Virtual Cane Creation for Glassblowers

Andrew Winslow[a], Kimberly Baldauf[b], Benjamin Lee[b], James McCann[c], Erik Demaine[b], Martin Demaine[b], Peter Houk[b]

[a] *Université Libre de Bruxelles,* `awinslow@ulb.ac.be`
[b] *MIT,* `{sakis,bleeben,edemaine,mdemaine,pbhouk}@mit.edu`
[c] *Disney Research and Carnegie Mellon University,* `jmccann@cs.cmu.edu`

Abstract

One of the fundamental color application techniques used by advanced glass artists is the creation and use of thin glass rods called *cane*. Producing and applying glass cane is a many-step process, making its use daunting for intermediate-level glassblowers. In this work, we present *VirtualGlass*, a computer program designed to help intermediate glassblowers plan and visualize all the steps involved in the application of cane. This allows them to rapidly iterate and explore virtually, rather than spending hours in a glass shop. By way of evaluation, we present side-by-side results of cane patterns designed in our program and the final pieces; and discuss the results of a user test among intermediate-level glass blowers.

Keywords: cane, glass, glassblowing

1. Introduction

The creation and use of cane is an advanced glassblowing technique which can create amazing color patterns in finished pieces. Canes are thin (0.2-0.6 inch) glass rods with internal color patterns. They are created by building short, fat, cylinders of glass with the desired pattern, then stretching these cylinders 5-20 feet to thin them out. Once created, canes can be used recursively to build new canes, or applied to finished glass pieces for amazing and intricate color effects (Figure 1).

In this paper, we discuss *VirtualGlass*[1], a program which can help glassblowers design cane and plan how to use it. We target our tool at glassblowers who have good basic skills, but have little-to-no experience actually creating cane.

The workflow in our tool mirrors the workflow in the glass studio (Figure 2), with the important difference that in our tool, nothing is immutable (e.g. you can go back and change the colors if the final piece doesn't look right). This encourages users to experiment and explore without worrying about available shop time.

Our software serves several purposes: foremost, it is a tool for design and ideation, specifically tailored to the constraints and processes of glass cane; additionally, it can help users to understand existing cane by attempting various creation methods; and, finally, it can be used as a communication tool, to explain goals to others helping in the glass studio.

2. Background

One of the foundational functions of computer graphics has been to help creative people design things more effectively. While the kind of general purpose 2D and 3D creation programs envisioned by early pioneers (e.g. [1]) are readily available to everyday users, domain-specific design programs can still provide additional benefit. Developing a modeling package with a specific set of users in mind also enables the set of operations to be oriented towards the users tasks and constraints. Computer systems for cel animation [2, 3] were designed around multi-person workflows and asset reuse – both essential for rapid-fire television production. Plushie, a stuffed-toy design application, accounts for domain-specific quirks like shape change due to cloth stretch and pattern layout [4]; whereas a knitted creature design variation calculates stitch-counts [5].

As a fluid with a dramatically varying viscosity, hot glass could be simulated effectively with traditional techniques for fluids with changing states (e.g. [6]). The Museum of Glass in Tacoma, Washington has a website that introduces users to the steps of glass-blowing using interactive animations, allowing color customization.[2] The Glasmuseet Ebeltoft, a glass museum in Denmark, has an interactive tabletop computer that allows visitors to engage in touch-based glass design as part of

[1] `http://virtualglass.org`

[2] `http://museumofglass.org/document.doc?id=142`

Figure 1: Two applications of cane by master glassblower Lino Tagliapietra. The form on the right uses cane inspired by an early version of our tool.

its “World of Glass” exhibit opened in 2010.[3] The “Digital Glass Blower” allows users to heat a piece and adjust its size and shape by touch or using tools, as well as adding color to regions of the vessel. This exhibit focuses on the shaping of glass forms, and only supports limited color application techniques.

3. Interface

Our interface is divided laterally into three parts (Figure 3). The leftmost part – the library – provides an overview of the colors, canes, and pieces designed during this session; it also indicates the current object and its dependencies/dependants. The middle and right parts (the editor and preview, respectively) change depending on the type of the current object. We describe their functions below.

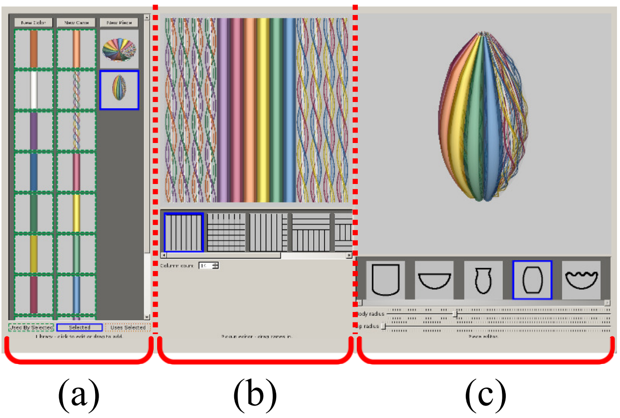
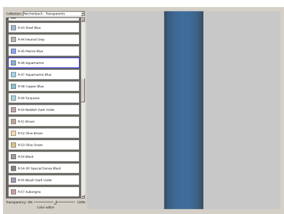


Figure 3: The parts of our interface: **(a)** the library, which provides selection; **(b)** the editor, which changes depending on the selected object; and **(c)** the preview.

3.1. Color



Most colored glass is produced by a few companies and distributed in the form of color rods. Glass color is expensive ($30-$50 a rod) and difficult to mix to form intermediate shades. As such, most glass artists only have access to (or want to buy) a few colors, and those colors are most likely drawn from the catalogue of one of a few manufacturers.

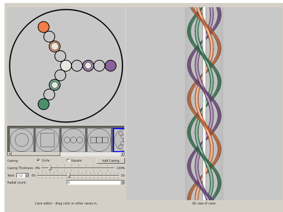
Thus, our editor for color shows a dropdown menu to select manufacturer (Reichenbach, Gaffer, or Kugler) and color type (transparent/opaque), and a list containing swatches mimicking the appearance of the currently selected color set. The lists of colors are stored in extrernal files in a human-readable, JSON-based format. Users can create and load their own color lists in this format using the interface.

[3]`http://www.glasmuseet.dk/en2010/events/world-of-glass.html`

3.2. Canes

When creating cane, glass artists first set up an internal color pattern in a compact *layout*. The glass in this pattern can be clear glass from the furnace, colored glass rods, or even sections of previously created cane. This pull setup is then stretched or *pulled* into a long rod – the finished *cane*.

Our interface represents these many construction scenarios uniformly as canes. Canes describe the location of colored elements in the cane layout, and the amount of twist applied to the layout during the pull. The cane itself remains an abstraction of a real cane: the radius and length of the pull plan are unspecified. This is done for ease in designing pickups (discussed in the next section), as it allows canes to be used freely in pickups.

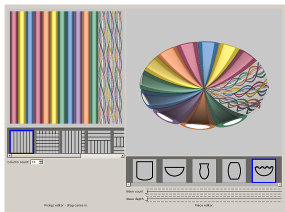


Our editor for pulls prominently features a 2D schematic, while the preview portion of the interface shows a 3D side view of the resulting cane. The schematic view provides easy manipulation of the internal structure of the cane, while the side-view shows how twisting will affect the internal structure.

The overall structure of the cane’s layout can be selected from a library of common layouts. These layouts are further customizable by sliders that vary, for instance, the number of color bars used or the width of the clear casing around the outside. The individual elements of the cane (colors or other canes) are selected by dragging from the library into the schematic view. Users can also choose to customize a layout completely in an alternate view, allowing new elements to be added, deleted, moved, and rescaled.

3.3. Pickups and Pieces

Once cane has been made, it is cut into sections, and these sections are arranged into a *pickup* that will be heated, picked up, and blown into a *piece*.



Our editor for pickups works in much the same way as the editor for canes; a pickup layout is selected and customized and canes are dragged from the library to fill it. The pickup editor occupies the center view of the interface, paired with a piece editor in the right view. The piece editor visualizes the pickup shaped into a piece. The piece’s shape can be selected from a library of common shapes, parameterized by popular manipulations done to real glass pieces.

For example, the vase shape has two parameters: the diameter of the body (corresponding to how much air is added when blowing), and the diameter of the lip (which can be flared to various radii with a hand tool). Increasing the size of the body is done by blowing air into the closed vessel, while increasing the lip diameter is done by working with a handtool to bend the glass outwards. This variety of forms and parameters is important, because the distortion of the cane caused during the inflation process can dramatically change its appearance.

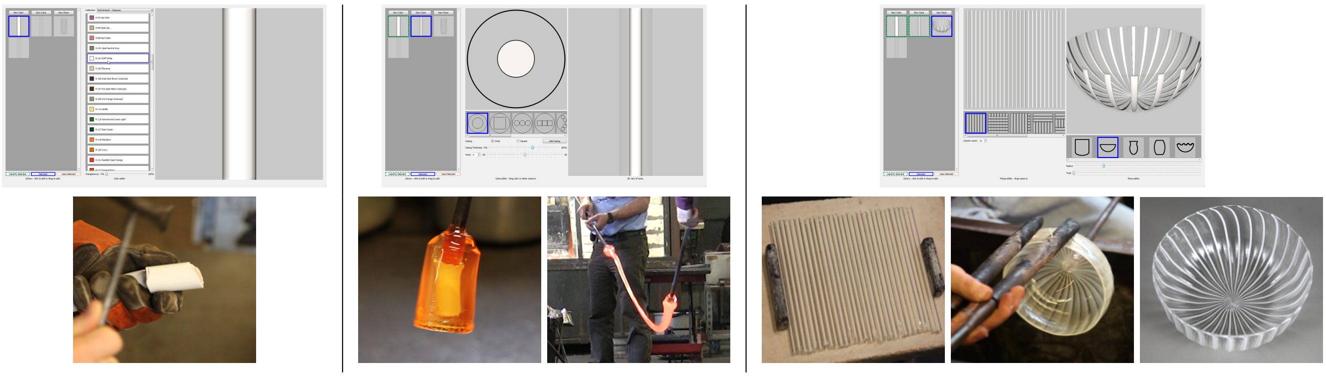


Figure 2: A glass piece with cane decoration is planned in our program (top), then executed in the hot shop (bottom). Columns show the steps corresponding to each of our interface sections.

Piece shapes are represented as curves spun around an axis. Shapes can be customized completely in an alternate view that gives users access to add, remove, and modify underlying spline control points.

4. Implementation

Under the hood, our system uses a DAG (directed acyclic graph) to represent each cane. Leaf nodes of the DAG correspond to color bar. Each internal node corresponds to a cane with specified twist; thickness and color of casing glass; and pattern of subcanes. For each subcane in the pattern, a pointer is stored to another DAG node. This allows canes to include canes created previously. We enforce the acyclic nature of the DAG so that no cane may be a subcane of itself. The resulting dependencies between objects, for instance, which canes are used in a piece, are represented in the library view of the interface.

To generate a 3D mesh for a cane, first 3D meshes are generated for all child nodes. Then these meshes are translated and scaled to their proper position within the pattern, cylindrical geometry is added for the casing, and twist is applied. For color bar, only cylindrical geometry is generated (as there are no child nodes). The resolution of the generated geometry is set heuristically based on the amount of twist and scaling that will be applied to the mesh. A background process continually rerenders geometry at increasing resolutions, allowing users to optionally view extremely high-resolution renderings simply by waiting a few additional seconds.

To generate a 3D mesh for a cane, first 3D meshes are generated for all child nodes. Then these meshes are translated and scaled to their proper position within the pattern, cylindrical geometry is added for the casing, and twist is applied. For color bar, only cylindrical geometry is generated (as there are no child nodes). The resolution of the generated geometry is set heuristically based on the amount of twist and scaling that will be applied to the mesh. A background process continually rerenders geometry at increasing resolutions, allowing users to optionally view extremely high-resolution renderings simply by waiting a few additional seconds. To generate a 3D mesh for a cane, first 3D meshes are generated for all child nodes. Then these meshes are translated and scaled to their proper position within the pattern, cylindrical geometry is added for the casing, and twist is applied. For color bar, only cylindrical geometry is generated (as there are no child nodes). The resolution of the generated geometry is set heuristically based on the amount of twist and scaling that will be applied to the mesh. A background process continually rerenders geometry at increasing resolutions, allowing users to optionally view extremely high-resolution renderings simply by waiting a few additional seconds.

When generating geometry for a pickup, different lengths of cane are requested from the DAG nodes to fill different length spots in the template. Dropping a cane into a pickup generates a version of the cane with appropriate length and diameter for the location. Diameter is adjusted by scaling, but length corresponds to a specified z-interval of the cane: shorter lengths of cane have fewer twists. The final piece geometry is created by deforming the pickup plan geometry, according to the chosen shape and parameters. Pickup geometry is transformed according to the requested into a piece. Pieces have a variety of shapes, and each shape has a set of parameters that fine-tune the transformation of the pickup into the desired shape. These transformations are approximately volume-preserving. For example, a large-diameter bowl will have thinner walls, changing the appearance of the cane's internal structure through flattening.

Because of the underlying graph-based representation of colors, canes, pickups, and pieces, designs can be stored compactly, and users have the option to save their designs in a human-readable, JSON-based file format. Users may choose to save a single design or many, and a dependency analysis ensures that all objects used in the design are saved along with it.

5. Results

The software was used in the design of a number of canes, pickups, and pieces (Figures 2, 4, 5, 6, 8). Figure 8 demonstrates a small design space explored in the software. Even in very complex pieces, the software is fast enough to allow real-time movement and modification of a piece in the editor.

The majority of glassblowers currently use hand drawings to plan their designs, and verbal descriptions are used to explain to team members the parts composing a finished piece (moving from a pickup to a finished piece usually requires two or more people). Giving the software to a number of experienced glassblowers, we observed the software being used in a number of ways in the glassblowing process: as a visualization tool (Figure 4), a design tool (Figure 5), and a communication tool (Figure 6).

A perfect correspondence does not always exist between the designed and executed pieces. Some of these differences are due to the glassblower making additional design decisions during the creation of the piece (e.g. Figure 4); while others result from modeling pieces using simple analytic functions (e.g. the slight twist in the cane in Figure 5 is a common production artifact, but is not modeled).

Glassblowers can also use our system to "reverse engineer" canes by experimenting with possible production processes and internal patterns. In Figure 7, a hand-drawn diagram of cane patterns belonging to the zanfirico style are shown, as well as several reproductions in our interface.

6. Evaluation

In order to evaluate our interface, we asked glassblowers to try it. Subjects were presented with a laptop computer with mouse and the software loaded. Each subject was first asked to produce a specified set of three canes and one bowl (ex: "An opaque cane with color R-61 Extra Enamel White Opaque cased in R-100 Lead Crystal (Clear)."). Next, the subjects were asked to create four sets of three objects with a common theme (ex: "Three pieces with the same shape but different canes."). Finally, an open-ended task was given in which the subject was asked to create a cane, pickup, and piece of their own design.

Five subjects aged 23-55 with glassblowing skills ranging from novice to advanced were asked to complete the list of tasks. All subjects were able to complete the entire list of tasks in less than 90 minutes. Afterwards, each subject was given a questionnaire about their experience using the software. The questionnaire contained 19 questions for evaluating engagement derived from those suggested by O'Brien, et al. [7] and 5 questions for measuring workload from the NASA TLX [8]. Aggregate user response means in six categories of engagement and five categories of workload on a 1-5 scale were computed, with 1 corresponding to most negative, 3 neutral, and 5 strongly positive. For instance, a score of 5 for the involvement category indicates that the subject felt highly involved in using the software, while a score of 5 for the effort category indicates the subject felt that completing the tasks required very little effort. Scores for each category were computed by averaging the responses by all subjects to all questions belonging to the category, with workload categories containing one question each. The results can be seen in Tables 1a and 1b.

Engagement	
Category	Score
Focused Attention	4.4
Perceived Usability	3.9
Aesthetics	3.6
Endurability	4.0
Novelty	4.1
Involvement	4.4

Workload	
Category	Score
Mental	3.0
Physical	4.8
Performance	4.4
Effort	3.2
Frustration	4.4

Table 1: Aggregate data of subject engagement and workload while using the software, measured by responses to a questionnaire completed after the study. Results in both tables are on a 1-5 scale of the software's performance, with 1 being most negative and 5 being most positive.

We also asked the subjects what they did and did not like about the software. Subjects responded that "I like the idea of being able to more easily visualize the results of design decisions without having to execute them in glass" and that they "enjoyed the fact that this will be useful to making glass". How-

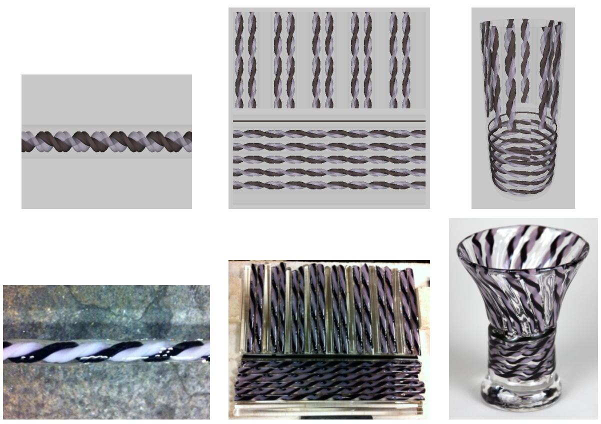


Figure 4: A piece with dark purple and pink twisted cane, designed (top) and created (bottom) by an intermediate-level glassblower. Our interface inspired the glassblower to try a pickup with perpendicular canes. The glassblower visualized the pickup on a cup during design, but decided during production to flare the top.

Figure 5: A striped cup, designed and created by an advanced glassblower who seldom uses cane (bottom). The glassblower had the purple and blue canes on hand, and modeled them in our program (top) before deciding on a pickup and shape.

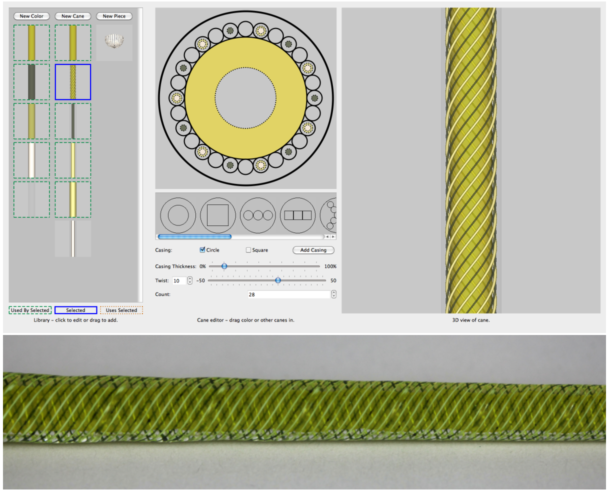


Figure 6: A complex cane designed in our interface and created by an expert glassblower. In this case, the design was used to communicate the intent of the expert to others present.

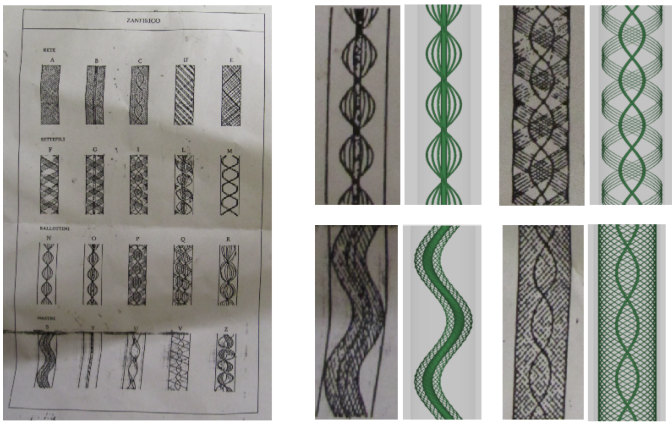


Figure 7: Hand-drawn diagrams of various zanfirico-style cane patterns (black), as reverse-engineered in our software (green).

ever, a common complaint was that there was no way "to delete unused work/mistakes." (We have since added a delete function.)

In total, subjects found the tool to be useful for exploration and visualization of cane and piece designs. One subject enthusiastically remarked, "I found I had to keep going back to the [task] list, because I wanted to keep going off and playing with the designs." The shortcomings of the software noted by the subjects were primarily related to missing features such as the ability to arrange and delete designs, create new pickup layouts, and the ability to save and load designs. We have since added the ability to delete designs, as well as save and load designs.

7. Discussion

In this paper, we describe a tool which enables glassblowers to rapidly experiment with cane design. This tool allows glass artists to play with glass cane virtually, so they can make the most of their limited hot shop time and glass color budget.

From a technical standpoint, one of the greatest limitations of the software is the rendering. While our rendering is suggestive of the finished structure and appearance of the glass pieces, and performs at interactive frame rates, it falls short of predictive display. In the future, a data-driven model of glass color could be constructed to rectify this deficiency. Better modeling of the interplay between light and glass, including the refractive properties of glass, would also give an improvement in the realism of the rendering.

While the software contains a basic set of common cane setups, pickups, and piece shapes, the set is far from being comprehensive. In particular, although users can customize cane layouts and piece shapes to some extent, the ability to completely customize the these, as well as pickup layouts would allow glassblowers to further experiment and explore glass designs.

Of course, being able to design a cane in our software does not imply that a glassblower can create or apply that cane successfully in the studio. The manual dexterity required to balance and shape glass, the understanding required to properly manipulate internal heat in a setup, and the control required to create perfect forms from pick-ups – these only come with experience. With this in mind, perhaps the most important feature of our system is that it gets glassblowers excited about cane. Everyone we have demonstrated the system to has gotten excited about using cane for the first time, adding to their stable of designs, or experimenting with a new pick-up style. And while this excitement may lead glassblowers reach beyond their ability, even their failures will pay dividends in skill.

Acknowledgements

We thank the members of the MIT Glass Lab and VirtualGlass users that provided valuable feedback on the software.

References

[1] Sutherland, I.E.. Sketchpad: A man-machine graphical communication system. Ph.D. thesis; MIT; 1963.

[2] Durand, C.X.. The "TOON" project: Requirements for a computerized 2d animation system. Computers & Graphics 1991;15(2):285–293.

[3] Fekete, J.D., Bizouarn, E., Cournarie, E., Galas, T., Taillefer, F.. TicTacToon: a paperless system for professional 2D animation. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95; ACM; 1995, p. 79–90.

[4] Mori, Y., Igarashi, T.. Plushie: an interactive design system for plush toys. In: Proceedings of SIGGRAPH 2007. ACM Press; 2007,.

[5] Igarashi, Y., Igarashi, T., Suzuki, H.. Knitty: 3D modeling of knitted animals with a production assistant interface. In: Proceedings of Eurographics 2008. Crete, Greece: Eurographics Association; 2008,.

[6] Carlson, M., Mucha, P., III, B.V.H., Turk, G.. Melting and flowing. In: Proceedings of the ACM SIGGRAPH Symposium on Computer Animation. 2002,.

[7] O'Brien, H.L., Toms, E.G., Kelloway, E.K., Kelley, E.. Developing and evaluating a reliable measure of user engagement. In: Proceedings of the American Society for Information Science and Technology. 2008,.

[8] Hart, S.G., Staveland, L.E.. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In: Hancock, A., Meshkati, N., editors. Human Mental Workload. Amsterdam: North Holland Press; 1988,.

Figure 8: A matrix of pieces designed in the software, varying by shape, pickup, and cane composition.