating sketching ability. In *Proceedings of the 4th International Conference on Design Creativity*. The Design Society, Article 43, 9 pages.

[37] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. 2013. The drawing assistant: Automated drawing guidance and feedback from photographs. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST'13)*. ACM, New York, New York, 183–192.

[38] Shari L. Jackson, Joseph Krajcik, and Elliot Soloway. 1998. The design of guided learner-adaptable scaffolding in interactive learning environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, 187–194.

[39] Elmer E. Jones. 1922. The correlation of visual memory and perception of perspective with drawing ability. *School Soc.* 15 (1922), 174–176.

[40] Jr. Joseph J. LaViola and Robert C. Zeleznik. 2004. MathPad2: A system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH'04)*. ACM, New York, NY, 432–440.

[41] Levent Burak Kara and Thomas F. Stahovich. 2005. An image-based, trainable symbol recognizer for hand-drawn sketches. *Comput. Graph.* 29, 4 (2005), 501–517.

[42] Swarna Keshavabhotla, Blake Williford, Ethan Hilton, Paul Taele, Wayne Li, Julie Linsey, and Tracy Hammond. 2017. Conquering the cube: Learning to sketch primitives in perspective with an intelligent tutoring system. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling (SBIM'17)*. ACM.

[43] Hong-Hoe Kim, Paul Taele, Stephanie Valentine, Jeff Liew, and Tracy Hammond. 2014. Developing intelligent sketch-based applications to support children's self-regulation and school readiness. In *Proceedings of the 2014 Intelligent User Interfaces Workshop on Sketch Recognition*. ACM, Haifa, Israel, 1–8.

[44] Hong-Hoe Kim, Stephanie Valentine, Paul Taele, and Tracy Hammond. 2015. EasySketch: A sketch-based fine motor skill recognizing educational interface for children emerging technology research strand. In *The Impact of Pen and Touch Technology on Education* (1st ed.), Tracy Hammond, Stephanie Valentine, Aaron Adler, and Mark Payton (Eds.). Springer Publishing Company, Heidelberg, Germany, 35–46.

[45] Joseph A. Koncelik and Kevin Reeder. 2008. *Conceptual Drawing* (1 ed.). Course Technology, Boston, MA.

[46] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. Shadowdraw: Real-time user guidance for freehand drawing. In *ACM Transactions on Graphics (TOG)*, Vol. 30-4. ACM, Association for Computing Machinery, New York, NY, 27.

[47] Wayne Li, Ethan Hilton, Tracy Hammond, and Julie Linsey. 2016. Persketchtivity: An intelligent pen-based online education platform for sketching instruction. In *Proceedings of the British Computer Society Electronic Visualisation and the Arts (BCS EVA'16)*. BCS, London, England, 133–141. DOI: http://dx.doi.org/10.14236/ewic/EVA2016.28

[48] Yong-Jin Liu, Xi Luo, Ajay Joneja, Cui-Xia Ma, Xiao-Lan Fu, and Dawei Song. 2013. User-adaptive sketch-based 3-D CAD model retrieval. *IEEE Trans. Automat. Sci. Eng.* 10, 3 (2013), 783–795.

[49] Yong-Jin Liu, Cui-Xia Ma, Qiufang Fu, Xiaolan Fu, Sheng-Feng Qin, and Lexing Xie. 2014. A sketch-based approach for interactive organization of video clips. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 1 (2014), 2.

[50] George Lucchese, Martin Field, Jimmy Ho, Ricardo Gutierrez-Osuna, and Tracy Hammond. 2012. GestureCommander: Continuous touch-based gesture prediction. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems (CHI EA'12)*. ACM, New York, NY, 1925–1930. DOI: http://dx.doi.org/10.1145/2212776.2223730

[51] Cui-Xia Ma, Yong-Jin Liu, Hong-An Wang, Dong-Xing Teng, and Guo-Zhong Dai. 2012. Sketch-based annotation and visualization in video authoring. *IEEE Trans. Multimedia* 14, 4 (2012), 1153–1165.

[52] Cui-Xia Ma, Yong-Jin Liu, Hai-Yan Yang, Dong-Xing Teng, Hong-An Wang, and Guo-Zhong Dai. 2011. KnitSketch: A sketch pad for conceptual design of 2D garment patterns. *IEEE Trans. Automat. Sci. Eng.* 8, 2 (2011), 431–437.

[53] David E. Meyer, J. E. Keith-Smith, Sylvan Kornblum, Richard A. Abrams, and Charles E. Wright. 1990. Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In *Attention and Performance 13: Motor Representation and Control*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 173–226.

[54] Erik G. Miller, Nicholas E. Matsakis, and Paul A. Viola. 2000. Learning from one example through shared densities on transforms. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 464–471. DOI: http://dx.doi.org/10.1109/CVPR.2000.855856

[55] Trevor Nelligan, Seth Polsley, Jaideep Ray, Michael Helms, Julie Linsey, and Tracy Hammond. 2015. Mechanix: A sketch-based educational interface. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion (IUI Companion'15)*. ACM, New York, NY, 53–56. DOI: http://dx.doi.org/10.1145/2732158.2732194

[56] Luke Olsen, Faramarz F. Samavati, Mario Costa Sousa, and Joaquim A. Jorge. 2009. Sketch-based modeling: A survey. *Comput. Graph.* 33, 1 (2009), 85–103.

[57] Jane Parker. 2005. A consideration of the relationship between creativity and approaches to learning in art and design. *Int. J. Art Des. Educ.* 24, 2 (2005), 186–198.

[58] Brandon Paulson and Tracy Hammond. 2007. A system for recognizing and beautifying low-level sketch shapes using NDDE and DCR. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'07)*. ACM, Newport, RI, 63–64.
[59] Brandon Paulson and Tracy Hammond. 2008. PaleoSketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI'08)*. ACM, New York, NY, 1–10. DOI: http://dx.doi.org/10.1145/1378773.1378775
[60] Brandon Paulson, Pankaj Rajan, and Tracy Hammond. 2008. What!?! No rubine features?: Using geometric-based features to produce normalized confidence values for sketch recognition. In *Proceedings of the HCC Workshop: Sketch Tools for Diagramming*. VL/HCC, Herrsching am Ammersee, Germany, 57–63.
[61] Dan Roam. 2013. *The Back of the Napkin: Solving Problems and Selling Ideas with Pictures* (1 ed.). Portfolio Publishing, London, England, United Kingdom.
[62] Dean Rubine. 1991. Specifying gestures by example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'91)*. ACM, New York, NY, 329–337. DOI: http://dx.doi.org/10.1145/122718.122753
[63] Irene Schiferl. 2008. Both sides now: Visualizing and drawing with the right and left hemispheres of the brain. *Stud. Art Educ.* 50, 1 (2008), 67–82.
[64] Tevfik Metin Sezgin and Randall Davis. 2007. Scale-space based feature point detection for digital ink. In *Proceedings of the ACM SIGGRAPH 2007 Courses (SIGGRAPH'07)*. ACM, New York, NY. DOI: http://dx.doi.org/10.1145/1281500.1281547
[65] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. 2006. Sketch based interfaces: Early processing for sketch understanding. In *Proceedings of the ACM SIGGRAPH 2006 Courses (SIGGRAPH'06)*. ACM, New York, NY, Article 22, 8 pages. DOI: http://dx.doi.org/10.1145/1185657.1185783
[66] Herbert A. Simon. 1983. Why should machines learn? In *Machine Learning*, Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell (Eds.). Springer, Berlin, 25–37. DOI: http://dx.doi.org/10.1007/978-3-662-12405-5_2
[67] Klara Sjölén and Allan MacDonald. 2011. *Learning Curves: An Inspiring Guide to Improve Your Design Sketch Skills*. KEEOS Design Books, Sundsvall, Sweden.
[68] Sheryl Sorby. 2009. Educational research in developing 3D spatial skills for engineering students. *Int. J. Sci. Educ.* 31, 3 (Feb. 2009), 459–480.
[69] Paul Taele, Laura Barreto, and Tracy Hammond. 2009. Hashigo: A next-generation sketch interactive system for japanese kanji. In *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference (IAAI'15)*. AAAI, Palo Alto, CA, 153–158.
[70] Paul Taele, Laura Barreto, and Tracy Hammond. 2015. Maestoso: An intelligent educational sketching tool for learning music theory. In *Proceedings of the 27th Innovative Applications of Artificial Intelligence Conference (IAAI'15)*. AAAI, Palo Alto, CA, 3999–4005.
[71] Paul Taele and Tracy Hammond. 2010. LAMPS: A sketch recognition-based teaching tool for mandarin phonetic symbols I. *J. Vis. Lang. Comput.* 21, 2 (Apr. 2010), 109–120. DOI: http://dx.doi.org/10.1016/j.jvlc.2009.12.004
[72] Stephanie Valentine, Raniero Lara Garduno, Julie Linsey, and Tracy Hammond. 2015. Mechanix: A sketch-based tutoring system that automatically corrects hand-sketched statics homework. In *The Impact of Pen and Touch Technology on Education*, A. Adler T. Hammond, S. Valentine and M. Payton (Eds.). Springer, Switzerland, Chapter 9, 91–103. http://dx.doi.org/10.1007/978-3-319-15594-4_9.
[73] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. 2012. Mechanix: A sketch-based tutoring system for statics courses. In *Proceedings of the 24th Innovative Applications of Artificial Intelligence Conference (IAAI'12)*. AAAI, Palo Alto, CA, 2253–2260.
[74] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong-hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. 2013. Mechanix: A sketch-based tutoring and grading system for free-body diagrams. *AI Mag.* 34, 1 (2013), 55–66.
[75] Antony White. 1990. *Harrap's Illustrated Dictionary of Art & Artists*. Harrap's Reference, Edin.
[76] Blake Williford, Wayne Li, Ethan Hilton, Julie Linsey, and Tracy Hammond. 2017. Everyone can sketch: Reviving hand-drawn sketching with an intelligent tutoring system. In *Extended Abstract for Workshop SketchingDIS: Hand-drawn Sketching in HCI Workshop at Designing Interactive Systems Conference (DIS'17)*. ACM.
[77] Blake Williford, Matthew Runyon, Adil Malla, Wayne Li, Julie Linsey, and Tracy Hammond. 2017. ZenSketch: A sketch-based game for improving line work. In *Proceedings of the ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'17)*. ACM, 591–598.
[78] Blake Williford, Paul Taele, Trevor Nelligan, Wayne Li, Julie Linsey, and Tracy Hammond. 2016. PerSketchTivity: An intelligent pen-based educational application for design sketching instruction. In *Revolutionizing Education with Digital Ink: The Impact of Pen and Touch Technology on Education*, Tracy Hammond, Stephanie Valentine, and Aaron Adler (Eds.). Springer International Publishing, Cham, 115–127. DOI: http://dx.doi.org/10.1007/978-3-319-31193-7_8

[79] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST'07)*. ACM, New York, NY, 159–168. DOI: http://dx.doi.org/10.1145/1294211.1294238

[80] Aaron Wolin, Brian Eoff, and Tracy Hammond. 2008. ShortStraw: A simple and effective corner finder for polylines. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling (SBM'08)*. Eurographics Association, 33–40. DOI: http://dx.doi.org/10.2312/SBM/SBM08/033-040

[81] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch: Face sketching assistance for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST'14)*. ACM, New York, New York, 407–417.

Received November 2016; revised September 2017; accepted January 2018