Facilitating Physical Computing with Computer Vision Markers

Clement Zheng[1]

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

Peter Gyory

ATLAS Institute, University of Colorado, Boulder, peter.gyory@colorado.edu

Ellen Yi-Luen Do

ATLAS Institute & Computer Science, University of Colorado, Boulder, ellen.do@colorado.edu

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. Over the last three years, we ran a series of studios with design participants to investigate how CV markers can participate in physical computing and the construction of physical interactive systems. 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. Taking these insights, we are developing a visual editor that enables designers to easily program marker behavior and connect it to keyboard events. We believe that such a platform will enable designers to develop physical and digital interfaces concurrently while minimizing the complexity of integrating both sides. In addition, this platform can also facilitate the construction of many alternative interfaces for existing software that cater to different people. We discuss our motivation, progress, and future work of this research here.

CCS CONCEPTS • Human-Computer Interaction

Additional Keywords and Phrases: Physical Computing, Computer Vision, Interaction Design

ACM Reference Format:
Clement Zheng, Peter Gyory, Ellen Yi -Luen Do. 2023. Facilitating Physical Computing with Computer Vision Markers.

---

[1]This work was presented at the **CHI2023 Workshop [WS2] - Beyond Prototyping Boards: Future Paradigms for Electronics Toolkits**
CHI '23, April 23-28, 2023, Hamburg, Germany

1 INTRODUCTION

As researchers and educators in the area of tangible interaction design, we have been tasked with both designing and building physical interactive systems, as well as facilitating other designers to construct such systems. We observed that building such TUIs (tangible user interfaces) pose many challenges for designers. Building a TUI is a complex design task that requires designers to integrate both physical materials and physical computing into a coherent working system that considers usability, aesthetics, and functional integrity. Currently, physical computing is largely driven by off-the-shelf electronic and microcontroller kits—notably, platforms like *Arduino*, *Micro:bit*, and *Raspberry Pi*. These platforms offer many advantages in terms of prototyping physical computing systems. For instance, they support a plug-and-play approach to explore and program electronic circuits, sensors, and actuators all while hiding low-level concepts like hardware registers and digital signal processing. However, they also pose many challenges to designers, particularly novices in this area. Booth et al. [1] 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. These challenges motivated us to investigate the question:

What if physical computing does not require electronics and circuits?

Over the last three years, we have been exploring an alternative approach to physical computing—one that uses CV (computer vision) markers instead of electronics for sensing TUIs. Our approach was inspired by previous work in this area, notably *ReacTable* [2] and *Sauron* [3]. These examples demonstrate how a CV marker-driven approach supports designers to build tangible interfaces that can be applied on to many different materials without any electronic circuits. Based on the design affordances offered by CV markers, we developed *Beholder*, a platform with a growing collection of tools that supports others in leveraging CV markers for tangible interaction design.

In this paper, we outline the work that we have done with *Beholder* and discuss our vision for this platform and how it might grow to support physical computing in different ways. The specifics of the work we discuss in this paper is covered in more detail in prior publications [4–7], including a paper that we will present at CHI 2023 [8].

1.1 Team & Project Background

This research developed from our prior work *Printed Paper Markers* [9]. That project concluded in 2020 just as the COVID-19 pandemic hit—and it revealed how CV markers could be used as a simple and low-cost (and therefore democratic) approach to sense tangible interactions with minimal setup and equipment. Inspired by *Printed Paper Markers*, we used CV markers during the pandemic to facilitate others to build tangible interfaces in our classes and research projects; developing tools around their use as a physical computing material in the process. This effort spanned the two institutions that we were teaching and researching at: National University of Singapore (NUS) and University of Colorado, Boulder (CU Boulder).

2 FACILITATING A CV-DRIVEN APPROACH TO PHYSICAL COMPUTING

From 2020 to 2022, we ran three studios (project-based classes typically conducted at university-level design programs) to probe how others might use CV markers for tangible interaction design. To support studio participants, we developed a software library based on a JavaScript port[2] of the ArUco [10] marker library. We packaged this software library in a web application template that has built-in marker detection and a live debugging view (Figure 1). Detected marker properties are accessed through the software library[3]. Studio participants built functional tangible interfaces that communicated with computers and smartphones by making use of the ID, position, and rotation of detected CV markers on a physical artifact.

Figure 2 captures the studio outcomes. We ran the first studio for a semester with undergraduate industrial design students at NUS. Inspired by the pandemic situation, we briefed participants to develop interactive systems for everyday activities that can be easily built by people at home. We ran the second studio with students and researchers at our lab at

[2]https://github.com/jcmellado/js-aruco
[3]https://github.com/project-beholder/

CU Boulder. For this studio, we developed a smartphone-operated cardboard arcade machine. We briefed participants to design a video game with a unique cardboard controller for this machine. We ran the third studio for a semester with a new group of undergraduate industrial design students at NUS. For this final studio, we honed in on games and briefed participants to develop new play or game experiences facilitated by CV markers.

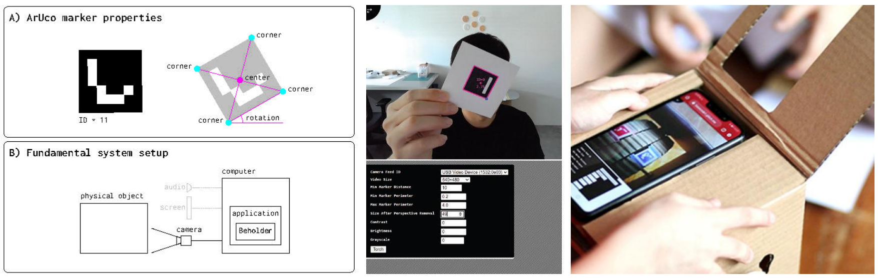

Figure 1: Left: Computational properties of an ArUco marker and fundamental setup of a CV marker-based physical computing system. Middle: An example debug view of the web-based template we developed for *Beholder*. Right: A smartphone running *Beholder* embedded in a student's cardboard prototype.

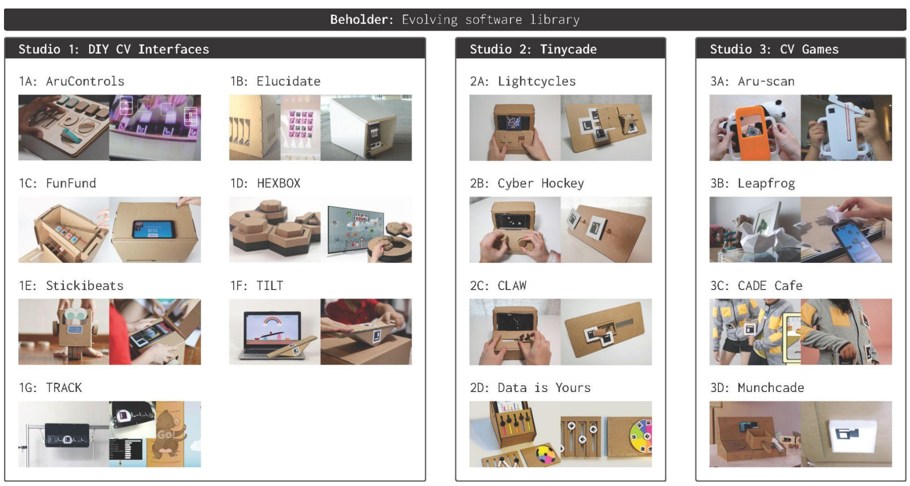

Figure 2: Studio outcomes facilitated with CV marker-based physical computing.

We documented and assembled the outcomes from these studios (Figure 2) and reflected on this body of work as an *annotated portfolio* [9]. By analyzing the studio outcomes as a whole, we noted several insights surrounding the practice of using CV markers for physical computing that differed from the electronics/microcontroller-driven approach:

(1) We use different electronic components to sense different interactions with microcontrollers. On the other hand, with the restriction of just using CV markers, we were still able to observe designers creating a wide range of interactions, and uncovered the computational materiality and sensing range of tangible interactions with markers. Figure 3 organizes the different tangible interactions found in our studio outcomes.

(2) Rather than sense the exact interaction that is happening, CV markers were used to "mark" interaction events that occurred. For example: placing markers at specific points along a slider and revealing them when the slider is at that point to indicate that the slider has reached a particular position. This enabled studio participants to offload software programming to the placement of markers, and determine interaction logic in-situ.

(3) Studio participants were able to debug their constructions by looking at how CV markers were acting; CV markers supported human vision as a proxy for the computer vision detection system. In the case of electronics

and circuits, debugging often requires reading invisible signals through meters or scopes. As educators, we found that the visual nature of CV markers lowered the barrier of working with physical computing for our studio participants.

(4) CV markers are simple physical tags that can be easily printed and attached to objects. This enabled studio participants to work with readily available materials and found objects to construct tangible interfaces. CV markers do not require a physical tether—unlike electronics. This reduced the complexity of integrating computational materials into a physical interactive system.

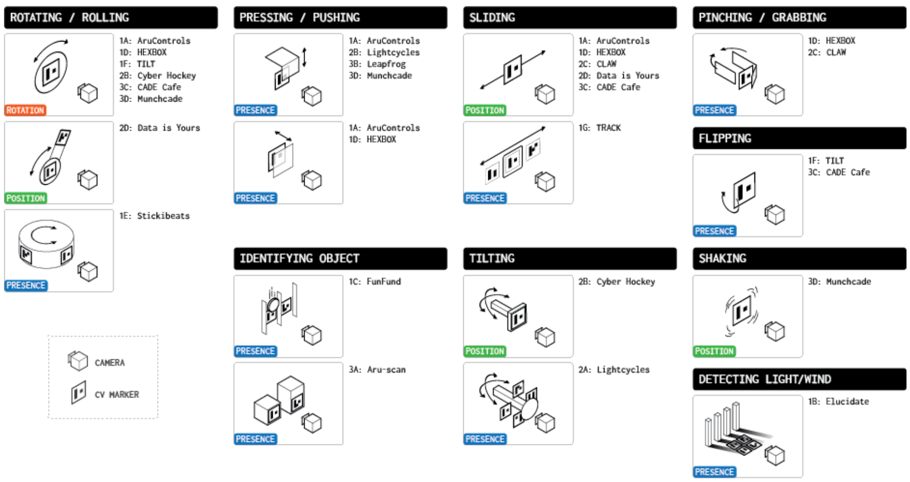

Figure 3: Different tangible interactions that CV markers support.

Despite these advantages, we observed several challenges posed by CV markers as a physical computing approach. Markers require a line of sight to the camera and detection depends largely on environmental conditions. Working around these factors constrained the size and internal structure of the interfaces that studio participants built. CV-based interfaces also required environment calibration, which affected their portability. Finally, the biggest challenge for our participants was programming marker logic in software through conventional coding syntax (web-based development in our case). This was an uncomfortable and foreign task (be it electronics-based or CV-based) within physical computing for our design students.

While we acknowledge that integrating camera vision and environmental factors are a fundamental part of using CV systems, we saw an opportunity to improve the way that others program marker behavior and interaction logic for CV marker-based physical computing.

3 A VISUAL PROGRAMMING PLATFORM FOR CV MARKERS

We were inspired by visual programming platforms for physical computing, notably Micro:bit's Makecode Editor[4] and Tinkercad circuit simulator[5]. In addition, we observed from the studios that CV markers, as a physical computing material, were useful for building tangible interfaces to control software applications (like video games or productivity software). With that in mind, we developed a visual programming editor that enables designers to define marker logic and connect it to keyboard events [5]. The program defined in this editor runs as a background process on the computer and spoofs keyboard input. This therefore connects CV marker-driven tangible controllers to any software that responds to keyboard input.

Through this, we hope to decouple programming the physical interface with programming the digital interface / interactive software. This also enables designers to build new physical interfaces for existing software that is controlled

[4]https://makecode.microbit.org/
[5]https://www.tinkercad.com/circuits

by a keyboard. As an added benefit, we see this enabling designers to build more accessible interfaces as an alternative for different people—such as interfaces for people with different physical abilities like elderly or young children.

We developed a first prototype of this editor using the C++ OpenCV library for ArUco markers and native keyboard emulation for Windows and MacOS devices (Figure 4). Compared to the JavaScript port, we found that the OpenCV library has faster and more robust marker detection—mitigating many of the environmental issues that we faced when running the studios. We are in the process of improving the visual editor, and plan to deploy it with other researchers and designers in workshops to study how it will be used.

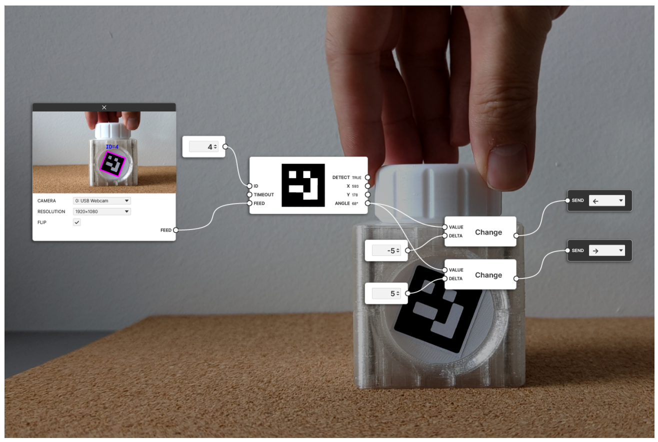

Figure 4: First version of the visual editor: connecting CV marker behavior to keyboard events.

4 DISCUSSION

As we continue developing this CV-based physical computing approach, it is also important for us to acknowledge the limitations of such an approach. Most notably—this approach does not support actuation or feedback beyond what the computing device offers (e.g. display, speakers, vibration). We plan to investigate how the visual editor might connect and control external microcontroller actuators (such as robotic arm kits). In addition, we observed that this approach does not perform well for tracking rapid movements due to image blurring, especially when using the cameras on board laptops and smartphones.

Moving ahead, we plan to deploy the visual editor in another round of design studios to study how using this editor compares to the first round of studios. We are also actively looking for collaborators in different design domains (e.g. game design, accessibility design, information design) and work with them to incorporate such an approach and toolkit to their work. Through these activities, we will also be evaluating and improving the usability of the visual editor. More broadly, we also plan to build up a repository of example projects, as well as building blocks of inputs or sensors based on CV markers (e.g. Figure 3). We hope that such a repository can inspire more people to explore CV marker-driven physical computing with us.

REFERENCES

[1] 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

[2] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07). Association for Computing Machinery, New York, NY, USA, 139–146. https://doi.org/10.1145/1226969.1226998

[3] 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 (UIST '13). Association for Computing Machinery, New York, NY, USA, 447–456. https://doi.org/10.1145/2501988.2501992

[4] Peter Gyory, Perry Owens, Matthew Bethancourt, Amy Banic, Clement Zheng, and Ellen Yi-Luen Do. 2022. Build Your Own Arcade Machine with Tinycade. In Creativity and Cognition (C&C '22). Association for Computing Machinery, New York, NY, USA, 312–322. https://doi.org/10.1145/3527927.3533023

[5] Peter Gyory, Krithik Ranjan, Zhen Zhou Yong, Clement Zheng, and Ellen Yi-Luen Do. 2022. Directing Tangible Controllers with Computer Vision and Beholder. In SIGGRAPH Asia 2022 Emerging Technologies (SA '22). Association for Computing Machinery, New York, NY, USA, Article 2, 1–2. https://doi.org/10.1145/3550471.3564764

[6] S. S. Bae et al., "Cultivating Visualization Literacy for Children Through Curiosity and Play," in IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 1, pp. 257-267, Jan. 2023, https://doi.org/10.1145/10.1109/TVCG.2022.3209442.

[7] Sandra Bae, Ruhan Yang, Peter Gyory, Julia Uhr, Danielle Albers Szafir, and Ellen Yi-Luen Do. 2021. Touching Information with DIY Paper Charts & AR Markers. In Interaction Design and Children (IDC '21). Association for Computing Machinery, New York, NY, USA, 433–438. https://doi.org/10.1145/3459990.3465191

[8] Peter Gyory, S. Sandra Bae, Ruhan Yang, Ellen Yi-Luen Do, Clement Zheng. 2023. Marking Material Interactions with Computer Vision. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI ’23). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3580643

[9] 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 (DIS '20). Association for Computing Machinery, New York, NY, USA, 909–923. https://doi.org/10.1145/3357236.3395578

[10] 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