Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI

Hiroki Kaimoto*
The University of Tokyo
Tokyo, Japan
University of Calgary
Calgary, Canada
hkaimoto@xlab.iii.u-tokyo.ac.jp

Kyzyl Monteiro*
IIIT-Delhi
New Delhi, India
University of Calgary
Calgary, Canada
kyzyl17296@iiitd.ac.in

Mehrad Faridan
University of Calgary
Calgary, Canada
mehrad.faridan1@ucalgary.ca

Jiatong Li
University of Chicago
Chicago, U.S.A.
jtlee@uchicago.edu

Samin Farajian
University of Calgary
Calgary, Canada
samin.farajian@ucalgary.ca

Yasuaki Kakehi
The University of Tokyo
Tokyo, Japan
kakehi@iii.u-tokyo.ac.jp

Ken Nakagaki
University of Chicago
Chicago, U.S.A.
knakagaki@uchicago.edu

Ryo Suzuki
University of Calgary
Calgary, Canada
ryo.suzuki@ucalgary.ca

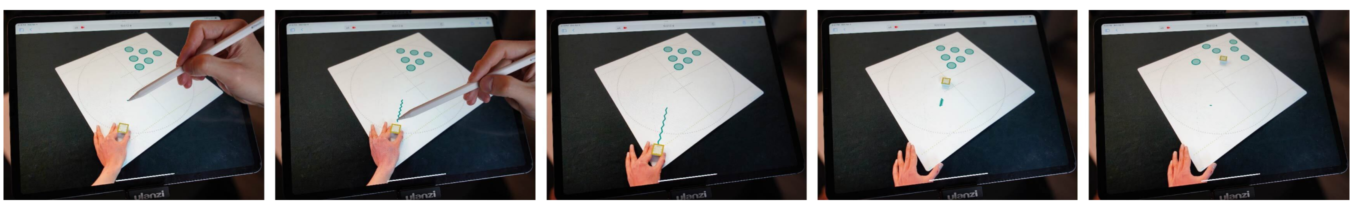

Figure 1: Sketched Reality explores bi-directional interactions between AR-based virtual sketches and actuated tangible UIs. When the user sketches a virtual spring (green lines), the user can start pulling the spring with a physical robot (yellow box), so that the sketched virtual spring *affects* the physical robot by applying the force like a slingshot game (virtual → physical). When the robot hits the virtual sketched circles, then the robot also *affects* the virtual objects, as if the physical robot collides with virtual sketched circles like a billiard board game (physical → virtual).

ABSTRACT

This paper introduces Sketched Reality, an approach that combines AR sketching and actuated tangible user interfaces (TUI) for ***bi-directional sketching interaction***. Bi-directional sketching enables virtual sketches and physical objects to *"affect"* each other through physical actuation and digital computation. In the existing AR sketching, the relationship between virtual and physical worlds is only one-directional — while physical interaction can affect virtual sketches, virtual sketches have no return effect on the physical objects or environment. In contrast, bi-directional sketching interaction allows the seamless coupling between sketches and actuated TUIs. In this paper, we employ tabletop-size small robots (Sony Toio) and an iPad-based AR sketching tool to demonstrate the concept. In our system, virtual sketches drawn and simulated on an iPad (e.g., lines, walls, pendulums, and springs) can move, actuate, collide, and constrain physical Toio robots, as if virtual sketches and the physical objects exist in the same space through seamless coupling between AR and robot motion. This paper contributes a set of novel interactions and a design space of bi-directional AR sketching. We demonstrate a series of potential applications, such as tangible physics education, explorable mechanism, tangible gaming for children, and in-situ robot programming via sketching.

*Both authors contributed equally to the paper

UIST '22, Oct 16–19, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**.

KEYWORDS

augmented reality; mixed reality; actuated tangible interfaces; swarm user interfaces

ACM Reference Format:
Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. In *UIST '22: The 35th Annual ACM Symposium on User Interface Software and Technology, Oct 16–19, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

With the advent of augmented and mixed reality (AR/MR) technology, many Human-Computer Interaction (HCI) researchers recently started exploring AR sketching as a means of interactive authoring and exploration of AR objects. In contrast to screen-based sketching [27, 46], AR sketching allows us to embed virtual sketches in the real world, which helps combine digital and physical worlds in more immersive and improvisational ways for designing [4], annotating [19], and collaborating [11] in the real world.

However, in the existing AR sketching tools, or in the existing AR interfaces in general, the coupling between AR sketches and the physical world is only ***one directional*** —meaning that virtual sketches have ***no effect*** on the physical environment. (Figure 2 left).

This paper introduces Sketched Reality, an approach that combines AR sketching and actuated tangible user interfaces (TUI) for ***bi-directional sketching interaction*** (Figure 2 right). In our paper, *bi-directional sketching interaction* refers to the interaction in-between virtual sketches and actuated physical objects that affect each other through physical actuation and digital computation.

To demonstrate this concept, we employ tabletop-size mobile robots (Sony Toio) and an iPad-based AR sketching interface to seamlessly couple physical robot and virtual interactive sketches. With our system, the user can sketch and embed lines onto a tabletop surface using an iPad and WebXR (A-Frame and 8th Wall). Similar to conventional AR sketching [52], these sketched lines are dynamic so that they can interact with tabletop robots that can represent an actuated tangible object. By dynamically calculating the collision and simultaneously moving these robots, it is possible to make illusions, as if virtual sketches and the physical robots affect each other.

This paper presents a set of design space of such a bi-directional sketching interaction in the following four categories: 1) *boundary constraints* (e.g., contour, walls), 2) *geometric constraints* (e.g., constant length, angle), 3) *applied force* (e.g., gravity, friction, attraction force), and collision (e.g., hitting objects). In total, we formulate the eight different interactions between virtual sketches and physical robots. By combining these elements, the user can quickly and easily create interactive effects on-demand in an improvisational and tangible manner, without any pre-defined programs or configurations. For example, sketched lines in AR can dynamically create a *virtual constraint* that can actually enclose and move actuated tangible objects. Or a sketched virtual material can provide a *force* to a physical object so that the user can shoot like a slingshot. We demonstrate the potential of this approach through various use cases and application scenarios, such as tangible physics education for children, explorable mechanism, tangible gaming, and in-situ robot programming and actuated TUI via sketching. We believe this paper opens up a new way of interactions and exploration for AR and tangible user interfaces.

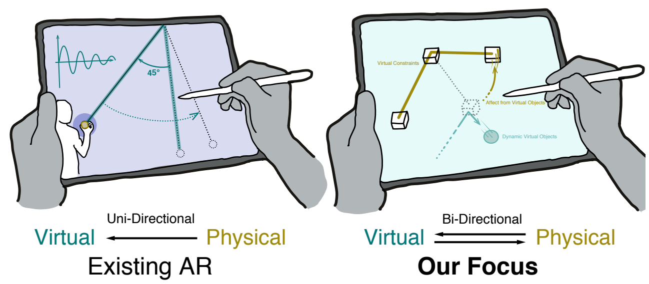

Figure 2: While, in conventional AR (such as in [52]), the virtual graphical information is affected by physical objects and actions, Sketched Reality explores how virtual can also affect physical objects by incorporating actuation with AR sketches.

The core novelty of this paper lies in the first exploration of the *concept* of bi-directional sketching interaction and its comprehensive *design space*. More specifically, this paper makes the following contributions:

(1) A concept and design space of Sketched Reality, allowing AR sketches to bi-directionally interact with actuated TUIs.
(2) An implementation of the system that uses a small tabletop robot (Sony Toio) as actuated objects and mobile AR (iPad and WebXR) as AR sketching interfaces.
(3) Interaction techniques and application scenarios enabled by our system, which includes tangible education, interactive games, and in-situ robot programming and control via sketching.

2 RELATED WORK

2.1 Augmented Reality Sketching Tools

In recent years, HCI researchers have explored various augmented and virtual reality (AR/VR) sketching interfaces. For example, Just a Line [17], TiltBrush [16], Gravity Sketch [18], and Vuforia Chalk AR [19] are commercially available sketching tools that support various 2D/3D sketching in AR/VR environments. In particular, one of the unique benefits of AR sketching over screen-based sketching is that the sketched elements can be embedded, situated, and contextualized in the real world, which allows the user interaction tightly coupled with the real environment. For example, Vuforia Chalk AR [19] allows the user to directly annotate a physical object via sketching so that the sketches can be embedded in the associated physical location, which is useful for many application scenarios (e.g., real-time remote assistant between experts and field technicians in a factory or construction site.) By combining a tablet as a sketching canvas, SymbiosisSketch [4], VRSketchIn [9], PintAR [11]) leverage a physical surface to serve as a geometric constraint for sketching in an immersive environment. Alternatively, SweepCanvas [31] and SketchingWithHands [26] use a physical object as a reference to create 3D shapes. These AR sketching tools enable a variety of applications, including education [43], entertainment [4], design [57], and collaboration [11].

While many of these tools only focus on embedding *static* sketches, more recent work has started exploring how to embed *dynamic* sketches to further blend virtual sketches with the real physical environment. For example, RealitySketch [52] explores *embedded and responsive* AR sketching, in which sketched objects can dynamically animate and interact with physical motion in the real world. To push the boundary of this AR sketching and tangible interaction research domain, we explore this novel idea and demonstrates this concept by combining augmented reality sketching with actuated tangible user interfaces.

2.2 Actuated Tangible User Interfaces

The original motivation for actuated tangible user interfaces was driven by the vision of coupling bits and atoms [41]. While traditional tangible interfaces can couple visual representation with physical interaction, these traditional TUI systems often face a challenge of digital-physical discrepancy — the physical manipulation can change the digital representation, but the digital computation cannot change the physical representation of passive and static objects. Such seamless coupling between virtual and physical worlds is not possible without *actuating* physical environment (e.g., changing physical environments, corresponding to the changes in the digital world). To address this limitation, HCI researchers have explored the research concept of actuated tangible user interfaces [41] and shape-changing user interfaces [2, 8, 44] since the early 2000s [39]. In actuated tangible user interfaces, physical objects are not merely augmented with digital overlays but are themselves dynamic and self-reconfigurable, so that they can change their physical properties to reflect the state of the underlying computation. These works have been greatly explored through different techniques such as magnetic actuation [40], ultrasonic waves [33], magnetic levitation [29], and wheeled and vibrating robots [38]. In recent years, a growing body of research started exploring tabletop mobile robots as actuated tangible objects. For example, Zooids [24, 28] introduces a swarm user interface, one of the class of actuated tangible interfaces which leverage tabletop swarm robots. Similarly, many different systems have also been proposed such as ShapeBots [54], HERMITS [35], HapticBots [53], Rolling Pixels [30], some of which also leverage the same commercially available Sony Toio robots [25, 35].

While most of these actuated tangible interfaces or swarm user interfaces are designed to interact based on *pre-programmed* behavior, other researchers have also explored the way to enable the user to program and design the motion on demand. For example, Topobo [42], MorphIO [37], and Animastage [36] allow the user to program physical behaviors with *direct manipulation*, as opposed to programming on a computer screen, so that the user can quickly control the actuation in improvisational ways. Most closely related to our work, Reactile [51] also leverages projection-based AR sketching to create virtual objects, which can be later bound and programmed for the dynamic motion of swarm user interfaces. However, Reactile only explores a small subset o bi-directional interaction (i.e., only bounded geometric parameters between virtual sketches and physical objects). In contrast, this paper provides a more holistic view of bi-directional sketching interaction, providing eight different categories of the *bi-directional virtual-physical interaction* of AR sketches and actuated tangible user interfaces. We show that by combining these novel sets of design space and interaction, we can achieve a more expressive and richer set of interactions and applications, that were previously unexplored in the literature.

2.3 Augmented Reality and Robotics

Augmented reality and robotics is a growing area over the last decades [32]. While the majority of augmented reality interfaces for robotics are considered AR-enhanced Human-Robot Interaction (AR-HRI) [50, 55], there are also a number of prior works that explore augmented reality for robotic and actuated tangible interfaces [50]. For example, exTouch [23] and Laser control Robot [21] use AR as a control and manipulation for robots. AR provides rich visual feedback and affordances, thus these AR interfaces provide a more intuitive understanding of how the robots or actuated interfaces should work. For example, AR is used to change the color, appearance, and added information to the robot [49]. Additionally, systems that allow human-robot interaction via sketching have been explored in multiple prior systems [6, 47, 48]. These prior works allow users to easily control physically actuated locomotive robots to navigate in space. Some other works have explored such sketch-based user interaction with augmented reality setup, letting users directly draw instruction over a real-physical environment [21, 34].

As the primary goals of these prior works were to control robots' behavior naturally, AR sketches (or reality overlaid virtual digital information) affect the robot in a single direction, but not another way around. In such a way, unlike conventional human-robot instructional interaction, we explore the combination of AR sketches and actuated tangible user interfaces, that enhance bi-directional interaction for users to affect digital information and physical objects interactively via in-situ embedded sketching.

As we can see, most of the existing works use AR to only *augment* the physical interface, but the *interaction* between AR objects and robots is not widely explored in the literature [50], except for a few examples. For example, Kobito [3] is one of the earliest explorations of synchronous coupling between AR and physical objects, which provides an illusion of virtual brownies pushing a physical cube. By taking inspiration from the prior art, this paper explores how we can support such interaction through improvisational AR sketching so that the user can create these effects more quickly and intuitively.

3 SKETCHED REALITY

3.1 Concept

3.1.1 Definition. This section introduces the concept of Sketched Reality. Sketched Reality is an approach to combining AR sketching and actuated tangible user interfaces (TUI) for *bi-directional sketching interaction.* A bi-directional AR sketching interaction allows the virtual sketches to move, actuate, collide, and constrain physical objects through synchronized coupling between physical actuation and virtual phenomenon (Figure 2 right). This enables us to further blend digital and physical worlds [20, 50].

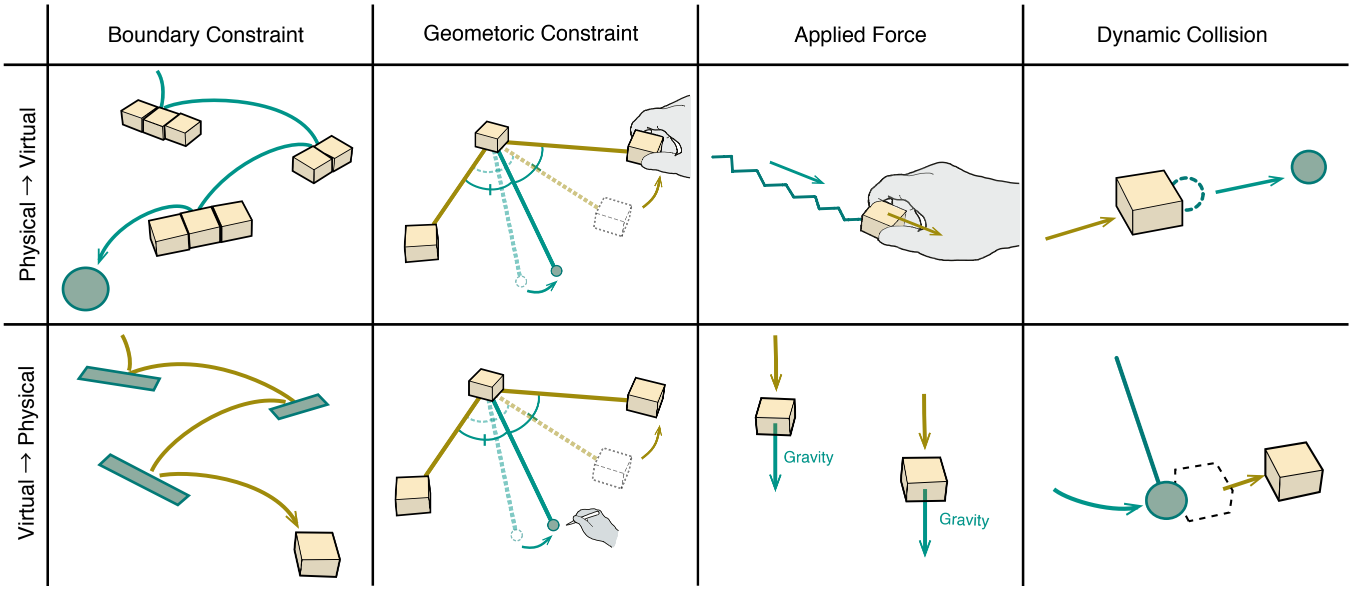

Figure 3: Design space of Sketched Reality: The horizontal axis shows four basic categories of our design space: 1) boundary constraints (left), 2) geometric constraints (middle left), 3) applied force (middle right), and 4) dynamic collision (right). The vertical axis shows descriptions for each element about 1) physical → virtual interaction: physical movement/interaction affects the behavior of the virtual objects (top line), and 2) virtual → physical interaction: virtual movement/interaction affects the behavior of the physical objects (bottom line).

3.1.2 Requirements and Scope. For Sketched Reality, the physical world needs to be synchronized with the virtual sketched animation and movement. To achieve this goal, we need to incorporate the physical actuation and reconfiguration through, for example, actuated tangible user interfaces [41], swarm user interfaces [28], shape-changing user interfaces [44], and reconfigurable environments [7]. The sketched interaction with all of these different configurations of the system is vast and goes beyond the scope of this paper. Therefore, in this paper, we specifically focus on bi-directional sketching interactions with *tabletop-scale small robots*, that can act as actuated tangible tokens and objects on a horizontal surface.

3.2 Design Space of Bi-Directional Interaction

In this section, we explore the design space of bi-directional interaction to understand how virtual and physical elements can affect each other. We have identified four basic categories: 1) boundary constraints, 2) geometric constraints, 3) applied force, and 4) dynamic collision (Figure 3). For each element, we describe 1) *physical → virtual interaction*: physical movement/interaction affects the behavior of the virtual objects, and 2) *virtual → physical interaction*: virtual movement/interaction affects the behavior of the physical objects.

3.2.1 Boundary Constraint. Boundary constraints refer to a virtual or physical constraint based on a static (stationary object), such as a wall or contour. The boundary constraint allows interaction between a dynamic and static object — a dynamic object can freely move in the physical or virtual space, whereas a static object stays at a certain position. This also applies to both virtual and physical worlds. For example, the physical boundary constraints also do the same thing for a virtual dynamic object (Figure 3 left top), and the virtual boundary constraints can prevent and reflect the physical object (Figure 3 left bottom).

3.2.2 Geometric Constraint. Second, geometric constraints refer to a fixed positional relationship between virtual and physical objects. For example, consider the situation where the user applies the geometric constraint as a middle angle between two lines, like a bisector line of a triangle or two lines. In physical to virtual interaction, when the user moves the endpoint of one line, the bisector line also moves accordingly to maintain an equal angle between two lines (Figure 3 middle left top). On the other hand, in virtual to physical interaction, the position of a physical robot can dynamically change based on the maintained geometric constraint. For example, in a similar situation, when the user moves the angle of the two lines by virtually moving one end of the line, then the position of the robot moves accordingly to maintain the equal angle of geometric constraint (Figure 3 middle left bottom). The geometric constraints can be various positional relationships such as length and angle, which can govern how virtual or physical objects behave through the connected and bound parameters.

3.2.3 Applied Force. Through bi-directional interaction, the user can also apply force to both virtual and physical objects. For example, in physical to virtual interaction, the user can apply an extension force to a virtual spring by deforming a virtual spring object (Figure 3 middle right top). In a similar way, the physical interaction can deform soft materials such as clothes or elastic string sketched in the virtual world. On the other hand, in virtual to physical interaction, the user can apply force to a physical object.

For example, the user can apply gravity force to the physical robot, so that the robot starts moving in a certain direction with a gravity force (Figure 3 middle right bottom). Such virtual force can take different forms such as magnetic force or friction.

3.2.4 Collision. Finally, dynamic collision refers to an interaction between virtual and physical objects, in which these objects collide with each other. In contrast to the boundary constraint, which focuses on the collision between *dynamic* and *static (stationary)* objects, the dynamic collision focuses on the interaction of multiple *dynamic* objects between virtual and physical worlds. For example, in physical to virtual interaction, a physical dynamic object hits a virtual dynamic object, which creates a movement and collision of the virtual object. In the same way, in virtual to physical interaction, a virtual object hits a physical object, then it moves the physical object.

4 DEMONSTRATING SKETCHED REALITY

4.1 System Overview

To demonstrate the concept of Sketched Reality, we develop a system that combines an AR sketching interface and actuated tangible interfaces, based on tabletop-size small robots. More specifically, our system employs iPad and WebXR for AR sketching and Sony Toio mobile robots [1] for actuated tangible objects.

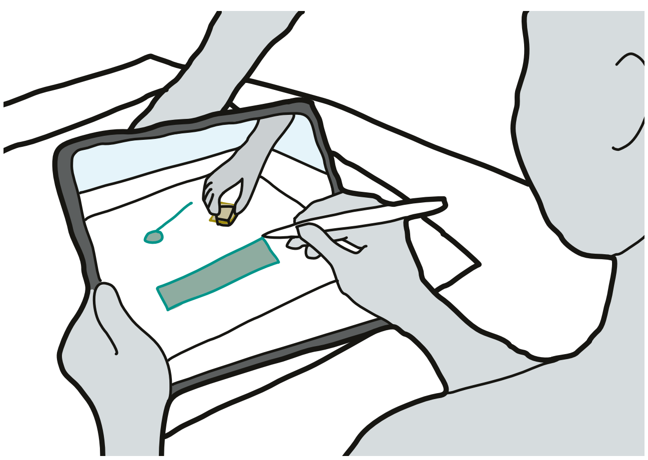

Figure 4: Basic setup of our system. The user draws virtual sketches through the iPad screen and Apple Pencil. After the simple calibration, the embedded 2D canvas onto a table surface in AR can be synchronized with the Sony Toio coordination. The green lines and objects represent the virtually sketched objects whereas a yellow cube object represents the physical Toio robot.

4.1.1 Interaction Workflow and Techniques. In our system, the user can *actuate* Sony Toio robots through AR sketching interactions. As we discussed in the Design Space section, there are several approaches, such as colliding virtual and physical robots, constraining robots' motion with virtual lines, and applying force to the robots through virtual property or environmental force. Similar to RealitySketch [52], the user can interact and manipulate the tangible Toio robots to modify the virtual sketched behaviors, but also modify or simulate the virtual sketches to change the behavior of tangible Toio robots through virtual geometric constraints or collision.

In our Sketched Reality system, the system allows the following interaction workflow for bi-directional sketching interaction:

(1) The user first selects a horizontal or vertical surface in the real world, then calibrates the position and coordination of the embedded 2D canvas to synchronize the coordination of the mobile robots.
(2) The user sketches a line in an AR screen using the iPad touch screen and Apple Pencil.
(3) The system recognizes a sketched line and creates a virtual body (closed shape like a circle or rectangle) or constraint (open shape such as lines or spring) based on the drawn shape.
(4) The user can apply property or force to both virtual sketched objects and physical robots in AR by selecting dynamic (affect gravity force) or static object (stationary object which does not affect gravity force).
(5) The user can interact with virtual or physical objects through virtually manipulating objects in AR scene or tangibly manipulating physical objects in the real world.

In the following section, we will describe each step in more detail.

4.2 Select a Surface and Calibrate the Position

4.2.1 Surface Detection. Sketched Reality system leverages mobile augmented reality (A-Frame and 8th Wall on iPad) to embed sketches onto a real-world surface. The user sketches with a pen on a touchscreen, where the sketched elements are overlaid onto a camera view of the real world. All of the sketched elements are 2D, so the user first needs to define the embedded 2D canvas by selecting a horizontal or vertical surface in the real world. Our system employs WebXR (8th Wall [1]) for the surface detection so that the system allows the user to sketch on a detected surface.

4.2.2 Calibration between AR Canvas and Physical Robot's Coordination. In Sketched Reality, the virtual and physical worlds need to be seamlessly coupled and synchronized with each other. Therefore, the coordination of the virtual and physical worlds is an important requirement for the system. To do so, once the system detects the surface, then the system allows the user to calibrate the position of the embedded 2D canvas in AR and Toio robot coordination. To calibrate the position, the user simply taps the center and left top corner of the physical square mat (Figure 5 left). Once the user taps two points, then the AR view shows the two red dots to confirm the position. When the calibration step is done, then the user can start drawing on a physical mat.

The main reason we chose the Sony Toio robot is its sophisticated and easily deployable tracking system. For tracking and localization of the robot, Toio has a built-in look-down camera at the base of the robot to track the position and orientation on a mat by identifying unique printed dot patterns, similar to the Anoto marker [2]. The

[1]https://www.sony.com/en/SonyInfo/design/stories/toio/

[2]https://en.wikipedia.org/wiki/Anoto

built-in camera reads and identifies the current position of the robot, enabling easy 2D tracking of the robots with no external hardware. Therefore, we can precisely track the location of multiple Toio robots on a physical square mat (Figure 5 left).

We leverage this built-in position and orientation tracking capability for synchronization between AR and the physical world. Based on the above simple calibration process, the system can match the coordinate systems between the Toio mat and the AR sketching canvas.

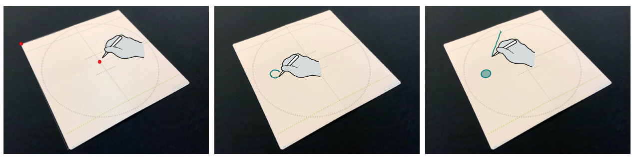

Figure 5: Position Calibration and Sketching to Create Circle Shapes: By tapping the screen and selecting the surface with two points, the user can calibrate the position between the AR canvas and the physical robot's coordination(left). Then, the user can create the 2D object in the AR scene by sketching a closed line. When the user draws a closed circle, the user can create a 2D ball shape of the body on a screen(middle). The user also can create an attached constraint with a single line by drawing to the sketched 2D body(right).

4.3 Sketch to Create Shapes and Constraints

4.3.1 Sketch a Line and Recognize a Shape. In Sketched Reality system, the user can create a virtual object with a simple line drawing. In our system, we have two virtual object categories: 1) body and 2) constraint. For constraint, it also has several variations such as fixed-line constraint, spring, or geometric relationship. The user can create these virtual objects by sketching them on a screen. In general, the closed line can be categorized as a body and the open line can be categorized as a constraint.

4.3.2 Create a Body with a Closed Line. First, the system detects the body or constraint based on the shape. By default, the system can basically recognize two different categories: 1) body and 2) constraint. The body refers to the virtual object such as circles or rectangles, whereas the constraint refers to the virtual relationship between bodies, such as a constant line or spring. For example, in Figure 5, the user draws a circle shape, then the system recognizes it as a virtual circular shape body, which is embedded in the AR view. Once the body is generated by sketches, the system shows the 2D object in the AR scene. In a similar manner, Figure 6 left and middle illustrates the situation where the user draws a rectangle shape and the system recognizes and embeds the rectangle body in the AR scene. By default, this 2D object is a floating body, which enables the collision with virtual or physical objects but does not affect gravity force.

4.3.3 Create a Constraint with a Non-Closed Line. Similarly, the user can also create a constraint by drawing a non-closed line. By default, the system supports two different constraints: 1) a static line constraint, and 2) a spring constraint. The system recognizes

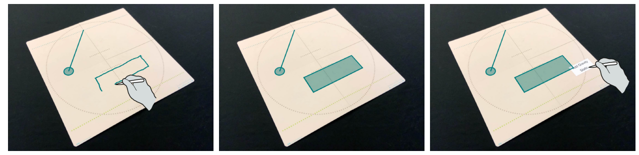

Figure 6: Sketching to Create Rectangle Shapes: The user can create a 2D rectangle shape of the body by sketching a closed rectangle-shaped line (left, middle). Then, the user applies static property to the sketched virtual rectangle from the AR menu (right).

these two constraints based on the sketched line shape. Similar to RealitySketch [52], if one end of the line is attached to a body (either virtual or physical), it creates an attached constraint in which when the body moves, the attached end of the constraint also moves and follows. For example, Figure 5 right and Figure 6 left show the user draws a line to the virtual sketched circle in AR. In this way, the user can create a fixed-length geometric constraint between the point on the surface and the attached virtual object. The end of these line constraints can not only be attached to the virtual sketched objects, but also attached to the physical robot.

4.4 Apply Force or Property to Virtual or Physical Object

4.4.1 Applying Force or Property to Virtual Objects. Once the user creates the virtual sketched object, the user can also change the property of the body or constraint through the AR interface. For the body, the user can choose the property of either 1) dynamic body or 2) static body. The dynamic body refers to the object that can move on a canvas, being affected by the gravitational force, such as a bouncing ball and block that falls down. On the other hand, a static body refers to a stationary object, such as a wall, boundary constraints, or slope that does not move. For example, the dynamic body can move based on the collision, while the static body does not move. Virtual objects with different properties can achieve different virtual-physical interactions, such as boundary constraints (static vs dynamic objects) or collisions (dynamic vs dynamic objects). To specify the property, the user can simply tap the sketched object, then the AR interface shows the menu of *"Add Gravity"* or *"Static"*. For example, Figure 6 right illustrates the user applying the static body property to the sketched rectangle shape. Once the user selects the menu, then the property of the selected virtual object will change. In a similar way, the user can also change other properties of the constraint. For example, the user can change the elasticity of the static line constraint or spring constraint by tapping and selecting the menu attached to these sketched constraints. Similar to RealitySketch [52], the user can also apply the geometric constraint, such as the parameter of the constraint (e.g., length or angle of the constraint).

4.4.2 Applying Force or Property to Physical Robots. In our system, physical robots also behave as dynamic or static bodies. Therefore, the user can also change the property or apply force to the physical robots, in the same way, we do with virtual sketched objects. This functionality helps the robots interact with virtual objects through

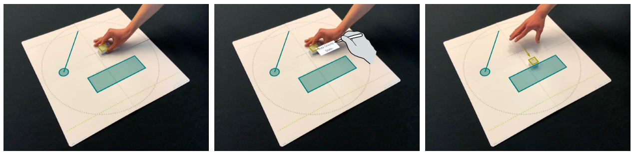

Figure 7: Apply Force or Property to Physical Robots: When the user places a physical robot on the mat (left), the user can see the menu of *"Add Gravity"* or *"Static"* (middle). The physical robot can collide with the virtual rectangle object by applying gravity (right).

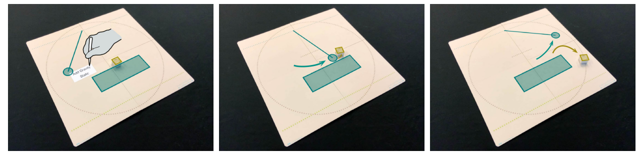

Figure 8: Interact between Virtual and Physical Objects: When the user applies gravity to the virtual sketched pendulum created with a circle-shaped body and an attached constraint (right), the virtual pendulum can affect the physical robot through collision, geometry constraints, boundary constraints, and applied force (middle). After the collision with the virtual pendulum, the physical object falls from the virtual rectangle scaffold through applied gravity force (left).

bi-directional interaction. When the user places a physical robot on the mat, then the system starts tracking the position of the robot and shows a yellow overlaid shape on top of the robot in AR view. In a similar manner, the user can tap the robot in the AR interface to show up in the menu of *"Add Gravity"* or *"Static"*. For example, if the user specifies the robot to be dynamic and effect by gravity, the robot can fall down in the direction of gravity. Furthermore, Figure 7 shows that the user selects the added gravity to apply the dynamic body property to a robot, then the robot starts falling down. Since the user applies the static object property to the virtual rectangle shape in Figure 6, the robot hits the ground and stays on top of the static body ground.

By applying force or property to virtual or physical objects, then these objects start interacting with each other. For example, Figure 8 shows that when the user applies gravity to a virtual pendulum ball, then it starts swinging to hit the robot to collide with each other so that the physical robot moves and starts falling down from the static ground. The user can also change the shape of the physical robot by tapping and drawing a physical robot. For example, when the user taps the physical robot and starts drawing, then it changes the virtual shape of the physical robot (e.g., change the yellow shape to a circular shape), which can change how to behave when interacting with virtual objects.

4.5 Interaction between Virtual and Physical Objects

4.5.1 Manipulating Virtual Objects. Once sketching and property change are done, the user can start interacting with the virtual objects. The dynamic body object can be basically manipulated with pen or touch interaction, so that the user can drag the object to a different position to interact with virtual or physical objects. These virtual interactions can affect the physical robots through collision, geometry constraints, boundary constraints, and applied force.

4.5.2 Manipulating Physical Robots. In the same way, the user can also manipulate and interact with physical robots through tangible and embodied interaction. With this interaction, the user can also affect and move the virtual objects in a seamless manner. This interaction allows a couple of different interactions for actuated TUI. First, it can support users in manipulating and controlling the actuated TUI and robot motion. Second, it can also support users to give another cycle to interact, observe, and interpret tangible physics models to explore and examine models. This allows users to 1) observe how the improvisational sketched models affect the motion of actuated TUIs based on the sketched relationships, 2) tangibly interact with the virtual physical models to perceive and explore the behaviors, 3) interpret to further understand how certain sketched relationship or physic model affects the behaviors. After users interact and interpret, they can go back to sketch mode to modify and improvise the models. In such a way, it can support iterative and explorative understandings.

5 IMPLEMENTATION

5.1 Mobile Robot Tracking and Control

We use Sony Toio, small tabletop-size robots, for actuated physical objects of our system. The system can track each Toio robot's position based on the pattern-printed tracking mat. Each robot has 3.2 cm x 3.2 cm x 2.5 cm and can travel at the speed of 24 cm/sec horizontally. The tracking mat covers 55 cm × 55 cm on top of the table and can track the position with a 1 mm error. To control the Toio robot, we need a Node.js server to run the control script, based on the open-source library of toio.js [3]. The Node.js server running on MacBook Pro 16-inch (Intel i7) can communicate with Toio robots through Bluetooth communication. One MacBook Pro can communicate and control up to 7 Toio robots simultaneously. For the controlling algorithm, we adapt to the open-source library of the existing Toio-based research project [35] [4] and rewrite the algorithm for Node.js based on toio.js.

5.2 AR Sketching Interface

For AR Sketching, we implement the interface based on WebXR, using A-Frame, Three.js, and 8th Wall. In A-Frame and Three.js, the HTML canvas can be embedded as an interactive 2D surface on top of the plane geometry. We show this canvas texture onto a square-shaped plane geometry, which can match with the Toio tracking mat. To detect the touch and pen interaction, we use ray casting from the camera to get the intersection between the ray and the canvas plane geometry. Then, by calculating the position within the 2D canvas based on UV mapping of the touched point, we convert the 3D touch position into the 2D coordinate of the canvas. To dynamically render 2D shapes, we employ Konva.js

[3]https://github.com/toio/toio.js/

[4]https://github.com/mitmedialab/HERMITS_UIST20

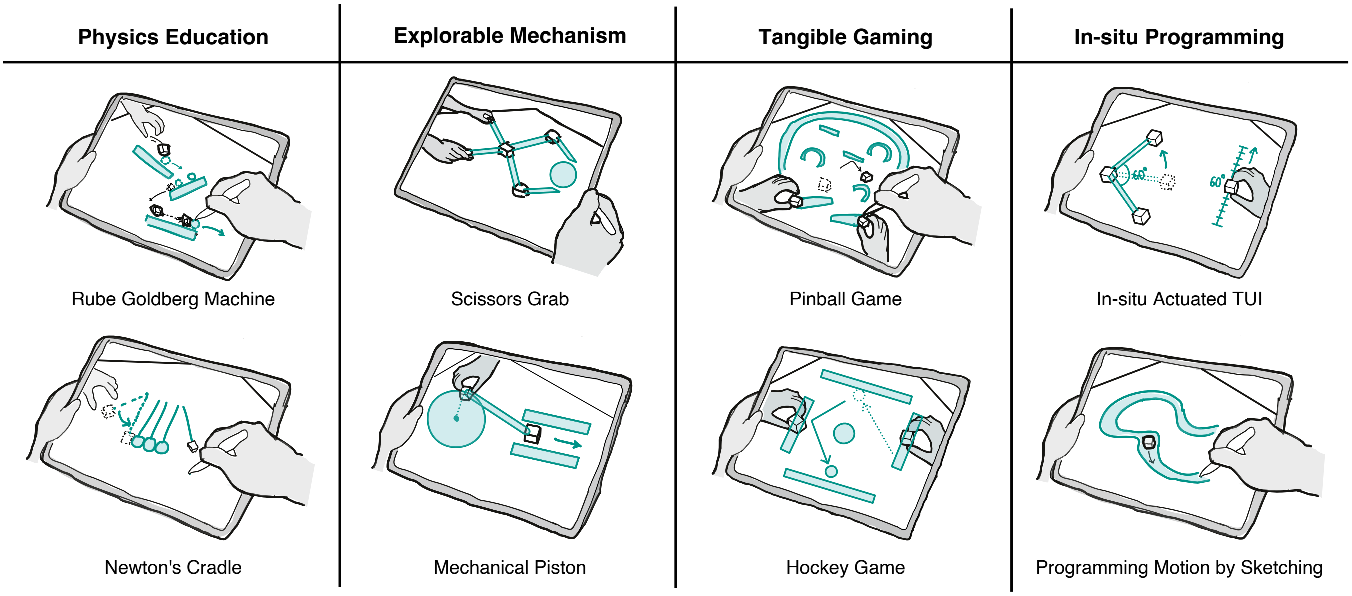

Figure 9: Application Sketches: Sketches highlighting the capability of Sketched Reality through Physical Education (left), Explorable Mechanism (middle left), Tangible Gaming (middle right), In-situ Programming (right).

for HTML Canvas drawing and manipulation and React.js for the JavaScript framework. Konva.js only supports the rendering of the virtual shape, thus we also use Matter.js for the 2D physics engine. By computing the position of each shape at each frame, we can animate the virtual sketched shape in Konva.js based on the physics simulation. In a similar manner, the system can also show the yellow object based on the robot's position. For the basic shape recognition, we use the JavaScript version of $1 Unistroke Recognizer, to detect the shape of a circle, rectangle, static constraint line, and spring.

5.3 Server and Communication

To synchronize between AR sketching and Toio robots, the system needs to communicate between the client-side web browser and Node.js server. To this end, the system uses the WebSocket protocol to 1) send the command to the next position of the Toio robot from the browser to the Node.js server, and 2) send the current position of each Toio robot from the Node.js server to the browser. In this way, the system allows synchronous communication between the AR interface and Toio control. The system runs a Node.js server on MacBook Pro (2021 16-inch Intel i7 CPU, 16GB RAM), which can communicate and control all of the robots.

6 APPLICATIONS

The bi-directional interaction between AR sketch and actuated user interface opens up a broad set of applications, allowing the user to interact with digital information in improvisational, responsive, reality-embedded, and tangible ways. This section introduces multiple application areas and examples within each area to highlight the capability of Sketched Reality (Figure 9).

6.1 Physics Education

Firstly, Sketched Reality can be used for physics education. While physics is commonly difficult to be taught only with equations and textbooks, through Sketched Reality, the combination of tangible objects together with overlaid sketches that define the abstract properties between objects can be useful for children and students to tangibly learn the concepts.

As shown in Figure 10 and 11, users can sketch different objects and strings in AR that interacts with actuated TUIs. By constructing their own physics setups that can be interacted tangibly, users could learn basic physics knowledge. In these setups, users can iteratively adjust different parameters of the physics sandbox through sketches.

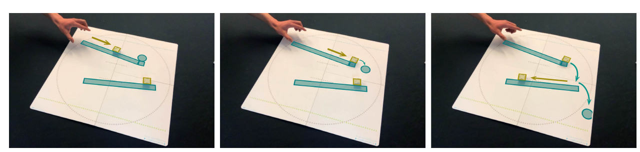

Figure 10: Physics Education Application - Rube Goldberg Machine: a user releases a physical Toio robot from the above (left), then the physical robot, with virtual gravity, rolls down the virtual slope then hits a virtual ball (middle). The virtual ball, in turn, affects another physical Toio robot, that is triggered to move (right).

The sketch can be done by students/learners, for them to exploratively and interactively learn physics behaviors, while it can also be done by teachers in classrooms for them to instantly construct physics simulations to describe and demonstrate different physics concepts intangible manner. Experiential and dynamic aspects give

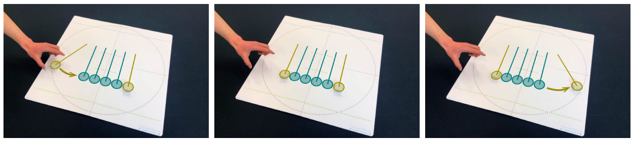

Figure 11: Physics Education Application - Newton's Cradle: users can sketch a set of virtual strings and balls that reacts to a physical robot's movement triggered by users. Once a physical robot, representing a ball, is released by a user (left), the kinematic force is propagated through the virtual balls after the collision with the virtual ball (middle), then the motion is transmitted to another physical robot (right).

tangibility to users. Through actuation, the actuated TUIs can represent physics properties in a dynamic and haptic way – e.g. users can feel the magnitude of gravity or mass via haptic feedback from the actuated tangibles [25].

6.2 Explorable Mechanism

Explorable Mechanism application demonstrates the use of Sketched Reality for mechanical design tasks. Similar to physics education, users can improvisationally design mechanisms where the virtual relationship is defined between actuated TUIs. Once interacted kinematic relationships are defined through AR sketches, users can grasp the Toio robots to control and examine the behavior of the mechanism. For example, as shown in Figure 12, a basic mechanism such as a piston can be verified through AR and actuated TUI, that dynamically responds to human input by grasping the actuated TUIs representing a handle of mechanism to control.

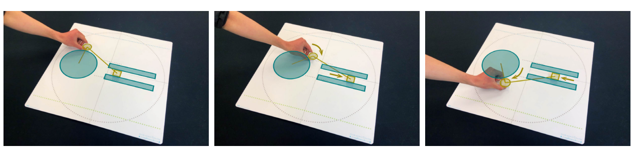

Figure 12: Mechanical Exploration - Piston Mechanism: As a user sketches a set of virtual link mechanisms attached to physical robot bodies, he/she can manually move one of the kinetic elements to explore and simulate the designed kinematics. The result is employed through the other robot's physical motion.

Mechanism design tasks could be difficult to get a tangible sense to simulate and test in screen-based CAD tools without tangibility. For example, some research works like Mechanism Perfboard [22] argue that the use of AR simulation improves the trial and error in mechanism design. In Sketched Reality, mechanism design and exploration application incorporate advantages from these tools for users to quickly iterate with sketches and physically explore and test through tangible interactions.

6.3 Tangible Gaming

Tangible Gaming applications bring virtual and graphical information together with immersive haptic and tangible interaction via the robots for creating storytelling and entertainment. For example, similar to Angry Birds [45], users can pull the robots using a virtually sketched spring or sling-shot to aim at targets either virtual or physical (Figure 1).

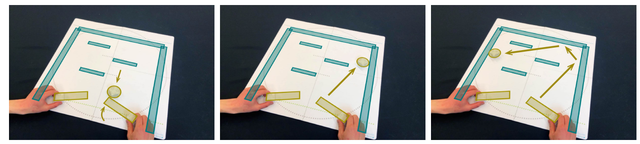

Figure 13: Tangible Gaming - Pin Ball: With the combination of virtually sketched obstacles and physical robots, users can enjoy playing virtual pinball by manually moving flippers (represented by Toio robots) to hit a robot, representing a ball. Users can feel the force through the flipper robots.

Users can feel the spring-like haptic feedback as gradually pulling the robots on the spring by dynamically controlling the actuation of robots, then see the trajectory to hit targets after releasing. Users can sketch different obstacles and stage gimmicks for making interactivity and gaming. Figure 13 represents how AR sketching can be used for pinball gaming, allowing users to sketch obstacles and play the game with a physical controller and moving physical balls. A similar gaming experience can be created for pong (Figure 14), where virtual and physical balls are mixed together for advanced game mechanics.

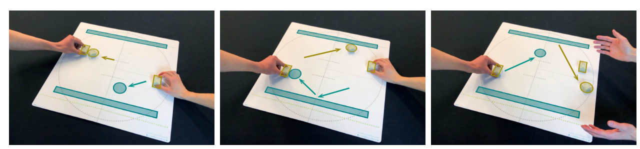

Figure 14: Tangible Gaming - Pong: a pong game can have both a virtual ball and a physical ball for two competing players to hit and play.

6.4 In-situ Programming

As AR interfaces and sketches have been used for defining and programming the behavior of IoT devices and robots [15, 50], Sketched Reality can be used to define the relationship of multiple robots as a way for users to program their behavior as everyday physical and actuated user interfaces. For example, as shown in Figure 15, by sketching lines between robots with variable angles in-between with an input slider, users can define the relationship in-between these components to bridge the input and output via AR sketch.

Such behavior can be dynamically used to develop bi-directional actuated interaction for users to define the relationship between actuated objects. In Figure 16, sketching allows users to create an in-situ virtual rope, that, after sketching, they can use this virtual rope to manipulate the behavior of multiple robots.

7 DISCUSSION AND FUTURE WORK

7.1 Different Approaches for AR Sketching

While our implementation mostly focuses on tablet-based AR, we acknowledge that *mobile AR* is just one of the possible approaches

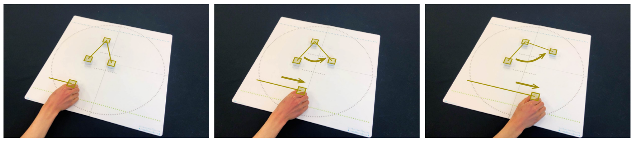

Figure 15: In-situ Programming - Actuated TUI control: by drawing linkage lines between the three robots as well as an input slider for another robot, users can tangibly define the angular relationship between robots through the input slider.

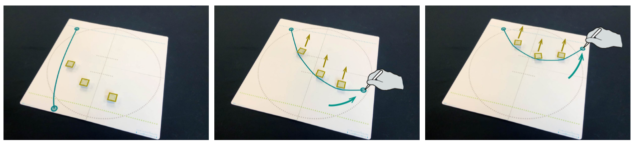

Figure 16: In-situ Programming - Rope Control: With the combination of a virtually sketched rope and physical robots, users can control the movement of physical robots through the control of virtual rope action.

among many. Based on AR and Robotics taxonomy [50], there are three possible approaches: *1) mobile AR, 2) head-mounted displays (HMDs), and 3) projection mapping*. In this section, we will discuss the pros and cons of each approach and give a holistic comparison of each of them.

7.1.1 Mobile AR. As we showed, tablet-based AR sketching tools allow an **easy setup** and implementation in a mobile setting. In addition, it allows using a pen or finger for tactile feedback while drawing. However, the biggest limitation of this approach is **non-hands-free interaction**. For example, in our setup, we used a tripod to overcome this issue, but this may not always be optimal. The use of a smartphone could alleviate this problem, but the screen size is limited.

7.1.2 HMDs. On the other hand, head-mounted displays (HMDs), like Hololens, allow **hands-free interaction**. This approach can address the current limitation, as the user does not need to hold the tablet when interacting with the virtual sketches and physical TUIs. In contrast, however, **precise sketching interaction** through finger and gestural interaction in HMD is still a challenge. In particular, the **lack of tactile feedback** of mid-air gestures makes the precise sketching interactions much harder [5]. Therefore, we should consider the sketching interface for touch interaction like MRTouch [56]. In addition, it is not trivial to share the experience with multiple users.

7.1.3 Projection Mapping. A projector allows the **collaborative multi-user experience**. This approach also enables hands-free interaction for multiple users, which is especially appropriate for educational use cases. However, The setup is not mobile and always requires a **tedious calibration process**. Moreover, in this way, the interaction and implementation can become more complex. For example, the system needs to distinguish between sketching, menu selection, virtual object dragging, and physical object dragging, which is not trivial and often requires another tracking mechanism for sketching.

7.1.4 Combination. Alternatively, an exciting future direction could be to combine multiple approaches. This allows us to leverage each benefit and overcome limitations of each approach. For example, previous research shows the benefits and advantages of using **Mobile AR + HMD** (e.g., BISHARE [58], SymbiosisSketch [4]) or **HMD + Projector** (e.g., ShareVR [13], AAR [14]). In future work, we should also explore different approaches for AR sketching and investigate how each approach would benefit the user interaction and experiences.

7.2 Benefits of Bi-Directionality

In this paper, we mostly focused on the exploration and demonstration of the Sketched Reality concept, and we did not formally evaluate the benefits of bi-directionality of AR and actuated TUIs. However, we believe AR sketches and actuated TUIs can benefit each other in many ways. Therefore, we would like to outline such benefits in this section.

7.2.1 How AR Sketching Benefits from Actuated TUIs. Actuated TUIs allow a number of benefits that cannot be done solely on AR. For instance, consider that in situations like Tangible Gaming, users can **feel the elastic force** of a virtual spring while pulling the robot (Figure 1). Also, in situations like exploration of mechanical linkage examples (Figure 12), the direct tangible manipulation allows the user to **feel the constraint** of the virtual object. In addition, if it enables the multi-user collaboration in a classroom, each user can interact with each other through virtually **inter-connected physical objects**. Such an experience can never be achieved solely with AR sketching.

7.2.2 How Actuated TUIs Benefit from AR Sketching. On the other hand, AR sketching allows the instantaneous creation of virtual objects, which may greatly increase the expressiveness of the actuated TUIs. For example, AR allows the scalable and flexible **object instantiation** (like virtual rope), which cannot be done with physical objects. The user can easily change, scale, and modify such a virtual object beyond the traditional physical constraints [40].

7.2.3 Beyond the Current Implementation. Due to the current limitation of mobile AR, such benefits may not be entirely clear as the experience seemingly takes place in the virtual world, not in the physical world. This is mostly because the mobile AR approach cannot fully blend the virtual and physical worlds, as it enforces the user to watch the tablet while their hands interact with the tangible. However, we envision the near future where everyone has an AR headset and the entire physical world becomes the interactive canvas. In such a fully blended reality world, we expect these benefits would become more interesting and clearer. We are interested in how our concept can be implemented in such a fully blended world in the future.

7.3 Exploration of Different Hardware

The concepts mentioned in this paper are not limited to mobile tabletop robots, but also can be expanded to different hardware. For example, larger-size robots like robotic vacuum cleaners or robot dogs could also be controlled through virtual sketched objects. Alternatively, we are also interested in further applying our concept to other actuated TUIs or shape-changing user interfaces. For example, the bi-directional interaction between AR and shape displays could be an interesting future exploration. Such interactions are partially explored in inFORM [10], but we believe there should be a larger design space for bi-directional virtual-physical interactions. In addition, we are also interested in exploring bi-directional interactions between virtual sketches and IoT devices. For example, the user could turn off or change the color of light bulb with sketched AR objects. By leveraging IoT and other robots, we could bring such an interaction to the everyday environment. We believe the ground concept and design space proposed in this paper help the future exploration in the future.

7.4 Advanced Sketch and Control Properties

In this paper, we have demonstrated the concept of Sketched Reality through preliminary sketching primitives (e.g., lines, blocks, spheres, and springs) with basic behaviors (e.g., bouncing, colliding, and linkage motions). A future implementation should incorporate more complex shapes, behaviors, and properties for users to flexibly sketch different elements and properties such as mechanical gears, friction/adhesion properties, or elastic surfaces. Increasing the library of sketch shapes and properties would enrich the versatility and adaptability of each application, for users to explore, improvise and touch AR sketches.

7.5 Scalability of Robots, Users, and Interaction Area

The scalability of robots, users, and interaction areas is obvious limitations in our prototype that can be further explored in the next steps. While a variety of user interaction methods with swarm robots have been proposed [24], we believe our proposed sketching interaction will further allow users to manipulate and interact with tens and hundreds of robots with in-situ sketches. Additionally, as sketching interactions are commonly introduced for multiuser interaction setup for collaborative and cooperative sketching/drawing, such a direction should be further explored in the future physical space, where a number of robots and a number of people exist. Additionally, while the current sketching area is limited to the mat of Toio, we plan to extend the interaction area much larger to, for example, our entire living space for full-spatial interactivity taking advantage of explorable AR, combined with embedded actuation in the real world.

7.6 User Study and Evaluation

While our paper primarily focused on the basic concept and prototype, we hope to further explore and evaluate our system with user studies. Such studies could compare the immersion, control efficiency, as well as enjoyment of user interaction with AR systems with and without our system. Application-specific evaluations could be explored further to understand the effect of our system. For example, for learning abstract physics or kinematics concepts, previous research like Mechanism Perfboard [22] or HoloBoard [12] provides evidence of the benefits of blending virtual and physical interactions for education. We believe the bi-directional sketching interaction proposed in this paper can also provide similar benefits in education and entertainment. As sketching interaction gives full freedom for people to control and design interaction with actuated TUIs, such evaluations should further suggest novel design space and functionality for bi-directional sketching interaction.

8 CONCLUSION

In this study, we proposed *Sketched Reality*, a concept of bi-directional virtual-physical interaction between AR sketches and actuated tangible user interfaces. Our design space categorized such a general interaction into four different categories, and we demonstrated this design space with a proof-of-concept prototype using tabletop mobile robots and an iPad-based AR sketching tool. With our implemented prototype, which combines AR, sketching interface, robot control, and primitive physics simulation, we have introduced a set of interaction techniques and demonstrated several applications, including tangible physics education for children, explorable mechanism, tangible gaming, and in-situ robot programming and actuated TUIs. We have discussed limitations and future work to highlight broader challenges to enable virtual-physical bi-directional interaction for the future physical environments in which digital computation and actuated robots are integrated.

ACKNOWLEDGMENTS

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Mitacs Globalink Research Award.

REFERENCES

[1] 8th Wall Inc. 2022. 8th Wall. https://www.8thwall.com/
[2] Jason Alexander, Anne Roudaut, Jürgen Steimle, Kasper Hornbæk, Miguel Bruns Alonso, Sean Follmer, and Timothy Merritt. 2018. Grand challenges in shape-changing interface research. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–14.
[3] Takafumi Aoki, Takashi Matsushita, Yuichiro Iio, Hironori Mitake, Takashi Toyama, Shoichi Hasegawa, Rikiya Ayukawa, Hiroshi Ichikawa, Makoto Sato, Takatsugu Kuriyama, et al. 2005. Kobito: virtual brownies. In *ACM SIGGRAPH 2005 emerging technologies*. 11–es.
[4] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 185.
[5] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR.. In *CHI*, Vol. 17. 5643–5654.
[6] Federico Boniardi, Abhinav Valada, Wolfram Burgard, and Gian Diego Tipaldi. 2016. Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2896–2901.
[7] Marco Cavallo, Mishal Dholakia, Matous Havlena, Kenneth Ocheltree, and Mark Podlaseck. 2019. Dataspace: A reconfigurable hybrid reality environment for collaborative information analysis. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 145–153.
[8] Marcelo Coelho and Jamie Zigelbaum. 2011. Shape-changing interfaces. *Personal and Ubiquitous Computing* 15, 2 (2011), 161–173.
[9] Tobias Drey, Jan Gugenheimer, Julian Karlbauer, Maximilian Milo, and Enrico Rukzio. 2020. VRSketchIn: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
[10] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation.. In *Uist*, Vol. 13. 2501–988.
[11] Danilo Gasques, Janet G Johnson, Tommy Sharkey, and Nadir Weibel. 2019. What you sketch is what you get: Quick and easy augmented reality prototyping with pintar. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[12] Jiangtao Gong, Teng Han, Siling Guo, Jiannan Li, Siyu Zha, Liuxin Zhang, Feng Tian, Qianying Wang, and Yong Rui. 2021. HoloBoard: a Large-format Immersive Teaching Board based on pseudo HoloGraphics. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 441–456.
[13] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2017. Sharevr: Enabling co-located experiences for virtual reality between hmd and non-hmd users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4021–4033.
[14] Jeremy Hartmann, Yen-Ting Yeh, and Daniel Vogel. 2020. AAR: Augmenting a wearable augmented reality display with an actuated head-mounted projector. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 445–458.
[15] Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 307–310.
[16] Google Inc. 2016. TiltBrush. https://www.tiltbrush.com/
[17] Google Inc. 2018. Just a Line. https://justaline.withgoogle.com/
[18] Gravity Sketch Inc. 2017. Gravity Sketch. https://www.gravitysketch.com/
[19] PTC Inc. 2017. Vuforia Chalk AR. https://chalk.vuforia.com/
[20] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: beyond tangible bits, toward transformable materials. *interactions* 19, 1 (2012), 38–51.
[21] Kentaro Ishii, Shengdong Zhao, Masahiko Inami, Takeo Igarashi, and Michita Imai. 2009. Designing laser gesture interface for robot control. In *IFIP Conference on Human-Computer Interaction*. Springer, 479–492.
[22] Yunwoo Jeong, Han-Jong Kim, and Tek-Jin Nam. 2018. Mechanism perfboard: An augmented reality environment for linkage mechanism design and fabrication. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.
[23] Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. 2013. exTouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. 223–228.
[24] Lawrence H Kim, Daniel S Drew, Veronika Domova, and Sean Follmer. 2020. User-defined swarm robot control. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[25] Lawrence H Kim and Sean Follmer. 2019. Swarmhaptics: Haptic display with swarm robots. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
[26] 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*. 797–808.
[27] Joseph J LaViola Jr and Robert C Zeleznik. 2006. Mathpad2: a system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2006 Courses*. 33–es.
[28] Mathieu Le Goc, Lawrence H Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th annual symposium on user interface software and technology*. 97–109.
[29] Jinha Lee, Rehmi Post, and Hiroshi Ishii. 2011. ZeroN: mid-air tangible interaction enabled by computer controlled magnetic levitation. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 327–336.
[30] Yujin Lee, Myeongseong Kim, and Hyunjung Kim. 2020. Rolling Pixels: Robotic Steinmetz Solids for Creating Physical Animations. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 557–564.
[31] Yuwei Li, Xi Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. 2017. SweepCanvas: Sketch-based 3D prototyping on an RGB-D image. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 387–399.
[32] Zhanat Makhataeva and Huseyin Atakan Varol. 2020. Augmented reality for robotics: A review. *Robotics* 9, 2 (2020), 21.
[33] Mark Marshall, Thomas Carter, Jason Alexander, and Sriram Subramanian. 2012. Ultra-tangibles: creating movable tangible objects on interactive tables. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2188.
[34] 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*. 599–606.
[35] Ken Nakagaki, Joanne Leong, Jordan L Tappa, João Wilbert, and Hiroshi Ishii. 2020. Hermits: Dynamically reconfiguring the interactivity of self-propelled tuis with mechanical shell add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 882–896.
[36] Ken Nakagaki, Udayan Umapathi, Daniel Leithinger, and Hiroshi Ishii. 2017. AnimaStage: hands-on animated craft on pin-based shape displays. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1093–1097.
[37] Ryosuke Nakayama, Ryo Suzuki, Satoshi Nakamaru, Ryuma Niiyama, Yoshihiro Kawahara, and Yasuaki Kakehi. 2019. Morphio: Entirely soft sensing and actuation modules for programming shape changes through tangible interaction. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 975–986.
[38] Diana Nowacka, Karim Ladha, Nils Y Hammerla, Daniel Jackson, Cassim Ladha, Enrico Rukzio, and Patrick Olivier. 2013. Touchbugs: Actuated tangibles on multi-touch tables. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 759–762.
[39] Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. 2002. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 181–190.
[40] James Patten and Hiroshi Ishii. 2007. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 809–818.
[41] Ivan Poupyrev, Tatsushi Nashida, and Makoto Okabe. 2007. Actuation and tangible user interfaces: the Vaucanson duck, robots, and shape displays. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. 205–212.
[42] Hayes Solos Raffle, Amanda J Parkes, and Hiroshi Ishii. 2004. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 647–654.
[43] Shwetha Rajaram and Michael Nebeling. 2022. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.
[44] Majken K Rasmussen, Esben W Pedersen, Marianne G Petersen, and Kasper Hornbæk. 2012. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 735–744.
[45] ROVIO. 2022. Angry Birds. https://www.angrybirds.com/
[46] Jeremy Scott and Randall Davis. 2013. Physink: sketching physical behavior. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*. 9–10.
[47] Vachirasuk Setalaphruk, Atsushi Ueno, Izuru Kume, Yasuyuki Kono, and Masatsugu Kidode. 2003. Robot navigation in corridor environments using a sketch floor map. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, Vol. 2. IEEE, 552–557.
[48] Danelle Shah, Joseph Schneider, and Mark Campbell. 2010. A robust sketch interface for natural robot control. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 4458–4463.
[49] Maki Sugimoto, Georges Kagotani, Minoru Kojima, Hideaki Nii, Akihiro Nakamura, and Masahiko Inami. 2005. Augmented coliseum: display-based computing for augmented reality inspiration computing robot. In *ACM SIGGRAPH 2005 Emerging technologies*. 1–es.
[50] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–32. https://doi.org/10.1145/1122445.1122456
[51] Ryo Suzuki, Jun Kato, Mark D Gross, and Tom Yeh. 2018. Reactile: Programming swarm user interfaces through direct physical manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[52] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 166–181.
[53] Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daniel Leithinger, and Mar Gonzalez-Franco. 2021. HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 1269–1281.
[54] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2019. Shapebots: Shape-changing swarm robots. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 493–505.
[55] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. 2018. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 316–324.
[56] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D Wilson, and Hrvoje Benko. 2018. MRTouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1653–1660.
[57] Emilie Yu, Rahul Arora, Tibor Stanko, J Andreas Bærentzen, Karan Singh, and Adrien Bousseau. 2021. Cassie: Curve and surface sketching in immersive environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
[58] Fengyuan Zhu and Tovi Grossman. 2020. Bishare: Exploring bidirectional interactions between smartphones and head-mounted augmented reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.