Marking Material Interactions with Computer Vision

Peter Gyory
peter.gyory@colorado.edu
ATLAS Institute
University of Colorado
Boulder, CO, USA

S. Sandra Bae
sandra.bae@colorado.edu
ATLAS Institute
University of Colorado
Boulder, CO, USA

Ruhan Yang
ruhan.yang@colorado.edu
ATLAS Institute
University of Colorado
Boulder, CO, USA

Ellen Yi-Luen Do
ellen.do@colorado.edu
ATLAS Institute & Computer Science
University of Colorado
Boulder, CO, USA

Clement Zheng
clement.zheng@nus.edu.sg
Division of Industrial Design &
Keio-NUS CUTE Center
National University of Singapore
Singapore

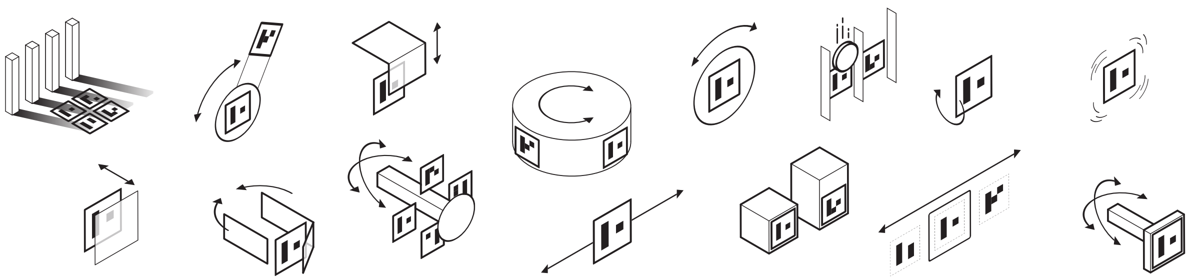


Figure 1: An abstract overview of the various physical interactions detected by computer vision markers used across fifteen projects in our annotated portfolio.

ABSTRACT

The electronics-centered approach to physical computing presents challenges when designers build tangible interactive systems due to its inherent emphasis on circuitry and electronic components. To explore an alternative physical computing approach we have developed a computer vision (CV) based system that uses a webcam, computer, and printed fiducial markers to create functional tangible interfaces. Through a series of design studios, we probed how designers build tangible interfaces with this CV-driven approach. In this paper, we apply the annotated portfolio method to reflect on the fifteen outcomes from these studios. We observed that CV markers offer versatile materiality for tangible interactions, afford the use of democratic materials for interface construction, and engage designers in embodied debugging with their own vision as a proxy for CV. By sharing our insights, we inform other designers and educators who seek alternative ways to facilitate physical computing and tangible interaction design.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580643

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Physical Computing, Tangible Interactions, Computer Vision, Making, Materiality

ACM Reference Format:
Peter Gyory, S. Sandra Bae, Ruhan Yang, Ellen Yi-Luen Do, and Clement Zheng. 2023. Marking Material Interactions with Computer Vision. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3580643

1 INTRODUCTION

As researchers and educators in the field of Tangible User Interface (TUI) design, we are tasked with both designing our own interactive systems as well as facilitating other designers. Building TUIs involves physical computing [36]—imbuing tangible artifacts with computational capabilities that can interact with the physical world. As both design practitioners and educators, we see our students as designers-in-training: budding creators who are at the beginning of their design careers and are actively acquiring the skills and knowledge they need to develop their own design practice. This paper is framed within our reflections on the experiences of other designers-in-training as they encounter physical computing for designing TUIs.

Physical computing is largely driven by off-the-shelf electronics and microcontroller units (MCU). TUI designers benefit from the wide-spread accessibility of MCU platforms, such as the Arduino [2] and Raspberry Pi [15]. As physical computing educators, we observed how these platforms lowered the barrier of entry for learning physical computing. For instance, when using these platforms, designers gain access to a wealth of prefabricated components designed to help novices bypass certain challenges associated with physical computing. The Arduino MCU and its integrated development environment, for example, support a plug-and-play approach to explore and program electronic circuits, sensors, and actuators all while black boxing [23] low-level concepts like hardware registers and digital signal processing. While these built-in features are meant to help designers-in-training focus on interactive experiences, such electronics-driven physical computing introduces various challenges when facilitating interaction design. Booth et al. [8] found that the most common errors physical computing novices face relate to circuit construction (e.g., putting an LED into a circuit backward, or forgetting to include a resistor). They also point out that these wiring errors can be compounded by the lack of approachable debugging tools for hardware. For instance, there is no analog to a program debugger that can pause during execution at the point of the error. As both interaction designers and educators, we have experienced these challenges ourselves and observed our students struggling with them first-hand. These experiences have encouraged us to explore approaches for physical computing beyond electronics.

In this paper, we discuss our facilitation of an alternative approach—using computer vision (CV) markers for physical computing to sense material interactions. We were inspired by previous work in this area, notably Reactivision [26], Sauron [41], and Printed Paper Markers [53]. The CV approach enables designers to build tangible interfaces with simple CV-recognizable graphics that can be applied to many different materials; and these interfaces do not require any electronic circuits. We were inspired by these design affordances of CV markers, and developed *Beholder*, a web-based programming library that leverages CV markers as a material for detecting tangible interaction.

With *Beholder*, we led a series of design studios with other designers, centered on building interactive artifacts with a CV-driven approach to physical computing. We looked at these studios as a whole and assembled fifteen project outcomes into a collection that we reflect on using the method of annotated portfolios [9].

1.1 Research Questions and Contributions

Annotated portfolios translate a set of individual design items into a "design space" that enables researchers to "make clear a domain of design" [19]. In our case, we leverage annotated portfolios to reflect on how designers approach CV markers as a material for interaction design and as an alternative approach to physical computing. Our reflections on the body of work captured by the portfolio were guided by the following research questions:

(1) How did designers use CV markers to detect physical interactions? What mechanisms did they construct to facilitate the use of CV markers for designing physical interfaces?
(2) How does the practice of physical computing with CV markers compare to the practice of physical computing with electronics and microcontrollers? How did designers make sense of the workings of the physical interfaces they built with CV markers?
(3) What interface materiality did CV markers facilitate?

From these questions, we contribute a series of annotations that reveal the *materials potential* of CV markers for physical computing and building physical interactive systems. Materials potential is described by Barati & Karana [4] as "the possibilities for action offered by a specific material beyond a means for achieving intended qualities in a (proposed) product application". In our case, we look beyond the conventional use of CV markers for tracking objects or augmented reality (e.g., robotic navigation and augmented reality), and suggest how they support an alternative mode of physical computing practice. By abstracting the body of work we assembled in the portfolio, we observed that CV markers, as a design material, offer versatile computational materiality for tangible interactions, afford the use of democratic materials for interface construction, and engage designers in debugging with their own vision as a proxy for computer vision.

In this paper, we describe the portfolio we assembled and detail the annotations that surfaced through our reflection process. We hope to inspire other design educators and interaction designers by offering a series of concepts to consider for taking up the CV-driven approach to physical computing. In addition to these annotations, we also contribute practically to interaction design facilitation by providing the software library we developed for CV-driven physical computing, as well as outlining the curriculum for each of the studios we conducted as a reference for educators.

2 RELATED WORK: APPROACHES TO PHYSICAL COMPUTING

To inform our research, we looked at current practices of physical computing and the common platforms that people use to facilitate it. We discuss how commercial electronic platforms support physical computing, as well as reflect on the rich and growing body of work in HCI that seeks to broaden the material expressions and practices surrounding interactive electronics. We then look at alternative approaches to physical computing that other researchers have proposed and discuss how they inspire our research.

2.1 Microcontrollers: the Dominant Approach to Physical Computing

HCI practitioners have a rich history of creating tools to democratize physical computing. Platforms, such as MetaCricket [30], Wiring [5], Arduino [2], Raspberry Pi [15], and Micro:Bit [14], all aim to be approachable systems by helping makers circumvent certain challenges they encounter when programming embedded hardware. For example, these platforms all offer simplified programming environments for microcontrollers (MCUs), which are often coupled with an ecosystem of compatible and easy-to-use components. Essentially, MCUs facilitate physical computing by enabling designers to build with *electronics*—hardware components like sensors and actuators that rely on electrical signals to transmit information to and from a computer.

The Arduino, for instance, contains all of the necessary electronic components and firmware loaded into a single plug-and-play board. This prefabricated assembly enables makers to skip directly to tinkering with a circuit. A common prototyping tool used in tandem with the Arduino is the breadboard, which is a block of conductive traces and plugs that enables circuit design with jumper wires. Crucially, Arduino also features a cross-platform Integrated Development Environment (IDE) to streamline the process of programming the MCU as a companion to the board. As an open-source platform, Arduino enables other companies and communities to contribute to its ecosystem. Companies like Sparkfun [12] and Adafruit [1] have created a plethora of off-the-shelf modules, accompanying software libraries, and tutorials for others to use. This ecosystem of tools and components lowers the floor as well as widens the walls of what can be built with electronics [39]. As such, microcontrollers and electronics have seemingly become the dominant approach to facilitate physical computing for interaction design at the universities (e.g. the curricula adopted in [13, 48]).

Despite its widespread adoption, MCU and electronics present challenges when facilitating interaction design courses. All of the physical computing platforms described above leverage *electronics* to support physical computing. However, novices struggle with the fundamental properties of electronics (beyond platform-specific features), such as the directionality of electric current and routing circuits correctly [8]. These challenges prompted us to consider how we might circumvent electronics when facilitating physical computing.

2.2 Materials as Electronics

Electronics give physical interfaces the ability to sense and act in the real world. Such electronics are a *material* [6] that designers use—along with other physical materials—to build tangible interactive systems. Beyond using prefabricated electronics as a material for physical computing, HCI researchers demonstrate leveraging other physical materials and making practices to create new electronic forms that contribute to physical computing. Such research weaves physical and computational materiality through crafting practices that come from traditions, cultures, and communities beyond conventional electronics [20]. This "material turn" in HCI [40] shifts the focus from electronics as material [6, 31], to investigate *materials as electronics.*

One such material focus in HCI research is centered on traditional textile practice. For example, the Lilypad [10] is an Ardunio compatible MCU that is designed to be sewn directly onto garments. Instead of jumper cables, it works with conductive yarns which can be stitched into a fabric to form soft circuits on fabrics and wearables. This enables a maker with sewing experience to naturally leverage skills they are already comfortable with when creating computational textiles [11]. Textile Game Controllers by Hartman et al. [21] extends this approach to integrate textile craft into the design space of custom game controllers. They demonstrate a variety of fabric interfaces from felt facsimile of traditional controllers, to rugs that sense players' movements all created with basic sewing techniques. HCI researchers have also translated other textile-related practices when facilitating physical computing. Such approaches include, weaving [16], knitting [kit of no parts], crocheting [37], and darning [25]. Other material practices that leverage electronics making include paper-based crafts, such as sketching circuits [38], or paper cutting and folding [54], or screen printing [45, 52].

These works prompt us to consider not only physical/computational materials that physical computing needs to integrate for TUIs—but also the making practices that are involved in this integration. As we developed our alternative approach to physical computing, we paid careful attention to the tools, materials, and making processes that are around the designers we worked with—and facilitated our design studios with their situations in mind.

2.3 Physical Computing beyond Electronics

Just as different material practices broaden how designers approach electronics for physical computing—different computing practices also offer alternatives to how designers can approach physical computing entirely. Physical computing can include any means of augmenting a physical object to communicate with a computer. One such approach is to sense the natural acoustic vibrations that permeate an interactive object. For example, Savage et al., exemplifies this approach with Lamello [42]. Lamello explores observing surface vibrations already present in 3D printed objects with cheap piezo sensors. To accompany Lamello, Savage et al. created a software that can be trained to map specific vibration patterns to distinct inputs. Similarly, Acoustruments [27] uses a microphone to detect changes in the acoustic vibration and airflow to convey different tangible interactions. The use of acoustic vibration illustrates the rich, unorthodox ways designers can facilitate physical computing.

This rich exploration also includes the wealth of work that features Computer Vision (CV) as a means to detect physical interactions. This body of work is characterized by a camera observing an interaction space and detecting user input via easily identifiable markers. For example, Reactivision [26] is an interactive tabletop library that uses CV to track fiducial markers and map them to interactions. The common template for CV-based tabletops is to have a translucent surface and an IR camera paired with a projector underneath. Slap Widgets [51] expands on Reactivision's approach by using distinct point configurations instead of fiducial markers. These configurations feature internal points that can be moved based on interactions with the widgets (e.g., the slider features a single moving point on a fixed axis). A similar technique is employed by Sauron [41], which places a camera inside of a 3D printed controller shell instead of underneath a table. Sauron enables the programmatic design of 3D printed controllers. Researchers have combined CV markers with paper crafting to create low-cost tangible interfaces and a variety of physical inputs that can be detected with CV [53].

The work surrounding CV-driven TUIs demonstrates the versatility of CV for sensing physical interaction. As a simple graphical element, CV markers enable designers to leverage a wide range of physical materials in the construction of TUIs (e.g. paper [54], 3D printing [41], laser cut plastic [51])—without the need for electronic circuits and wiring. We took reference from this body of related work to develop our CV marker-driven approach to facilitate physical computing.

3 METHOD

We adopted the design research method of annotated portfolios [9, 18, 29] in our work. Annotated portfolios serve a valuable role between formal theory and design instances by enabling researchers to reflect on a collection of artifacts to generate concepts and guidance to further design work [18]. Single artifacts "occupy a point in design space" and embody a "specific configuration of properties". Through a collection of work, annotated portfolios support the "comparison of multiple items [to] make clear a domain of design and relevant dimensions", informing one about "fruitful locations and configurations to develop on those directions". Along with revealing these broader design directions behind a group of artifacts, annotating a portfolio can also provide "abstraction [that] take[s] place from the level of particular artifacts to a higher level in order to produce a knowledge yield that is applicable across a broader range of situations" [29]. Through this abstraction, such annotations can therefore be used as a form of "intermediate-level knowledge" [24] to generate or inform future design activities in a specific area—bridging the gap between isolated artifacts/projects and general design theory [29].

Design researchers have used annotated portfolios in different ways within HCI. For example, Hauser et al. [22] used a portfolio of research products to propose a new way to *frame design research methods* ("doing philosophy through things"). Murray-Rust et al. [32] assembled a large portfolio of public workshops and installations that they facilitated across many years centered around the topic of blockchains. By reflecting on this portfolio, they raise *social considerations and design guidelines for future systems.* Tsaknaki et al. [47] facilitated a workshop with diverse design researchers working on biodata, and reflected on the collective work of the participants to surface *future directions that the research community might pursue.*

In this research, we leverage annotations to surface the *materials potential [4] of CV markers for physical computing*; advocating for their use as an alternative way to facilitate designing and building physical interactive systems. Through these annotations, we reveal the broader dimensions and configurations, as well as constraints, of working with CV markers for physical interaction design for other design researchers and educators.

3.1 A Portfolio of Design Studio Artifacts

The body of work in this annotated portfolio comprises a series of design studios that we facilitated over the last three years. A design studio is a creative and "highly material" site for developing HCI artifacts [7]. As university educators situated in design departments, we leveraged the studio format to introduce CV markers to designers-in-training as a means of building physical interactive systems. It is important to note that the annotated portfolio we present was not our original intention in running these studios. We ran these studios as a means of facilitating tangible interaction design for designers-in-training, while probing the potential of the approach we were developing to facilitate physical computing via CV markers. Through these studios, we had first-hand experience in facilitating other designers to use this approach. The studios also generated a collection of interactive artifacts. "A designed artifact can be seen as a kind of position statement from its designers, not only about what is important to consider in a given design situation, but also about how to best respond to those considerations." [18] In this sense, each artifact produced by the studio embodied an authentic translation of CV markers and other materials into prototypes that address specific real-world situations identified by the participants.

We assembled these studio outcomes into a portfolio of artifacts that rely on CV markers for physical computing. As facilitators of these studios, we (as a research team) had unique access to the design process taken by each project, as well as a comprehensive overview of how CV markers were employed within and across each studio. We, therefore, leveraged these design studios—through the portfolio—as a research site to reflect on the key questions raised in subsection 1.1 to surface the material potentials of CV markers for physical computing.

4 PORTFOLIO: THREE DESIGN STUDIOS SUPPORTED BY BEHOLDER

The portfolio we present in this paper includes 15 projects that were produced from 2020 to 2022 (Figure 2). This work was done in design studios conducted by two separate facilitators at different locations: Clement Zheng at NUS (National University of Singapore) and Peter Gyory at CU Boulder (University of Colorado, Boulder). Throughout this period, the two authors kept in constant contact with each other through online meetings—sharing progress and insights about the projects they were facilitating. As such, the different pieces presented in this portfolio can be seen as part of a larger process. The design studios form a series of courses that we reflected and iterated on; rather than separate efforts that developed independently.

The work in this portfolio is organized into three distinct design studios. Each studio is a long-term design engagement with a specific brief oriented around building tangible interfaces. These studios are all supported by *Beholder*, an evolving software library for detecting CV markers that we developed and continuously refined across each studio we ran. The three studios incorporated into the portfolio are:

(1) DIY CV Interfaces: An 11-week design studio class with undergraduate industrial design students at NUS.
(2) Tinycade: A year-long design studio by CU Boulder researchers designing alternative video game controllers.
(3) CV Arcade: A 13-week design studio class with undergraduate industrial design students at NUS.

We included all studio projects into the portfolio that we reflect on with the exception of two projects in Studio 3 (CV Arcade). We excluded these two projects as the designers did not make use of CV markers for building the interactive system, and opted instead to use other platforms to develop their ideas. We describe the portfolio in the following sections. First, we detail the *Beholder* software library that we developed to support the studios, then, we outline each studio in terms of its design brief, facilitation tools, and outcomes. This portfolio and constituent projects are also available for others to browse via an interactive website[1] that we provide in the supplementary materials.

[1] https://project-beholder.github.io/markinginteractionsportfolio/

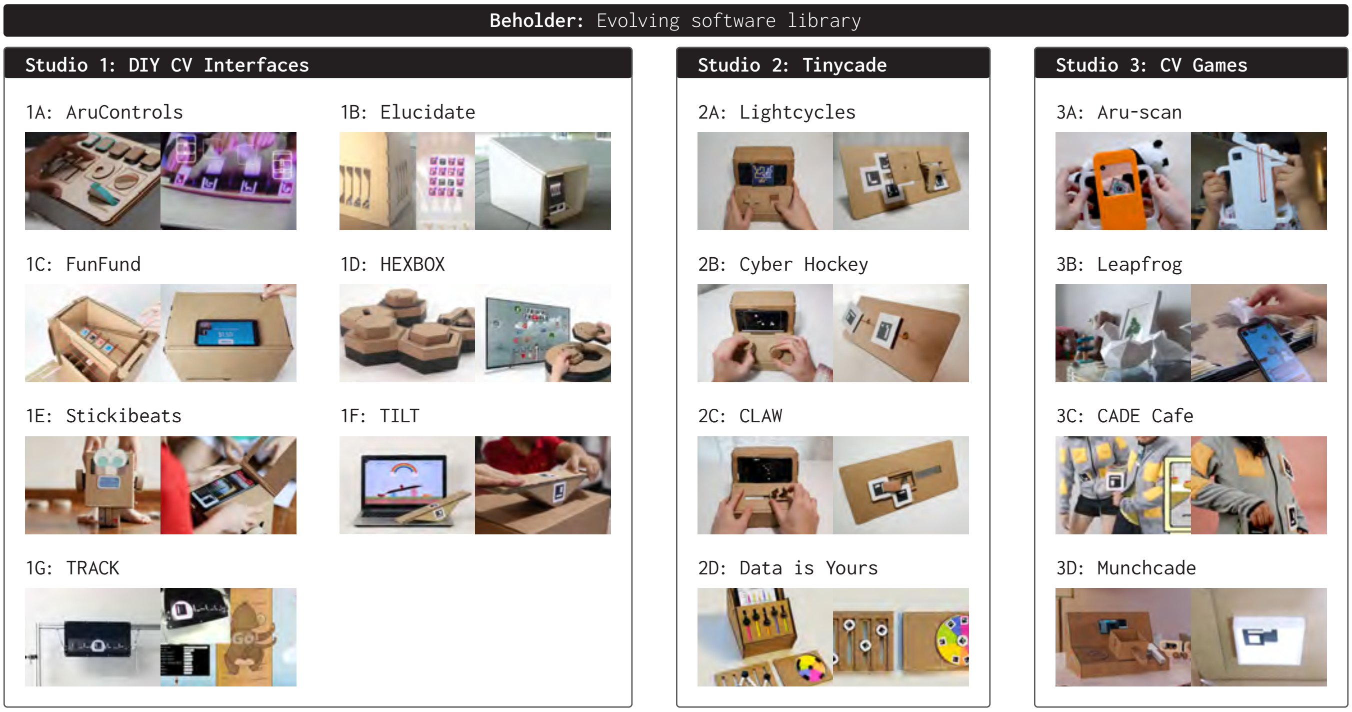


Figure 2: Overview of the 15 projects created using *Beholder*. The 15 projects are the outcomes from three distinct design studios. We used an annotated portfolio methodology to uncover insights on how CV markers can be used for physical computing. See Sections 4.2, 4.3, and 4.4 for more details on each individual project.

4.1 CV Markers and Beholder Software Library

We developed a general-purpose software library to detect CV markers. *Beholder* is built with JavaScript and intended for web applications such that it can run on a diverse range of computing platforms (e.g., mobile devices, tablets, and personal computers). To use *Beholder*, designers simply add and initialize the library in their web application project at the head of the web page. *Beholder* library runs in the background of a web application. It offers a few convenient features, such as selecting which camera to use, changing the camera resolution, and flipping the camera view. *Beholder* also injects a debug overlay onto the webpage that displays the current camera feed, shows which markers are detected, and provides inputs for adjusting detection parameters (e.g., minimum marker perimeter or image contrast). The library is also compatible with novice-friendly coding platforms, such as p5.js, as well as other web development frameworks used by professional developers (e.g., HTML5 Canvas, WebGL). This library is available through the Node Package Manager (NPM) ecosystem and can be included on any web page. When introducing this library to designers, we also created a template for them to build upon using the online web development platform Glitch.[2]

Beholder is designed to detect ArUco markers [17]. As a computational material, an ArUco marker provides useful properties including a unique ID number and 2D position data of its corners (Figure 3). The 2D position data can be used to easily derive the center of the marker as well as the marker's rotation angle. As demonstrated by [53], CV markers also offer its *presence* (i.e., CV markers being detected or not detected) as an interaction technique (Figure 9). *Beholder* computes these properties in the background for each marker automatically and ties them to the marker ID as meta-data (Figure 3). Designers can access this data through *Beholder* library methods. *Beholder* also offers a debug window that displays the live video feed (Figure 4), which can be used to view

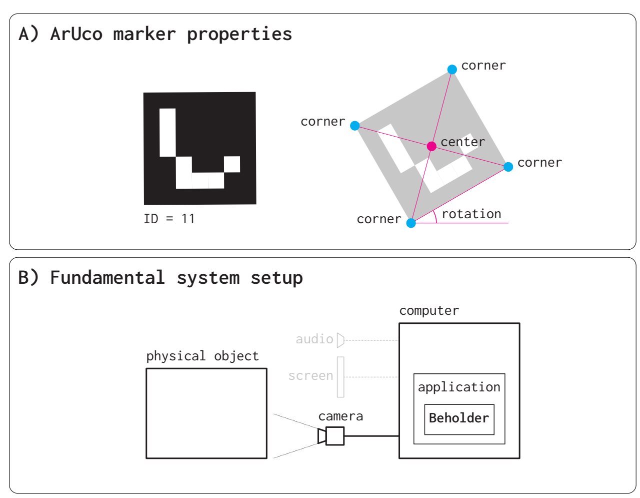


Figure 3: A: An ArUco marker with the ID of 11 demonstrating the meta-data provided by *Beholder*. B: Fundamental system setup for using *Beholder* to instrument a tangible interface.

[2]https://glitch.com/~diy-ar-beholder

the CV markers. Detected CV markers are outlined in pink in the live feed, along with information on the marker's unique ID.

We subsequently use "CV Markers" in this paper to refer specifically to ArUco markers. We used ArUco markers in this work as it is conveniently integrated into many open-sourced software platforms (e.g. OpenCV [35], JavaScript). It is important to note that while we focus on ArUco markers, many of the computational concepts that they demonstrate can be translated to other CV marker libraries that provide a similar set of properties (e.g. Reactivision [26], AprilTag [34]).

Figure 3b illustrates the overall system architecture of physical interfaces built with Beholder. The computer vision setup forms the physical requirement of using Beholder: a camera connected to a computer. This can take the form of a web camera attached to a laptop computer or a smartphone which provides an all-in-one mobile package.

We developed different packages for *Beholder* based on the needs of each design studio we facilitated. While we do not see it as a key contribution of this research, we provide the library, as well as the examples we developed for each studio, as separate resources for others to follow.[3]

Figure 4: The debug window of *Beholder* and how it is used for Stickibeats (project 1E)

4.2 Studio 1: DIY CV Interfaces Studio

DIY CV Interfaces was an 11-week design studio class that ran starting from August 2020 with 21 undergraduate industrial design students. In this studio, students were challenged to develop DIY computer interfaces for everyday activities, anchored on the themes of *work*, *learn*, and *play*. This brief was provoked by the fragility of the global supply network highlighted by the COVID-19 pandemic and the increased interest in DIY activities that the situation sparked. In studio 1, we created a simple web application with the *Beholder* pre-loaded for this studio. This application served as a template for designers to remix, as well as a debugging tool to inspect CV marker behavior (i.e., view marker meta-data in real time). This example web application contained the p5.js creative coding environment where designers developed the digital aspects of their prototypes. As a warm-up activity, we got designers to individually explore and demonstrate three different physical interactions that can be detected by using CV markers.

[3]https://www.npmjs.com/package/beholder-detection

Designers then worked in teams on a studio project for approximately 8 weeks. The studio culminated in a demo day where teams showcased their work for visitors to interact with.

4.2.1 *Outcomes.*

We included all seven projects from this studio in the portfolio (Table 1). These projects covered a wide range of contexts, such as alternative inputs for computers and educational kits for children.

4.2.2 *Reflection on Studio 1 proceedings.*

The warm-up activity served a crucial role in shaping the design participants' approach to CV markers as a material for making physical interfaces. Notably, TRACK's (project 1G) slider was directly informed by a mechanism that the designers built during warm up. This activity enabled designers to "play" with CV markers and *Beholder* without any context; first coming to terms with the affordances and constraints of this set of materials and tools before deciding on a real-world situation to address.

It is likely that this playful start to the studio, as well as the DIY focus of the studio brief, led many projects to be framed as play or games—even for projects that addressed work or learning activities. With the DIY focus, we also observed designers use physical materials that were immediately available to them as industrial design students. All teams used sheet materials like corrugated cardboard, foam core boards, and acrylic. Apart from manually cutting and assembling these materials for their prototypes, teams also relied heavily on the laser cutter that was available to them in the university workshop.

It is also important to note that programming the digital components of each project in p5.js was the most challenging aspect for all design teams in this studio as the industrial design students were mostly unfamiliar with programming. We addressed this challenge as facilitators by providing dedicated office hours for programming support.

4.3 Studio 2: Tinycade

Taking cues for the process and outcomes from Studio 1, we honed in on play as a good context to apply CV markers for tangible interactions. Designing for play encourages designers to focus on the experience of the end user without concerning too much over practicality. We also wanted to explore the potential of mobile devices as an "all in one" platform for marker-based physical computing which projects (1B, 1C) demonstrated the potential of. With these goals in mind, we organized Studio 2 to run within our research group. These researchers were already familiar with *Beholder* so we did not need to provide any new examples for them. The group of participating designers for this studio was only 8, and all were working together to create a Game controller platform that works with a smartphone as the only electronic component, and multiple games to showcase its potential. Unlike the other studios which took place during 13-week courses, Studio 2 occurred over the course of a year without a concrete schedule. Each researcher averaged around 2-4 hours per week towards the project during this studio. The main motivating deadline for the project initially was the ALT.CTRL.GDC contest of the Game Developers Conference (GDC),

Table 1: An overview of Studio 1's projects ($n = 7$). Each row showcases a picture of each individual project, a short description, and the physical interactions detected by CV markers

Studio 1 Projects	Project Description	Marker Mediated Interactions
	**AruControls (1A)**: AruControls is a desktop software application that maps DIY controller modules to computer keyboard events. For example, users can map detecting a CV marker to a specific keypress (e.g., delete). The team demonstrated a number of physical inputs with different interactions, and used their application to control video games or digital audio workstations on the computer.	Rotating Pressing Sliding
	**Elucidate (1B):** Elucidate is a box-like ambient interface that translates changes in light and wind to sound. Slits and flaps are placed at the opening. As light or wind interact with these slits and flaps, shadows shift around in the inside of the box hiding and revealing CV markers on the inside. A smartphone observes these CV markers and maps the change in detected CV markers to different sound samples.	Detecting light/wind
	**FunFund (1C):** FunFund is a cardboard coin bank designed to help children learn about tangible and digital money. Children can deposit coins into the coin bank, and a cardboard mechanism will sort the coins into their specific denominations. This is logged in real time into a digital "savings" application by a smartphone placed on top.	Identifying objects
	**HEXBOX (1D):** HEXBOX is an alternative video game platform with interchangeable inputs designed with cardboard and household materials like elastic bands. Each input offers a different action (pressing, sliding, squeezing) and provides haptic feedback. CV markers are positioned at the base of each input, and the game makes use of marker presence, position, and rotation to control game actions.	Rotating Sliding Grabbing Pressing
	**Stickibeats (1E):** Stickibeats is a cardboard robot designed to help children learn about musical rhythm. This robot has a single large wheel with four different tracks on which CV marker stickers can be placed. The wheel turns as children move the robot across the floor, and the smartphone on top plays the corresponding sound sample for each marker that appears.	Rolling
	**TILT (1F):** TILT is a simple cardboard controller that takes the form of a see-saw. Design students developed three cooperative games that make use of this controller. The see-saw controller facilitates tangible interactions that require two people. CV markers are placed around the controller and are used to determine the orientation of the controller.	Tilting Flipping
	**TRACK (1G):** TRACK is an occupational therapy device to exercise the shoulders. It comprises of a pulley system connected to a railtrack lined with CV markers. As the exercise progresses, a white background shifts back and forth, revealing the CV markers one at a time. This system connects to a simple video game designed to motivate elderly physical therapy patients.	Sliding

which is a showcase of alternative game controllers available to all in attendance at GDC.[4]

The designers in this studio created Tinycade, a DIY platform inspired by the arcade machines from the 1980s that exclusively features the use of commonplace materials (e.g., cardboard, paper, toothpicks). To fit this form without requiring users to purchase more electronics, the designers employed a smartphone for dual purposes: the rear-camera on a smartphone as a detection window and the screen as an interactive display. Tinycade features two mirrors on the inside, which have the functionality of a periscope to redirect the vision of the phone camera to the back of an interchangeable control panel. From this platform, three games and one educational toolkit were developed.

4.3.1 *Outcomes.* We included four projects built for Tinycade over the course of one year, three games and one educational kit (Table 2). These projects were designed around the strict constraints of the Tinycade form factor.

[4] https://gdconf.com/alt-ctrl-gdc

Table 2: An overview of Studio 2's projects ($n = 4$). Each row showcases a picture of each individual project, a short description, and the physical interactions detected by CV markers

Studio 2 Projects	Project Description	Marker Mediated Interactions
	**Lightcycles (2A):** Lightcycles is a 4 player competitive area control game inspired by classic arcade racing machines and TRON. The controller features a D-pad and toggle switch which control the player movement toggle a trail on and off which can be used to cause competing players to crash.	Pressing Tilting
	**Cyber Hockey (2B):** Cyber Hockey is a two player abstract sport game played with a single Tinycade where the goal is to score 10 points first. Each player character is comprised of a goal and a paddle which can be rotated to protect the goal and launched to hit the ball. The input uses a single marker on a spoke and works as a knob, joystick, and button all at once.	Rotating Tilting
	**CLAW (2C):** CLAW is a single player arcade shooter where the player must survive waves of enemies flying down the screen. The controller is a claw mounted on a slider. By moving the slider they player sets the position of the ship on the screen, and pinching the input causes the in-game claw to launch out and grab an enemy.	Sliding Pinching
	**Data Is Yours (2D):** Data Is Yours is a set of interchangeable visualization panels designed to help teach data visualization literacy to children ages 5–11. Each visualization panel consists of a single chart (bar, line, and pie) which have CV markers on the back. Children use the chart panels to enter their own data and author visualizations. When the chart is adjusted, the changes are reflected on a phone app.	Rotating Sliding

4.3.2 Reflection on Studio 2 proceedings. The constraints for this studio meant that there were very subtle variations between the projects. Ultimately the designers settled on a standard shape for Tinycade and used interchangeable control panels to support different interactions. In particular projects, 2A, 2B, and 2C all take the same exact template shape. Over the course of the year, the Studio 2 designers produced many prototypes and ended up relying heavily on lasercutting in order to precisely cut their cardboard components. This use of advanced fabrication tools diverges from the initial goal of keeping accessible to players and designers with limited resources.

4.4 Studio 3: CV Games Studio

CV Games was a 13-week design studio with 15 undergraduate industrial design students that ran from January 2022. This studio was inspired by the Tinycade platform (Studio 2), as well as the many game-based projects from DIY CV Interfaces (Studio 1). In this studio, designers were challenged to develop new DIY video game platforms for a specific audience. As with studio 1, we facilitated a warm-up activity to introduce students to CV markers for tangible interaction design. Each student was given a Tinycade kit (the same cardboard frame used in Studio 2). We developed a simple rock-paper-scissors game for this platform, and students were tasked to design a new controller to allow users to play this game. This activity resulted in 15 different controllers that work on the same platform—demonstrating what designers can aspire toward as project outcomes for this studio.

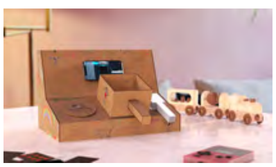
Designers worked in teams for their studio project. Compared to Studios 1 and 2, we had the opportunity to visit a children's makerspace in the middle of this studio, and teams playtested their game concepts with members of that makerspace. As with studio 1, this class culminated in a demo day. This demo day took the format of a games festival where visitors dropped by to play and interact with the different projects that teams developed. Learning from Studio 1, we supported participants by providing office hours to scaffold the programming required for their projects.

4.4.1 Outcomes. We included four projects from this studio in the portfolio (Table 3). Two projects pursued other avenues and did not use CV markers[5].

4.4.2 Reflection on Studio 3 proceedings. The Tinycade kit served as a good platform to introduce students to CV markers and *Beholder* for the specific purpose of developing tangible games. More importantly, collaborating with a children's makerspace was a critical facilitation component that encouraged designers to go beyond the kit provided and develop specific tangible interfaces that considered how different people play games. For example, CADE Cafe (project 3C) was inspired by the full body interactions that the

[5] One project opted to explore a smartphone's capability for sensing movement through the inertial measurement unit, while another project opted to explore holographic projections with a smartphone using the pepper's ghost technique. These point toward certain constraints of the CV marker-based approach for physical computing which we discuss at the end of this paper (subsection 5.5)

Table 3: An overview of Studio 3's projects ($n = 4$). Each row showcases a picture of each individual project, a short description, and the physical interactions detected by CV markers

Studio 2 Projects	Project Description	Marker Mediated Interactions
	**Aru-scan (3A):** Aru-scan is a CV marker scanner smartphone app built for preschool children. Inspired by observing young children playfully interacting with barcode scanners at the grocery store, Aru-scan enables caregivers to map CV markers to specific images and sound files. The media is played back when the app scans the correct corresponding marker. Aru-scan includes a 3D printed case for the smartphone with a mechanical shutter for the camera lens.	Identifying Objects
	**Leapfrog (3B):** Leapfrog is a DIY paper diorama of a pond scene with paper animals. This diorama transforms into a mobile game with the frog as the protagonist. The characters in the game are controlled by their corresponding paper animals. Each paper animal is instrumented with CV markers at their base, and interactions with the animal are identified up by the smartphone. This game was inspired by hobbyists who craft and display their own work, and designed this game for papercraft enthusiasts.	Pushing
	**CADE Cafe (3C):** CADE Cafe is a game platform that facilitates full-body interactions inspired by commercial game platforms like the Kinect and Wii. The controller consists of two clothing jackets with velcro patches placed at different locations (e.g., shoulders, forearm, chest), as well as a series of large fabric patches with CV markers printed on them. The platform offers a series of minigames where players compete against each other. Games are controlled by manipulating the CV markers placed on the body.	Rotating Sliding Flipping
	**Munchcade (3D):** Munchcade is a miniature cardboard kitchen that controls a video game where players run a food truck. The kitchen offers a variety of tangible interactions that mimic cooking scenarios (e.g., chopping vegetables, controlling oven temperature, and stirring a pot). The set-up for this controller resembles the Tinycade platform (studio 2), and each input is detected via CV markers placed on the interior of the controller.	Rotating Pushing Shaking

designers observe children make while playing with an early prototype; while Aru-scan (project 3A) was inspired by the observation of young children scanning barcodes at a grocery store.

4.5 Organizing and Annotating the Portfolio

We assembled the portfolio and began work on annotating it after the conclusion of Studio 3. Besides the studio facilitators, we also invited three other researchers from CU Boulder who had prior experience with *Beholder* as external reviewers of the portfolio. We aimed to balance our approach in reflecting on the work presented through this mix of studio facilitators who were intimately familiar with each project's process and external reviewers who could comment on the project outcomes objectively.

To support our reflection process, we organized individual projects in the portfolio on a Miro workspace (an online whiteboard platform). This includes the title and description of each project, as well as a video demonstration of the interactive prototypes. Additional materials such as project documentation and images were provided separately through an online repository.

We began the annotation process by onboarding all reviewers during an online meeting to the main objective of the research (probing CV markers for facilitating physical computing) and the key research questions driving our reflections. We used our research questions to scaffold our reflections on each project:

(1) *How did designers use CV markers to detect physical interactions? What mechanisms did they construct to facilitate the use of CV markers for designing physical interfaces?*
We deconstructed each project into individual physical interactions (e.g. pushing, rotating) and observed how CV markers facilitated detecting this interaction. We paid attention to the broader physical construct (mechanism) that CV markers were used in for interaction design.
(2) *How does the practice of physical computing with CV markers compare to the practice of physical computing with electronics and microcontrollers? How did designers understand the mechanics of building physical interfaces with CV markers?*
We compared CV marker mechanisms to electronic components that achieved the same function. We examined the set up that CV markers required (including CV related constraints) to their electronic counterparts (considering physical set up as well as software set up). From our role as studio facilitators, we also reflected on how designers approached designing and building physical interfaces with CV markers, including how they "debugged" their constructions.

(3) *What interface materiality did CV markers facilitate?* Following [20], we reflected on the materiality of the physical interfaces that designers built through their *physical*, *computational*, and *craft* aspects. We took note of the physical materials that designers used to build their prototypes, paying attention also to the materials that we introduced as studio facilitators. We observed how designers built interfaces to facilitate detecting interaction events and the design choices they made to accommodate CV. We also observed the making or crafting practices that designers used to build their interfaces.

Our annotation process was an iterative process of asynchronous individual reflection on the work, and synchronous group sharing and discussion. We created individual areas on the Miro workspace (Figure 5) for each researcher to individually note down what they observed in each project based on the research questions. We also conducted online discussions with the whole team to share findings. During these meetings, we coalesced similar observations across projects into broader concepts. The insights we discuss in the following section stem from this iterative reflection and discussion process.

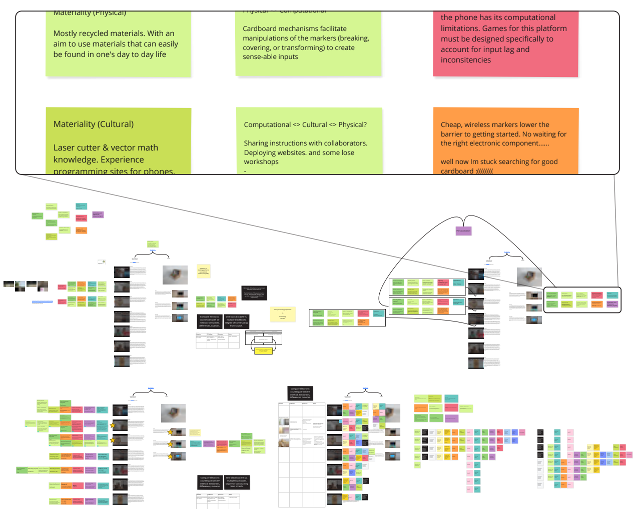


Figure 5: A screen capture of the Miro whiteboard we used to annotate the projects in the portfolio. We zoom in on a small section of the board to reveal example reflections that were color-coded based on the theme.

5 DISCUSSION: THE POTENTIAL OF CV MARKERS FOR PHYSICAL COMPUTING

In each studio, designers were able to build functional interfaces that supported various tangible interactions. They accomplished this feat by combining CV markers with physical materials in specific ways for the computer's camera to detect. In reflection, our approach of facilitating physical computing with CV markers can be broadly defined as "marking material interactions with computer vision". We unpack this broad approach by discussing the insights that surfaced from the annotated portfolio.

Table 4: A comparison of the sensors a designer might use when implementing an interaction with electronics, and how we observed those same interactions implemented with CV markers.

Interaction	Electronic Component(s)	CV Markers Analogs
Rotation	Potentiometer, rotary encoder	Marker attached to the rotating wheel reporting its rotation
Sliding	Linear potentiometer	Marker attached to a slider reporting its position
Pressing	Contact switch	Uncovering a marker for detection (presence) when an object is pressed
Shaking	Accelerometer	Marker attached to a shaker reporting its position
Flipping	Accelerometer	Uncovering a marker for detection (presence) when an object is flipped

5.1 CV Markers Mark Identity & Interaction Constraints

Electronic sensors translate tangible interactions from physical to computational materiality [49]. A contact switch, for example, translates a physical movement to a mechanical opening and closing of electrical contact. When this switch is wired up to a microcontroller, it sends a "high" or "low" voltage signal that corresponds to physically pressing and releasing the switch.

CV markers serve a similar function as electronic sensors in the paradigm of CV-driven physical computing. As a physical material, CV markers are in essence simple 2D graphics, and therefore can be conveniently applied as a physical material—such as printing it out on paper and sticking it to cardboard (e.g. studio 2 projects), soft wearables (e.g. project 3C), or paper (e.g. project 3B). As a computational material, CV markers provide data to a computer vision system, such as ID, position, and rotation for ArUco markers Implicitly, CV markers also report if they are present or absent. See Section 4.1 for more details.

We identified all of the tangible interactions observed in the portfolio's outcomes and organized our findings into Figure 6. Despite being a single type of physical/computational thing, this figure illustrates how designers used CV markers in versatile ways to support different functions of physical computing. As educators who have taught physical computing with electronic kits, we observed analogous uses between CV markers and electronics for detecting tangible interactions (Table 4). Rather than using a kit of individual, different parts, the portfolio findings highlight how it is possible for one type of "material" (i.e., CV markers) to accomplish several functionalities.

Other researchers have previously demonstrated using CV markers in similar ways, such as tracking the rotation of a physical token [53]. Extending from these related work, we observed some novel uses of CV markers for tangible interface design in the portfolio—notably, using markers to detect changes in light and wind (project

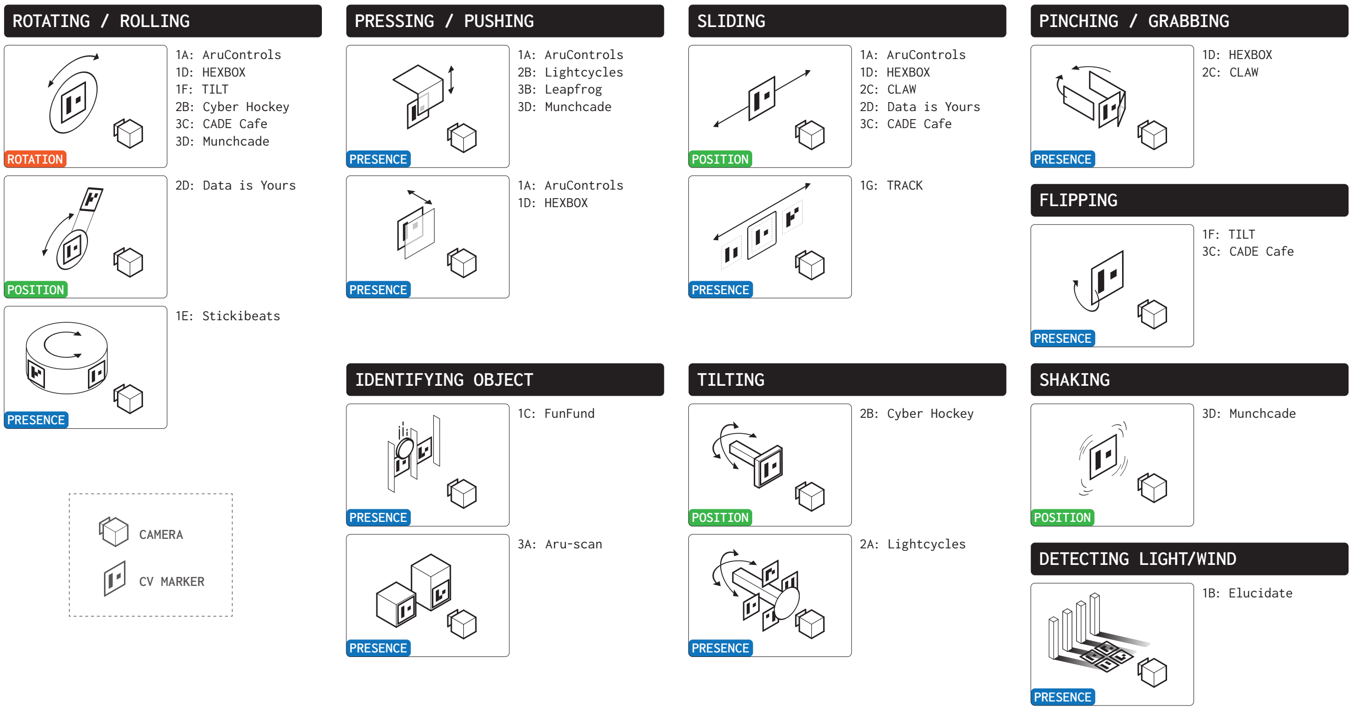


Figure 6: The range of CV marker configurations we observed in use through the portfolio to detect tangible interactions. Designers often found multiple ways to detect each desired interaction. Listed next to each technique illustrated are the projects that used them.

1B). More importantly, our objective with consolidating portfolio projects into Figure 6 is to gain a broad overview of what designers perceived CV markers can do. We are primarily interested in the affordances that CV markers offer for tangible interaction design, and took note of the different ways that designers connected marker properties to detect physical interactions in the studios.

By examining the overview that Figure 6 offers, we observed that CV markers offer two types of "marks" that enable versatile tangible interaction sensing: (1) marking an object's identity, and (2) marking physical interaction constraints.

First, CV markers mark specific physical objects and are able to identify them through their ID number. Though CV markers are commonly used as trackers for augmented reality systems, earlier work like Reactivision [26] demonstrates how object tracking can participate in tangible interactions. Using identity markers, Reactivision enables the computational system to differentiate between each input within a complex system. Further, this identification can be accomplished with a high degree of tolerance. Compared to systems like Sauron [41] and Nintendo Labo [33], the computational system does not need to know the expected positions of the markers. Drawing parallels to electronics again, a CV marker's ID functions like an "input pin" that connects a sensor to an MCU. Each ID can be used to define a relationship between the marker and the object/interaction to that it is attached to. Unlike electronics, establishing the relationship does not require a physical tether. For example, in Cyber Hockey (project 2B; Figure 7), marker $ID = 1$ and marker $ID = 2$ are used to differentiate and identify two similar inputs.

Second, CV markers mark physical interaction constraints. These constraints are achieved in the physical mechanisms that designers build, and enable CV markers to report on a specific tangible interaction. For instance:

(1) The position data of a CV marker constrained to move along a track can report the location of a slider (`projects 1A, 1D, 2C, 2D, 3C`)
(2) The rotation data of a CV marker constrained to rotate about an axis can report the direction of a knob (`projects 1A, 1D, 1F, 2B, 3C, 3D`)
(3) The presence of a CV marker constrained to be revealed or hidden based on the position of a cast shadow can report on the direction of a light source (`project 1B`)

The wealth of data that a single CV marker offers as a computational material also enables designers to naively define new tangible interactions based on constraints that are otherwise challenging to realize with electronics. Each input for Cyber Hockey, for example, is constrained to both the tilt-like (i.e., a joystick) and rotate-like (i.e., a knob) mechanisms. In contrast, creating such a similar design mechanism with electronics will likely require mechanically and electrically coupling these two components (joystick + knob). Yet, with CV markers, this hybrid function input is easily achieved with CV markers as a single marker provides both 2D position and rotation data (Figure 3).

These two forms of marks—marking physical identity and marking physical constraints—work in concert to enable designers to build interfaces capable of detecting both a large number of interactions (e.g. projects 1A, 1D), as well as varied types of interactions as illustrated in Figure 6.

5.2 CV Markers Mark Interaction Events in the Physical World

In addition to building the physical interface and defining and constraining physical interactions with CV markers, designers also need to write software that connects physical interactions to an application (e.g. a video game). Throughout the different studios, we observed that designers "offload" programming software logic by strategically situating CV markers on the physical interface itself. These CV markers report interaction events that occur.

Figure 8a demonstrates two different ways to construct a functional slider with CV markers. In the first scenario, a CV marker is attached to the moving slider; the position of the CV marker indicates the position of the slider. To detect if the slider has reached a certain point, the designer will have to define a conditional logic in software:

```
if (marker.position.x > 300) { ... }
```

In the second scenario, two CV markers are placed at the ends of the slider (Figure 8b). The CV marker is revealed and detected when the slider reaches a specific point. In this second case, the CV markers report on a specific event—whether the slider has reached a point marked by the CV marker. In this scenario, the designer defined an interaction event by manipulating the physical location of a CV marker, thus engaging with programming in the physical world.

While both scenarios accomplish the same functionality, each scenario serves different interaction needs. The CV marker provides a continuous slider position in the first scenario, while the CV markers in the second scenario provide checkpoints to approximate slider position. CLAW (project 2C; Figure 8a) and TRACK (project 1G; Figure 8b) are two examples that illustrate this difference. CLAW adopts a moving CV marker and detects the slider position with the marker's position data. CLAW's slider is used to control the exact horizontal position of the player's character in the video game. The continuous position data of a marker is a direct approach to connecting physical interaction to game behavior. On the other

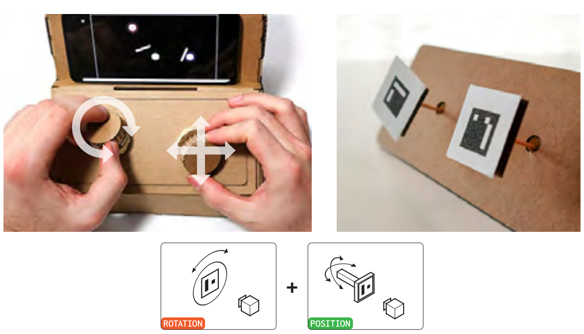


Figure 7: Each input on the Cyber Hockey controller uses only a single CV marker to sense multiple interactions. Players can rotate the input (and by extension the CV marker) to rotate their character in game. Tilting the input acts like a joystick which moves the players character, and is sensed by the position of the CV marker.

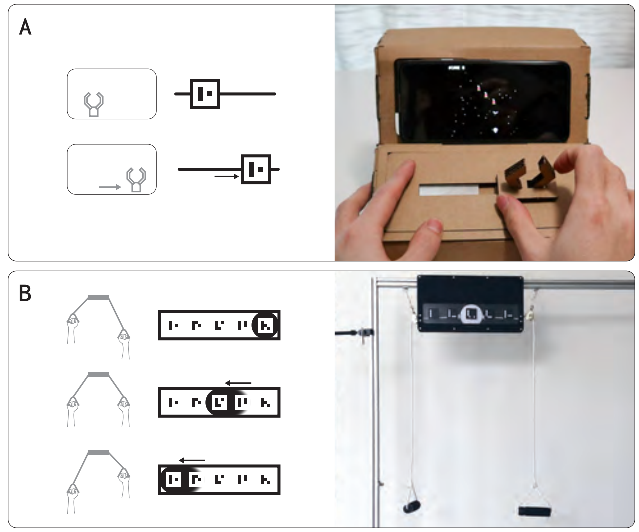


Figure 8: These projects showcase two different approaches to sense slider interactions. A: CLAW (project 2C) uses a single CV marker on the bottom to track the slider's position in pixel values. B: TRACK (project 1G) uses an array of markers in a line that become detectable as a white background moves behind them, discretely marking the position of the slider.

hand, TRACK employs 5 CV markers spaced along the slider's path: one marker marks the center point, two markers mark the respective midpoints, and two markers mark the respective ends. TRACK is designed to quantify the repetitions of a shoulder pulley activity for physical therapy. The physical movements occur quickly, and TRACK's designers adopted the "event marking" strategy with static CV markers to avoid the motion blur of a moving marker that disrupts CV detection.

Marking interaction events was also accomplished through the use of "marker completion" (i.e., to intentionally reveal or conceal markers in plain sight). Beholder can only detect a CV marker if it has sufficient "whitespace" around its border, or if the entire square pattern is revealed. These simple detection properties add to the versatility of CV markers as a material that mediates between the physical and computational worlds. We observed designers employ various means of partially disrupting marker detection to sense physical interaction, as illustrated in Figure 9.

Figure 10 illustrates more examples of how different projects in the portfolio deconstruct tangible interactions into events that can be "marked". For instance, the triggering point of a button can be adjusted by shifting the position of the marker so that it is revealed earlier or later during the pressing interaction (e.g. Project 1A: AruControls); similarly, the triggering point of a turning wheel can be modified by the shifting the position of the marker along the rim (e.g. Project 1E: Stickibeats). Marking interaction events with CV markers in the physical world enables designers to define and modify system logic in-situ. We believe that this facilitated the construction of the physical interface for the industrial design students that participated in Studio 1 and 3. In contrast to the challenges they encountered in programming the digital aspects of their projects, we observed that this tangibility of modifying the interaction logic of the physical interface enabled designers to

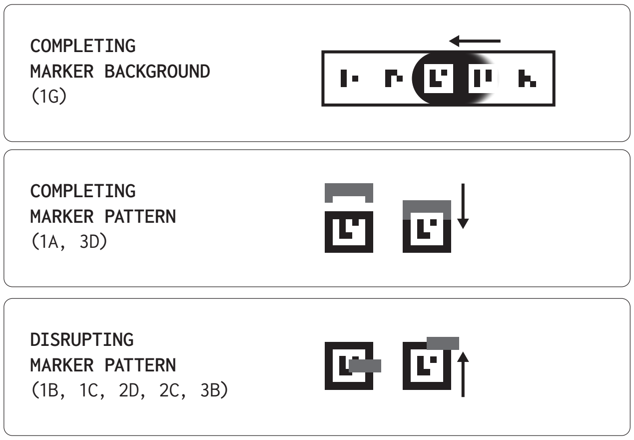


Figure 9: We observed three main approaches to applying CV marker presence for interaction. From Left to Right: altering the background to reveal or hide a marker, dividing a marker into multiple parts to be mechanically separated or joined, and disrupting a marker pattern with another material.

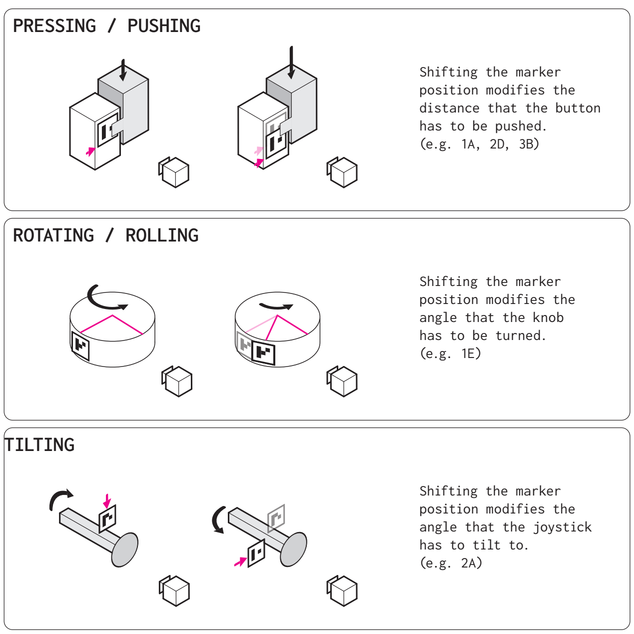


Figure 10: "Marking events" with CV markers. Pressing/Pushing: the CV marker's position determines the distance to be pressed/pushed. Rotating/Rolling: the CV marker's position determines the angle to rotate by. Tilting: the CV marker's position determines the direction to tile to.

quickly iterate and improve on their prototypes. We continue to elaborate on the embodied benefits that CV markers provide as a tangible interaction design material in the following section.

5.3 Debugging CV Markers with Human Vision

CV markers do not fail invisibly, unlike electronics [8]. When viewing the camera feed with the Beholder library, designers can immediately see if the CV marker they are testing is detected properly. This helps designers reason through any system errors they encounter and identify if the error is in their code or in the construction of the physical interface—a challenge that Booth et al. [8] identifies with electronics-driven physical computing.

The detection algorithm that a CV system uses to detect CV markers is a black box. Yet, the input into this black box (a camera feed), and output of this black box (detected markers and its properties from the image), is naturally understood by sighted people. For instance, a camera feed of a rotating marker is parsed by the CV system which returns the same image that a designer sees, along with labeled information such as the marker's angle of rotation (Figure 3).

On the other hand, electronic sensors convert physical phenomena into intangible electrical signals that we can only "see" through multimeters, oscilloscopes, or computer consoles. Even then, these signals contain abstract protocols for a computer to read (e.g. binary data packets), and require layers of software libraries and programming before it is parsed to data that is usable for interaction design. For instance, for an accelerometer to detect rotation, it sends electrical signals to an MCU via the I^2C protocol, which decodes the signal via a library into X, Y, and Z axis readings, which can be computed into an angle of rotation with trigonometric programming functions (Figure 12).

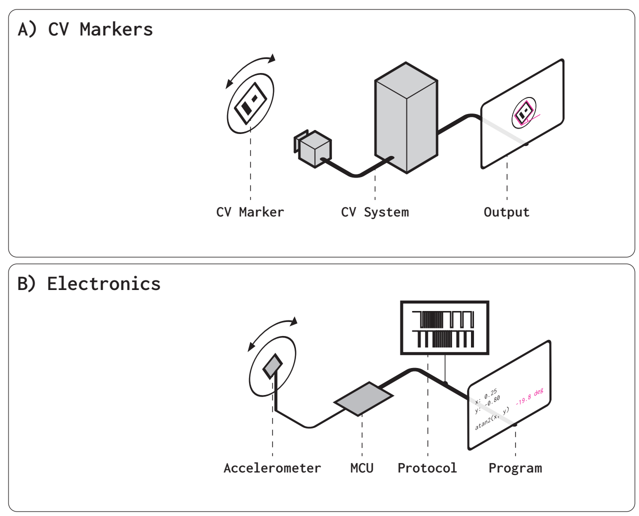


Figure 11: The different flow of information in CV marker systems and electronics systems. A: A CV marker is observed in the real world by the camera, processed by the CV algorithm, then presented on the computer screen as an image with marker properties labeled. B: An accelerometer sends electrical signals to a MCU, which then communicates to the computer. A software library is required to read the data from the accelerometer, and a program parses the data into the accelerometer's rotation angle.

By exposing input and output in a manner analogous to human vision, we argue that CV markers instill designers with confidence that "what they see is what they get". Throughout all the design studios, we observed some designers rely entirely on their sight to make design decisions on the construction of their interaction mechanisms—only checking with the computer system when they have finalized their design. CV markers thus support a "reflective conversation" [44] with other materials in the construction of TUIs. They enable and encourage designers to fluidly "move" between interaction design ideas and (literally) "see" across physical and computational materialities [43].

We are by no means claiming that CV markers are therefore the better approach to facilitate physical computing. CV systems and the algorithms they use are as much a black box as electronic signals and protocols. There are many factors that affect this black-boxed CV marker detection system (e.g. in *Beholder*). These include lighting, camera field of view (FOV), image contrast, and camera resolution. These factors have a big influence on the reliability of marker detection—and are "immaterials" [46] that designers need to understand in order ultimately design functional TUIs with CV markers. Even as they instill designers with confidence, we also observe the same designers express frustration at unpredictable detection by the CV system, such as when environmental conditions change.

5.4 Materiality of Physical Computing With CV Markers

In this portfolio, we presented a range of projects that apply CV markers to TUIs in a variety of ways. During our facilitations, we introduced the markers on printed paper, which can be cut and glued to any surface. Throughout the three studios, we had several designers liken the black and white square graphics of the CV markers to QR codes upon first seeing them. While not equivalent to the ArUco markers, QR codes turned out to be a helpful analog. QR codes are physical elements that are specially designed for a computer to read. This comparison conveniently served as an introduction point to computer vision. The relationship between QR codes and CV markers is most directly demonstrated by AruScan (project 3A) which is built around the scanning interaction associated with QR codes.

5.4.1 The Intangible Materials of Computer Vision. We discussed in the previous section that working with CV markers requires designers to attend to intangible concerns like environmental lighting. To elaborate, we observed that these "immaterials"—such as light, shadows, and space—are as relevant to CV as the physical materials that make up the tangible interface.

Cameras require space and adequate lighting to focus and capture images for CV. This limits the minimum size of artifacts that are built from CV marker-driven physical computing. In FunFund (project 1C) and HEXBOX (project 1D) for example, designers had to use a fairly large box to ensure that the camera was able to capture all the interactions. Mirrors were therefore a clever way to redirect the path of light and alleviate the space constraints of CV systems, and we see many projects in the portfolio use it to great effect (specifically all projects in the Tinycade studio, and projects 3B and 3D). Mirrors have similarly been used in other CV-driven TUIs to reduce the size of the system, such as in *Sauron* [41] and ClipWidgets [50].

Beyond a limitation to work around, these intangible materials were also leveraged for tangible interaction design. Notably, the two ambient sound interfaces presented by Elucidate (project 1B) are centered on light, shadow, and wind. Both interfaces feature a grid of CV markers which are revealed and hidden in CV by the positions of shadows cast on the inside of a box. The first interface monitors the moving light of the sun, and the second interface features CV markers attached to moving flaps which flap under different wind conditions.

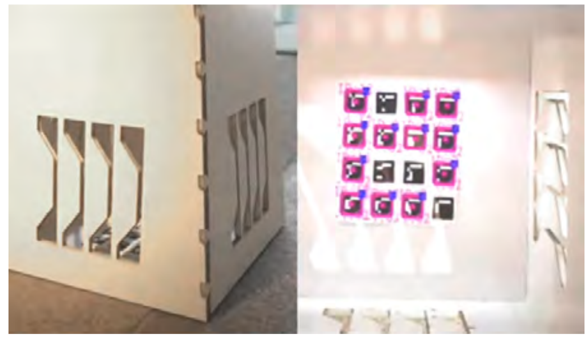


Figure 12: Elucidate (project 1B) uses a grid of CV markers that appear and disappear from CV detection due to the shadows cast by the movement of sunlight over time.

5.4.2 The Democratic Materiality of CV Markers. Besides being CV-driven, artifacts in the portfolio we present are also unified through their use of democratic materials. By democratic materials, we refer to the use of physical materials that are commonplace (e.g., paper and cardboard), computational tools that are widely available (e.g., webcams and smartphones), and making processes that are easily accessed at home or at a design workspace (e.g., scissors, paper-knives, hot glue). This was a conscious decision that we facilitated through the design briefs that we wrote for each studio. As design educators, we were interested to explore a physical computing approach that maximized "economy of means" in terms of prototyping [28]. Leveraging everyday materials and tools was also necessitated by the situation that our designers found themselves in during the COVID-19 pandemic.

Notably, Data Is Yours (project 2D) made democratic making the project's focus [3]. The designers created a toolkit for other educators to replicate with simple tools as their audience of educators does not have access to advanced fabrication tools (e.g., 3D printing and laser cutting). As for the other projects in the portfolio, while we observed most projects use cardboard and paper, some projects applied CV markers to other physical materialities—such as wood (project 1A), plastic sheets (projects 1G, 3B), and soft wearables (project 3C). As a web-based library, *Beholder* also supports cross-platform on all web browsers. During our studios, this enabled designers to choose the platform that best fits their experience (i.e., mobile or desktop).

5.5 Limitations

While we advocate for CV markers as an approach to facilitate physical computing, we observed a few constraints that this approach place on designers in our studios. We took note of the limitations and trade-offs that one encounters when adopting this approach in comparison to the conventional electronics-driven method, and organized them into a few important constraints that designers need to consider.

5.5.1 Camera and Field of View. A significant difference between electronics and computer vision is the need to accommodate the camera FOV in the interface. We observed designers addressing this by either pointing a camera from the outside at an exposed set of CV markers (e.g., projects 1F, 1G, 3A, and 3C) or by integrating the camera into the artifact and leaving a large empty space for it to observe (e.g., project 1C, 1D, and 3D). Designers also used mirrors to redirect the camera view in order to capture a wider angle within a more compact space (e.g. Studio 2 projects). Accommodating the camera also extends to the device level. For example, every different smartphone model has a different form factor, including its camera placement and FOV. This was a main challenge for Data Is Yours (project 2D) [3] as the designers resolved to only support one phone model.

5.5.2 Beholder Specific Limitations. While *Beholder* is intended to support a multitude of platforms, relying on web browsers presents some challenges. In particular, the web API for image processing is much slower than native versions of OpenCV [35]. This increases potential input latency to at least 15 milliseconds (the capped browser frame rate) where an electronics approach could process input events instantly. Where processing power is more limited (i.e., smartphones), it is common to see a minimum delay of 50 milliseconds. High-resolution camera feeds can also slow down detection due to the increased processing load from a larger pixel count. To combat this, we added a feature to *Beholder* which enables designers to select only a portion of the video feed to run detection on, which can dramatically increase speed. However, this approach requires designers to finely tune the detection area by hand for each device.

5.5.3 Sensing Only. In its current form, our approach does not support actuation—one important component of electronics-driven physical computing. We crafted our studio briefs with this in mind, and studio projects used screens and speakers already present in computing devices (smartphones, laptops) for visual/audio feedback. This limits the design space to tangible interfaces that provide input only. Due to this, we view CV markers as a companion or alternative for electronics *sensors* in physical computing rather than a replacement.

Even as inputs, we observed that one team from studio 3 (an excluded project) opted to use the built-in inertial measurement unit (IMU) of a smartphone to realize their tangible interface instead of CV markers. Computer vision with markers is less reliable for interactions with rapid movements due to the motion blur of the image feed. The IMU works better in interaction cases with fast physical gestures.

5.5.4 Challenges For Facilitation. Devices can also pose a challenge for facilitation. Where there are dozens of Arduino starter kits available for cheap, we had to ensure each participant had access to a CV-capable device before starting a workshop. This means either surveying the devices participants had or providing devices as part of facilitation. The latter situation represents how the Data Is Yours [3] designers chose to proceed. We also encountered some expected friction when introducing the coding aspect for Beholder. This challenge is shared with Arduino. We observed that teams of designers would divide their roles across fabrication and programming. There were also inconsistencies with the markers themselves. Ink on paper wears down rather quickly with a lot of friction, this can be remedied by filling in gaps with a black marker, but needs to be clearly explained when teaching how to use CV markers.

6 CONCLUSION

In this annotated portfolio, we present 15 projects that used a CV marker-based approach for physical computing. Through reflecting on these projects, we uncovered the following insights about designing with CV markers:

(1) They can be applied to physical materials to sense a wide range of interactions through a CV system.
(2) They can be used to directly mark interaction events, enabling designers to "program" system logic in the physical world.
(3) They can be debugged using human vision, thus enabling designers to draw connections between what they see and what a CV system detects.
(4) They enable designers to leverage a diverse materiality, including democratic physical materials. They are particularly sensitive to intangible environmental factors, which should be considered as an intangible material to shape when working with CV markers.

We continue to use the approach described in this paper—*marking material interactions with computer vision*—to facilitate physical computing and the building of tangible interfaces for ourselves and other designers. We are constantly iterating on the tools for this approach. For example, we plan to extend *Beholder* to be a standalone application that removes the requirement for software programming. We also plan to create a library of CV marker mechanisms that translate physical interactions to CV marker detection, as templates to build on top of when engaging with our approach.

In this paper, we report exclusively on the project outcomes from the studios, our facilitation as studio instructors, and our reflections on them. We hope to expand upon our insights in future work by conducting follow-up interviews with studio participants to understand their first-person experience of adopting a CV marker-driven approach to physical computing, and see if it has impacted their design practice following the studio.

We hope that this work will contribute to the range of alternatives that designers can employ for physical computing—and inspire more research in this area.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. National Science Foundation under Grant No IIS-2040489 and National University of Singapore Startup Fund A-0008470-01-00. We also want to thank our students for testing Beholder and providing us with invaluable feedback.

REFERENCES

[1] Adafruit. 2022. *Adafruit Industries, Unique & fun DIY electronics and kits.* Adafruit. https://www.adafruit.com/
[2] Arduino. 2022. *Arduino - Home.* Arduino. https://www.arduino.cc/
[3] S Sandra Bae, Rishi Vanukuru, Ruhan Yang, Peter Gyory, Ran Zhou, Ellen Yi-Luen Do, and Danielle Albers Szafir. 2022. Cultivating Visualization Literacy for Children Through Curiosity and Play. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 257–267.
[4] Bahareh Barati and Elvin Karana. 2019. Affordances as Materials Potential: What Design Can Do for Materials Development. *International Journal of Design* 13, 3

(Dec. 2019), 105–123. http://www.ijdesign.org/index.php/IJDesign/article/view/3419/879
[5] Hernando Barragán. 2022. *Wiring*. Wiring. http://wiring.org.co/
[6] Ayah Bdeir. 2009. Electronics as material: littleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (2009-02-16) *(TEI '09)*. Association for Computing Machinery, New York, NY, USA, 397–400. https://doi.org/10.1145/1517664.1517743
[7] Eli Blevis, Youn-kyung Lim, Erik Stolterman, Tracee Vetting Wolf, and Keichi Sato. 2007. Supporting design studio culture in HCI. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* (2007-04-28) *(CHI EA '07)*. Association for Computing Machinery, New York, NY, USA, 2821–2824. https://doi.org/10.1145/1240866.1241086
[8] Tracey Booth, Simone Stumpf, Jon Bird, and Sara Jones. 2016. Crossed Wires: Investigating the Problems of End-User Developers in a Physical Computing Task. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016-05-07) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3485–3497. https://doi.org/10.1145/2858036.2858533
[9] John Bowers. 2012. The logic of annotated portfolios: communicating the value of 'research through design'. In *Proceedings of the Designing Interactive Systems Conference* (2012-06-11) *(DIS '12)*. Association for Computing Machinery, New York, NY, USA, 68–77. https://doi.org/10.1145/2317956.2317968
[10] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008-04-06) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 423–432. https://doi.org/10.1145/1357054.1357123
[11] Leah Buechley and Benjamin Mako Hill. 2010. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems* (2010-08-16) *(DIS '10)*. Association for Computing Machinery, New York, NY, USA, 199–207. https://doi.org/10.1145/1858171.1858206
[12] SparkFun Electronics. 2022. *SparkFun Electronics*. SparkFun Electronics. https://www.sparkfun.com/
[13] MIT's Center for Bits and Atoms. 2022. HTMAA 2022. https://fab.cba.mit.edu/classes/MAS.863/
[14] Micro:bit Educational Foundation. 2022. *Micro:bit Educational Foundation*. Micro:bit Educational Foundation. https://microbit.org/
[15] Raspberry Pi Foundation. 2022. *Teach, learn, and make with the Raspberry Pi Foundation*. Raspberry Pi Foundation. https://www.raspberrypi.org/
[16] Mikhaila Friske, Shanel Wu, and Laura Devendorf. 2019. AdaCAD: Crafting Software For Smart Textiles Design. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019-05-02) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300575
[17] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005
[18] Bill Gaver and John Bowers. 2012. Annotated portfolios. *Interactions* 19, 4 (2012), 40–49. https://doi.org/10.1145/2212877.2212889
[19] William Gaver. 2012. What Should We Expect from Research through Design?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 937–946. https://doi.org/10.1145/2207676.2208538
[20] Shad Gross, Jeffrey Bardzell, and Shaowen Bardzell. 2014. Structures, forms, and stuff: the materiality and medium of interaction. *Personal and Ubiquitous Computing* 18, 3 (2014), 637–649. https://doi.org/10.1007/s00779-013-0689-4
[21] Kate Hartman, Emma Westecott, Izzie Colpitts-Campbell, Jennie Robinson Faber, Yiyi Shao, Chris Luginbuhl, Olivia Prior, and Manisha Laroia. 2021. Textile Game Controllers: Exploring Affordances of E-Textile Techniques as Applied to Alternative Game Controllers. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (2021-02-14) *(TEI '21)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3430524.3446069
[22] Sabrina Hauser, Doenja Oogjes, Ron Wakkary, and Peter-Paul Verbeek. 2018. An Annotated Portfolio on Doing Postphenomenology Through Research Products. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 459–471. https://doi.org/10.1145/3196709.3196745
[23] Cindy E. Hmelo and Mark Guzdial. 1996. Of black and glass boxes: scaffolding for doing and learning. In *Proceedings of the 1996 international conference on Learning sciences* (1996-07-25) *(ICLS '96)*. International Society of the Learning Sciences, Evanston, Illinois, 128–134.
[24] Kristina Höök and Jonas Löwgren. 2012. Strong concepts: Intermediate-level knowledge in interaction design research. *ACM Transactions on Computer-Human Interaction* 19, 3 (2012), 23:1–23:18. https://doi.org/10.1145/2362364.2362371
[25] Lee Jones. 2021. The E-darning Sampler: Exploring E-textile Repair with Darning Looms. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (2021-02-14) *(TEI '21)*. Association for Computing Machinery, New York, NY, USA, 1–5. https://doi.org/10.1145/3430524.3444700
[26] Martin Kaltenbrunner and Ross Bencina. 2007. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (2007-02-15) *(TEI '07)*. Association for Computing Machinery, New York, NY, USA, 69–74. https://doi.org/10.1145/1226969.1226983
[27] Gierad Laput, Eric Brockmeyer, Scott E. Hudson, and Chris Harrison. 2015. Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015-04-18) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2161–2170. https://doi.org/10.1145/2702123.2702414
[28] Youn-Kyung Lim, Erik Stolterman, and Josh Tenenberg. 2008. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction* 15, 2 (2008), 7:1–7:27. https://doi.org/10.1145/1375761.1375762
[29] Jonas Löwgren. 2013. Annotated Portfolios and Other Forms of Intermediate-Level Knowledge. *Interactions* 20, 1 (jan 2013), 30–34. https://doi.org/10.1145/2405716.2405725
[30] F. Martin, B. Mikhak, and B. Silverman. 2000. MetaCricket: A designer's kit for making computational devices. *IBM Systems Journal* 39, 3 (2000), 795–815. https://doi.org/10.1147/sj.393.0795
[31] David A. Mellis, Sam Jacoby, Leah Buechley, Hannah Perner-Wilson, and Jie Qi. 2013. Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an "untoolkit". In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (2013-02-10) *(TEI '13)*. Association for Computing Machinery, New York, NY, USA, 83–90. https://doi.org/10.1145/2460625.2460638
[32] Dave Murray-Rust, Chris Elsden, Bettina Nissen, Ella Tallyn, Larissa Pschetz, and Chris Speed. 2022. Blockchain and Beyond: Understanding Blockchains through Prototypes and Public Engagement. *ACM Trans. Comput.-Hum. Interact.* (feb 2022). https://doi.org/10.1145/3503462 Just Accepted.
[33] Nintendo. 2022. *Create new ways to play with Nintendo Labo!* Nintendo. https://www.nintendo.co.uk/Nintendo-Labo/Nintendo-Labo-1328637.html
[34] Edwin Olson. 2011. AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*. IEEE, IEEE, New York, NY, USA, 3400–3407.
[35] OpenCV. 2022. *OpenCV*. OpenCV. https://opencv.org/
[36] Dan O'Sullivan and Tom Igoe. 2004. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Course Technology Press, Boston, MA, United States.
[37] Irene Posch and Ebru Kurbak. 2016. CRAFTED LOGIC Towards Hand-Crafting a Computer. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (2016-05-07) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 3881–3884. https://doi.org/10.1145/2851581.2891101
[38] Jie Qi and Leah Buechley. 2014. Sketching in circuits: designing and building electronics on paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014-04-26) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1713–1722. https://doi.org/10.1145/2556288.2557391
[39] Mitchel Resnick and Eric Rosenbaum. 2013. Designing for Tinkerability. In *Design, Make, Play*. Routledge, Abingdon, Oxfordshire, UK.
[40] Erica Robles and Mikael Wiberg. 2010. Texturing the "material turn" in interaction design. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction* (2010-01-24) *(TEI '10)*. Association for Computing Machinery, New York, NY, USA, 137–144. https://doi.org/10.1145/1709886.1709911
[41] Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (2013-10-08) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 447–456. https://doi.org/10.1145/2501988.2501992
[42] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015-04-18) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1277–1280. https://doi.org/10.1145/2702123.2702207
[43] Donald A. Schon and Glenn Wiggins. 1992. Kinds of seeing and their functions in designing. *Design Studies* 13, 2 (1992), 135–156. https://doi.org/10.1016/0142-694X(92)90268-F
[44] D.A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5, 1 (1992), 3–14. https://doi.org/10.1016/0950-7051(92)90020-G Artificial Intelligence in Design Conference 1991 Special Issue.
[45] Michael Shorter, Jon Rogers, and John McGhee. 2014. Practical notes on paper circuits. In *Proceedings of the 2014 conference on Designing interactive systems*

(2014-06-21) *(DIS '14)*. Association for Computing Machinery, New York, NY, USA, 483–492. https://doi.org/10.1145/2598510.2602965

[46] Cesar Torres, Jessica Chang, Advaita Patel, and Eric Paulos. 2019. Phosphenes: Crafting Resistive Heaters within Thermoreactive Composites. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (2019-06-18) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 907–919. https://doi.org/10.1145/3322276.3322375

[47] Vasiliki Tsaknaki, Pedro Sanches, Tom Jenkins, Noura Howell, Laurens Boer, and Afroditi Bitzouni. 2022. Fabulating Biodata Futures for Living and Knowing Together. In *Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1878–1892. https://doi.org/10.1145/3532106.3533477

[48] New York University. 2022. ITP Physical Computing. https://itp.nyu.edu/physcomp/

[49] Anna Vallgårda and Johan Redström. 2007. Computational composites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2007-04-29) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 513–522. https://doi.org/10.1145/1240624.1240706

[50] Aaron Visschedijk, Hyunyoung Kim, Carlos Tejada, and Daniel Ashbrook. 2022. ClipWidgets: 3D-printed Modular Tangible UI Extensions for Smartphones. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction* (2022-02-13) *(TEI '22)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3490149.3501314

[51] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. 2009. SLAP widgets: bridging the gap between virtual and physical controls on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009-04-04) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 481–490. https://doi.org/10.1145/1518701.1518779

[52] Yang Zhang and Chris Harrison. 2018. Pulp Nonfiction: Low-Cost Touch Tracking for Paper. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018-04-19). ACM, Montreal QC Canada, 1–11. https://doi.org/10.1145/3173574.3173691

[53] Clement Zheng, Peter Gyory, and Ellen Yi-Luen Do. 2020. Tangible Interfaces with Printed Paper Markers. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (2020-07-03) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 909–923. https://doi.org/10.1145/3357236.3395578

[54] Clement Zheng, HyunJoo Oh, Laura Devendorf, and Ellen Yi-Luen Do. 2019. Sensing Kirigami. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (2019-06-18) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 921–934. https://doi.org/10.1145/3322276.3323689