AdaptiBrush: Adaptive General and Predictable VR Ribbon Brush

ENRIQUE ROSALES, University of British Columbia, Canada and Universidad Panamericana, México
CHRYSTIANO ARAÚJO, University of British Columbia, Canada
JAFET RODRIGUEZ, Universidad Panamericana, México
NICHOLAS VINING, University of British Columbia, Canada and NVIDIA, Canada
DONGWOOK YOON, University of British Columbia, Canada
ALLA SHEFFER, University of British Columbia, Canada

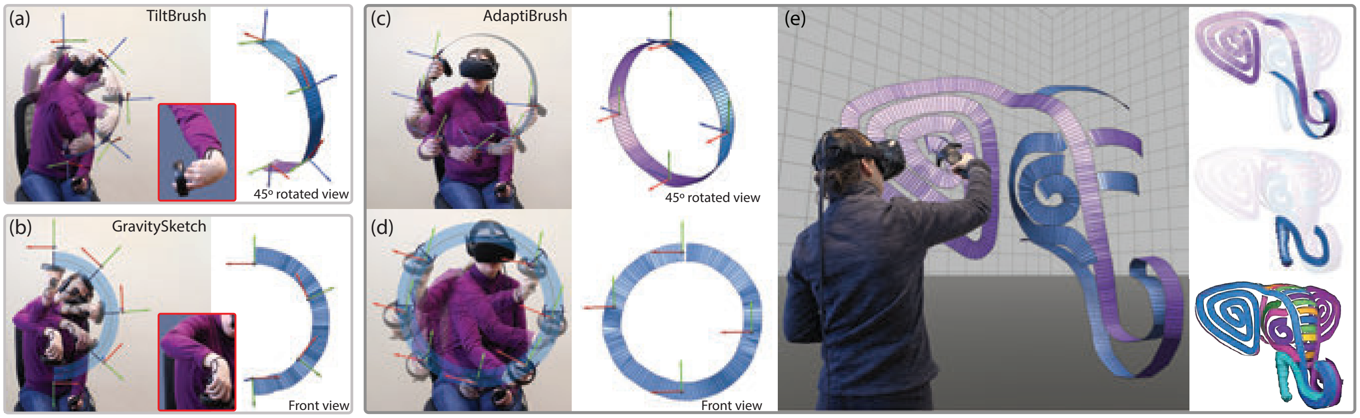


Fig. 1. VR ribbon drawing interfaces let users draw ruled surface ribbon strokes (shown via double-sided blue-purple rendering) by moving a 6DOF controller in space. (a) *Cross-product based* methods (e.g. TiltBrush [2019]) require unnatural wrist twisting (inset) to draw consistently oriented curved ribbons such as semi-cylinders, and are prevented by joint biomechanics from drawing complete cylinders (left, overlay of user poses during the drawing process; right, output ribbon and representative controller positions/orientations during drawing; inset, wrist pose corresponding to the last frame user was able to draw). (b) *Direct interfaces* (e.g. GravitySketch [2019]) require similar unnatural wrist exertion when drawing bent ribbons such as semi-circles; biomechanical constraints prevent their users from drawing shapes such as full circles. (c-e) Our AdaptiBrush VR ribbon brush lets users comfortably draw curved ((c) and (e), tusks) and bent ((d) and (e), ear) ribbons of varying complexity. User study participants strongly prefer AdaptiBrush over all existing alternatives. See supplementary video for complete motion. Elephant: © Elinor Palomares.

Virtual reality drawing applications let users draw 3D shapes using brushes that form *ribbon* shaped, or ruled-surface, strokes. Each ribbon is uniquely defined by its user-specified ruling length, path, and the ruling directions at each point along this path. Existing brushes use the trajectory of a handheld controller in 3D space as the ribbon path, and compute the ruling directions using a *fixed* mapping from a specific controller coordinate-frame axis. This fixed mapping forces users to rotate the controller and thus their wrists to change ribbon normal or ruling directions, and requires substantial physical effort to draw even medium complexity ribbons. Since human ability to rotate their wrists continuously is heavily restricted, the space of ribbon geometries users can comfortably draw using these brushes is limited. These brushes can be unpredictable, producing ribbons with unexpectedly varying width or flipped and wobbly normals in response to seemingly natural hand gestures. Our *AdaptiBrush* ribbon brush system dramatically extends the space of ribbon geometries users can comfortably draw while enabling them to accurately predict the ribbon shape that a given hand motion produces. We achieve this by introducing a novel *adaptive* ruling direction computation method, enabling users to easily change ribbon ruling and normal orientation using predominantly translational controller, and thus wrist, motion. We facilitate ease-of-use by computing predictable ruling directions that smoothly change in both world and controller coordinate systems, and facilitate ease-of-learning by prioritizing ruling directions which are well-aligned with one of the controller coordinate system axes. Our comparative user studies confirm that our more general and predictable ruling computation leads to significant improvements in brush usability and effectiveness compared to all prior brushes; in a head to head comparison users preferred AdaptiBrush over the next-best brush by a margin of 2 to 1.

ACM Reference Format:
Enrique Rosales, Chrystiano Araújo, Jafet Rodriguez, Nicholas Vining, Dongwook Yoon, and Alla Sheffer. 2021. AdaptiBrush: Adaptive General and

---

Authors' addresses: Enrique Rosales, University of British Columbia, Canada, Universidad Panamericana, Facultad de Ingeniería, Zapopan, Jalisco, 45010, México, albertr@cs.ubc.ca; Chrystiano Araújo, University of British Columbia, Canada, araujoc@cs.ubc.ca; Jafet Rodriguez, Universidad Panamericana, Facultad de Ingeniería, Zapopan, Jalisco, 45010, México, arodrig@up.edu.mx; Nicholas Vining, University of British Columbia, Canada, NVIDIA, Canada, nvining@cs.ubc.ca; Dongwook Yoon, University of British Columbia, Canada, DongwookYoon<yoon@cs.ubc.ca>; Alla Sheffer, University of British Columbia, Canada, sheffa@cs.ubc.ca.

0730-0301/2021/12-ART1 $15.00
https://doi.org/10.1145/3478513.3480511

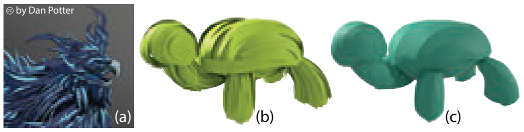


Fig. 2. Representative ribbon brush drawings (a,b); methods such as [Rosales et al. 2019] algorithmically convert ribbon stroke surface drawings (b) into ready to use surface models (c). Bird: © Dan Potter, Turtle: © Elinor Palomares.

Predictable VR Ribbon Brush. *ACM Trans. Graph.* 40, 6, Article 1 (December 2021), 15 pages. https://doi.org/10.1145/3478513.3480511

1 INTRODUCTION

Spatial drawing, using different types of brushes, is an increasingly popular mode of content creation in immersive Virtual Reality (VR) applications (Sec. 2). *Ribbon brushes* [GravitySketch 2019; Keefe et al. 2007, 2001; Mozilla 2021; TiltBrush 2019] enable users to draw 3D shapes using ruled-surface *ribbon* strokes. They define the ribbon geometry by sweeping a fixed-length straight-line *ruling* along the spatial trajectory, or path, traced by the tip of the user's handheld six degrees of freedom (6DOF) controller (Fig. 3). Ribbon brushes provide both experienced and inexperienced users with the means to draw both elaborate 3D artworks and free-form surfaces (Fig. 2). While existing ribbon brush interfaces can be used effectively to draw simple ribbon geometries, they limit the space of ribbons that users can comfortably trace (Fig. 1ab) and can generate unexpected artifacts in response to seemingly natural hand gestures (Fig. 3,bottom). These limitations negatively impact drawing effectiveness and the quality of surfaces that users can produce (Sec. 5). Our *AdaptiBrush* ribbon drawing interface allows users to comfortably draw complex ribbons unsupported by previous interfaces (Fig. 1c-e), while enabling them to better predict the outcome of their gestures, resulting in greater usability (Sec. 5).

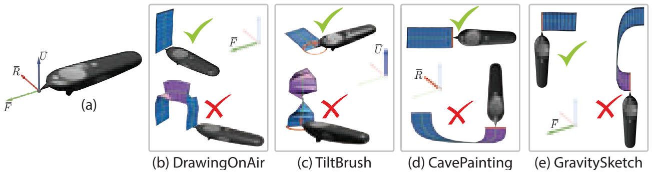


Fig. 3. (a) Canonical VR controller coordinate system. Cross-product based brushes compute the ruling direction as the cross-product of the brush trajectory and a fixed controller axis $\vec{F}$ (Drawing on Air [Keefe et al. 2007], b) and $\vec{U}$ (TiltBrush [2019], c). Direct brushes use a fixed axis $\vec{R}$ (CavePainting [Keefe et al. 2001], d) and $\vec{F}$ (GravitySketch [2019], e) as the ruling direction. (b-e bottom) Moving each brush along a trajectory coinciding with its fixed axis results in unexpected changes in ribbon normal direction and, for direct brushes, ribbon width. Brush signifiers in orange.

Existing ribbon brushes leverage the orientation of a *fixed* controller axis to compute the direction of the ribbon rulings. One family of methods [GravitySketch 2019; Keefe et al. 2001] directly uses the orientation of a specific controller axis as the direction of the rulings (Figs. 1b, 3de), while the other [Keefe et al. 2007; Mozilla 2021; TiltBrush 2019] uses a specific controller axis as approximate ribbon normal and defines the ruling direction as a cross-product of the direction of this fixed axis and the tangent of the controller trajectory, resulting in rulings which are strictly orthogonal to this controller axis (Figs. 1a, 3bc). Changing ruling orientation using the former methods, or explicitly changing the ribbon normal using the latter, requires users to rotate the controller. Unfortunately, humans have a limited ability to rotate their wrists during continuous motion, impacting their ability to orient the controller as they may desire. During continuous motion some controller orientations may be impossible to achieve, since human joints have limited degrees of freedom and ranges of motion; others may be achievable but require uncomfortable wrist-twisting (Sec. 2, Fig 1ab). Consequently, while in theory these brushes may satisfy *controller generality*, or the property that the controller itself offers sufficient degrees of freedom to draw any ruled ribbon, in practice due to biomechanical constraints on human joint motion users can draw only a subset of ribbon geometries using them. Additionally, when the fixed axis these methods use coincides with the trajectory tangent, the *moving frames* of the ribbon (defined by the trajectory tangent, ribbon ruling, and normal) degenerate, resulting in unpredictable changes in ribbons' normal orientation or width (Figs. 3,bottom, 6). Mathematically, the controller can always be rotated around itself to prevent the fixed axis and controller trajectory from coinciding; however, this rotation cannot always be achieved in practice due to biomechanical constraints on human elbow and wrist motions.

We aim to expand the space of ribbons that can be drawn with a 6DOF controller using comfortable arm and wrist motions, a property which we refer to as *biomechanical generality*, while also aiming to avoid unpredictable brush behaviors (Sec. 3.1). We achieve both goals by introducing an *adaptive* brush design that casts the computation of the ruling direction in each time step as a solution of a constrained optimization problem (Sec. 3.2). Our formulation is centered around three key design choices. First, we relax the fixed linkage between controller axes and ribbon ruling orientations, allowing the angles between all controller axes and the rulings to vary. Second, we constrain the ruling directions to be orthogonal to the path trajectory at all times. Third, we use the remaining degree of freedom to maximize brush *predictability* by computing ruling directions that change gradually in both the world and controller coordinate systems, and deviate from controller axes only in response to clear user prompts. Combined, these design choices allow users to change ribbon ruling and normal orientations using predominantly translational motion, greatly extending biomechanical generality compared to prior methods (Fig. 1de). Since rulings are trajectory-orthogonal at all times, a ribbon's moving frame is never allowed to degenerate; consequently, ribbons drawn with AdaptiBrush always maintain the user-prescribed width, and exhibit no unpredictable changes in normal orientation. Promoting predictability and encouraging rulings to *align* with one of the controller axes makes AdaptiBrush easier to use and learn. We solve the resulting per-frame optimization problems in real-time using an efficient parallel solver, avoiding any lag during user interaction (Sec. 3.3).

We confirm the effectiveness of AdaptiBrush via two user studies (Sec 4). The first compares the biomechanical generality of AdaptiBrush against that of prior brushes; it confirms that AdaptiBrush enables users to draw a large range of ribbon geometries of varying complexity that users cannot comfortably trace using prior brushes.

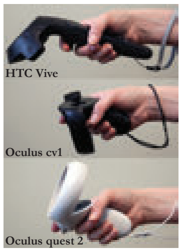
Our second, two-part study compares the user experience of operating AdaptiBrush compared to operating prior brushes. Our first set of participants assessed AdaptiBrush and four other brushes in terms of effectiveness, ease-of-use, output drawing quality, and ease-of-learning. Participants deemed AdaptiBrush as superior in the first three categories, and deemed it on par with the closest competitor in terms of ease-of-learning. We then conducted a head-to-head comparison of AdaptiBrush and the brush the first set of participants deemed as next best. Study participants deemed AdaptiBrush as significantly more effective and easier to use, deemed its output drawings to have superior quality, and overall strongly preferred it over this alternative (58% versus 33.4%; on par 8.6%).

2 RELATED WORK

VR Stroke Drawing Interfaces. Researchers have experimented with a range of techniques and interfaces for directly drawing strokes in 3D space [Amores and Lanier 2017; Diehl et al. 2004; Grossman et al. 2002; Israel et al. 2009; Jackson and Keefe 2016; Kim et al. 2018; Tano et al. 2003; Yu et al. 2021], as surveyed by [Bhattacharjee and Chaudhuri 2020]. These methods typically render the captured strokes as either tubular shapes [Keefe et al. 2007; PaintLab 2019] or ruled ribbons [GravitySketch 2019; Mozilla 2021; Oculus 2019; TiltBrush 2019] that follow a curved trajectory.

Early research [Keefe et al. 2007; Schkolne and Schroeder 1999] suggests that ribbon strokes are effective not only for drawing individual ruled ribbons, but for directly depicting 3D surfaces (Fig. 2). Ribbon strokes visually act as patches of surface and enable human observers to easily envision 3D surfaces depicted using dense collections of ribbons [Keefe et al. 2007] (Fig. 2ab). Dedicated surfacing methods [Huang et al. 2019; Rosales et al. 2019; Schkolne and Schroeder 1999] successfully reconstruct the artist intended surfaces from such ribbon-stroke collections (Fig. 2c). Rosales et al. [2019] suggest that even inexperienced users can model complex 3D shapes by first drawing them using a ribbon brush, and then using these reconstruction methods to obtain their intended model. Our quest for an effective ribbon brush interface is motivated both by the increasing popularity of these brushes, as evidenced by the number of commercial VR systems that support them, and by their potential to simplify and democratize 3D content creation.

Ergonomics and Accuracy of VR Interfaces Current VR controllers are designed for a fixed grip [Oculus 2021] (inset); thus while users can freely translate a controller in space using a combination of shoulder rotation and elbow bending, rotating the controller around itself requires either wrist or elbow joint twisting and is limited by the biomechanical constraints of these joints [Porter and Kaplan 2011]. These constraints impact both the ergonomics and accuracy of VR interfaces.



VR interface research indicates that mid-air interactions lead to upper-arm fatigue and addresses fatigue as a function of shoulder joint torque [Hincapié-Ramos et al. 2014; Jang et al. 2017; LaViola et al. 2017]. Research suggests that overconstrained VR interfaces can cause physical strain [Grossman et al. 2003; Zhai 1998] and that 80% of expert users experienced ergonomic issues such as neck and shoulder pain when using a VR sketching system over an extended period [Arora et al. 2017]. Keefe et al. [2007] observe that care must be taken when designing ribbon-based drawing systems in order to avoid users being forced to move their wrist into uncomfortable positions in order to maintain a correct ribbon orientation. AdaptiBrush drastically reduces the amount of wrist rotation users need to employ when drawing complex ribbons, as compared to prior brushes (Figs. 1, 4, 5, Sec. 4).

Researchers analyzed user accuracy when drawing in space, and concluded that 3D drawing is far less accurate than its 2D counterpart [Arora et al. 2017; Keefe et al. 2007; Rausch et al. 2010]. While user spatial ability influences shape quality [Barrera Machuca et al. 2019], Wiese et al. [2010] demonstrate that a user's VR drawing skills improve rapidly as they gain more experience. Arora and Singh [2021] highlight the ergonomic importance of hand/controller rotation required to draw mid-air strokes. Forman et al. [2020] and Kumar et al. [2020] study the effects of wrist fatigue on hand motion accuracy, and show that sustained or dynamic wrist flexion or extension significantly decrease accuracy. Our user study (Sec. 5) suggests that AdaptiBrush enables users to draw surfaces more accurately than when using other brushes; we speculate that this may partly be due to our novel ruling computation which reduces the amount of wrist flexion and extension users perform during drawing, thus lowering wrist fatigue and improving accuracy.

VR Ribbon Brushes. Many research and commercial VR drawing tools allow users to draw ribbon, or ruled-surface brush strokes by moving a handheld controller in 3D space [GravitySketch 2019; Keefe et al. 2007, 2001; Oculus 2019; TiltBrush 2019]. In all cases, the constructed ribbons form ruled surfaces whose path tracks the trajectory of the tip of the handheld controller in space (Fig. 3), and have rulings whose length is pre-defined by the user. The tools complete the definition of the ribbon geometry by specifying the position and orientation of each ribbon ruling at each point along the trajectory. Ruling directions are by construction signed, with the trajectory tangent, ruling and ribbon normal at each point along the trajectory defining a right-handed *moving frame*. Existing approaches for assigning ruling directions can be divided into two categories: *direct* and *cross-product based.*

Direct Ribbon Brushes use the direction of one of the axes of the controller coordinate system as the ruling direction (Fig. 3de). The GravitySketch [2019] VR brush uses the forward-pointing axis (Fig. 3e), and places the starting point of the ruling at the controller's tip. An alternative choice, motivated by the CavePainting system [Keefe et al. 2001], originally designed for a CAVE immersive virtual reality environment, is to use the rightwards-pointing axis of the coordinate system as the ruling direction and to place the center of the ruling at the controller tip in each frame (Fig. 3d). Changing ruling orientation using direct brushes requires users to rotate the controller around itself, a task that requires rotational wrist or elbow motion and is constrained by the range of motion of these joints. These restrictions make drawing ruled ribbons with naturally changing ruling orientations, such as circles or planar arcs, challenging or even infeasible (Fig. 4).

Cross-Product Based Ribbon Brushes treat one of the controller axes as an approximation of the ribbon normal, and define the

Fig. 4. Direct brushes (a) [Keefe et al. 2001], (b) [GravitySketch 2019] force users to twist their wrists into awkward positions to change ruling orientation, making it difficult for them to draw shapes such as circular arcs for all choices of drawing planes. Biomechanical constraints prevent users from rotating the ruling a full circle. (c) AdaptiBrush users can rotate ruling directions around a fixed axis while keeping their wrist orientation fixed, enabling them to easily draw complex planar ribbons. Columns left to right: HTC Vive, Oculus Quest 2, and Oculus CV1. See supplementary video for complete motions.

ruling direction as the cross product of this axis and the controller trajectory direction (Fig. 3bc). These brushes center the rulings at the tip of the controller at each frame. Drawing On Air [Keefe et al. 2007] uses the forward axis of a tethered virtual pen controller as the *approximate normal* (Fig. 3b). Commercial VR systems such as TiltBrush [2019], Quill [2019], and Mozilla A-Painter [2021] use the up axis (Fig. 3c). Since these three systems have essentially identical implementations, any reference in the text below to one of the three applies to all three. Directly changing the ribbon normal in these interfaces requires rotating the controller (Fig. 5), and is therefore limited by biomecanical constraints on continuous wrist rotation.

For both direct and cross-product based brushes, when the fixed axis used and the trajectory direction become colinear, the default moving frame becomes degenerate. For direct methods, this leads to ribbon width shrinkage (Figs. 3cd, 6a). Since the cross product of colinear vectors is zero, to continue to function cross-product based systems retain the last stable ruling direction when the fixed axis and trajectory become colinear. When the angle between the axis and trajectory direction nears the threshold used for the colinearity test, their ruling computation becomes unstable, resulting in wiggly ribbons (Figs. 3de, 6b). In the presence of moving-frame degeneracies, both families of methods can produce unexpected changes in normal orientation (Figs. 3, 6). Our analysis of user drawings created using these methods (Sec. 5) suggests that users try to avoid such degenerate configurations; for direct methods less than 4% of rulings have an angle of 45° or less with the trajectory tangent, and for TiltBrush [2019] the angle between the up vector (approximate normal) and the tangent is less than 45° in less than 14% of frames.

AdaptiBrush is designed to maximize the space of ribbon geometries users can draw while either keeping the controller orientation fixed or minimally rotating it, and avoids unexpected changes in ribbon width or orientation (Figs. 4c, 5c). It consequently allows users to comfortably and easily draw the ribbons they wish to form, including arbitrarily long and complex ones (Fig. 1e, Sec. 5).

3 ADAPTIBRUSH FRAMEWORK

3.1 Problem Statement

Problem Setup. In a ribbon-based VR drawing system, the user traces strokes in space using a six-degrees-of-freedom (6DOF) handheld VR controller. At each frame, the system captures the controller's position and orientation in three-dimensional space; the sequence of controller positions forms a polyline and defines the trajectory, or path, of the ribbon (Figs. 3, 7). Ribbon drawing systems use the trajectory and the orientation information to specify the location and orientation of the new ribbon ruling in each frame. Each ruling is defined as an oriented straight-line edge connecting start and end vertices; the lengths of all rulings along each ribbon are set to a fixed user specified value. Oriented rulings corresponding to consecutive frames are connected at their start and end vertices using quads to form a discrete ruled surface, or *ribbon* (Fig. 7). While all systems share this common setup, they differ in the way they compute the rulings given the per-frame controller input.

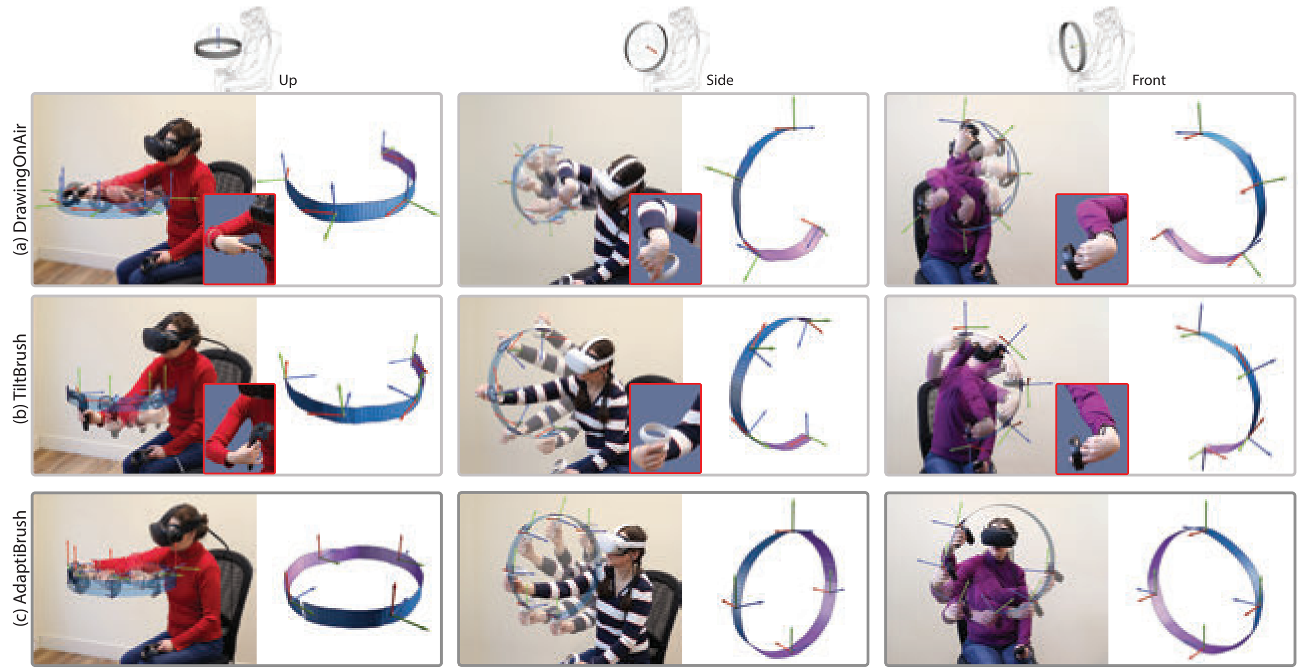


Fig. 5. Cross-product based brushes (a) [Keefe et al. 2007] , (b) [TiltBrush 2019] force users to twist their wrists into awkward positions to smoothly change ribbon normals, making curved ribbons challenging to draw and preventing users from drawing even medium complexity curved ribbons such as cylinders. Using AdaptiBrush, users can draw curved surfaces such as cylinders while keeping their wrist orientation fixed for any choice of cylinder axis (c). Columns left to right: HTC Vive, Oculus Quest 2, and Oculus CV1. See supplementary video for complete motions.

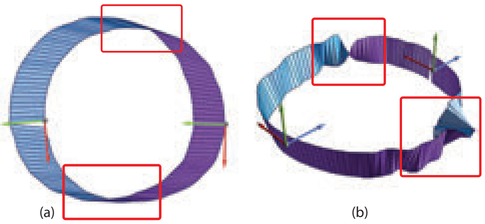


Fig. 6. (a) When the fixed axis and trajectory of a direct brush (e.g. [GravitySketch 2019]) coincide, ribbon width shrinks to zero and ribbon orientation can flip. (b) When the fixed axis and trajectory of a cross-product brush (e.g. [TiltBrush 2019]) are close to colinear, ribbon normals become unstable and can flip.

Objective. The overarching objective of our work is to design a ribbon brush that outperforms prior methods in terms of effectiveness and ease-of-use, while remaining equally easy to learn. We cast these high-level goal in terms of the following technical objectives.

Real-time Interactivity. To facilitate ease-of-use, a ribbon brush must operate in *real time* and at interactive frame rates. To avoid VR fatigue and motion sickness, VR literature [Kelkkanen et al. 2020; Luks and Liarokapis 2019] specifies that tools such as our brush must operate at a minimum of 90 FPS, the target frame rate for consumer display headsets. For a brush to be effective, users must be able to instantaneously see the results of their gestures. Accordingly, our brush must also follow the what-you-see-is-what-you-get principle: once a portion of a ribbon is drawn, it should not be changed by future user input; therefore the geometry of each ruling must be fully determined by the controller input in the corresponding and preceding frames.

Controller-level Generality. For a ribbon brush to be effective, it needs to enable users to draw as many ruled ribbon geometries as possible. We therefore aim for a control scheme that supports all possible planar (zero mean and Gaussian curvature), parabolic (non-zero mean, zero Gaussian curvatures), and hyperbolic (negative Gaussian curvature, zero or non-zero mean curvature) ribbons [do Carmo 2016; Pottmann and Wallner 2009]. Note that as ruled surfaces have zero normal curvature along the ruling direction, by construction they cannot have positive Gaussian curvature.

Biomechanical Generality. Controller-level generality alone is insufficient, since biomechanical constraints on human arm motion prevent users from tracing many combinations of controller locations and orientations. To maximize effectiveness, we need to maximize the space of ribbon geometries that users can draw *subject to biomechanical constraints*. Based on prior observations about human biomechanics (Sec. 2), we interpret this requirement in technical terms as a preference for enabling users to form ribbons using shoulder rotation and elbow bending motions, as opposed to wrist rotation or elbow twist. Wrists have a relatively small range of comfortable motion compared to larger joints [Palmer et al. 1985], and the comfortable range of elbow twist or rotation around the lower arm is limited to approximately 180° [Nan et al. 2019]. Consequently, we aim for an interface that maximizes the space of ribbon shapes that users can draw by translating the controller in space, while keeping the orientation of the controller with respect to its

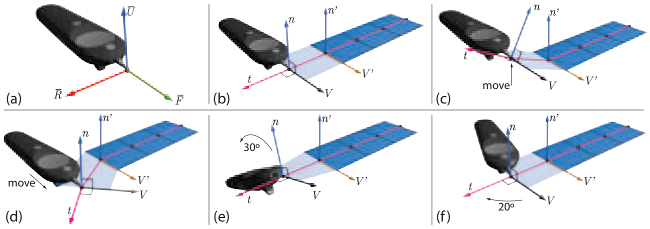


Fig. 7. (a) Controller coordinate frame. (b-f) Example ruling computations in response to controller translation (b-d) and rotation (e,f). While rotation around the trajectory changes the ruling direction (e), due to ruling tangent-orthogonality rotating the controller around the normal has no such impact (f).

own coordinate system fixed. We therefore require a method that enables users to affect a range of ribbon ruling and normal orientation changes via purely translational controller motion. Biomechanical generality, per our definition, is strongly correlated with the physical effort required when utilizing our brush, or its physical ease-of-use. This effort is directly related to the range of motions considerations above; expanding the space of ribbons users can draw without rotating the controller reduces the physical strain required to operate the brush, and thus improves its physical ease-of-use.

Predictability. Maximizing ease-of-use requires minimizing both the physical and mental effort required to operate the brush. We believe that mental effort is directly correlated to ease of control. To effectively control the brush, users must be able to understand the correlation between their hand gestures and the ribbons these gestures would produce; in other words they should be able to anticipate, or predict, the ribbon that their controller motion will create and should be able to anticipate which hand gesture will produce the ribbon they seek to draw. This desire for predictability has three consequences. First, the ribbon's moving frame should never degenerate, as such degeneracies lead to visible and unexpected artifacts. Second, ribbon width should remain constant absent user input indicating otherwise. Finally, and most importantly, the orientation of the rulings, and hence that of the moving frame, should not change without reason. To be predictable, unless the user explicitly indicates otherwise, we aim for the ruling orientation to change as little as possible between frames in both the world and controller coordinate systems.

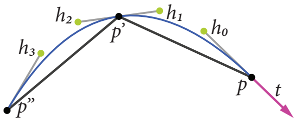
Alignment. Finally, we observe that the VR controller, when held in the user's hand, defines a natural and intuitive coordinate frame (Fig 3a). We speculate that using this frame to define the moving frame of the ribbon makes it easier for the users to control the brush and makes the tool easier to learn. This is confirmed by our studies (Sec. 5), which indicate that users find direct interfaces, where users directly manipulate the ruling directions, easier to learn than cross-product based interfaces, where users manipulate only an approximation of the ribbon normal.

We note that it may not be possible to strictly satisfy all the criteria above at once. For instance, as just observed, satisfying alignment argues for using a direct brush; however, such brushes perform suboptimally in terms of biomechanical generality and predictability. In our setup we prioritize biomechanical generality over predictability and prioritize both over alignment.

3.2 Ruling Computation Algorithm

Following the observations above, to facilitate real-time interactivity and similar to prior methods, we compute ribbon geometry one frame at a time. At each frame, we use as input the current controller location p and orientation expressed via a 3-axis right-handed coordinate frame $\langle \vec{R}, \vec{U}, \vec{F} \rangle$, consisting of *right*, *up* , and *forward* unit vectors respectively (Fig. 7a). In addition to the current controller position and coordinate frame, we use the controller position p', frame $\langle \vec{R}', \vec{U}', \vec{F}' \rangle$ and ruling direction $\vec{V}'$ in the preceding time frame, and its position two frames earlier denoted p'', as input.

Setup. We compute the trajectory tangent t at p using a geometric construction that is motivated by the expectation that users tend to draw stroke paths with constant or gradually changing curvature, consistent with prior research [Baran et al. 2010; McCrae and Singh 2009]. We therefore follow the construction in the figure below, which keeps the rate of change of the tangent orientation between consecutive control points as constant as possible. We consider the current and previous two control points as elements lying on a composite cubic Bezier curve with handle points h_0, h_1, h_2, h_3 (h_3 is not used except to illustrate the construction). In order for this curve to have $C2$ continuity, associated handle points must be mutually opposed and colinear with each control point, and each handle point must be equidistant from the control point. We therefore set



$$\overrightarrow{h_1h_2} = \frac{1}{3}(\overrightarrow{pp''})$$
$$\overrightarrow{h_0h_1} = \frac{1}{3}(\overrightarrow{pp'}).$$

Using this construction $\vec{h}_1 = 1/6(\overrightarrow{p''p}) + \vec{p}'$ and $\vec{h}_0 = 1/3(\overrightarrow{p'p}) + \vec{h}_1$, enabling us to define

$$t = \overrightarrow{h_0p} / ||\overrightarrow{h_0p}||.$$

We formulate the computation of each new ruling direction $\vec{V}$ as a constrained minimization problem (Sec 3.2.1). We leverage the constraints to reduce the solution space to a finite one-dimensional interval, enabling us to compute the minimizer in real-time using brute-force parallel local search (Sec 3.3). We use the computed directions to generate the start and end points r_s, r_e of the new ruling as

$$\begin{aligned} r_s &= p - \vec{V} \cdot w/2 \\ r_e &= p + \vec{V} \cdot w/2 \end{aligned} \quad (1)$$

where w is the user specified ruling length for the current ribbon.

3.2.1 *Formulation.*

At the core of our formulation is the desire to enable users to control ribbon shape through predominantly translational motion. Our key idea making this control mechanism possible is to decouple the strict linkage between ruling and controller orientations; instead, we directly link ribbon ruling and trajectory directions in each time frame. To this end, we constrain the rulings in each frame to be strictly orthogonal to the trajectory tangent:

$$t \cdot \vec{V} = 0$$

Since we search for *unit* direction vectors, we enforce:

$$\|\vec{V}\|^2 = 1$$

With these constraints enforced, changes in ribbon trajectory lead to changes in ruling directions. Enforcing orthonormality has two additional advantages. First, with these constraints in place, the moving frames can never degenerate, improving brush predictability and eliminating the need for users to consciously avoid controller configurations that result in such degeneracies. Second, orthonormality guarantees that our ribbons have widths equal to the input parameter w at all times. Automatically avoiding degeneracies and preserving ribbon width removes the mental burden from the users of having to adjust the controller to achieve these goals.

Enforcing orthogonality and unit length reduces the solution space for our sought after vectors $\vec{V}$ to a one-dimensional space of directions orthogonal to the tangent t. Our choice of optimal direction in this space balances predictability, controller generality and alignment. We express these goals by searching for directions that balance two energy terms. Our first energy term $E_{\text{world}}(\vec{V})$ seeks to satisfy predictability in the world coordinate frame by minimizing the angular difference between the new ruling direction and the previous one, while implicitly accounting for tangent orthogonality. To this end, we note that a direction that is orthogonal to the tangent t and maximally close to the previous ruling direction $\vec{V}'$ can be computed analytically by rotating the previous ruling $\vec{V}'$ using the following formula:

$$\bar{V} = t \times (\vec{V}' \times t) \tag{2}$$

$$\tilde{V} = \begin{cases} \bar{V}, & \text{if } \bar{V} \cdot \vec{V}' > -\bar{V} \cdot \vec{V}' \\ -\bar{V}, & \text{otherwise} \end{cases} \tag{3}$$

We express predictability in the world coordinate frame as keeping the output direction as close to this analytically computed direction as possible.

$$E_{\text{world}}(\vec{V}) = (1 - (\vec{V} \cdot \tilde{V})^2). \tag{4}$$

We constrain V and $\tilde{V}$ to have the same orientation by enforcing

$$\vec{V} \cdot \tilde{V} > 0.$$

Minimizing E_{world} in isolation produces new ruling directions that are orthogonal to the tangent and close to the previous ruling vector. However, this term alone is agnostic to changes in the orientation of the controller, and using it in isolation reduces the number of degrees of freedom users can employ to control the shape of the ribbons they draw. More formally, since V' and $\tilde{V}$ are coplanar, ribbons produced by minimizing $E_{\text{world}}(\vec{V})$ alone will be strictly developable (i.e. isometric to a planar ribbon) and thus will always have zero Gaussian curvature [Pottmann and Wallner 2009]; accordingly, minimizing E_{world} alone would prevent formation of negative Gaussian curvature ribbons.

Our second term, $E_{\text{control}}(\vec{V})$, allows the controller's orientation to affect the ruling orientation, enhancing controller generality. It does so by weakly promoting alignment with cardinal controller axes, while maintaining predictability in the controller coordinate frame.

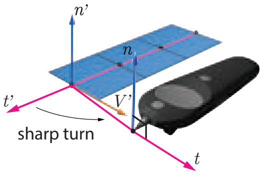
$$E_{\text{control}}(\vec{V}) = \sum_{\vec{D} \in \{\vec{R}, \vec{U}, \vec{F}\}} \left| \vec{D} \cdot \tilde{V} \right| \, (1 - (\vec{V} \cdot \vec{D}')^2). \tag{5}$$

Here $\vec{D}'$ is the vector $\vec{D}$ projected to the plane orthogonal to t and oriented to maximize $\vec{D}' \cdot \tilde{V}$. If an axis and the trajectory are nearly colinear (that is, they have an angle less than $\epsilon = 5°$ between them) we remove this axis from the sum above. This formulation causes the output ruling direction to gradually move toward the closest coordinate frame axis and away from the farthest axis, while implicitly forcing it to remain orthogonal to the trajectory tangent; thus in addition to increasing controller generality this term weakly promotes alignment.

Final Energy. Combining these two terms together, each new ruling direction $\vec{V}$ is computed as the minimizer of:

$$\min_{\vec{V}} \quad E(\vec{V}) = E_{\text{world}}(\vec{V}) + E_{\text{control}}(\vec{V}) \tag{6}$$

Initialization and Trajectory Discontinuities. The formulation above assumes the existence of a prior ruling direction $\vec{V}'$. When users start drawing a ribbon, however, no such prior direction exists. Instead, we initialize the ribbon as follows. Once users activate the brush to draw a ribbon by pressing the trigger on the VR controller, we record the first two points along the trajectory p' and p and use the direction of $\overrightarrow{p'p}$ as the trajectory tangent t. We compute the ruling at p as follows: if the angle between t and $\vec{U}$ is larger than 45°, we set $\vec{V} = t \times \vec{U}$; otherwise, we set $\vec{V} = t \times \vec{F}$. This choice promotes rulings that are orthogonal to the up vector given approximately horizontal user motion, and ones orthogonal to the forward direction given weakly vertical motion; this choice facilitates maximal alignment between rulings and controller axes subject to tangent orthogonality.



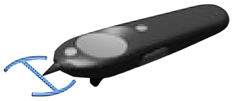
When drawing ribbons, users may sharply change their trajectory of motion, leading the tangent t to practically coincide with the previous ruling direction $\vec{V}'$. Such a sharp change of trajectory represents a deliberate choice on the part of the user, and indicates an intentional discontinuity in the shape of the output ribbon; in this case, similarity between the current and previous ruling directions is likely undesirable and should not be attempted. Thus, when the angle between t and $\vec{V}'$ is less than ε we set $\vec{V}' = t'$, where t' is the previous frame's tangent.

3.3 Solving For A New Ruling

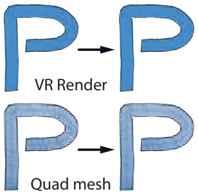
Finding a new ruling requires us to solve a non-linear constrained optimization problem every frame. We solve it efficiently using a brute-force discretized approach that can be trivially parallelized. We observe that each new ruling must be a unit vector lying on the plane orthogonal to the tangent. Consequently, we can recast our problem as finding an angle θ such that the vector $V(\theta)$ formed by rotating $\tilde{V}$ around t by θ minimizes $E(V(\theta))$. We find the optimal θ by brute-force searching the range $-90° \le \theta \le 90°$, employing a search step of 0.04° and evaluating candidates in parallel; at this sampling density we achieve equivalent accuracy to a commercial solver [Gurobi Optimization 2020] with a tolerance of 10^{-6}; see Sec. 5 for timing information.

3.4 Brush Visualization and Implementation

Brush Visualization. Inspired by the signifiers used by commercial VR brushes (Fig. 3), [GravitySketch 2019; TiltBrush 2019], we improve the usability of our brush by using a *signifier* that communicates the current moving frame to the user. We shape our signifier as two circular arcs bounding a common diameter line, centered at the controller tip (inset). The diameter line depicts the current ribbon ruling, and the circular arcs bounding the diameter lie in the plane orthogonal to the current ribbon normal. Since both normal and ruling directions in our system change smoothly and gradually, visualizing the moving frame helps users ideate how their next movement will affect their ribbon. When the brush is not engaged, the signifier is aligned with the right controller axis $\vec{R}$, and the arcs lie on the plane whose normal is the up axis $\vec{U}$. This initialization promotes alignment and is consistent with our ribbon initialization process that preferrentiates horizontal rulings.



Quad-Mesh Fairing. We note that the trajectory tangents used while the user draws the ribbon are an estimation of the final trajectory tangent after the ribbon is complete, and are impacted by local trajectory inaccuracies. Once the ribbon is complete and the user has released the trigger, we fair the generated quad mesh to better align the rulings with the final less noisy tangents, while preserving the moving frame normals. Specifically, we rotate each mid-ribbon ruling, keeping its center point p fixed around the moving frame normal, to make it as orthogonal as possible to the final tangent. This step improves the alignment between quad and moving frame normals; while the visible effect on VR ribbon rendering is limited (inset, top), this correction (inset,bottom, trajectory in green) makes ribbons more suitable for downstream post-processing with methods such as that of Rosales et al. [2019].



Technical Implementation. We implemented AdaptiBrush as a Unity application, using the OpenVR SDK and the SteamVR Unity plugin [Unity 2019]. The user interacts with AdaptiBrush using two controllers, one in the dominant hand and one in the non-dominant hand. The dominant hand performs drawing actions; the non-dominant hand controls additional user interface options, such as undo and redo functionality, mimicking the TiltBrush interface [TiltBrush 2019]. We also provide users with a choice between "draw" and "erase" modes. To form the ribbon, we sample the dominant hand controller positions when the "draw" trigger is engaged, generating new rulings, each time the controller moves ε-distance away from the last ribbon endpoint; we set $\varepsilon = 1/5(w)$ to empirically match observed behavior in other packages.

4 COMPARATIVE EVALUATION

We validate the effectiveness of AdaptiBrush against that of prior spatial brushes via two comparative studies (Sec. 4.1 and 4.2). The first study evaluates the biomechanical generality of different brushes by asking participants to replicate differently shaped exemplar ruled-surface ribbons using a minimal number of strokes (Fig. 8). This study confirms that AdaptiBrush enables users to trace a larger range of ribbon geometries using single continuous strokes than previous methods, confirming its superiority in terms of biomechanically generality. User feedback collected during this study suggests users view AdaptiBrush as more effective and easier to use than existing brushes. The second study compares the effectiveness and usability of the different brushes “in-the-wild”: users were asked to use each brush to depict different 3D shapes, but were given no instruction as to the properties of ribbon strokes to use (Figs. 9, 10). The study consisted of two parts: the first compared AdaptiBrush to all four prior brushes (Sec 4.2.1); the second performed a more in depth comparison between AdaptiBrush and the next best brush as identified by the first study (Sec. 4.2.2). The results of the combined two-part study confirmed that AdaptiBrush dominates the other brushes in terms of effectiveness, ease-of-use, and perceived output quality. We first describe the experimental setting of the studies, then elaborate the format and outcomes of each study; see Appendix A for additional details.

Experiment Design. We used within-subject design. Each participant was asked to draw the same set of strokes or surfaces using all brushes given in each study, and was asked the same series of questions about their experience and outputs. In all studies to minimize tool order bias, we choose an initial random order for the tools and then shifted the order by one for each new participant. To control order effects, the order of shapes in each study was randomized.

Data Collection. After each drawing task, participants rated their subjective experience drawing the target shapes using a given tool. The metrics used included effectiveness, ease-of-use, ease-of-learning, output quality, and brush preference, rated using a 5-point Likert scale (“Strongly Agree”, “Agree”, “Neutral”, “Disagree”, “Strongly Disagree”). Each study used a set of metrics that reflected its objective. Further details of each study’s data collection procedure can be found in Appendix A.2. All participants answered a pre-task demographic survey, with the result of each survey summarized in the corresponding subsection of Appendix A.1.

Participants. Each study used a distinct set of participants. For the biomechanical generality assessment (Sec. 4.1) we recruited experienced users of VR drawing tools to minimize learning effects. For the effectiveness evaluation, to enable fair comparison, we sought out users with minimal VR brush experience and diverse expertise. Due to COVID-19 pandemic social distancing orders, the majority of participants completed their studies remotely using their own VR sets, while two did the study in Sec. 4.2.1 in person; see Appendix A.1 for detailed participant demographics for each study.

Procedure. Each remote participant received an executable containing the assessed brushes and study setup and performed the study independently, with one of the researchers available on call to answer general setup questions. In person participants used a pre-installed executable, with the rest of the protocol being identical. At the beginning of each study, participants were introduced to the study structure and the spatial navigation interface in a video played inside the VR environment and then were asked to spend some time to familiarize themselves with the interface. For each brush, we asked participants to spend at least three minutes drawing the four practice strokes shown in the inset. We designed each of the studies to be doable in 1-2 hours, accounting for setup time, time to draw each shape (approximately 3 min.), and time to answer each question in a VR setup. When answering questions about the shapes they drew during the studies, participants could move, rotate, and zoom in on the 3D shapes dispalyed to examine and compare their drawings. In all questions and instructions we referred to the brushes as "brush" A, B, C, D, E according to their order of appearance for each participant. For participant safety, they were asked to remain seated during the study.

Implementation. To provide fair conditions for comparisons against prior art, and focus the comparison on the brush interface differences, rather than polish levels, we implemented all assessed brushes using the same Unity plugin interface that was used for AdaptiBrush (Sec. 3.4), providing the same undo, ribbon width choice, manipulation and other functionalities. When reimplementing the commercial brushes (e.g. [GravitySketch 2019; TiltBrush 2019]) we accurately reproduced their observed behaviour, while for brushes for which no public implementation exists [Keefe et al. 2007, 2001] we followed the description in the respective papers.

4.1 Biomechanical Generality Evaluation

The objective of our first study was to evaluate the biomechanical generality of the different brushes by assessing the space of ruled surface ribbons that can be traced with single strokes. To this end, we evaluated how well users can reproduce different example ribbon geometries using a single stroke, or a minimal number of strokes if they fail to do so using just one. Study participants were specifically asked to trace each given ruled surface using the smallest possible number of separate strokes possible, and ideally a single stroke when possible.

Study Design and Procedure. The study used a 5 (Tools) by 4 (Ribbons) within-subject factorial design. Participants were asked to trace all four ribbons with one tool at a time. The tools were AdaptiBrush and four prior ribbon brushes: CavePainting [Keefe et al. 2001], GravitySketch [2019], DrawingOnAir [Keefe et al. 2007], and TiltBrush [2019]; the latter is identical to the brushes used by [Oculus 2019] and [Mozilla 2021]. The target ribbon set includes a planar spiral (zero Gaussian and mean curvatures), a cylindrical spiral (zero Gaussian and positive mean curvature), a stroke consisting of planar sections connected via smooth corners (a mixture of positive, zero, and negative Gaussian curvatures), and a twisted ribbon (negative Gaussian curvature) (Fig. 8). In selecting the ribbons for users to draw, we aimed to balance generality against simplicity. We selected ribbons that are representative of the spectrum of ruled ribbon geometries, are easily recognizable and thus easy to ideate, and that users can draw within a reasonable time frame using tools they have not used before.

For each tool we rendered the four target ribbons side by side using semi-opaque material, and instructed the participant to "Draw the given shapes on top of each transparent scaffold, defining the surface as accurately as possible. Try your best to draw each shape with a single brush stroke."

Our core evaluation metric was the number of strokes users employed for drawing each ribbon. We also collected participants' subjective ratings of the effectiveness, ease-of-use, and ease-of-learning, of the brushes, and their impression of brush output quality.

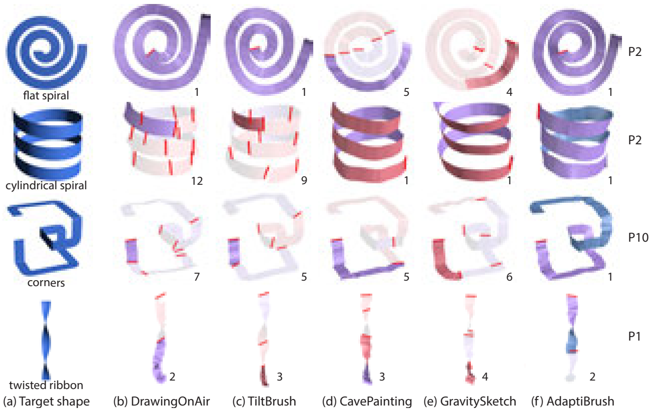


Fig. 8. Representative drawings from the ribbon-tracing study. First stroke of each drawing is opaque to indicate how far the participant could proceed before stopping due to biomechanical constraints. Red lines indicate the first ruling of each new ribbon. Total number of strokes used for each target ribbon is shown at the bottom right of each image. On average participants required significantly fewer ribbons (often just one) to draw the target shapes with AdaptiBrush than with other tools.

	DrawingOnAir	TiltBrush	CavePainting	GravitySketch	AdaptiBrush
Flat Spiral	1.2/1 (0.6)	1/1 (0)	4.7/5 (1.5)	6.8/6.5 (2.5)	1/1 (0)
Cylindrical Spiral	5.8/6 (3)	6/5.5 (2.5)	1.6/1 (1.8)	1/1 (0.3)	1/1 (0)
Corners	7.8/7 (2.4)	7.6/6.5 (3.4)	6.8/6 (3.2)	8.9/7.5 (3.5)	1.5/1 (0.7)
Twisted Ribbon	4.4/3 (3.4)	2.7/3 (1)	2.7/2 (1.35)	3.4/3 (0.8)	3.2/3 (1.8)
Total	4.8/3.5 (3.5)	4.35/3 (3.4)	3.95/3 (2.9)	5/4 (3.7)	1.7/1 (1.3)

Table 1. Number of strokes per target ribbon in the ribbon-tracing study (average/median, and standard deviation in brackets). Overall, AdaptiBrush requires significantly less strokes to reproduce target ruled-surfaces than all prior methods ($p < 0.05$, compared across all ribbons and users per-brush).

	DrawingOnAir	TiltBrush	CavePainting	GravitySketch	AdaptiBrush
Ease of Use	2.4/2.5* (0.9)	3.3/3.5 (0.8)	3.3/3 (0.9)	2.1/2* (0.5)	**4.1**/4 (0.5)
Ease of Learning	3.3/3* (0.6)	4/4 (0.6)	**4.3**/4 (0.5)	4.2/4.5 (1)	**4.3**/4.5 (0.9)
Output Quality	2.7/3* (0.8)	3.2/3 (0.8)	3.3/3.5 (0.8)	2.3/2* (0.8)	**3.7**/4 (0.6)
Effectiveness	2.3/2* (0.6)	2.9/3* (0.5)	3/3* (0.8)	2.5/2* (0.7)	**4.2**/4 (0.4)

Table 2. Subjective ratings from the ribbon-tracing study (average/median and standard deviation in brackets). Highest average per row highlighted in bold. Entries marked "*" in the first four columns indicate statistically significant difference ($p < 0.05$) in comparison against AdaptiBrush.

Results and Findings. A total of 10 participants successfully completed all the tasks. On average participants took 94 minutes to compete the study. Tabs. 1 and 2 summarize the study findings, and Fig. 8 shows representative examples of ribbons users traced using different tools. Complete details are included in the supplementary.

Our results convincingly demonstrate that using AdaptiBrush increases the space of stroke geometries users can comfortably draw; our ANOVA measurement shows a significant brush type effect on stroke count ($F_{4,36}$=11.8, $p \leq 0.05$). While users required an average of 1.7 strokes to draw the example geometries using AdaptiBrush, this number rose to 3.95 using the next best performing brush. Notably most users required only one stroke to draw three of the four representative ribbon geometries using AdaptiBrush. For each of the other methods two of the shapes required most users to employ at least five strokes each. The twisted ribbon represents a particularly challenging case for all methods as its rulings need to perform a full rotation around its path. Eight AdaptiBrush users used three strokes or less to depict this shape; the only brush where most users did better was CavePainting, where over half the users used two ribbons or less. This difference was not statistically significant.

Qualitative assessment collected after the users completed the four drawings using each tool suggests that our increased generality led users to assess AdaptiBrush as more effective and easier to use than all alternatives, and to judge the outputs generated using it as having better quality. They judged AdaptiBrush as on par with other brushes in terms of ease-of-learning. Data shows significance on subjective ratings ($F_{16,129} = 4.0164$, $p \leq 0.05$, $\Lambda = 0.29$ on a repeated-measures ANOVA). t-tests conducted on the study results show that AdaptiBrush offers a statistically significant effectiveness improvement against all other methods, and statistically significant improvement in terms of ease-of-use an output quality against DrawingOnAir and GravitySketch ($t = 60$, $p \leq 0.05$; two-tailed paired t-tests).

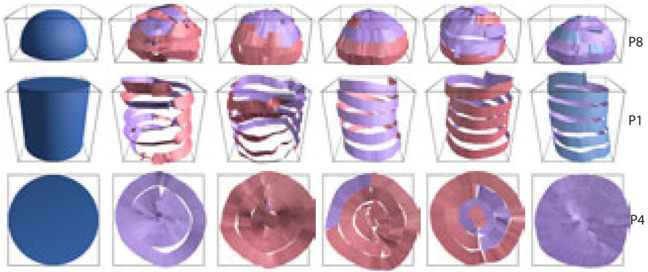
4.2 Effectiveness Evaluation: Drawing 3D Shapes

Our second study aims to assess the effectiveness of AdaptiBrush in a typical real-world VR drawing setup, where users draw 3D content using free-hand strokes. Study participants were instructed to draw a range of shapes as accurately as possible without restricting them to use a specific pattern for their individual ribbon strokes. The core evaluation metrics were participants' subjective ratings, including tool effectiveness, ease-of-use, ease-of-learning, and perceived quality of the shapes they drew. Directly comparing our brush against the four alternative brushes on even a single shape requires over 30 minutes of participant time; thus assessing the tools on four shapes or more would require over two hours. Since we aim for most participants to spend under two hours completing our studies, we opted for a two-part study format. In part one (Sec. 4.2.1), we compared AdaptiBrush against all four prior brushes on three representative shapes. In part two (Sec. 4.2.2), we performed a head-to-head comparison between AdaptiBrush and the best performing prior brush from the part one study, using eight shapes.

4.2.1 Multi-Brush Side-by-side Evaluation.

The first part of our study compares AdaptiBrush against the four methods in Sec. 4.1.

Study Design and Procedure. The study used a 5 (Brushes) by 3 (Shapes) within-subject factorial design. We asked participants to use each brush to draw 3 different shapes, one after the other, as accurately as possible. The shapes we selected for participants to draw were a circular disk (planar), a cylindrical surface (no caps, singly curved), and a hemisphere (no cap, doubly curved). This selection was motivated by generality and simplicity considerations. Our selected shapes are representative of commonly used surface-patch geometries, are easily recognizable, and can be drawn even by non-experts within a reasonable time frame using a tool they have not used before.



(a) Target shape (b) DrawingOnAir (c) TiltBrush (d) CavePainting (e) GravitySketch (f) AdaptiBrush

Fig. 9. Target shapes, and representative drawings from the multi-brush side-by-side evaluation.

	DrawingOnAir	TiltBrush	CavePainting	GravitySketch	AdaptiBrush
Ease of Use	2.1/2* (1.1)	2.8/3* (1.2)	3/3 (1.2)	3/3 (1)	**3.3**/4 (1)
Ease of Learning	2.45/2* (1.1)	3.65/4 (0.9)	3.95/4 (0.7)	**4.2**/4 (0.8)	4/4 (0.7)
Output Quality	1.9/1.5* (1.2)	2.3/3* (1.4)	3.25/3 (1.2)	3.3/3 (1.4)	**3.55**/4 (1.1)
Effectiveness	2/2* (1.2)	2.8/3* (1.3)	2.9/3* (1.25)	3.1/3 (1)	**3.3**/4 (1)

Table 3. Multi-brush performance evaluation summary (average/median, and standard deviation), highest average per row highlighted in bold. Entries marked "*" in the first four columns indicate statistically significant difference ($p < 0.05$) in comparison against AdaptiBrush. Study participants scored AdaptiBrush as more effective and easier to use than prior methods, and deemed surfaces produced using AdaptiBrush to have superior quality compared to those produced by alternatives.

Participants were introduced to one tool at a time and asked to use the given tool to draw one target shape at a time. The target shape was displayed enclosed in a bounding box, and an empty same-size bounding box was displayed to the right of it. We instructed participants to "draw the same shape in the right box, so the resultant shape is similar to the target model as far as you can."

Study Findings. 20 participants successfully completed all shape drawing tasks. See Appendix A.1 for their demographic information. On average participants took 1.5 hours to complete the study, as expected. Tab. 3 summarizes the study findings and Fig. 9 shows representative examples of ribbons users traced using different tools; complete details are included in the supplementary.

Overall, AdaptiBrush was rated as more effective and easy to use than the other four brushes ($F_{12,749} = 5.61$, $p \leq 0.05$, $\Lambda = 0.79$ on a repeated-measures ANOVA), and ranked approximately on par with the others in terms of ease of learning ($F_{4,76} = 14.3$, $p \leq 0.05$ on ANOVA measurement). t-tests conducted on the study results show that AdaptiBrush offers a statistically significant improvement across all metrics versus DrawingOnAir; and on effectiveness, ease-of-use, and quality vs TiltBrush ($t = 60$, $p \leq 0.05$; two-tailed paired t-tests).

Overall participants rated the quality of output shapes they drew with AdaptiBrush to be higher than the quality of the outputs they produced with the other brushes; in this study, this improvement was shown to be statistically significant vs both DrawingOnAir and TiltBrush ($p < 0.05$). We base our output quality measure solely on participant perception rather than on Euclidean space distance between the target and drawn shapes, as Euclidean space distances are heavily dependent on viewer perception of the target shapes. For instance, some participants added top and bottom caps to the

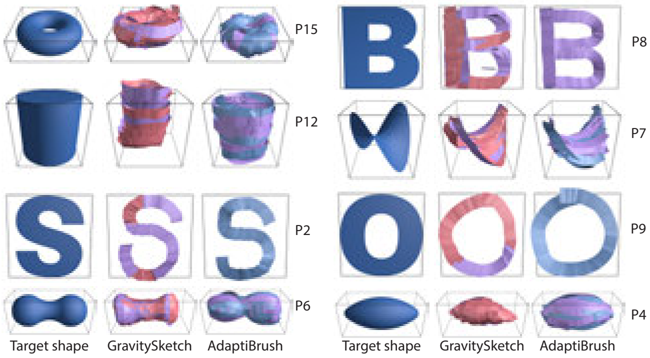


Fig. 10. Representative user drawings from the head-to-head comparison of AdaptiBrush and GravitySketch.

cylinders they drew, even though the target shapes did not contain such caps. Such perception-based discrepancies make Euclidean space distance metrics inadequate for our needs. In contrast, the subjective rating is a direct measure of how well our participants succeeded in drawing the shape they aimed to draw.

4.2.2 *Head-to-Head Comparison Against GravitySketch.*

The second part of our effectiveness study directly compares AdaptiBrush against the GravitySketch [2019] approach, which was ranked as second best in the multi-tool study. This format allows us to perform a more in depth comparison of these tools, in terms of both the number of example inputs, and the opportunity to ask head to head comparative questions immediately after participants use both tools (and thus can base their answers on just-completed tasks).

Study Design and Procedure. The study used a 2 (Brushes) by 8 (Shapes) within-subject factorial design. Participants were asked to draw eight shapes (Fig 10), first with one tool and then with the other. Half participants assigned at random used GravitySketch first, the other half used AdaptiBrush first. Shape order was randomized for the first participant, and rotated moving forwards. The shapes included the letters “B” (planar, mix of straight and positive curvature paths), “O” (planar, full-circle positive curvature path), and “S” (planar, path curvature changes sign), a cylinder (with caps, singly curved), an ellipsoid (doubly curved, positive Gaussian curvature), a hyperbolic paraboloid (doubly curved, negative Gaussian curvature), a torus (doubly curved, both positive and negative Gaussian curvatures), and a metaball (doubly curved, both positive and negative Gaussian curvatures). These shapes cover the full spectrum of curvature types, are easy to recognize, and can be drawn by a non-expert within a reasonable amount of time.

Participants were shown one target shape at a time, enclosed in a bounding box, and an empty same-size bounding box to the right of it. Participants were instructed to “Draw the same shape in the right box, so the resultant shape is similar to the target model as far as you can”. After drawing each shape with both tools they were asked to assess the *relative* ease-of-use, output quality, and effectiveness of each tool; and to indicate their tool preference for that shape. The answer options had the general form of: ‘definitely tool A’, ‘probably tool A’, ‘both are equal’, ‘probably tool B’, ‘definitely tool B’; see Appendix A.2 for exact question and answer wording.

	Definitely GravitySketch	Probably GravitySketch	Both are equal	Probably AdaptiBrush	Definitely AdaptiBrush
Ease of Use	23 (17.9%)	20 (15.6%)	9 (7.0%)	41 (32.0%)	35 (27.3%)
Output Quality	17 (13.3%)	22 (17.2%)	9 (7.0%)	43 (33.6%)	37 (28.9%)
Effectiveness	22 (17.2%)	16 (12.5%)	14 (11.0%)	41 (32.0%)	35 (27.3%)
Preference	24 (18.6%)	19 (14.8%)	11 (8.6%)	33 (25.8%)	41 (32.0%)

Table 4. Head-to-head study comparisons summary across all participants and drawings. Each entry includes both number and percentage of responses. Participants preferred AdaptiBrush across all assessed modalities, with strong statistical significance ($p < 0.01$).

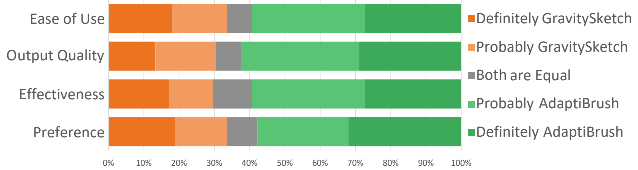


Fig. 11. Bar-chart visualization of the findings in the head-to-head comparison of GravitySketch and AdaptiBrush.

Study Findings. The study included 16 distinct participants. On average, participants took 2.4 hours to complete this study. Table 4 and Fig. 11 summarize the study findings, and Fig. 10 shows representative examples of ribbons users traced using the two tools; complete details are included in the supplementary.

The subjective ratings (Tab. 4, Fig. 11) show that AdaptiBrush is preferred over GravitySketch **across all metrics** with strong statistical significance ($p \leq 0.01$). Participants saw AdaptiBrush as easier to use and more effective than GravitySketch, and judged the outputs produced with AdaptiBrush to be higher quality. Overall, AdaptiBrush was “definitely” preferred 32% of the time and “probably” preferred 26% of the time across all input shapes and participants; the two tools were judged as on-par 8.6% of the time. These findings confirm that AdaptiBrush provides a significant improvement over all prior brushes in terms of effectiveness, ease of use and output quality.

4.2.3 *Qualitative Feedback.*

In both parts of the effectiveness study, participants were invited to provide optional qualitative feedback on the brushes used, and to identify what they liked or disliked about each brush. Below we provide some representative participant quotes. For complete feedback summary please see supplementary material.

Cross Product-Based Methods. Positive comments about DrawingOnAir included: *“Good for drawing on a plane but because it is fixed it makes it difficult to make some movements fluidly”* (P4) *“It is a good tool for 2d [shapes] that simulate a tape.”* (P16), *“Clean width”* (P17). Negative ones included *“more complicated than I expected”* (P20), *“difficult to learn and use”* (P1), *“impossible to predict the direction”* (P3), *“wobble effect when going in circle”* (P5), *“the moves of the brush are so random”* (P11),

For TiltBrush positive comments included *“Great for planar shapes. The flat circle tip really helped guide flat shape drawing.”* (P7), *“width [preservation]”* (P17). On the negative side participants commented: *“It is necessary to keep the tool perfectly perpendicular to the trace which is not always easy to achieve.”* (P3) *“Hopeless for cylindrical shapes ... I gave up on rings entirely and ended up making it out of vertical planks instead.”* (P7), *“you have to rotate your arm and shoulder too much”* (P6), *“hard to make domes”* (P13).

These comments suggest that biomechanical generality is strongly correlated with user perceived effectiveness and ease of use, and that while the users found these tools comfortable for planar shapes where they can comfortably use longer ribbons which follow the shape without triggering moving-frame degeneracies, they found them challenging on curved surfaces where this is no longer the case. The comments also highlight the importance of predictability for both usability and ease of learning; and point to the importance of preserving constant ribbon width, as enforced by both these tools and AdaptiBrush.

Direct Methods. For CavePainting positive comments included: *"Straight forward"* (P5), *"fine for drawing cylinders"* (P12), *"Intuitive"* (P7), *"effective with the dome"* (P1). Negative comments included *"poor results for the circle"* (P1), *"Terrible for planar curves"* (P7), *"Hard to make curves"* (P17).

For GravitySketch positive comments (part 1) included *"simple and easy to control"* (P6), *"great for some specific stroke types"* (P15), *"like a roller taken by one end"* (P16), *"most noble tool to work with and easy to learn"* (P19) . Positive comments (part 2) included: *"basic and intuitive tool, anyone who has drawn digitally before fits without problems."* (P4), *"It's really easy to predict."* (P9). Negative comments (part 1) included: *"It needed twists and multiple hand movements to recreate certain surfaces."* (P2), *"difficult and tricky with the circle"* (P1),*"have to turn your shoulder and arm a lot"* (P6), *"difficult to use for curved objects given the wrist turns that must be made to follow curves"* (P16). In part 2: *"The turns became physically uncomfortable."* (P9, referring to drawing letters) , *"Depends on how much you rotate your wrist."* (P13), *"You have to move a lot and make more moves or do more brushes to reach the result wanted"* (P15).

These observations are consistent with our analysis that these tools require significant wrist twisting to draw shapes with naturally changing ruling orientations, and that this biomechanical constraint impacts both physical ease of use and generality. They also are consistent with our conjecture that axis alignment, strictly enforced by direct tools, makes tools easier to learn.

AdaptiBrush. Representative positive feedback included (part 1): *"easy to learn and use and effective with all the figures"* (P1), *"Easy for drawing in every direction"* (P3), *"My favorite...fluid and predictable behavior"* (P4), *"Really good movement and turn on the strokes path this is my favorite"* (P14), *"Good overall brush tip. orientation gizmo is predictable, I could get a good stroke on all but the most awkward planes and adjustments. ."* (P15). In part 2 participants commented: *"Curves were much easier to make . I [didn't] have to move the wrist as the tool would detect my movement."* (P1), *"Feels easier for curved movements."* (P3), *"It feels easier to draw circular designs and give the illusion of 3D. With practice one can sense the orientation of the stroke and it becomes an intelligent tool that allows more flexibility."* (P7).

Negative comments (part 1) included: *"Very sensitive"* (P6), *"tricky to figure out at first"* (P7), *"hard to learn"* (P9). In part two, these included: *"If swipes are made slow it would jitter"* (P2), *"could only be rotated automatically"* (P4), *"Some times it turns in an angle I don't want and make kinda difficult to correct between shapes"* (P9).

These comments suggest high overall satisfaction, but indicate that our brush may require more time to learn to control. A few comments also suggest that our method may benefit from reduced sensitivity. It is not clear if either concern is widely shared.

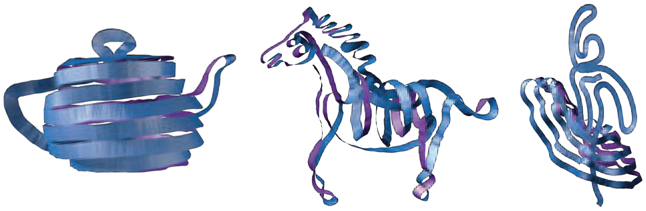


Fig. 12. Single stroke drawings created by artists using AdaptiBrush. Teapot and horse: © Jafet Rodriguez, flamenco dancer: © Elinor Palomares.

5 ADAPTIBRUSH RESULTS

We further evaluate AdaptiBrush via empirical assessment and ablation. We empirically assess the biomechanical generality of AdaptiBrush, by commissioning two artists with VR experience to create 3D art using it and encouraging them to create complex drawings with a minimal number of strokes and to use the full range of stroke geometries that they can draw. Figures 1e, 12 and 13 show the 3D drawings they created. Notably all drawings in Fig. 12 were created with single strokes. Such results are impossible to create with prior methods, and demonstrate the wide range of geometries supported by AdaptiBrush. The surface drawings in Fig 13 are not only visually appealing, but can be used as input to a modeling tool designed to convert ribbon drawings into manifold surfaces [Rosales et al. 2019]; the resulting models can be 3D printed or used as virtual assets.

We validate our hypothesis that users find it more intuitive and thus easier to learn and use brushes whose rulings satisfy alignment by measuring the angles between the rulings of the ribbons they draw using AdaptiBrush and controller axes in the corresponding frames. The median angle between the output rulings and closest axis across all user inputs was 12°, and the median angle between the rulings and farthest axis was 88°. These numbers suggest that users leverage the opportunity provided by our method to align the ribbon normals and rulings with controller axes, validating our design choice to promote alignment. At the same time, the flexibility provided by our adaptive linkage allows for the ribbon moving frames to diverge from the controller frames, facilitating user ability to comfortably draw complex ribbons.

Runtimes. Our method takes 0.018 milliseconds on average to compute a new ruling direction, measured on an AMD Ryzen 7 1800X running at 3.6 GHz with 32 GB of RAM, supporting a far higher frame rate than the required 90 FPS [Kelkkanen et al. 2020; Luks and Liarokapis 2019].

Formulation Ablation. We ablate our choice of energy to optimize for (Sec 3.2, Eq. 6) by assessing the impact of dropping either of the two terms it consists of (Fig. 14). Disabling the control term $E_{control}$ (Eq. 5) results in ribbons whose geometry is dependent only on the controller trajectory and limits the space of output ribbons to zero-curvature, or developable ones, severely restricting controller level brush generality (Fig. 14a). In contrast, removing the continuity term E_{world} (Eq. 4) results in more jaggy and thus less predictable ribbons, (Fig. 14b). These experiments validate the

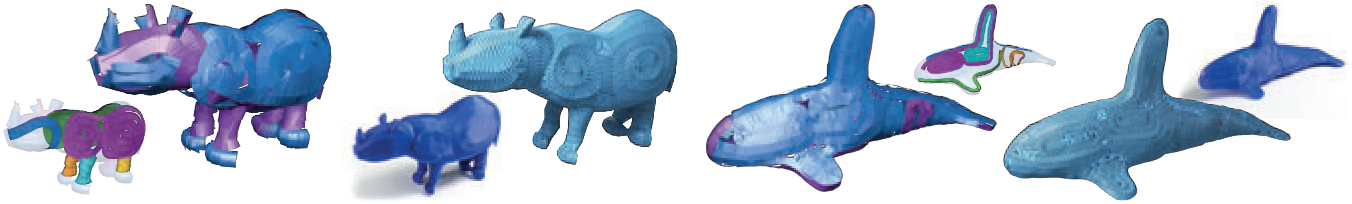


Fig. 13. 3D models created from AdaptiBrush drawings using the method of Rosales et al.[2019]. For each input, drawing shown on the left and surface on the right. Insets show complex strokes and 3D printed models. Rhino and Orca: © Elinor Palomares.

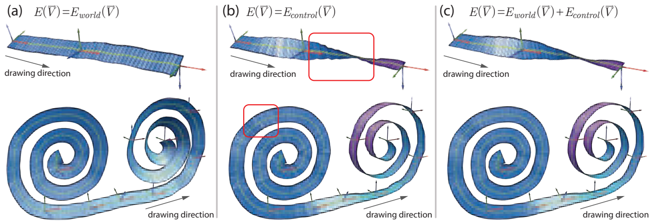


Fig. 14. Formulation ablation: (a) Minimizing only E_{world} prevents users from drawing non-developable surfaces, and disables the impact of controller rotation on the output shape. (b) using the same sequence of controller frames and minimizing only E_{control} results in jaggy ribbons with sudden changes in ruling directions; (c) minimizing our full energy $E(V)$ and using the same sequence of controller frames produces smooth, predictable ribbons with both developable and negative Gaussian curvature regions, facilitating drawing of twisted ribbons (top) and transition between differently oriented ribbon sections (bottom).

need for our formulation which accounts for both generality and predictability (Fig. 14c).

5.1 Limitations and Future Work.

As demonstrated, AdaptiBrush is more biomechanically general than prior brushes and does not exhibit the unexpected artifacts due to degenerate moving frames that made prior methods less predictable. At the same time, the range of ribbons users can draw with AdaptiBrush is still constrained by human biomechanics, as demonstrated by the twisted ribbon example in 4.1, which necessitates multiple strokes to depict with any existing brush. While the exact range depends on the flexibility of user joints, we expect ribbons with large regions of high negative Gaussian curvature to be hard to draw using either AdaptiBrush or other existing brushes. However, we expect users to rarely need such ribbons when drawing surfaces, as research shows that most surfaces can be well approximated by developable (zero curvature) ribbons [Pottmann and Wallner 2009]. Given the limitations of human joint motion, it is unclear whether any gesture driven brush utilizing off-the-shelf commercial controllers can be fully biomechanically general. We speculate that as human hands have < 6DOF during continuous motion, the space of single continuous ribbons humans can draw using any brush is limited.

Two of thirty-six participants made negative comments about AdaptiBrush's learning curve when specifically asked to list negative comments for each brush. Our t-tests (Tables 2, 3) comparing user assessments of ease-of-learning across all tools show that AdaptiBrush is significantly easier to learn than DrawingOnAir and show no statistically significant differences between it and the other tools in terms of ease-of-learning. It would be interesting to develop drawing tutorials that assist users in learning to operate it effectively.

6 CONCLUSIONS

We presented AdaptiBrush, a new ribbon brush that outperforms prior brushes in terms of effectiveness and ease of use. The first key observation behind our brush design is that user ability to rotate a controller around itself, while simultaneously translating it along a desired spatial path, is limited by human joint biomechanics. We achieve much higher biomechanical generality than prior brushes by enabling users to draw a larger range of ribbons using largely translational controller motion. Specifically, we enable users to directly impact ribbon ruling orientation via predominantly translational gestures by forming rulings that are strictly orthogonal to the controller trajectory all times, and facilitate ease of use by smoothly and predictably changing ruling directions in response to user input. We extensively validate our method, confirming that it outperforms prior approaches across multiple modalities.

ACKNOWLEDGMENTS

We are deeply grateful to Elinor Palomares for her artistic inputs and Livio Cambranis for his help with user studies recruitment. The authors were supported by NSERC and CONACYT.

REFERENCES

Judith Amores and Jaron Lanier. 2017. HoloARt: Painting with Holograms in Mixed Reality. In *Proc. Human Factors in Computing Systems*. 421–424.

Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proc. Human Factors in Computing Systems*. 5643–5654.

Rahul Arora and Karan Singh. 2021. Mid-Air Drawing of Curves on 3D Surfaces in Virtual Reality. *ACM Trans. Graph.* 40, – (2021), 17.

Ilya Baran, Jaakko Lehtinen, and Jovan Popović. 2010. Sketching clothoid splines using shortest paths. In *Computer Graphics Forum*, Vol. 29. 655–664.

Mayra Donaji Barrera Machuca, Wolfgang Stuerzlinger, and Paul Asente. 2019. The Effect of Spatial Ability on Immersive 3D Drawing. In *Proc. Creativity and Cognition*. 173–186.

Sukanya Bhattacharjee and Parag Chaudhuri. 2020. A Survey on Sketch Based Content Creation: from the Desktop to Virtual and Augmented Reality. *Computer Graphics Forum* (2020).

Holger Diehl, Franz Müller, and Udo Lindemann. 2004. From raw 3D-Sketches to exact CAD product models Concept for an assistant-system. In *Sketch Based Interfaces and Modeling*.

M.P. do Carmo. 2016. *Differential Geometry of Curves and Surfaces*. Dover Publications.

Davis A Forman, Garrick N Forman, Maddalena Mugnosso, Jacopo Zenzeri, Bernadette Murphy, and Michael WR Holmes. 2020. Sustained isometric wrist flexion and extension maximal voluntary contractions similarly impair hand-tracking accuracy in young adults using a wrist robot. *Frontiers in Sports and Active Living* 2 (2020).

GravitySketch. 2019. Gravity Sketch. (2019). https://www.gravitysketch.com/

Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton. 2002. Creating Principal 3D Curves with Digital Tape Drawing. In *Proc. Human Factors in Computing Systems*. 121–128.

Tovi Grossman, Ravin Balakrishnan, and Karan Singh. 2003. An Interface for Creating and Manipulating Curves Using a High Degree-of-freedom Curve Input Device. In *Proc. CHI Conference on Human Factors in Computing Systems.* 185–192.

LLC Gurobi Optimization. 2020. Gurobi Optimizer Reference Manual. (2020). http://www.gurobi.com

Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions. In *Proc. CHI Conference on Human Factors in Computing Systems.* 1063–1072.

Zhiyang Huang, Nathan Carr, and Tao Ju. 2019. Variational Implicit Point Set Surfaces. *ACM Trans. Graph.* 38, 4, Article 124 (2019).

J.H. Israel, E. Wiese, M. Mateescu, C. Zöllner, and R. Stark. 2009. Investigating three-dimensional sketching for early conceptual design-Results from expert discussions and user studies. *Computers and Graphics* (2009), 462 – 473.

B. Jackson and D. F. Keefe. 2016. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Trans. on Visualization and Computer Graphics* (2016), 1442–1451.

Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling Cumulative Arm Fatigue in Mid-Air Interaction Based on Perceived Exertion and Kinetics of Arm Motion. In *Proc. CHI Conference on Human Factors in Computing Systems.* 3328–3339.

D. Keefe, R. Zeleznik, and D. Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *IEEE TVCG* 13, 5 (2007), 1067–1081.

Daniel F. Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H. Laidlaw, and Joseph J. LaViola, Jr. 2001. CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience. In *Proc. I3D.* 85–93.

Viktor Kelkkanen, Markus Fiedler, and David Lindero. 2020. Bitrate Requirements of Non-Panoramic VR Remote Rendering. In *Proc. 28th ACM International Conference on Multimedia.* 3624–3631.

Yongkwan Kim, Sang-Gyun An, Joon Hyub Lee, and Seok-Hyung Bae. 2018. Agile 3D Sketching with Air Scaffolding. In *Proc. Human Factors in Computing Systems.* 238:1–238:12.

Robert I Kumar, Garrick N Forman, Davis A Forman, Maddalena Mugnosso, Jacopo Zenzeri, Duane C Button, and Michael WR Holmes. 2020. Dynamic wrist flexion and extension fatigue induced via submaximal contractions similarly impairs hand tracking accuracy in young adult males and females. *Frontiers in Sports and Active Living* 2 (2020), 135.

Joseph J LaViola, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice.* Addison-Wesley Professional.

Roman Luks and Fotis Liarokapis. 2019. Investigating Motion Sickness Techniques for Immersive Virtual Environments. In *Proc. 12th ACM International Conference on Pervasive Technologies Related to Assistive Environments.* 280–288.

James McCrae and Karan Singh. 2009. Sketching piecewise clothoid curves. *Computers & Graphics* 33, 4 (2009), 452–461.

Mozilla. 2021. A-Painter. (2021). https://blog.mozvr.com/a-painter/

Peng Nan, Amnad Tongtib, and Theeraphong Wongratanaphisan. 2019. Evaluation of Upper Limb Joint's Range of Motion Data by Kinect Sensor for Rehabilitation Exercise Game. In *Proc. Medical and Health Informatics.* 92–98.

Oculus. 2019. Quill. (2019). https://quill.fb.com/

Oculus. 2021. Oculus VR best practices guide. "https://developer.oculus.com/learn/". (2021).

PaintLab. 2019. PaintLab VR. (2019). http://paintlabvr.com/

Andrew K. Palmer, Frederick W. Werner, Dennis Murphy, and Richard Glisson. 1985. Functional wrist motion: A biomechanical study. *The Journal of Hand Surgery* 10, 1 (1985), 39–46.

Robert S Porter and Justin L Kaplan. 2011. *The Merck manual of diagnosis and therapy.* Merck Sharp & Dohme Corp.

Helmut Pottmann and Johannes Wallner. 2009. *Computational line geometry.* Springer Science & Business Media.

Dominik Rausch, Ingo Assenmacher, and Torsten Kuhlen. 2010. 3D Sketch Recognition for Interaction in Virtual Environments. In *Workshop in Virtual Reality Interactions and Physical Simulation.* The Eurographics Association.

Enrique Rosales, Jafet Rodriguez, and Alla Sheffer. 2019. SurfaceBrush: From Virtual Reality Drawings to Manifold Surfaces. *ACM Transaction on Graphics* 38, 4 (2019).

S. Schkolne and P. Schroeder. 1999. *Surface Drawing.* Caltech Department of Computer Science Technical Report CS-TR-99-03.

Shun'ichi Tano, T. Kodera, Takashi Nakashima, I. Kawano, K. Nakanishi, G. Hamagishi, M. Inoue, A. Watanabe, T. Okamoto, K. Kawagoe, K. Kaneko, T. Hotta, and M. Tatsuoka. 2003. Godzilla: Seamless 2D and 3D Sketch Environment for Reflective and Creative Design Work. In *INTERACT.*

TiltBrush. 2019. Google TiltBrush. (2019). https://tiltbrush.com/

Unity. 2019. SteamVR Plugin. (2019). https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

E. Wiese, J. H. Israel, A. Meyer, and S. Bongartz. 2010. Investigating the Learnability of Immersive Free-hand Sketching. In *Proc. SBIM.* 135–142.

Emilie Yu, Rahul Arora, Tibor Stanko, J. Andreas Bærentzen, Karan Singh, and Adrien Bousseau. 2021. CASSIE: Curve and Surface Sketching in Immersive Environments. In *Proc. CHI.* 1–14.

Shumin Zhai. 1998. User Performance in Relation to 3D Input Device Design. *SIGGRAPH Comput. Graph.* 32, 4 (Nov. 1998), 50–54.

A STUDY DETAILS

A.1 Participant Demographics

We recruited a different set of participants for each evaluation.

A.1.1 Biomechanical Generality Assessment. The first study included 10 participants, two females and eight males. The participants were between 21 and 44 years old. Seven users had VR Drawing experience using TiltBrush [2019]: one had two years of experience and the rest ranged from six month to a year. Three users had VR Drawing experience using GravitySketch [2019]: one had four years of experience, one six months, and one under six months.

A.1.2 Multi-brush Side-by-side Evaluation. The multi-brush evaluation study included 20 participants, six females and fourteen males. The participants were between 26 and 55 years old. Ten participants had no previous experience using VR. Five participants had more than two years of experience. The rest had less than six months of experience. Two participants were experienced in drawing in VR using TiltBrush: one three months and one five years respectively. The rest had no prior experience drawing in VR.

A.1.3 Head-to-Head Comparison Against GravitySketch. The head-to-head comparison study included 16 participants, six females, and ten males. The participants were between 27 and 45 years old. Seven participants had more than one year of experience using VR. The rest of the participants had less than one year of experience. Five participants had previous experience drawing in VR using TiltBrush. The rest had no prior experience drawing in VR.

A.2 Subjective Rating Data Collection

A.2.1 Biomechanical Generality Assessment. We asked the participants to rate the effectiveness, ease of use, ease of learning, and the perceived quality of the shapes they drew . After drawing the set of shapes with each tool, participants were asked to rank the following statements using a 5-point Likert scale while their output shapes: "It was effective to draw the given shape using brush X", "It was easy to draw the given shape using brush X", "It is easy to learn the brush X" with five options (i.e., "Strongly disagree", "Disagree", "Neutral", "Agree", "Strongly agree"), then we asked: "Rank the quality of the set of shapes you drew using brush X" with five options (i.e., "Bad", "Poor", "Neutral", "Good", "Excellent").

A.2.2 Multi-brush Side-by-side Evaluation. We asked the participants to rate the effectiveness, ease of use, and the quality of the outputs they produced independently for each shape they drew with each tool, and to rate the ease of learning of each tool after they drew all shapes using it. After drawing each shape, participants were asked to rank their level of agreement with the following statements using a 5-point Likert scale, from "Strongly agree" to "Strongly disagree": "It was effective to draw the given shape using the brush X" and "It was easy to draw the given shape using the brush X". After drawing all the shapes with each tool, participants answered the following statement: "It is easy to learn to use the brush X". The same procedure was repeated with each tool. Once participants completed all drawing tasks with all brushes, we showed them each target shape (on top) and the five drawings of this shape that they made using different brushes in random left to right order. We asked them to "rank the drawings by how well they accurately represent the surface of the target model. 5 being the highest quality and 1 being the lowest. Label each drawing with only one number".

A.2.3 Comparison Against GravitySketch. We asked the participants to rate the relative effectiveness, ease of use, output quality, and tool preference for each shape they drew. After drawing each shape with the two tools, anonymized as A and B, participants were asked to rank the following statements using a 5-point Likert scale while seeing the target shape on top and their output shapes side-by-side on the bottom (A to the left, B to the right) : "Which tool did you find more effective for drawing this shape?", "Which tool was easier to use while drawing this shape?", and "Which tool would you prefer to use if you wanted to draw this shape again?" with five options (i.e., "Definitely tool A", "Probably tool A", "Both are equally effective / easy to use / good", "Probably tool B", "Definitely tool B"), and "Please rate the output shapes you created using these tools in terms of their quality" with five options (i.e., "A is definitely better", "A is probably better", "Both shapes are of equal quality", "B is probably better", "B is definitely better").