SCIENCE CHINA
Information Sciences

• MOOP •

May 2017, Vol. 60 053101:1–053101:6
doi: 10.1007/s11432-017-9063-2

Tilting-Twisting-Rolling: a pen-based technique for compass geometric construction

Fei LYU[1,2,3], Feng TIAN[1,3*], Guozhong DAI[1,3] & Hongan WANG[1,3]

[1]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*School of Digital Media and Design Arts, Beijing University of Posts and Telecommunications, Beijing* 100876*, China;*
[3]*Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China*

Received February 10, 2017; accepted April 1, 2017; published online April 10, 2017

Abstract This paper presents a new pen-based technique, Tilting-Twisting-Rolling, to support compass geometric construction. By leveraging the 3D orientation information and 3D rotation information of a pen, this technique allows smooth pen action to complete multi-step geometric construction without switching task states. Results from a user study show this Tilting-Twisting-Rolling technique can improve user performance and user experience in compass geometric construction.

Keywords Tilting-Twisting-Rolling, pen-based tool, virtual compass, geometry construction, interaction

Citation Lyu F, Tian F, Dai G Z, et al. Tilting-Twisting-Rolling: a pen-based technique for compass geometric construction. Sci China Inf Sci, 2017, 60(5): 053101, doi: 10.1007/s11432-017-9063-2

1 Introduction

Pen-based devices offer a direct way to construct geometry. Besides rough sketches, users also need to build precise geometry. One common approach to create precise geometry is to beautify users free stroke based on geometric constraints [1, 2]. However, in areas like education, its desirable to present precise geometry during the drawing process. For example, in geometry classes where geometry construction with ruler and compass is used to understand fundamental concepts1), the sketch-first-and-then-beautification approach may not represent spatial attributes of objects, such as size, location, and distance dynamically.

Currently, there are some applications support geometric construction by the ruler and compass. The Geometer's Sketchpad can help teachers and students build mathematical models, objects, figures, diagrams, and graphs2). GRACE is an interactive ruler and compass construction editor for teaching the fundamental geometry concepts to high school students1). QuickDraw is a prototype sketch-based drawing tool that can recognize and beautify sketched diagram containing components and can enable users draw diagrams fast and precisely [3]. HabilisDraw is a drawing application that includes pens, ink wells,

* Corresponding author (email: tianfeng@iscas.ac.cn)

1) GRACE. http://www.cs.rice.edu/ jwarren/grace.
2) The Geometer's Sketchpad. http://www.keypress.com.

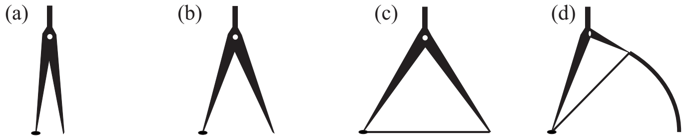

Figure 1 Steps in constructing an arc. (a) Setting the centre point; (b) adjusting the radius; (c) fixing the radius; (d) drawing the arc.

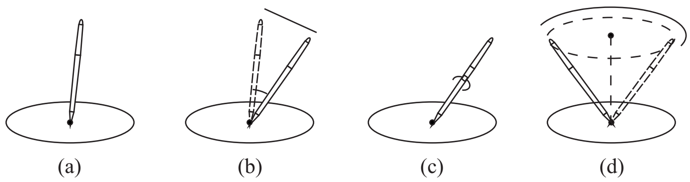

Figure 2 User's actions with Tilting-Twisting-Rolling. (a) Pen tapping to set the center; (b) pen tilting to adjust the radius; (c) pen twisting to set the radius; (d) pen rolling to draw.

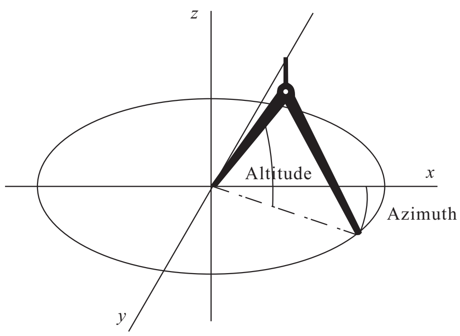

Figure 3 3D coordinates for modeling the virtual compass.

pushpins, compasses, rulers, and lenses [4]. Gulwani et al. [5] propose a ruler/compass-based geometry construction method, which can encourage the interactive learning activity in the classroom.

Tilting and rolling extend input channel in pen-based user interface [6–8]. To better support geometry construction with a pen, we propose a Tilting-Twisting-Rolling technique to facilitate drawing complex geometry like arcs through an uninterrupted pen action. Figure 1 illustrates key steps involved in drawing an arc with a compass on paper, and Figure 2 shows how the Tilting-Twisting-Rolling technique accomplishes each step.

2 Tilting, twisting, and rolling technique

2.1 Virtual compass design

Our design is based on a virtual compass metaphor. Figure 3 shows 3D Cartesian coordinates in modeling the compass.

The original of the coordinates is the pivot point of the compass. The viewpoint is at the positive infinity of a vector which has 45° angel both with positive y and positive z axis. The coordinates of the tip of the drawing leg of the compass can be calculated with the following:

$$\begin{aligned}
&\Delta x = \left(\text{altAdjust} - \frac{|\text{altitude}|}{\text{altF}}\right) \times \sin\left(\frac{\text{azimuth}}{\text{aziF}}\right) \times \cos(\text{altitude}),\\
&\Delta y = \left(\text{altAdjust} - \frac{|\text{altitude}|}{\text{altF}}\right) \times \cos\left(\frac{\text{azimuth}}{\text{aziF}}\right) \times \cos(\text{altitude}),\\
&Mx = Ox + 2 * \Delta x, \quad My = Oy - 2 * \Delta y,
\end{aligned}$$

where Δx /Δy is the difference between the end and the head of the cursor in x/y axis; altAdjust is the altitude zero adjust; altF/aziF is the altitude/azimuth factor; Mx and My are coordinates of the compass marginal point $M(Mx,\ My)$; Ox and Oy are coordinates of the compass original point $O(Ox, Oy)$.

The position of the top point of the compass can be calculated according to the following steps:

$$\text{Height} = \sqrt{\text{ArmLen}^2 - \Delta x^2 - \Delta y^2},$$
$$Tx = Ox + \Delta x, \quad Ty = Oy - \Delta y - \text{Height},$$

where ArmLen is the length of the arm in the compass, which can be calculated by computing the maximum of the length between original point and marginal point of the compass. Tx and Ty are the coordinates of the top point of the compass Top(Tx, Ty).

2.2 Pen twisting gesture recognition

Pen twisting gesture imitates the traditional screwing operation and is used to fix the radius of the compass. Twisting could go two directions, backward or forward. We capture pen information and detect pen twisting gestures every 50 ms. Pen twisting is related to pen position, pen orientation, and time. The pen operation from time period 1 to time period n could be represented as

$$\text{Info}\,(1, n) = \{(x_1, y_1, \text{alt}_1, \text{azi}_1, \text{twist}_1)\,, ..., (x_t, y_t, \text{alt}_t, \text{azi}_t, \text{twist}_t)\,, ..., (x_n, y_n, \text{alt}_n, \text{azi}_n, \text{twist}_n)\}\,,$$

where (x_t, y_t) is the pen position at time t, alt_t is the altitude, azi_t is the azimuth and twist_t is the twist information at time t.

To identify whether a twisting is backwards and forwards, we use three adjacent twisting extremums twist_{e1}, twist_{e2}, and twist_{e3} in Info(1,n) that satisfy the following conditions. e stands for the time stamp of a twisting extremum. These conditions indicate the temporal constraints, the position and the tilting constraints during pen twisting. We find a pen twisting gesture when the three extremums are found.

$$e2 - e1 > \text{thresTime}, \;\; e3 - e2 > \text{thresTime},$$
$$\frac{\sum_{i=e1}^{e3-1} |x_{i+1} - x_i|}{e3 - e1 - 1} < \text{thresOffsetX}, \;\; \frac{\sum_{i=e1}^{e3-1} |y_{i+1} - y_i|}{e3 - e1 - 1} < \text{thresOffsetY},$$
$$\frac{\sum_{i=e1}^{e3-1} |\text{alt}_{i+1} - \text{alt}_i|}{e3 - e1 - 1} < \text{thresOffsetAlt}, \;\; \frac{\sum_{i=e1}^{e3-1} |\text{azi}_{i+1} - \text{azi}_i|}{e3 - e1 - 1} < \text{thresOffsetAzi},$$

where thresTime, thresOffsetX, thresOffsetY, thresOffsetAlt and thresOffsetAzi are thresholds that ensure the twisting operation should last for a certain time period and the moving and tilting of pen do not change much during the pen twisting gesture. Currently, we set thresTime = 3, thresOffsetX = 100, thresOffsetY = 100, thresOffsetAlt = 100, and thresOffsetAzi = 100.

2.3 Interaction state transitions

Figure 4 shows a state transition diagram of the compass. Tapping a pen sets the pivot point of the compass as well as the center of a circle or an arc (State 1). The tip of the drawing leg automatically follows the position of the pen in 3D space and its coordinates can be calculated from formulas mentioned above. By tilting the pen, user can adjust the radius (State 2). Twisting the pen (the pen tip is on the pad) can set the desired radius (State 3). Then, the user can roll the pen to draw an arc (State 4). When drawing finished, the user can lift the pen and twist it to reset the pen state (St). It should be noted that if the pen is lifted only, the radius of the compass will remain fixed. In this situation, the user can use the compass to draw other arcs with the same radius in different position, unless the user twists the pen hovering above the pad.

2.4 Visualization techniques

We provide users with real-time visual feedback about the drawing action. The virtual compass follows pen movement and indicates the current state. When a task to draw an arc or a circle is initiated, the compass appears on the workspace and moves with the pen. When the user taps the pen on the screen, the pivot leg of the compass is anchored at the tapped position. Then, tilting the pen leads to the

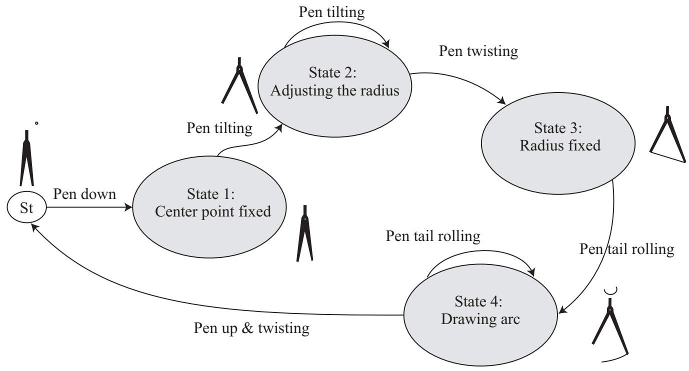

Figure 4 State transitions of the compass.

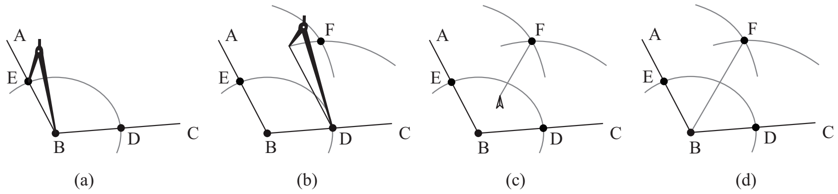

Figure 5 Drawing the bisector of an angle.

adjustment of the angle between the pivot leg and the drawing leg of the compass. After the radium is locked by twisting the pen, the compass turns by following pen rolling. When the user lifts the pen and changes the center with a fixed radium, the compass moves with the pen while the angle between the two legs keeps untouched.

Furthermore, four colors are used to render the compass to help users perceive state transitions(black-initial state, green-tilting state, red-fixed, yellow-rolling state). When the pen taps the screen, the compass changes from black to green and becomes active. Twisting the pen turns the compass to red, indicating the compass radium has been set and fixed. Then, the compass turns yellow when the user rolls the pen. When the user lifts and twists the pen, the compass becomes black and returns to the initial state, waiting for the user to reset the center and radius. If the user lifts the pen without twisting it, the compass becomes red and with a fixed radium. As long as the pen tags the screen again, the compass becomes yellow and is ready to draw.

Currently, almost all errors occur when user lifts the pen by accident. When the error occurs, the compass turns black, which indicates that it returns to the initial state and needs to restart drawing.

3 Application example

Figure 5 illustrates the process of using the Tilting-Twisting-Rolling technique to draw the bisector of an angle. The user first draws an arc that intersects with the two edges of the angle at points D and E (Figure 5(a)). In Figure 5(b), the user draws two arcs with points D and E as the center respectively and with the same radius. These two arcs intersect at point F. Figure 5(c) shows the process that the user taps point F and draws a straight line to point B. Figure 5(d) shows the final bisector. This process includes two steps that require drawing precise arcs. Points D and E should have the exactly same distance to point B. Point F needs to be generated from two arcs with the exactly same radius. Arcs created by beautifying sketches cannot guarantee these two conditions to be met.

4 Experiment

We conducted an experiment to compare user performances in drawing arcs between the Tilting-Twisting-Rolling (TTR), and traditional state-switching tool HabilisDraw (HD) [4]. Constrcting an arc with the

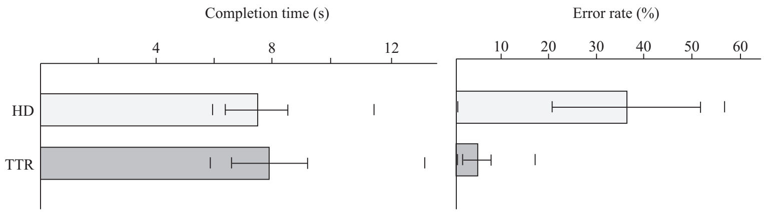

Figure 6 Mean movement time and mean error rate for two techniques. Mean values and the standard error for each series are shown.

compass tool in the HD treatment can be seen in Figure 1 in [4].

4.1 Subjects, apparatus and procedure

Twelve subjects participated in the experiment. To minimize experimental bias due to handedness, we ensured that all participants are right-handed and have normal or corrected-to-normal vision via self-report.

The pen device used in the study was a Wacom 6 in × 8 in touchpad, which provides 3D orientation and 3D rotation information of pen. The study used 17 in LCD screen with the resolution of 1024×768.

We adopted a within-subject design. Each treatment had 18 trials. In each trial, a template arc was displayed on the screen, and subjects were told to replicate the arc with the given tool, the TTR or HD tool, as quickly as possible. Nine different arcs were used in the study, and each arc appeared twice during 18 trials. All these nine arcs were 30°, and had three different radium, three of each, of 60, 80, and 100 pixels. Among each three arcs with the same radium, different starting and ending arms were applied: one arc started at 0°, relative to the positive x axis, and ended at the 30°; one started at –30° and ended at 0°; and one started at 30° and ended at 60°. The order of the experimental conditions for each participant was counterbalanced using a Latin square control for order effects. In order to limit carryover effects, the arc layouts of the TTR and HD are arranged in different orders. Each participant was given the opportunity to practice arc drawing before the real test.

User performances were measured by task completion time and error rate. Task completion time was the time interval between the time a template arc appeared and the time a subject lifted the pen. An error was recorded when a subject failed to follow the correct step required by arc construction under each condition. After completing all tasks, participants completed a questionnaire to rate these two tools from six aspects: fast to construct an arc, error prone, smooth in using, easy to learn, comfortable to use, and fun to use. Rating is on a Likert scale from 1 (worst) to 7 (best).

4.2 Results

Figure 6 compares user performances between these two conditions. No significant difference is found in terms of task completion time ($t_{11} = 0.743$, $p = 0.473$)[3](Figure 6(a)), but the TTR tool is found to led to significantly less errors than the state-switching HD tool ($t_{11} = 3.534$, $p = 0.005$) (Figure 6(b)).

Figure 7 compares subjects' perception of these two tools. Results from the Wilcoxon signed rank test indicate that the TTR tool is perceived more smooth ($Z_{11} = 2.78$, $p = 0.0027$) and more interesting ($Z_{11} = 2.78$, $p = 0.0027$) than the HD tool. Differences in the other four aspects are not significantly.

5 Discussion and conclusion

Results of the experiment demonstrate that our Tilting-Twisting-Rolling technique can significantly reduce the error rate without sacrificing task speed. We further examined pen trajectories and behavior data, and found that most errors in the HD condition were related to subjects' misconception of current

3) t is the value of t-test. It is used to determine if two sets of data are significantly different from each other. Z is the value of the Wilcoxon signed rank test. p-value is the probability for a given statistical model. When $p < 0.05$, it would be considered extremely significant. The subscript number 11 is the degrees of freedom.

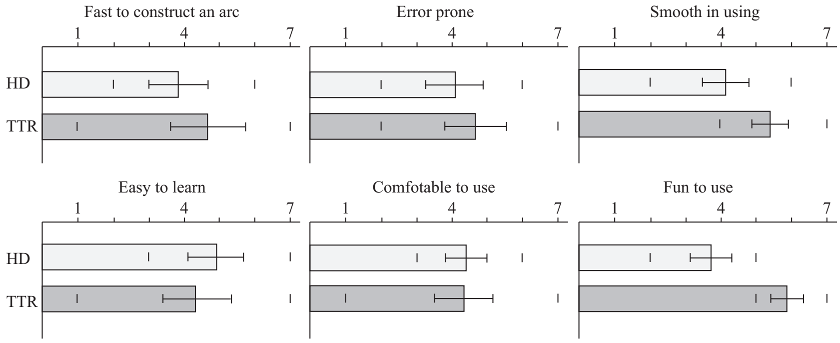

Figure 7 Subjective ratings. Median value and the standard error for each series are shown.

states. Since different states require different pen actions, subjects had to remember exactly what state has been chosen in order to construct arcs correctly. Comparatively, in the Tilting-Twisting-Rolling condition, subjects could merge all steps under one coherent pen action. Our results also indicate that this new design can improve the user experience in geometry construction. Users feel the tool is smooth and fun to use. It is interesting to find that although the Tilt-Twisting-Rolling tool leads to significantly less errors, subjects did not perceive this benefit. This might be due to the fact that subjects were familiar with the state-switching tool in the HD condition, a design popularly seen in many applications, and consequently underestimated the error they may make.

Our research can be extended in two ways. First, we like to study the integration of tilting, twisting, and rolling and investigate how they may be better coupled and how they may interfere with each other. Second, we want to go beyond geometry drawing and design tilting, twisting, and rolling tools for more generic multi-state interaction.

Acknowledgements This work was supported by National Key Research and Development Plan of China (Grant No. 2016YFB1001405), National Natural Science Foundation of China (Grant Nos. 61232013, 61422212, 61303162), Fundamental Research Funds for the Central Universities, and Beijing Municipal Social Science Foundation (Grant No. 16YTC033).

Conflict of interest The authors declare that they have no conflict of interest.

Supporting information The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Igarashi T, Matsuoka S, Kawachiya S, et al. Interactive beautification: a technique for rapid geometric design. In: Proceedings of the ACM Symposium on User Interface Software and Technology. New York: ACM, 1997. 105–114
2 Fiser J, Asente P, Schiller S, et al. Advanced drawing beautification with ShipShape. Comput Graph, 2016, 56: 46–58
3 Cheema S, Gulwani S, Laviola J. QuickDraw: improving drawing experience for geometric. In: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems. New York: ACM, 2012. 1037–1064
4 Daughtry J M, Amant R S. Power tools and composite tools: integrating automation with direct manipulation. In: Proceedings of the 8th International Conference on Intelligent User Interfaces. New York: ACM, 2003. 233–235
5 Gulwani S, Korthikanti V A, Tiwari A. Synthesizing geometry constructions. In: Proceedings of ACM SIGPLAN Confenrence on Programming Language Design and Implementation. New York: ACM, 2011. 50–61
6 Tian F, Lu F, Jiang Y Y, et al. An exploration of pen tail gestures for interactions. Int J Hum-Comput Stud, 2013, 71: 551–569
7 Xin Y Z, Bi X J, Ren X S. Acquiring and pointing: an empirical study of pen-tilt-based interaction. In: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems. New York: ACM, 2011. 849–858
8 Bi X J, Moscovich T, Ramos G, et al. An exploration of pen rolling for pen-based interaction. In: Proceedings of the ACM Symposium on User Interface Software and Technology. New York: ACM, 2008. 191–200