Skywriter: Drawing Curves from a First-Person Perspective

Steven Rybicki
Department of Computer Science
University of Cape Town
Private Bag X3, Rondebosch, 7701
South Africa
rybste001@myuct.ac.za

James Gain
Department of Computer Science
University of Cape Town
Private Bag X3, Rondebosch, 7701
South Africa
jgain@cs.uct.ac.za

ABSTRACT

Most research into sketching 3D non-planar curves is concerned with how best to take 2D input from a user to specify a 3D curve, exploiting familiarity with 2D sketches. *Skywriter* employs a different approach: instead of specifying a curve from a third-person perspective, users create curves by piloting a camera with a first-person perspective. We use depth cues, a cockpit-like heads up display and a simplified editing system to provide a system that is easy to learn without sacrificing accuracy. After testing with university students from different disciplines, we found that in comparison with a more traditional curve sketching approach, *Skywriter* is less accurate for complex curves that require large amounts of editing but is more accurate for simpler curves. Our usability tests also showed that *Skywriter* is slightly easier to use for the first time.

Categories and Subject Descriptors

I.3.3 [**COMPUTER GRAPHICS**]: Picture/Image Generation—*Line and curve generation*

General Terms

Design Experimentation

1. INTRODUCTION

The specification of non-planar 3D curves is a highly researched area in computer graphics since it is a critical task for 3D modelling and animation. Most approaches to solving this problem rely on taking a person's aptitude for sketching a two dimensional representation of the curve from the third-person perspective and then - using a wide variety of methods [1, 5, 7, 8] - extrapolate the curve.

This popular approach allows the user to sketch curves in a manner with which they are familiar, but suffers from problems caused by the extrapolation step. Extrapolation is a delicate balance between user control and time savings. If the 3D modelling program gives little help specifying the curve then the user has to undertake significant work to get any result, but the user gets exactly what they want. Conversely if there is a large amount of extrapolation from the user input then the design can be specified quickly but a large amount of time may be spent tweaking the results.

Skywriter presents a different approach to this problem allowing the curve to be specified directly from the first-person. Instead of sketching a curve using techniques inspired by pencil on paper, the user pilots a camera with the curve being specified as the path the camera takes. This is analogous to a skywriting plane writing messages with smoke.

Because *Skywriter's* input is simply piloting a camera, it avoids the problem of lost control suffered by some curve systems, while still maintaining the direct and simple input that such systems allow. The persistent popularity of flight simulators and first-person shooters indicates that navigation of a camera viewed from the first-person perspective is potentially a intuitive, and possibly developed, skill for a significant segment of the computer-literate population.

The elimination of 2D sketches from the modelling process could make the creation of 3D curves more accessible to those who lack adept fine motor skills. Fewer people have the ability to sketch accurately than those who have the skill involved in navigating a camera with a mouse and/or keyboard. This means that those who do not consider themselves artists and have traditionally been forced to chose between time intensive traditional 3D suites or sketching interfaces with which they have little skill, now have another option.

While *Skywriter* is robust, it is not intended to be used as a full editing suite. It is meant to be a proof of concept for a curve sketching program that is able to be both efficient and powerful. It enables a user to quickly sketch a relatively accurate approximation of the curve desired, with the intention of exporting it to a third party editing suite. The problem of editing has been solved by such suites already, *Skywriter* enables them to have a point from which to start this process.

2. RELATED WORK

Due to space constraints, here we outline some representative approaches for 3D curve sketching. For a more detailed survey, see Cook 2009 [2].

Curve creation from 2D sketches tends to be the focus for curve sketching interfaces. ILoveSketch [1] is a 3D curve sketching system that allows users to draw a series of 2D sketches on different axes. It uses these sketches to create a spline representing a three dimensional curve. This allows for rapid prototyping as the user can quickly make a number of freehand sketches. However, ILoveSketch was targeted at industrial modellers and as such requires a high level of accuracy in its input. This makes it inaccessible to many in the 3D modeller community.

Another 2D sketching approach was that taken by Teddy [5]. Here the user designs rotund objects such as stuffed animals. The user specifies a model by drawing a 2D sketch of the it head on and Teddy creates a 3D model by extrapolating on the widths of the areas defined: wide areas become 'fat', narrow areas become 'thin'. They found this

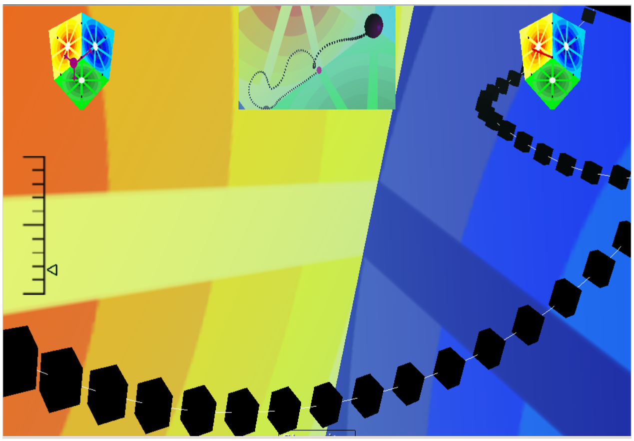

Figure 1: User interface of *Skywriter* displaying a partially sketched curve

to be a simple and effective curve sketching interface for creating such models. However, it has limited potential as a general purpose sketching interface as its performance is linked to its extensive extrapolation which is not possible for general curves.

To reduce the complexity of creating a sketch from scratch, suggestive sketching approaches [8] have been proposed. These use two dimensional sketching but are aided by images of the object that is being sketched. This allows the user's sketch to be guided by the image and the system's interpolation to be augmented by the image's shape. While useful for creating curves based on standard shapes, this approach is not practical for curves where providing an image is equivalent to drawing the curve itself.

3. INTERFACE

3.1 Views and Heads Up Display

As *Skywriter* is intended to be as intuitive as possible, it includes a number of redundant and overlapping means of viewing the curve and piloting the *Skywriter* camera.

The principal view for *Skywriter* is a central perspective view which can display either the "Observer" or *Skywriter* cameras, the former positioned to evaluate the curve being drawn, while the latter camera defines that curve.

To try help the user keep their orientation and position, some Heads up Display (HUD) elements were added, as shown in Figure 1. These included a direction arrow on the top right (which shows where the user is headed) a position indicator on the top left (showing where the user is in space) as well as a view at the top showing the Observer view. The Observer camera is navigable like the Skywriter camera and can be used to provide a 'birds eye view' to determine if the curve is turning out as expected. A speedometer is displayed on the left hand side of the screen indicating what speed the camera is travelling at.

3.2 Controls

The movement of the cameras is controlled by mouse or keyboard depending on the user's preference. The mouse had two distinct methods of movement. The first allows users to control pitch and yaw depending on the direction of mouse movement. The second calculates the speed of rotation based on the position of the cursor on the screen. Speed of rotation was based on the distance of the cursor from the center. The pitch and yaw were inferred from the angle between the cursor and a horizontal line through the center of the screen. The former method provides the user more fine grained control. The latter gives users the ability to make circular arcs more easily.

The keyboard input is similar to the first mouse control method: the angle of rotation was controlled by the direction indicated by the user using the arrow keys.

Editing in the *Skywriter* camera is done by the user indicating that they want to reverse course. This makes the camera backtrack along the curve erasing points as it comes into contact with them. The speed of the camera is controlled through the keyboard or the mouse wheel. This was added to allow the user to slow down for intricate details or speed up for larger curves.

4. DESIGN OF THE VISUAL CUES IN THE ENVIRONMENT

Particular attention was paid to the design of the environment where the user pilots the camera. To draw a curve, the user judges where they are, evaluates how accurate their curve is and chooses where to fly next. This is based almost entirely on the view from the camera they are piloting. Therefore the environment must not only provide this information but do so quickly and accurately in order for curve drawing to be robust.

The environment is contained within a cube with brightly coloured faces. Each face of the cube has differently coloured concentric circles on it, with opposing faces sharing colour schemes. This allows users to quickly orientate themselves and plan their movement based on a face's appearance alone, saving time that other environments lose by forcing the user to orientate themselves using orthographic planes.

Circles were chosen as they provide a strong indication of how far the *Skywriter* camera is away from the center of each of the containing cube's faces, allowing users to gauge how far they have to travel before making their next turn. The curved edges of the circle's also allow users to quickly gauge the steepness of the curve they are producing as they can compare it to the circles curvature.

Nine spheres are placed as depth cues within the containing cube: one in the center of the environment with the other eight arranged around it to form a cube with side length approximately half of the larger containing cube. These allow users to estimate their distance from the center.

Fog is also used as a depth cue to allow the user to assess larger distances across the cube.

The curve itself is rendered as a white line with black cylindrical markers dashed along it. This allows the curve to be picked out easily from the other colours present in the environment while allowing portions of a curve to be created without obstructing any element of the environment.

5. TESTING METHODOLOGY

5.1 Experimental Procedure

To test the efficacy of *Skywriter*, an usability and accuracy test was run with 19 students from different faculties and differing levels of experience with three dimensional thinking and modelling.

The accuracy test required subjects replicate physical wire curves that were hung in front of them. They had to replicate four of the six available curves (described in

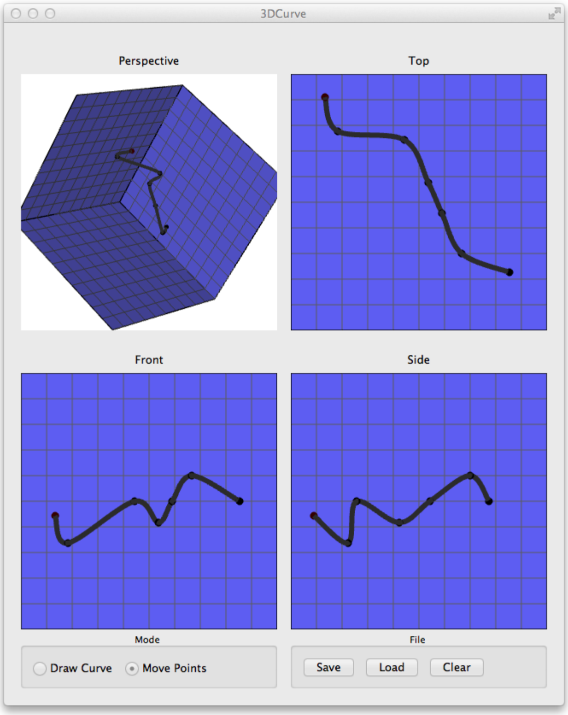

Figure 2: User interface of the more traditional interface used to compare the accuracy of *Skywriter* curves

the next section): two in *Skywriter* and two in a more traditional curve sketching interface of similar sophistication depicted in Figure 2. The traditional software enabled users to sketch curves in one of the orthographic planes and then adjust the points as needed in a three dimensional view. To control for familiarity and experience, the selection of curves as well as the order in which the software was used were randomised.

Each participant was given a manual describing how each program worked and allowed eight minutes to familiarise themselves. After this, the participant was given five minutes to replicate each curve. They then repeated this procedure with the alternate program.

After the accuracy test was completed, participants were given a questionnaire based on the IBM computer usability satisfaction questionnaire [6]. This was aimed at determining which approach was the most usable after a brief exposure to determine whether *Skywriter* has a different difficulty curve from traditional approaches.

5.2 Assessing Curve Accuracy

To assess the accuracy of the participants' curves, digital reference curves were created for each wire curve. These were done using pictures of the curves taken on orthographic planes to ensure a high degree of accuracy.

The six curves were:

S-Shaped Curve A period of a sinusoidal graph

Spiral A tight spiral with several coils

Bird's Camera Path a path that mimics a bird circling, dipping down then returning to the starting point

Intersecting ovals Two oval paths perpendicular to each other, intersecting at two of their edges

Local perturbations A sinusoidal curve with several small perturbations

Points of inflection A large cubic-like curve with a single point of inflection

Since the curves were captured as piecewise linear approximations for the purposes of analysis, we finely subsample them using linear interpolations. This gives us a curve defined by a sequence of vertices. We then define a distance metric d between two curves A, B each comprised of n vertices A_i, B_i as follows:

$$d(A,B) = \frac{1}{n}\left(\sum_{i=0}^{n} \min_{j=0}^{n} |A_i - B_j|\right) \tag{1}$$

.

To compare the shapes of the curves, we used Procrustes analysis [4]. This removes the translational, scaling and rotational components so that curves with the same shape will coincide exactly. It does this by this translating and scaling the curves so that they are both centered and contained in the same bounding box. We then remove the rotational component of the curve using the Kabsch algorithm [3]. This calculates a rotation matrix that when applied minimises the root-mean-square deviation between the curves.

6. RESULTS

6.1 Curve Accuracy

The results are summarised in Table 1. Although the results are not statistically significant, the test suggest that *Skywriter* performed slightly better on the S-Shaped and Spiral curves while under performing on the Bird's Camera Path, Intersecting Circles, and Local Perturbations. Both approaches seem to perform comparably on the Points of Inflection curve.

To fully explain these results requires further experimention. Below we offer some hypotheses to explain the results found that can be tested in future studies.

For the S-Shaped Curve and Points of Inflection, one could see that sketching these is a simple task while operating from a first-person perspective (flying upwards then downwards in succession) whereas from a third-person perspective this could involve large amounts of calibration. Similarly, the performance of *Skywriter* for the Spiral curve could also be explained by the difference of controls: spirals in *Skywriter* involves simply holding an angle and going downwards but requires large amounts of recalibration in the third-person.

From the results, we can see that Skywriter under performs for the more complex curves: the Bird's Camera Path, the Intersecting Ovals, and the Local Perturbations. According to the survey comments of users who sketched these curves in Skywriter, this is due to the difficulty of determining which movements will lead to the correct shape while in the first-person view. While for the simpler curves, one can determine this by the heading alone, for these curves you need to keep in mind your heading in relation to the general shape of what you have drawn already. For longer curves, it becomes difficult to keep track of especially in time-limited situations.

6.2 Usability

Responses to both approaches tended to be polarised; participants tended to rank each approach towards the

Table 1: Results from the accuracy testing of *Skywriter*. These are calculated from the curves created by users during the experiment outlined in Section 5. Distances are calculated between the user curves and the reference curves.

Description of curve	Mean distance (third-person)	Mean distance (Skywriter)	Standard Deviation (third-person)	Standard Deviation (Skywriter)	Student t-test	Student two-tailed probability
S-Shaped Curve	0.2782	0.2245	0.0572	0.0509	1.7195	0.1112
Spiral	0.3705	0.3310	0.1105	0.0771	0.6930	0.5015
Bird's Camera Path	0.3205	0.4170	0.0908	0.0406	-2.8015	0.0134
Intersecting Ovals	0.3868	0.4214	0.0581	0.0459	-1.1014	0.2942
Local Perturbations	0.2536	0.4435	0.0548	0.0459	-4.2940	0.0010
Points of inflection	0.4342	0.4449	0.1169	0.0609	-0.2184	0.8301

extremes according to their experience. As such, few statistically significant usability results were found.

Overall, users seemed more satisfied with the traditional approach with some vocal users saying that the first-person perspective made them feel like they had less control. Participants did say that *Skywriter* presented information more effectively, enhancing their ability to recover easily and quickly from mistakes as well as providing information that was easy to find and understand.

They also stated that they thought that this information did not help them accomplish tasks as efficiently as the traditional approach did. In the comments, the users stated that the reason for this was the disorientation of trying to edit their curves in the first-person.

Creating first-person visual cues to address this deficiency is not sufficient as we found that users need to see their position from the third person in order to orientate themselves. During heuristic testing users indicated that they were not be able to determine their position in the environment without that position being projected onto orthographic planes.

7. CONCLUSION

We have shown that *Skywriter* outperforms traditional interfaces for certain simpler curves as well as presenting an interface that is easier to interpret. However, we also found that it is not able to cope with more complex curves due to problems with its editing functionality.

If *Skywriter* could be incorporated as a drawing mode within a more sophisticated editing suite, it would then be possible to leverage the advantages of both approaches. Users found it difficult to define an initial shape for a curve in the traditional editor but found this easier in *Skywriter*. Users found it easier to edit points on a curve in the third-person as it was harder to get perspective in the first-person. This suggests that a productive pattern would be to create curves piece by piece, using *Skywriter* to sketch out the basic shape and then using a traditional suite to refine the curves.

Skywriter solves some problems with traditional third-person editing and should be an effective way to help sketch three dimensional curves in a user friendly way. While it does not solve the problem of curve sketching, it provides a method to provide rough drafts of curves that can be refined, helping accelerate the process of 3D curve design.

Future work could examine the efficacy of different controllers used for first person curve sketching. Here we provided users with mouse or keyboard input. Other inputs such as joysticks and other game controllers could provide a more familiar and effective method of control and present a possible avenue for further investigation.

8. REFERENCES

[1] S.-H. Bae, R. Balakrishnan, and K. Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 151–160, New York, NY, USA, 2008. ACM.

[2] M. T. Cook and A. Agah. A survey of sketch-based 3-D modeling techniques. *Interacting with Computers*, 21(3):201–211, 2009.

[3] E. A. Coutsias, C. Seok, and K. A. Dill. Rotational superposition and least squares: The svd and quaternions approaches yield identical results. reply to the preceding comment by g. kneller. *Journal of computational chemistry*, 26(15):1663–1665, 2005.

[4] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991.

[5] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[6] J. R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[7] L. Markosian, J. M. Cohen, T. Crulli, and J. Hughes. Skin: a constructive approach to modeling free-form shapes. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 393–400, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[8] S. Tsang, R. Balakrishnan, K. Singh, and A. Ranjan. A suggestive interface for image guided 3D sketching. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 591–598, New York, NY, USA, 2004. ACM.