WireDraw: 3D Wire Sculpturing Guided with Mixed Reality

Ya-Ting Yue[1], Xiaolong Zhang[1], Yong-Liang Yang[2], Gang Ren[3], Yi-King Choi[1], Wenping Wang[1]

[1]The University of Hong Kong, Hong Kong, China, {ytyue, xlzhang, ytchoi, wenping}@cs.hku.hk
[2]University of Bath, Bath, UK , y.yang@cs.bath.ac.uk
[3]Xiamen University of Technology, Xiamen, China , rengang@xmut.edu.cn

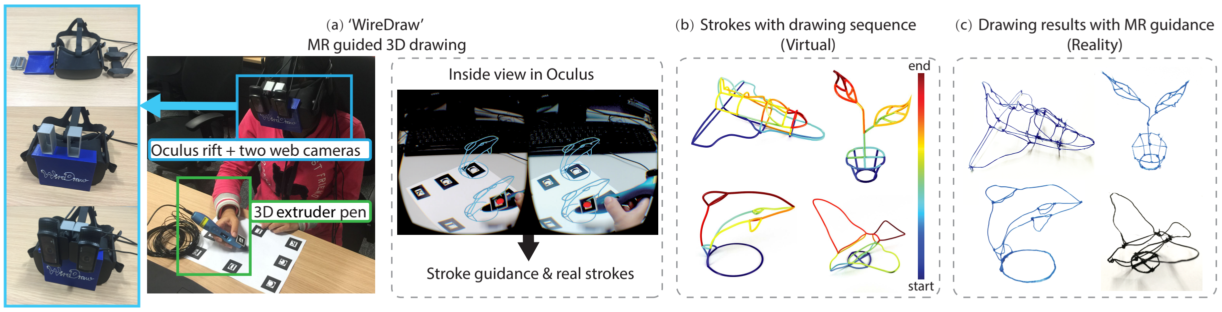
(a) 'WireDraw'
MR guided 3D drawing
Oculus rift + two web cameras
3D extruder pen
Inside view in Oculus
Stroke guidance & real strokes
(b) Strokes with drawing sequence
(Virtual)
end
start
(c) Drawing results with MR guidance
(Reality)

Figure 1. (a) We present a mixed reality system, called 'WireDraw', to immersively guide 3D wire objepct drawing using a 3D extruder pen. (b) Given an input wire model composed by curve segments, our system optimizes the combination and ordering of the curve segments based on a set of drawing principles for easy wire object sculpturing. (c) High-quality wire objects can be easily sculpted by novices using our system.

ABSTRACT

The availability of commodity 3D extruder pen allows direct drawing of 3D wire sculptures for novice users, enabling many novel applications such as intuitive spatial intelligence development for school students. However, the lack of spatial and structural cues among individual pen strokes makes the 3D drawing process challenging, which often leads to highly distorted and even incomplete wire sculptures. We present a mixed reality system, called 'WireDraw', to immersively guide the 3D drawing for easy wire sculpturing. The system design is based on novel 3D drawing principles and the subsequent optimization, making the stroke sequence of the wire model drawable and easy to draw. On-the-fly edits on unsatisfactory strokes are also allowed for creative design. We demonstrate the effectiveness of our system by testing on a variety of wire models and a user study. The results show that the visual guidance provided by our system is extremely helpful for drawing high-quality wire sculptures.

ACM Classification Keywords

H.5.1. Multimedia Information Systems:Artificial, augmented, and virtual realities; H.5.2. User Interfaces

Author Keywords

3D extruder pen, wire sculpture, stroke generation, drawing optimization, mixed reality

INTRODUCTION

Drawing is a fundamental skill for humans to visually convey shapes. The traditional drawing process requires drawing tools, such as pencil, inked pen, etc., to mark sketches on two-dimensional media such as a piece of paper. Since the sketches are usually drawn in 2D, when depicting 3D contents, one needs to pay significant efforts to mentally map 3D shapes onto the 2D canvas to avoid visual distortions and unrealistic artifacts. Freeform drawing in 3D space is possible based on trackable stylus or carefully designed 2D graphics interface, benefiting tasks such as sketch-based modeling and computer-aided design (CAD). However, the sketches are virtually created and require careful refinement for generating physically valid objects.

With the fast development of modern fabrication technologies, commodity 3D extruder pen is largely available recently. Compared with ordinary pen, 3D extruder pen has similar size and shape, but stores fabrication materials other than pigments. The user can easily move a 3D extruder pen in the air to generate physical wires, allowing direct drawing of real objects in 3D. This enables many novel applications for amateurs and novices, such as spatial intelligence development, fast prototype design exploration, and stylistic art creation.

While drawing with 3D extruder pen is intuitive, forming a plausible 3D sculpture using multiple wires remains challenging due to the lack of spatial and structural cues during the drawing process. First, the trajectory control of a 3D wire is tedious and error prone, which often leads to unacceptable geometry error. Second, it is difficult to keep the scale consistent among wires, causing severe shape distortion. Further, how to continuously draw multiple wires to form the entire object requires significant efforts, since keeping structural soundness and a clear drawing context is rather difficult. Note that pre-

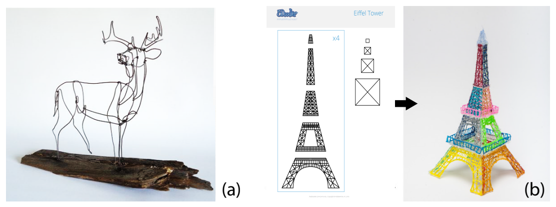

Figure 2. (a) 3D metal wire sculptures (By Artist: Bud Bullivant – WiredbyBud). (b) Objects drawn with 3D extruder pen. The strokes are drawn in 2D then assembled in 3D (photo courtesy of 3Doodler).

viously, a wire sculpture is often formed by metal wires (see Figure 2(a)). This requires professional skills for deforming and assembling metal wires, and is not feasible even for people well-trained in drawing and sketching.

In this work, we present a novel mixed reality system which provides intuitive guidance for easy wire sculpturing with a 3D extruder pen (see Figure 1(a)). This is based on considering two aspects in the 3D drawing process. First, the wire model should be physically *drawable*. Second, the drawing process should be *simple* for novice users. We first identify a set of 3D drawing principles concerning the above aspects. The principles are then quantitatively measured and optimized by analyzing the spatial and structural relations of constituent wires, which results in a sequence of strokes with corresponding viewing directions for a valid and easy drawing process (see Figure 1(b)). To further facilitate 3D wire object drawing, we involve mixed reality (MR) technology to guide the drawing process. The optimized stroke sequence is immersively mixed with the real environment, providing intuitive drawing references. Moreover, the 3D extruder pen is robustly tracked to help user control the wire trajectory and perform further edits if needed, resulting in high-quality wire objects (see Figure 1(c)).

We evaluate our system using a variety of wire models from different categories, including synthetic structures, articulated shapes, CAD designs, and man-made models. A user study is also conducted to verify the effectiveness of our system. The results show that our system can meet all the drawing requirements and is extremely helpful for easy object drawing using a 3D extruder pen.

In summary, our work makes the following contributions: 1) We identify a set of principles for easy drawing of 3D wire objects; 2) We present novel wire analysis and optimization algorithms that realize the proposed principles; 3) We develop the first MR-based system for intuitive 3D object drawing.

RELATED WORKS

Sketching interfaces. Humans are gifted of using sketches to convey geometric shapes [14]. Designing intuitive computer sketching interfaces is an attractive topic and benefits many applications. 'FreeDrawer' [35] allows the user to directly draw curves in a virtual environment, using a tracked stylus as the input device. 'Drawing on Air' [18] uses haptic-aided techniques for more controllable 3D line drawing. 'ILoveSketch' [4] integrates a coherent set of 3D curve sketching methods based on symmetry analysis and epipolar geometry. Schmidt et al. [31] present a pure-inference interface for 3D lines and curves from single-view sketches. 'JustDrawIt' [12] provides efficient sketching tools for adapting a new stroke to existing strokes. Fu et al. [10] investigate how to simulate artist drawing by automatically ordering of 2D sketches. 'Mockup Builder' [7] enables 'on-and-above-the-surface' sketching by tracking hand and fingers in a stereoscopic environment. 'SketchingWithHands' [19] incorporates tracked hand information for sketching handheld product with proper scale and usage. Our goal is different from all previous works. Other than creating virtual sketches, we aim at sketching physical strokes to compose real 3D wire objects.

Interactive fabrication and smart handheld tools. With the development of modern fabrication techniques, researchers have paid more attention on how to facilitate human interactions in the fabrication process for creative design and fast prototyping. To help the user communicate with the fabrication device (e.g., CNC machine, laser cutter, 3D printer), various interaction techniques have been employed via different media, such as digital pen [33], touch screen [36], laser pointer [26], etc. More recently, the rise of smart handheld fabrication tools [40] allows direct user accessability for intimate fabrication. 'FreeD' [39, 41] presents a handheld digital milling device that combines digital fabrication and craft. 'CAP' [32] allows novices to experience spray painting with an augmented airbrush. 'D-Coil' [28] provides a digital modeling system with a tailor-made handheld extruder, allowing wax coiling for tangible design and fabrication. Using also a commodity extruder pen, Roumen et al. [30] propose mobile fabrication to allow 'personal fabrication on the go' by referring to 2D shapes (in a similar format as in Figure 2(b) left) displayed on a mobile phone. Unlike previous works that often rely on tailor-made fabrication equipment, our work is based on commodity fabrication device - a 3D extruder pen, which is more accessible to novice users. Also, our work facilitates the user to draw 3D freeform curves in the air instead of being constrained by composing 2D planar strokes.

Wire-based fabrication. There exist several works on how to fabricate wire-based objects for stylistic design or fast prototyping. Garg et al. [11] present a wire mesh design framework that can approximate an input 3D shape using metal wires woven in a regular grid. Iarussi et al. [17] present a system called 'WrapIt' to fabricate 2D metal wire sculptures for wire-wrapped jewelry. Miguel et al. [24] investigate how to design stable planar-rod structures from a set of contours of a 3D shape. For fabricating general 3D wire models with curved strokes, traditional 3D printing techniques based on additive layers cannot be applied. This is because the wire thickness can largely affect the printing quality. Enforcing print head to extrude wires according to our optimized stroke sequence is also not feasible, since the fabricated strokes can easily occlude the print head when extruding the remaining strokes. Recently, tailor-made 3D printing devices are developed to efficiently fabricate wire-framed triangular and quadrilateral meshes with straight edges [25, 27, 37, 15], but cannot handle 3D curved wires. Compared with prior work, our wire model is composed of freeform wires represented by space curve in 3D. We focus on how to guide novice users to easily sketch real objects. The usage of a commodity extruder pen allows broad applications for education, prototype design, and stylized fabrication.

MR-guided drawing and fabrication. Mixed reality (MR) bridges real-world activities and digital experience, allowing novel interactive graphics systems for various applications. A recent survey on related techniques and applications can be found in [5]. Here we only discuss the most relevant work on MR-guided drawing and fabrication. In terms of drawing, Flagg and Rehg [9] employ multiple projectors to create an interactive display on the canvas to guide artists to paint using traditional media and tools. Laviole and Hachet [22] demonstrate that traditional 2D drawing can be enhanced by projecting real photos or virtual renderings on tracked sheets of paper. Lee et al. [23] utilize a 'shadow image' of suggestive contours that are updated in real time to guide 2D drawing. Iarussi et al. [16] enhance traditional 'drawing-by-observation' techniques by providing drawing guidance with construction lines extracted from a model photograph. Regarding fabrication, Lau et al. [21] present a system called 'Modeling-in-context', which uses a single photo as reference to design fabricatable objects. Rivers et al. [29] project virtual guidance to help sculpt polymer clay. Gupta et al. [13] demonstrate a real-time system called 'DuploTrack' that can track the Duplo® block assembly process and provide visual guidance for easy model construction. Weichel et al. [34] introduce a mixed-reality environment called 'MixFab' for personal fabrication, where virtual designs and physical objects are naturally mixed to help the design and fabrication process. Our system differs from pervious works in a way that it guides 3D drawing of high-quality objects for novice users, which would otherwise be very difficult. To the best of our knowledge, it is the first MR system of its kind.

PRINCIPLES FOR DRAWING 3D WIRE OBJECTS

3D extruder pen provides the possibility of intuitive sketching 3D physical strokes to form a real object. However, compared with 2D drawing using an ordinary pen, it is much more challenging for novice users to control the trajectory of a 3D stroke. As a result, the usage of 3D extruder pen is often limited. The most common scenario is very similar to 2D drawing. The user just draws 2D strokes (e.g., on paper) as building blocks and then assembles all the blocks to build up the whole object [1] (see also Figure 2(b)). In this case, all the strokes are planar curves and difficult to represent features of freeform shapes. Also, it requires significant efforts to keep the compatibility among building blocks and assemble them afterwards, which largely affects the quality of the resultant wire object. One possibility is to use a 3D solid object as a reference and draw strokes on it. This simplifies the trajectory control but a reference object needs to be fabricated beforehand.

In this work, we aim at 3D drawing of freeform strokes to directly compose a wire object which realizes the corresponding virtual wire model. We first perform a pilot study in which we asked a few participants to draw wire objects with reference models and collected their feedbacks. Then we identify a set of drawing principles accordingly concerning two aspects to guide the 3D drawing process. First, the wire model should be *drawable*. Second, the drawing process should be *simple* for the user. The proposed principles serve as the foundation of our system to optimize the 3D drawing process.

Pilot study for 3D drawing. We asked five participants to draw a few wire objects with 20-30 curve segments according to virtual wire models shown on a computer display. We gave them 5 minutes to practice the 3D extruder pen by drawing a simple cube. Then we asked them to draw the target object with the virtual model aside as reference. After the drawing session, most of the participants expressed that drawing in 3D is a very challenging task and Figure 3 shows four typical results. A summary of their feedbacks is as follows. 1) Participants were likely to be confused during the drawing process. They often drew extra curves or missed curves, resulting in wrong structure of the wire object. 2) It is difficult to control the trajectory of the stroke according to the reference model, leading to erroneous scale and/or geometry. The right distance between strokes is difficult to infer. 3) Participants had to spend a lot of time observing the virtual wire model to figure out which part of the model to draw next. 4) A poor stroke drawing sequence can easily cause problem and even failure. For instance, by referring to the virtual model, participants often started with wires that were close to them, resulting in occlusions for drawing rear wires. Then the wire object needs to be rotated to move the occluded wires closer, which might again occlude the remaining wires. Whether or not a stroke sequence is optimal is difficult to infer.

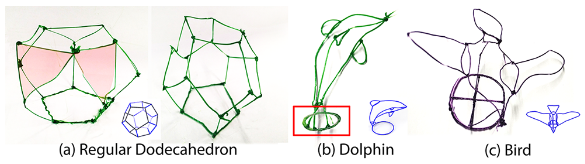

Figure 3. Typical drawing results from the pilot study. The challenges of 3D drawing can easily lead to (a) wrong structure and inaccurate stroke length; (b) erroneous scale and unbalance (the circular base is too small in this case); and (c) severe shape distortion.

Drawability. To make the wire model drawable, the constituent strokes should obey 1) *Dependency*: The strokes to be drawn should be supported by existing strokes that have been drawn. This is to keep the drawing context during the drawing process and avoid further assembling.

Simplicity. To make the drawing process simple for the user. The strokes and their drawing order should be optimized to meet the following requirements. 2) *Low complexity*: The wire object should be composed by a small number of strokes. 3) *Smoothness*: A single stroke should be smooth for continuous drawing. 4) *Planarity*: It is much easier for the user to control the trajectory if a stroke is planar. Then the drawing is very similar to 2D case since the user only needs to move the pen with two degrees of freedom. 5) *Clarity*: When drawing a stroke, it should be viewed from a direction, in which the stroke is clear to the user and without depth ambiguity. 6) *Reachability*: Each stroke should be reachable for the 3D extruder pen when drawing along its trajectory. In other words, the stroke should not be blocked by existing strokes. 7) *Closeness*: The strokes should be ordered in a way that neighboring strokes are close to each other in terms of spatial distance and viewing direction. This is to reduce user's transition efforts when moving from one stroke to the next.

Note that some of the above principles share similar characteristics as [17] (principle 2 and 3) and [10] (principle 7). However, our work is in a different scenario. The drawability constraint enforces hard constraints on the stroke drawing order, making our problem more difficult to solve. Moreover, our principles are not only realized by computer algorithms, but also illustrated by an MR system to further facilitate the drawing process.

3D WIRE OBJECT DRAWING OPTIMIZATION

In this section, we elaborate how to quantitatively measure and optimize the proposed 3D drawing principles, such that a drawable and simple stroke sequence can be provided to the user for easy 3D drawing of high quality wire objects.

3D Drawing Conventions

We focus on the most common 3D object drawing setup similar to 2D drawing. It involves a 3D extruder pen and a flat supporting domain (e.g., a piece of paper on table) to hold the generated physical strokes. The size of the wire model is scaled within $20cm \times 20cm \times 20cm$, such that the user could sit still and draw all the contents. The user's eye position is roughly $30cm$ above the supporting plane. The pen's position is on the right/front side of the wire model and above the support plane for right-handers (left/front side for left-handers respectively). The bottom strokes will stick on the supporting domain and the user can rotate the supporting domain along with the wire object. To help problem formulation, we assign a 3D coordinate system onto the supporting domain, where the origin is the domain centroid, xy-plane is the supporting plane, and z axis is straight up.

We denote the wire model as $\mathscr{W}(\mathscr{V},\mathscr{E},\mathscr{C})$, where $\mathscr{V}$ contains all the vertices of the wire model, $\mathscr{E}$ is the set of edges connecting neighboring vertices, and $\mathscr{C} = \{c_n | (1 \leq n \leq N)\}$ consists of N curve segments divided by those vertices with valance not equal to 2. The wire model can be generated computationally [8, 38] or by sketching interfaces such as those discussed in the "Related Works" section, as long as all the curve segments form a connected layout. We assume that the wire model is physically in balance, otherwise a supporting base is added to ensure the stability of the strokes to be drawn (see the circular base of the dolphin model in Fig2ure 1 (b)).

Similar as in 2D, 3D drawing is also performed incrementally where physical strokes are assembled one by one to form the entire object. As such, we simply use a sequence of M strokes $\boldsymbol{S} = (s_1, s_2, ..., s_M)$ to denote this incremental procedure, where a single stroke s_m is an Eulerian path composed by one or more curve segments, and can be continuously drawn without breaking the stroke. On the other hand, in contrast to 2D drawing where the viewing direction w.r.t. the drawing is pretty much fixed, for 3D drawing, it is better to allow viewpoint change to easily draw different parts of the 3D object. In our system this is achieved by rotating the supporting domain and the attached wire object (an analogous example is rotating the model when assembling Duplo® blocks). Let $\boldsymbol{\Phi} = (\phi_1, \phi_2, ..., \phi_M)$ denote the corresponding viewing directions when drawing consecutive strokes, the 3D drawing process can be represented by $\boldsymbol{S}$ and $\boldsymbol{\Phi}$.

3D Drawing Principle Measurements

Based on the above 3D drawing conventions, we now present the quantitative measurements for the proposed 3D drawing principles, which further help the formulation of the subsequent 3D drawing optimization.

Dependency. The structural dependency among strokes plays a crucial role for the drawability of the wire model. We use a directed acyclic graph (DAG) for this measurement, where graph nodes represent individual strokes, and graph edges (directed) encodes that the target node depends on the support of the start node during the drawing process. The directed edges are iteratively generated based on a modified breath first search (BFS), which starts from a (hidden) root node representing the supporting domain. More specifically, each node is traversed based on the connectivity between nodes under an additional height restriction, which ensures that lower strokes should be visited first. The height constraints involve structural consideration and provide valid dependency between strokes for feasible drawing. Figure 4 shows an example of DAG-based dependency measurement. The height constraint guarantees that lower strokes will be drawn first, otherwise there may be wrong dependency highlighted by the red arrow. Note that in most cases, the dependency is from bottom to up, expect for some special cases, where lower stroke might depend on the stroke above (denoted by green arrow). Also, the strokes sharing one supporting stroke do not have dependency with each other. The initial order among these strokes (generated from modified BFS) will be further refined in a subsequent optimization step.

Low complexity. This principle is simply measured by the number of strokes:

$$E_{complex}(\boldsymbol{S}) = M \tag{1}$$

Smoothness. The stroke smoothness is measured by the angle between consecutive curve segments at their junctions:

$$E_{smooth}(\boldsymbol{S}) = \sum_{m \in [1,M]} \sum_{i} \left| \pi - \arctan \frac{t_i^m \times t_{i+1}^m}{t_i^m \cdot t_{i+1}^m} \right|, \tag{2}$$

where t_i^m and t_{i+1}^m are the tangents of the i-th pair of incoming curve segment and subsequent outgoing curve segment along s_m. Note that the above two terms are directly adopted from [17].

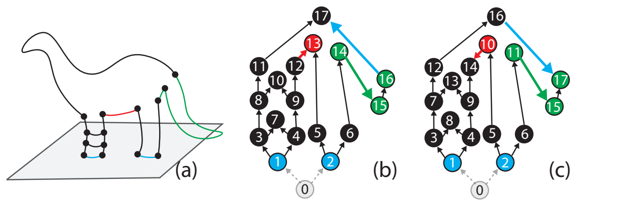

Figure 4. The drawing dependency among strokes is measured by a directed acyclic graph (DAG), where graph nodes represent individual strokes. (a) An example wire model. (b) Starting from the root node (in gray), we use a modified breath first search (BFS) with height constraints to build the DAG with ordered nodes. (c) BFS without height constraints leads to unrealistic dependency where lower curve depends on higher curve (denoted by the red and blue arrows). Green arrow shows the only case where lower curve actually depends on higher curve, otherwise there will be no support from below.

Planarity. For a given stroke, we construct a local coordinate system based on Principal Component Analysis (PCA). Then the planarity is measured by the range covered by the stroke along the normal of the maximum principal plane (z-axis of the local coordinate system) . For all strokes, we have:

$$E_{planar}(\boldsymbol{S}) = \sum_{m\in[1,M]} |z_m^{max} - z_m^{min}|, \tag{3}$$

where z_m^{max} and z_m^{min} are the maximal and minimal z values of s_m in the local coordinate system. Please note that the planarity term is only considered if a stroke consists of more than one curve segments. This is because we do not optimize the planarity of a single curve segment in order to preserve the geometric features of the wire model.

Clarity. Due to the depth ambiguity of the human vision system, the stroke shape may not be clear to the user if being viewed from certain direction, along which the stroke depth has big variation. For instance, a roughly straight stroke is difficult to draw if it is nearly along the current viewing direction, while it would be much easier from a side view due to a clearer shape (see Figure 5(a)). To ensure that each stroke can be easily drawn in a good viewing direction, we define the clarity measurement based on the angle α between the viewing direction and the principal direction(s) of the stroke. More specifically, as shown in Figure 5(a), we suppose the user's viewing direction is varying in a range above the supporting plane (within $[30°, 60°]$ to the z-axis as in our experiments). The principal direction of a stroke is computed by PCA. It is unique for elongated stroke, while for general stroke, the principal directions span a principal plane. We empirically set $\alpha_{\min} = 20°$ as the minimum angle to ensure clarity. Based on the clarity measurement, we would like to further compute a viewing direction with guaranteed clarity when drawing each stroke. In practice, we find that restricting the user to a unique viewing direction is very tedious. Hence we provide a flexible solution, where the space above the supporting plane is divided into eight domains (see Figure 5(b)). The user is free to change viewing direction within the domain when drawing a stroke. For each stroke, we check the angle α within each domain, and only keep the domains with $\alpha > \alpha_{\min}$ as valid candidates for further optimization. For each domain, we use the longitude of the domain's bisecting plane (latitude is set to 0 for simplicity) to represent the viewing direction (see Figure 5(c)).

Reachability. When drawing a stroke, only adjusting viewing direction is not enough. Even with guaranteed clarity, the stroke may still be occluded by existing strokes. Then the user needs to manually relocate or even break the blocking strokes so that the pen nip could reach the target position of the current stroke, making the drawing process cumbersome. Therefore, how to measure the reachability of the current stroke is important. This is similar to the reachability problem for a mill-drill in CNC machining [6]. However, it is difficult to formulate the problem in the same way as the pen is held by a human hand with much more flexible movements. Hence we define the measurement by only considering the user's drawing hand, and also the spatial relation between strokes. Specifically, for right-handers, the pen is usually placed on the right/front side of the stroke (left/front side for left-handers). Moreover, if the current stroke depends on a lower stroke (i.e., the supporting dependency is bottom-up), the pen should be placed above, otherwise below. As shown in Figure 6, we first build a local coordinate system where y-axis is the selected viewing direction and z-axis is the global one. Then an axis-aligned bounding box is constructed for the current stroke. The origin of the coordinate system is placed at one corner of the bounding box, so that the current stroke lies in the forth octant (or eighth octant if support dependency is top-down). Figure 6 illustrates the two cases for right-hander and the left-hander respectively. In the local coordinate system, the existing strokes intersecting with the octant of the current stroke are possible to cause occlusion. The amount of occlusion is measured by the shortest distance between the current stroke s_m and the part of existing stroke s_n which lies in the corresponding octant, denoted as $d(s_m, s_n)$. If no occlusion occurs between s_m and s_n, $d(s_m, s_n)$ is ∞. The reachability of s_m w.r.t. s_n is defined as: $r(s_m, s_n) = \frac{1}{d(s_m,s_n)+1}$, where smaller value means more reachable. The overall reachability is simply an integration over the entire stroke sequence:

$$E_{reach}(\boldsymbol{S}, \boldsymbol{\Phi}) = \sum_{m\in[2,M]} \sum_{n\in[1,m-1]} r(s_m, s_n), \tag{4}$$

where s_m is the current stroke, s_n is one of the existing strokes, and $r(s_m, s_n)$ is the reachability of s_m w.r.t. s_n, which depends on the viewing direction ϕ_m as described before.

Closeness. The closeness between two strokes are measured by two terms in different aspects. The first term measures the transition efforts due to changing viewing directions between consecutive strokes:

$$E_{view}(\boldsymbol{S}, \boldsymbol{\Phi}) = \sum_{m\in[1,M-1]} \angle(\phi_m, \phi_{m+1}), \tag{5}$$

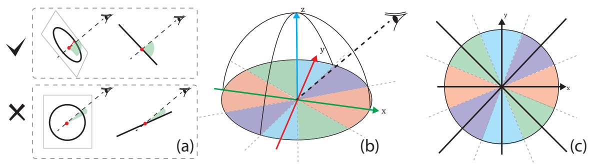

Figure 5. (a) The clarity is measured by the angle between the viewing direction of the current stroke and the stroke's principle direction/plane. (b) The space above support plane is divided into eight domains. (c) The clarity of a stroke is measured when being viewed within each domain.

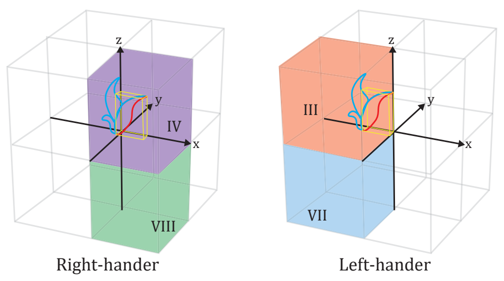

Figure 6. The reachability measurement for right-hander (Left) and left-hander (Right) respectively. Red stroke is the current stroke to be drawn.

where $\angle(\phi_m, \phi_{m+1}) \in [0, \pi]$ is the viewing direction difference between two consecutive strokes s_m and s_{m+1} in the drawing sequence. The second term measures the transition efforts caused by different locations of consecutive strokes:

$$E_{dist}(\boldsymbol{S}) = \sum_{m \in [1, M-1]} |p_{m+1} - q_m|, \tag{6}$$

where p_{m+1} is the start point of s_{m+1}, q_m is the end point of s_m. The start/end point when drawing one stroke is determined by the order and connectivity between strokes (details in supplemental document).

Drawing Optimization

Given a connected wire model $\mathscr{W}(\mathscr{V},\mathscr{E},\mathscr{C})$, our method jointly optimizes 1) the combination of curve segments to form individual strokes, 2) the viewing direction for drawing each stroke, and 3) the drawing sequence of all strokes, with the help of the quantitative measurements of the 3D drawing principles. The output of the optimization is a sequence of strokes with corresponding viewing directions.

We formulate the optimization as a coupled grouping and ordering problem of all N curve segments. During the optimization, we use an index $x_i \in \{1, 2, ..., M\}$ to encode the group and order of curve segment $c_i \in \mathscr{C}$. All curve segments with index $m \in [1, M]$ are in the the same group, and form the m-th stroke s_m in the drawing process. It is easy to see that all the indices x_i's indicate both stroke formation and ordering, resulting in the entire sequence of M strokes ($M \leq N$). We iteratively optimize the stroke sequence $\boldsymbol{S}$ and the viewing directions of individual strokes $\boldsymbol{\Phi}$ by minimizing a constrained objective function respecting the proposed principles.

$$\begin{aligned} \text{min. } & E_{complex}(\boldsymbol{S}) + \lambda_s E_{smooth}(\boldsymbol{S}) + \lambda_p E_{planar}(\boldsymbol{S}) + \\ & \lambda_v E_{view}(\boldsymbol{S}, \boldsymbol{\Phi}) + \lambda_d E_{dist}(\boldsymbol{S}) + \lambda_r E_{reach}(\boldsymbol{S}, \boldsymbol{\Phi}). \qquad (7) \\ & \text{s.t. each stroke } s_m \text{ is an Eulerian path} \\ & \text{\& the strokes satisfy } \textit{dependency} \text{ constraints} \\ & \text{\& each viewing direction ensures } \textit{clarity} \end{aligned}$$

Similar as in [17], we also apply simulated annealing [20] to optimize the drawing process due to its effectiveness of sampling the solution space (pseudo code in supplemental document). Four perturbation operators (see Figure 7) are employed to explore the configuration space of $(\boldsymbol{S}, \boldsymbol{\Phi})$ while satisfying constraints from *dependency* and *clarity* principles. The main difference between our work and [17] is that our optimization is in a more challenging scenario where 3D drawing requirements (e.g., dependency) need to be met, while they only optimize simple 2D wires. Besides, other than simple stroke partitioning, we also optimize stroke ordering and viewing directions due to structural and fabrication considerations. Now we specifically discuss the optimization initialization and perturbation operators.

Optimization initialization. We generate the initial configuration of stroke sequence based on the various measurements presented previously. First, each curve segment c_i is simply treated as a single stroke. Then the corresponding index x_i is generated by the modified BFS-based graph node traversal, which guarantees the structural dependency among strokes.

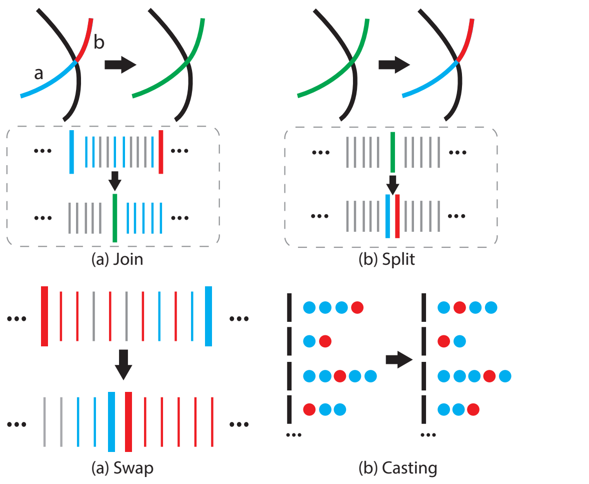

Figure 7. Four perturbation operators used in our optimization. (a) Join. Blue and red strokes are merged to a new green stroke. (b) Split. Split one stroke into two consecutive strokes. (c) Swap. Red and blue strokes are swapped. (d) Cast. Randomly select one viewing direction (red dot) from valid candidates (blue dots). Colored line segments in a row denote ordered strokes. Thinner and shorter line segments denote ancestors/descendants of the stroke with the same color (depending on the order).

Finally, the viewing direction ϕ_i for drawing c_i is determined by the clarity measurement. We randomly select one direction from the valid candidates as initialization.

Perturbation operators. Figure 7 shows the four perturbation operators used to explore the configuration space during the optimization. The first three operators update the stroke sequence $\boldsymbol{S}$, and the fourth operator updates the stroke viewing directions $\boldsymbol{\Phi}$ from valid candidates to ensure clarity.

- **join.** The *join* operator conditionally merges two connected strokes into one while keeping the *dependency* constraints. Specifically, we only join two strokes s_i and s_j $(i < j)$ if s_j is not an indirect descendant of s_i in the DAG, otherwise it violates the dependency of another stroke s_k which lies between s_i and s_j in the stroke sequence. While this operator keeps the stroke dependencies, it changes the number of nodes of the DAG. A straightforward way is to rebuild the DAG from the virtual root node and perform modified BFS to reorder the updated strokes. However, this would destroy the stroke sequence optimized by the *swap* operator (see later discussion), as BFS leads to unique ordering which only depends on the stroke configuration. Instead, we locally update the sequence by analyzing the dependency of all the strokes between s_i and s_j in the current sequence. Among these intermediate strokes, we select all the descendants of s_i in the DAG and put them after the merged stroke while keeping their order in-between. The same rule applies for ordering the remaining intermediate strokes except that they are relocated before the merged stroke.

- **split.** The *split* operator splits a random stroke s_i, which has been merged before, back into two consecutive strokes s'_i and s'_{i+1} without affecting the dependency and order of the remaining strokes. The ordering of the two new strokes follows the trajectory of the stroke s_i (see supplemental document for stroke trajectory estimation).

- **swap.** The *swap* operator conditionally swaps two random strokes in the sequence while keeping the *dependency* constraints. Specifically, we only swap two strokes s_i and s_j $(i < j)$ if s_i and s_j are both ground strokes or non-ground strokes, and s_j does not depend on s_i (no path from s_i to s_j). After swapping, the ordering of the in-between strokes are updated according to the dependency w.r.t. s_i and s_j (see Figure 7), which is similar as for the *join* operator.

- **cast.** For a randomly selected stroke, the *cast* operator randomly selects a different viewing direction from the valid candidates.

Remarks. Please note that since our perturbation operators never create new internal vertices of odd valence, the optimization is guaranteed to maintain the Eulerian property of each stroke. This means each stroke can be continuously drawn. Also, when joining two strokes, the valid candidate viewing directions of the combined stroke is the intersection of those from two sub-strokes. This is to ensure the clarity for drawing every part of the combined stroke. If the intersection is empty, the two stokes remain separated.

Figure 8 shows the effectiveness of individual terms in the objective function. The convergence of the optimization with all energy terms is illustrated in Figure 9.

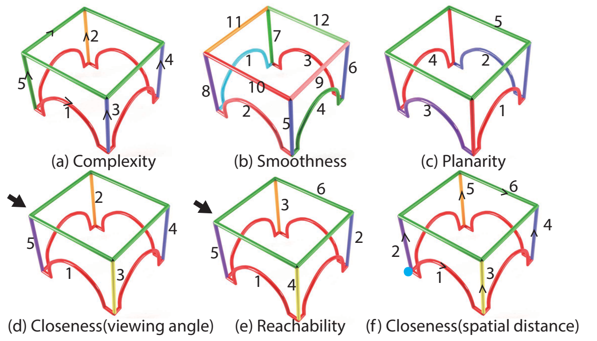

Figure 8. Effect of each energy term. (a) Complexity minimizes the number of strokes. (b) Smoothness avoids stroke with sharp turns. (c) Planarity prefers planar stroke. (d) Closeness (viewing direction) avoids frequent viewing direction change during the drawing process. In this case, the viewing direction (denoted by the arrow) is unique for all strokes. (e) Reachability avoids occlusion when drawing strokes in corresponding viewing directions. The four vertical strokes are ordered from far to close. (f) Closeness (spatial distance) minimizes the transition efforts (distances) when moving from one stroke to the next. The blue point is the start point when drawing the first stroke.

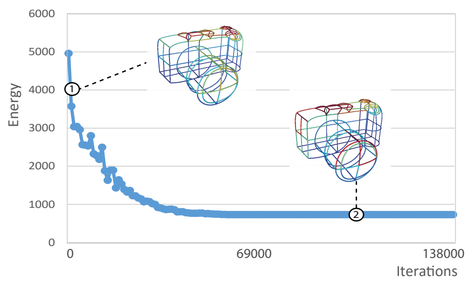

Figure 9. Energy convergence

MR-GUIDED DRAWING

Based on the 3D drawing optimization in the previous section, we could already provide the user with a step-by-step drawing instruction with incremental strokes illustrated in 2D or 3D. However, even with the help of such instruction, it is still very challenging for the user to draw plausible wire objects in practice (see also the user study in the “Results” section). The reason behind this is twofold. First, unlike drawing 2D strokes using an ordinary pen, it is very difficult to control the stroke trajectory when sketching in 3D. The additional dimension makes 3D drawing rather difficult, which often leads to highly distorted strokes. Second, the spatial relations between strokes are difficult to comprehend in 3D, which usually results in non-uniformly scaled and distributed strokes.

In this work, we present a novel mixed reality (MR) system, called ‘WireDraw’, for guided 3D wire sculpturing with a 3D extruder pen. Our system provides an immersive drawing environment by fusing the real space with virtual wire drawing guidance. The virtual guidance not only provides the user with important spatial and structural cues, but also practical hints in real-time to ensure the drawing quality. We now present the detailed configuration and functionality of our MR system.

System Setup

As shown in Figure 1(a), our MR drawing guidance system consists of an Oculus rift head-mounted display (HMD) mounted with two Logitech C920 webcams. The two webcams form a stereoscopic camera that captures the real scene, while the HMD exhibits virtual stroke sequence in the scene to guide the 3D drawing process. We use publicly available library ARTookit [3] to fuse real and virtual contents.

In our experiments, we use a piece of A4 paper as a simple platform to hold the physical wire object. Eight markers are attached to the paper to robustly align the coordinate systems in real and virtual worlds for immersive visual experience. Our drawing device could be any 3D extruder pen with melted plastic filament (ABS/PLA), such as 3D Scribbler [2] as used in our experiments.

MR-based Drawing Guidance

Drawing contents. To assist 3D drawing, we render a virtual wire model onto the real scene as a reference for the user. The strokes are displayed one at a time according to the optimized stroke sequence. Each stroke trajectory is animated from the starting point to the end point to simulate the real drawing process (see supplemental video). The user can trigger display/redisplay before drawing the real stroke. After a real stroke is drawn, the corresponding virtual stroke is hidden by default to avoid visual cluttering (see Figure 10). Besides, to provide a big picture to the user, we still show the entire wire model with high transparency. Hence, the user will know exactly where the current stroke is so as the current drawing progress.

Drawing position. Although the virtual stroke sequence provides highly intuitive guidance, the quality of the resultant wire object still largely depends on the real stroke trajectory created by the 3D extruder pen. To precisely control the trajectory, we put one marker close to the material button to track

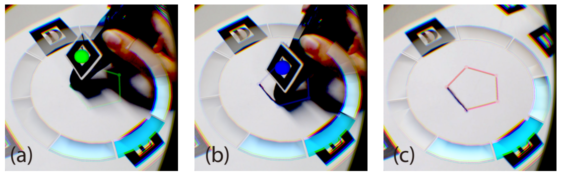

Figure 10. Drawing one stroke with visual guidance. (a) The virtual wire and an extra indicator on the pen become green when the pen touches the start/end/key point of the stroke. (b) The virtual wire and the indicator become blue when the pen touches other parts. (c) The drawn part (real wire in dark purple) and the remaining part (virtual wire in red due to no pen nib contact).

the pen nib position (see Figure 11(a)). The tracked position is utilized to check whether the pen nib touches the guiding stroke (see Figure 10). The stroke is highlighted in red if the pen nib is apart from it, otherwise in blue. The start and end points, and key points with high curvature are crucial for controlling stroke trajectory. We therefore show these points on the stroke and when the pen nib is in touch, the stroke is colored in green. The above position guidance provides strong spatial cues for drawing high-quality strokes. Note that when drawing one stroke, the virtual stroke might interfere the appearance of the real one created by the user. To avoid the interference, we set the part of the virtual stroke that has been drawn to transparent based on tracked pen nib.

Drawing angle. Based on the 3D drawing optimization, we render a color wheel around the virtual wire model, and highlight the optimized view direction for drawing the current stroke. Figure 12 shows an example.

Drawing context. We also realize that a good visualization of the overall drawing context is crucial for the user experience. To address this, we show aside a reference wire model with the current stroke highlighted (Figure 12). Also, it is important to well present the context between pen and strokes, since the user mainly focuses on them during the drawing. If we simply overlay the virtual strokes onto the real pen (captured by webcams), the spatial/depth relation between pen and strokes are difficult to infer by the user, resulting in non-immersive user experience and severe confusion when trying to place pen nib at the right position (see Figure 11 (b)). To solve this problem, we import a virtual pen model into the scene and align it with the real pen. Then instead of displaying the real pen underneath the strokes, we render the virtual pen together with the (virtual) strokes using Z-buffer (see Figure 11 (c)). Thanks to the markers placed on the real pen for robust alignment, the

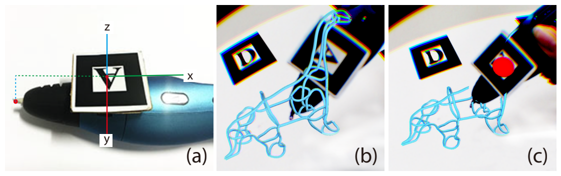

Figure 11. Marker-based pen nib tracking. (a) Marker centroid can be tracked by ARTookit. The pen nib position can thus be estimated by applying a fixed translation in a local coordinate system according to the marker. (b) The real pen is occluded by virtual strokes even though it is closer to the user. (c) To give user stronger spatial cues for placing the pen nib in the right position, we import a virtual pen aligned with the real pen to resolve occlusion.

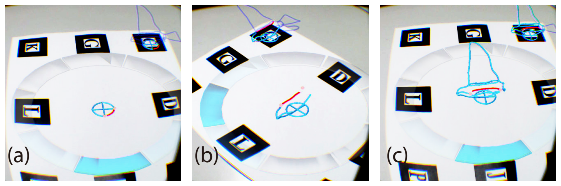

Figure 12. The optimized viewing direction for drawing each stroke is highlighted in our system. The blue arc indicates the optimal viewing direction (within a range) for a specific stroke. A reference model is also shown aside to provide a global context.

imported virtual pen does not affect the user experience, but provides much stronger spatial cues for placing the pen nib in the right position.

On-the-fly Stroke Editing

With the help of robust pen nib tracking, our system also allows on-the-fly edits on unsatisfactory strokes for creative design. During the drawing process, the user could pick and delete a stroke, then sketch a new stroke (see Figure 13) for replacement. When sketching a new stroke, the pen nib position is continuously tracked and stored, forming the basic shape of the stroke. The basic shape is then automatically refined by curve smoothing and re-sampling. Smart snapping and alignment are further applied to conjoin to the other strokes. Due to stroke shape change, we check the stroke sequence after the edit (mainly for dependency), and only re-optimize the stroke sequence if necessary. Figure 14 shows a more complicated example created by the user.

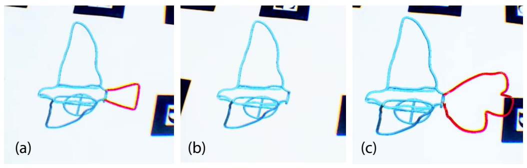

Figure 13. On-the-fly stroke editing. (a) Select one stroke (red). (b) Delete old stroke. (c) Draw new stroke.

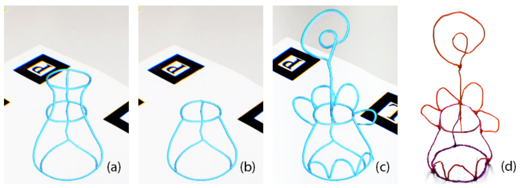

Figure 14. Stroke editing for creative design. Given an input wire model (a), our system allows the user to delete part of the model (b), and add new wires by on-the-fly stroke editing (c). (d) The final drawing result aided by 'WireDraw'.

RESULTS

We evaluate our system on various models from different categories, including synthetic structures, articulated shapes, CAD designs, and man-made models. The results can be found in Figure 18 and the supplemental materials. Taking an input wire model composed by curve segments, our optimization can effectively generate a stroke sequence which is drawable and easy to draw for novice users (see also the user study). This verifies that the proposed principles on drawability and simplicity are successfully realized.

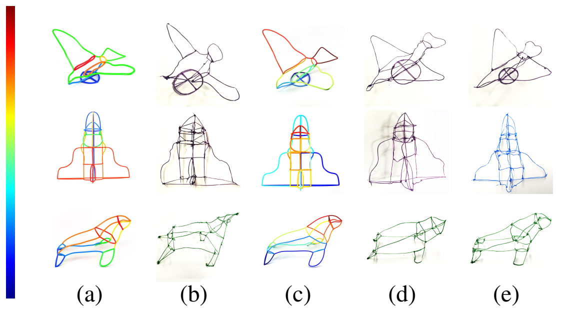

Figure 15. 3D drawing results in different scenarios. (a) Stroke sequence of free drawing (determined by the user). (b) Free drawing results. (c) Our optimized sequence. (d) Results by referring to our optimized stroke sequence shown on computer display. (e) Results under guidance of 'WireDraw'.

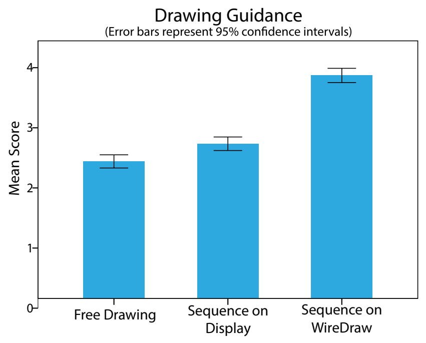

Figure 16. Visual evaluation of drawing quality in different scenarios.

User study. We conducted a user study to evaluate the effectiveness of our MR-based drawing guidance. We recruited 10 volunteers from graduate students in computer science. We asked them to draw wire objects in three different scenarios with random order. The scenarios are: 1) *Free Drawing*: Only the input wire model was rendered on a computer display for drawing reference. 2) *Sequence on Display*: The optimized stroke sequence was illustrated on a computer display for reference. Note that the participants were allowed to rotate/translate/zoom the virtual scene in the first two scenarios. 3) *Sequence on WireDraw*: Drawing based on our MR-based guidance system. We asked each participant to select a preferable model to draw for the three scenarios, while making sure a variety of models were covered by the participants overall. For each scenario, the participants were given a short period of time (around 5 mins) to practice by drawing a simple cube. Then the participants started to draw the wire object with help of the reference/guidance in the scenario. We also made a note if the drawing was not completed.

The last column of Figure 18 shows randomly picked wire objects drawn by participants with the help of our system. These objects well approximate the shape and structure of the input wire models. Figure 15 shows the wire objects drawn in different scenarios. More results can be found in the supplemental document. To evaluate the drawings from different scenarios (3 drawings corresponding to 3 scenarios from each participant), we asked 27 people (not the participants for drawing) to carefully rate the drawing quality by comparing the photo of the result object with the rendering of the reference model (from roughly the same view). The score was an integer between 1 (very bad) and 5 (very good), and treated as interval data such that parametric ANOVA could be applied. We then performed a repeated measure ANOVA to analyze the scores from different scenarios.

Main effects were found for drawing guidance [1] ($F_{1.46,37.87} = 173.20, p < 0.001$). Post hoc Bonferroni pairwise comparisons showed that *Sequence on WireDraw* is significantly better than the other two ($p < 0.001$), indicating 'WireDraw' provides the best support for our users. Besides, *Sequence on Display* is significantly better than *Free Drawing* ($p < 0.001$), which proves that our drawing sequence optimization is also effective. The mean score of drawing quality is shown in Figure 16. Figure 17 shows some typical failure cases of *Free Drawing* caused by the lack of spatial and structural cues. For some cases, the participants even cannot complete the drawing due to wrong stroke ordering. The unreachable strokes are impossible to draw unless breaking existing strokes.

Note that we have tried to quantitatively measure the approximation error between the real wire object and the virtual reference model. We planned to compare them by 3D reconstruction of the wire object or 3D printing of the virtual model. But the reconstruction (based on stereo-vision and laser scanning) and fabrication (by 3D printer with additive manufacturing) are not reliable due to the thin shape of wires, preventing accurate approximation error estimation. However, visual evaluations are already convincing since the quality differences are distinctive.

User feedback. Overall the participants responded very positively to the visual guidance provided by 'WireDraw': "The system helps a lot and makes drawing much easier.", "I am not getting confused and making mistakes.", "It saves me a lot of time to look at the reference model." The stroke editing tool was also appreciated by the user: "I like the stroke editing tool to create new stuff." For wire models with many strokes, it still requires some user efforts: "The task becomes feasible but is still complicated.", "It requires a lot of patience." We also found that due to the mounted cameras are closer to the drawing platform than the user's eyes, the captured scene has

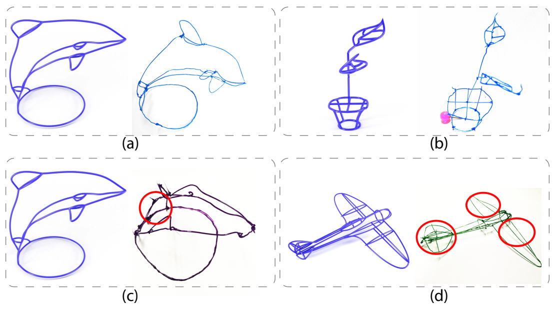

Figure 17. Failure cases of free drawing. The drawing results in (a) and (b) are highly distorted and cannot stand by themselves without MR guidance. (c) and (d) show the failure cases caused by wrong stroke sequence. (c) The participant drew the three major curves on dolphin first. Then the circular stroke at the tail could not be drawn due to occlusion. (d) The wings and tail of the plane are incomplete. The participant drew upper strokes first and could not draw strokes at the bottom.

[1]The sphericity assumption was not met so the Greenhouse-Geisser correction was applied, the corrected degrees of freedom were shown.

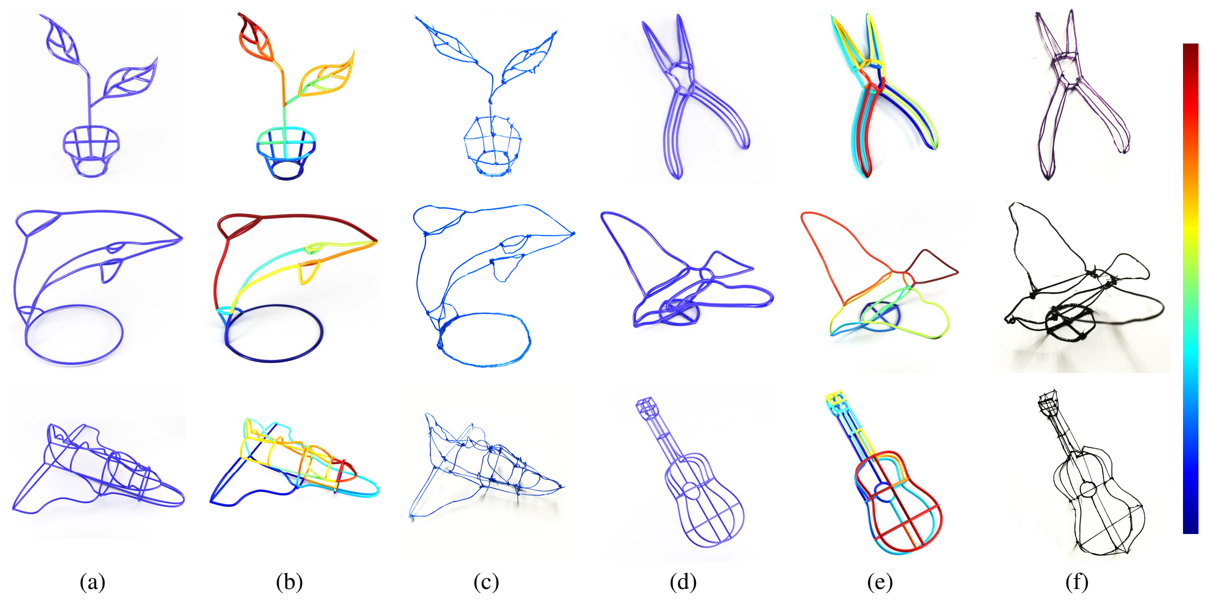

Figure 18. 3D wire object drawing results guided by our system. (a)(d) Input wire models to our 3D drawing optimization. (b)(e) Strokes sequence generated from 3D drawing optimization. The ascending stroke order is color coded from blue to red. (c)(f) Drawing results with MR guidance. All results are generated with the same parameter setting ($\lambda_s = 1, \lambda_p = 1, \lambda_v = 1, \lambda_d = 10, \lambda_r = 30$).

an ‘amplification’ effect. Luckily, the cube drawing practice helped the user to adapt to this: "The objects in the scene are somehow getting closer. I need to practice a bit to get used to this." Finally, we realized that the drawing speed also has some influence on the drawing quality, especially for curvy wires, since the extruded stroke requires some time to become solid: "The stroke can be stretched if I draw too fast."

Performance. Our MR-guided system runs in real-time (see supplemental video) based on the drawing sequence pre-computed from the optimization. The optimization takes less than 1 minute for a typical wire model with 50 curve segments (table with full statistics can be found in supplemental document). All the algorithms are implemented in C++ on a desktop PC with 3.4GHz CPU.

CONCLUSION AND FUTURE WORK

We have presented the first mixed reality system for guided 3D drawing of wire sculptures using a 3D extruder pen. Our system is based on novel 3D drawing principles concerning the drawability of the reference wire model and the simplicity of the drawing process. We use the proposed principles to guide the stroke sequence optimization which optimizes the formation of individual strokes as well as the drawing dependency among strokes. The optimized stroke sequence guarantees the structural soundness of the wire object during the drawing process and facilitates 3D wire object drawing for novice users. We further illustrate the stroke sequence using a mixed reality system to intuitively guide the 3D drawing process. The pen nip is also tracked in our system to help the users control the trajectory of individual strokes, and edit unsatisfactory strokes if needed. The evaluation and user study verify the effectiveness of our system.

Limitation and future work. While the structural symmetry is partly encoded in the DAG (i.e., symmetric strokes share similar dependency characteristics in the DAG and are likely to be drawn consecutively), shape symmetry is not specifically formulated in the optimization. From the user study of free drawing, we found that users prefer to draw symmetric strokes in strictly consecutive order for easier trajectory control (although this might cause problem for reachability). In the future, we would like to further involve the symmetry between strokes into the optimization.

The current system is optimized for drawing wire objects composed of smooth strokes. We are interested in extending the current system for guided drawing of surface-based objects. Then more sketching types need to be allowed for efficiently drawing surface patches, such as scribbling, other than tracing smooth strokes as in the current system.

We also plan to improve the stroke trajectory animation to better guide user’s drawing speed. The curvature information of individual strokes can be taken into account here.

Last but not least, we only allow the user to delete and replace strokes in the system. Based on our robust pen nib tracking, more stroke sketching and editing options, such as user guided stroke deformation can be added into the system to provide more degrees of freedom for creative design.

ACKNOWLEDGMENTS

We thank all the volunteers for participation, reviewers for helpful comments, and Garoe Dorta-Perez for proofreading. This work was supported in part by Innovation and Technology Fund (ITS/072/16), Microsoft Research Lab – Asia (MSRA), EPSRC Grant (EP/K02339X/1 and EP/M023281/1), State Intellectual Property Office National Patent Development Strategy (PS2016-005), Xiamen Overseas Scholar Project (XRS2016 314-10), and the Social Sciences Planning Projects of Fujian Province (FJ2016C095).

REFERENCES

1. 2015. *Master Your 3D Pen: Tips, Techniques, and Inspiration for 3D Designs*. 3DTotal Publishing.
2. 3DScribbler. 2016. http://www.scribbler3dpen.com/. (2016).
3. ARTookit. 2016. http://artoolkit.org. (2016).
4. Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: As-natural-as-possible Sketching System for Creating 3D Curve Models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. 151–160.
5. Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Found. Trends Hum.-Comput. Interact.* 8, 2-3 (2015), 73–272.
6. Thomas O. Boucher. 1996. *Computer Automation in Manufacturing*. Springer US.
7. Bruno De Araùjo, Géry Casiez, and Joaquim Jorge. 2012. Mockup builder: direct 3D modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012 (GI 2012)*. 173–180.
8. Fernando De Goes, Siome Goldenstein, Mathieu Desbrun, and Luiz Velho. 2011. Exoskeleton: Curve network abstraction for 3D shapes. *Computers & Graphics* 35, 1 (2011), 112–121.
9. Matthew Flagg and James M. Rehg. 2006. Projector-guided Painting. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. 235–244.
10. Hongbo Fu, Shizhe Zhou, Ligang Liu, and Niloy J. Mitra. 2011. Animated Construction of Line Drawings. *ACM Trans. Graph.* 30, 6 (2011), 133:1–133:10.
11. Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph.* 33, 4 (2014), 66:1–66:12.
12. Cindy Grimm and Pushkar Joshi. 2012. Just DrawIt: A 3D Sketching System. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*. 121–130.
13. Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. 2012. DuploTrack: A Real-time System for Authoring and Guiding Duplo Block Assembly. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 389–402.
14. Arthur Guptill and Susan Meyer. 1997. *Rendering in Pen and Ink*. Watson-Guptill Publications Inc.
15. Jiyang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. FrameFab: Robotic Fabrication of Frame Shapes. *ACM Trans. Graph.* 35, 6 (2016), to appear.
16. Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. 2013. The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. 183–192.
17. Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. 2015. WrapIt: Computer-assisted Crafting of Wire Wrapped Jewelry. *ACM Trans. Graph.* 34, 6 (2015), 221:1–221:8.
18. D. Keefe, R. Zeleznik, and D. Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 1067–1081.
19. Yongkwan Kim and Seok-Hyung Bae. 2016. SketchingWithHands: 3D Sketching Handheld Products with First-Person Hand Posture. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. 797–808.
20. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680.
21. Manfred Lau, Greg Saul, Jun Mitani, and Takeo Igarashi. 2010. Modeling-in-context: User Design of Complementary Objects with a Single Photo. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. 17–24.
22. Jeremy Laviole and Martin Hachet. 2012. Spatial Augmented Reality for Physical Drawing. In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct Proceedings '12)*. 9–10.
23. Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. ShadowDraw: Real-time User Guidance for Freehand Drawing. *ACM Trans. Graph.* 30, 4 (2011), 27:1–27:10.
24. Eder Miguel, Mathias Leputre, and Bernd Bickel. 2016. Computational Design of Stable Planar-Rod Structures. *ACM Trans. Graph.* 35, 6 (2016), to appear.
25. Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 273–280.
26. Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. 599–606.
27. Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 887–896.

28. Huaishu Peng, Amit Zoran, and François V. Guimbretière. 2015. D-Coil: A Hands-on Approach to Digital 3D Models Design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 1807–1815.

29. Alec Rivers, Andrew Adams, and Frédo Durand. 2012. Sculpting by numbers. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 157.

30. Thijs Roumen, Bastian Kruck, Tobias Dürschmid, Tobias Nack, and Patrick Baudisch. 2016. Mobile Fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. 3–14.

31. Ryan Schmidt, Azam Khan, Karan Singh, and Gord Kurtenbach. 2009. Analytic Drawing of 3D Scaffolds. *ACM Trans. Graph.* 28, 5 (2009), 149:1–149:10.

32. Roy Shilkrot, Pattie Maes, Joseph A. Paradiso, and Amit Zoran. 2015. Augmented Airbrush for Computer Aided Painting (CAP). *ACM Trans. Graph.* 34, 2 (2015), 19:1–19:11.

33. Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. 2006. ModelCraft: Capturing Freehand Annotations and Edits on Physical 3D Models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. 13–22.

34. Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. 2014. MixFab: A Mixed-reality Environment for Personal Fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3855–3864.

35. Gerold Wesche and Hans-Peter Seidel. 2001. FreeDrawer: A Free-form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 167–174.

36. Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2011. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. 69–72.

37. Rundong Wu, Huaishu Peng, François Guimbretière, and Steve Marschner. 2016. Printing Arbitrary Meshes with a 5DOF Wireframe Printer. *ACM Trans. Graph.* 35, 4 (2016), 101:1–101:9.

38. Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. *ACM Trans. Graph.* 33, 4 (2014), 131:1–131:13.

39. Amit Zoran and Joseph A. Paradiso. 2013. FreeD: A Freehand Digital Sculpting Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. 2613–2616.

40. Amit Zoran, Roy Shilkrot, Pragun Goyal, Pattie Maes, and Joseph A. Paradiso. 2014a. The Wise Chisel: The Rise of the Smart Handheld Tool. *IEEE Pervasive Computing* 13, 3 (2014), 48–57.

41. Amit Zoran, Roy Shilkrot, Suranga Nanyakkara, and Joseph Paradiso. 2014b. The Hybrid Artisans: A Case Study in Smart Tools. *ACM Trans. Comput.-Hum. Interact.* 21, 3 (2014), 15:1–15:29.