

# InvisiShapes: A Recognition System for Sketched 3D Primitives in Continuous Interaction Spaces

Paul Taelle<sup>(✉)</sup> and Tracy Hammond

Sketch Recognition Lab, Texas A&M University, College Station, TX 77840, USA  
{ptaelle,hammond}@cse.tamu.edu  
<http://srl.tamu.edu>

**Abstract.** Continued improvements and rising ubiquity in touchscreen and motion-sensing technologies enable users to leverage mid-air input modalities for intelligent surface sketching into the third dimension. However, existing approaches largely either focus on constrained 3D gesture sets, require specialized hardware setups, or do not deviate beyond surface sketching assumptions. We present InvisiShapes, a recognition system for users to sketch 3D geometric primitives in continuous interaction spaces that explore surfaces and mid-air environments. Our system leverages a collection of sketch and gesture recognition techniques and heuristics and takes advantage of easily accessible computing hardware for users to incorporate depth to their sketches. From our interaction study and user evaluations, we observed that our system successfully accomplishes strong recognition and intuitive interaction capabilities on collected sketch+motion data and interactive sketching scenarios, respectively.

**Keywords:** Sketch recognition · Gesture recognition · Continuous interaction spaces · 3d drawing · Mid-air interaction

## 1 Introduction

Designing intelligent user interfaces for automatically recognizing digital sketches have conventionally focused on sketches made on physical surfaces, and we can see this ranging from the traditional interactions of a stylus contacting pen-enabled monitors, to the ubiquitous interactions of a finger contacting a smartphone screen, and even to the emerging interactions of several fingers contacting the surface of large tabletop displays. However, the relatively recent trend of commercially-available motion-sensing hardware devices continues to shift the landscape of how people interact with computing devices. These trends can be seen from previous mainstream commercial hardware devices such as the wand-based controls of the Nintendo's Wii or the far-field motion-sensing controls of Microsoft's Kinect, to more recent releases of near-field motion-sensing devices such as Leap Motion's namesake sensor or Creative's Senz3D. Therefore, users are no longer restricted to interactions such as sketching made on physical computing surfaces, but can further broaden these interactions either continuously or disjointly into mid-air using motion-sensing hardware devices.

As these motion-sensing hardware devices are experiencing growing reliability, shrinking form factors, and wider ubiquity, researchers and developers can tap into these resources and explore the design of intelligent spatial sketch user interfaces to recognize sketches other than solely from the established settings of familiar touchscreens. These interfaces can take advantage of expanding sketching interactions beyond current surface interaction spaces for a variety of applications, and into novel interaction spaces that continuously extend from surfaces (i.e., continuous interaction spaces [9]). Sketching scenarios in continuous interaction spaces can therefore motivate both realized and potential applications, whether it is rapidly prototyping or representing entities in design and planning, quickly creating artifacts for gaming or immersive environments, more intuitively drawing three-dimensional concepts, or so on.

However, adapting intelligent user interfaces for spatial sketching interactions first involves seriously considering the challenges inherent with motion-sensing technologies distinct from conventional pen- and touch-enabled computing devices. These challenges include but are not limited to: determining appropriate non-surface sketching analogs to surface sketching, addressing imprecise motion-sensing sensors compared to touch display sensors, discovering optimal domain contexts that naturally benefit from the use of a third spatial dimension, and accommodating non-surface sketching factors that are unique to working in continuous interaction spaces [15]. Furthermore, prior research have little explored intelligent user interfaces specific to spatial sketches. Previous works for surface interaction spaces are constrained by solely surface sketching assumptions [14], existing works for mid-air interaction spaces focus on command gestures that have limited gesture vocabularies [12], and current works for continuous interaction spaces focus instead on other forms of interactions such as selection (e.g., [11]) and modeling (e.g., [3]).

In this paper, we therefore describe our approach that adapts existing recognition approaches from the sketch and gesture recognition research field, and leverages existing technologies with commercial motion-sensing hardware for recognizing freehand sketched 3D geometric primitives in continuous interaction spaces, with the hope that such an approach allows people to intuitively extend the expressiveness of their surface sketches and without resorting to mode switching (e.g., selectable menu options). We developed this approach by taking advantage of corner-finding and primitive geometric shape recognition techniques from the sketch recognition field, and then adapting them to continuous interaction spaces that utilizes a conventional touch-enabled surface display (e.g., notebook computer screen) and a lightweight motion-sensing hardware device (e.g., Leap Motion sensor). From evaluating our approach for a set of 3D geometric primitives, we discovered that users were able to intuitively draw automatically-recognized primitives with reasonable accuracy.

## 2 Related Work

### 2.1 Surface Interaction Spaces

Various directions for automatically recognizing sketches from the surface sketch recognition community have focused on addressing the challenges of recognizing raw digital surface ink strokes at different levels. At the low-level stage, techniques such as IStraw [18] proposed processing the raw strokes and segmenting out candidate corners within these strokes. The corner information is then used for the next recognition stage, where techniques such as PaleoSketch [10] and QuickDraw [2] rely on the segmented surface stroke information from corner-finding techniques in order to classify the original raw strokes into various simple and complex geometric shapes. Furthermore, top-level sketch recognition systems such as LADDER [4] can then take advantage of this combined information such as from primitive geometric shape classifiers to recognize fuller sketches. While these techniques perform very well for sketches made on surfaces interaction spaces, they are also constrained to surface sketching assumptions and do not account for the diverse challenges of recognizing noisier sketches with different sketching behaviors beyond surface interaction spaces [14].

### 2.2 Continuous Interaction Spaces

Research work for designing interfaces that explored continuous interaction spaces – or interaction spaces that occur both on and above surfaces [9] – have taken advantage of different types of computing input devices at both the on-surface and above-surface level. Work by [13] introduced the idea early on for mid-air selection and movement operations using a tabletop display augmented by a digital pen recording mid-air spatial positions. As tabletop display systems became more sophisticated, researchers began exploring continuous interaction spaces that adopted more diverse forms of interaction. For example, work by [9] described broadening the possible input modalities to include multi-touch and tangible objects, work by [11] provided more refined guidelines for previous explored interaction tasks, and work by [3] expanded interaction tasks using a system called Mockup Builder to include sketch-based modeling of three-dimensional objects. While Mockup Builder enables users in continuous interaction spaces to model three-dimensional objects with a combination of gestures and motions in a spatial sketch user interface, our work differs by focusing on a recognition approach for automatically recognizing users’ sketched 3D geometric primitives within an intelligent spatial sketch user interface.

### 2.3 Mid-Air Interaction Spaces

With continuing improvements made to commercial motion-sensing technologies and growing shifts of natural user interfaces, researchers have strongly capitalized on existing surface gesture recognition techniques, and then adapting

them to automatically understand motion gestures performed in mid-air interaction spaces. For example, \$3 [6, 7] provide mid-air analogs to Dollar recognizers (e.g., [1]), while [5, 8] further improve upon the lessons from prior mid-air motion gesture recognition techniques. However, due to the limited vocabularies of mid-air motion gesture recognition techniques [12], and since the gesture sets are generally minor 3D variants of flat 2D gestures, they are not optimal approaches for recognizing the greater complexity of 3D geometric primitives.

### 3 Interaction Methodology

In order to develop a system that is able to classify users' interactions of sketched 3D primitives, it is important to better understand how users would produce them in continuous interaction spaces with accessible commercial hardware devices such as touch-enabled screens and inexpensive motion-sensors. Therefore, we first conducted a short-term interaction study that involved observing users informally demonstrating such interactions offline, and then taking insights from their interactions to produce a representative list of both 3D geometric primitives to classify and also corresponding interactive cues to draw them in a continuous interaction space.

**Table 1.** A representative list of the 3D geometric primitives that InvisiShapes can classify that were derived from the interaction study. The shape name is the real-world label presented to users, the formal term is its geometric label, and the surface, transition, and mid-air classes refer to the components that the geometric primitive is composed of.

Shape name	Formal term	Surface	Transition	Mid-air
Path	Horizontal line	Path	Click	Empty
Pole	Vertical line	Dot	Tip	Empty
Wall	Vertical plane	Path	Flat	Equal
Cone	Cone	Ellipse	Tip	Empty
HalfCone	Ellipse frustum	Ellipse	Flat	Greater
Cylinder	Cylinder	Ellipse	Flat	Equal
Bowl	Reverse ellipse frustum	Ellipse	Flat	Lesser
Pyramid	Pyramid	Quadrilateral	Tip	Empty
HalfPyramid	Rectangle frustum	Quadrilateral	Flat	Greater
Box	Cuboid	Quadrilateral	Flat	Equal
Pan	Reverse rectangle frustum	Quadrilateral	Flat	Lesser

### 3.1 Interaction Study

We initially recruited a group of nine participants – two females – from ages 18 to 33 years, all of whom self-reported strong experience using touch-enabled devices, and ranged from average to strong familiarity with motion-sensing devices. The study participants were told that they would be individually taking part in an interaction study that involved demonstrating how to create various 3D geometric shapes on a surface space. We further expanded our explanation by having the participants assume that their demonstrated interactions of these shapes would later be understood by a touchscreen and external camera.

After introducing the scenario to the participants, we then sat them at a table and provided the participants with both a pen to draw on paper and a touch-enabled notebook computer to draw on a basic drawing application, in order to act out their roles in the described scenario. Once the participants communicated to us that they understood their scenario, we verbally prompted the users to demonstrate a list of different geometric primitive shapes conventionally found in geometry math textbooks and computer graphic applications.

Summarizing our general observations of the user participants, we discovered that the participants came to a consensus on how they demonstrated drawing the 3D geometric primitives. The participants first drew the intended shape base of the verbally-prompted shape on the flat surface (e.g., on paper or touchscreen display), then extended away from the completed sketched base, and lastly expressed the depth of the shape in mid-air before connecting their pen and finger on paper and screen, respectively, on the edge of the shape’s sketched base. We observed that the participants’ demonstrated motions align with similar interactions from prior systems based in continuous interaction spaces (e.g., [3]) and is supported by generalized drawing behaviors of shapes in other domains from the cognitive psychology (e.g., [16]).

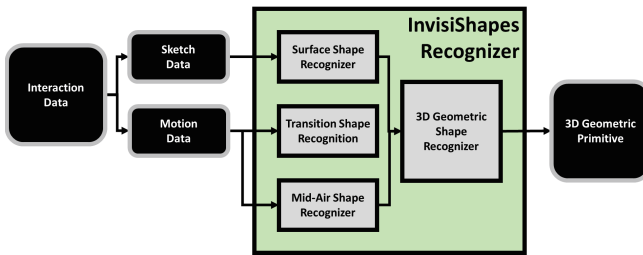
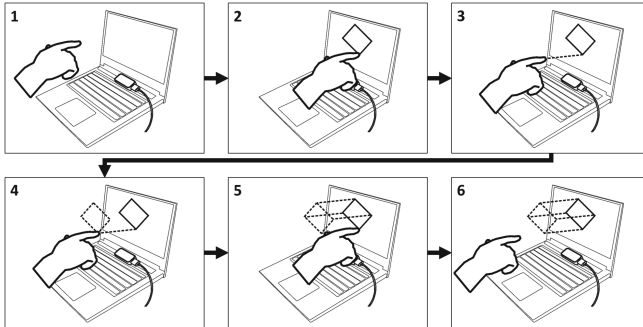


Fig. 1. A generalized system overview of the InvisiShapes recognition system.

### 3.2 Interaction Process

From the insights of our interaction study, as well as the lessons from the interaction cues of related interactive systems and the observational findings of related

cognitive psychology works, we first derived a representative list of eleven geometric shape primitives (Table 1) that also happen to be analogs in 3D to most of the eight geometric shape primitives found in surface geometric shape primitive recognizer PaleoSketch [10].



**Fig. 2.** A visual representation of the user’s interactive steps with their finger to draw a tall cuboid using a touch-enabled notebook computer screen and an accompanying Leap Motion sensor device: the user is (1) about to draw the primitive, (2) drawing the primitive’s surface base, (3) extends out from the surface base, (4) draws the corresponding mid-air base, (5) motions towards to tap screen to complete the sketch, and (6) hovers away from the completed drawing.

We additionally derived a series of interactive steps for users to draw these primitives in continuous interaction spaces (Fig. 2). Our particular interaction setup combines stylus or touch strokes that are visualized on a display monitor, and then viewed as a visualized cursor during mid-air interactions. We also normalized the motion-sensed coordinates to the display screen coordinates by offsetting them relative to the most recent position recorded from the surface sketch point, so that the user can see a cursor indicating where their mid-air stylus or finger relative to the display.

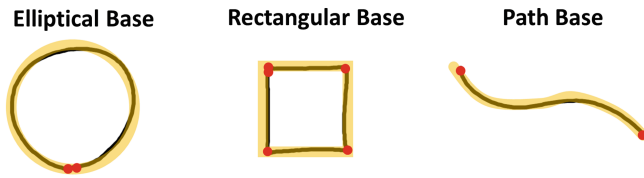
In regards to the primitives list, we briefly elaborate on three notable omissions: freeform shapes, non-quadrilateral polygons, and sphere-based primitives. For the first group of freeform shapes, our recognition system can trivially label these shapes as non-geometric primitives. For the second group of non-quadrilateral polygons, our system can trivially recognize them in the same process that is used to recognize quadrilaterals, but we chose to not list them in the paper for brevity. For the third group of sphere-based primitives, we discovered from our interaction study that participants had difficulty in coming to a consensus on how to demonstrate sketching them, so we omit these primitives from our initial list and discuss potential directions near the end of the paper.

## 4 Recognition Methodology

The InvisiShapes recognition system builds upon a variety of sketch and gesture recognition techniques and heuristics to classify the diverse types of components that construct the 3D geometric primitives (Fig. 1). The system takes in the user’s interaction data that is composed of both the sketch data performed on a touchscreen made by either stylus or finger, and also the motion data performed in the air on a motion-sensing device that extends from the sketch. The sketch and motion data are subsequently sent to their respective recognizers before combining their results to a final recognizer that outputs the most likely 3D geometric primitive.

### 4.1 Surface Sketch Recognition

The surface sketch recognizer was designed for data produced from touch or stylus input, and relies on corner segmentation information from IStraw [18] and individual surface sketch shape tests and closed shapedness derived from various sketch recognition techniques such as PaleoSketch [10] and ShortStraw [17] to classify the surface sketch (Fig. 3).



**Fig. 3.** Examples of surface bases that users sketched of different geometric primitives that were segmented with IStraw and classified with various surface sketch recognition techniques (L-R): the elliptical base of a sketched square with two detected corners, the rectangular base of a sketched square with five detected corners, and the path base of a sketched curve with two detected corners.

Path sketches consist of either polylines or curvilinear lines that form either path or wall shapes, and these sketches rely solely on identified endpoints not demonstrating closed shapedness.

Dot sketches consist of pole shapes and require merely a tap on the screen. These sketches are defined by their bounding box not exceeding a small area threshold (i.e., 100 pixels squared).

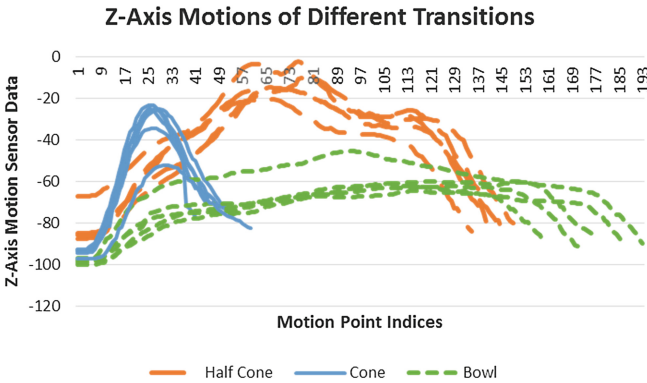
Ellipse sketches follow the ellipse test from PaleoSketch in that they must contain at most three corners and closed shapedness. Then, the two furthest points are first located from the stroke, and then rotated by the opposite of the angle that is formed from the line between these endpoints. Afterwards, the smaller value between the ellipse stroke’s path length and the Ramanujan approximation of the ideal circumference length is divided from the larger value to form a ratio that must exceed a certain threshold (i.e., 0.9).

Polygon sketches similar follow polygon tests from PaleoSketch in that they must contain  $n + 1$  corners, where  $n$  is the number of vertices in the polygon, and closed shapeness. Then, a line test is performed between each segmented corner, where the ratio of the line’s path length and ideal line length is calculated and must exceed a certain threshold (i.e., 0.9).

### 4.2 Transition Motion Recognition

Identifying the shape of both bases of the user’s interaction data is crucial to determine the type of 3D geometric primitive it forms. However, due to the noisy nature of current commercially available motion-sensing devices, it is challenging to separate what part of the motion data is the base itself and what part is the transition to the base. As a result, the motion data is classified from two different recognizers. The first is the transition motion recognizer, which identifies whether the base potentially exists initially, and then determines whether it is tipped (e.g., pyramids or cones) or flat (e.g., cylinders, cuboids, frustums).

A useful feature of the motion data to help identify the type of transition of the motion data is from how the z-axis motion is graphed with respective to the number of points (Fig. 4). We empirically observed that the smaller the mid-air base, the more steep the curve is formed from the z-axis motion. As a result, we first calculate the angle of the left and right lines formed from the endpoints to the peak of the graph, and then average the two angles.



**Fig. 4.** A plot of z-axis motions from the Leap Motion sensor device for a subset of motioned 3D geometric primitives, where shapes with smaller mid-air bases contain fewer collected points and steeper overall slopes.

For tip motions, we classify them if their averaged angles exceed a certain threshold (i.e.,  $60^\circ$ ). Contrary to tip motions, flat motions are classified as such if their averaged angles do not exceed the tip motion’s angle threshold requirement.

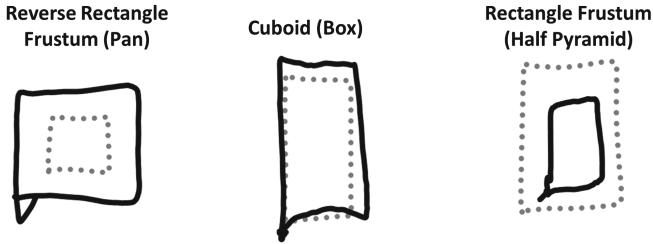
A special case of the bases involve clicks, which are rapid clicks that form for shapes that are not 3D such as paths in order to denote the lack of depth. If



the area of the motion does not exceed a certain area threshold (i.e., 1000 stroke pixels squared from a Leap Motion sensor), then it is classified as a click motion.

### 4.3 Mid-Air Motion Recognition

In conjunction with the information received from the transition motion recognizer, we can finally classify the mid-air base of the demonstrated 3D geometric primitive. We first trim the tails that form from the motion data due to the unintentional surface sketch noise produced as the user transitions from the surface sketch base to the intended mid-air base. We empirically set the trimming of the tails to the first and last 10% of the motion stroke, since we empirically observed that this adequately approximates the users' intended motioned mid-air base. Due to the noisiness of the motion data, we also define the mid-air base by first resampling both the sketch and motion strokes so that they have the same number of points, and then compare the ratio of points within each stroke's bounding box (Fig. 5).



**Fig. 5.** Examples of sketched and motioned data strokes of shapes with varying mid-air bases, where the dotted gray lines represents the surface stroke and the solid black lines represent their corresponding mid-air stroke. The two frustums visually demonstrate surface stroke points that lie either completely or dominantly inside or outside the bounding box of their corresponding mid-air stroke. For geometric primitives with congruent bases, points from their surface and mid-air strokes more frequently overlap each others' bounding boxes.

With geometric primitives that consist of congruent sketched and motioned bases, we take the bounding boxes of both bases and compare the ratio of number of points of the sketched base to the motioned base and vice versa. If the greater and lesser ratios do not exceed certain thresholds (i.e., 0.9 and 0.1, respectively), we then classify the mid-air motion as equal to the surface base.

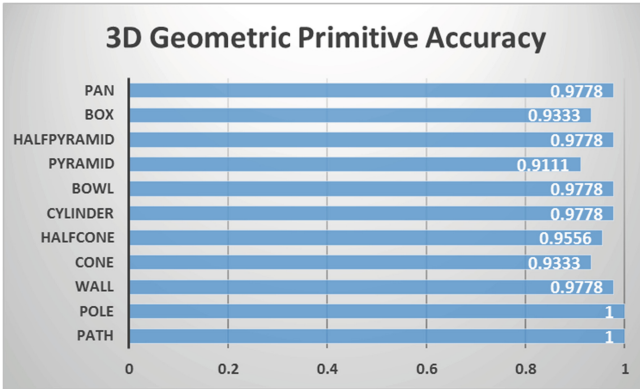
For the larger mid-air base, we classify the motion as greater if the ratio of sketched points within the motioned point's bounding box exceeds 0.9. On the other hand, we perform the opposite ratio test for lesser motions representing a larger surface base, where the motioned points contained within the sketched points' bounding box must exceed 0.9. However, for shapes that do not have bases, such as those that are tipped or clicked, we rely on the transition shape information to automatically classify their motions as empty.

#### 4.4 3D Geometric Shape Recognition

Once the sketch and motion data are processed through their respective classifiers, the three labels of surface, transition, and mid-air that are generated from the classifiers are then sent to the final 3D geometric shape recognizer. We rely on the labels produced from Table 1 to then determine the interaction data’s associated primitive type. If the interaction data’s three labels do not fit appropriately to the list of surface, transition, and mid-air labels, we instead classify the primitive as a freeform shape.

### 5 Evaluation and Discussion

To evaluate our approach, we utilized a touch-enabled Wacom tablet and a Leap Motion sensor to record users’ surface and mid-air sketching, respectively. Prior to performing our data collection, we performed a one-time calibration of the Leap Motion’s motions to that of the screen dimensions of the tablet screen, and placed the Leap Motion in front of the screen lying within several inches directly from the tablet screen’s center. We also ran these commercially-available hardware devices within their optimal interaction settings of a table setting in a normally-lit room.



**Fig. 6.** Recognition accuracy of the different 3D geometric primitives using all-or-nothing classification.

For the data collection study, we recruited nine individuals – two females – between the ages of 25–35 years, all of whom self-reported some experience with motion-tracking controls from commercial video game systems but not from research-driven motion-tracking applications. Each user was provided instructions on the type of interactions that they were performing and the shapes that they were about to draw, but were not given specific instructions on how to draw the shapes. After allowing the users to spend at most five minutes to freely draw

in this continuous interaction space setup, they were then prompted to sketch five consecutive iterations of each shape listed in Fig. 1 for a total of 495 shapes. We then classified their sketched shapes with our approach using all-or-nothing accuracy Fig. 6, where shapes were considered as correctly classified if the user’s actual input completely matched the expected label without exception.

From our approach, we demonstrate that users were able to successfully perform three-dimensional drawing of representative geometric primitives with reasonable accuracies, where accuracies for each shape did not fall below 90%. The most common types of mis-classifications came from users drawing bases that were either congruent for expected frustum shapes or not congruent for expected prisms or cylinders, or drawing mid-air bases that were not accurately detected from the motion sensor.

## 6 Conclusion and Future Work

In this paper, we describe our work on InvisiShapes, a 3D geometric primitive shape recognizer for continuous interaction spaces. From our evaluation, we demonstrate that not only were users able to intuitively sketch geometric primitives using commercially-available touchscreen and motion-sensing hardware, but also with their interaction data classified to their 3D geometric primitives from our recognition system with reasonable accuracy. From our current work’s progress, we propose several potential future directions such as expanding our recognition system to incorporate more challenging 3D geometric primitives, developing appropriate spatial sketch user interfaces that take advantage of our recognition system for various domains, observing how users draw in more varied interaction scenarios, and expanding the recognizer to other motion-sensing hardware.

## References

1. Anthony, L., Wobbrock, J.O.: \$N\$-protractor: a fast and accurate multistroke recognizer. In: Proceedings of Graphics Interface 2012. Canadian Information Processing Society, Toronto (2012)
2. Cheema, S., Gulwani, S., LaViola, J.: QuickDraw: improving drawing experience for geometric diagrams. In: 2012 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York (2012)
3. De Araùjo, B.R., Casiez, G., Jorge, J.A.: Mockup builder: direct 3d modeling on and above the surface in a continuous interaction space. In: Proceedings of Graphics Interface 2012, pp. 173–180. Canadian Information Processing Society, Toronto (2012)
4. Hammond, T., Davis, R.: LADDER, a sketching language for user interface developers. *Comput. Graph.* **29**, 518–532 (2005)
5. Hoffman, M., Varcholik, P., LaViola Jr., J.J.: Breaking the status quo: improving 3D gesture recognition with spatially convenient input devices. In: Proceedings of the 2010 IEEE Virtual Reality Conference. IEEE Computer Society, Washington (2010)

6. Kratz, S., Rohs, M.: A \$3 gesture recognizer: simple gesture recognition for devices equipped with 3D acceleration sensors In: Proceedings of the 15th International Conference on Intelligent User Interfaces. ACM, New York (2010)
7. Kratz, S., Rohs, M.: Protractor3D: a closed-form solution to rotation-invariant 3D gestures In: Proceedings of the 16th International Conference on Intelligent User Interfaces. ACM, New York (2011)
8. Kratz, S., Rohs, M., Essl, G.: Combining acceleration and gyroscope data for motion gesture recognition using classifiers with dimensionality constraints. In: Proceedings of the 18th International Conference on Intelligent User Interfaces. ACM, New York (2013)
9. Marquardt, N., Jota, R., Greenberg, S., Jorge, J.A.: The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011. LNCS, vol. 6948, pp. 461–476. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-23765-2\\_32](https://doi.org/10.1007/978-3-642-23765-2_32)
10. Paulson, B., Hammond, T.: PaleoSketch: accurate primitive sketch recognition and beautification. In: Proceedings of the 13th International Conference on Intelligent User Interfaces, pp. 1–10. ACM, New York (2008)
11. Spindler, M., Martsch, M., Dachselt, R.: Going beyond the surface: studying multi-layer interaction above the tabletop. In: 2012 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1277–1286. ACM, New York (2012)
12. Steins, C., Gustafson, S., Holz, C., Baudisch, P.: Imaginary devices: gesture-based interaction mimicking traditional input devices. In: Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 123–126. ACM, New York (2013)
13. Subramanian, S., Aliakseyeu, D., Lucero, A.: Multi-layer interaction for digital tables. In: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology. ACM, New York (2006)
14. Taele, P., Hammond, T.: Initial approaches for extending sketch recognition to beyond-surface environments. In: Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts, pp. 2039–2044. ACM, New York (2012)
15. Taele, P., Hammond, T.: Developing sketch recognition and interaction techniques for intelligent surfaceless sketching user interfaces. In: Proceedings of the Companion Publication of the 19th International Conference on Intelligent User Interfaces, pp. 53–56. ACM, New York (2014)
16. van Sommers, P.: Drawing and Cognition: Descriptive and Experimental Studies of Graphic Production Processes. Cambridge University Press, Cambridge (1984)
17. Wolin, A., Eoff, B., Hammond, T.: ShortStraw: a simple and effective corner finder for polylines. In: Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling, pp. 33–40. ACM, New York (2008)
18. Xiong, Y., LaViola Jr., J.J.: A ShortStraw-based algorithm for corner finding in sketch-based interfaces. *Comput. Graph.* **34**(5), 513–527 (2010)