Recognizing Perspective Accuracy: An Intelligent User Interface for Assisting Novices

Blake Williford
Sketch Recognition Lab
Texas A&M University
bwilliford@gmail.com

Matthew Runyon
Sketch Recognition Lab
Texas A&M University
matrunyon@tamu.edu

Tracy Hammond
Sketch Recognition Lab
Texas A&M University
hammond@tamu.edu

Figure 1: Our algorithm and intelligent interface can provide real-time assistance to novices trying to sketch in perspective. In the example above, incorrect convergence is happening in the roofs of the buildings above the horizon, and in the sidewalk lines (both common novice mistakes). Awareness of the mistakes during the sketch can help novices learn perspective and produce more accurate sketches.

ABSTRACT

Sketching in perspective is a valuable skill for art, and for professional disciplines like industrial design, architecture, and engineering. However, it tends to be a difficult skill to grasp for novices. We have developed an algorithm that can classify rectilinear perspective strokes, as well as classify which of those strokes are accurate on a stroke-by-stroke basis. We also developed an intelligent user interface which can provide real-time accuracy feedback on a user's free-hand digital perspective sketch. To evaluate the system, we conducted a between-subjects user study with 40 novice participants which involved sketching city street corners in 2-point perspective. We discovered that the participants who received real-time intelligent feedback improved their perspective accuracy significantly ($p < 0.0005$) in a subsequent unassisted sketch, suggesting a mild learning effect and knowledge transfer. Through qualitative feedback we

also derived principles for how intelligent real-time feedback can best assist novices in learning perspective.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; **Interaction techniques**; • **Applied computing** → **Computer-assisted instruction**;

KEYWORDS

Human-Computer Interaction; Drawing; Sketching; Art Education; Design Education; Sketch Recognition; User Experience Design; User Interface Design; Intelligent User Interface

ACM Reference format:

Blake Williford, Matthew Runyon, and Tracy Hammond. 2020. Recognizing Perspective Accuracy: An Intelligent User Interface for Assisting Novices. In *Proceedings of 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17–20, 2020 (IUI '20)*, 12 pages.
<https://doi.org/10.1145/3377325.3377511>

1 INTRODUCTION

Drawing is a valuable skill to learn for a wide variety of reasons. It can benefit internal cognition and problem solving ability [26]. It can improve peripheral skills like writing, brainstorming, and visual communication [4]. It can even promote better overall academic achievement in some students [36]. In this work we focus on a practical form of drawing commonly known as design sketching or just *sketching*, and specifically on perspective sketching.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IUI '20, March 17–20, 2020, Cagliari, Italy

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7118-6/20/03...\$15.00

<https://doi.org/10.1145/3377325.3377511>

Sketching in perspective is a core skill that artists, designers, architects, and engineers should learn in order to depict objects in a manner that closely resembles how reality is seen by the human eye. It tends to be a skill that novices struggle grasping however, and as a result there is much focus on it in the educational literature [16, 23], as well as in curricula that emphasize sketching skills [12, 13]. Our discussions with domain experts with teaching experience also confirmed that perspective tends to be the skill that beginners struggle with the most.

Our goal with this research was to develop an intelligent user interface that could help novices grasp the concepts rectilinear perspective more readily with their own digital sketches and in real-time. We accomplished this via a perspective accuracy recognition algorithm and technology probes which explore different ways of giving real-time feedback to users. The algorithm can classify perspective strokes versus non-perspective strokes, as well as classify which perspective strokes are accurate and which are not. The algorithm does this on a stroke-by-stroke basis. We also wanted to determine if sketching novices could improve their perspective drawings by using the feedback and which form of feedback was most useful to them. The research questions associated with this effort were:

R1—*Can novices improve their sketching perspective accuracy from exposure to intelligent feedback driven by the algorithm? Can it transfer to subsequent sketches?*

R2—*What are the advantages and disadvantages of real-time versus summative feedback in this context, and which form of feedback is most useful to novices?*

Our contributions include:

- A general-purpose rectilinear perspective accuracy sketch recognition algorithm with strong performance and novel sketch recognition features.
- The promising results of a user study and exploration of providing intelligent real-time feedback to novices using the algorithm. On average, participants could improve their perspective accuracy by a statistically significant amount by using the intelligent feedback, and this improvement carried over when the feedback was removed.
- Insights about the optimal way to utilize the different forms of feedback for faded scaffolding to best help novices learn perspective sketching skills.

2 RELATED WORK

There have been a variety of sketch-based intelligent tutoring systems [27], tutorial generation tools, and creativity-support tools [25] aimed at helping people learn drawing skills in recent years. This section explores the most relevant

ones and how this work improves on and expands such directions.

Sketch-based intelligent Tutoring Systems

Numerous sketch-based intelligent tutoring systems have explored helping people improve drawing skills, but many of them have only explored specific domains and subjects. *iCanDraw* [5] was a system that utilized sketch recognition to assist novices in drawing human faces. Similarly, *EyeCanDraw* [3] explored assisting novices in drawing human eyes. In this work we avoid specific subjects and instead focus on *general* perspective sketching ability, however we build on the sketch recognition techniques utilized in these works.

SketchTivity [10, 15, 29–31] is an intelligent tutoring system for perspective sketching fundamentals focused on the basic primitives of sketching. This system utilizes sketch recognition to provide real-time feedback on accuracy, but does not offer support for more general drawings in perspective. Similarly, Kuribayashi et al. built a system that focused purely on evaluating handwritten cubes in perspective [17]. We aim to expand the potential of this work to include *any* rectilinear perspective drawing.

Tutorial Generation Tools

How2Sketch [11] is a system that converts a 3D model into a step-by-step sketching tutorial. This work is highly relevant as the generated sketching tutorials are in perspective, however this system provides limited real-time feedback to the user on their performance. We build on the potential of this work by contributing a general-purpose algorithm that can provide real-time feedback on any digital rectilinear perspective sketch.

Sketch Sketch Revolution [8] generates tutorials from user-generated sketches, and provided stroke-by-stroke guidance and feedback. We draw inspiration particularly from this real-time feedback, which can help a user correct mistakes as they sketch.

Creativity-Support Tools

PortraitSketch [33] is a creativity-support tool that utilizes beautification and automatic adjustment of user's strokes to assist novices in producing better portraits. We considered this beautification approach for our own work, however we ultimately decided to retain the original sketch of the user, as it can preserve stylistic choices they make and details from their imagination which are not classified as being in perspective.

ShadowDraw [18] utilizes a database of sketches and real-time classification of what a user is sketching to provide assistive underlays to the user. As the user traces, it can lead to more accurate sketches, however it is limited to a finite set of sketches. We appreciate that this tracing approach

can be helpful for novices, and it influences our approach to scaffolding which is discussed in the user interface section.

The Mental Canvas [6] is a tool for helping architects design in three dimensions by converting 2D strokes into a 3D space and helping them to analyze and refine their concepts. While the system does not provide feedback on a stroke-by-stroke basis, it is conceptually similar to our system and is another approach to providing assistance in perspective sketching.

3 PEDAGOGICAL INFLUENCE

In an effort to fully understand this problem space, we consulted two sketching experts with more than ten years of experience teaching sketching at the university level. We also referenced some of the leading literature in drawing education [7, 16, 23].

Our discussions with experts were about teaching sketching in general, but it quickly became clear that one of the biggest obstacles for novices is learning how to sketch in *perspective*, particularly two-point perspective. In one-point perspective, there is only convergence to one vanishing point, so it is much easier to grasp, however in two-point perspective, there are multiple convergences. This often confuses novices as they have to make decisions about where lines should converge to properly depict the object in perspective.

Typically, perspective is taught through demonstrations, where the instructor produces a perspective drawing in front of the students and relays concepts such as vanishing points, the horizon, convergence, and foreshortening. A very common exercise for both understanding and practicing perspective is to think of city streets and corners. One-point perspective is analogous to a city street, in which all lines parallel to the street are converging at the single vanishing point. Two-point perspective is analogous to a city corner, in which the same convergence is happening but for two streets perpendicular to each other. One of the other reasons why city streets and corners are a great way to practice perspective is because buildings are generally *rectilinear* forms, meaning they are cuboid shapes which are far easier to sketch in perspective than curvilinear shapes. Figure 2 shows a two-point perspective drawing of a city street.

These insights drive our focus on two-point perspective for both the algorithm, and for the user study, however it's important to note that it is not difficult to adjust the algorithm and intelligent interface to assess one-point perspective, or even three-point perspective.

4 ALGORITHM

There are up to four steps in the perspective recognition algorithm: pre-processing, perspective identification, convergence identification, and perspective accuracy.

Figure 2: A freehand two-point perspective sketch of a city corner generated by an expert during an educational demonstration.

Pre-processing helps remove noise from the sketch data. A stroke is represented by a series of x , y , and time values for each sampled point from pen down to pen up in a format based on SketchML [1]. Since a user can draw multiple segments in a single stroke, each stroke must be segmented into substrokes. The corners of a stroke are found via an algorithm called ShortStraw [32], and these corners are used to segment the original stroke into its substroke components. After this pre-processing step, we can extract features for every substroke in the sketch.

The extracted features are then used to determine if a substroke should be considered a perspective stroke or a non-perspective stroke. If the substroke is deemed to be perspective, then we determine which point the stroke is converging towards: left (VP1), right (VP2), or vertical. Finally, we determine if the substroke was drawn accurately converging towards the point from the previous step.

Features

We chose 26 features that we believed would be highly relevant for the algorithm. In general, these features are in one of two categories: *gestural* features and *contextual* features.

Our gestural features are extracted from the stroke itself and are influenced primarily by Rubine's seminal work in gesture recognition [24]. These features were primarily aimed at classifying whether or not a stroke was a perspective stroke or a non-perspective stroke.

The contextual features are more novel and explore relationships between the stroke and the provided perspective grid. For example, we believed the angle between the stroke's angle and the "perfect" convergence to each vanishing point as well as perfect vertical and horizontal lines would be highly relevant for both convergence classification and accuracy classification. Therefore, most of the contextual features

are different ways of measuring that angle including cosine, sin, and degrees. Table 1 reveals all of the features we explored and which subset of features were most relevant for at least one stage of the algorithm (in bold).

Feature Selection

We performed a best-first forward subset selection to determine the most important set of features for each stage of the algorithm. These features were then used in training the random forest classifier used in this section.

For determining if a stroke was a perspective or non-perspective stroke, the most important features were the length of the bounding box diagonal, angle of bounding box diagonal, distance between first and last point, sum of squared values of angles, line similarity ratio, and substroke siblings (3, 4, 5, 11, 12, 13 in Table 1). These features make sense because the sum of squared angle values and line similarity ratio can determine the roundness of a substroke which helps classify non-perspective strokes. In addition, the substroke siblings feature was intended to classify non-perspective strokes that occur from embellishments drawn with a single stroke such as a bush that would result in multiple substroke siblings.

For determining which point the stroke was converging towards if it was perspective, the most important features were the angle of the bounding box diagonal, cosine between first and last point, sine between first and last point, cosine made with vanishing points 1 and 2, and the angle made with vanishing points 1 and 2 (4, 6, 7, 15, 16, 23, 24 in Table 1). These features all deal with the direction the stroke is point as a whole as well as the relative difference between an ideal line to each of the respective points that the stroke may converge.

For determining the accuracy of the perspective line, the most important features were the angle of the bounding box diagonal, cosine between first and last point, sine between first and last point, and the cosine, sine, and angle with respect to the vanishing point determined in the previous step (4, 6, 7, 15, 16, 17, 19, 20, 23, 24 in Table 1). These features closely correlate to similarity with a perfect line drawn to the vanishing point and thus act as thresholds for accuracy within the classifier.

Performance

In order to evaluate the performance of the algorithm we compiled a set of 22 digital perspective drawings from seven individuals with varying skill levels. These individuals were given the city corner template as shown in Figure 4 but with no assistance. The drawings ranged from having very accurate perspective to having very poor perspective and many mistakes. This provided a rich and varied set of data for training.

Table 1: Features used to classify perspective strokes, their convergence, and their accuracy. The features in bold were determined significant for at least 1 classifier through subset selection.

Gestural Feature	Novel?
1. Cos of Initial Angle	No [24]
2. Sin of Initial Angle	No [24]
3. Length of Bounding Box Diagonal	No [24]
4. Angle of Bounding Box Diagonal	No [24]
5. Dist Between First and Last Point	No [24]
6. Cos of Angle Between First and Last Point	No [24]
7. Sin of Angle Between First and Last Point	No [24]
8. Total Stroke Length	No [24]
9. Total Angle Traversed	No [24]
10. Sum of Absolute Values of Angles	No [24]
11. Sum of Squared Values of Angles	No [24]
12. Line Similarity Ratio (#8 / #5)	No [20]
13. # of Substroke Siblings	Yes
Contextual Feature	Novel?
14. Above Horizon?	Yes
15. Cos to VP1	Yes
16. Cos to VP2	Yes
17. Cos to Vertical	Yes
18. Cos to Horizontal	Yes
19. Sin to VP1	Yes
20. Sin to VP2	Yes
21. Sin to Vertical	Yes
22. Sin to Horizontal	Yes
23. Degrees to VP1	Yes
24. Degrees to VP2	Yes
25. Degrees to Vertical	Yes
26. Degrees to Horizontal	Yes

Table 2: Classification Performance

Classification	Precision	Recall	F-Score
Perspective Stroke?	0.892	0.892	0.892
Convergence	0.957	0.958	0.957
Convergence Accuracy	0.884	0.885	0.883

One of the authors who is a sketching domain expert manually labeled 2026 strokes across the 22 drawings. The drawings were labeled in three different ways:

- Is it a perspective stroke or not? (Binary)
- If so, where is it converging? (VP1, VP2, Vertical)
- Is it accurate or not? (Binary)

After labeling the data we trialed a variety of classifying algorithms and found random forest [19] to perform best.

Random forest also computes very quickly, making it ideal for real-time feedback in a fluid user experience. The hyper parameters used were default. The performance can be seen in Table 2.

While this performance could improve, it's important to note that this domain is highly subjective and perfection is not realistic. In particular, the scrutiny with which a perspective stroke can be considered “accurate” does not have a strong answer. A computer can demand a perfectly converging line, however a threshold this demanding would be frustrating to a human sketching freehand. A line directly below a vanishing point could be considered converging to the vanishing point or a vertical line, depending on context.

Since the training data was labeled by a human domain expert, the binary accuracy classification of this algorithm can be considered “good enough” to the human eye. In other words, the algorithm is forgiving if a stroke is reasonably accurate enough. We believe this is very appropriate for freehand sketching, in which some minor inaccuracies are to be expected. We considered also use a continuous floating point value instead of a binary value for perspective accuracy, however that would demand a more arbitrary threshold. The manually trained binary value serves as that threshold.

Flexibility

It is important to describe the flexibility of the algorithm and the ways it can be generalized. Because of the way the features are extracted contextually, vanishing points can move anywhere in the 2D euclidean space of the sketching canvas and this will be embedded in the features. This can allow the user to adjust vanishing points in the interface itself with no detriment to algorithm performance. Vanishing points can also be added or removed, allowing for 1-point and 3-point perspective accuracy to be recognized. This flexibility allows the algorithm to be used for almost any rectilinear perspective drawing. The algorithm can also be useful when there are curvilinear forms in perspective such as cylinders, spheres, or ovoids. In these cases, the algorithm can help the individual to establish the rectilinear “scaffolding” for those forms. For example, cylinders are based in a cube or cuboid scaffolding. If the cube or cuboid is in accurate perspective, the cylinder will be in accurate perspective (see Figure 3).

5 USER INTERFACE

We designed a web-based interface which can be accessed with any device that has an internet connection. We provide the user with a basic set of drawing tools and some scaffolding in the form of a city corner sidewalk and horizon (see Figure 4). This establishes the perspective of the sketch for the user. For purposes of the study, we intentionally do not provide more power to the user such as the ability to adjust

Figure 3: An example of a sketch with a curvilinear form (cylinder). The algorithm is still helpful for these forms because they are typically constructed within a rectilinear scaffolding (see blue construction lines). The more accurate the scaffolding, the more accurate the curvilinear form will be.

Figure 4: (A) The sketching canvas, which provides two vanishing points and a sidewalk. (B) Pen and Eraser. (C) Undo and Save sketch. (D) 5-minute timer. (E) *Real-time probe only*—Text feedback on most recent stroke. (F)(G) *Summative probe only*—An evaluate button and “remove bad strokes” button for removing incorrect strokes.

the perspective grid or a wide variety of drawing options. These features can be added in the future.

We provide the horizon at two-thirds the height of the screen of the device, and the vanishing points a short distance from the edges of the screen so that they can be seen by the user. Depending on the device resolution, this can translate to different X and Y coordinates which are captured during feature extraction. While this “forces” the perspective and makes it quite dramatic, we believed this was the best approach for novices. Our discussions with sketching instructors informed us that forcing the perspective this way and keeping the vanishing points visible can be very helpful for novices.

In an effort to answer one of our research questions and understand the best way to provide feedback to the user, we designed three technology probes using this interface; an unassisted probe, a real-time feedback probe, and a summative feedback probe. The details of each are described below.

Figure 5: The perspective grid is a form of scaffolding in a light blue color commonly associated with construction lines in design sketching. It appears briefly when an incorrect perspective stroke is sketched in the *real-time feedback* probe, and is always present when the evaluation mode is toggled in the *summative feedback* probe.

Unassisted Probe

The unassisted probe was designed to assess baseline performance of participants and does not provide any feedback on the accuracy of the perspective. It is little more than a sketching canvas. However, the algorithm is still running in the background, allowing us to gain perspective accuracy data. As will be discussed in the methodology, the control group uses a slight variation of this probe that adds a static perspective grid in the background (without any intelligent feedback).

Real-time Feedback Probe

The real-time feedback probe provides feedback after each and every stroke sketched by the user. This comes in the form of turning the stroke green or red for three seconds, along with all of the previously sketched strokes. If a stroke turns red, the user would know that it is an inaccurate perspective stroke, and can either erase or undo that stroke. Additionally, text feedback either provides positive encouragement “Great job so far!” or constructive feedback “This stroke appears to converge to a vanishing point but it is inaccurate.” If the stroke is labeled incorrect, the perspective grid also appears for three seconds (see Figure 5).

It’s important to note that for users who may be red-green color blind, we use a light green color that has noticeable contrast between the dark red color used for incorrect strokes.

Summative Feedback Probe

The summative feedback probe places the power with the user and gives them the ability to evaluate the whole sketch at any time. For example, the user can sketch for a few minutes, then press the “Evaluate” button to see if any strokes were found to be inaccurate by the system. This variation also gives the user the useful ability to remove all inaccurate

strokes in the sketch in one fell swoop. The evaluation mode also provides the perspective grid (see Figure 5) and the user can leave it on for as long as they like while they grasp what could be wrong with their sketch.

It’s worth noting that while these probes could provide automatic correction to perspective strokes such as what is available in some commercial tools like Sketchbook Pro, we believed there would be more educational value in letting users make mistakes. This is supported by the educational literature [21]. When individuals are free to make their own mistakes, and then are corrected, it makes them more aware of the mistakes. This awareness can translate into learning more readily.

6 METHODOLOGY

We conducted a between-subjects user study with 40 participants in order to answer our research questions. Participants were undergraduate and graduate students from a variety of majors who were recruited primarily through convenience sampling. The group included 26 males and 14 females. The participants were predominantly novices at sketching with an average self-rated drawing ability of $2.50 (\pm 0.88)$ out of 5, and an average self-rated ability to draw specifically in perspective of $2.17 (\pm 0.87)$ out of 5.

After the initial questions on drawing ability, participants were asked to make four perspective sketches of city street corners. Each sketch was given a time limit of five minutes and participants were encouraged to be creative with the sketches, but draw them in 2-point perspective to the best of their ability. In order to ensure that any improvements in perspective accuracy were truly caused by the intelligent real-time feedback and not just practicing with a grid, half of our participants (20) were in a control condition in which they still produced four drawings, but with just a static perspective grid (see Figure 6).

Sketch 1 - Pre Test. Initially, all participants were asked to sketch with the unassisted probe, which served as a pre-test and baseline of sketching performance.

Sketches 2 + 3. Participants in the experimental group were exposed to the two feedback probes. For each of those participants, we alternated between exposing one feedback probe first versus the other to avoid learning effects from one specific probe being more pronounced.

Meanwhile participants in the control group continued to use the unassisted probe, albeit with a static perspective grid in the background added. A perspective grid is something that students can practice drawing in perspective with readily, without using any technology. However, this may not be enough for novices to truly grasp the rules of perspective and we hypothesized that practice with the grid would not

Figure 6: The study involved all participants producing four sketches. The first and last sketch were unassisted and served as pre and post tests. The second and third sketches differed for the two groups. The experimental group ($n = 20$) used the feedback probes while the control group ($n = 20$) were only assisted with a static perspective grid.

lead to any significant improvement in perspective accuracy afterwards.

Sketch 4 - Post Test. Lastly, all participants were asked to sketch with the unassisted probe one more time, which served as a post-test.

Data Analysis. For each sketch, we logged the number of strokes drawn, perspective strokes classified, accurate perspective strokes classified, and any corrections that were made (including “undo,” “erase,” and the unique “remove bad strokes” for the summative feedback probe). To mitigate imperfection in the algorithm, a sketching expert reviewed the sketches and the logs to ensure the numbers were as accurate as possible.

Between each sketch, participants answered survey questions centered around self-rating their sketch, and self-rating their confidence during the sketch. For the experimental group specifically we asked what they thought about the intelligent feedback provided by the feedback probes. Additionally, think aloud protocol [9] was used to elicit rich qualitative data from those participants about the intelligent feedback during the sketches.

7 RESULTS

During the course of the study the 40 participants produced 160 perspective sketches. The average sketch in the experimental group contained around 87 strokes (± 34.59) of which an average of 44.45% ($\pm 16.04\%$) were classified as perspective strokes. Individuals varied quite a bit with regards to how much non-perspective detail they chose to include in their sketches, with some focusing mostly on the main rectilinear buildings in perspective and others choosing to add people, signage, cars, etc. Unless otherwise noted, the following analysis is focused only on the perspective strokes generated

Table 3: Average Performance In Control Group

Condition	Perspective Acc.	Corrections
Unassisted (Pre)	56.91% ($\pm 15.87\%$)	7.55 (± 11.51)
Static Grid	72.74% ($\pm 17.51\%$)	8.00 (± 11.93)
Static Grid	63.70% ($\pm 16.69\%$)	9.50 (± 13.13)
Unassisted (Post)	56.64% ($\pm 15.30\%$)	11.59 (± 13.38)

Table 4: Average Performance In Experimental Group

Condition	Perspective Acc.	Corrections
Unassisted (Pre)	64.43% ($\pm 18.12\%$)	11.95 (± 11.79)
Real-time	99.27% ($\pm 1.80\%$)	22.45 (± 6.86)
Summative	93.36% ($\pm 11.31\%$)	19.15 (± 10.53)
Unassisted (Post)	78.94% ($\pm 11.58\%$)	17.85 (± 12.58)

by participants and how accurate they are. All p-values reported are from paired t-tests.

R1 Improvements in Perspective Accuracy

Not surprisingly, the real-time feedback provoked participants in the experimental group to make nearly twice as many corrections on average, a statistically significant increase ($p < 0.001$). This resulted in sketches that were much more accurate, also a statistically significant increase ($p < 0.0001$). Since the real-time feedback immediately showed participants any perspective error they made, it was easy for them to erase or undo that stroke and make a correct one. 17 of the 20 participants produced 100% accurate perspective sketches with this probe.

The summative feedback resulted in similar statistically significant increases in corrections and accuracy ($p < 0.05$ and $p < 0.0001$, respectively), though slightly less dramatic. Participants used this feedback in varying ways, and they did not always make all corrections suggested by the feedback. Since the feedback in this probe was suggesting multiple corrections, they would sometimes forego the “remove bad strokes” feature and only correct certain mistakes. 11 of the 20 participants produced 100% accurate perspective sketches with this probe.

What is perhaps most promising is that when the feedback was taken away and the participants returned to the unassisted probe, there was still a statistically significant increase in corrections ($p < 0.05$), and improvement in perspective accuracy ($p < 0.0005$) compared to the initial unassisted sketch. This suggests that there was a mild learning effect from exposure to the intelligent feedback for many participants, with some transfer of knowledge happening in the final unassisted sketch. 16 of the 20 participants in the experimental group improved their perspective accuracy

Figure 7: Accuracy improved with both feedback conditions, particularly real-time feedback. More importantly, when feedback was removed in the final sketch, participants retained some improvement in perspective accuracy on average ($p < 0.0005$), suggesting a learning effect and transfer of knowledge. Error bars are 1 standard deviation.

in the final sketch (see Figure 9 for two examples). This is supported by the qualitative data which included participants sharing what they thought they learned about perspective.

“I understood that basically all vertical strokes need to go straight, perpendicular to the horizon, while horizontal strokes need to be drawn at angles with perspective from your drawing to the horizon. Also you can’t see the tops of objects above your perspective to the horizon.”—P2

“It really helped in understanding how the perspective changes depending on how far away from the vanishing point I get.”—P8

Another result that is very promising is the improvement in self-rated sketching confidence in the experimental group. After each sketch participants were asked to rate their sketching confidence during the sketch on a scale from 1 to 5 (5 being very confident). The average rating after the initial unassisted sketch was $2.10 (\pm 0.79)$ out of 5, and the average rating after the final unassisted sketch was $3.45 (\pm 0.51)$ out of 5, a statistically significant increase ($p < 0.0001$). This suggests the tool could be very useful for improving sketching self-efficacy [2], an important factor when learning sketching and staying motivated to practice.

Meanwhile, participants in the control group showed signs of improvement in their perspective accuracy on average by using the static grid ($p < 0.0005$ for second sketch, statistically significant; $p = 0.072$ for third sketch, not statistically significant). However, they did not show any improvement at all in the subsequent unassisted final sketch ($p = 0.36$). This can be readily seen in Figure 8. This suggests that practice with a perspective grid alone does not help individuals

Figure 8: When comparing the pre and post sketch performance between the two groups, it can be readily seen that participants in the control group showed no improvement in perspective accuracy on average ($p = 0.36$), suggesting no learning effect. Meanwhile participants who were exposed to the real-time feedback improved by a statistically significant amount ($p < 0.0005$). Error bars are 1 standard deviation.

fundamentally improve their understanding of perspective, but the intelligent real-time feedback provided by our system can have that effect.

R2 Both Forms of Feedback Considered Useful

A content analysis of feedback given from both surveys and think aloud protocol revealed insights on how useful participants in the experimental group found the intelligent feedback provided by the probes.

In general, participants found both approaches to feedback useful, but for different reasons. When asked how useful they found the feedback on a scale from 1 to 5 (5 being very useful), participants rated the real-time feedback $4.25 (\pm 0.64)$ on average, and the summative feedback $4.20 (\pm 0.77)$ on average. The following sections reveal more nuances of how the participants felt about the different forms of feedback.

Advantages and Disadvantages of Real-time Feedback

Many participants appreciated that the real-time feedback allowed them to maintain a constant flow in their sketching, correcting individual mistakes as they progressed.

“It’s easier to stay on task with what you are drawing. The feedback rate is consistent, you’re getting feedback after every step.”—P2

“Since the feedback is real-time I am less worried about making mistakes and having to check them later. This improves my workflow significantly.”—P14

However, some participants found that the feedback could be distracting, since it appears after every single stroke. Some got concerned when they were making many mistakes in a row, and not fully grasping why the lines were inaccurate.

“The feedback is somewhat distracting while sketching.”—P12

“Its easy to get flustered and want to start over completely.”—P13

Advantages and Disadvantages of Summative Feedback

Many participants appreciated that the summative feedback gave them the power to choose when to get help.

“More ‘freedom’ to draw without getting immediately notified at if you do something wrong”—P3

“Receiving Feedback when I asked for it allowed me to work at my own pace without having to stop and retrace every line”—P5

“I got to choose when I wanted to receive feedback, meaning I could just check when I wasn’t confident in a line”—P7

However, some participants did not like the idea of making many mistakes only to later find out that much of their drawing was incorrect. In a few cases, this was an overwhelming amount of mistakes that was difficult to correct all at once.

“If you had a lot of bad lines, then the program would suggest removing a large portion of the image. Sometimes causing you to change what you are drawing completely.”—P1

“You might end up drawing the whole thing out of perspective and not check until it’s too late.”—P3

There was also a consensus among many participants that the real-time feedback was more appropriate for absolute beginners, and the summative feedback was better for when one is more intermediate to advanced.

“This type of feedback would be for those who already had some understanding of drawing in perspective and made minor mistakes. I like being able to draw larger items and then check how ‘good’ it was at one time.”—P1

“This type of feedback seems good for intermediate level.”—P9

8 DISCUSSION AND FUTURE WORK

We gained insights from this work which are worth discussing and sharing with other researchers and technologists working in the areas of creativity-support, intelligent tutoring systems, and sketch recognition.

Fading Scaffolding and Changing Feedback Based on Skill Level

Faded scaffolding [14] is a technique used in educational contexts to ease beginners in to a subject area while keeping them in their proximal zone of development [28]. It involves giving a beginner extra support until they improve their ability and confidence, at which point the support can begin to fade away. Adding, then removing training wheels on a bicycle is a classic example of faded scaffolding.

We believe the insights gained from the user study reveal an ideal way to utilize faded scaffolding in the context of helping novices learn perspective sketching with intelligent software. Initially, a persistent perspective grid can help novices to essentially trace perspective gridlines, and make accurate perspective sketches immediately. This would be used in combination with real-time feedback for every stroke to catch any mistakes from the individual immediately. Then, the persistent perspective grid can be removed, and only shown when mistakes are made. When the individual has reached a more intermediate to advanced level, the real-time feedback can be replaced with summative feedback, bringing them the power to receive help only when desired, and giving them more freedom to sketch without scaffolding. Lastly, when the individual has shown mastery of perspective, all forms of feedback can be removed and the individual can have full freedom to sketch anything they desire on any medium (digital or traditional).

Algorithm Limitations

It’s important to emphasize that this algorithm is limited to rectilinear perspective accuracy recognition. Recognizing perspective accuracy of *curvilinear* forms is significantly more complex, and can be subjective even for human sketching experts. Such recognition would likely have to involve more robust and nuanced perspective stroke classification, since the current algorithm does not classify curves as being perspective strokes. It would also need many more contextual features which describe the relationship between strokes in the sketch and what they are trying to depict. The systems built by Xie et al. in automatically completing sketches and paintings [34, 35] use contextual features and could be a source of inspired improvement.

Edge cases can also detriment the potential educational benefit of such an algorithm. For example, a common novice mistake was to draw sidewalk lines as a series of vertical strokes, even though they should converge to the vanishing points (see Figure 10). More contextual features could help with these edge cases and it may be possible to subdivide provided templates into sections where specific perspective convergences are expected.

Examples of Perspective Accuracy Improvement

Figure 9: These participants improved noticeably from the initial unassisted sketch to the final unassisted sketch. Note that in the initial sketches both participants make mistakes with the roofs of the buildings (you should not be able to see the roof of a building above the horizon). In their final sketches they have properly depicted the buildings and their roofs.

Figure 10: (A) shows a common novice mistake which is to miss convergence in sidewalks. The algorithm assumes these are "correct" vertical strokes because it has no context of knowing that it is meant to depict a sidewalk. (B) shows the correct convergence.

Supplementing Existing Pedagogy

For reasons mentioned in the previous section, we believe this user interface would be best used in combination with traditional pedagogy. In a traditional studio environment where sketching is typically taught, a qualified instructor may perform demonstrations, show videos, and thoroughly explain the concepts behind perspective. An interface such as what we've presented in this work can allow the students to practice, ensuring they at least grasp the concept of rectilinear two-point perspective before moving on to other advanced topics. The real-time feedback provided by the tool could allow students to self-regulate their learning [22], helping them avoid common novice mistakes without instructor input.

Additionally, the instructor could offload the task of grading to such a tool, significantly reducing the time needed for grading. They can then observe the performance of individual students, and better personalize their instruction for the students that need it most.

9 CONCLUSION

We've presented a flexible algorithm for rectilinear perspective accuracy recognition, its performance, and the most important subset of features it utilizes. We've also presented an intelligent user interface that can provide different forms of feedback to users and assist their freehand perspective sketching in real time. The results of a user study with 40 participants showed that the feedback allowed participants to improve the accuracy of their sketches significantly, and we observed that there was at least some transfer of knowledge when the feedback was taken away, suggesting the tool can be helpful for novices learning perspective. We gained an understanding of the advantages and disadvantages of real-time versus summative feedback in this context and synthesized principles for fading scaffolding and changing feedback based on an individual's skill level. We believe this is a promising educational tool, particularly if it used as a supplement to existing pedagogy.

ACKNOWLEDGMENTS

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 nor Microsoft.

REFERENCES

- [1] Danilo Avola, Andrea Del Buono, Giorgio Gianforme, Stefano Paolozzi, and Rui Wang. 2009. SketchML: a representation language for novel sketch recognition approach. In *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*. ACM, Corfu, Greece, 31.
- [2] Albert Bandura. 1997. *Self-efficacy: The exercise of control* (1st ed.). Worth Publishers, New York, New York, USA.
- [3] 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.
- [4] Karen Ernst daSilva. 2001. Drawing on Experience: Connecting Art and Language. *Primary Voices K-6* 10, 2 (2001), 2–9.
- [5] 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, USA, 897–906.
- [6] Julie Dorsey, Songhua Xu, Gabe Smedresman, Holly Rushmeier, and Leonard McMillan. 2007. The mental canvas: A tool for conceptual architectural design and analysis. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. IEEE, Maui, Hawaii, 201–210.
- [7] Betty Edwards. 1999. *The New Drawing on the Right Side of the Brain* (2nd ed.). Tarcher, New York, New York, USA.
- [8] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-Sketch Revolution: an engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, Santa Barbara, CA, 373–382.
- [9] Marsha E Fonteyn, Benjamin Kuipers, and Susan J Grobe. 1993. A description of think aloud method and protocol analysis. *Qualitative health research* 3, 4 (1993), 430–441.
- [10] 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 Transactions on Interactive Intelligent Systems (TiiS)* 8, 3 (2018), 19.
- [11] James W. Hennessey, Han Liu, Holger Winnemöller, Mira Dontcheva, and Niloy J. Mitra. 2017. How2Sketch: Generating Easy-to-follow Tutorials for Sketching 3D Objects. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '17)*. ACM, New York, NY, USA, Article 8, 11 pages. <https://doi.org/10.1145/3023368.3023371>
- [12] 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 *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers (ASME), Charlotte, North Carolina, V003T04A013–V003T04A013.
- [13] Ethan Hilton, Myela Paige, Blake Williford, Wayne Li, Tracy Hammond, and Julie Linsey. 2017. Improving the sketching ability of engineering design students. In *International Conference on Engineering Design (ICED)*, Vol. 9: Design Education. The Design Society, Vancouver, BC, Canada, 217–224. ISBN: 978-1-904670-97-1.
- [14] 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., Los Angeles, CA, 187–194.
- [15] 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 *SBIM '17 Proceedings of the Symposium on Sketch-Based Interfaces and Modeling*, Vol. 2. ACM, Los Angeles, CA, USA, 1–11. ISBN: 978-1-4503-5079-2, DOI: 10.1145/3092907.3092911.
- [16] Joseph A. Koncelik and Kevin Reeder. 2008. *Conceptual Drawing* (1st ed.). Course Technology, Boston, Massachusetts, USA.
- [17] Yuki Kuribayashi, Eito Yamaga, and Takeshi Sasaki. 2019. Handwritten Cube Recognition and Evaluation Based on Perspectives for Sketch Training System. In *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, Paris, France, 34–39.
- [18] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. ShadowDraw: Real-time User Guidance for Freehand Drawing. In *ACM Transactions on Graphics*, Vol. 30:4. ACM, New York, NY, USA, Article 27, 10 pages. <https://doi.org/10.1145/2010324.1964922>
- [19] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.
- [20] A Chris Long Jr, James A Landay, Lawrence A Rowe, and Joseph Michiels. 2000. Visual similarity of pen gestures. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, The Hague, The Netherlands, 360–367.
- [21] Henry J Perkinson. 1979. Learning from our mistakes. *Et cetera* 36 (1979), 37.
- [22] Paul R. Pintrich, Christopher A. Wolters, and Gail P. Baxter. 2000. Assessing Metacognition and Self-Regulated Learning. In *Metacognitive Assessment*, G. Schraw (Ed.). University of Nebraska Press, Lincoln, Nebraska, USA.
- [23] Scott Robertson and Thomas Bertling. 2013. *How to Draw: Drawing and Sketching Objects and Environments from your Imagination* (1st ed.). Design Studio Press, Culver City, California, USA.
- [24] Dean Rubine. 1991. Specifying Gestures by Example. *SIGGRAPH Comput. Graph.* 25, 4 (July 1991), 329–337. <https://doi.org/10.1145/127719.122753>
- [25] Ben Shneiderman. 2002. Creativity support tools. *Commun. ACM* 45, 10 (2002), 116–120.
- [26] Kathy G. Short, Gloria Kauffman, and Leslie H. Kahn. 2000. "I Just Need To Draw:" Responding to Literature across Multiple Sign Systems. *The Reading Teacher* 54, 2 (Oct 2000), 160–171.
- [27] Derek Sleeman and John Seely Brown. 1982. *Intelligent tutoring systems*. Academic Press, London.
- [28] Lev Vygotsky. 1987. Zone of proximal development. *Mind in society: The development of higher psychological processes* 5291 (1987), 157 pages.
- [29] Blake Williford, Matthew Runyon, Josh Cherian, Wayne Li, Julie Linsey, and Tracy Hammond. 2019. A Framework for Motivating Sketching Practice with Sketch-based Gameplay. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. ACM, ACM, USA, 533–544.

- [30] Blake Williford, Matthew Runyon, Adil Hamid Malla, Wayne Li, Julie Linsey, and Tracy Hammond. 2017. ZenSketch: A Sketch-based Game For Improving Line Work. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*. ACM, ACM, USA, 591–598.
- [31] 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 (Human-Computer Interaction Series)*, A. Adler T. Hammond, S. Valentine (Ed.). Springer, Switzerland, 115–127. ISBN: 978-3-319-31193-7, http://dx.doi.org/10.1007/978-3-319-31193-7_8.
- [32] Aaron Wolin, Brian Eoff, and Tracy Hammond. 2008. ShortStraw: A Simple and Effective Corner Finder for Polyline. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling (SBIM)*. Eurographics Association, Annecy, France, 33–40. ISBN: 978-3-905674-07-1.
- [33] 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, USA, 407–417.
- [34] Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. 2014. Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 172.
- [35] Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete hand-drawn animations. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 169.
- [36] Ruhizan M. Yasin, Lilia Halim, and Azaman Ishar. 2012. Effects of Problem-Solving Strategies in the Teaching and Learning of Engineering Drawing Subject. *Asian Social Science* 8, 16 (2012), 65–79.