Phasking on Paper: Accessing a Continuum of PHysically Assisted SKetchING

Soheil Kianzad
Department of Computer Science, University of British Columbia
Vancouver, Canada
skianzad@cs.ubc.ca

Yuxiang Huang
Department of Computer Eng, University of British Columbia
Vancouver, Canada
huang@ece.ubc.ca

Robert Xiao
Department of Computer Science, University of British Columbia
Vancouver, Canada
brx@cs.ubc.ca

Karon E MacLean
Department of Computer Science, University of British Columbia
Vancouver, Canada
maclean@cs.ubc.ca

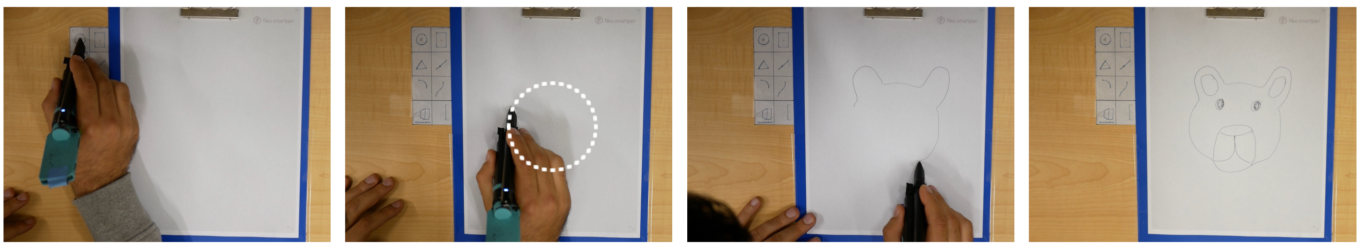


Figure 1: The steps of a phasking interaction. (a) The user selects the "circle" tool from the Phasking tool palette, (b) then selects the centre and one point on the circle to establish a circular *bring* constraint (dotted line). (c) The Phasking pen actively *brings* the user along the circular path with its ball-drive motor, but the user can modify *control sharing* by applying pressure to the pen (Figure 10b). This causes the system to scale down its constraint force, allowing the user to diverge from the path. (d) Phasking supports passive constraints as well as fully unconstrained drawing, enabling the user to quickly sketch out a cartoon character.

ABSTRACT

When sketching, we must choose between paper (expressive ease, ruler and eraser) and computational assistance (parametric support, a digital record). PHysically Assisted SKetching provides both, with a pen that displays force constraints with which the sketcher interacts as they draw on paper. Phasking provides passive, "bound" constraints (like a ruler); or actively "brings" the sketcher along a commanded path (*e.g.*, a curve), which they can violate for creative variation. The sketcher modulates constraint strength (control sharing) by bearing down on the pen-tip. Phasking requires untethered, graded force-feedback, achieved by modifying a ballpoint drive that generates force through rolling surface contact. To understand phasking's viability, we implemented its interaction concepts, related them to sketching tasks and measured device performance. We assessed the experience of 10 sketchers, who could understand, use and delight in phasking, and who valued its control-sharing and digital twinning for productivity, creative control and learning to draw.

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
https://doi.org/10.1145/3313831.3376134

Author Keywords

Haptics; force-feedback; sketching; stylus interaction; computer aided drawing; shared control drawing

CCS Concepts

•Human-centered computing → Haptic devices; Systems and tools for interaction design; •Hardware → Haptic devices;

INTRODUCTION

From scribbles to detailed, elaborated productions, sketching is both intellectual play and can help us form, develop and communicate our thoughts, a key part of conceptualization. Pen-and-paper sketching is direct, improvisational, expressive, resists distraction, and may promote deeper cognitive processing [23]. The freedom and functional control afforded by physical drawing is unmatched in electronic media; but paper sketching lacks digital enhancements, and it is laborious to move fluidly between paper and digital media.

Meanwhile, freehand drawing is poorly supported in graphical and CAD (Computer Aided Design) environments, as evinced by many professionals' preference for paper. Capturing the subtlety and nuances of physical drawing and painting remains elusive [37].

Active force feedback is an intriguing approach to supporting on-paper drawing, with different opportunities for expert and novice sketchers. But active force feedback (rather than brakes, which cannot actively guide, or tactile vibrations, which cannot even constrain) usually entails a grounded device [31] with a fixed and impractically limited workspace, often costly. What if a user wants to roam with their physical drawing support, accessing guidance and constraints for everyday drawing on arbitrary media, or to support big strokes on big surfaces?

The goal of this project is explore the use of active force feedback in paper sketching, to provide a user with the digital support they need in the moment while maintaining their originality, authorship, and continuity of expression.

Physically Assisted SKetching with Variable Control

Phasking enables a user to access drawing supports in a continuum from full physical guidance to expressive freehand sketching, on their media of choice (Figure 2). It is free-roaming (untethered and portable); and our demonstration prototype is constructed of low-cost commodity components and DIY construction. It is based on two key interaction concepts.

Constraints: Phasking's bring/bound system captures and extends the range of assistance explored in previous work, by accessing an interactive, constructable virtual environment (VE). The user can place structures in the VE to constrain their movements both passively (*bound*) and actively (*bring)*, and can interact with constraints and the sketch itself.

Control sharing: Users can fluidly move between levels of assistance, simply by bearing down on the pen tip. Phasking differs from adaptive force feedback [3], where the system chooses the degree of assistance based on user performance. Phasking gives this choice to the user.

Usage Scenarios

A wide spectrum of users sketch, in many contexts. We consider situations where phasking will be valued, to prioritize feature development, and consider them in our evaluation.

Rapid technical sketching: Professional architects, engineers and other designers sketch copiously, rely on it for conceptualization and communication, and move between paper and digital media. Many experts value drawing assistance when on paper, whether to construct a perfect circle or perspective, as evinced by their heavy use of physical guides. Some find physical tools cumbersome and "in the way", *e.g.*, wanting to draw on both sides of a ruler. Large-scale drawing (*e.g.*, during public communication), is a situation where even skilled sketchers may have difficulty with alignment and clean curves. Finally, professionals spend a long time *learning* to draw. Phasking's large workspace, portability and digital-twinning capabilities could assist in all these contexts, both manually and by off-loading some cognitive demand.

[Re]Learning to sketch: Learners may be children, hobbyists or recovering stroke patients. All may need assistance in drawing a straight line, getting proportions right for an animal or a face, or figuring out perspective. There is evidence that CAD can help creative self-efficacy [32]. Learners' needs might range from manual control (*e.g.*, drawing a circle), to the cognitive challenge of using proportion. Learning is best accomplished on paper, with its friction, focus and room to spread out, but would benefit from computational supports. Phasking's constraints can assist learners and patients in reinforcing motor programs, and advanced skills like perspective drawing. Once learning is achieved, they might continue to phask in more expert ways, or no longer need it.

Artistic 2D Sculpting: Creative expression is inspired by constraints [18]. Physical constraints that can be stretched and violated stimulate a resonant collaboration between user and system [18, 33]. Phasking is founded on collaborative control-sharing. Artistic sketchers can follow a basic shape, or creatively modify it, by altering control authority. To draw a face instead of that circle, they press down to take control to form an ear or a sketchy line of hair. Or, they can pull and bounce off an active node to draw sweeping trapeze-like trajectories, bound to the node with an invisible elastic string.

Contributions

1. *A conceptual framework for phasking* which highlights fluid transition of control sharing from assisted to freehand drawing, making use of both bound and bring constraints.
2. *A force-feedback digital manipulative* capable of implementing this framework. We made major mechatronic, ergonomic and control extensions to a previous ballpoint drive display to enable screen-free, shared control phasking.

BACKGROUND

Our work is founded in manual and computer-aided drawing practices, virtual and augmented environment creation and manipulation, haptic force feedback and that field's knowledge of control-sharing and past frameworks for sketching support.

Figure 2 arrays the mediums and examples mentioned here on the spectrum between fully freehand and fully computer-aided drawing. Vertically, this figure highlights how *control sharing* is related to type of drawing. Most examples occupy only one point in this space, whereas physically assisted sketching, as presented here, can theoretically cover all of it.

Non-Haptic Assistance of Digital and Manual Drawing

Professional CAD tools are increasingly accessible, online and learnable: basic ideas of parametric drawing have high penetration for even minimal expertise. But despite many conveniences (pen type/color, copy/paste/undo) the experience is still fundamentally different than pen-and-paper drawing due to tactility and limited canvas.

Graphical drawing systems can provide visual corrective feedback for drawing or give stroke suggestions, *e.g.*, iCanDraw to sketch a human face from a source image [7], and ShadowDraw for high-level arbitrary objects [17]. Tthese systems typically restrict users to graphical screens.

Paper-oriented digital styli, such as Anoto and Neo smartpens, feature realtime digital capture of handwriting and translation to digital task, requiring use of watermarked paper (*e.g.*, pre-printed with microdots) [13, 14]. While capturing natural drawing, this approach cannot offer added support *during* sketching. Our conceptual prototype incorporates a Neo Smartpen with its pen-tip vision system and watermarked paper as a convenient way to mockup position localization.

For *visual* guidance, PenLight [34] combines an Anoto pen with a miniature projector to add information to pen and paper interaction, but faces a technical barrier of image stability In virtual reality, Nomoto et al presents a "corrected" sketch which encourages the user to configure their own hands appropriately for drawing a shape [26]. These are promising approaches but do not offer physical constraints. In phasking, real forces on the hand – in the real world, not a VR headset – convey drawing suggestions.

Force-Based Support of Screen-Based Constraints

Force-feedback devices can generally support constraint systems; here we give examples of the different ways they tend to be *tethered* and/or *non-portable*.

Grounded desktop devices such as the Phantom Omni have long been used to support handwriting training tool and rehabilitation, *e.g.*, [24, 8] Another approach uses a Cobot mechanism (a passive steered wheel which blocks motion in certain directions but cannot drive. In mouse form, the Robotic Touchscreen Totem presents cobot-style boundaries and paths, and is untethered; but has no active forces – meaning no active constraints or graded forces [29].

The MOTORE [2] device (a 3-wheeled handheld mouse-bot) and Cellulo [27] (3-ball omni-directional drive with Anoto pen for localization) supply substantive forces for rehab and educational applications. Technically untethered, its inherent size, weight and 3-wheeled base makes MOTORE non-portable, whereas Cellulo is ergonomically unsuited for drawing due both to size and a large three-wheeled base.

Haptic Support of Pen-and-Paper Sketching

Passive Constraints – A ballpoint stylus able to impose passive constraints by *constricting* a rolling contact ball by means of electromagnetic or mechanical brakes has some similarity to our ballpoint drive and was used to render roughness on 3D objects, but cannot provide active forces [6]. Comp*Pass [25] offers a semi-active solution using DC motors; however, a user is not actively involved in sketching. With I-Draw, a cobot-type drawing assist for passive constraints [10], the authors "explore the seamless switching between guided and freehand modes," as do we.

Active Guidance – Muscle-Plotter generates force feedback to the hand by electrically stimulating the user's own muscles [22] While creatively satisfying the criteria of free-roaming, it has drawbacks of 1.5 DOF, and a potential for temporal adaptation by muscles [15][39].

dePENd [38] exploits the ferromagnetic property of pen balltips to assist sketching by providing directional force feedback on regular pen and paper interaction. The main drawback of magnet-based haptic assisted sketching devices [21] is the tradeoff between backdriveability and perceptible force levels. Increasing magnetic coupling provides higher forces, but draws the pen tightly to the interaction surface (higher normal force) and makes it difficult to move freely.

I-Draw, dePENd, and Muscle-Plotter each demonstrate a concept of guided drawing (using actuation to turn a user's hand into a computer-guided drawing implement in a screen-free context), one passive and the other active. They are a departure point for the contributions described here.

Using Haptics to Facilitate User-System Control Sharing

Control-sharing with haptic systems has been utilized for physical therapy [19] and handwriting control [36]. In driving, haptic shared control can improve speed and accuracy of human / system collaborative tasks, and lower the need for visual involvement control effort [12, 1] It has also been used to support expressive drawing on a screen, *e.g.*, Snibbe's Dynasculpt and GridDraw [33]. While I-Draw [10] is framed in moving between freehand and supported drawing, its passive nature does not provide an ideal mechanism for doing so.

We have drawn from these functional and expressive approaches to form our own control-sharing, which prioritizes simplicity and intuitiveness in modulating control authority in instances where users need a collaboration rather than a binary choice. The Phasking Pen's capabilities support this.

Frameworks

Steimle et al's framework of non-sketching pen-and-paper interactions separates conceptual activities (annotating, linking, tagging) from core interactions (inking, clicking, combining, associating) [35]. While its domain differs, we are inspired by its approach in our own support framework.

I-Draw presents an initial framework for passive guidance, of interaction primitives allocated between physical (guided and freehand drawing) and digital (digital manipulation) spaces [10]. We re-organize and extend it with the capacities afforded by active haptic guidance.

PHASKING FRAMEWORK

The added capabilities of a fully force-performant but free-roam, screen-free device has several implications. First, the availability of active, omnidirectional forces in a handheld format permit fundamental changes in interaction, notably active guidance and control sharing. This physical support can work in both graphical and screen-free contexts, widening scope and altering how digital-physical transitions can occur. Finally, the active force's scalability means that constraints and guidance can be modulated, from hard to soft. Together, these necessitate a deep revision and extension beyond past conceptual framings (*e.g.*, [10, 35]).

Like [35], our framework articulates *conceptual activities* that users need to do, elaborated in (I) below; leading to *core interactions* that support them (II): bound and bring-type constraints and variable control authority (constraint hardness).

Figure 2 shows how bound/bring constraints interact with shared control in the phasking framework.

I. Conceptual Activities

We articulate the foundational activities which our framework needed to support, as elements that mediate a dialogue between user and system in Table 1, a potentially extendable list. These emerged from our observation of users' expectations formed through interacting with conventional tools, as well as consideration of the basic operations of freehand paper drawing, CAD and virtual environments.

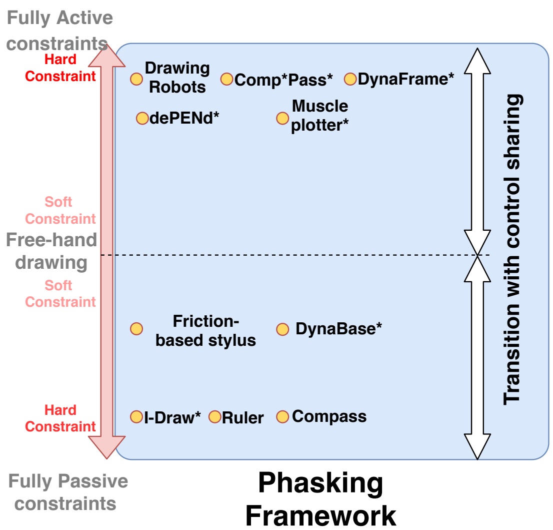


Figure 2: *A one-dimensional model of sketching control authority.* The y-axis denotes strength of a given system's constraint, *active* up / *passive* down. The x-axis has no meaning. Existing types of assistance are located approximately in this space, both generic (*e.g.*, a physical ruler) or published works (denoted with *, see Related Work). Phasking can fluidly access all points on this axis, by bearing down on the pen-tip while drawing.

II. Core Interaction Concepts

These concepts demonstrate how phasking's key conceptual activities are supported. We use a constraint-based virtual environment (VE) which a user constructs then sketches within, with tools created by drawing a palette on the paper's margin.

(a) Constraints – Bounding and Bringing

Constraints can be expressed as a gain on an error function (of position, velocity or other parameters): $u = K(x_{des} - x_{act})$. With active force assistance, phasking constraints can passively *bound*, or actively *bring* (Figure 3). They do both to varying degrees (Control Sharing concept, below).

Bound – Movement is free up to the boundary, then constrained. A binary boundary (no shared control) could be implemented with a passive force device, *e.g.*, a brake, because it just prevents the user from going somewhere. Examples of *bound* constraints include one-sided walls, and path constraints which the user can traverse at will: the constraint blocks path departure, but allows free movement along it.

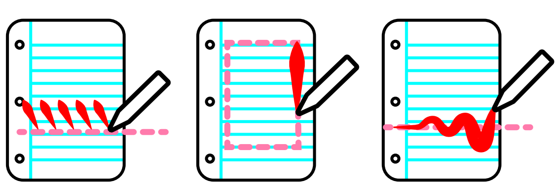


Figure 3: The framework's (a) *bound* and (b) *bring* constraints, and (c) the concept of *control sharing*, where the user can diverge from a guiding line by bearing down on the pen.

Table 1: Core conceptual activities of the phasking framework.

Activity	Description
Free-draw marks	Create marks manually and at will, optionally within user-set and system-maintained boundaries.
Create objects	Form basic shapes on command (*e.g.*, parametrically specified). Produced objects may conform perfectly to the digital guidance, or the user can overcome guidance to construct personalized or expressive variations.
Place & arrange elements	Receive assistance as to where, how large and what angle; *e.g.*, perspective drawing, or sizing different regions of a multi-part sketch
Interact with active constraints	Set up constraints (*e.g.*, attractive nodes, or lines and curves to push/pull against) for modulated creative control in variably-guided drawing.

Bring – A force field draws the user in a particular direction or rate, and requires an active force feedback device; it always entails active guidance[1]. Examples of *bring* constraints include point magnetic attraction or repulsion (snap); and spatiotemporal and temporal trajectories, in which the user is guided to traverse a path in time and space respectively.

As with a ruler, passive VE elements are only felt when the drawing tool touches them. Active elements, *bound* or *bring*, can be felt at a distance, as a force field.

(b) Control Sharing

In phasking, what's shared is control authority ("who gets to drive"), under continuous user control (*e.g.*, the pressure with which the drawing tool is squeezed or pushed into the drawing surface). The constraint can vary between absolute and soft – a suggestion or a jumping-off place, *e.g.*, if one wishes to draw a wiggly line along a path (Figure 3c). This scaling is available for both *bound* and *bring* constraints.

Control sharing can be implemented simply by changing the control gain on the error between pen and constraint. A wall can be softened, and an actively guided geometry (such as an oval) can be sketched into something more expressive and detailed, like a face. More complex implementations are available to address system stability issues [11, 1].

(c) Tool Selection

Like other digitally assisted drawing systems, phasking is modal. We deliver the function of tool selection with a paper tool palette with hand-drawn (and extemporaneously creatable) icons that the pen's vision system can recognize (see *System*). Because interactions are brief, tool selection also supplies modal awareness, together with the device's physical response.

[1] *Bring* contrasts with what is called guidance (but is passive) in some related work (*e.g.*, I-Draw), in which a passive mechanism such as a a Cobot [5] or brake restricts movement in some direction.

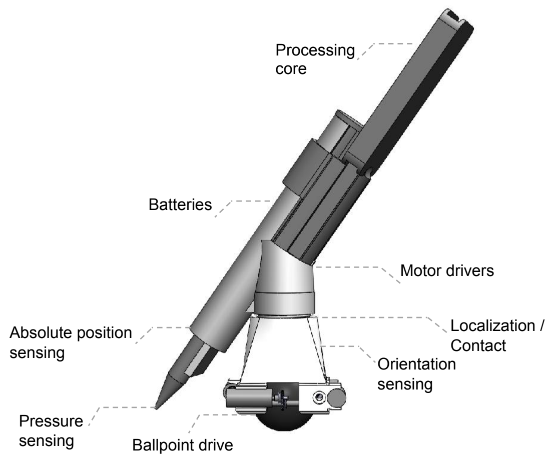


Figure 4: The Phasking Pen: an untethered ballpoint drive produces forces by driving a contact ball over a surface. Friction between the ball and surface prevent slip, and provide a "ground" back to the user's hand.

(d) Constructing Constraint Environments

In free-hand drawing, a user sets a passive boundary with a tool, like placing a physical ruler to help draw a straight line or set a CAD drawing plane.

Phasking constructs and tracks constraints through a virtual environment: the user draws the environment within which they then operate. For example, the user places a virtual bound (*e.g.*, a line, circle or channel) or an attraction point for a *bound* or *bring* constraint, respectively. The VE is portrayed to the user through both the visible marks on the drawing medium, and what they feel.

Constraints support expressive drawing, and force feedback has been used for this [33]. But when working screen-free, constraint creation *is required* to access assistance. Here, our VE is a basic functional implementation, but the approach opens other design spaces as well.

SYSTEM

We implemented the Phasking Pen's mechatronics, system architecture and controls, and phasking primitives to assess the concept's feasibility and usefulness. The device reported here significantly extends a previous basic demonstration of the ballpoint drive mechanism as a 2D force-feedback display [16], as required for this application.

and paper-based operation. Phasking demonstrations are original with this paper. Here, we overview the full system for the reader's benefit, but focus on novel or modified elements.

Phasking Pen Mechatronics

Mechanism: Untethered 2D Forces Via Ballpoint Drive

In the ballpoint drive [16], pairs of opposing motors drive a surface-contact ball to create directional force-feedback, generated between ball and arbitrary two-dimensional surface

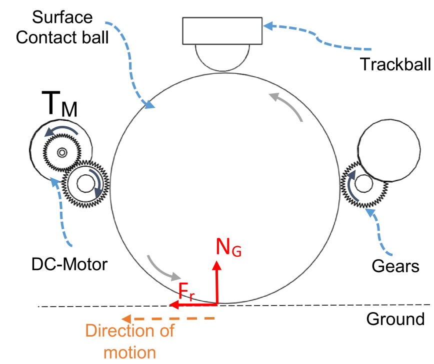


Figure 5: Ballpoint drive and sensing in 2D. Motors generated torques T_m are transmitted to the surface contact ball through the gears causing the ball to roll over the surface. The friction force F_r between the contact ball and the surface produces motion in x-y direction.

(Figure 5). We completely re-engineered the ballpoint drive mechanism to improve backdrivability (freedom of unpowered motion), reducing passive impedance by 51%. We also reworked the gear drive to achieve higher force without slip, significantly reducing vibration and skidding. We customized the drive train with low-cost commodity micro gears, motors and a surface contact ball. To achieve required backdriveability and power transfer, we iterated component configuration, size and material properties (Figure 6).

Gear drive optimization – Ensuring non-slip coupling between motors and surface contact ball can add impedance to the drive train, which then degrades rolling free-ness when motors are not actuated. We found a solution by matching contact-ball material properties with cog size. A 1-inch diameter rubber ball with tensile strength of 144 MPa, with metal gears with diametral pitch of 187 teeth/inch [9] gave the best results.

Drive motor coordination – We can achieve optimal control over the surface contact ball with four motors (as opposed to two drivers and two passive castors; or a 3-point contact which cannot reach the same control space). To generate a rolling movement, each opposing motor pair work together to apply a balanced torque to the ball. While it is possible to electrically connect the opposing pairs, we found out that for fast movements when the motors are not in active mode, paired motors work as a generator and produce back-emf voltages which pass through the opposing pair. To avoid the consequent resistance, we drove each motor separately.

Sensing: position, orientation and pressure

The two primary sensing needs for phasking operations are (a) localization of device on the interaction surface, and (b) internal ball motion and stylus orientation for closed-loop control on position and velocity, to generate desired forces.

Position for external localization and internal control –

There are many possible approaches to external localization of contact point on the interaction surface, depending on applica-

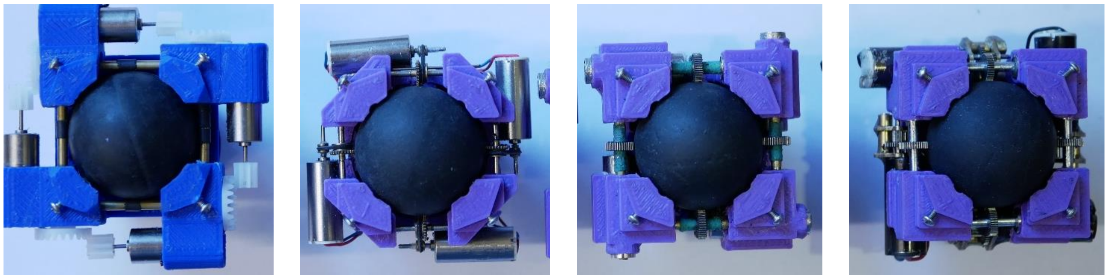


Figure 6: Phasking Pen ballpoint Drive design iterations. From left (earliest): (a) Early version with plastic gear between motors and rollers, and rubber connecting ball between roller and surface contact ball. (b) Pulley belt connection between motors and rollers, and clock gears with micro cogs between roller and the contact ball. (c) Metal gears between motors and roller, for a lower gear ratio of 1:1. (d) Similar mechanism with a customized higher gear ratio of 4:1.

tion setting, priorities and constraints. We exploited existing technology of digital pens and watermarked papers to obtain absolute position sensing. An embedded camera (here, a Neo SmartPen [14]) decodes optical microdot patterns on watermarked paper to determine absolute position of the stylus to an accuracy of 0.1mm. This enables accurate sketches and useful interactions such as the tool palette.

Internally, a micro-trackball senses the motion of the larger surface contact ball in each rolling direction and sends the pulses to the device controller.

Orientation – Stylus orientation corrects position estimates generated by the above methods. A twist of the user's hand impacts the contact relation between rolling ball and interaction surface: when non-vertical, the ball's contact point does not coincide with the stylus axis. The global localization signal can also suffer from low update rate, gaps or spatial inaccuracy.

We used a low-cost orientation sensor (Bosch BNO055) which fuses accelerometer, magnetometer and gyroscope data to supply Euler roll, pitch and yaw [4].

Pressure – The Neo Smartpen supplies pen tip pressure [0-255], which we use for control sharing. During early prototyping, we tested several locations for pressure sensing. We found out that users have most control over pressure application when the pressure sensor is located at the pen tip, rather than behind the trackball or under the pen finger's grip.

Battery

The early version of the ballpoint drive [16] was electrically tethered to external power and communications. The present prototype uses two 3.7V Ultrafire 14500 rechargeable batteries in parallel, capable of approximately one hour of free-roam performance. The most significant power draw is from the CPU due to I/O interrupts related to orientation and localization sensing. A future version using an embedded CPU will be significantly more efficient.

System Architecture and Control

Computing processor, motor drive and communications – The primary processing unit (Raspberry-Pi Zero W, or RPi) takes sensor data, updates a state model, then computes proportional-derivative (PD) control commands and sends them to two Pololu DRV8835 dual-motor driver carriers (one per axis).

We sample micro-trackball velocity, integrated to get position at 5 kHz (RPi external interrupts), and BNO055 for orientation sensing at 100 Hz. The Neo Smartpen sends its data (x,y position and pen-tip pressure) to the RPi controller via Bluetooth Low-Energy at 100Hz. We built a custom Linux driver for the Neo's BLE protocol to reduce latency and enable custom features (e.g. on-demand beeping).

State model and closed-loop control – A 1KHz control loop checks for a command, samples internal position and orientation, receives x,y and pressure data from the Neo (100 Hz), and optionally sends it to the monitor. Control then branches:

Free mode	No command is running; wait for next iteration.
Command mode	If new command registered, collect command parameters parameters (position taps). Then: ▷ Update command target reference. ▷ Adjust absolute contact position estimate based on pen orientation, with internal position change. ▷ Compute motor command command using a PD controller on the error signal. ▷ Output motor command to PWM motor.

Implemented Drawing Support Features

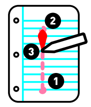
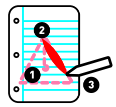
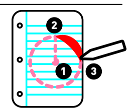
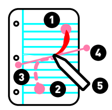
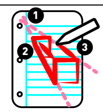
2D CAD software implementation

Our new untethered design required a custom, lightweight CAD platform for the RPi, and custom low-latency hardware drivers for the ballpoint drive and digital pen.

To implement the Phasking Pen's drawing functions, we developed a simple 2D CAD software system using Python and the PyQt4 library (a Python interface for Qt, a popular cross-platform graphics library) [20], which runs on the pen's RPi controller. With this software, the user can either free-draw, or access primitive CAD functions by clicking on a paper tool palette (described below).

The CAD software also implements a graphical user interface view (referred to as the *GUI monitor*), which can be displayed on a screen connected by HDMI cable to the onboard RPi for debugging. The GUI monitor view shows real-time updates of the user's drawing, and provides additional functions such

Table 2: Phasking primitive descriptions. Each operation begins by touching the corresponding icon on the tool palette.

Primitive	Description	Illustration
Line, Ruler	(1) Touch end point (2) Touch starting point (3) *Ruler*: draw along invisible barrier between the two points *triangleright Line*: Phasking Pen *brings* to end point	
Triangle, Arc	(1-3) Touch three points to define Triangle or Arc ▷ Phasking Pen *brings* across triangle edges, or arc.	
Circle, Rectangle	(1) Touch center (2-3) Touch radius (circle) or top left corner (rectangle) ▷ Phasking Pen *brings* to endpoint.	
Bezier spline	(1-4) Touch at least four control points ▷ A cubic (4 points) Bezier curve is defined (5) Touch curve ▷ Phasking Pen commands motor velocity according to the tangent line to the curve at each point.	
Perspective function	(1) Touch vanishing point (2-n) define any geometry, *e.g.*, *Rectangle* (center, corner) ▷ Phasking Pen draws the object (*e.g.*, *Rectangle*) in perspective	

as saving the digitized drawing locally and changing the color and thickness of the pen.

(a) Bound/Bring Constraints for 2D Geometry Construction

We implemented eight phasking primitives (Table 2): seven to construct basic shapes or constraints, and a perspective function for use with other primitives. Each uses *bound*/*bring* constraints, with force guidance modulable through control sharing (below). Jointly, these primitives implement all of the core conceptual activities of Table 1.

(b) Sharing Control Authority

When drawing, the user can start with one of these primitives and deviate from the pen's guide by applying a small force, to sketch more complex shapes (Figure 1).

(c) Tool Switching: paper-based tool palette

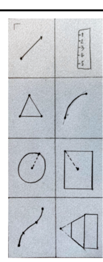
Function	Palette	Function
line (*bring*)		ruler
triangle		arc
circle		rectangle
B-spline		perspective

Figure 7: Paper tool palette, which can be hand-drawn and customized, or printed on a full sheet or slip of paper.

Phasking requires extendable access to the drawing primitives. We used a paper palette as a simple physical ac2cess point; the user can draw tools on the sheet, selects a command by tapping on a box, then taps on the drawing to define parameters (Table 2). Table 7 shows the watermark-paper implementation. The tool palette has an added advantage of logging user commands, as one route to saving geometry; *e.g.*, for copy/paste on paper, or a screen-based reconstruction.

EVALUATION

Our evaluation objectives focused on conceptual viability: we needed to know whether the ballpoint drive approach could perform well enough to support phasking operations, and to get insight into users' experience of phasking.

We tested force-feedback standards of force and position control and disturbance rejection. These address whether the novel drive can localize itself with its dual sensing system even as the user's hand rotates the pen's axis (pitch and yaw), follow a commanded path, provide a usefully large commanded impedance, and reject disturbances, all fast and smoothly enough for at least moderate-paced drawing. We sought performance that would let us try phasking out, an assessment to be rendered in part by our user study. These tests also set a benchmark for comparison with future progress.

Performance Characterization

Prior to involving participants, we evaluated the Phasking Pen's mechanical and control performance.

Test 1 – Force generation: We measured passive and active 2D forces with a BOSE ElectroForce TestBench®, which held the Phasking Pen with two arms (Figure 8a).

Passive force, step response – Low mechanical impedance is crucial for control sharing interactions. We recorded Phasking Pen's resistive force while unpowered, while one BOSE arm imposed an 8mm sin-wave position displacement to the Phasking Pen at 0.5 Hz for 5 periods. We found an impedance of 37.5 Ns/m, computed by dividing the recorded value of mechanical resistance force over the speed. This is approximately half that measured for the previously reported version of the ballpoint drive (77 Ns/m) [16].

Active force, step response – We measured the force the ballpoint drive produced in response to pulsed drive input ($\tilde{0}.2$Hz), while held isometrically between two measuring load cells, one on each arm (Figure 8b. The pen generated up to 1N in continuous force (close to peak output) without slip between contact ball and gears, after which the drive gears slipped and the ball started spinning.

Active force, sine response – As seen in Figure 8c, generated force closely followed a continuous sine pulsewidth modulated (PWM) voltage input. There was minor nonlinearity at higher voltages, as excitation approached motor saturation.

Test 2 – Position control: We required the Phasking Pen to follow a sine wave trajectory using only the internal microtrackball (relative position sensing), measured with the setup of Figure 9. This test demonstrates the ballpoint drive's capacity to achieve agile, omnidirectional control in the absence

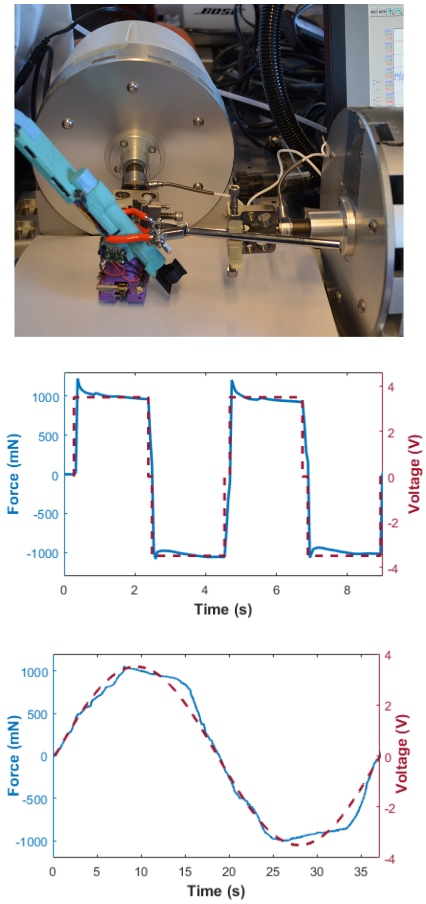


Figure 8: Force generation performance. (a) BOSE test setup. (b) Max output force response to PWM voltage pulses. (c) Force-tracking response to a slow sinusoid of PWM excitation.

of disturbances. We chose a sine wave target to capture a full range of movement in a 2D plane, and used a steady roll offset of 10°. This means that for the surface contact ball to, *e.g.*, move along an x-trajectory, it would need to adjust x,y motor commands using orientation data rather than simply turn the x-motor on and the y-motor off. Here, the task is to follow an x-y sinusoid. Due to the low impedance of the second trackball, the contact ball rolls at full speed (20mm/second) which slightly reduces the accuracy of the controller.

The Phasking Pen followed a 4.0cm peak-to-peak sine trajectory with 20% initial error and 40.0cm path length, using only the trackball (relative position sensing) corrected by orientation data for position control, with an error (mean squared distance to reference) of 6.78 mm (std 4.88mm), or 1.4%.

Test 3 - Disturbance Rejection: The controller needs to compensate when the stylus is twisted due to movements of the user's hand, adhering to a straight x trajectory with no y deviation. To do this, it just continually change motor velocity to maintain contact ball motion in the x direction alone.

Figure 9b shows the system following a line as we sweep through roll angles of 3-38°(a range observed for handheld usage) with mean square error 6.89 mm (std 2.29mm) 1.3%.

Introducing a pitch angle of 0-25°can reduce the speed in the x direction up to 10%, as the surface contact ball rolls on a smaller circle. This can be compensated either by multiplying a $cos(\theta)$ coefficient [16], or using absolute position sensing.

Test 4 - Sensor fusion: By using the absolute position sensing from the NeoPen, the controller can compensate for the offset and achieve higher precision. Figure 10 shows (left) the performance of the device in drawing a circle with a hard *bring* constraint, while a user gently holds the Phasking Pen; and (right) with control sharing activated while the user authors creative modifications to the basis circle.

User Evaluation

We conducted two user studies: (a) A performance assessment with novices (N=7) to assess usability and impact of core phasking interactions on normal users. We measured users' deviation from a predefined trajectory (error) and investigated

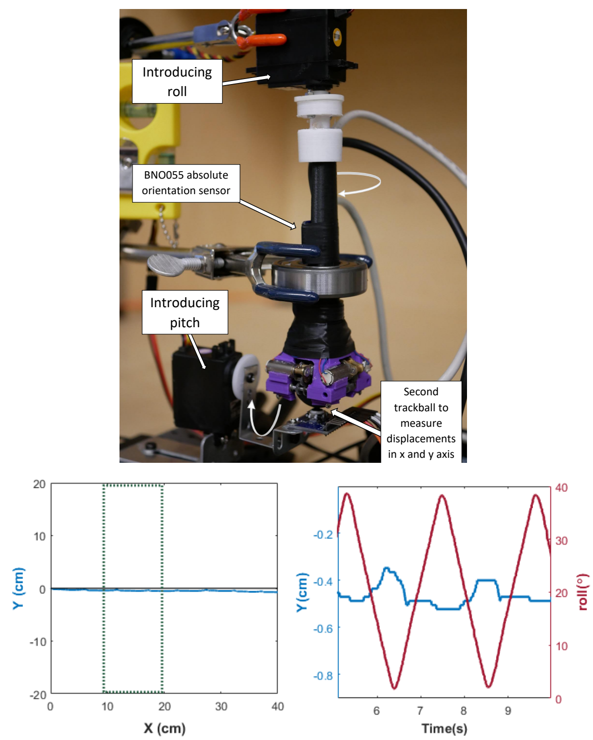


Figure 9: Position and disturbance tracking. (a) Test setup. The Phasking Pen's shaft is held in a vice, with roll and pitch disturbances applied at the top of the shaft and through the contact ball, respectively. The onboard processor controls the contact ball trajectory using orientation (roll/pitch/yaw) and internal ball motion. An external trackball measures ball movement relative to a global reference. To assess performance, we compare error between command and externally measured trajectories. (b) Test 3 (Disturbance Rejection) results shows the system's response to rapid yaw (0.5Hz, 35° peak-to-peak over 22 seconds), mimicking significant wrist rotation.

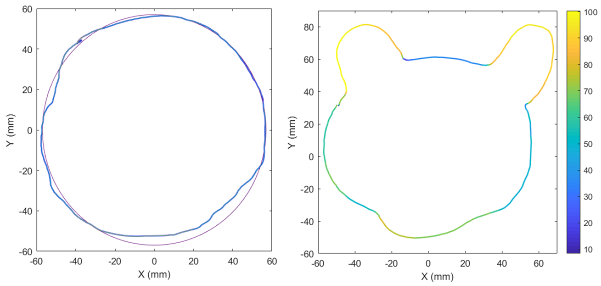


Figure 10: Human's aided circle following. (Left) System control: with control-sharing off, we fuse absolute and local position sensing to guide user N1 in drawing a circle. The deviation is a result of the pen rotating in N1's hand. (Right) With control-sharing on, a user violates a *bring* circle guide to draw a bear's face by pressing down on the pen. The color scale indicates applied pressure (yellow at 100% user control, blue when user has relaxed and is letting the system drive).

Table 3: Evaluation tasks, by execution and complexity.

Concept	Task
***Bring* constraints**	[1] Draw a straight line [2] Draw a rectangle [3] Draw a rectangle in perspective (novices) [4] Draw a circle (novices)
***Bound* constraints**	[5] Draw diagonal lines meeting an invisible barrier (coarse cross-hatch on a line)
Shared control	[6] Draw a sine wave as Phasking Pen guides along a straight line (pulling towards the guide) [7] Draw a sine wave as Phasking Pen sets an invisible line barrier at the center of the sine wave (resisting their crossing of it).

their pressure profile to better understand the shared control (SC) concept in practice. (b) An interview evaluation with domain experts (N=3, architects who sketch in their professional work) to assess value, fit to needs and potential for a enriching user experience, based on a functional conceptual prototype.

Procedure, both studies: We collected profile information on how participants used sketching in their work, and their personal attitudes to it. After a system familiarization session, we asked participants to perform several tasks (Table 3) with the Neo Smartpen on its own, and with the Phasking Pen. Neo/Phasking Pen order was counterbalanced by task and participant. Tasks were performed in the same order for each participant, as a progression in complexity.

After the tasks, all participants were asked to complete a Likert scale (1:7) on: (1) *How likely are you to use this system again in the future?*; and, (2) *Would you use feature again (assume a more refined version of the tool)*? for each of Line (Bring), Line (Ruler), Rectangle, and Shared Control. (3) *The movement speed and force is appropriate for me*.

This general procedure was the entirety of the Novice evaluation (30 minutes/session). Seven participants (aged 21-30, 4 female) had backgrounds of Computer science and Forestry.

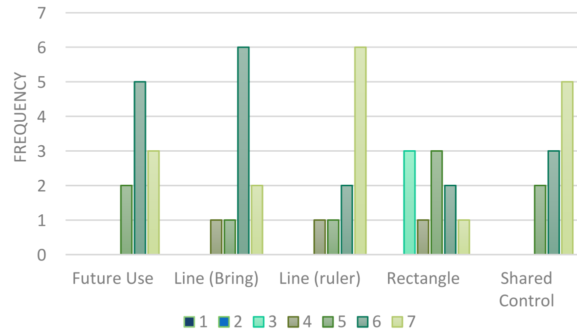


Figure 11: Users' Likert scale responses for the Phasking Pen(N=10, 7 novices and 3 experts; 7 is positive).

Experts: Procedure for Qualitative Interview – Following the task-based interaction period, we carried out a 30-minute semi-structured interview with our domain experts, covering task experience, relevance to their professional work, and potential impact. A complete session for experts took 60 minutes.

Results

Quantitative Likert Responses (N=10) – In total, participants performed 204 trials. Figure 11 shows the responses of all participants (novices and experts) to the survey questions. The Line function (ruler, or *block)* was the most popular feature, followed by SC, with *Bring* close behind and a strong interest in adoption. Rectangle was the only feature to receive any responses below neutral.

Table 4 shows task precision of manual drawing *vs.* phasking. Straight lines cover the basics; circle shows the system's ability to generate force feedback in a full range of 0-360 degrees, and rectangle+perspective a more complex task and guidance. Our data show shared control reducing error, with both *bound* and *bring* constraints.

Experts' Profiles and practice – The three participants (2 female, all right-handed) had practiced in the area of landscape architecture for periods ranging from 3-12 years; E1 in outdoor, E2 urban and E3 residential design. None had prior haptics experience. All confirmed that their process started with hand-drawn conceptual sketches, an important aspect of both ideation and client and collaborator communication,

Table 4: Precision of manual drawing *vs.* phasking.

Task	System	Mean / std abs error, mm
Line	Neo	1.56 / 0.43, N=10
Line *(Bring)*	SC	1.38 / 0.33, N=10
Line *(Bound)*	SC	1.20 / 0.30, N=10
Circle	Neo	5.39 / 5.34, N=7
Circle *(Bring)*	SC	5.20 / 5.65, N=7
Rect	Neo	5.91 / 3.87, N=7
Rect *(Bring)*	SC	3.38 / 3.21, N=7
Rect+persp	Neo	9.21 / 6.86, N=7
Rect+persp *(Bring)*	SC	2.22 / 3.21, N=7

Shared control (SC) or just Neo Pen (manual: single point contact, no guidance).

including in public meetings (E2). While estimating hand drawing as 10-20% of their entire process, after which continuity and precision required CAD, they wished for more:

> *My personal preference is hand drawing and sketching but I also like and appreciate the precision and the tools that other CAD basically provides you.* [E1]

> *It's not about preference. It's about the tools that we currently have.* [E2]

E3 additionally mentioned the importance of color in hand-drawn work, for impact in public communication.

Basic task performance, quality and experience – Participants rated appropriateness of speed and force at [5.5,7,5] on a 7-point Likert scale; E1 requested more flexibility in speed and force control. They could feel and understand the feedback, and control it to degrees estimated at 60-100%; E2 (60%) described "*an invisible barrier that sort of holds you ... keeps your line nice and tidy*". Some noted initial awkwardness, with greater comfort by the end of the tasks, and that some CAD functions had unfamiliar steps (*e.g.*, sequence of marking).

They found its use generally intuitive; E1 liked the ability to construct a perfectly straight line, and E3 noted that "*Line [ruler] part was really interesting because it really helped me to draw a straight line and it was the most interesting ... aspect of using it.*" [E3]

All were enthusiastic of SC's value and intuitiveness: "*an amazing sort of transition between a hundred percent computer drafting and hand drawing*" [E1]; "*in a way the device starts reacting smartly to what I'm intending to do.* " [E2]. E3 sometimes pressed too hard, then found SC less controllable. Precision (E1) and bulk (all) were identified as primary issues.

Relevance to professional work – All valued the potential to cycle between paper and digital work, in contrast to their present one-way transition. Of features evaluated (geometry, barriers, SC), all identified SC as most useful. Among specific widgets, they preferred Line (Likert responses [6.5,6.5,6]), with more mixed but still generally positive reception for Rectangle [5.5,3,6] and Ruler [4,5,7]. For screen-free and large-surface potential, E1 noted the difficulty of maintaining control on large surfaces (where Phasking Pen could help); E2 mentioned value in a public engagement process, and communicating extemporaneously with an audience. E3 wanted a phasking ruler for section and building elevations, now done by hand.

Overall impact and interest in adoption – Experts responded to the adoption question with [6,7,6]. Presuming a slimmed-down and more precise device, participants were positive on productivity (*e.g.*, by integrating paper sketching later in process). E3 indicated great interest in precise technical sketching, rather than conceptual work where roughness was fine, and liked the efficiency – "*you put away the ruler and then you have two things in one (pen and ruler)*". E1 predicted value in education, noting 10 years of training with constant practice.

DISCUSSION

We have presented and implemented *phasking*, a form of computer-assisted drawing that brings a virtual environment constraint system to pen and paper, and alloed users to access a continuum of assistance (type and hardness of constraints) via fluid sharing of control authority.

We created this framework out of the varied purposes that people bring to sketching. Phasking requires active force feedback, because it entails active and passive constraints, and user-controlled gradation in, *e.g.*, a restoring force upon violating a helpful constraint. Paper sketching requires an implement that can operate screen-free. For large, free movements, the device needs to be free-roaming (untethered).

In creating the Phasking pen, a major extension of a previous ballpoint drive device, we focused on strength and backdrivability, attributes difficult to jointly optimize but crucial for feature rendering and unimpeded movement. Objective performance metrics provide a benchmark for future improvements. We implemented an essential set of framework primitives and a paper-based tool palette to access them, with which users can carry out a complete drawing task on paper.

We shared phasking with novice and professional sketchers. They could feel and understand the forces, and found strength and speed adequate while wanting more precision. They also told us of a strong desire to be able to hand-sketch more, which requires integration throughout their process, not just at the start. Because of the volume of their use, they valued physical supports for productivity and prized fluid control sharing; and suggested that geometry construction would have been valuable when they were learning to sketch.

Limitations: Rotation in the drawing direction (pitch) is constrained by two-point contact, addressable by delivering ink via the ball itself. Our evaluation revealed notable individual differences, particularly in magnitude and smoothness of force profile, signalling a need for training. While new users learn to deploy pressure to optimize SC use, graphical or auditory feedback will be valuable. A screen as training-wheels could also assist with learning CAD features. Finally, a nonlinear relationship between pressure and control share might work better, a topic of future work.

CONCLUSIONS AND NEXT STEPS

Our evaluations indicate that this prototype was sufficient to assess phasking's potential. The concept found an enthusiastic reception, and feedback points to key improvements.

With a full interactive experience in place, we have proved possible many other interesting functions within this framework, including modifying elements (*e.g.*, resize, rotate, amend) as well as copy, paste, undo; identifying free-drawn marks as parametric objects; combining objects into a virtual construct, and even simulating dynamic virtual systems. It is a small straightforward step to full *digital twinning*: modify it on-screen then bring it back to paper with guided tracing.

Phasking is too different from other digital tools to know its full potential. Being untethered, portable and self-contained, Phasking Pen can, with attainable modifications, be used on arbitrary surfaces. This could lead to a new way of 'drawing on a napkin', support blind mobility by revealing maps on a corridor wall, and allow drawing and playing with simulations on a whiteboard – an 'object to think with' [28, 30].

ACKNOWLEDGMENTS

This work was primarily supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). We thank our participants and reviewers for their scientific inputs.

REFERENCES

[1] David A. Abbink, Mark Mulder, and Erwin R. Boer. 2012. Haptic shared control: smoothly shifting control authority. *Cognition, Technology Work* 14, 1 (2012), 19–28. http://dx.doi.org/10.1007/s10111-011-0192-5

[2] Carlo A. Avizzano, Massimo Satler, Giovanni Cappiello, Andrea Scoglio, Emanuele Ruffaldi, and Massimo Bergamasco. 2011. MOTORE: A mobile haptic interface for neuro-rehabilitation. In *2011 RO-MAN*. IEEE, 383–388.

[3] B. Bayart, A. Pocheville, and A. Kheddar. 2005. An adaptive haptic guidance software module for I-TOUCH: example through a handwriting teaching simulation and a 3D maze. In *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*. 6 pp.–. DOI: http://dx.doi.org/10.1109/HAVE.2005.1545651

[4] BOSCH. 2015. Absolute orientation sensor. (2015). https://www.bosch-sensortec.com/bst/products/all_products/bno055

[5] J.E. Colgate, W. Wannasuphoprasit, and M.A. Peshkin. 1996. Cobots: Robots for Collaboration with Human Operators. In *the 5th Ann. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME/IMECE*, Vol. DSC:58. Atlanta, GA, 443–439.

[6] Heather Culbertson and Katherine J Kuchenbecker. 2017. Ungrounded Haptic Augmented Reality System for Displaying Roughness and Friction. *IEEE/ASME Transactions on Mechatronics* 22, 4 (2017), 1839–1849. https://ieeexplore.ieee.org/abstract/document/7917356

[7] Daniel Dixon, Manoj Prasad, and Tracy Hammond. 2010. iCanDraw: Using Sketch Recognition and Corrective Feedback to Assist a User in Drawing Human Faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Atlanta, Georgia, USA New York, NY, USA, 897–906.

[8] Mohamad A. Eid, Mohamed Mansour, Abdulmotaleb H. El Saddik, and Rosa Iglesias. 2007. A Haptic Multimedia Handwriting Learning System. In *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education (Emme '07)*. ACM, New York, NY, USA, 103–108. DOI: http://dx.doi.org/10.1145/1290144.1290161

[9] Pololu Rootics & Electronics. 2019. Micro Metal Gearmotors. (2019). https://www.pololu.com/category/60/micro-metal-gearmotors

[10] Piyum Fernando, Roshan Lalintha Peiris, and Suranga Nanayakkara. 2014. I-Draw: Towards a Freehand Drawing Assistant. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures the Future of Design - OzCHI'14*. ACM Press, New York, New York, USA, 208–211.

[11] Ben Forsyth and Karon E. MacLean. 2006. Predictive Haptic Guidance: Intelligent User Assistance for the Control of Dynamic Tasks. *IEEE Trans on Visualization and Computer Graphics* 12, 1 (2006), 103–113. http://www.computer.org/csdl/trans/tg/2006/01/v0103-abs.html

[12] Paul Griffiths and R. Brent Gillespie. 2004. Shared Control between Human and Machine: Haptic Display of Automation during Manual Control of Vehicle Heading. In *the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*. 358–366.

[13] Anoto Inc. 2017. Anoto Digital Pen. (2017). http://www.anoto.com/

[14] Neo Smartpen Inc. 2018. Neo smartpen M1. (2018). https://www.neosmartpen.com

[15] Oliver Beren Kaul, Max Pfeiffer, and Michael Rohs. 2016. Follow the Force: Steering the Index Finger Towards Targets Using EMS. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2526–2532. DOI: http://dx.doi.org/10.1145/2851581.2892352

[16] Soheil Kianzad and Karon E MacLean. 2018. Harold's purple crayon rendered in haptics: Large-stroke, handheld ballpoint force feedback. In *2018 IEEE Haptics Symposium (HAPTICS)*. 106–111. DOI: http://dx.doi.org/10.1109/HAPTICS.2018.8357161

[17] 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.

[18] Golan Levin. 2000. *Painterly Interfaces for Audiovisual Performance*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge.

[19] Y. Li, J. C. Huegel, V. Patoglu, and M. K. O'Malley. 2009. Progressive shared control for training in virtual environments. In *3rd Worldhaptics Conference (WHC '09)*. IEEE Press, Salt Lake City, UT, 332–337.

[20] Riverbank Computing Limited. 2019. PyQT. (2019). https://riverbankcomputing.com/software/pyqt/intro

[21] Long-Fei Lin, Shan-Yuan Teng, Rong-Hao Liang, and Bing-Yu Chen. 2016. Stylus Assistant: Designing Dynamic Constraints for Facilitating Stylus Inputs on Portable Displays. In *SIGGRAPH ASIA 2016 Emerging Technologies (SA '16)*. ACM, New York, NY, USA, Article 14, 2 pages. DOI: http://dx.doi.org/10.1145/2988240.2988255

[22] Pedro Lopes, Doaa Yuksel, Francois Guimbretiere, and Patrick Baudisch. 2016. Muscle-plotter: an Interactive System based on Electrical Muscle Stimulation that Produces Spatial Output. In *29th Annual Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 207–217.

[23] Pam A. Mueller and Daniel M. Oppenheimer. 2014. The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking. *Psychological Science* 25, 6 (2014), 1159–1168. https://doi.org/10.1177/0956797614524581

[24] J. Mullins, C. Mawson, and S. Nahavandi. 2005. Haptic handwriting aid for training and rehabilitation. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3. 2690–2694 Vol. 3. DOI: http://dx.doi.org/10.1109/ICSMC.2005.1571556

[25] Ken Nakagaki and Yasuaki Kakehi. 2014. Comp*Pass: A Compass-based Drawing Interface. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 447–450. DOI: http://dx.doi.org/10.1145/2559206.2574766

[26] Akira Nomoto, Yuki Ban, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2016. Supporting Precise Manual-handling Task Using Visuo-haptic Interaction. In *Proceedings of the 7th Augmented Human International Conference 2016 (AH '16)*. ACM, New York, NY, USA, Article 10, 8 pages. DOI: http://dx.doi.org/10.1145/2875194.2875216

[27] Ayberk Özgür, Wafa Johal, Francesco Mondada, and Pierre Dillenbourg. 2017. Haptic-Enabled Handheld Mobile Robots: Design and Analysis. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 2449–2461. DOI: http://dx.doi.org/10.1145/3025453.3025994

[28] S. Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

[29] Mark Price and Frank C. Sup. 2016. A robotic touchscreen totem for two-dimensional haptic force display. In *2016 IEEE Haptics Symposium (HAPTICS)*. IEEE, 72–77.

[30] Mitchel Resnick, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. 1998. Digital Manipulatives: New Toys to Think with. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., Los Angeles, California, USA New York, NY, USA, 281–287.

[31] Hasti Seifi, Gunhyuk Park, Farimah Fazlollahi, John Sastrillo, Jessica Ip, Ashutosh Agrawal, Katherine Kuchenbecher, and Karon E. MacLean. 2019. Haptipedia: Accelerating Haptic Device Discovery to Support Interaction Engineering Design. In *Conference on Human Factors in Computing Systems (CHI'19)*. ACM, Glasgow, UK, 12 pgs.

[32] Yu shan Chang, Mavis Yi-Ching Chen, Meng-Jung Chuang, and Chia hui Chou. 2019. Improving creative self-efficacy and performance through computer-aided design application. *Thinking Skills and Creativity* 31 (2019), 103 – 111. DOI: http://dx.doi.org/https://doi.org/10.1016/j.tsc.2018.11.007

[33] Scott Snibbe, Sean Anderson, and Bill Verplank. 1998. Springs and Constraints for 3D Drawing. In *Third Phantom User's Group*, Vol. M.I.T. Artificial Intelligence Laboratory Technical Report AITR-1643. 1–4.

[34] Hyunyoung Song, Tovi Grossman, George Fitzmaurice, François Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. 2009. PenLight: Combining a Mobile Projector and a Digital Pen for Dynamic Visual Overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 143–152. DOI: http://dx.doi.org/10.1145/1518701.1518726

[35] Jürgen Steimle. 2009. Designing pen-and-paper user interfaces for interaction with documents. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction - TEI '09*. ACM Press, New York, New York, USA, 197–204.

[36] A Teranishi, G Korres, W Park, and M Eid. 2018. Combining Full and Partial Haptic Guidance Improves Handwriting Skills Development. *IEEE Transactions on Haptics* 11, 4 (2018), 509–517. https://ieeexplore.ieee.org/abstract/document/8400473

[37] P. Vandoren, T. Van Laerhoven, L. Claesen, J. Taelman, C. Raymaekers, and F. Van Reeth. 2008. IntuPaint: Bridging the gap between physical and digital painting. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. 65–72. DOI: http://dx.doi.org/10.1109/TABLETOP.2008.4660185

[38] Junichi Yamaoka and Yasuaki Kakehi. 2013. dePENd: Augmented Handwriting System Using Ferromagnetism of a Ballpoint Pen. In *26th Annual ACM Symposium on User Interface Software and Technology*. ACM, St. Andrews, Scotland, United Kingdom New York, NY, USA, 203–210.

[39] Raphael F Zory, Delphine Boërio, Marc Jubeau, and Nicola A Maffiuletti. 2005. Central and peripheral fatigue of the knee extensor muscles induced by electromyostimulation. *International journal of sports medicine* 26 10 (2005), 847–53.