LOW-RESOLUTION CUSTOMIZABLE UBIQUITOUS DISPLAYS

by

ANTONY ALBERT RAJ IRUDAYARAJ

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirements for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2023

EXAMINING COMMITEE MEMBERSHIP

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Ehud Sharlin Professor, Dept. of Computer Science, University of Calgary
Supervisors	Daniel Vogel Associate Professor, School of Computer Science, University of Waterloo
	Omid Abari Assistant Professor, Dept. of Computer Science, University of California, Los Angeles
Internal Member	Craig S. Kaplan Associate Professor, School of Computer Science, University of Waterloo
	Jian Zhao Assistant Professor, School of Computer Science, University of Waterloo
Internal-external Member	Rob Gorbet Associate Professor, Dept. of Knowledge Integration, University of Waterloo

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

STATEMENT OF CONTRIBUTIONS

This dissertation includes first-authored peer-reviewed material that has appeared in conference and journal proceedings published by the Association for Computing Machinery (ACM). The ACM's policy is as follows[1]

> *"Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected)."*

This dissertation also includes first-authored peer-reviewed material that has appeared in Graphics Interface conference published by OpenReview [2].

The following list serves as a declaration of the works included in this dissertation. This material is expanded and revised from the original publication.

Portions of Chapter 3

Antony Albert Raj Irudayaraj, Rishav Agarwal, Nikhita Joshi, Aakar Gupta, Omid Abari, and Daniel Vogel. 2021. PocketView: Through-Fabric Information Displays. In The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21). Association for Computing Machinery, New York, NY, USA, 511–523. https://doi.org/10.1145/3472749.3474766

Nikhita Joshi and Rishav Agarwal contributed to the survey and running of user studies in the PocketView project. Aakar Gupta introduced the idea of a PDLC film to view information through the pocket. The PDLC pocket is one of the baselines for the user study (Chapter 3).

Portions of Chapter 4

Antony Albert Raj Irudayaraj, Jeremy Hartmann and Omid Abari, and Daniel Vogel. 2023. Scatterpixels: Ad Hoc Reconfigurable Physical Pixel Displays, Graphics Interface 2023 https://openreview.net/forum?id=t-IRQBTmDXV

Jeremy Hartmann contributed with the *'Generalized registration with multiple capture poses'* section in the Scatterpixels project (Chapter 4).

---

[1] https://authors.acm.org/author-resources/author-rights

[2] https://openreview.net/about

ABSTRACT

In a conventional display, pixels are constrained within the rectangular or circular boundaries of the device. This thesis explores moving pixels from a screen into the surrounding environment to form ubiquitous displays. The surrounding environment can include a human, walls, ceiling, and floor. To achieve this goal, we explore the idea of customizable displays: displays that can be customized in terms of shapes, sizes, resolutions, and locations to fit into the existing infrastructure. These displays require pixels that can easily combine to create different display layouts and provide installation flexibility. To build highly customizable displays, we need to design pixels with a higher level of independence in its operation. This thesis shows different display designs that use pixels with pixel independence ranging from low to high. Firstly, we explore integrating pixels into clothing using battery-powered tethered LEDs to shine information through pockets. Secondly, to enable integrating pixels into the architectural surroundings, we explore using battery-powered untethered pixels that allow building displays of different shapes and sizes on a desired surface. The display can show images and animations on the custom display configuration. Thirdly, we explore the design of a solar-powered independent pixel that can integrate into walls or construction materials to form a display. These pixels overcome the need to recharge them explicitly. Lastly, we explore the design of a mechanical pixel element that can be embedded into construction material to form display panels. The information on these displays is updated manually when a user brushes over the pixels. Our work takes a step forward in designing pixels with higher operation independence to envision a future of displays anywhere and everywhere.

ACKNOWLEDGMENTS

I would like to thank my supervisors Dan and Omid for providing me the opportunity to do my Ph.D. at the University of Waterloo. It has been a great journey for five years with a lot of ups and down. A sincere thanks to Dan for being an extremely supportive supervisor. Thank you Omid for all the discussion about my projects. I would like to thank Aakar Gupta for introducing me to the field of HCI research.

I would like to extend my sincere thanks to my Dad, Mom, and Sister for their constant support throughout the course of my Ph.D.

I would like to thank my Toronto family for motivating me to do my Ph.D. and for their support during difficult times. Thank you Vimal for being my constant support during my hard times and my partner in building cool things. Thank you Goutham for your motivation and constant support. Thank you Sushmitha Akka for your support during my hard times and helping me with reviewing my SOP and CV. Thank you Sowjanya for keeping in touch during my hard phase. Thank you Deepak, Sowmya, Sagar, Ashish, Anand anna, and Joshua for all your support.

I would like to thank my Waterloo family for all the memories and their support during my Ph.D. I am listing people based on alphabetical order. Thank you Ashwini Akka for all the memories at 139A. Thank you Atish for being my roommate and for cricket games. Thank you Ganesh for being my roommate for 2 years and for cricket memories. Thank you Harshitha for checking on me during my hard times. Thank you Laksh for being my trip buddy, for sharing food, and for always calling to check on me. Thank you Manoj for being my roommate for 4 years, for late-night discussions, trips, cooking food together and inviting people, and being my constant support for everything. Thank you Ravi for the cricket games, cricket discussion, and fish biriyani. Thank you Neha Buddy for patiently listening to all my stories, infinitely long calls, messages, and trips. Thank you Ruta for the infinite coffee walks. Thank you Sneha Ditcher for being my trip/activity buddy, for annoying and fighting with me, spamming with messages, EVS & social issues discussion, and for cute cafe trips. Thank you Sakthi Ji for greeting me the first time I came to Waterloo and being my roommate. Thank you Sanjay for all the biking trips. Thank you Zeba madam for being my game buddy, for offering food & biryani, late-night discussions, and listening to my stories. Thank you Mahesh and Vaishali for all your memories. Thank you everyone for being part of this important journey of my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Mark Weiser's vision of ubiquitous computing states [133]:

> *'The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it'*

Inspired by Weiser's vision, we explore the concept of ubiquitous displays that integrate pixels into the surrounding environment. Imagine a scenario where pixels can be placed on surfaces, garments, or embedded into the architectural surroundings to display pertinent information or create aesthetic patterns. Installing displays into existing infrastructure significantly benefits from easy installation and flexibility. To provide this flexibility, we explore the space of customizable displays by combining pixels in different ways to enable ubiquitous display applications. The pixels could augment an existing surface, replace a particular segment of construction materials, or integrate into clothing.

A display is an output interface to visually present information from computing devices through text, images, or graphics. Advancements in display technologies have focused on increasing the resolution to render superior image quality for enhanced user experience. Despite high-resolution displays becoming mainstream, low-resolution displays are still used in digital signage, advertisement boards, wearable displays, and media facades. We focus on customizable displays that have low resolution, which trades the fidelity of conventional displays for a high level of flexibility in display configurations, enabling new kinds of ubiquitous display use cases and novel aesthetic display experiences. The main idea is to move the pixels constrained within a conventional display screen to the surrounding area, which includes the human body, wall, ceiling, and floor.

We define *customizable display*s as displays that use physical pixels to create custom configurations to provide flexibility in shapes, sizes, resolutions, and locations. A *physical pixel* is an individual entity containing a display element whose visual state is controlled from a base station, displaying a small piece of information. Different technologies, such as LCD, LEDs, or E-Ink can be used for the display element. Figure 1.1 illustrates the typical workflow for a generalized customizable display system. A host device (e.g., phone) is used to program the visual state of the pixels forming the customizable display. The host device sends the programmed display configuration to a base station through wired or wireless communication. Subsequently, the base station sends control signals to set the visual state of one or more physical pixels. Pixels are updated through either contact (wires) or non-contact methods (e.g., radio waves, light waves, magnetic fields).

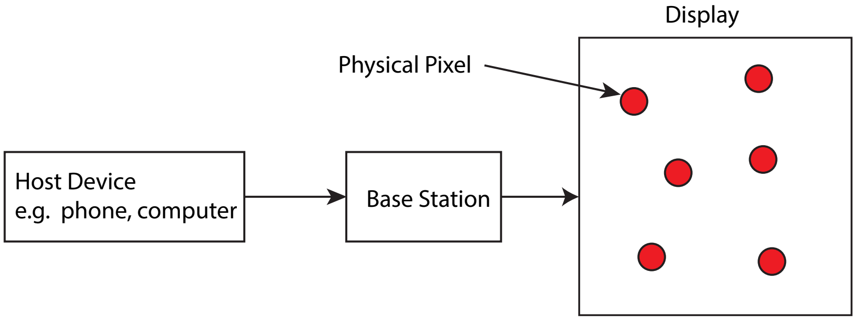


Figure 1.1: Typical workflow of a customizable display system: A host device sends information to be shown on display to a base station. Using this information, the base station sends control signals to update the physical pixels of the display.

The primary objective of customizable displays is to facilitate easy installation and flexibility in creating diverse display layouts that seamlessly integrate into existing infrastructure. To achieve this goal, the design of pixels should prioritize *independence* of operation. A pixel with higher independence enables the creation of highly customizable displays. The ideal characteristics of an independent pixel for customizable displays are: small, portable, reusable, self-powered/unpowered, and untethered.

A design space of customizable displays can be expressed as a continuum, defined by the level of pixel independence (Figure 1.2). At one end, traditional display screens use tethered pixels that draw power from a wall outlet and receive data through wired connections, offering minimal pixel independence. On the other end, completely manual displays use untethered pixels that are unpowered and do not share a physical connection with the neighbouring pixels, offering a high level of pixel independence.

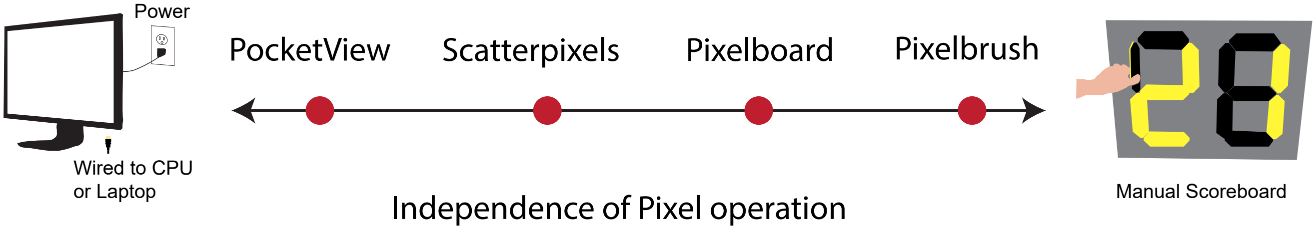


Figure 1.2: Design Space: a continuum based on the level of independence of pixel operation. The left extreme of the continuum is a traditional display screen and the right extreme is a manual scoreboard

In this thesis, we explore different display designs that use pixels with different levels of independence, ranging from low to high. We investigate the space of customizable displays through four projects. First, PocketView explores integrating battery-powered tethered pixels into pockets in clothing. Then, Scatterpixels, Pixelboard, and PixelBrush explore integrating pixels into the architectural surroundings, such as walls, ceilings, and floors. The transition from wearable context to architectural surroundings requires increased flexibility in display configurations, wiring, and the technical expertise required for installation. To address these challenges, we explored the design of physical pixels with more independence, ensuring they can operate without relying on wiring with the base or other pixels. These pixels can enable seamless integration into architectural surroundings. We discuss the hardware design of these pixels, registration methods to locate them in space, and ways to control the displayed information.

Initially, we use tethered LED pixels to shine information through the fabric of a pocket. The LED pixels receive power from a battery and communicate with a base station through wires. The pixels are updated with the information sent from a phone. We experimented with LED pixels arranged in different shapes and sizes that can fit easily into a wide range of pockets in clothing. The display can show information such as fitness stats while running, an email notification while walking, and navigation instructions while biking without retrieving the phone out of the pocket.

Later, we explored the concept of ad hoc reconfigurable displays using untethered physical pixels. This approach allows for instant creation of diverse display configurations, ranging from one-bit displays to 2D displays. For example, a set of pixels placed on a wall can function as a game timer, and the same set of pixels can be reconfigured as a floor display to show a welcome message. Each physical pixel is a red LED ball, powered by a battery and wirelessly controllable from a phone. Once these pixels are arranged in the desired configuration and the display is registered, it can show images or animations. Since the pixels are battery-powered, they require charging every 5 – 8 hours.

To overcome the need for frequent recharging and increase pixel independence, we developed a battery-free pixel element controllable from a laser base station. These pixels harvest energy from ambient light sources through solar panels, storing it in a supercapacitor, to power the pixel circuitry. The base station sends modulated laser signals to control a semi-bistable electrochromic display element on the pixel. The pixels can be utilized as programmable ad stickers on buses, signboards in remote environments, or attached to a building exterior to form a media facade.

While battery-free pixels increase the independence of pixel operation, they still rely on an electrical energy source for driving the associated circuitry. To further increase pixel independence, we explored the design of a mechanical flip dot pixel that minimizes electronics on the pixel itself, resulting in a more compact form factor. These pixels use NFC tags for pixel detection, but their visual state is set when a human brushes over them using an external device. When the brush detects a pixel, it updates the visual state accordingly. Displays formed using these pixels are well-suited to show information that requires occasional updates. They could be used as programmable advertisement boards, signboards, and building media facades.

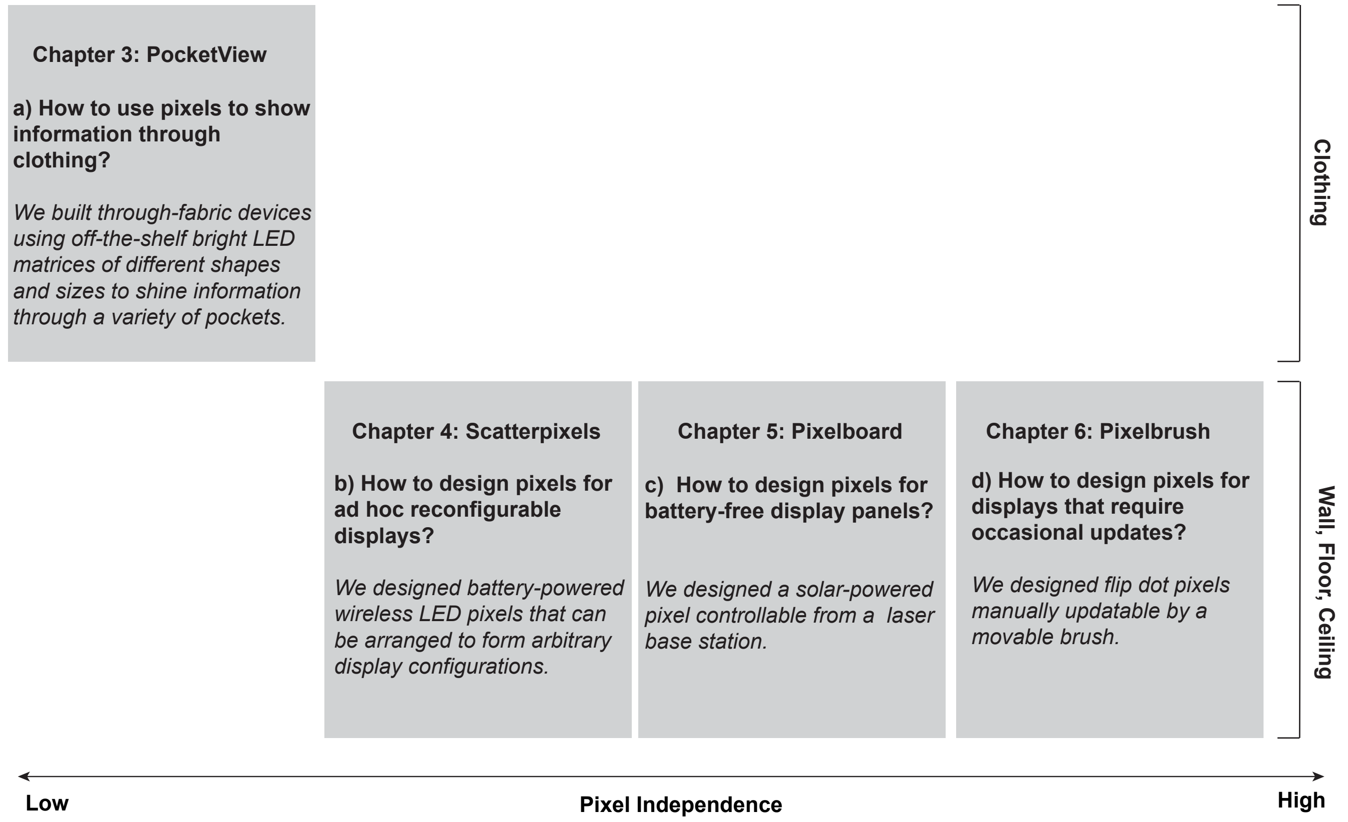


Figure 1.3: Research Outline: Research path showing the projects used to explore different levels of pixel independence for integrating them onto human body (clothing) and the architectural surroundings (wall, floor, ceiling).

1.1 RESEARCH OBJECTIVES AND OVERVIEW

The overall research problem investigated in this thesis is:

> *How to build displays that provide flexibility in creating custom display configurations in terms of shape, size, resolution, and location, to fit into the existing infrastructure?*

We investigate the overall research problem through a series of research questions focused on different domain applications:

(a) How to use pixels to show information through clothing?

(b) How to design pixels for ad hoc reconfigurable displays?

(c) How to design pixels for battery-free display panels?

(d) How to design pixels for displays that require occasional updates?

We built different hardware and software prototypes to answer these research questions (Figure 1.3):

1. To answer question (a), we built through-fabric devices using off-the-shelf bright LED matrices of different shapes and sizes to shine information through a variety of pockets.
2. To answer question (b), we designed battery-powered wireless LED pixels that can be arranged to form arbitrary display configurations.
3. To answer question (c), we designed a solar-powered pixel controllable from a laser base station.
4. To answer question (d), we designed flip dot pixels manually updatable by a movable brush.

1.2 CONTRIBUTIONS

We explored our research questions through four different projects, and this section describes the contributions.

1.2.1 *Modular Clothing Displays using a Through-Fabric Pixel Panel*

In chapter 3, we explore the idea of integrating displays into the fabric of a pocket. We conducted a survey with 112 respondents to find pocket locations in clothing and the objects commonly stored in it. We used off-the-shelf LED matrices to shine information through the fabric of a pocket. We built different form factors of through-fabric devices that resemble objects commonly stored in pockets: a phone prototype using an 8x8 LED matrix, a pen prototype using an 8x1 LED strip, a car remote prototype using a 15x7 LED matrix, and an earbuds prototype using a denser 8x8 LED matrix. We conducted a technical experiment to test the ability of the LED matrices to shine through common garment fabrics. We also conducted a user study with 12 participants to compare our LED through-fabric devices against two extreme baselines of through-fabric devices: a standard phone displaying low-resolution icons using high-contrast imagery and a PDLC pocket that turns transparent to show the phone screen stored inside the pocket.

1.2.2 *Reconfigurable Physical Pixel Displays*

In chapter 4, we explore the concept of ad hoc reconfigurable displays to create different display configurations on the desired surface instantly. To achieve this goal, we designed battery-powered spherical LED pixels (4cm diameter) that can be wirelessly controlled from a phone to show content at 20 FPS. Once the pixels are arranged in the desired configuration, a spatial registration process finds the relative locations of the pixels in space. We built a system to interactively guide users to position the pixels in optimal locations when the dictionary of images to be shown is known beforehand. These pixels enable different applications and configurations, which include one-bit displays, 1D displays, 2D wall displays, 2D floor displays, and 2D ceiling displays. These pixels allow displays to be customized in shape, size, resolution, and location.

1.2.3 *Solar-powered Independent Pixel Module*

In chapter 5, we describe the design of a solar-powered independent pixel module controlled by a laser base station. Each pixel contains two solar panels to harvest energy from ambient light sources and store charge in a supercapacitor, photodiodes to receive control signals from the base station, and an electrochromic display element. The base station uses a laser galvanometer to accurately position and send modulated laser signals to the photodiode to update the electrochromic display. A laser beam scanning process registers the spatial location of the pixels. We built two exemplar pixels as a proof of concept. We conducted technical experiments to estimate the charging time for different lighting conditions, the number of updates per charge, the time required to reach a steady state in different lighting conditions, and the ability to update the pixel from an angle. The aspirational goal is to embed these battery-free pixels into construction materials like wood or drywall to form “display panels".

1.2.4 *Manually updated displays using a handheld brush*

In chapter **??**, we describe the design of mechanical flip dot pixel elements that can be embedded into construction material or existing building infrastructure to form a display. The pixels can be arranged in a grid or arbitrary configuration on the desired construction material (e.g., wood, plexiglass) or directly integrated into existing building surfaces (e.g. concrete wall). The information on the display is updated by manually brushing over it using a handheld brush. Each flip dot pixel contains an NFC chip and a tiny permanent magnet. The handheld brush has an NFC reader antenna stacked with an electromagnet and an Aruco marker tag to track the brush location. When the brush goes over the display surface and detects a pixel underneath, the electromagnet is triggered to set the visual state of the pixel. We conducted technical experiments to test the ability of the brush to update pixels at different brushing speeds, brushing angles, and brushing offsets.

1.3 DISSERTATION OUTLINE

The remainder of the dissertation follows the outline given below:

In Chapter 2, we discuss prior work related to combining pixel elements to form customizable displays and technologies required to design pixels for such displays.

In Chapter 3, we describe the design of through-fabric devices that shine smartphone information through the fabric of a pocket.

In Chapter 4, we describe the design of ad hoc reconfigurable displays that use battery-powered LED pixels to create arbitrary display configurations.

In Chapter 5, we describe the hardware design of a solar-powered independent pixel module and how it can be controlled from a laser base station.

In Chapter 6, we describe the design of a mechanical flip dot pixel that is updated using an electromagnet brush.

In Chapter 7, we discuss the technical challenges and limitations of our display designs, ways to make them more interactive, and propose directions for future work.

In Chapter 8, we conclude the thesis and summarize the findings.

2

RELATED WORK

In this chapter, we describe previous approaches and technologies to build customizable displays. First, we describe approaches to building customizable displays using physical pixels. Second, we review different technologies to enable the design of customizable displays.

2.1 CUSTOMIZABLE DISPLAYS

We recognize that definitions of a customizable display could include systems that actuate or combine conventional display units. For example, using actuated projectors to form displays on different surfaces [42, 99, 136], creating a modular set of back-projection display "bricks" [78, 112], combining multiple device screens together to create large high resolution displays, [73, 80, 104, 113], robotic large display panels that "shape shift" [123], even using drones with projectors to create flying high-resolution displays [66, 90, 111]. However, our focus is on previous work that explicitly, or conceptually, considers each display pixel as an independent and distinct element that can be re-positioned to create different displays.

2.1.1 *Using many phones or tablets as pixels*

One possible approach in some settings, is to create a large ad hoc display using many phone or tablet screens as individual pixels. Schwarz et al. [70, 115] and Chungkuk Yoo et al. [140] investigate variations of this general idea, where many people in a crowd run a special application that communicates a unique code (such as flashing different colour transitions) to a centrally positioned camera, so the relative location of all phones can be determined. Once registered, and assuming everyone maintains a similar pose, imagery and patterns can be presented on the display created by a collection of phones, each acting as a single pixel. This type of display is truly ad hoc, meaning there is little a priori control of its shape or how it might change over time. However, the usage setting and possible configurations are not general, and using phones as pixels may not be practical due to size and cost.

2.1.2 *Using drones or robots as pixels*

If dynamic deployment and real-time control of the display shape are important, independent physical actuation of pixel elements is possible. Bitdrones [30] is an actuated 3D display using nano-quadcopter drones, demonstrations used up to 12 drones each acting as a single RGB pixel. The system highlights methods for real-time tracking and absolute position control when faced with challenging conditions resulting from many small drones flying close together. Bitdrones can only create sparse displays since drones can not fly too close to each other due to turbulence and downdraft. The system requires a high-quality Vicon motion tracking setup in the environment, and each drone can only run for about seven minutes before recharging. Commercial groups have used many drones to create dynamic outdoor displays [54]. These are intended for large public events since significant planning and setup with a team of people is required, and there are still limitations for run time. However, the outdoor setting enables the use of a new generation of error-corrected GPS positioning.

Tangentially related to our work are systems that use small robots, each acting as a single pixel that can be actuated to form dynamic two-dimensional displays. Morphogenesis [118] uses circular robots called kilobots [106] that contain a battery, vibrating motors, multi-color LED, and an IR LED to communicate with neighbouring pixels. Communication is line-of-sight with an operating range of 10 cm. The system does not centralize the control of robot positions to form specific shapes for a conventional information display. Instead, the robots self-arrange in periodic patterns or shapes to mimic phenomena like cell behaviour in tissue growth.

Other systems centrally control and track individual wheeled robots. For example, PixelBots [4] use an overhead camera, Hiraki et al.'s robots [45] decode invisible projected light patterns [62] from a specialized projector, and Zooids [69] use a related method of detecting projected grey-code patterns. These robots can be compact: for example, Zooids [69] are 2.6 cm diameter cylinders. Scatterpixels uses the same radio transceiver module, voltage regulator, and battery charging chip as Zooids.

A customizable display composed of self-actuated robot pixels is well suited for applications like physical animations or dynamic optimization of layouts. Indeed, a central focus of these works are algorithms to optimally re-arrange robots to convey a given image or display dynamic animations [3]. However, self-actuation has a significant trade-off with cost, complexity, and flexibility. An instrumented environment is required for accurate registration and continuous tracking (dead reckoning remains a difficult problem and self-localization is limited by physical constraints like line-of-sight). In some cases, the increased time to perform actuated movements may detract from the display. In many cases, miniature robots will not work on diverse surfaces like carpets or grass without specific customization. A stationary customizable display trades self-actuation capability for a system that is simpler, less expensive, and useful for a different set of applications across different display form factors. For example, the Scatterpixels system requires pixels to be positioned by hand and a smartphone app is used to register the static location of the pixels.

2.1.3 *Using individual stationary LEDs as pixels*

There are several examples of art displays composed of physically separate pixels. LED Throwies [126] are individually powered LEDs that can be attached to ferromagnetic surfaces to create sparse, abstract displays by manually positioning them. They are always on until the battery drains. Six-forty by four-eighty [13] is an art installation with 220 individual pixel "blocks." Each has a small screen to display animations and colours when touched. DisplayBlocks [101] are cube-shaped pixels with small OLED screens on each side. Each cube displays images or videos independently, from data pre-loaded onto an internal SD card. None of these examples use pixels that are spatially registered and none feature communication between pixels, so creating coordinated, dynamic displays is not possible.

Physically connected with fixed layouts — Other examples connect individual pixels by wires to enable communication but use fixed or manual spatial configurations. Lightset [47] hangs chains of wired LEDs on exterior building walls to prototype and explore ideas for urban displays. LED positions are known on each chain, and the demonstrated layouts are regular 2D grids created by multiple chains, both of which remove the need for custom registration. The distance between pixels can only be adjusted between 5 to 30 cm because of wiring, making this approach unsuitable to create diverse display layouts. Sato et al. [110] create large displays for an airport ceiling by arranging individual LEDs to display imagery like stars and simple animations. This semi-permanent, purpose-built installation places each LED at a pre-computed position according to the specific ceiling site.

Physically connected and customizable — More related are systems that form larger images using individual display modules as a pixels. Siftables [84] are 36 mm square "tiles" each containing a 128 × 128 px colour LCD display, IR transceiver to communicate with nearby tiles within a 1 cm range, and RF modules to communicate with a central base station. These can be arranged to form larger displays in rows or grids with each tile rendering a section of the combined image or application. Pickcells [29] is a similar system composed of small modular colour LCD screen tiles created from commercial smartwatches that can be physically connected to form different shapes. Both projects demonstrate using each tile to render an image or a piece of a larger image. In theory, a large number of tiles could be connected to form a very large display with each LCD tile forming single RGB pixels, but this was not the focus and was not tested or demonstrated.

Wired and customizable — Yet other examples use physically connected or wired "pixel" modules and support some limited forms of registration. Pushpin [74, 75] is a modular system for designing table-top wireless sensor networks composed of nodes built by stacking individual 18 × 18 mm modules for power, communication, processing, and application-specific functions like an LED or light sensor. Nodes are centrally programmed by an IR spotlight with communication among nodes using capacitive coupling or IR LEDs. Each node is powered from a common "power plane" of layered aluminum foil and polyurethane foam, which reduces the reconfigurability within the area of the plane. Blinky [63] are 40 mm cube-shaped physical blocks each containing multiple RGB LEDs to produce the same colour in all directions, an orientation sensor, and contact connectors to communicate with neighbouring blocks. Multiple blocks can be reconfigured into rectilinear shapes by connecting them in lines and stacks. The kind of displays demonstrated are limited since the focus is on using the system to teach programming concepts.

Twinkly [143] is a commercial product for creating light decorations. It uses a string of LEDs or lamps wired together in configurations suitable for different scenarios, like seasonal decorations for a building exterior or enhanced Christmas tree lighting. Multiple strings are interfaced with a controller, which communicates with a phone. Calibration uses the phone camera, but the method is not specified. Firefly [11] is a semi-permanent display formed using hundreds of individually addressable lighting elements. Each lighting element contains a microcontroller and an LED, which connects to a common rail to obtain power and control signals. A sample display installation was formed by attaching 2940 lighting elements to a building exterior spread over $40m^2$. The spacing between the lighting elements can be adjusted to form arbitrary display configurations and accommodate for existing building infrastructure. The physical location of the pixels is obtained by decoding the flashing sequence of pixels using a centrally positioned high-quality camera. Scatterpixels uses a similar method to register the spatial location of our pixels but supports a combination of multiple captures for flexibility in deployment, camera requirements, and display shape. Wiring constraints between the pixels reduce the flexibility to create different display layouts and usage in different applications.

Wireless and customizable — A more flexible approach is to design each LED pixel with on-board power and wireless communication. Bloxels [71] are translucent cube-shaped pixels which are stacked to form arbitrary-shaped volumetric displays. Each cube contains two RGB LEDs, nine IR LEDs for communication, and a battery. Invisible light patterns are projected from under the table to communicate with the bottom Bloxels, with information passed to upper Bloxels using the IR LEDs. SteganoScan Orbs [64] are transparent spherical balls that can be rolled inside a large parabolic dish. Each ball contains six LEDs and 18 photo sensors, and they are tracked in real-time and the LED state is updated by decoding invisible projected light patterns. The parabolic dish causes the orbs to pack close together when at rest to form a regular display grid. Urban Pixels [116] is an art installation to "paint building surfaces" with LED pixels. The battery-powered LED pixels are 4-inch acrylic balls, and they support wireless communication from a base station to display coordinated images and animations. The spatial locations of the pixels are hard-coded, no spatial registration method is described. NetworkedPixels [26] create abstract light patterns across a large open garden area. The system is a network of 923 wireless LED nodes controlled asynchronously by a base station. Given the outdoor application and large distances between nodes, onboard GPS is used for a simple spatial registration based on distance from the base station. Using the system as a single display to show coordinated imagery is discussed but not implemented. ParticleDisplay [108] uses 100 individually powered LED nodes (S-Node RFID module), each over 3cm square and 1cm thick, that can be controlled wirelessly from a base station. The base station communicates with the LED nodes at 4800 bps using a 303MHz radio module, which is unsuitable for displaying content in real-time or for animations. The spatial locations of pixels are obtained using a simple method: a camera captures the entire display to record a video as each LED illuminates one-by-one in a known sequence. The LED node also contains an acceleration sensor to sense input and directly control it.

Scatterpixel's spherical pixel form factor and battery-powered wireless approach are most similar to Urban Pixels, but we reduce the size by more than a factor of two, and we develop spatial registration methods that significantly advance the simple and constrained methods used by NetworkedPixels and Particle Display.

The design of the physical pixel is the key to building customizable displays. Most of the customizable displays discussed previously use simple monochrome LEDs for the display elements and still convey meaningful information. Different approaches highlight the flexibility of these displays to be installed outdoors, indoors, or on different surfaces. In our work, we designed a customizable display that can render real-time content. We also explored building battery-less pixels that can embed into construction materials to enable architects to construct walls or furniture with integrated displays.

2.1.4 *Wearable Customizable Displays*

This section describes previous approaches for designing customizable displays within the context of wearable e-textiles. Electronic sensors, actuators, and a power supply are integrated into clothing like a shoe, a shirt, or a hat to form wearable and programmable e-textiles. Electronic modules can be configured within the clothing for different applications.

Akira et al. [131] developed $7cm^2$ square fabcell modules, fabric material coated with specialized liquid crystal ink, which can switch between eight different colours by controlling the temperature. Fabcells are arranged on a substrate fabric (140cmx110cm), which is an array of conductive yarns that can selectively heat segments of the fabric. The fabcells can be arranged in an arbitrary fashion on the substrate fabric to form a wearable customizable display. This approach is fashionable with clothing-like aesthetics, but very sensitive to temperature and slow to respond. Prior work on e-textiles has explored designing electronically programmable fabrics. i*CATch [89] uses electronic modules that attach to sockets integrated into the fabric. The modules contain a microcontroller, sensors, and actuation elements like LEDs, which can be programmed to perform different actions. MakerWear [60] demonstrates using hexagonal electronic modules (25.5 mm) to retrofit into existing clothing. These modules serve as a power source, sensor, or actuating elements like LEDs, which tiled together operate as a single circuit. Similarly, Makershoe [61] attaches modules similar to MakerWear on an instrumented shoe. The shoe is stitched with hexagonal conductive patches to provide power to the module, instead of using a module with the built-in battery as in Makerwear. Rewear, Makershoe, and i*CATch modules can be used as LED pixels for a customizable display, which has not been explored yet.

The big challenge in designing wearable customizable displays is to make the result fashionable and comfortable using technologies that can easily retrofit into existing clothing while taking into account power and space constraints. In PocketView, LED matrices of different shapes and sizes were used to fit into different pocket types and shine information through them. However, since the system uses tethered LED pixels, these displays are not pixel-level customizable. Instead, they use pre-customized displays to fit into a wide range of clothing pockets. This is possible since the location of the display installation is known in advance. These devices enable seamless integration of digital information into clothing without instrumenting them and look fashionable.

2.2 TECHNOLOGIES FOR CUSTOMIZABLE DISPLAYS

A customizable display for integrating into the architectural surroundings requires the design of physical pixels, spatial registration methods to locate them, and methods to update their visual state to show the required information. This section discusses the available technologies for energy sources, display elements to show pixel visual state, communication link, display update methods, and spatial registration methods to locate the pixels. We discuss some display solutions mentioned in the previous section but mainly focus on the technologies used to enable them.

2.2.1 *Energy Source*

An energy source is required to drive the electronic circuit of an electrical physical pixel. Energy can be obtained from a power outlet, a battery, or a supercapacitor charged from an ambient energy source. The pixels can be powered through a power outlet, a battery, or using different energy harvesting approaches.

Wired to Power Outlet — The simplest way to power a display is to connect it to a power outlet. For example, a television or a decorative light receives power when connected to the wall outlet. These solutions require the pixels to be wired to a common bus to receive power. Displays using tethered LED pixels are connected to a wall outlet to receive power [11, 47, 143].

Battery Powered — Batteries are commonly used to build self-powered electronic devices (e.g. phones and smartwatches) and portable gadgets. They recharge slowly but provide power for long periods of time. The pixels of the display are wired together and connected to a battery for power. Maureillo et al. [81] used a LED matrix attached to a human's back or LED strips attached to a shoe [15] , which are powered by a battery. Some scenarios require the pixels to operate independently without sharing wiring with their neighbouring pixels. To achieve this, each pixel is powered by its own battery. The pixels are powered using a battery bank [26] or a rechargeable lithium-ion battery [116]. Some pixels are powered by a single-use coin cell battery [109], which will require frequent battery changes.

Energy Harvesting Displays — An alternative to powering pixels using batteries or a wall outlet is harvesting energy from ambient sources such as light or radio waves. The harvested energy is stored in a supercapacitor and used to drive the pixel circuitry. Supercapacitors can be charged quickly but provide power only in short bursts. They can be charged and discharged frequently without wear and tear. We discuss different strategies for building radio and solar-powered displays.

Radio signals are well suited for energy harvesting in a wide range of settings. Short-range RF communication systems like NFC can harvest power and communicate when an NFC transponder is in close proximity (< 15 mm) or almost touching a receiver coil interfaced with a display element. NFC-WISP [141] is an open-source NFC display platform that harvests energy and updates a small E-Ink display from a phone. Similarly, AlterWear [20] and AlterNail [22] are wearable E-Ink displays that harvest energy and communicate using NFC signals from a phone. Zanzibar [130] is an electronic mat that combines NFC and capacitive sensing to track and send data to tagged objects placed above it. One example shows a playing card with an embedded NFC antenna and an E-Ink display that updates when placed on the mat. These NFC solutions use individual display screens that are not tiled to form large display surfaces. UHF RF signals operating in the frequency range of 800–1000 MHz enable power harvesting over a longer range compared to NFC. Farsens [25] and PowerCast [145] develop RFID chipsets to harvest power from a dedicated RFID transmitter. The harvested energy can power LEDs and sensors in battery-free systems. However, power harvesting is not reliable when multiple devices are placed nearby due to interference, and it requires a large (≈15 cm) antenna.

Energy can be also harvested from Wi-Fi, cellular, or television RF signals (> 2.4 GHz). Olgun et al. [92] designed a prototype circuit (9 × 9 × 1.5 cm) with an array of patch antennas that harvest energy from ambient Wi-Fi signals. They demonstrated powering an off-the-shelf temperature and humidity device with a built-in LCD display. Liu et al. [76] designed an ambient backscattering system to harvest energy from ambient cellular and TV signals to drive a touch sensor and microcontroller and flash an LED. Similarly, the PoWiFi system [124] can harvest power and communicate using Wifi signals for temperature monitoring applications. Radio frequency-based energy harvesting systems require a large antenna, limit multiple devices operating near each other because of signal interference, and relatively complex circuit design.

Solar panels harvest energy from ambient indoor lighting or direct sunlight, which can drive energy-neutral displays. Grosse-Puppendahl et al. [33] designed an 11 × 12-pixel custom E-Ink display powered by a 54 × 52 mm solar panel attached to the back of the display. The circuit wakes up every 60 seconds to communicate over Bluetooth Low Energy Module (BLE) to update the display. Engage [137] is a battery-free implementation of a Game Boy (a gaming platform) that runs using power harvested from a solar panel and button presses. The harvested power runs a microcontroller, FRAM memory, and a low-power display for mobile gaming. The San Diego airport has 2100 rectangular 'E-Ink prism' color-changing display tiles attached to the exterior of a 1600-foot long building [144]. Each tile harvests energy from a solar panel to drive a microcontroller and communicates with a base station through wireless signals. These tiles do not operate together to form a large display for showing imagery but only show animations. The process of computing the spatial locations of these tiles is not specified.

Yogesh et al. [82] designed solar-powered display tiles that integrate into different surfaces, like a kitchen floor or table. They design a 4 × 4 array of display panels, with each 28 × 28 mm tile made up of a transparent solar panel and a low-power 128 × 128 memory display screen. The display screen in each tile is wired to a microcontroller and a PIR sensor. The PIR sensor detects basic hand gestures to update the display with notifications and reminders. This system is not easily scalable to larger surfaces, and it cannot form non-rectangular display configurations without having to rewire it to a central controller for every new configuration. In the Pixelboard system, we designed pixel elements that harvest energy from solar panels, operate independently from one another, and can be arranged in arbitrary configurations to form large display surfaces. We also implemented an energy-efficient strategy to update them from a central laser-based control unit.

This section gave an overview of different approaches for power management for an electrically powered pixel. The choice of energy source for a physical pixel is dependent on the application of the display.

2.2.2 *Display Element to show Pixel Visual State*

This section describes the different technologies available for the display element of the pixel. Non-Emissive displays modulate their optical properties to convey their visual state and emissive displays convert electrical energy to light energy. Non-Emissive displays are subtle, low power, and update slowly, whereas emissive displays are bright, have higher power consumption, and update faster. Some of these display technologies can operate in continuous mode in which the contrast of the display element is proportional to the trigger signal. But, for our applications, it is sufficient to operate in binary mode, either completely on or off.

Bi-stable e-ink displays are non-emissive displays that require power only to update their state, but not for holding it. They have low power consumption and are commonly used with energy-harvesting displays. AlterNail [23] and AlterWear [21] are small, simple, and minimal E ink displays that can be integrated into clothing like hats, shoes, and shirts. Yogesh et al. [82] used a low-power memory display that is driven using energy harvested from solar panels. An electrochromic display is a semi bi-stable display technology whose contrast falls with time. It switches its visual state when an electric current is supplied to it. Transprint [55] enables custom fabrication of flexible and transparent electrochromic displays that can switch from being completely transparent to a dark blue opaque state. Pixelboard system uses an electrochromic display element that changes from blue to grey and vice versa when supplied with a reverse polarity. Mechanical flip dots are bi-stable display elements that require a lot of power to update them. Marius et al. [46] design low-resolution displays using electromechanical flip dots, which can flip between white and black sides. When power is supplied to an electromagnet, it rotates the disc on the flip dot and sets its visual state and bi-stable since it is mechanical. The PixelBrush system uses a modified version of the flip dot that moves the electromagnet to a movable brush. The brush moves over the pixels and sets its visual state.

Thermochromic ink can be used as a non-emissive display element whose colour changes when heated [96, 131]. For example, Ebb [18] demonstrates how thermochromic yarn can be woven to create a low-resolution, non-emissive textile display. Ambikraf [97] uses thermochromic ink controlled by a peltier element to enable rapid heating and cooling.

Electroluminiscent (EL) panels are emissive displays that glow when a high voltage is applied to them. Printscreen [91] enables custom displays with arbitrary 2D or 3D shapes and sizes on different substrates like wood, stone, or paper. The display can be fabricated with a single segment, multiple segments, or a matrix of segments. Each segment can be individually controlled for displaying content. Protospray [38] enables the design of displays on curved and irregular surfaces by spraying an electroluminescent layer on conductive 3D-printed structures. EL panels of 45cm by 45 cm in size, can be chained together to build large displays up to $2m^2$ [5]. Shirts are printed with a design on a special EL panel, which can glow when connected to a battery pack [142]. Vitaboot [56] embeds an EL panel on the shoes which lights when the heart rate is above a threshold to encourage physical activity. As the number of pixels or segments scale up in the display, wiring, and controller circuitry becomes complex.

LEDs are active display elements and require a simple interface to control them. LEDs are emissive display elements that are bright, easy to control, and can be easily combined together to form large-scale displays. LEDs enable RGB states and allow faster refresh rates. The downside is high power consumption, so they are not ideal for energy-constrained situations. A matrix of LEDs can form a low-resolution display, which can be attached to a building exterior to act as a social display [7]. Mauriello et al. [81] used an LED matrix display fixed to the back of exercise clothing to display fitness statistics. Online fashion brands like LED Clothing [103] sell clothes and shoes with integrated LED lights for fashion and costumes. LED strips attached to a shoe act as ambient displays to show their running speed [15]. LED displays are attached inside a t-shirt, which can be viewed through a transparent slot [17]. Scatterpixels uses a LED as a display element.

2.2.3 *Communication and Display Update Methods*

We discuss methods to communicate with pixels and update their visual states. The communication link is used to send control signals to update the display element of a pixel or identify the ID of a pixel.

Radio communication includes NFC, WiFi, Bluetooth, and RFID systems. AlterNail [22] and AlterWear [20] use NFC communication to update their displays with information sent from a phone. NFC communication is intended for very short-range applications. Pixelbrush system uses pixels embedded with an NFC tag to uniquely identify it. Grosse-Puppendahl et al. [34] used a BLE module to communicate with a tablet or phone to update its display. Zooids [69] use a 2.4 GHz proprietary RF module to communicate between a base station and a group of mobile robots. RF backscattering systems enable battery-free operation and communication. These systems harvest power from radio signals and communicate information by selectively reflecting the received signal back to a reader or an access point. Ambient backscattering [117] enables battery-free communication between a specialized transmitter and receiver by backscattering TV and cellular signals. Similarly, Wi-Fi [1] and RFID [83] backscattering techniques enable the design of battery-free communication systems.

Wireless optical communication can control a physical pixel in controlled lighting environments and when light-of-sight operation is possible. These systems use a combination of a modulated optical light source along with a light-sensing device to decode the information. DarkVLC [127] enables optical communication by modulating the LEDs to encode information which is then decoded using photodiodes. This system makes visible light modulation imperceptible to the human eye by sending very short light pulses that can only be detected by a photodiode. Similarly, a visible laser beam modulated using On-Off-Keying (OOK) can be decoded by an array of LEDs [85]. Infrared (IR) LEDs can be used for invisible light communication, similar to traditional TV remote operation. Siftables [84] use infrared transceivers to communicate and track their neighbouring pixels, by sending and receiving IR codes.

For systems that share a wired interface, communication can be achieved using serial I2C communication or a 1-Wire interface. i*CATch [89] connects electronic modules in a wired mesh network. The mesh network is connected to a base station that communicates with individual modules through I2C communication. MakerWear [60] and Makershoe [61] use a 1-Wire interface to communicate between the interconnected modules.

Electrical or radiant energy signals (radio, light waves) are used to send control signals to set the visual state of a pixel. But, in certain scenarios, the visual state of a display element is set using manual methods. For example, a handheld brush mounted with a servo motor goes over a carpet [120] or a grass turf [121] to either raise or flatten its fibers to show an image. SweepScreen [88] uses a magnetophoretic material as a display surface which requires a human to brush over it to show information. The brush has a series of electromagnets to create a magnetic field to update the display surface to show the desired image.

The communication protocol and display update methods discussed above are not exhaustive but give an overview of existing technologies to establish them. The choice of communication link and update methods for the display element depends on the usage scenarios, power constraints, and environmental conditions.

2.2.4 *Spatial Registration*

Once the pixels are installed in the desired configuration, a spatial registration method finds the positions of all the pixels and their IDs relative to each other and possibly relative to the environment. This generates a spatial map that keeps track of the relative pixel locations and their corresponding IDs, which are later used to render content.

Physical objects embedded with a microcontroller can be stacked together, such that they share a common bus for communication. When one object is connected to a base station, it can find the relative locations of all the interconnected tiles. Triangles [31], Anderson et al. [6], RFIBricks [49], and ActiveCube [132] build physical blocks that can be stacked together to form arbitrary configurations that are tracked by a base controller. These systems do not operate as a display, but the tracking methodology is useful for spatial registration of physically connected pixels.

For certain applications, physical pixels have to be separate, so they cannot share a common wired communication link. In these cases, computer vision methods register these pixels by detecting their locations within a camera frame. Particle display [108] uses an RGB camera to detect the light flashes from pixels to spatially locate it. LightAnchors [2] and Infoled [139] encode information in an LED by controlling its flashing sequence. The flashing sequence is decoded using a smartphone camera. In our work, we used computer vision methods to register the locations of the pixels. Scatterpixels uses an RGB camera to locate the pixels similar to Particle Display; Pixelboard tracks a laser pointer and detects IR flashes using RGB and IR cameras respectively, and Pixelbrush uses Aruco markers to track the location of the handheld brush.

The spatial locations of wired pixels arranged in different configurations can be obtained using computer vision methods. Firefly [11] spatially registers the pixel by decoding an LED flashing sequence using a centrally positioned RGB camera. Smartphone cameras are used to spatially register LED chains arranged in arbitrary configurations [143]. But, the registration process is more difficult when the pixels are spread over a very large area and challenging lighting scenarios. So, NetworkedPixels [26] uses GPS sensors to locate pixels spread over a wide garden area.

3

POCKETVIEW:THROUGH-FABRIC INFORMATION DISPLAYS

3.1 INTRODUCTION

Mobile phones are an indispensable part of daily life, and we carry them everywhere. But, accessing information on them is not always convenient. For example, when a phone is in a pocket and emits a sound or vibration to signal a new notification, the phone must be retrieved from the pocket to see the information. This retrieval process can be socially awkward during meetings, it can be cumbersome when carrying something in your hands, and it can be difficult, or dangerous, when walking or biking.

The question is, how can smartphone content be made visible when the smartphone itself is stored in a pocket? Possible solutions include wearing a smartwatch, headphones, or augmented reality glasses to receive smartphone information. However, this introduces additional cost, technical complexity, and requires additional visible accessories to be carried or worn by the user, which may not be suitable in all settings. Other more radical ideas could add a flexible LED display to clothing [81], or integrate displays directly into the fabric using thermochromatic ink [18], E-ink [21], or woven optical fibres [67]. Instead of placing a display on fabric, or weaving a display into fabric, we explore how to make phone information visible *through* fabric, so it is always accessible even when the phone is stored inside a pocket. This could be used for applications like viewing notification types or turn-by-turn directions (Figure 5.1). A through-fabric device can complement other wearables as well. For example, viewing smartwatch information hidden under a sleeve or augmenting a headphone-based audio interface with additional visual information.

We conduct a small preliminary survey followed by a more extensive main survey to understand different types of pockets in clothing, the objects stored in them, and the need to access information when the phone is inaccessible. We find that for almost all participants (>90%), irrespective of age and gender, phones are the most popular object stored in various types of pockets. Men prefer storing phones in pockets located in the lower body area while women prefer the stomach area. We then conduct a technical experiment to validate the ability of an LED matrix to shine through common fabrics. The results show that LED pixels can shine through common fabrics, while light transmission is affected by fabric thickness, knit, and weave type, and irregularity is affected by patterns such as checkered designs. Motivated by the survey and technical experiment, we designed an initial through-fabric display prototype using a matrix of bright LEDs that users can place in their pocket and interact with using simple knock gestures. We evaluated the prototype in a 12-person user study to validate the general approach, including a baseline using a

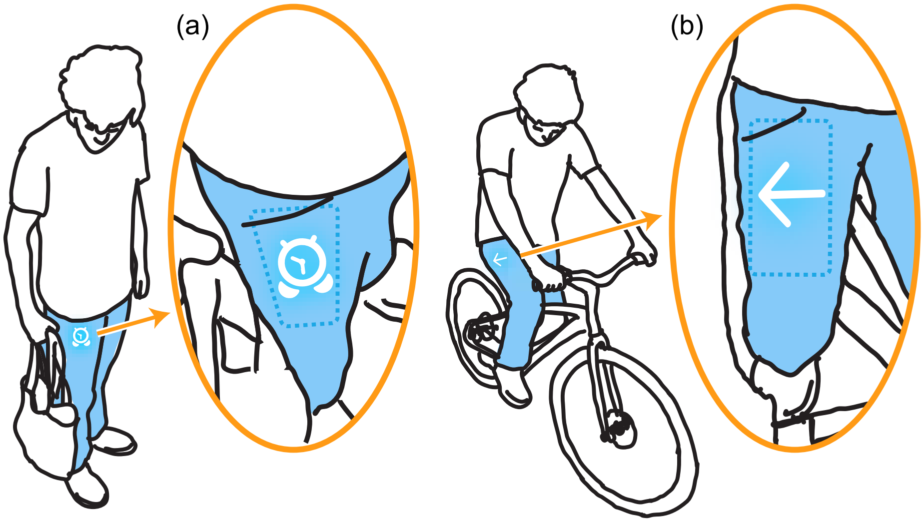


Figure 3.1: A through-fabric display for a pant pocket: (a) receiving a notification during encumbered walking; (b) viewing directions while bicycling.

standard phone display with bright, high-contrast imagery and a futuristic pocket that can be made transparent on demand using Polymer Dispersed Liquid Crystal (PDLC) film. Our results show the feasibility of the concept, with participants favouring the LED matrix for comfort. Comments about the futuristic PDLC pocket approach show there is a desire for selectively viewing information through a pocket in terms of usability, ease of interaction, visibility, and amount of information. We built different form factors using LED matrices that can attach to an earbuds case, pen, and keyfob. Using multiple, smaller objects makes through-fabric displays more inclusive to objects commonly stored in smaller pockets, typically found in women's clothing [19]. We contribute what we believe, is the first investigation into creating a through-fabric pocket display. These wearable displays are a hybrid between smart textiles, ambient displays, and traditional wearable devices like a smartwatches.

3.2 BACKGROUND

We discuss prior work related to smart textiles, on-body displays, "see-through" displays, and interaction through pockets.

3.2.1 *Smart Textiles*

Clothing can be instrumented with smart materials or sensors to sense input. For instance, garment fabrics can be augmented with iron-on sensors, as in Klamka et al. [65] and Polysense [48], or even sewn or woven into garment fabrics, like conductive threads in Project Jacquard [102] and Pinstripe [59], electrospun nanofiber-based materials [12], and others for detecting moisture [138] or pressure in RESi [95]. A common goal of smart textile input is to control a smartphone, but the output remains tied to the phone. A through-fabric display complements these input methods by providing a method for integrating a display into clothing.

More relevant to our work, is past research on using smart textiles as displays. One approach is thermochromic textiles that use heating elements to change colour, and create displays using the fabric itself [96, 131]. For example, Ebb [18] demonstrates how thermochromic yarn can be woven to create a low-resolution, non-emissive textile display and Ambikraf [98] animates patterns on common fabric with the help of thermochromic inks and peltier semiconductor elements. Using thermochromic textiles enables fashionable, clothing-like aesthetics, but they are very slow to change, and tend to be more suitable for ambient information. Methods like Optical Fiber Displays [67] aim to spin optical fibres directly into clothing to serve as flexible displays. However, Braunder et al. [9] survey the broader area of interactive smart textiles and conclude that there is a lack of reliable conductive yarns technologies and they can currently be used for demonstration purposes only.

3.2.2 *On-body Displays*

Apart from smart textiles, researchers have also integrated LED or E-ink displays on clothing. For instance, Mauriello et al. [81] used LED-based displays fixed to the back of a shirt or jacket to display fitness statistics and Colley et al. [16] integrated RGB LED strips into shoes to help runners visualize their pace. Grosse et al. [32] studied suitable locations to wear display and built LED display prototypes for the arm and back. When worn, they can indicate turn and stop signals while biking. Similarly, the Idle stripe shirt [39] uses fibre-optic threads to generate display patterns. Online fashion brands like LED Clothing sell clothes and shoes with integrated LED lights for fashion and costumes [103]. AlterNail [23] and AlterWear [21] are small, simple, and minimal E-ink displays that can be integrated into clothing like hats, shoes, and shirts. AlterWear is battery-free and relies on NFCs for powering and communication, however, fabrics still need to be instrumented to accommodate these devices.

Schneegass et al. [114] explore on-body displays to extend the display area of a smartwatch using a low-resolution LED matrix. They describe a prototype using a 16×8 LED matrix that shines through white t-shirt fabric, but the goal is to simulate low-resolution garment-based displays, not explore its through-fabric nature. Their focus is on finding suitable locations for on-body displays, visualization methods, and the efficacy of visualizing off-screen data in a navigation task. In contrast, we focus on the motivation and potential of a pocket-based through-fabric display, including light transmission capabilities, device form factors, and usability.

3.2.3 *See-through Displays*

A transparent material can enable access to a display in a stored location. Colley et al. [14] creates a transparent slot in a handbag to view a tablet display. They explore how this can be used to customize the bag colour for fashion, to view and interact with objects stored inside the bag (including a mobile phone), and as a social display with a personal message. Sugiura et al. [119] create a wrist-worn prototype for simultaneously showing private and public information. The system uses a sandwich of retro-reflective material and electronically controllable Polymer-Dispersed Liquid Crystal (PDLC) film with a head-worn projector for content. The PDLC film rapidly switches between an opaque state, in which projected content is visible to nearby people, and a transparent state where the retro-reflective material makes private projected content visible only to the user.

We use PDLC film to create a switchable version of Colley et al.'s slot in the form of an instrumented pant pocket. Unlike shining light through the fabric, a PDLC pocket requires the garment to be specially modified, making it less practical. However, in our usability study, it provides an extreme baseline for upper limits of the through-fabric approach since it enables a standard phone display to be easily viewed inside a pocket.

3.2.4 *Interaction On and Around Pockets*

Previous work has explored using front pant pockets, and the upper thigh in general, for sensing input. Thomas et al. [125] found using a mouse on the front thigh is most favoured by participants when sitting, kneeling, or standing. Smart pockets [129] uses pocket-based gestures (e.g., placing hands in a certain pocket) as input for a large ambient display. PocketThumb [24] is a touch interface integrated into a pocket to control wearables like AR glasses. PocketTouch [107] investigates the practicality of adding touch input to a pocket, or through the fabric of a pocket. The results suggest that using a specially modified capacitive sensor, smartphone touch input could work while in a pocket, through many fabrics. Ronkainen et al. [105] and Hudson et al. [50] explored using tapping (or "whacking") gestures as input for mobile devices. We also adopt this simple method to interact with a phone when in a pocket, but a through-fabric display could be extended to use more advanced input methods like PocketThumb [24] or PocketTouch [107].

3.3 PRELIMINARY SURVEY

We conducted a short preliminary survey to establish if there is a need to access information when a phone is inaccessible and to begin to understand phone storage preferences in different scenarios. The online survey had 10 questions about phone storage when walking or in a meeting, frequency of accessing phone information, and the need to access information when hands are occupied. There were 106 respondents, ages 17 to 68 (79 male, 23 female, 1 genderfluid, 2 did not answer). The results show that respondents generally want to access information on their phones in different scenarios. When walking, 28.3% indicated they wanted to access information on their phones every 1 to 6 minutes, and 35.8% every 6 to 20 minutes. When asked about the importance of accessing information on their phones when their hands were occupied, 37.7% indicated high importance (4 or more on a 5-point scale). These results show that many people want to access information on their phones, even when it may not be convenient to do so. In response to where respondents kept their phones in different scenarios, relatively few women used their pant pockets. While walking, 97.4% of the male respondents stored their phones in their pant pockets, whereas only 30.4% of the female respondents do the same. Similarly, during a meeting, most male respondents (57%) kept their phones in their pant pocket, but most female respondents (69.6%) kept it on a desk or table.

Overall, men commonly store their phones in pant pockets, but women less so. Two related studies, one interviewing people on the street [51] and the other semi-structured interviews and an online survey [135], also found men predominantly store phones in their pant pockets, while women prefer shoulder bags or purses. They note phone storage location is affected by societal perceptions of gender, culture, and age, as well as physical constraints due to pocket size and clothing. For example, women's clothing typically has smaller pockets [19].

While our preliminary survey motivates a need for accessing information from an inaccessible phone in different scenarios, the survey design was limited in terms of understanding phone storage preferences and gender diversity. The questions only asked about storing a phone in a limited range of clothing pockets (pant, shirt, and coat pockets), which women may not use, let alone wear, frequently. But there are many other clothing pockets of varying sizes and on different parts of the body that could be leveraged to create more inclusive through-fabric displays. Likewise, asking only about storing a phone in a pocket may be too limiting. There are other smaller objects that people place in pockets, like keys and credit cards, that could be augmented as well.

3.4 MAIN SURVEY

We conducted an extensive follow-up survey to understand whether people wear clothing with pockets, where pockets are located, and the types of objects stored in each pocket. Results from the survey are also used to understand the effect of gender on the pocket location and stored objects. In addition, this survey confirms the preliminary survey result showing a need to access information on an inaccessible phone, and expands this to include what alternative methods respondents are using now in that situation.

3.4.1 *Protocol*

The survey was conducted online, and disseminated to the general public through social media. There was no remuneration. It had three main sections with 39 questions total. The first section asked respondents about pocket locations on clothing they typically wear and what kinds of items they store in different pockets. These questions used illustrations of representative types of clothing, such as pants, jackets, and skirts, to convey pocket locations. The second section asked respondents about the importance, frequency, and methods for accessing information on their phone when it is inaccessible, like when in a pocket. The third section asked about demographics like age and gender. Respondents were told to consider their behaviour both during and before the COVID-19 pandemic.

3.4.2 *Results*

There were 112 people who completed the survey. The respondent sample has reasonable gender balance, with 57 identifying as male, 52 as female, 1 non-binary, and 2 did not answer. 93 respondents provided their ages. They span 19 to 71 years, but are skewed slightly younger overall with 68% between 19 to 35 years, 22% between 35 to 50 years, and the remaining 10% 50 or older. Although our survey was distributed internationally, we did not record the geographic location or climate of where our respondents live. We believe indoor garments are reasonably consistent across regions and cultures, but our sample may not adequately capture all clothing types (such as winter parkas).

Table 3.1: Percentage of respondents who wear clothing with eight pocket locations.

	Female	Male	Overall
Upper Thigh	96.2	98.2	95.5
Back of Leg	92.3	94.7	92.0
Stomach	90.4	80.7	83.9
Lower Thigh	40.4	35.1	36.6
Chest Area	36.5	66.7	51.8
Waist/Waistband	30.8	1.8	15.2
Arm	17.3	10.5	13.4
Back	0	0	0

Clothing Pocket Locations — The survey asked participants whether they wore clothing with pockets in any of 8 body locations: on the chest area (e.g., dress shirt); on the arm (e.g., sleeve pocket); near the stomach (e.g., front hoodie pocket); on the waist/waistband (e.g., waist pockets on leggings/-workout shorts); on the front upper thigh (e.g., front jean pocket); on the back

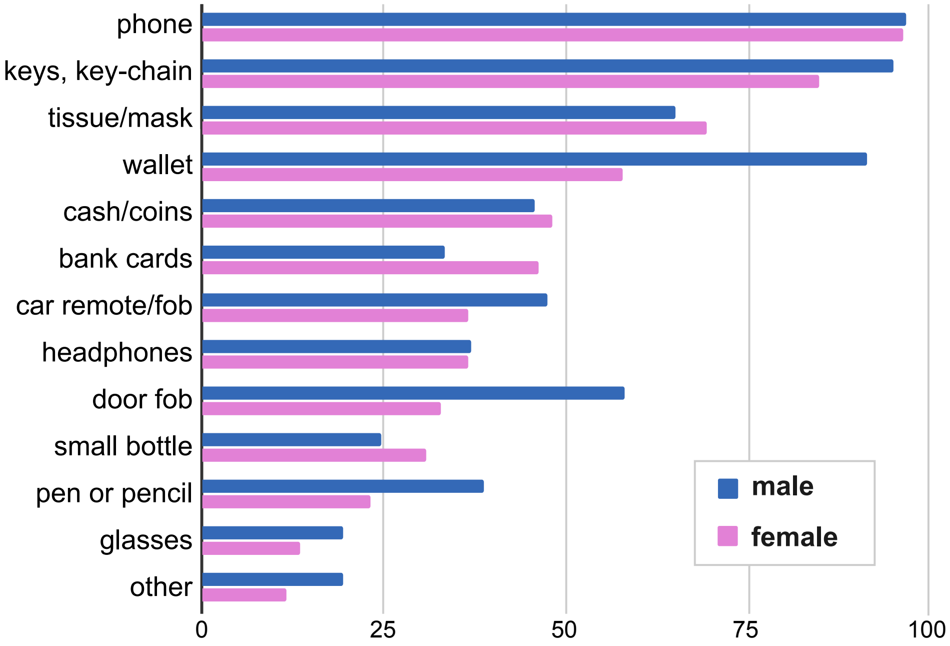


Figure 3.2: Types of items stored in pockets by gender (x-axis is % of respondents).

of the leg (e.g., back jean pocket); on the side of the leg (e.g., side pockets on cargo pants); and on the back (e.g., back of a sports bra).

Responses indicate that participants wear clothing with pockets located on the upper thigh area (95.5%) and the back of the leg (92%). Clothing with pockets on the arm (13.4%) and back (0%) were least-commonly worn. Among our respondents, women wore more clothing with pockets on the stomach (90.4% F, 80.7% M) and waist areas (30.8% F, 1.8% M). Men wore more clothing with pockets in the chest area (36.5% F, 66.7% M).

Items Stored in Pockets — If the participants indicated that they wore clothing with pockets in the specified locations, the survey asked them to select the types of objects stored in these types of pockets. Possible answers were nothing, or choosing one or more options from a list of 12 common types of items: phone; wallet; keys or key-chain; door fob; car fob/remote; loose bank cards; loose cash/coins; headphones and/or case; pen or pencil; glasses; tissue or face mask; or small bottle (e.g., hand sanitizer). An open text "other" option was also provided.

Overall, when considering the objects kept in any pockets, phones were most popular (94.6%), with other popular items being keys/key chains (88.4%), wallets (74.1%), and tissue/face masks (65.2%). For most objects, men and women reported similar storage preferences; for example, both men and women placed their phones in a pocket (on any location of the body) > 96% of the time. However, a higher proportion of men placed wallets in pockets than women (57.7% F, 91.2% M), but women were more likely to place loose bank cards in their pockets than men (46.2% F, 33.3% M). Men

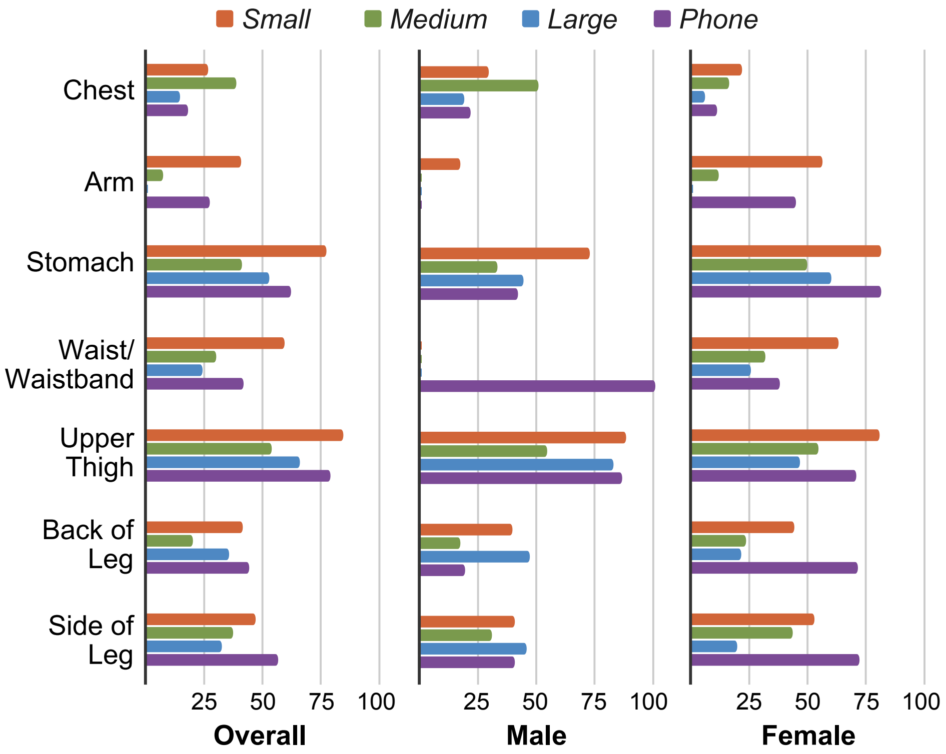


Figure 3.3: Pocket locations where phones and three sizes of items are stored overall, and by gender. The x-axis shows the conditional percentage of respondents who both answered they wear garments with pockets in a given area and that they store one or more items in that pocket. The back is excluded as no respondents indicated they wore clothing with pockets in this area.

were more likely to store pens/pencils in their pockets than women (23.1% F, 38.6% M), as well as door fobs (32.7% F, 57.9% M) (Figure 3.2).

To examine the specific objects placed in each pocket location, we first group the types of items into four categories by size for reporting purposes: "phone"; "large" for headphones/headphone case, wallets, and glasses; "medium" for bank cards, car remote/fob, pen or pencil, small bottle; and "small" for door fob, keys/key-chain, cash/coins, tissue/mask. We calculate the percentage of respondents that store a group of objects in a specific pocket location. Note that these percentages are the percentages of total respondents who reported wearing clothing with pockets at the indicated pocket location, rather than the percentage of all respondents. For instance, if the respondent did not report wearing clothing with arm pockets, they were not asked to indicate the types of objects they stored in arm pockets. Overall, people stored many different objects of varying size in different pockets (Figure 3.3). Respondents stored objects of all groups in every pocket type, with the exception of storing large objects in arm pockets. The upper thigh area is the only pocket location where the majority of respondents stored objects of all groups (all > 53%), and is the most common location for small objects (84.1%) and phones (78.5%). However, men were more likely to store large objects (46% F, 82.1% M) and phones (70% F, 85.7% M) in upper thigh pockets than women, who store small objects in these pockets instead. Women also tended to use a wider range of pocket locations to store objects; for example, they stored a wider variety of objects in arm and waist pockets than men (only 1 male respondent reported wearing clothing with pockets on the waist). Phone storage was spread across more pocket locations for women. With the exception of the single male respondent who reported using a waist pocket, men primarily relied on upper thigh pockets to store their phones, but women stored their phones in pockets located at the side of the leg, back of the leg, upper thigh, and stomach area (all $\geq$ 70%). The stomach area in particular, was the most common location for storing a phone for women and more commonly used than men (80.9% F, 41.3% M).

Accessing Information on an Inaccessible Phone — The survey asked a series of questions to understand the need and methods for accessing information on an inaccessible phone. In response to the question, "are there ever times where you cannot access your phone even though you wish to", a majority, 67.8%, responded yes and 40.1% felt that their ability to access their phones is moderately to extremely important.

A series of questions also asked how respondents currently access information normally viewed on a phone. Only 28.6% of our respondents wore a smartwatch and among those, 56.7% used it "about half the time" or more to access information on their phones. 29.1% of our respondents used a voice assistant "about half the time" or more to do the same. Among other devices, 93.7% participants use a laptop to access information they would typically view on phones, but this of course is only possible in a non-mobile context.

3.4.3 *Discussion and Implications*

These results validate the general idea of making information on inaccessible phones more accessible. Although a smartwatch or audio-based virtual assistant can fill this need, our survey suggests these methods are not frequently used. Our results also show phones are often kept in the various on-body pockets of both men and women. Most men placed their phones in the thigh area but more women used pockets in the stomach area for their phones. This confirms that our preliminary survey was limited in terms of understanding where women store their phone since it did not cover a comprehensive range of possible pocket locations. We use these results to motivate our initial design of a smartphone case through-fabric display for an initial prototype and usability test.

It is important to recognize the pocket used to hold a phone differs for women. This could be due to smaller front pockets in women's jeans and pants, making them hard to fit even medium sized-phones [19]. Our results do show that women place a phone in other front pockets that would be visible, but also that the back pocket is commonly used, a location which would make a personal through-fabric display on the phone case less practical.

However, we also find a large diversity of other items kept in pockets, including medium and large sized objects that would have enough surface area for a through-fabric display and internal space for necessary electronics. Importantly, we find that many of these items are kept in pockets that would be visible to the individual. We explore the idea of augmenting other objects like wallets, car remotes, headphone cases, and pens to create working through-fabric prototypes in Section 3.6. Before we describe any prototypes, we first report on an experiment that answers another set of fundamental questions about how LED light shines through fabric.

3.5 LIGHT TRANSMISSION EXPERIMENT

This section describes a technical experiment to validate and understand the ability of an LED matrix display to shine through common garment fabrics. Prior work has studied light transmission through fabrics to understand characteristics relevant to normal applications, such as curtains that block light or how sheer fabric may not work well for clothing. Relevant to our work are these general approaches and how both light sensors and image processing are used as measurement methods. Past work examining light transmission through knitted fabrics [52] and curtains [68, 122] used a lux sensor or light intensity meter. Some approaches process images captured from a camera to compute light transmission, such as an investigation of 40 different weave types [86] and polyester and cotton blends [35]. We use a camera to capture images of different light patterns shining through a fabric sample and also measure light illumination with a lux sensor. Using the images, we compute light transmission and irregularity values which are indicative of the optical properties of a through-fabric display in a garment.

3.5.1 *Apparatus*

A 3D printed rectangular frame was designed to hold an 8 × 8 RGB LED matrix (Adafruit 1487) measuring 71 × 71 mm. Each LED in the matrix operates at 300mW, all powered by a single 5V, 4 amp source. These LEDs are sufficiently bright to shine through a wide range of fabrics, and represent a best-case scenario for our tests. We trigger each LED as a binary "pixel", either completely turned off or as a white pixel operating at maximum brightness setting.

Each fabric sample is firmly secured to the display using a square hoop (Figure 3.4). A Canon Rebel T5i DSLR camera captures images of the LED patterns shining through the fabric sample. The camera view direction is co-linear with the fabric sample normal, with the camera 21 cm away from the fabric surface. A Rohm BH1750 digital light "lux" sensor is also placed 11 cm above the fabric to measure light intensity reflecting from, or shining through the fabric in lux. The lux sensor provides a relative baseline for light

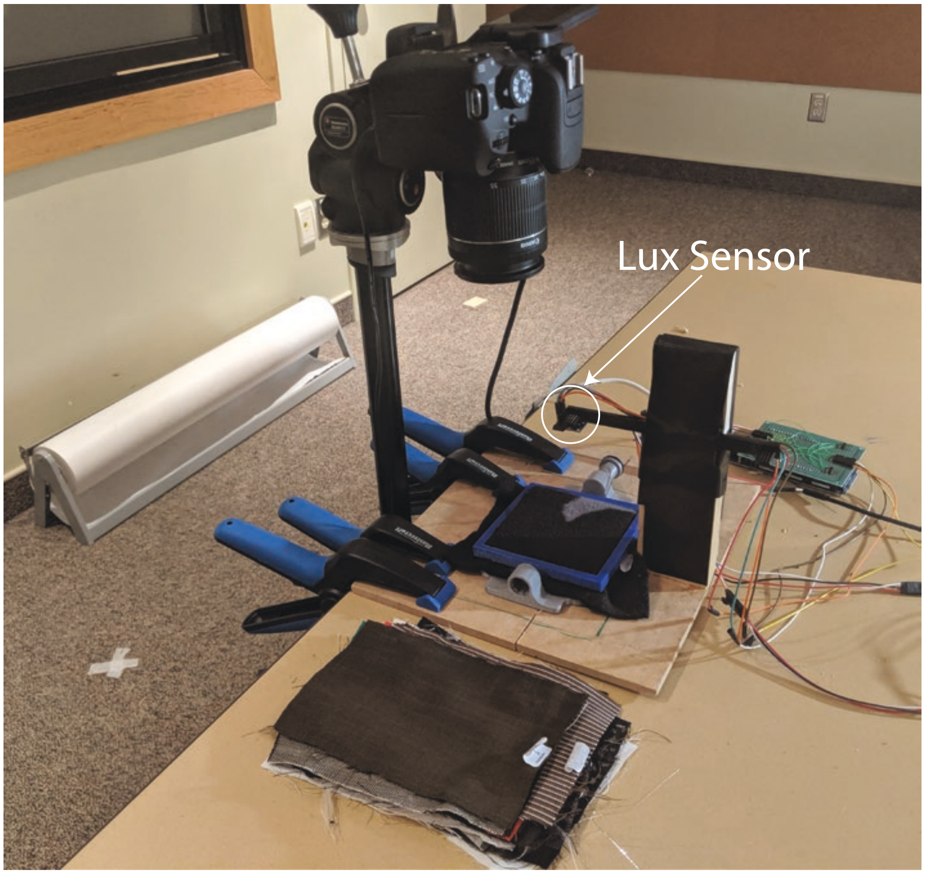


Figure 3.4: Fabric light transmission apparatus: each fabric sample is placed over an LED matrix display housed in plastic frame and a DSLR camera and light sensor are used for measurements. A lux sensor is used as a baseline for light measurements using the camera.

transmission values measured using the camera. The camera measures light at different positions of the fabric, which makes measures for patterned and irregular fabrics more reliable, and this is critical when measuring irregularity across individual LEDs. The images are captured inside a dark room, and the camera is set in manual mode with a 1/100 second shutter speed (TV=100), f10 aperture (AV=10), and 400 ISO. These values are chosen as the upper threshold in which no light enters the camera when the LED matrix is turned off. The images are captured with a resolution of 1728×2592 pixels and stored as a 24-bit JPG file. These images are cropped to extract the region within the rectangular hoop and then processed to calculate different metrics. A desktop C# application is interfaced with an Arduino Mega to control the LED matrix and to issue commands to the camera to capture images with specific settings.

3.5.2 *Fabric Samples*

Clothing fabrics are composed of one or more types of raw material fibres which are combined together using a manufacturing process. Fibre material is classified as natural (e.g. cotton), synthetic (e.g. polyester), or mixed fibre[1] (when fibre content is unknown and cannot be accurately determined). Manufacturing process is primarily categorized as woven (e.g. denim) or knitted (e.g. barcelona knit). We worked with an experienced salesperson at a large textile retail store to select a range of representative fabric samples that are typically used for garments.

The raw materials used for fibres in our samples include two natural types: Cotton (C) and Ramie (RA); six synthetic types: Polyester (P), Rayon (R), Spandex (S), Metallic Fiber (MF), Polypropylene (PLP), and Nylon (N); as well as Mixed Fibre (MIF) types.

The manufacturing processes used to combine fibres in our samples include ten woven types: Flannel (FA), Satin (S), Denim (D), Chiffon (CF), Poplin Prints (PP), Velvet (V), Metallic Jacquard (MJ), Georgette (G), Plaid (PL), and a generic weave (W); five knitted types: French Terry (FT), Kluffy Knits (KK), Lorie Lace (LL), Fleece (FL), Barcelona Knits (BK), and Tuscany Knits (TK); and a spunbond type (SB). Note some manufacturing processes use proprietary names.

In the results that follow, each fabric sample is labelled with an ID using the raw material and manufacturing process codes above, as well as the mm thickness in parentheses. For example, a fabric with ID 'D-C (0.59)' corresponds to a denim manufacturing process with cotton fibres with a thickness of 0.59 mm and 'W-CPS (0.67)' corresponds to a woven fabric with fibres composed of cotton, polyester, and spandex with thickness 0.67mm. Each fabric sample is cut into 20×15cm swatches to fit over the image capture frame.

3.5.3 *Results*

This section discusses the quantitative findings from the experiment in terms of light transmission and irregularity.

Light Transmission — Light transmission measures the amount of light that passes through the fabric. To calculate our relative light transmission measure, we first capture a reference image with the matrix turned on without any fabric sample on top. This is used with binary thresholding to find regions of interest for each LED pixel, and the intensity at each pixel is used to normalize light transmission measures. Then, each fabric sample is placed over the LED matrix and an image is captured with all the LED pixels turned on. Using the region of interest, transmittance is calculated as the ratio between sum of grayscale pixel intensity with the fabric to the sum of grayscale pixel intensity without the fabric.

---

[1] "mixed fiber" is a standard term, e.g.: `https://www.competitionbureau.gc.ca/eic/site/cb-bc.nsf/eng/01544.html`

Due to the reference image normalization, our transmittance measure ranges from 0 to 1: '0' implies that the fabric completely blocks out the light, and '1' implies that the fabric completely allows light to pass through the fabric. Transmission values near '0' would be more visible in a darkroom but not in sunlight, values from 0.3 to 0.6 would be visible in a well-light room, and values greater than 0.8 will be visible even in sunlight. Transmittance values of 1.01 are possible due to sensor noise in bright images and suggest an estimated measurement precision of $\pm.01$. The lux values correlate with our transmission metric, and they provide an absolute measure of overall light transmission.

Table 3.2 shows the light transmission values for the different fabric samples. Transmittance is high for very thin fabrics like 'CF-P (0.2)' and 'KK-MIF (0.39)', and very low for thicker and darker fabrics like velvet 'V-C (0.72)' and thick denim cotton fabrics 'D-C (0.67)', 'D-C (0.92)', and 'D-C (1.02)'. Barcelona Knit fabric 'BK-PS (0.4)' is thin, but light transmission is affected due to its knitting type. We expected waterproof fabrics to have low transmission, but for the two Nylon waterproof fabrics we tested, 'W-N (0.19)' and 'W-N (0.13)', one has very high full transmission and even the other has a lower, but still usable 0.21 transmission.

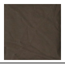
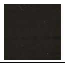
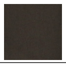
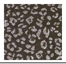
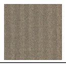
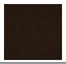
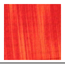
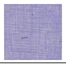
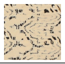
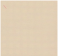
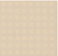
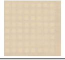
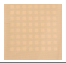
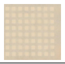
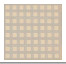
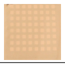
Irregularity — Irregularity measures how evenly light is transmitted through different areas of a fabric. Specifically, it is the standard deviation of light transmission for 16 individual LEDs that span the area of the display. This is computed by capturing a fixed sequence of 16 images, each with only a single LED illuminated. Similar to light transmission, reference images are used to obtain the region of interest (ROI) around each LED. The mean grayscale pixel intensity is calculated for the illuminated LED ROI in each of the 16 images. Irregularity is then the standard deviation of these 16 mean intensity values.

Table 3.2 shows the irregularity values. The irregularity value is low for fabrics that can shine light evenly across the fabric and vice versa. Irregularity is high for fabrics with dyed designs, textures or patterns, and low for solid fabrics without any texture. Fabrics with designs or patterns tend to have higher irregularity values because of the uneven light distribution across the fabric sample. Our Fleece Polyester fabric 'FL-P (0.94)' is dyed with an image of a bear and our poplin prints 'PP-C (0.2)' has a design with contrasting black and white regions, increasing the regularity value because of the patterns on the fabric. The checkered patterns on fabrics 'FA-C (0.49)' and 'MJ-PS (0.57)' also increase their irregularity values. For fabrics with designs or patterns, the irregularity could also vary based on the location of the fabric sample on the LED matrix.

3.5.4 *Discussion*

The experiment validates the ability of an LED matrix to shine through certain garment fabrics. Light transmission is affected by fabric thickness, knit type, weave type, and material. Regularity is affected in fabrics with patterning,

Table 3.2: Fabric experiment results for Transmittance, Irregularity, and Lux

Fabric (thickness mm)	**Trans**	**Irreg**	**Lux**	**Fabric** (thickness mm)	**Trans**	**Irreg**	**Lux**
 ‘BK-PS (0.4)’	0.0	0.54	0	 ‘W-CPS (0.67)’	0.0	0.18	0
 ‘D-C (0.67)’	0.0	0.73	0	 ‘V-C (0.72)’	0.01	2.8	0
 ‘D-C (0.92)’	0.05	3.48	1	 ‘W-P (0.43)’	0.08	4.7	3
 ‘D-C (1.02)’	0.09	9.07	3	 ‘D-C (0.88)’	0.14	5.88	5
 ‘FA-C (0.5)’	0.21	12.76	16	 ‘W-N (0.13)’	0.21	5.56	10
 ‘FL-P (0.94)’	0.28	49.73	14	 ‘MJ-PRMF (0.29)’	0.31	6.59	20
 ‘MJ-PS (0.57)’	0.32	36.38	46	 ‘PL-MIF (0.44)’	0.36	3.0	49
 ‘W-C (0.3)’	0.48	12.81	53	 ‘PL-PR (0.44)’	0.51	9.26	49
 ‘W-P (0.21)’	0.62	5.0	65	 ‘FA-C (0.49)’	0.64	38.88	90
 ‘D-C (0.59)’	0.66	8.98	76	 ‘TK-RS (0.38)’	0.72	8.88	165
 ‘W-CS (0.23)’	0.85	2.99	640	 ‘PP-C (0.2)’	0.85	14.06	550
 ‘PP-C (0.19)’	0.89	7.89	466	 ‘G-P (0.38)’	0.9	3.65	564
 ‘W-C (0.25)’	0.94	2.68	884	 ‘W-C (0.53)’	0.94	0.61	280
 ‘FT-CS (0.66)’	0.95	3.29	290	 ‘W-CS (0.35)’	0.97	0.85	363
 ‘KK-MIF (0.39)’	1.0	0.26	650	 ‘W-C (0.27)’	1.0	0.08	713
 ‘LL-MIF (0.58)’	1.0	0.03	2332	 ‘S-P (0.21)’	1.01	0.01	1759
 ‘D-C (0.58)’	1.01	0.0	1614	 ‘W-C (0.25)’	1.01	0.01	2375
 ‘W-RA (0.27)’	1.01	0.0	2150	 ‘W-N (0.19)’	1.01	0.01	2178
 ‘SB-PLP (0.37)’	1.01	0.0	3046	 ‘CF-P (0.2)’	1.01	0.02	3509
 ‘TK-RS (0.53)’	1.01	0.0	2495	 ‘W-CP (0.37)’	1.01	0.01	1271

dyed images, and checkered designs. These metrics help in understanding the feasibility, limitations, and design considerations for a through-fabric display. It is important to acknowledge that not all fabrics will work, thicker and darker fabrics generally have lower transmission levels. Overall, these results show that many types of garment fabrics transmit enough light generated by an LED matrix to be visible for a user.

Visual Separability — Visual separability of the light pattern transmitted through the fabric is another factor that affects the usefulness of through-fabric displays. This measure would capture how well people could distinguish individual pixels in different patterns, which is likely affected by different types of weaves, interaction with fabric patterns and material blends, and adherence of the fabric to the LED matrix. For example, the high contrast, high-frequency floral pattern of the Poplin Print fabric sample 'PP-C (0.2)' affects separability because the LED pattern visually interacts with the fabric pattern, causing some LED pixels to appear to merge, creating "bridging" patterns (Figure 3.5a).

Separability is affected by the distance from the LEDs to the fabric. When the LEDs are not tight against the fabric, the resulting gap increases diffusion making the through-fabric display blurrier (examples in Figure 3.5b,c). This effect is most prominent in thicker fabrics. This is not a pronounced problem with more tailored or form-fitting clothing or when a device in a pocket naturally lays against the pocket fabric. To mitigate this issue, an internal clip or magnet can hold an LED through-fabric display tightly against the inside of the pocket fabric.

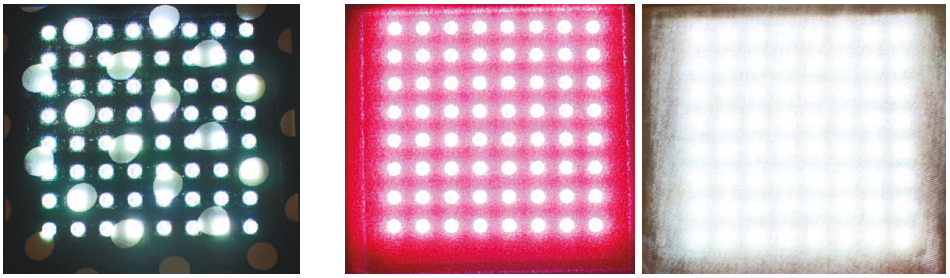
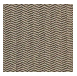
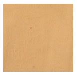
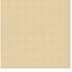

(a) (b) (c)

Figure 3.5: Visual separability examples: (a) bridging patterns in a polka dot fabric 'PP-C (0.2)'; diffusion effect when fabric is placed 2mm above the LED matrix for (b) thin fabric 'G-P (0.38)'; (b) thick fabric 'W-C (0.53)'.

We tested several possible objective quantitative measures for separability, but were not able to find one that was repeatable and represented the subjective experience of a person interpreting a through fabric display pattern. We note that many contemporary clothing fabrics have little or no high contrast patterns, so in practice this may not be a common issue.

Multilayered Fabrics — Another consideration is that light transmission may be affected by multiple fabric layers. For example, pockets are often lined with a thin cotton material like 'W-C (0.25)'. Although we did not test fabric

Table 3.3: Transmittance for other display sources (results from Table 3.2 for the "bright 8×8 LED matrix" used in main experiment provided for comparison).

Fabric (thickness mm)	**Phone**	**8×8 matrix**	**15×7 matrix**	**from Table 3.2**
 'D-C (0.88)'	0.0	0.02	0.03	0.14
 'PL-PR (0.44)'	0.0	0.14	0.22	0.51
 'FT-CS (0.66)'	0.01	0.43	0.6	0.95
 'W-C (0.53)'	0.02	0.43	0.68	0.94
 'W-C (0.25)'	0.1	0.47	0.71	0.94
 'S-P (0.21)'	0.16	0.93	1.01	1.01

in layers, given the high transmittance of this type of fabric, we believe it will have little effect on light transmission when used as an inner lining. Some garments use multiple layers of thick fabric, such as a formal suit jacket or winter jacket. We plan to test these more extreme examples in the future, but note that even with these garments, there are typically some external pockets that have a single or minimal layers of fabric, for example, a shirt or a hoodie pocket.

Other LED Matrices — It is also informative to compare these results with other types of through-fabric displays. We measured transmittance for a standard phone (Google Pixel 2, P-OLED display) displaying high contrast pixels at maximum brightness, a smaller 1.2×1.2 inch 8×8 LED matrix (Adafruit 1614), and a 2×0.9 inch Charlieplex Feather Wing 15×7 LED matrix (Adafruit 3163). We calculated light transmission similar to the main experiment on a sub-sample of six fabrics chosen to cover a range of transmittance with the high power LED matrix. The results are shown in Table 3.3. The phone screen image is visible through some fabrics, but transmittance is much lower and becomes too low to be visible with thicker fabrics. The 15×7 matrix has slightly better transmittance than the 8×8 matrix. The bright LED matrix used in the main experiment has very high transmission values compared to all the matrices and phone, thus making it suitable to design through-fabric displays that can work on a wider range of fabrics. These results validate the ability of other variations of LED matrices to shine through fabrics, while a standard phone can only work through thin fabrics.

3.6 POCKETVIEW DEVICE PROTOTYPE

Motivated by the surveys and technical evaluation results, we created a hardware and system design with a simple interaction vocabulary for a through-fabric display device suitable for a pocket. We use available electronic components to create our novel device. This initial prototype is used in the user study that follows, after which variations on this first prototype are presented to demonstrate additional form factors and interaction design variations.

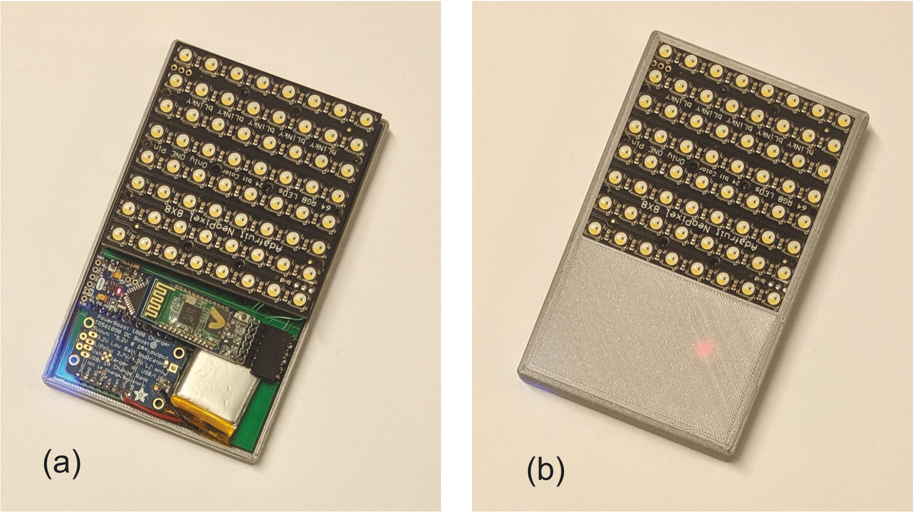


Figure 3.6: Device prototype: (a) self contained battery-powered, wireless device with 8×8 LED display; (b) as assembled in a 3D printed case with a phone-sized form factor.

3.6.1 *Hardware and System*

A RGBW Neopixel 8x8 display (Adafruit 2872) is mounted on a custom PCB and controlled by an Arduino promini micro-controller (Figure 3.6a). The prototype board measures 115×71×15mm and can be enclosed inside a 121×77×18mm 3D printed case (Figure 3.6b). A HC-05 bluetooth module communicates with the smartphone to receive content to be displayed on the LED matrix. The entire system is powered using a 3.7V, 420mAh Lithium battery and can be recharged using a USB power supply. Each LED (SK6812) has a maximum current rating of 60 mA. As an approximate estimate of run time, we considered typical usage with occasional notifications and temporary information. We model this power consumption as half the LEDs illuminated for 5 seconds every 5 minutes, and calculate the prototype would run for 2 hours.

The form factor of this prototype can represent a through-fabric display housed in a custom phone case, or as a stand-alone device carried in the front pocket resembling a wallet (with the phone placed in a back pocket or bag). Our single-sided prototypes must be inserted into a pocket with the LED matrix facing out to work as a display. This also provides an explicit way to silence or hide a through-fabric display by simply changing the orientation. The Android app sends a bit stream required to display appropriate imagery on the matrix display. This would enable an Android app to sync with other apps like health, email, and calendar to display appropriate through-fabric content.

3.6.2 *Interaction Vocabulary and Applications*

We designed simple graphic icons to convey information related to weather conditions, arrows for navigation directions, and various types of notifications, like a message or a reminder (Figure 3.7). A set of numerals in a similar graphic style is used for quantitative information like calories burnt, time left before the next meeting, and fitness tracking. Low-resolution icons are displayed on the standard phone and LED case prototype. Interaction uses single taps on the pocket [50, 105], to cycle through different information sources (like weather to navigation to fitness and back to weather). Double taps dismiss notifications after they arrive, or turn off the display. The tap gestures are intended to provide simple, quick (and ideally subtle) interaction while viewing the information displayed through the pocket. For simplicity, our initial prototype is placed over the phone like a phone case, and the built-in microphone of the smartphone is used to detect single and double taps.

Figure 3.7: Interaction vocabulary imagery demonstrated with different kinds of clothing and pockets: (a) numeral 5 through cotton pants; (b) fitness icon through knit dress; (c) mail notification icon through hoodie; (d) message notification icon through front pocket of lycra tights.

3.7 USER STUDY

The goal of this qualitative user study is to test our initial through-fabric prototype in a simulated usage setting to validate the general approach of the hardware, interaction design, and potential usage scenarios. For a relative comparison, we include two baselines.

3.7.1 *Baselines*

The baselines serve as extremes in through-fabric device approaches.

Standard Phone — This baseline approach uses a standard phone display to shine information through fabric (Figure 3.8a). The screen is set to maximum brightness and uses high-contrast 8×8 pixelated white-on-black imagery approximating the fidelity of the LCD matrix display. This approach is simple and immediately applicable, but limited to shining through light coloured, thin fabrics, in low ambient light conditions. The built-in phone sensors are used to detect single and double taps for interaction.

PDLC Transparent Pocket — This baseline is a radical approach which imagines future fabrics that can dynamically change from opaque to transparent. The intention is to provide participants with a device example that could enable "perfect" through-fabric viewing. The device is a "window" of Polymer Dispersed Liquid Crystal (PDLC) film over a phone-sized hole cut out of a front pant-pocket (Figure 3.8b). This film can switch between opaque and transparent states by controlling the current passed through the film. Otherwise, the condition is the same as the phone baseline.

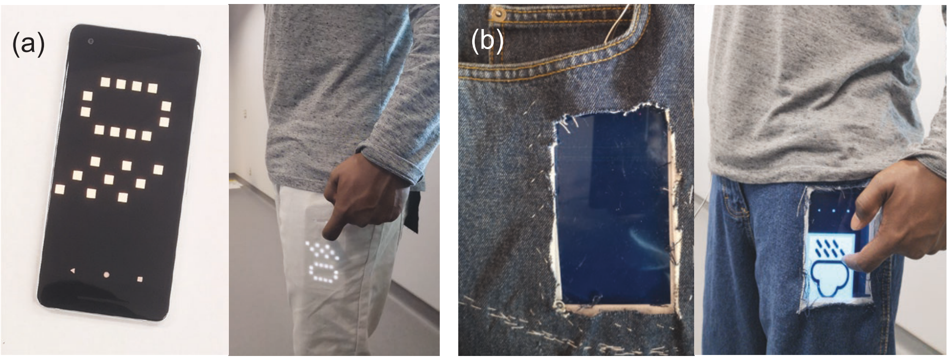


Figure 3.8: User study baseline device conditions: (a) a standard phone with high contrast pixelated imagery; (b) a futuristic PDLC transparent pocket to make a standard phone display completely visible "through fabric".

3.7.2 *Protocol*

We recruited 12 participants ages 22 to 31 (1 female, 11 male) from a university student population. Based on a short questionnaire, 10 stored their phone in a pant pocket, the others used a backpack and coat-pocket. With one female participant, this study is limited in terms of generalizing to women.

During the session, the participant used all three through-fabric device conditions, one at a time: the standard phone baseline; the PocketView LED phone case prototype device; and the PDLC transparent pocket. They were provided with light, white-coloured pants for the first two prototypes, and blue jeans fitted with the PDLC film as the third prototype. Most chose to wear the supplied loose-fitting pants over their existing clothes. The experimenter used a desktop application to trigger notification events on the smartphone and LED matrix display. A custom Android app running on the smartphone, received these commands from the experimenter's application, and rendered the corresponding icon to the screen, or interfaced with the micro-controller to render it on the LED matrix.

While wearing each prototype, the participant was asked to stand, sit in a chair, and sit on a bicycle. They then used the prototype interaction vocabulary to view different information sources with single taps, and the experimenter sent notification alerts at random times, which the participant dismissed with a double tap. During this time, they were prompted to "think out loud" to externalize their thoughts and experiences for observation [93]. After trying all three device conditions, they ranked each for visibility, comfort, usefulness, and ease of interaction. They also provided an overall preference for each device using a 5-point numeric scale. After, a semi-structured interview was conducted. Interviews were conducted following best practices [134], and all but two were audio recorded (due to a technical error). Each session lasted approximately 30 minutes.

3.7.3 *Results*

In terms of overall preference, 91.7% assigned scores of 4 or higher for the LED matrix and 83.4% for the PDLC transparent pocket. Meanwhile only 66.7% assigned a score of 4 or higher to the standard phone baseline. Participants ranked all three prototypes similarly in terms of ease-of-interaction. From rankings, think-aloud observations, and interviews, there were six themes that emerged.

Phone Visibility — The standard phone baseline was ranked the lowest on visibility. Participants expressed skepticism on its utility outdoors, *"not sure how usable it is in sunlight"* [P12]. Meanwhile, most users preferred the LED prototype owing to its high visibility, though one participant thought it might be *"too loud"* [P12]. Another participant wondered whether a *"[standard phone and LED phone case] would not work with absolutely all types of fabric"* [P1]. We expected participants to comment on the viewing angle when standing, sitting, or when on the bicycle, but no one specifically commented on viewing angle as an issue.

Use in Different Scenarios — 7 participants indicated that they often need to access information on their phone while their hands are occupied. While walking with hands encumbered, 6 participants preferred the PDLC pocket and 5 preferred the LED case. In meeting scenarios, 9 participants preferred having a display so that they would not miss out on importation notifications while having their phones on silent, for example *"a visual indication would be better in environments where phone has to be kept on silent"* [P5]. One participant also mentioned the general convenience of being able to view through the pocket, *"sometimes it's difficult to take out the phone when you're sitting and so this can be useful even when my hands are free"* [P3].

Use for Different Tasks — Regardless of though-fabric device, participants imagined several tasks such as controlling music, reading messages, or navigating using Maps being done directly from the pocket. *"nice to have phone in the pocket while running"* [P7]. *"Having Maps here is the most interesting feature"* [P11]. One participant said that *"even though it divides my attention but it would be really useful if I can interact with the phone on the bike and answer calls"* [P2]. Another wished to *"have special pockets like this for the gym"* [P8] where they could work out without having to take the phone out of their pockets.

Less Reliance on Third Party Devices — Participants commented on reducing the reliance on third-party devices for accessing information. One participant mentioned that *"headphones do it somewhat but [controlling music] is better if you can do it directly from your phone"* [P1]. Participants also commented on how all approaches obviate the need for information to be synced, *"I can use it [PDLC] with any kind of phone without worrying about iOS, Android compatibility or Bluetooth syncing"* [P11]. However, in practice, this only really applies to a standard phone. A transparent pocket technology like PDLC would need a connection to the phone to synchronize transparency with display events on the phone. The same is true for the PocketView device, it needs a wireless connection with the phone to receive image rendering patterns. In both cases, these connections only need to support real-time output events, not synchronization of data stores which adds additional considerations for security and privacy.

Managing Privacy — Several participants raised concerns related to privacy, for example *"I would not use this if it showed too much information in public"* [P3]. In particular, some feared the PDLC might accidentally become transparent and show too much information: *"I would not like to use a transparent pocket in a social setting"* [P5]; and *"I would not use this [PDLC] when someone is walking towards me, for example, my prof and I'm getting a lot of messages."* [P11]

Fashion and Aesthetics — While people did not complain about the aesthetics of the LED matrix prototype, they did not find the PDLC prototype to be visually pleasing. One participant said that the *"only issue is how it looks and feels"* [P4], and several participants mentioned that being fashionable is very important.

3.7.4 *Discussion*

Overall people preferred the LED prototype in terms of managing privacy and fashion aesthetics. While the PDLC approach was able to provide more

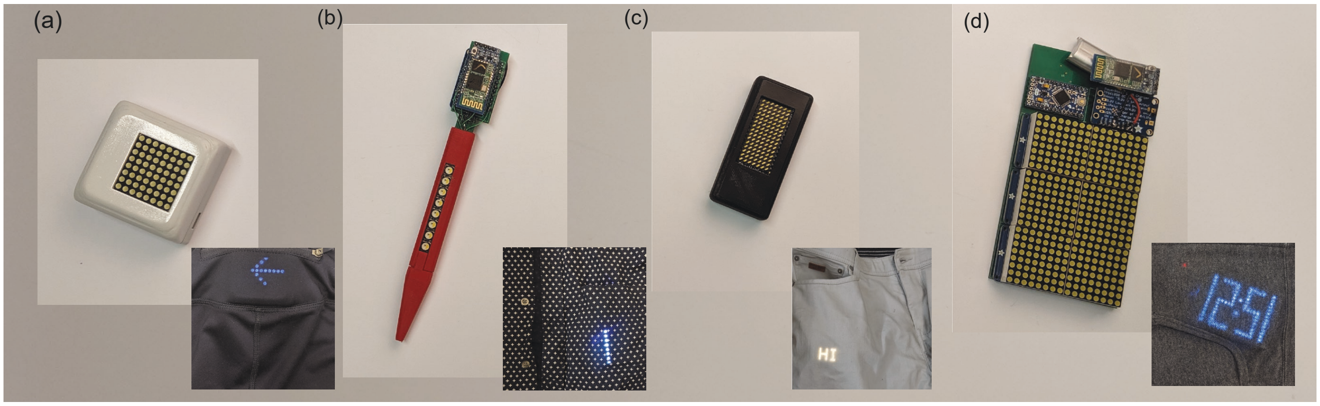


Figure 3.9: Prototypes showing form factor variations: (a) earbud headphone case; (b) pen; (c) car remote; (d) phone case.

information, some users were comfortable with the minimal information provided with the LED device, given that it allowed multitasking and reduced the reliability on other third party devices like a smartwatch. There was also a positive reception to the "through-fabric" aspects of the PDLC. We interpret this as a validation of the general concept of a through-fabric display. This prototype has clear practical limitations: PDLC does not feel or flex like fabric given its stiff and plastic properties, and even full opacity is quite transmissive compared to a fabric like denim. However, as one baseline in our study, it was effective for helping participants make a relative comparison between a "perfect" through-fabric pocket display in terms of transparency and image fidelity.

3.8 DESIGN VARIATIONS AND LIMITATIONS

In this section, we present other device form factor variations and discuss limitations and considerations for the general approach.

3.8.1 *Device Variations*

To demonstrate how the PocketView through-fabric display concept can be extended, we built different prototypes resembling other items that might be stored in a pocket. Like our initial prototype, all but the car remote prototypes are self-contained with a microcontroller (Arduino Pro mini), Bluetooth chip (HC-05), lithium-ion battery, and Powerboost 1000C (Adafruit 2465) module for boosting the battery voltage and USB battery charging. The car remote prototype has all the components except a Bluetooth chip. These form factors demonstrate different use cases of through-fabric devices, and they are informed by the results of our main survey showing the diversity of garment pockets and what kinds of objects are placed in pockets.

Earbud Case — We built a prototype resembling an earbuds case (Figure 3.9a). It would be small enough to fit in many different pockets, most notably the front pocket of women's jeans. It contains a small 1.2 × 1.2 inch 8 × 8 square LED matrix interfaced to a driver circuit (Adafruit 1614) all enclosed in a 62 × 60 × 28mm 3D printed case with rounded corners. It is powered using a 250mAh lithium-ion battery.

Pen — We also explored a small prototype with a restricted display in the form factor of a pen (Figure 3.9b). A linear 8 × 1 LED strip (Adafruit 2869) is mounted on a custom PCB. The LED strip and on-board circuitry are powered from a 110mAh battery. The LED strip is enclosed in a 3D-printed case resembling a pen, which measures 121 × 14 × 11mm. Most electronics remain external to the case which simplified this demonstration prototype development. The low-resolution one-dimensional display necessitates a simplified version of the interaction vocabulary. Numeric values, such as fitness counters or meeting timers, can be shown as a bar along the strip. Different notification types can be conveyed using patterns and animations.

Car Remote — A car remote (or "car key fob") is another convenient form factor for a PocketView device (Figure 3.9c). Our prototype uses a Charlieplex Feather Wing 15 × 7 LED matrix (Adafruit 3163). It measures 76 × 34 × 17mm and its small form factor can also fit into a wide range of pocket sizes.

Phone Case — We also experimented with a higher resolution display prototype in a phone case form factor (Figure 3.9d). It uses six 8 × 8 LED matrices along with a driver board (Adafruit 2308) tiled together to form a 24 × 16 through-fabric display. It uses the same LED matrix as the earbuds case. All the components are mounted on a custom 138 × 74 × 17 mm PCB. This prototype can display information like scrolling text, for example, a grocery shopping list or more details about a specific notification, like an email or text.

3.8.2 *Applications*

Even with a simple interaction vocabulary and low-resolution display, PocketView through-fabric displays can show notifications, reminders, track the progress of an activity, or act as social displays. Different form factors will be suitable for different pocket locations, sizes, and usage scenarios.

Notification Assistant — When inserted in a front pant pocket, navigation instructions can be shown while walking or biking, or it can act as signal indicators when inserted in a back pocket while cycling [32]. Users can also manually tap on the system to view weather updates or time before the next meeting when their phone is inaccessible to retrieve.

Fitness Tracking — Prototypes placed in a pocket of athletic wear can show fitness statistics like step count, calories burned, heart rate, or track fitness goals while jogging, walking, or working out. Pen prototypes can visualize progress towards a goal as a bar plot, or the higher-res phone case prototype can show the fitness stats with more detail.

Social Displays — A prototype placed in a back pocket, for example, a sports bra back pocket, could function as social or public display [81]. The wallet prototype placed in the pant back pocket can display a social message or trigger emergency medical notification to notify the public of a potential medical emergency. These displays can act as digital ID cards at conferences when placed inside a neck wallet pocket.

3.8.3 *Limitations and Design Considerations*

We discuss current device implementation limitations and privacy implications for a personal through-fabric display.

Power consumption — The LED matrix display in the original prototype can consume up to 20 watts with all pixels illuminated. In practice, the display would be used for short periods to convey information at certain moments, and it can be made to run at reduced power consumption with a proportional reduction in brightness. For example, by intelligently reducing brightness based on the ambient lighting conditions and fabric transmission properties.

Prototype Size — Although we attempted to make the prototypes small, they remain slightly bulky because they are built using commercially available components. With more engineering, they can be made lighter and sleeker to more closely resemble different items, or even integrate with those items. For example, Apple iPhone "Magsafe" is a magnetic accessory attachment method with power and communication that could support a PocketView through-fabric display on a phone.

Privacy — We also note the privacy aspects of a through-fabric device like PocketView. Unlike third-party devices like smart watches or voice assistants, our device has no capacity to store, share, or analyze any data. While the minimal information that is displayed can be seen by other people, it can be configured to convey no more than what a glowing phone or smartwatch notification would show. We can also imagine users might create custom obfuscated imagery that are uninterpretable by others.

Display Location — The location of a wearable display affects visual accessibility, interaction subtlety, and social acceptance. Harrison et al. [41] studied reaction times to visual notifications generated by LED nodes placed on different parts of the body. Wrist and shoe locations had the fastest and slowest reaction times respectively as participants predominantly spent time in a seated position. In another study, Harrison et al. [40] examined suitable locations to project content for on-body interfaces when standing or sitting. Arm and hands were most suitable, but notably, the thigh area received positive feedback for a seated posture. Some areas of the body are not socially acceptable for displaying content, and these positions can vary by gender. For example, women may be less comfortable with a display placed on the chest than men. Body shape influences visual accessibility. For example, people with a larger hip size may have more difficulty viewing content displayed on the lower body. Our PocketView prototypes are suitable for diverse wearable display locations, which may alleviate and compensate for the guidelines and issues above. Future studies can examine suitable locations and social acceptability.

Challenges with Cold Weather Outerwear — Our results show a general trend of lower light transmission with thicker fabric. This poses a limitation for using a through-fabric display in the pocket of insulated clothing like winter parkas. In some cases, these garments have thin-walled outer pockets that are sewn on the outer wall of the jacket, which could be used.

3.9 CONCLUSION

We investigated how to create an unencumbered, always-accessible display for smartphone content through a pocket, a concept we call through-fabric displays An online survey explored different pocket locations in garments, the items stored in them, and the need to access information when the phone is inaccessible. To explore the feasibility of through-fabric displays, we performed a technical experiment to validate the ability of an LED matrix to shine through common garment fabrics. Motivated by these results, we built a preliminary prototype for a through-fabric display using an 8×8 RGBW LED matrix in a phone-sized form factor. Then, a qualitative study conducted with 12 participants suggested that the approach can be useful, and the general device form factor is reasonable. Finally, we showed that these ideas can be generalized to other items typically stored in a pocket, like a pen, headphone earbud case, and car remote. Beyond creating a new type of wearable, our through-fabric devices could be used for prototyping smart textile interactions where the ultimate goal is to embed or weave a display into fabric.

We hope our work opens up a new space for designing interactions with smart devices without having to remove them from their stored location.

4

SCATTERPIXELS : AD HOC RECONFIGURABLE PHYSICAL PIXEL DISPLAYS

4.1 INTRODUCTION

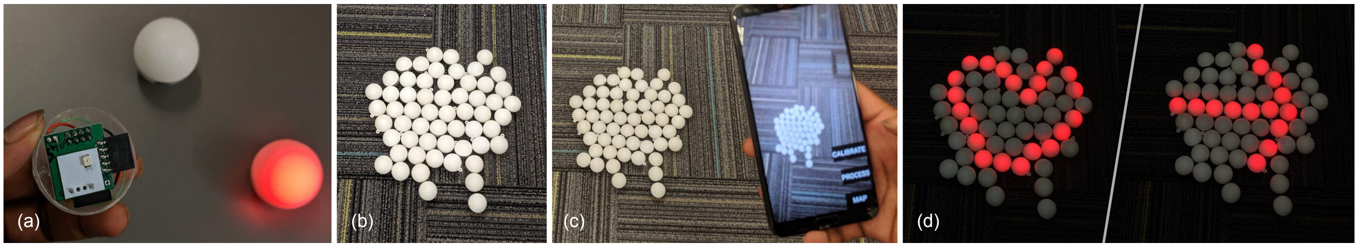


Figure 4.1: Scatterpixel system: (a) each 4cm spherical "pixel" is an independently addressable unit with a rechargeable battery, microprocessor, and radio to control a red LED; (b) one example usage where pixels create an ad hoc floor display; (c) the relative spatial locations of the pixels are registered through computer vision using a smartphone application; (d) once registered, rendering algorithms display patterns or symbols at optimal positions using a communication protocol capable of 20 FPS animation.

Traditional computer displays are a matrix of small, precisely aligned pixels that maximize fidelity within a standardized, defined area. We explore an alternative visual display concept called *ad hoc reconfigurable displays* where pixels are individual physical entities that can be positioned and combined to create re-usable displays with arbitrary shapes, sizes, and functions. The goal is to create a portable and easy-to-use system that trades the fidelity of conventional displays for a high level of flexibility in display configuration, enabling new kinds of ubiquitous display use cases and novel aesthetic display experiences. For example, the same physical pixels can be reconfigured to form a long corridor display to nudge people to a meeting room in the morning, attached to a wall to show the score of a game of basketball in the afternoon, and scattered on a lawn to create a fun welcome sign for a party in the evening.

Previous work has explored different approaches to enabling related types of reconfigurable displays. For example, using many mobile phones or tablets held by people in a crowd where each device acts as a pixel in a large temporary display [110, 115, 140]. A compelling concept, but tailored to a specific setting and use case. Miniature robots or drones have been created to arrange themselves as pixels in dynamic displays (e.g. [30, 69, 118]), but they are complex, expensive, have high power requirements, and require an instrumented environment for continuous tracking. They are well suited to applications that benefit from the capabilities of a realtime dynamically reconfigurable display.

We focus on a different problem space for temporary *manually reconfigurable displays*, and target use cases and applications that are impractical or undesirable to create with a swarm of robots. For example, a one-dimensional wayfinding display created along a long corridor during an afternoon of meetings, or a sign for a party placed on a grassy lawn for an evening. The most related approaches to our work are systems that use addressable LEDs as pixels [11, 26, 47, 108, 110, 116, 143]. But these systems remain incomplete: some use approaches that rely on hard wiring for either communication or power; some use communication protocols not capable of real-time animations; and many are limited in the method and versatility for how pixels are spatially registered. *No previous work has holistically examined all technical aspects necessary to realize the potential of using individual LED pixels for reconfigurable displays.*

This paper describes Scatterpixels, a comprehensive, and flexible, and reconfigurable ubiquitous display system. The core of the system is a 4 cm spherical "physical pixel" using a single red LED (Figure 5.1a) that is inexpensive, wireless, rechargeable, and scalable. Many of these pixels can temporarily be distributed along a corridor, attached to a ferromagnetic surface, fastened to a wall or window, or spread out on the floor or ground (Figure 5.1b). Smartphone-based computer vision methods register the positions of addressable pixels in arbitrary arrangements (Figure 5.1c), including providing interactive augmented reality guidance so the operator can fine-tune pixel positions. Once registered, algorithms optimally render content on the available pixel layout (Figure 5.1d,e). A compact and fast communication protocol enables animation frame rates on the 70 pixels we fabricated, and will scale to hundreds of pixels. Using our system, we demonstrate different display configurations and applications, ranging from individual visual indicators, to one-dimensional way-finding, to two-dimensional dynamic event signage.

In summary, we contribute a complete end-to-end reconfigurable display system using faster, smaller, more self-contained hardware, more comprehensive registration methods enabling a greater variety of display configurations, new methods to fine tune pixel layouts for specific content, and integrated techniques that optimize content for a given layout. Our methods are packaged in a way to make the complete system usable: simple deployment requiring only a base station, phone, and remote server; a convenient charging method; and a phone-based app to control all functions. Our open source schematics, design files, and code are all available[1].

4.2 HARDWARE AND SYSTEM DESIGN

This section provides the hardware details for a "physical pixel," as well as associated parts of the Scatterpixels system for pixel charging and communication.

---

[1] https://github.com/exii-uw/scatterpixels

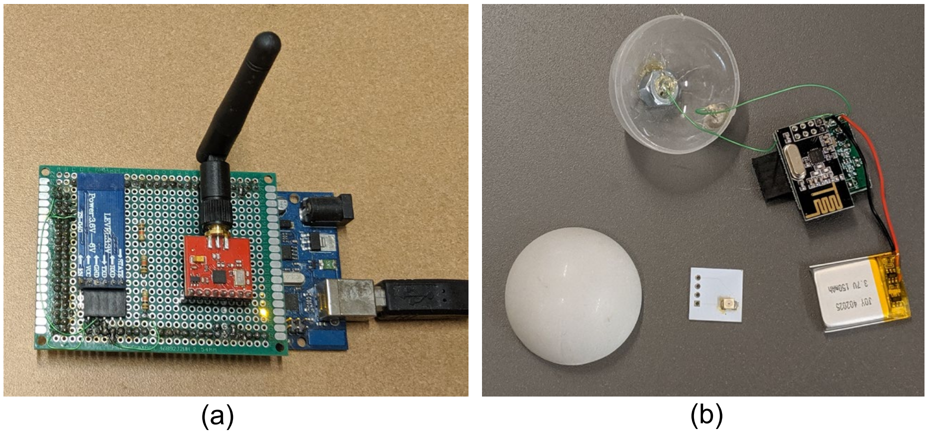


Figure 4.2: Hardware setup: (a) Bluetooth base station; (b) pixel disassembled into two halves showing primary components.

4.2.1 *Physical Pixel*

Each pixel (Figure 4.2b) is a 4cm diameter plastic sphere, which can opened into two hemispheres. The bottom half contains all components: a 25 by 20 mm 150mAh Lithium-ion rechargeable battery; a 23 by 24 mm custom PCB with a microcontroller (Atmega328), 3.3v voltage regulator (ADP122ACPZ), and other small components; a 2.4 GHz wireless transceiver board with a PCB trace antenna (NRF24L01+); and a 15 by 15 mm custom PCB board to mount a LED. The red LED is a 2.8 × 3 mm SMD (VLMS334AABB-GS08), with 1600 mcd luminous intensity and ±60 angle of half intensity). Two metal bulletin board thumb tacks, with flattened heads, are mounted through the bottom and side of the lower hemisphere to create battery charging contact pins.

The LED board is white to maximize reflection and designed to position the LED in the lower centre of the top hemisphere when the pixel is assembled. The top hemisphere is left empty and the inside is coated with a thin film of white spray paint to evenly diffuse the LED light. The outsides of both hemispheres are coated with super matte transparent spray paint to eliminate specular reflections that would otherwise cause issues when computer vision methods are used for registration. The spherical shape of the pixel allows it to roll on surfaces and encourages more creative, less precise display layouts. A standard 1/4 inch hardware nut is placed inside the lower hemisphere at the very bottom to act as a ballast to further lower the centre of mass of the pixel. Each pixel weights approximately 17 g. This weight distribution, and slightly flat bottom profile created by the bottom charging pin, means when the pixel is placed or rolled on the floor, it eventually rights itself so that the LED diffuser half is up. The hardware nut can be replaced with a neodymium ring magnet of similar size, which provides enough attractive force through the bottom charging pin to attach the pixel to ferromagnetic surfaces.

4.2.2 *Charging*

A custom charging station (Figure 4.3a), resembling a large egg tray, charges 25 pixels at once. It is a custom 40 by 31 cm PCB board suspended above a wooden base. Each pixel rests in a 39 mm circular hole which has an exposed conductive rim connected to charging circuitry. When in the hole, the pin on the side of the pixel makes contact with the rim, and the pin at the bottom of the pixel contacts a ground plane. The ground plane is fabricated from copper tape bonded to thin plastic film, which is attached to the charging station wooden base in a way that sections of the strip form very light springs (Figure 4.3b). Wood spacers suspend the charging board 16mm above the base. The charging circuitry for each hole is a MCP73831T-2ACI Lipo charge controller chip with an LED to monitor charging status.

A pixel begins charging as soon as it is inserted into the circular opening, and it takes approximately 90 minutes to charge. After charging, a pixel will run continuously for 5 to 8 hours, depending primarily on LED illumination time. Note that pixels have no "on and off" switch: as soon as the firmware has power, the radio runs in receive mode and processes data from the base station. This makes the system simple to deploy, but further optimizations like a "sleep mode" could drastically increase stand-by time for charged pixels. We created four of these charging stations, so in practice, all pixels can be left charging until needed.

4.2.3 *Base Station*

The system uses a single base station (Figure 4.2a), which is an ArduinoMega microcontroller, bluetooth module HC-05 and a 2.4 GHz NRF24L01+ wireless transceiver board (https://www.sparkfun.com/products/705) operating as a transmitter with an SMA connector. An external LCW Dipole high-gain antennae is connected to the SMA port. The base station connects to a standard smartphone through the HC-05 Bluetooth radio. The typical workflow in sending data to the pixels involves creating the data packet on the phone app, which is sent to the base station through bluetooth. The Arduino Mega decodes the received packets and sends appropriate signals to all physical pixels using the NRF24L01+ radio. The base station is programmed with calibration routines used for pixel registration. In informal tests, we found the base station could communicate with pixels more than 7 m away without major obstructions like walls.

4.2.4 *Firmware and Communication Protocol*

During assembly, custom firmware is loaded into each pixel microcontroller using an USB-UART programmer through a 5-pin header on the pixel PCB. The program assigns a unique id, operates the radio in receive mode to continuously listen for data packets from the base station, and when data is received, decodes the data and performs the required action. The firmware also sets the radio power amplifier to LOW level (to reduce power consumption), sets the radio channel, and sets the air data rate to 250 Kbps. For a display to show meaningful content, all the pixels of the display should update synchronously. We did not use a standard mesh network protocol because it introduces a dependency on certain pixels and increases latency as data traverses through the network.

Instead, our method is multicast, where all pixels receive the same data packet from the base station at the same time. This means all pixels can be updated simultaneously, which avoids latency during registration or when displaying content across many pixels forming a single display. In addition, there is no dependency on any pixel for communication, so the system is robust if a pixel battery drains before others.

The communication protocol uses a single 26-byte data packet. The first byte encodes one of three commands, and the remaining 25 bytes encodes which pixels must execute the command. The commands are: `activate`, to signal a pixel to illuminate the LED; `register`, to run a pre-determined registration routine of flashing the LEDs at 33Hz for two cycles; and `flash`, to flash the LED in a binary sequence representing the pixel's unique ID. Since the same data packet is broadcast to all pixels, multiple pixels can execute the same command and which pixels should execute is determined by a bit mask. Our current implementation supports 100 pixels, where the position of each bit in a 100-bit binary string indicates whether the corresponding pixel id should execute the command ('1') or not ('0'). For convenience when decoding the packet on the microcontroller, the 100-bit binary string is encoded in a 25 byte hexadecimal string.

Consider a simple example. If a pixel with id '7' receives a packet with first byte indicating `activate` and decodes the remaining bytes to determine the 7th bit is '1', then it will illuminate its LED. Correspondingly, if a pixel with id '11' receives the exact same packet with `activate` in the first byte, but decodes the remaining bytes to find the 11th bit is '0', it will turn off.

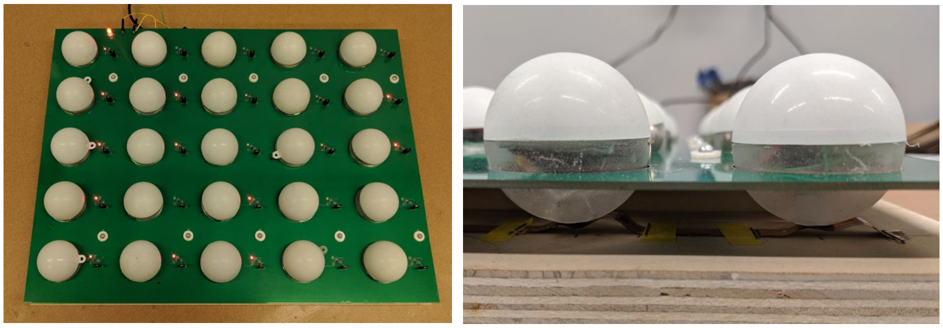

(a) (b)

Figure 4.3: Pixel charging station: (a) 25 pixel charging board; (b) detail showing pixel contact with ground plane through bottom push pin and with charging circuit through conductive rim and side push pin.

Our system can update up to 100 pixels at 20 Hz from the phone, the limiting factor for the refresh rate is Bluetooth communication between the phone and base station. Using our simple broadcast protocol, the base station and pixel hardware is capable of 100Hz updates. For applications that do not require interactive phone input, a one-time configuration packet could be sent to the base station to to control a sequence of pixel updates at 100 Hz.

4.3 SPATIAL LOCATION REGISTRATION

The physical pixels can be arranged in different configurations, like small and large 2D displays, and long 1D displays. Once arranged, the relative location of each pixel and its ID are required to display content appropriately. In a basic form, registration may be accomplished by holding the phone camera to capture all pixels at once, then recording a video while the pixels to execute a flashing registration sequence. A more generalized form of registration, suitable for large, disconnected, or dispersed displays, is to repeat this capture sequence multiple times by moving the phone camera over the display, pausing when one or more pixels become visible in the frame, and capturing a video of the flashing registration sequence each time. This method reconstructs a single spatial representation of the display in 3D. In essence, the interaction and goal is similar to taking a panorama photo: separate or overlapping portions of the display are captured from different view perspectives, and then "stitched" together to create a single spatial registration for all physical pixels.

In both registration forms, an Android application uses computer vision code to extract the image coordinates and IDs for each pixel, which are combined and stored as a single *spatial frame*. In the generalized form, this is repeated for each capture sequence, with the relative position and orientation of multiple spatial frames determined using the Android ARCore smartphone pose. Combined, a single spatial frame is composed of all the IDs and image coordinates for each physical pixel location captured within the frustum of the smartphone camera along with supplementary pose data obtained from ARCore. For the generalized form of registration, all data is sent to a separate Unity server using GRPC[2] to combine all spatial frames capturing portions of the display into a final spatial registration of the complete display.

4.3.1 *Basic registration using single capture pose*

This form of registration is sufficient for a display that can fit within the frustum of the smartphone's camera from a medium distance. The `register` and `flash` commands allow for two types of pixel identification techniques: *one-shot* and *sequential*.

One-Shot Identification — The approach to extract the image coordinates and IDs for each physical pixel is inspired by Firefly [11]. All visible pixels within

---

[2] www.grpc.io/

the camera frustum are located at once by decoding this pattern, which is encoded as a series of LED flashes.

During video processing, we detect the video frame in which all physical pixels are illuminated as the starting point of the binary sequence. This is converted into a binary image from which brightness and circular contours are used to find a region of interest (ROI) for each physical pixel. A single flash of an LED lasts for exactly 100 ms (approximately 12 video frames). A complete binary sequence can then be reconstructed by analyzing each ROI over time. Finally, the decoded ID and centre of the ROI are saved as a JSON string for further processing and visualization. For a 54 pixel display, the capture time is approximately 2.5 seconds and video processing is less than 1 minute. Capture time remains constant and processing time is linear to the total number of pixels. Our calculations suggest that a 1,000 pixel display would require the same capture time of 2.5 seconds and processing time of about 15 minutes without optimization. Video processing time could be further optimized using GPU acceleration and SIMD instructions on the phone.

Sequential Identification — In challenging lighting conditions and display configurations, the one-shot method can fail, so we provide an alternative more robust method with the trade-off of more time to capture pixel flashes. This is a similar approach to what is used in Particle Display System [108]. When the user initiates capture, the base station broadcasts packets to create an initialization time marker sequence, where all pixels are illuminated for 200 ms, then off for another 200 ms. Then, the base station broadcasts packets to request each pixel, one at a time in ascending order, to execute the `register` routine of their assigned ID. The time window for each pixel to flash is 200ms. When the capture is finished, the video is processed similar to above to locate each pixel, with the advantage that the problem is more constrained. Only one pixel will be flashing at a time, and the temporal order provides a degree of error checking. Video processing is similar to *one-shot*, where binary thresholding and frame subtraction is used on the video frames. As an example of performance, it takes approximately 38 seconds to process the recorded video to obtain all image coordinates and IDs for a 54 pixel display. To get the best performance, the phone should be held stationary.

4.3.2 *Generalized registration with multiple capture poses*

Single-pose registration is sufficient when the physical pixels can be framed within a single camera view that is held roughly parallel to the plane spanning them. However, if they are distributed over a wider area or have non-planar shapes, multiple captures are needed to reconstruct their spatial relationships. The sequence of captures, along with pose information from ARCore, allow us to build a 3D representation of the physical pixels using an incremental optimization technique over the parameter space of each pose and image coordinate for each captured frame. The time to register a large display is proportional to the number of single frame captures needed for reconstruction, where each capture uses the method described in Section 4.3.1.

As a starting point, we frame this as a variation on a structure from motion (SfM) [87] problem found in large scale computer vision tasks, with two added assumptions: 1) the correspondences between physical pixels are known and 2) the poses for each capture is approximately known with noise. This brings the total parameter space for each capture with N detected physical pixels to be equal to $\Phi = 6 + N + 2N$ parameters: 6 parameters representing pose, N parameters representing the distance z for where each physical pixel lies on the ray projected from the camera, and $2N$ parameters that represent the image coordinates for each captured physical pixel. To limit the scope of the parameters space into manageable chunks, we formulated an incremental optimization routine that iterates over each pair of captured frames in three separate phases, where each phases is responsible for updating only a subset of the parameters.

PHASE 1: IMAGE COORDINATE PROJECTION. This initial phases utilizes assumption 1 and 2, that we know correspondences between physical pixels and we know the approximate pose for the captured frame. For each pair of captured frames, we project the recorded image coordinates out into a shared world coordinate space to find the optimal value z for each physical pixel that minimizes the distance between the projected 3D world points from one frame to the other. This results in an initial guess on where each physical pixel is located in the world.

PHASE 2: POSE ADJUSTMENTS. Based on the results of *Phase 1*, we have an initial guess of where the physical pixels are located in a world coordinate frame of reference. In *Phase 2*, we refine the 6 parameters that represent the pose of each captured frame. For each pair of captured frames, we find the optimal pose that minimizes the distances between each pair of corresponding 3D points representing the physical pixel. This results in further refinement of the captured frames' poses and 3D points of the physical pixels.

PHASE 3: IMAGE COORDINATE REFINEMENT. In the final phase, we again project the captured image coordinates into a world frame of reference utilizing the recovered z values from *Phase 1*. We then iterate over each pair of captured frames and directly refine the image coordinates by minimizing the distances between each corresponding 3D point.

All three phases are applied to each pair of captured frames, where K captured frames give $K^2/2$ iterations. Each phase uses a non-linear least square Levenberg-Marquardt [72, 79] algorithm to minimize their cost function. Further refinement can be accomplished through multiple iterative applications of our optimization routine that use parameter regularization to further constrain the dimensionality of the overall parameter space. The final result gives the 3D points for each physical pixel, with an accurate sense of scale and space. We use these final 3D points to create an accurate 2D image of the physical pixels by projecting them back into image space using an orthographic projection that encapsulates the entire physical display.

4.4 MAPPING AND RENDERING IMAGERY

Mapping images onto an ad hoc reconfigurable display is not always straightforward since the pixels can be arranged in an arbitrary fashion. In this section, we describe different methods to display content on the ad hoc display which includes directly controlling the physical pixels from the phone to create animations and mapping existing binary images to it. We also describe our interactive layout assistant that guides the user to optimally place the pixels when content is known beforehand.

Interactive Display and Animation — This mode allows the user to directly control the display pixels using the phone. A phone app shows the spatial map of the display on which the user can directly draw and create animations. As the user draws an image, the input events can be saved and played back as animations. The user can also create individual frames that can be saved and played back at a specified frame rate. The pixels update in real-time to reflect the drawings and animation frames.

Image Rendering — Rendering a given bitmap image onto registered positions of physical pixels, uses a simple proximity mapping. First, the image is downsampled and binarized. Then, given an image position and scale in the physical display, the closest physical pixel to each image pixel is determined. For manual control, the phone app enables the user to position and scale each image in a set with a live preview on the display. These adjustments are saved, and used to display each of the images at the configured locations and scales to create the dynamic display.

Optimal Image Mapping — Rather than manual positioning, an optimal location can be found. Given an image, a stochastic algorithm iteratively places the image at different positions within the display until it finds an optimal position. Optimal corresponds to minimal error computed as the sum of Euclidean distances between the image pixels and its corresponding nearest display pixel.

4.4.1 *Interactive Layout Assistant*

We also created an interactive layout assistant that guides a user in placing the pixels at optimal locations, when the set of images to be shown on the display is known beforehand. The pixels are initially spread out in some arbitrary configuration and a quick pixel registration is performed to obtain the initial layout of the pixels. The user chooses a set of images from an image gallery and sends them Python server, which generates a layout for the display. A python server binarizes all the received images, and a resultant binary image is obtained by performing a boolean OR with all the binary images. The resultant high-resolution binary image is downsampled using k-means clustering, by setting the cluster size equal to the number of physical

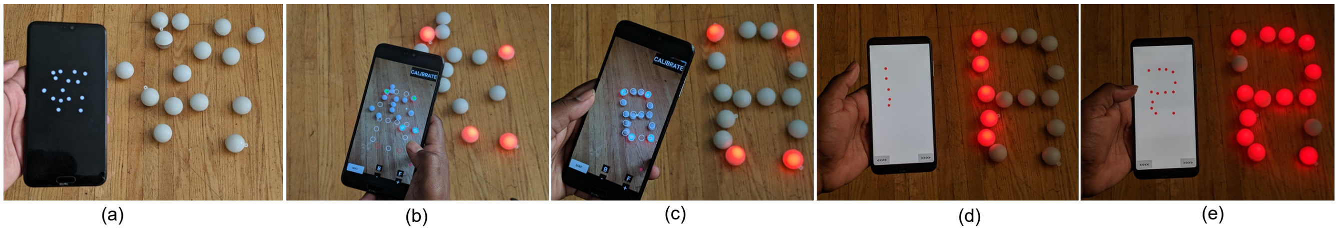


Figure 4.4: Interactive layout assistant: (a) initial layout with pixel arranged in random configuration; (b) guide overlay with red circles showing where to place illuminated anchor pixels; (c) all the pixels are arranged within the white guidelines; (d,e) examples of imagery.

pixels present in the initial layout. The output cluster centres gives the optimal locations to place pixels. The suggested pixel location values are scaled to the camera preview frame size and sent to the phone.

The pixel locations obtained from the server are overlaid on the camera preview as circular guides to assist the user in arranging the pixels according to the suggested display layout. The layout assistant automatically turns on four pixels acting as anchor pixels, which should be placed within the red circular guide overlaid (Figure 4.4 b). Once the anchor pixels have been placed in the appropriate locations, the remaining pixels are placed within the white circular guides (Figure 4.4 c). The white circular guides are always displayed with respect to the anchor pixels, such that even if the phone moves around, they are re-positioned with respect to the anchor pixels. This is achieved by continuously tracking the anchors pixels in real-time and computing the homography with respect to its initial position. The new position of the white circular guides are obtained by warping with the computed homography. The user can add or reduce the number of pixels in the display, and the layout assistant generates a new layout for it. Once the pixel locations are fine-tuned by placing them within the white circular guides (Figure 4.4 c), a quick pixel registration is performed to obtain a spatial map. Now, the display can cycle through the dictionary of images (Figure 4.4 d,e).

4.5 APPLICATIONS USING DIFFERENT CONFIGURATIONS

This section describes a broad collection of possible real world applications for the Scatterpixels system using different display geometries and environment locations. The emphasis is on non-permanent impromptu installations that might be typically created on a smaller scale by non-professionals. The display applications are typically designed to exist for a short time within the five to eight-hour battery life of the pixels, such as an hour up to an afternoon or evening.

Our goal in presenting many different applications is to demonstrate the versatility of the system, and show it can easily be reconfigured into different display configurations with minimal setup time. Note the same set of pixels is reconfigured to form all applications, so these demonstrations serve as a simple validation of system reconfigurability in terms of pixel density,


(a) (b) (c) (d) (e)

Figure 4.5: Applications: (a) 1D guidance display (b) 2D floor display showing 'HI' (c) 2D floor queue displays ; (d) 2D White board timer display; and (e) 2D Ceiling displays

shapes, sizes, and locations. We note that the Scatterpixels system was designed with a vision of enabling display applications using hundreds of pixels. As a research proof-of-concept system, we have a limited number of physical pixels to implement and illustrate the examples below. However, the system and application concept can be expanded to much larger scales. An accompanying video shows a large subset of these demonstrations, including how the implemented display changes over time or animates.

4.5.1 *Individual Pixel Displays*

The simplest configuration is using each pixel as single bit visual indicator to convey a state or status which changes over time. Each pixel can be individually controlled by ID, and could optionally be associated with a specific person or object.

Facilitating Games with Large Groups — Pixels could be distributed to people attending a party, banquet, film showing, or other event. Without knowing which pixel ID is held by each person, the activation status of a pixel could still be used to form random teams to compete in a game, or enable ice breaker activities, like random groups of an audience singing parts of a song. This can be achieved by selecting a random subset among all pixels IDs that were distributed, and flashing those pixels together with instructions by the organizer to form a team, or as a cue to sing during the performance. The pattern of flashes could even be used to indicate a team leader among the subset, or divide the subset into different musical parts.

Personal Notification — Pixels can be used to generate visual notifications to track the progress of an activity or convey basic information like navigation instructions while walking. The flashing frequency of the pixel can indicate the progress of a microwave timer or a reminder for an upcoming appointment. Custom flashing patterns can be used to provide simple navigation instructions, like slow flashing for a right turn, fast flashing for a left turn, and solid red to indicate that the destination is reached.

4.5.2 *One-Dimensional Displays*

Pixels can be arranged along a path or in lines to form one-dimensional displays. Once registered, animated patterns can convey information like direction, activity, and time along the display.

Meeting Location Display — Pixels can be arranged in a line along floors through corridors, and show animations to help attendees locate a meeting room and remind them about the time remaining before the meeting starts. For example, each pixel can light up in sequence along the path to indicate the path direction(Figure4.5a). This animation can speed up to indicate the meeting is starting soon, all the pixels illuminate when the meeting has started and then flash together for a few minutes to hurry people to the meeting. Once it is too late to join the meeting, they can all turn off.

Wait Queue Indicator — At an event, the pixels can be arranged in multiple lines beside different queues to indicate the waiting time or type of queue using different patterns and animations. This would allow people to choose the correct or optimal queue.

4.5.3 *Two-Dimensional Sparse Displays*

Multiple pixels can be distributed throughout a space, and then spatial registration methods can enable individual pixels to communicate information about their locations.

Targeted Class Participation — At the start of a large lecture, each student can be provided with a pixel with a known ID. This could allow the teacher to call on specific students based on records of past participation, to award prizes, or to split students strategically who are sitting near each other for a classroom activity. The association between pixel ID and a student could be done using a variation of our registration method using a very high-resolution camera in the lecture hall to capture the pixel ID flashes and an enhanced algorithm to associate the pixel location with the recognized face of the student.

Location Indicators — Pixels could be placed throughout a temporary setting like a banquet hall, weekend craft workshop, or farmer's market, and the illuminated pixels used to indicate an area or item. For example, to guide a guest to an open table, assist a workshop participant in locating a specific material, or highlighting sales items to a shopper. More than one pixel could be used at each location to convey more information based on illumination pattern, such as stock level or urgency.

4.5.4 *Two-Dimensional Dense Displays*

Multiple pixels can be placed together on the floor, or attached to a metal wall structure like a whiteboard or architectural panels, or even to metal fixtures and elements in a ceiling. The pixels can be placed randomly or formed into specific shapes, and they can be separated into different clusters.

Floor Sign for an Event — The pixels can be arranged on the floor or ground to create an ad hoc display. Once calibrated, the display could show a welcome note during a party as an example (Figure4.5b). The people in the party could also use their phones to create drawings, which could provide additional amusement for the people.

Exam Timer or Sport Scoreboard — During an exam in a large lecture hall, pixels can be arranged on a whiteboard to indicate the time remaining (Figure4.5d). This could be conveyed using numerals, shown at optimized positions on a randomly assembled cluster of pixels, or on a more intentionally laid out display using the layout assistant. Alternatively, the time left could be communicated through more abstract patterns. A similar, but equally compelling application is creating a scoreboard at an ad hoc location for an amateur or informal sports event.

Queuing Signs — As an extension of the event queue example above, multiple clusters of 2D pixel displays could be placed at the beginning of different queues. Each cluster could show symbols or patterns to indicate which queue is open or slow, and suggest alternate queues. For example, displaying shapes like an up arrow for open, flashing for slow, or left and right arrows to suggest the direction of other queues (Figure4.5c). This could be useful to direct crowds at large festivals, open markets, or even in emergency response situations.

Hanging Displays — Some pixels have a tab with a hole extending from the case, which can be attached to strings or hooks to create different types of hanging displays. With many pixels, this could be an alternate form of vertical 2D display, like a sign. Or it could be used for decorative, ambient effects, such as outside in a garden like the NetworkedPixels [26] project.

Ambient Effects — Pixels arranged on the ceiling or walls (Figure4.5e) can be used for ambient effects, like calming patterns at yoga retreat, or accentuating dance music at a festival. They could be attached arbitrarily to any available ferromagnetic elements, like light fixtures or steel structural beams.

4.6 DISCUSSION

We discuss current limitations of our system and future enhancements to further expand the capabilities and possibilities for this type of ad hoc display.

Pixel Size and Display Resolution — The 40mm physical size of each pixel means that the effective display resolution is limited unless viewed from a great distance. Moving to a higher frequency communication would reduce the antenna size and effectively reduce the pixel size.

Display Colour — For simplicity, we use a red LED to create monochrome displays. Each pixel can be easily extended to support a RGB LED to create a colour display, in fact our LED daughter board is designed to support the pins for an RGB LED. The challenge is the impact on power requirements to drive an RGB LED, and a significantly expanded communication protocol that increases from 1-bit to activate an LED to many bits per pixel to specify the colour as well. Another related way to expand the fidelity of the display is to add more bits to select an intensity level of a single colour LED through Pulse Width Modulation (PWM). A useful range may be as little as two or three bits for three to seven intensity levels. Enabling reduced intensities would also reduce amortized LED power consumption.

Run Time — The run time of a pixel could be extended with more efficient hardware and software. For example, adding a "sleep mode" that only checks for base station packets every minute before waking up could drastically increase stand-by time for charged pixels. However, the battery-to-weight ratio will always impose some limits on maximum run time. Other approaches like battery-free pixels that harvest power from ambient energy sources [34] could eliminate this run time ceiling. We initially experimented with an RFID-based approach using a Rocky 100 [146] chip to control an LED and harvest power from RF signals sent from the RFID reader antenna. However, the power harvesting capabilities and communication link are very unreliable, and the large antenna limits how closely pixels could be packed.

Scalability in Terms of Number of Pixels — Using the current base station setup and communication protocol, we can control up to 124 pixels. This number can be increased using the same radio by partitioning hundreds of pixels into sets of 124, sending commands to each set of pixels in batches, and using an offset update signal sent synchronously to all the pixels to make it appear like all pixels update at the same time. This method will reduce the frame rate of the display. Another approach is to modify the base station to support more transmitters. All transmitters can work in parallel, controlling all sets of 124 pixels simultaneously, without compromising the frame rate. However, this requires a fixed association between pixel and transmitter which increases complexity and cost.

Pixel Tracking for Registration — The optical tracking of the pixels using a RGB camera does not work reliably in all lighting conditions. Non-visible light methods can be used for registration, such as the phone NFC reader to scan NFC tags attached to each pixel, or the phone IR receiver to decode a pixel ID flashing sequence transmitted from an IR led in the pixel.

Interactive Pixels — Our system works as an output device by controlling an LED based on the signal received from the base station. But, the pixels can be instrumented with additional sensors to sense touch, motion, light, or sound. The sensed information can be used to directly manipulate the pixel's state or the information shown by the display. Instead of instrumenting all the pixels with sensors, we could design specialized "super" pixels, which are instrumented with sensors. For example, a microphone embedded into the pixel can listen to the user's question and show an appropriate response on the display, or like Particle Display [108] pixels instrumented with an accelerometer allows to interact directly with it through motion.

3D Displays — In principle, these pixels could also be used to create 3D displays. For example, pixels wrapped around a cylinder, or even pixels hanging in a cluster. Our registration method supports the basics of finding pixel locations in 3D, but we would need to relax and refine optimization assumptions, and possibly require more images and more guidance to perform a full 3D registration. Displaying images on a clustered 3D configuration would require a significant extension to our image-fitting algorithms.

Automatic Layout — The process of physically laying out pixels to create a display can be made more convenient by using some time of "paint roller" loaded with pixels. Pixels can also be loaded onto a cylindrical metal stack which can spit out the pixels, when a button is triggered. This process can be completely automated by using a robot programmed with the layout configuration to place the pixels at appropriate locations.

4.7 CONCLUSION

We presented Scatterpixels, a system using custom-built wireless LED pixels that can be arranged in multiple ways to form different kinds of ad hoc displays. Unlike previous work, we describe a full end-to-end solution including hardware, software, and user interfaces for setup. Our individual pixels are simple to set up in many different layouts, and can be conveniently controlled from a smartphone. We developed a comprehensive set of spatial registration methods to accommodate different display configurations, and we provide methods to map content to the displays, including an interactive layout assistant to guide the optimal placement of pixels when expected display content is known. We show how these pixels enable flexible display configurations ranging from one-bit indicators, to 1D lines, to different 2D shapes, clusters, and surface orientations. Our work is a step towards a grander vision, in which individual display pixels are even smaller, powered wirelessly, and inexpensive enough to be "painted" on surfaces, scattered across floors, and embedded in building materials — creating a future where pixels, and displays of all shapes and sizes, can literally be everywhere.

5

PIXELBOARD: A SOLAR-POWERED INDEPENDENT PIXEL MODULE CONTROLLED BY LINE-OF-SIGHT LASER

5.1 INTRODUCTION

Architects have been increasingly embedding digital displays into large building facades for aesthetics and public information, referred to more generally as “media architecture” [44, 128]. In graphics and HCI research, the idea of merging virtual worlds into the physical environment can be traced back to the origins of spatial augmented reality (SAR) [8]. An aspirational goal is to develop ubiquitous display technologies that enable interior designers and architects to incorporate digital information into exterior and interior walls (Figure 5.1).

One approach to add digital content onto arbitrary surfaces uses digital projectors for projection mapping [57, 100], but this requires high power, it can be very expensive because large or complex surfaces require multiple projectors, and line-of-sight is needed for all display surfaces at all times. Given the scale and purpose of media architecture, absolute display resolution, fidelity, and geometric regularity are often less important than economical cost, ease of installation, and long-term maintenance. This relaxation of display requirements enables other approaches, such as a system of addressable pixel elements that can be attached to surfaces to form displays of arbitrary shapes, sizes, and resolutions. These individual pixels can be connected by wires to deliver power and enable display control [11], but this limits how pixels can be placed and introduces significant installation time and expertise for wire routing and connections. A more flexible approach is to use batteries and wireless communication [109, 116], but these require maintenance for charging. If display and communication power are managed carefully, battery-free pixels are possible by harvesting energy from ambient energy sources like NFC [20], RF [77], and light [33, 82, 137]. Typically, the energy harvested from solar panels drives a single display unit that repeatedly updates at a regular time interval. Instead, we update individual pixel elements only when required, resulting in efficient use of the harvested power.

We contribute a technical approach using solar panels to enable a compact battery-free pixel element that can be triggered only when required, to efficiently manage the harvested energy. We call this a “pixelboard element”. The 5 × 5 cm pixelboard element (Figure 5.1c) harvests solar energy to power a simple control circuit for a low-power semi bi-stable single-bit electrochromic display. After a one-time registration, a steerable laser in a central control unit (Figure 5.1d) sends frequency-modulated signals to control the pixel state. Cameras mounted on the central control unit and simple computer vision methods track the laser pointer and acknowledge communication with individual pixels. A steerable high-power spotlight on the control unit can

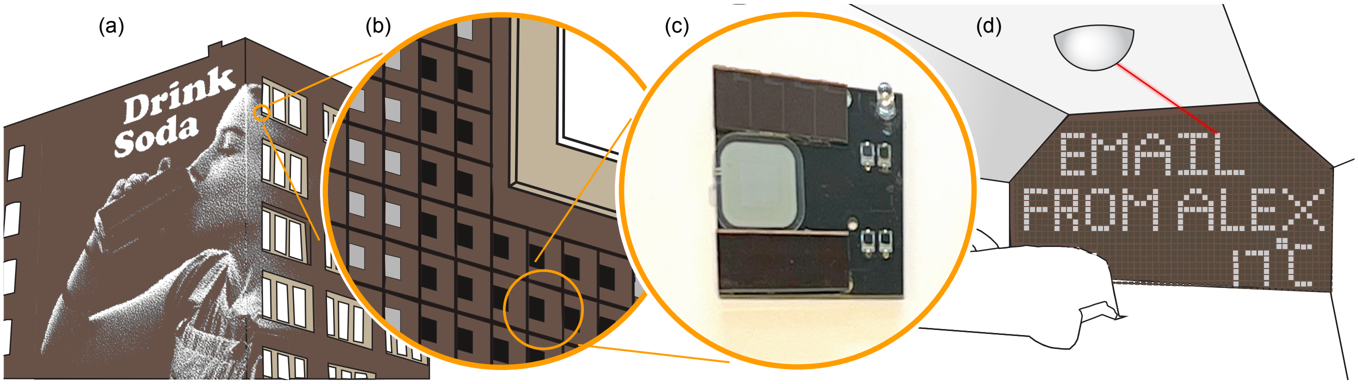


Figure 5.1: Concept for low-resolution independent pixel displays: (a) illustration of low-resolution display as building media facade; (b) closer view showing matrix of independent pixelboard elements; (c) our prototype independent pixelboard element; (d) illustration of interior application showing panels cut to fit wall geometry, and a ceiling-mounted control unit which updates each independent pixel using a laser.

also supplement light energy in off-peak hours. Two pixelboard elements are built and tested to measure charging time, update frequency, and angular operation range. Each pixelboard element takes 3 seconds to update its visual state and slowly loses contrast with time. The update time can be made faster with an improved hardware design and a fully bi-stable e-ink display can overcome the loss in pixel contrast. We discuss how the pixelboard element design could be embedded in construction materials to create building facades and interior design elements, as well as use cases for signs.

5.2 HARDWARE DESIGN

The pixelboard element is the fundamental component that can be placed in the environment to form display surfaces. It is equipped with a solar panel and energy harvesting chip to power itself, and a bi-stable display that conveys one bit of information. A laser control unit is equipped with a two-axis galvanometer laser along with IR and RGB cameras, which are used to identify the locations of the pixels in the environment and control their states. The control unit uses frequency-modulated laser signals to change the pixel display states.

5.2.1 *Pixelboard Circuit Design*

Each pixelboard element is a 5 × 5 cm square-shaped PCB with mounts for two solar panels, a S6AE103A energy harvesting chip, photodiodes, a frequency decoder, and a bi-stable display element (Figure 5.2a). For testing, we attached our pixelboard PCB to a CYALKIT-E04 S6AE103A evaluation kit instead of mounting a standalone S6AE103A chip directly on the PCB (Figure 5.2b). The evaluation board was used to circumvent supply chain issues with individual S6AE103A chips, and it enabled us to explore the features of the chip and easily customize settings during iterative design of

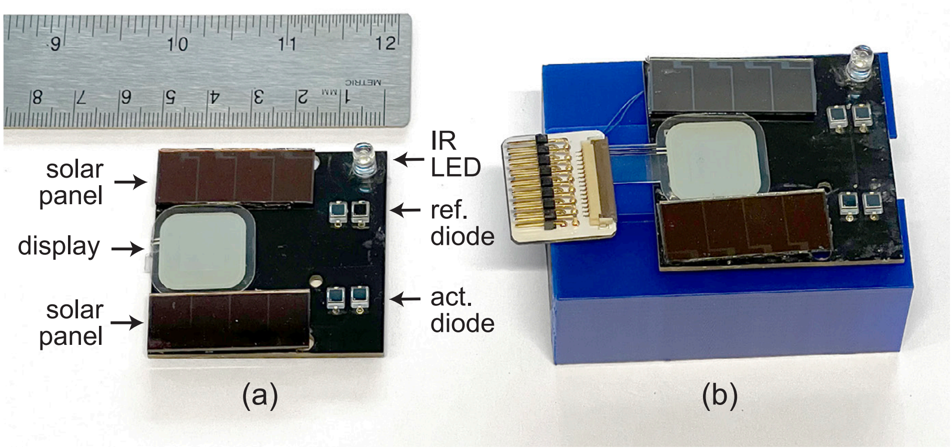


Figure 5.2: Independent pixelboard element: (a) custom PCB as it would be deployed; (b) custom PCB connected to S6AE103A evaluation board enclosed in case for testing.

the PCB. In future versions, the harvester chip can be mounted directly on the PCB, making the pixel slim and ready to be embedded into construction materials.

The energy harvesting (EH) chip has a low startup power (1.2 µW) suitable for harvesting electrical energy in low-lighting conditions. It harvests energy from two solar panels (*Solar1*, *Solar2*, AM-1417) connected in series (Figure 5.3) and stores it in a 0.1F supercapacitor (C_{store}). The EH chip can deliver the energy stored in the supercapacitor to external circuitry from when its voltage reaches the upper voltage threshold (V_{out}^{H} = 3.392V) until it drops to the lower voltage threshold (V_{out}^{L} = 2.196V). These voltage thresholds can be adjusted by changing the resistor network on the evaluation board, but we found the default values worked well in our prototype.

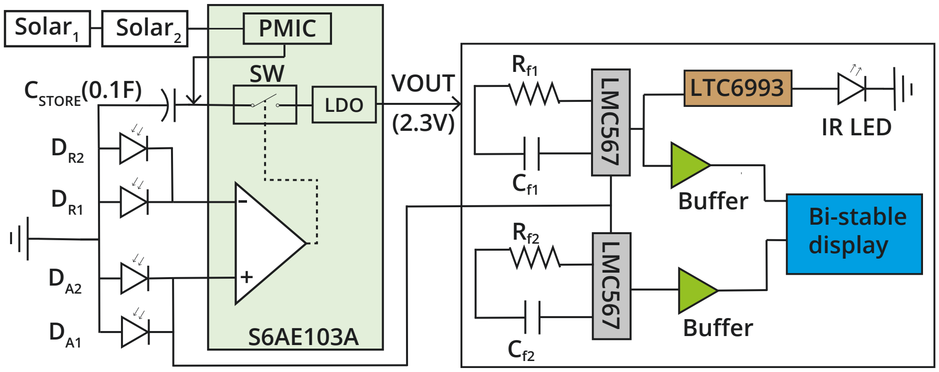


Figure 5.3: Pixelboard circuit design: A S6AE103A energy harvesting chip harvests energy from two solar panels and stores it in a 0.1F supercapacitor. A frequency detection circuit decodes frequency-modulated laser pulses from the laser control unit to update an electrochromic display.

For triggering and communication (Figure 5.3), a photodiode pair (BPW34) connected in parallel is placed to the right side of each solar panel. The two photodiode pairs are placed 1.8 cm vertically apart from each other on the PCB and connect to the built-in low-power comparator of the EH chip. One pair connects to the inverting terminal that acts as the reference photodiode (D_{R1}, D_{R2}), and the other pair connects to the non-inverting terminal that acts as the activation photodiode (D_{A1}, D_{A2}), as shown in Figure 5.3. The activation photodiode pair is covered with a square-shaped 650 nm optical filter to be more sensitive to red light. The filter partially blocks the ambient light falling on the photodiode which sets the comparator to logic '0' by keeping the non-inverting voltage lower than the inverting voltage when no laser trigger pulse is supplied. A 5 Hz square pulse (Figure 5.4) hitting the activation photodiode triggers the comparator and closes the *SW* switch for 2.58 seconds (T0). By closing the switch, the energy stored in the supercapacitor is regulated to 2.3V (*VOUT*) using the internal low-dropout regulator (LDO) of the EH IC and powers the remaining pixel circuitry.

Bi-stable displays are commonly used in energy-neutral systems because they only require power to update their state, but do not require power to hold their state. We used a commercially available semi-bistable electrochromic (Ynvisible) square display in our pixel. Semi-bistable displays lose their contrast over time; for example, the Ynvisible display contrast drops to 80% in 2.5 minutes. Our pixelboard circuit design will be able to drive a fully bi-stable display designed using E-Ink film [36], which can hold its visual state indefinitely.

A frequency detection circuit decodes two frequency-modulated laser signals to control the electrochromic display state. Two low-power tone decoder ICs (LMC567) are programmed using a resistor-capacitor (R_{f1}, C_{f1} and R_{f2}, C_{f2}) combination to detect two frequencies $f1$ and $f2$, respectively. The activation photodiode signal is the input to both the LMC567 ICs. The output of LMC567 is pulled to VDD and pulls down to ground when it detects the programmed frequency. The output from both the LMC567 ICs is connected to the display terminal through a dual digital buffer IC (ZW12445).

A 5 mm IR LED (940 nm) is placed on the top-right edge of the PCB to provide feedback that the laser signal is hitting the activation photodiode. When an $f1$ signal hits the activation photodiode, the output of the LMC567 connected to the LTC6993 is pulled to ground, generating a negative-going pulse. The LTC 6993 monostable multivibrator chip detects the negative-going pulse and turns on the IR LED for 150 ms (Figure 5.4).

5.2.2 *Control Unit*

The control unit consists of a two-axis galvanometer laser, a motorized pan-tilt light source, and a pair of RGB cameras (Figure 5.5). The galvo laser attached with mirrors can precisely control the position of a laser with an optical rotation of ±20 degrees in both the X and Y axis. An Arduino Uno board controls the position of the galvo laser by adjusting the control

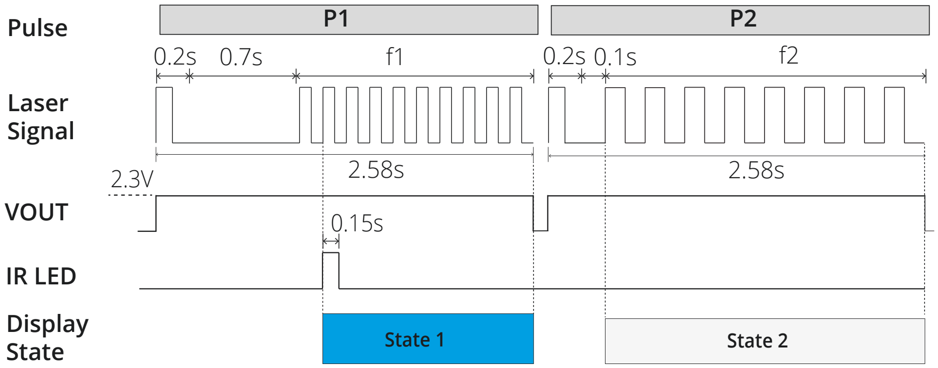


Figure 5.4: Laser Control Signal: Laser switching pulses P1 and P2 are used to change the bi-stable display to 'STATE1' and 'STATE2', respectively.

voltage supplied to the driver board of the X and Y galvo axes. The variable control voltage is generated using a digital-to-analog chip (MCP4921, $0-5$V) interfaced to the Arduino. Op-amps convert the unipolar voltage from the DAC chip to bipolar signals (–5V to 5V) required to obtain full-scale deflection of the galvo.

A 5mW 650 nm Class B Red laser (Adafruit) is focused onto one of the galvo mirrors. The laser can be either off, on, or flash a P1/P2 pulse (Figure 5.4). An LED spotlight source is mounted onto a pan-tilt setup controlled by two 180° servo motors. The spotlight source is used to charge pixels faster or provide light energy to pixels located in low-light environments. The galvo and pan-tilt servo mechanism is mounted on top of a wooden platform.

Stationary cameras are mounted on the bottom part of the wooden platform to track the location of the laser pointer and IR feedback signal emitted from the pixel. One RGB camera is fitted with two stacked IR optical filters to detect an IR LED flash. The cameras are positioned such that their field-of-view covers the pixels of the display panel. The cameras are connected to a PC through USB, and computer vision methods locate the light pulse in the camera frame using OpenCV. Specifically, a frame difference is computed between two consecutive frames in the camera and binary thresholding is performed. Contour detection on the binary image gives the locations of the laser pointer and IR LED flash in the camera frame. The computer vision approach is described in more detail in Section 5.3.1.

5.3 PIXEL REGISTRATION AND CONTROL

To determine where pixelboard elements are in the environment, the control unit performs a pixel registration process. At a high level, the galvo scans across its range of motion and uses a simple computer vision approach to determine the locations of pixels in the environment. It stores these locations in a spatial map. Once the registration process is complete, modulated laser pulses are used to turn individual pixels on or off. To accommodate for errors

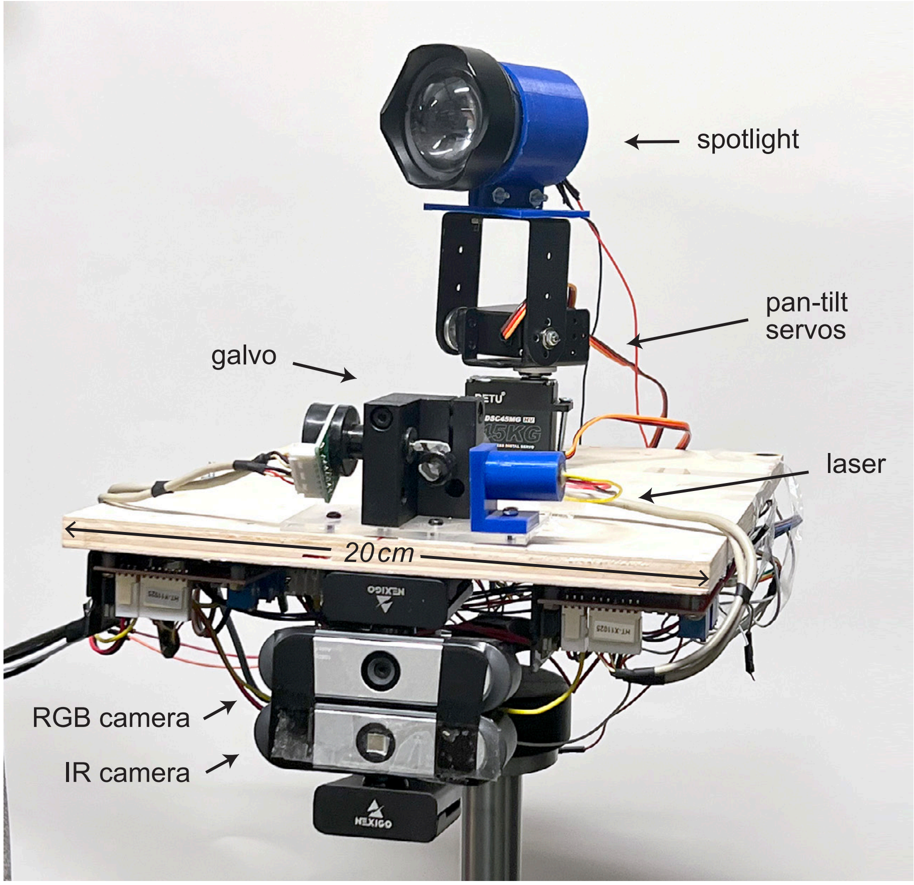


Figure 5.5: Control Unit: A galvo laser and a motorized pan-tilt light source mounted on top of a wooden platform, with two cameras attached to the bottom of the platform.

in galvo positioning, a two-step process is used to accurately locate each pixel's activation photodiode.

5.3.1 *Scanning*

The control unit is installed at a suitable location such that the display panels are within the camera's field-of-view. A one-time laser scanning process is performed to compute and store the spatial information of the pixels after installing sections of display panels in a desired configuration.

The galvo sequentially scans the installed display area by adjusting the control voltage of the X and Y galvo axes, such that full scale deflection is achieved in both axes. For every galvo position, the laser flashes the P1 sequence and looks for an IR LED flash. During the P1 sequence (Figure 5.4), the RGB and IR cameras accumulate the frames in a buffer for post-processing. A frame difference is computed between two consecutive frames in the buffer. A binary threshold is computed on the difference image and contours are detected from it. This process is repeated for all the frames in the buffer and the contour with the maximum area will ideally correspond to the light pulse. A minimum enclosing circle is fitted onto the contour to obtain the location of the laser pointer and IR LED.

If an IR LED flash is detected during the scanning process, then a potential pixel is found. The laser pointer location in the RGB camera's frame of reference, the X-Y control voltage of the galvo, and the location of the IR LED flash in the IR camera's frame of reference are stored. After the laser scans all the possible locations for potential pixels, the information of the detected pixels is saved in a spatial map configuration file. The spatial map is used to control the pixels and display meaningful content on the display panel. The user draws a bounding box on the RGB camera image such that it encloses all the pixelboard elements to reduce the scan area. It would take approximately 20–30 minutes to scan a 1 m^2 display panel.

5.3.2 *Controlling Pixelboard Display State*

The spatial map is used to control and update each pixelboard element's visual state. It contains information about the location of each pixel's activation photodiode and other parameters required to trigger the pixel. The two visual states of a pixel are 'STATE1', corresponding to a blue color, and 'STATE2', corresponding to white color. To update a specific pixel's visual state, the galvo positions itself based on the control voltage parameters obtained from the spatial map, and then the laser flashes the P1 sequence. An IR flash is detected on the IR camera if a pixel is located. In this case, the galvo is accurately positioned to update the pixel. The P1 flashing sequence used to locate the pixel automatically puts the pixel's display to 'STATE1'. A P2 pulse sets the display to 'STATE2', assuming the galvo is accurately positioned.

Ideally, the laser pointer will be able to point at the activation diode and update the pixel's visual state simply by applying the galvo control parameters obtained from the spatial map. However, due to mechanical and electrical noise in the galvo positioning, sometimes the laser does not accurately point at the activation diode. To overcome this, the control unit executes a two-step process to locate the activation photodiode. The first step computes an error representing the absolute difference between the laser pointer's current location in the RGB camera frame and the target location obtained from the spatial map. The error is the calculated separately for the X and Y axes. For example, if the current laser pointer's coordinates are (200, 200) and the target coordinates are (250, 300), then $error_X = 50$ and $error_Y = 100$. The control voltages for the X and Y axes of the galvo are proportionally adjusted based on the $error_X$ and $error_Y$, respectively. This step converges either when the IR LED flashes or the error values fall within a certain tolerance. If the first step converges without the IR LED flash, a second step performs a local circular scan to locate the activation photodiode. The galvo control voltage obtained at the end of the first step is set as the center with the radius equivalent to a small deflection (12mV). The local scan starts by moving along the circumference with the angle incremented by 10° at every step until the IR LED flash is detected. The laser pulsing and IR LED acknowledgement are the same in the scanning process.

5.4 TECHNICAL EVALUATION

We conducted experiments to evaluate the pixelboard element system, including the time required to charge a pixel in different lighting conditions, the number of updates per charge, the update frequency considering a steady power state, and the ability to control the pixel when placed at different angles. These experiments explore the ability of the pixel to operate in different common environments and how the environment affects the different parameters of the pixel.

5.4.1 *Charging Time*

The charging time is dependent on the amount of power available from the solar panel, which varies based on the lighting conditions. We conducted an experiment to estimate the charging time in different lighting settings: outdoors and an indoor office environment, with and without using the spotlight source at different distances from the pixel. Table 5.1 shows the time to charge a 0.1F supercapacitor from 0 to V_{out}^H (3.392V) in different lighting conditions. Initially, the supercapacitor charges from 0 to V_{out}^H, but in subsequent cycles, it charges only from V_{out}^L (2.192V) to V_{out}^H. The supercapacitor continues to charge more than V_{out}^H up to 5V, but the system can be activated and start delivering power once it reaches V_{out}^H. The charging time would increase for a larger supercapacitor, but also allow for a longer operation time before needing to be recharged. The results show that in sunlight, full charging can complete in less than an hour, and that a nearby spotlight can improve indoor charging time by nearly a factor of three.

Table 5.1: Charging time in different settings (all using 0.1F supercapacitor).

Setting	Time (h)		Lux
	$0 - V_{\text{out}}^H$	$V_{\text{out}}^L - V_{\text{out}}^H$	
Sunlight	0.37	0.15	3662 ± 83
Indoor	6.63	2.75	236 ± 5
Spotlight @ 1.5m	2.41	1.00	679 ± 25
Spotlight @ 3m	6.84	2.87	238 ± 3

5.4.2 *Number of Updates per Charge*

Once the laser is exactly pointing at the activating diode, the minimum time required to trigger and update the pixel is 3 seconds, which includes the time to send the laser pulse and process the frames obtained from the camera to locate the laser pointer and IR flash.

Table 5.2: Number of immediate updates for different supercapacitor sizes before depleting the usable energy.

Supercapacitor (F)	No. of Updates		Charge Time (h)
	P1 switch	P2 Switch	
0.047	14	16	0.17
0.1	48	52	0.37
0.33	134	167	1.221
1	415	487	3.699

We conducted an experiment to measure the number of possible updates before the pixels needs to be recharged. The results show that using a large supercapacitor increases the charging time, but allows for a higher number of updates before the pixel stops operating. This is useful when the display requires frequent updates within a short period of time, like for showing an animation. For different supercapacitor sizes, the number of possible updates for P1 and P2 trigger pulses sent once every 5 seconds is shown in Table 5.2. This table also shows the estimated time to charge from 0 to V_{out}^{H} in sunlight using the average power calculated from Table 5.1.

5.4.3 *Update Frequency for Steady Power State*

We calculated the amount of time it would take to replenish the charge consumed for a single P1 switch in different lighting conditions. A 0.1F supercapacitor is charged to 3.48V ($\approx V_{\text{out}}^{H}$) and when a P1 pulse is sent, its voltage drops from 3.48V to 3.46V. Table 5.3 shows the time required to replenish the energy consumed by each switch and maintain the steady state voltage (3.48V) in different lighting conditions. The results show that even in less ideal lighting conditions, updates can still trigger in less than about four minutes while maintaining steady state.

5.4.4 *Angle Study*

We tested the ability of the control unit to trigger the pixel and detect the IR flash from it when placed at different angles and distances. The pixel is 'triggered' when the P1 pulse is flashed on the activation diode, and 'detected'

Table 5.3: Time to maintain steady state after a P1 pulse trigger in different light settings (all using 0.1F supercapacitor).

Setting	Update Time (s)
Sunlight	13
Indoor	227
Indoor & spotlight @ 1.5 m	86
Indoor & spotlight @ 3 m	245

when the IR flash can be located by the IR camera. The pixel is mounted on a tripod and tested at three distances (d): 1 m, 2 m, and 3 m. At every distance, the angle between the normal vectors from the laser galvo and pixel activation photodiode (θ) is varied between 0° and 90° (Figure 5.6).

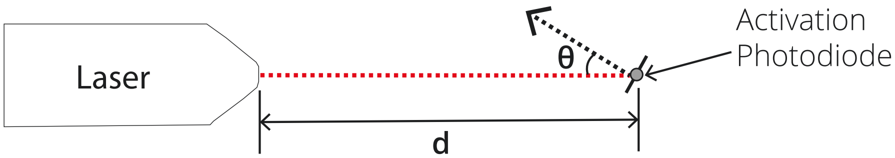


Figure 5.6: Angle study setup: A pixelboard element is placed d m away from the control unit at different angles (θ) between the normal vectors from the galvo laser and pixel activation photodiode

A local scan is repeated five times around the area of the activation photodiode to locate a pixel for different combinations of angle and distance. If the pixel is triggered or detected in more than 50% of the trials, then the trial is considered successful. Table 5.4 shows the different angles and distances at which the pixel can be triggered and detected. The results show that the pixel can be reliably detected even when it is not placed directly facing the control unit, and it can be triggered at an angle up to 84° even when it cannot be detected. Our current hardware uses an IR LED with a beam angle of 20°, but it could be replaced with an IR LED with a wider beam angle to improve the detection angle.

5.5 APPLICATIONS

There are many potential applications for independent pixel elements. They can be used to build new objects, can be attached to existing surfaces or objects, or can replace existing building infrastructure. Here, we propose applications that illustrate how pixelboard elements could be used to form large display surfaces in a variety of environments.

Table 5.4: Pixelboard element triggering and detection from different angles.

Distance (m)	Detect Angle (deg)	Trigger Angle (deg)
1	0 – 48°	0 – 90°
2	0 – 18°	0 – 90°
3	0 – 12°	0 – 84°

Indoor Wall Display — Pixelboard elements could be attached to the wall or ceiling of a bedroom or a living room. The wall display could show pictures, notifications, or weather information; the ceiling could simulate a stargazing environment. Figure 5.1d illustrates how this arrangement could display an email notification and the current temperature.

Programmable Bus Display — A bus fitted with pixelboard elements on its exterior could act as a digitally programmable sticker. The bus display could show daily weather updates, advertisements or a social message. When the bus is parked overnight, the display could be programmed with new information. Multiple control units could be installed at strategic locations in the bus shelter to update all the pixels on the bus.

Advertisement/Road Sign Update — Pixelboard elements could be used to build dynamic billboards or road sign indicators in remote environments where access to electrical power or cellular networks is unreliable. To update the pixels with new information, a vehicle fitted with the control unit could drive to the display location and program the pixels by sending appropriate laser pulses.

Media Facade — Pixelboard elements attached to a building exterior could show information about the building, advertisements, or a list of tourist attractions along with a map. The pixels could be arranged to fit the existing building infrastructure; for example, to accommodate windows, pipes, and other building structures. Figure 5.1a shows a building exterior augmented with pixels elements to show an advertisement.

When pixelboard elements are unevenly spaced or sparse, showing raster graphics is not as straightforward as with a conventional rectangular display. When an existing image needs to be mapped onto a pixelboard display panel, image-fitting algorithms can be used account for the visual state of each pixel in the display panel to closely resemble the given image. Sato et al. [109] developed algorithms that map an image onto irregularly shaped and sparse pixel displays. The algorithm determines which pixels have to be turned on and turned off. This information could used by our laser control unit to update the pixels to show the required content.

5.6 LIMITATIONS AND FUTURE WORK

We discuss current limitations of our pixelboard prototype and potential solutions to overcome them.

Reducing the pixel bezel — Our implementation has bezels surrounding the active display area (electro-chromic display), comprising opaque solar panels and photodiodes. The bezel could be reduced by replacing the amorphous solar panels with two transparent solar panels [82], each covering half of the pixel PCB area. The solar panels could then be placed directly on top of the active display element. One solar panel would be covered with a 650 nm optical filter to act as the activation photodiode for controlling the pixel. The size of the active display element(electrochromic display) could also be made smaller, as long as a PCB can fit all the electronics beneath it. A pixel element with a smaller bezel and display element would enable the design of denser display panels.

Fully bi-stable display element — A fully bi-stable display element can hold its visual state indefinitely without a change in contrast, and it does not require power to do so. Our pixel uses a semi-bistable electrochromic display element, which loses 50% of its contrast within 7 minutes. A E-Ink pixel is a truly bi-stable display element, but we were not able to source affordable 1-bit square E-Ink panels in small volumes. We investigated building a rectangular one-bit bi-stable pixel by re-purposing the E-Ink film from a Kindle e-reader and had initial success. However, the pixel was not reliable because the conductive adhesive used to attach the film to a substrate was not stable, and we also had difficulty accessing the terminal leads needed to control the display.

Scanning pixels spread over a large area — The current version of the control unit hardware can scan a 170 × 170 cm area when placed three meters away from the display surface. The scanning area is limited by the full-scale deflection of the galvo. This could be overcome by attaching the galvo to a precise motorized pan-tilt mechanism such as a servo or stepper motor. Further, multiple control units strategically positioned in the environment could update pixels that are spread over a large area or do not fall within the triggering or detection angular range (Table 5.4).

Update Speed — In our current implementation, the laser control unit takes 3 seconds to update a pixelboard element. This time includes the time for the laser to send the 5 Hz trigger pulse, P1/P2 pulse, and detect the IR LED acknowledgement. When the 5Hz trigger pulse is sent, the switch (SW) is closed for 2.58 seconds (T0) to supply power to pixel circuitry (Figure5.3). The default setting for T0 is 2.58 seconds on the CYALKIT-E04 S6AE103 evaluation board, but can be changed by tuning a timing capacitor. For simplicity, we left T0 at the default setting, but it can be reduced to 1 second which will be sufficient to update the pixel.

5.7 CONCLUSION

We propose a design for an "independent pixel" that can be used in architecture and interior design settings. It features an energy-efficient approach to control pixels only when needed, using solar panels and a line-of-sight laser control system. We prototyped two pixel elements and we conducted technical experiments to calculate the charging time, the number of updates per charge, update frequency to maintain a steady power state, and maximum trigger and detection angle for the implementation.

6

PIXELBRUSH: AN UNPOWERED CUSTOMIZABLE FLIP-DOT DISPLAY UPDATED USING A HANDHELD BRUSH

6.1 INTRODUCTION

Traditionally, the pixels of a display are wired to a central control unit to receive power and control signals to show information. The pixels are constrained within a screen because of wiring constraints which reduce the flexibility to create custom display configurations. However, displays can be simplified when the information shown requires infrequent updates. For example, an advertisement display that is updated once every six months with a new ad or a sign board display that is updated to indicate a road closure during an event. The question is: how can we build simple displays that occasionally update information and provide installation flexibility?

One approach requires human intervention to update a display using an external device. Previous work updated E-ink displays [20, 22] using NFC signals when a user holds a phone nearby (<15mm). The system required 2.4 seconds to update a high-resolution E-ink screen. Grafitti Fur [120] and Grassfiti [121] convert a carpet and artificial grass turf into a display screen. A brush mounted with a servo motor rolls over the display surface to raise or flatten the fibers to show information. Using a carpet or grass turf as a display is an original concept, but they have low contrast, require precisely aligned movement of the brush, and the displayed information changes when people walk over it.

A compelling type of display technology that was popular decades ago is the flip dot. These were once ubiquitous in train stations, airports, and on buses. A conventional flip dot has a two-sided magnetized disc that rotates along an axis parallel to the display surface when triggered by an electromagnet placed beneath it. It is thick because of the space required for the electromagnet and the disc to rotate as it flips. Physical stops make the disc flip to one of the two sides. The two sides of the disc have different colours to represent the visual state of a monochrome pixel. By controlling the direction of the current through the electromagnet, the disc is flipped to the appropriate side. A display built with flip dot pixels consumes a huge amount of power to drive the electromagnets and requires wiring to a central control unit to update it.

In our system, we move the electromagnet to a movable brush and use an NFC reader to detect the pixels. This removes the wiring constraints in the displays and makes the pixel simple and cheap. Our custom flip dot pixel is a 3D-printed two-sided disc embedded with an NFC tag and a permanent magnet enclosed in a cylindrical housing (Figure 6.1). The pixel is monochrome with one side black and the other side white, and fully bi-stable as it is mechanical. When the brush detects a pixel underneath it, the electromagnet is triggered with the correct polarity to set the pixel to the desired colour state. These pixels enable the design of general-purpose displays, have high pixel contrast, and can be updated by a conventional back-and-forth brushing action reminiscent of erasing a blackboard.

We conducted technical experiments to test the brush mechanism's ability to update the pixel at different speeds, angles, or offsets from the pixel center. The results show that a single pixel element can be updated reliably at a brushing speed of 30 cm/s, 4 mm away from the pixel center, and from all angles. We propose different applications and discuss how our system can potentially replace the existing display solutions.

We contribute the design of a mechanical pixel element that can be embedded into any surface to form a display panel whose information is updated occasionally using a movable electromagnet brush.

6.2 RELATED WORK

We discuss prior work on displays that require human intervention to update. Unlike traditional displays, which automatically update content from a central controller, these approaches require manual updates using an external device, such as a mobile phone or a purpose-built actuation device.

Some E-ink displays harvest energy and communicate via NFC signals when a smartphone with NFC compatibility is placed near it. Alterwear [20] and Alternail [22] harvest power and communicate through NFC signals from a phone to update a high-resolution E-ink display screen. These displays require complex circuitry to harvest power and communicate with them and are slow to update.

Some approaches use a purpose-built brush that moves over a display surface to program it and show the desired information. The display surface can be carpet, grass turf, or a magnetophoretic surface. Graffiti Fur [120] converts a carpet into a high-resolution display screen by either raising or flattening the fibers of the carpet. A brush containing a servo motor attached to a rod either raises the fiber or leaves it flat. A movement sensor on the brush locates its position on the carpet and sets the fiber direction according to the image to be shown. Similarly, Grassfiti [121] converts a conventional grass surface into a large-scale display by raising or flattening its fibers. These displays do not require power to show a static image, but the information is affected when a person walks over it.

SweepScreen [88] uses a magnetophoretic surface to form displays of custom shapes and sizes, which can be brushed to show the desired image. A magnetophoretic surface is made up of ferromagnetic particles suspended between two substrates which do not require power to show a static image. A brush contains a series of 20 electromagnets and a movement sensor to locate its position within the display area. When the brush moves over the display, based on its location, the electromagnets are triggered to set the desired image. The contrast of the display is dependent on the brushing speed, with the best contrast achieved at a brush speed of 4.5 cm/s. It requires the display to be flipped over to erase an existing image and requires an aligned movement of the brush.

Our work explores the design of purpose-built pixels that support higher contrast, do not rely on the absolute position of the brush to update the pixels, and use a brushing action reminiscent of erasing a blackboard.

6.3 PIXEL HARDWARE AND DISPLAY PANEL DESIGN

This section describes the design of the standalone flip dot pixel and display panels formed by embedding the pixels into construction material.

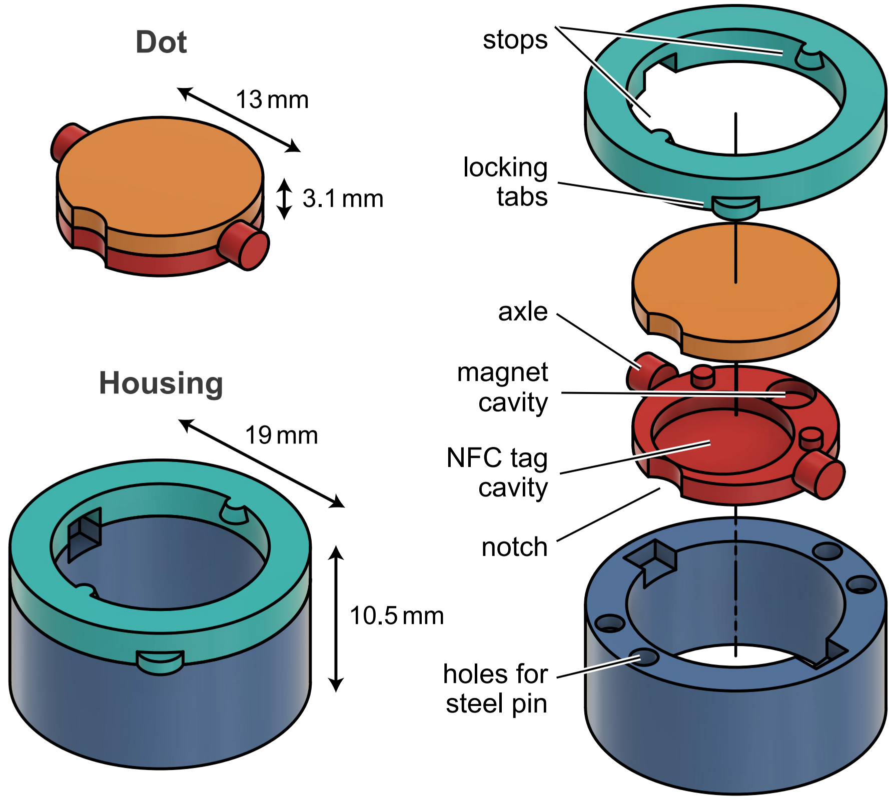


Figure 6.1: Exploded view of the flip dot pixel and shows the dimensions of the different components

6.3.1 *Flip Dot Pixel*

The flip dot pixel consists of a circular disc enclosed in a cylindrical casing (Figure 6.1). The disc is made from two 3D-printed halves, one white and the other black, representing the two colour states of the pixel. The black half has axles on diametrically opposite sides for rotating the disc. The disc has slots for an NFC tag and a magnet, and a small cutout on its periphery (Figure 6.2b).

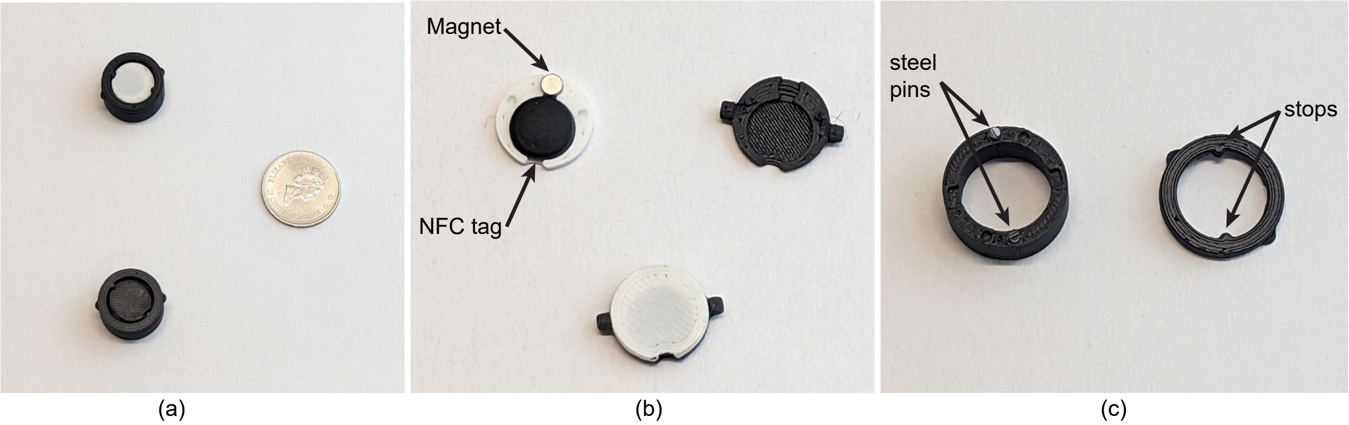


Figure 6.2: Flip Dot Pixel Design: (a) two fully assembled flip dot pixels showing the black and white colour states; (b) rotating circular disc embedded with an NFC tag and a permanent magnet; (c) a cylindrical housing containing the bottom part with two steel pins and the top part printed with two stops.

The NFC tag (*GoToTags: KJ8LQ4BA73*) has an NTAG213 chip enclosed inside a 9mm diameter machine-washable casing. Each tag has a unique ID that identifies each pixel. A cylindrical permanent magnet (3mm diameter and 1.3mm height) is placed on the edge of the disc. Once the NFC tag and magnet are placed inside the disc, the two halves are glued together using superglue.

The circular disc is placed inside a 3D-printed cylindrical housing (19mm diameter and 10.5mm tall), consisting of two components (Figure 6.2c). The bottom part contains two small holes on the edge, diametrically opposite each other. Two steel pins made from 3.6 mm long segments of 1.6mm diameter 'mechanics' wire are inserted into these holes. The steel pins serve to pull the disc to a flat resting position when it rotates. The top half of the housing is printed with a tiny protrusion/stop on its edge to limit the disc's rotation to 180 degrees and prevent it from rotating 360 degrees. Two fully assembled flip dot pixels representing the two visual states of the pixel are shown in Figure 6.2a.

6.3.2 *Display Panel Design*

The flip dot pixels are directly embedded into construction materials or into architectural surroundings, forming a display surface. This can be achieved by inserting the pixels into holes created in construction material like wood or into existing building surfaces such as a concrete wall. The pixels can be arranged on the desired surface either in an evenly spaced grid or in an arbitrary configuration.

A display panel is created by placing the pixels between the top and bottom substrates of the desired construction material. Circular holes with a diameter of 19 mm are cut into the substrates using a laser cutter or a drill press to

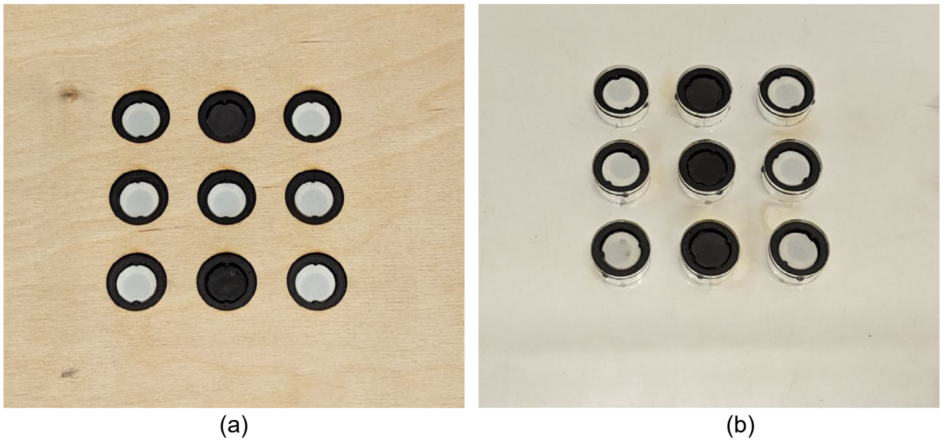

(a) (b)

Figure 6.3: Display panel design in evenly spaced 3 x 3 grid: (a) pixels embedded into wood; (b) pixels embedded into acrylic.

accommodate the pixels. The holes are arranged in an evenly spaced grid with 28.9 mm spacing between them. The pixels are inserted into the holes on the bottom substrate, and then the top substrate is placed on top to secure the pixels in place. The top and bottom surfaces are glued together to form the display panel. Figure 6.3a shows pixels embedded into wood with a 2.8 mm thick top substrate and 9 mm thick bottom substrate. Similarly, Figure 6.3b shows pixels embedded into acrylic with a 3 mm thick top substrate and 10.3 mm thick bottom substrate. We imagine these panels could be manufactured in a facility and produced at scale. They could be purchased, customized, and installed by construction workers without requiring technical expertise. Instead of using display panels, the pixels can also be embedded into existing surfaces to form a display, for example, by creating holes in concrete or drywall to insert pixels into them.

Some scenarios do not require the pixels to be arranged in a grid when the content to be shown on the display is known prior. The pixels can be arranged in arbitrary display configurations in terms of shapes, sizes, and pixel densities on construction material or directly into the architectural surroundings.

6.4 BRUSH DESIGN

This section describes the design of a handheld brush used to set the visual state of the pixels in the display. The brush detects the ID of each pixel and triggers an electromagnet to flip the pixel's disc to the appropriate side.

6.4.1 *Automatic Pixel State Brush*

The automatic pixel state brush contains an electromagnet placed on top of an NFC reader antenna and electronic control unit to drive the different

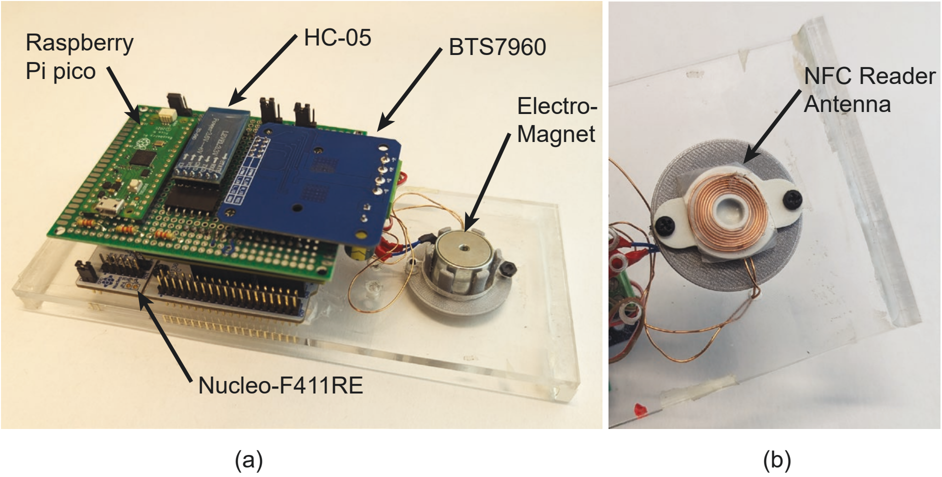


Figure 6.4: Automatic Pixel State Brush: (a) top view of the brush showing the electronic control unit and electromagnet mounted on a plexiglass; (b) bottom view of the brush unit showing the NFC reader antenna.

components on the brush (Figure 6.4a). The NFC reader antenna is a spiral coil attached to a 3D-printed circular substrate (Figure 6.4b). The antenna has a 6 mm inner diameter and 22mm outer diameter, wound with a 24-gauge enameled magnet wire. The antenna is connected to a 1.4W NFC reader evaluation board (NFC05A1). The NFC reader is interfaced with a microcontroller evaluation board (Nucleo-F411RE) through a Serial Peripheral Interface (SPI) protocol. When the brush moves over the pixels, the microcontroller reads the ID of the NFC tag embedded into the flip dot pixel. The detected pixel ID is then transmitted via serial communication to a Raspberry Pi Pico microcontroller through serial communication. The Pi Pico is programmed using micropython, making use of its dictionary functionality to efficiently manage pixel IDs and their corresponding visual states ("1' white or "0' black). To enable communication between a phone and the brush, an HC-05 Bluetooth module is interfaced with the Nucleo-F411RE.

The electromagnet on the brush is used to rotate the pixel's disc and set the colour state of the pixel. The direction of rotation is controlled by adjusting the polarity of the electromagnet. The strength of the magnetic field generated by the electromagnet is proportional to the square of the current flowing through it, and the polarity is determined by the direction of the current flow. A cylindrical electromagnet with a holding force of 25N, measuring 20mm in diameter and 20 mm tall is placed directly on top of the antenna. To control the electromagnet, the brush uses a BTS7960 H-Bridge motor driver board controlled by the Pi Pico. The Pi Pico signals the motor driver to supply current to the electromagnet and also control the direction of current flow.

As the brush moves and the NFC reader antenna detects a pixel (Figure 6.5), the electromagnet is triggered with appropriate polarity to update the pixel's visual state. Since the brush is moving, it is essential for the electromagnet to respond instantly before the brush moves to the neighbouring pixels. To

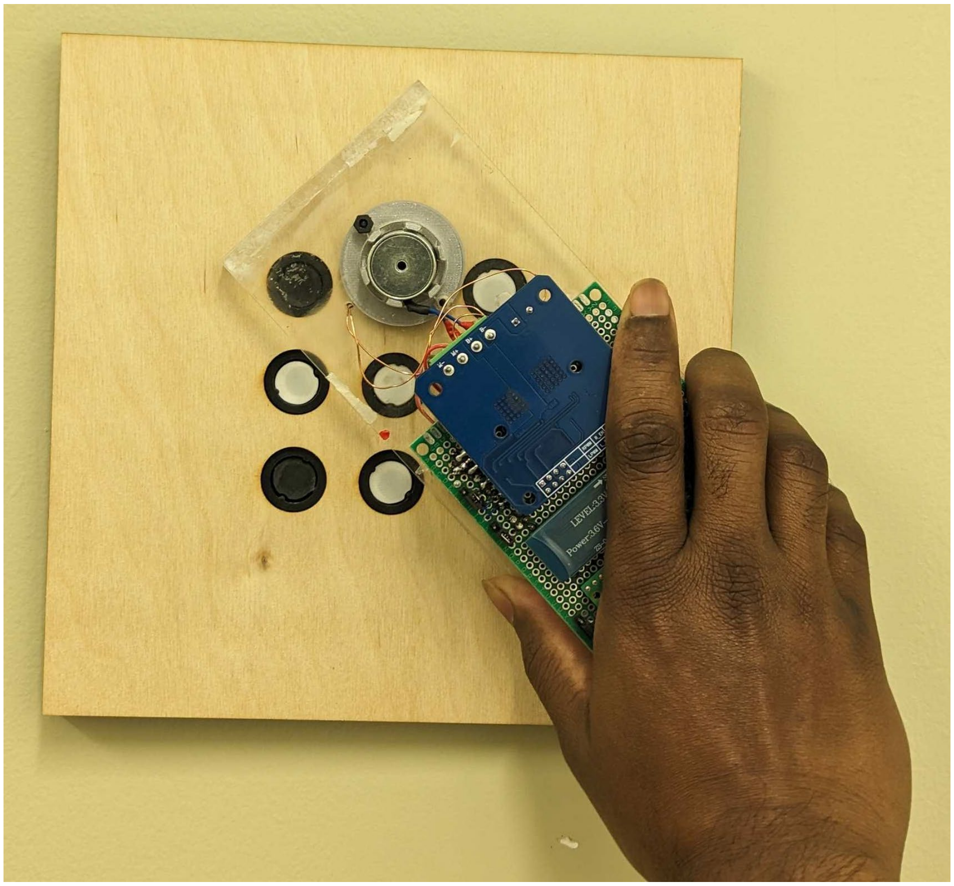


Figure 6.5: A human brushing over the pixels using the automatic pixel state hand-held brush.

achieve this, the magnetic field strength must be maximized within a short time frame, ensuring reliable and accurate pixel updates. The electromagnet is equivalent to an inductor, resisting the increase of current through it, which influences the magnetic field strength. To force the current to reach a higher value within a short period, a voltage greater than the rated voltage of the electromagnet is applied for 15 milliseconds whenever the antenna detects a pixel. The electromagnet is rated at 5 volts, but we supplied 25 volts in short time bursts whenever a pixel is detected by the NFC antenna.

An Aruco marker is attached to the top of the brush to perform a one-time registration of the spatial location of pixels, as explained in Section 6.5.

6.4.2 *Manual Pixel State Brush*

We built a simplified brush for updating the pixels without the need to detect their individual IDs (Figure 6.6a). This brush can be imagined as a digital pencil, enabling users to manually draw on the display panel. It contains an electromagnet controlled by a BTS7960 motor driver module, an Arduino pro mini microcontroller, and two momentary switches (Figure 6.6b). The switches control the direction of the current flow to set the polarity of the electromagnet. One of the switches sets the pixel to the white side, and the other switch sets it to the black side. Using these two switches, a user can draw a desired image on the display.

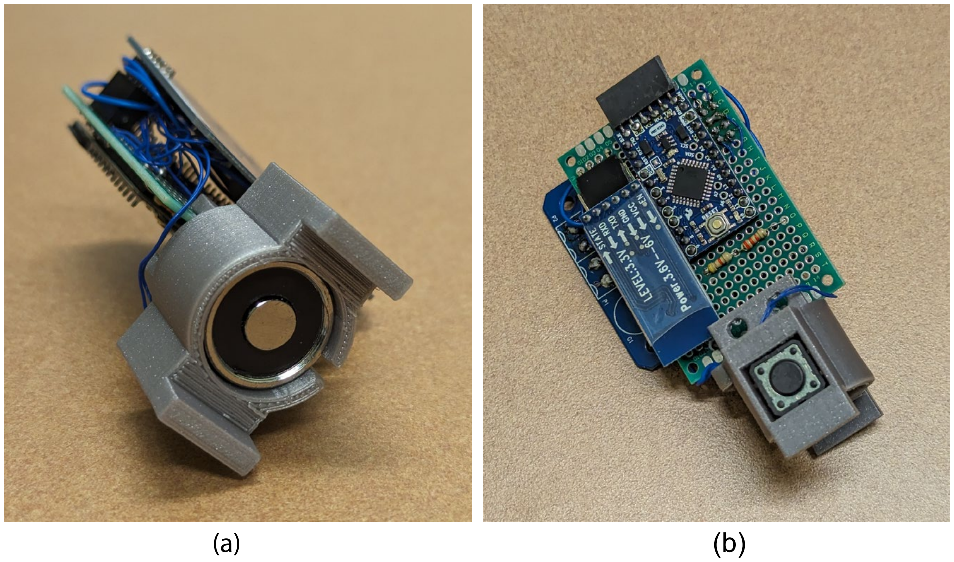

(a) (b)

Figure 6.6: Electromagnet Only Brush: (a) Electromagnet Setup; (b) buttons are assembled in a 3D printed case to resemble a pen

6.5 SPATIAL REGISTRATION AND DISPLAY CONTROL

This section describes the registration method to generate a spatial map and how to control the display to show the required information using the automatic pixel state brush.

Spatial Registration — Once the pixels are installed in the desired configuration, a one-time registration process finds the relative locations of the pixels and their IDs. This information is then stored as a spatial map. The spatial map is used to control the pixels of the display to show meaningful information. A smartphone (Google Pixel7) running a custom Android application is mounted on a tripod and positioned such that the smartphone's camera views all the pixels. The camera tracks the location of the Aruco marker mounted on the brush in real time using OpenCV. The process provides 2D image coordinates corresponding to the four corners of the marker. The marker's center is computed as the average of the four corner points.

As the brush moves over the pixels, the NFC antenna detects the ID of the pixel directly underneath it and transmits this information to the phone through Bluetooth. Whenever the phone receives an ID from the brush, the marker's center is stored along with the received ID. This process continues until the brush has traversed all the pixels, resulting in a complete spatial map. The spatial map is stored as a text file for later use.

Control the display — The spatial map information is read from the stored text file and displayed as a 2D map on the phone. Users tap on the map to program the visual state of the pixels. This pixel configuration is sent to the brush through Bluetooth. To update the pixels with new information, the user moves the brush over them, and the electromagnet is triggered to set the desired colour state of each detected pixel.

6.6 TECHNICAL EVALUATION

We conducted technical experiments to evaluate the ability of the automatic pixel state brush to update the pixel at different brushing speeds, brushing angles, and brushing offsets.

To run these experiments, the brush is mounted on an Ultimaker 2+ 3D printer extruder head to simulate different brushing patterns (Figure 6.7a). The printer head can move at a maximum speed of 30 cm/s on both axes. The antenna-electromagnet is attached to the bottom of the head using a 3D-printed mount (Figure 6.7b). A cooling fan is placed above the electromagnet to dissipate heat during the experiment. The motor driver on the brush is powered by a regulated bench power supply (25V, maximum current limited to 1.5A) which drives the electromagnet. The brush communicates with a PC running Python to send commands to move the brush and receive the ID of the detected pixel.

The wooden display panel (3x3 grid) described in Section 6.3.2 is used to run the technical experiment. The display panel is placed on the build tray of the 3D printer, and an RGB camera mounted on a tripod captures images of the display panel during the experiment (Figure 6.7a). Images are captured before and after every brushing sequence and are saved for further processing. The images are later processed using OpenCV to determine the visual state of each pixel (black or white) and detect whether the pixels were updated correctly after the brushing action.

We conducted three experiments: Brushing Speed, Brushing Angle, and Brushing Offset. Each experiment has two conditions: update a single target pixel (1x1) and update three target pixels in a line (3x1). We measured the success rate of updating the target pixels and the false rate of updating the non-target pixels. An update corresponds to rotating the disc of the pixel from the black to the white side. Success (1x1) corresponds to the success rate of updating a single target pixel. False (1x1) is the false rate of updating the non-target pixels. For the 1x1 experiment, P5 is the target pixel for all the experiments, and the remaining pixels are non-target pixels (Figure 6.8a).

Similarly, we repeated the experiments with three target pixels in a line (e.g. P3, P5, P9 or P3, P4, P7). Success (3x1) corresponds to the success rate of updating three target pixels along the line. False (3x1) corresponds to the false rate of updating non-target pixels. For the 3x1 condition, P4, P5, P6 are the target pixels for the speed and offset test. But, the target pixel varies for

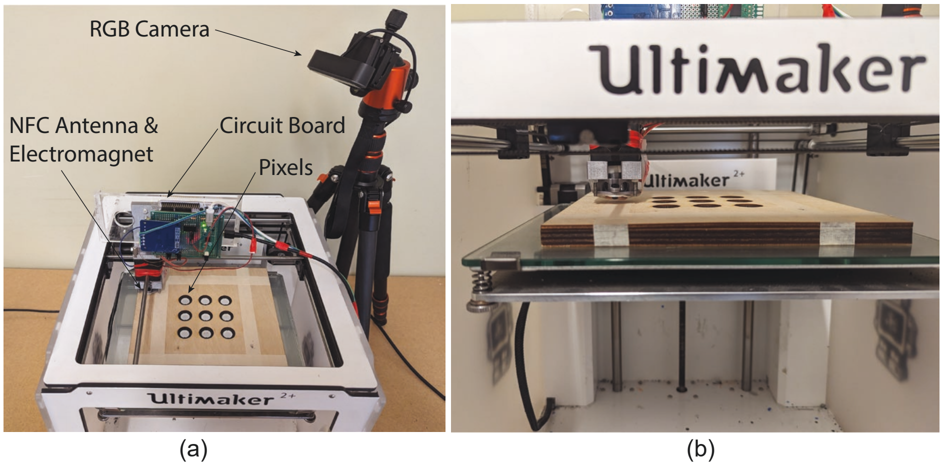


Figure 6.7: Technical experiment setup: (a) Automatic pixel state brush mounted on 3D printer head, display panel placed on the build tray, and RGB camera mounted on a tripod; (b) close-up view of the NFC antenna-electromagnet unit attached to the printer head.

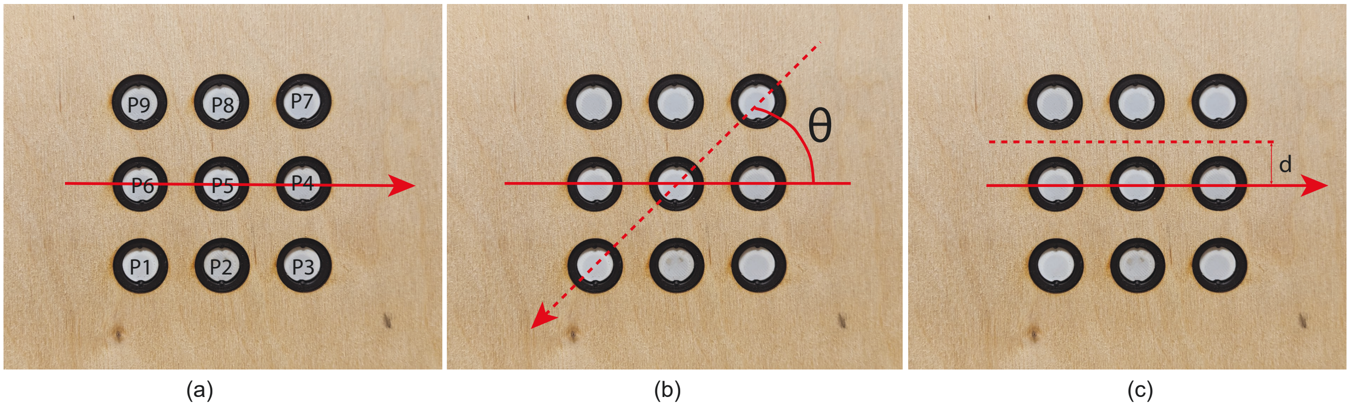


Figure 6.8: Technical Experiment: (a) Brushing Speed test; b) Brushing Angle test; c) Brushing Offset Test.

the angle test for different brushing angles. For example, P8, P5, P2 are the target pixels when the brushing angle $\theta = 90$. The pixels apart from the target pixels are treated as non-target pixels. The Count (1x1) and Count (3x1) value corresponds to the average number of NFC reads of the target pixels during a brushing sequence.

The brushing sequence is repeated 10 times to calculate the success rate, false rate, and average detection count. A success rate of 90% means that the brush successfully updated the target pixels in 90% of the attempted trials. A false rate of 10% means that the brush falsely updated at least one of the non-target target pixels in 10% of the attempted trials. After every brushing sequence, the pixels are reset to black.

6.6.1 *Brushing Speed*

This experiment evaluates the ability of the brush to update the target pixels at different speeds. The brush is swept in a straight line across the pixel grid (Figure 6.8a). We experimented with brushing speeds from 5 cm/s to 30 cm/s and calculated the detection count, success rate, and false rate for the single target pixel and three target pixels (Table 6.1). For the single target pixel condition (1x1), the brush can detect and update the target pixels with a 100% success rate up to 30 cm/s. For the three target pixels condition, the brush can detect target pixels up to 30 cm/s but can be updated with an 80% success rate at 20 cm/s. We used a brushing speed of 15 cm/s as a reasonable baseline to run the angle and offset tests.

Speed (cm/s)	Count (1x1)	Success (1x1)	False (1x1)	Count (3x1)	Success (3x1)	False (3x1)
5	15.1	100.0	0.0	15.3	100.0	0.0
10	7.2	100.0	0.0	7.1	90.0	0.0
15	4.8	100.0	0.0	4.3	60.0	0.0
20	3.4	100.0	0.0	3.0	80.0	0.0
25	2.5	100.0	0.0	2.2	70.0	0.0
30	2.0	100.0	0.0	2.0	20.0	0.0

Table 6.1: Brushing Speed Test: NFC detection Count, Success Rate, and False Rate for 1x1 and 3x1 conditions for different brushing speeds

6.6.2 *Brushing Angle*

This experiment evaluates the ability of the brush to update the target pixels when brushed at different angles (θ) (Figure 6.8b). We experimented with different brushing angles from $\theta = 0^\circ$ to 360° in steps of 45° and set the brushing speed at 15 cm/s. Table 6.2 shows the detection count, success rate, and false rate for the single target and three target pixel conditions at different brushing angles. For the single target pixel condition (1x1), the brush can update the target pixels with 100% success rate for all brushing angles. For the three target pixel conditions, the pixels are updated with more than 90% success rate for the majority of the brushing angles. Overall, the results show the pixels can be updated when the brush moves over them at different angles.

6.6.3 *Brushing Offset*

This experiment evaluates the ability of the brush to update the pixels when the brush is offset (d) from the pixel center (Figure 6.8c). We experimented with offset values $d = 0, 2, 4, 6, 8$ mm and set the brushing speed at 15 cm/s.

Angle	Count (1x1)	Success (1x1)	False (1x1)	Count (3x1)	Success (3x1)	False (3x1)
0	4.7	100.0	0.0	3.9	90.0	0.0
45	4.7	100.0	0.0	4.7	90.0	0.0
90	4.8	100.0	0.0	4.6	90.0	0.0
135	4.9	100.0	0.0	5.0	70.0	0.0
180	4.7	100.0	0.0	3.3	40.0	0.0
225	4.6	100.0	0.0	2.5	90.0	0.0
270	4.6	100.0	0.0	3.9	100.0	0.0
315	4.5	100.0	0.0	3.9	100.0	0.0

Table 6.2: Brushing Angle Experiment: NFC detection Count, Success Rate, and False Rate for 1x1 and 3x1 conditions for different brushing angles

Offset	Count (1x1)	Success (1x1)	False (1x1)	Count (3x1)	Success (3x1)	False (3x1)
0	5.0	100.0	0.0	4.0	70.0	0.0
2	5.0	100.0	0.0	4.0	70.0	0.0
4	5.0	70.0	0.0	4.0	0.0	0.0
6	4.0	0.0	0.0	4.0	0.0	0.0
8	3.0	0.0	0.0	3.0	0.0	0.0

Table 6.3: Brushing Offset Experiment: NFC detection Count, Success Rate and False Rate for 1x1 and 3x1 conditions for different brushing offset

Table 6.3 shows the detection count, success rate, and false rate for the single target pixel and three target pixel conditions at different brush offsets. For the single target pixel (1x1) condition, the brush can update the target pixel with a 70% success rate when it is 4mm offset from the pixel center. For the three target pixel (3x1) condition, the brush can update the target pixels with 70% success rate at a 2 mm brush offset. Table 6.3 shows that pixels can still be detected even when the brush is 8 mm away from the pixel center, but cannot be updated. To solve this issue, an electromagnet with a stronger or wider magnetic field could update pixels at a brush offset of 8 mm.

6.6.4 *Discussion*

The technical experiment results vary each time the experiment is conducted due to the mechanical nature of the pixels and overheating of the electromagnet. To achieve more consistent results, the pixels should be designed with increased robustness, and the number of trials should be increased. Despite the variability, the results provide insights into the brush's ability to update the pixels. To manage overheating of the electromagnet and experiment run time, we limited the number of trials to 10. The heating of the electromagnet affects its magnetic strength. To partially overcome this problem, we provided breaks between different experimental conditions and used cooling fans for heat dissipation.

6.7 APPLICATIONS

We propose different applications of the flip dot display panels and how they can replace the existing display system. The proposed applications exploit the different aspects of our pixels which include the bi-stability of the visual element, simple installation since it does not require cables, and infrequent update of the displayed information.

Advertisement boards — Portable advertisement boards use characters and numeric symbols arranged on a board to show information. For example, ad boards placed outside a shop show the products on sale and their price. The symbols are usually limited to a specific language. Changing the advertised content requires a cumbersome process of searching through a repository of symbols and arranging them. These ad boards can be replaced with our flip dot display panels. The display panel can be brushed to show a new product on sale which might need to be updated occasionally, either daily or weekly.

Outdoor LED billboards and large-sized vinyl banners installed on roads show ads. LED displays require complex wiring and a power supply to show a static image. Vinyl banners require reprinting a new banner for every new ad and involve a cumbersome process of mounting them. The billboards and vinyl banners can be replaced with flip dot display panels. These panels do not require power to show a static image and can be brushed with new information which eliminates the need to reprint a new banner for every new ad. The information can be updated using a brush attached to a drone, which reduces the cumbersome process of manually mounting a banner.

Media Facade — The exterior of a building integrated with display panels forms a media facade. The display panels can cover the entire building or only a segment of it. The building exterior can show information related to the building itself or the neighbouring area or show ads. For example, a college building can show information like the name of the college, courses offered, university ranking, or the tuition fees for different programs offered in that specific building. When display panels attach to the exterior of a mall or a commercial building, they can show the list of shops/services provided inside the building or show an ad.

Bus display — A display panel attached to the exterior of the bus can be programmed to show a social message, weather forecast, or ads. When the bus is parked in the overnight shelter, the pixels are updated to show the required information. The shelter is equipped with a motorized 2D gantry mechanism that moves along the X and Y axes. The brush mounted on the gantry system moves the brush and updates the pixels on the bus to program the required information. Since the panels are unpowered and do not require wiring, it is flexible to install them on the bus and accommodate existing infrastructure (e.g. windows and doors).

Displays integrated into the architectural surrounding — A display panel attached to a wall of an indoor room can show a low-resolution photo. Display panels attached to the outside of a shop can show operating hours and update it when required during the holiday season. The display panels integrated as part of the floor can show a welcome message or customize floor designs or patterns.

6.8 LIMITATIONS AND FUTURE WORK

We discuss the limitations of our display system and potential solutions to overcome them and propose future work.

Pixel Size — The size of the pixel can be adjusted based on the application requirement. The current version of the pixel is 13mm in diameter (9mm NFC tag + 3mm magnet). By using smaller NFC tags (5mmx5mm) or larger NFC tags, other pixel sizes can be achieved. However, for rotating a larger pixel disc, a stronger electromagnet would be needed to create a more powerful magnetic field.

Brush to update multiple pixels simultaneously — The current brush prototype uses a combination of a single NFC antenna and electromagnet to detect and update the pixels. However, by arranging an array of NFC antennas and electromagnets in a grid on the brush, it will be possible to detect and update multiple pixels simultaneously, which will significantly improve the overall updating process and make it faster.

Robotic brushing to update large display surfaces — The human brushing action can be automated using a wheeled robot or drone. By attaching the automatic pixel state brush to the bottom of a wheeled robot, the brushing process can be automated. The robot could then move over a floor installed with display panels to update the pixels. The spatial registration method and pixel updating methods will still remain the same, except the wheeled robot replaces the human brushing action.

The handheld brush is limited to updating displays installed in locations accessible by a human and may require more time to update large display surfaces. We require alternative methods to update the display panels installed in inaccessible locations. For example, displays installed on a building exterior or roadside billboard could be updated using a drone mounted with the brush. These robotic solutions will enable efficient updates for large display panels and remote locations.

Viewing information in the dark — The flip dot pixels use reflective display technology, which means they require a light source to view the information. To enable the operation of these displays in the dark, one side of the pixel could be printed with glow-in-the-dark material, while the other side remains black. The glow-in-the-dark material stores energy during day time and this stored energy is used to shine in the dark.

.

6.9 CONCLUSION

We described the design of a mechanical pixel element that can be integrated into construction material to form a display panel. These display panels are flexible and easy to install as they do not require wiring or technical expertise. To enable these display panels, we show the hardware design of a 3D-printed flip dot pixel that can be controlled using an electromagnet brush. When the brush moves over the display, it detects the pixels and triggers the electromagnet to set the pixel's colour state. We conducted technical experiments on the brush's ability to update pixels at different speeds, angles, and offsets. We propose ways to replace existing display solutions with our system.

7

DISCUSSION AND FUTURE WORK

7.1 DISCUSSION AND LIMITATIONS

In this section, we discuss the technical challenges and hardware limitations of the physical pixel design, ways to make them more interactive, integrate them into construction materials, and understand their potential applications in live stage performances or as lighting installations.

7.1.1 *Technical Challenges & Hardware Limitations*

An electrical physical pixel contains an electrical energy source to power the pixel circuitry, a display element, control circuitry, and a communication link connecting it to the base station. The selection of these components depends on the specific usage scenarios of the pixels, which, in turn, determines the overall size of the pixel. We discuss the challenges in choosing the appropriate display element and energy source, and also the difficulties in scaling up the pixel count to form large display surfaces.

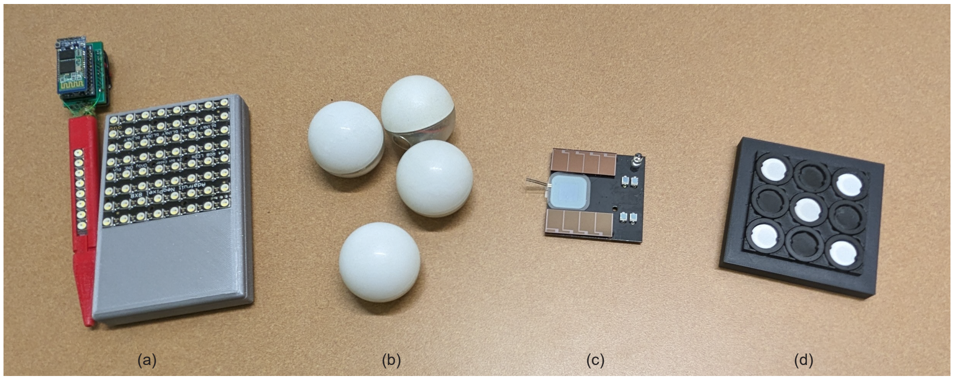


Figure 7.1: Display designs discussed in this thesis: (a) PocketView prototype for phone and pen form factors; (b) Scatterpixels wireless pixel elements; (c) Pixelboard solar-powered pixel element; (d) Pixelbrush flip dot pixel arranged in a 3x3 grid.

Display Element — Each pixel has a monochrome display element to show binary visual states. LEDs are easily available, simple to control, and cheap for use as a display element. They require a continuous power supply to maintain their visual state, making them suitable for pixels with sufficient energy supply. But, for pixels that harvest power from ambient sources (e.g. sunlight, RF signals), the available energy is limited and requires efficient management of the resources. Typically, energy harvesting displays use bi-stable display elements that only need power to update but not to hold their state.

E-ink and flip dot are common technologies used as fully bi-stable display elements. However, commercially available e-ink displays are high-resolution screens not suitable for our pixel design as they require a complex interface with a microcontroller for updates. Additionally, custom e-ink displays are not easily obtainable from the market, and the raw materials required to build them are not readily available due to intellectual property protection. To address this problem, Ollie et al. [37] extracted e-ink film from old e-readers to create custom e-ink displays. However, the extraction process is complex, unreliable, and not suitable for mass manufacturing of displays. We initially attempted to extract the e-ink film from a Kindle reader to make a custom one-bit e-ink display element. Although we achieved some initial success, the display was not reliable, and the process of adding electrical contact terminals proved challenging.

We initially intended to use e-ink technology for the pixelboard's display element. However, due to the unavailability of these displays, we opted for a semi bi-stable electrochromic display, which experiences a drop in contrast over time. As display technology continues to advance, we envision the possibility of simpler solutions for electronically controlled bi-stable displays. Flip dots are fully bi-stable mechanical pixels that contain a disc that rotates when an electromagnet is triggered. However, flip dots require a huge amount of power to trigger the electromagnet, making them unsuitable for battery-powered or energy-harvesting displays. So, in Pixelbrush, we designed a simpler version of a flip dot by moving the electromagnet to a movable brush. When the brush moves over a flip dot, the electromagnet is triggered setting the visual state of the pixel. This method decouples the electromagnet from the rotating disc, reducing the pixel size and power required to update a display panel made with flip dots, and eliminating the need for wiring between the pixels and a central control unit.

Energy Source — The electrical energy source powers the pixel circuitry, which could be a battery or a supercapacitor charged from an energy-harvesting device (e.g. solar panel).

Energy can be harvested from various ambient sources, such as light or radio waves. Solar panels harvest energy from light radiations and have the highest energy density compared to other energy harvesting technologies. So, our Pixelboard elements used two solar panels to harvest energy and store it in a supercapacitor. However, the use of solar panels increased the overall size of the physical pixel large and reduced the area available for the display element. To address this challenge, transparent solar panels can be directly stacked on top of a display. For example, Yogesh et al. [82] used transparent solar panels to harvest energy and power display screens placed directly underneath them. Unfortunately, small form factors of transparent solar panels are not readily available in the market and their custom fabrication requires specialized equipment and intensive engineering processes.

Initially, we also explored designing pixels that use RFID signals to both control and power an LED. In an RFID system, a high-power reader antenna communicates with passive tags. A passive tag contains an RFID chip and does not have a battery. The tags harvest energy from the RF signals sent by the reader antenna to power the chip. Once the chip receives sufficient power, it can communicate by either reflecting or not reflecting the signals from the reader, commonly used for identification purposes. We experimented with an evaluation board for a Rocky 100 chip[1], which can harvest enough energy to both power and control an LED. The reader antenna sends signals to program the register on the Rocky 100 chip to either turn on or off. After initial experimentation with the evaluation board, we designed our custom pixel element, a circular PCB (Figure 7.2a bottom), to house the Rocky 100 chip. We replaced the PCB trace antenna in the evaluation module with simple copper wires. These pixel elements can be mounted on different surfaces to form a display. For example, we attached three pixel elements to a coffee cup (Figure 7.2b) which can be controlled to show the progression of some physical quantity. Two pixels turn on (Figure 7.2c) to indicate the time left before the next meeting.

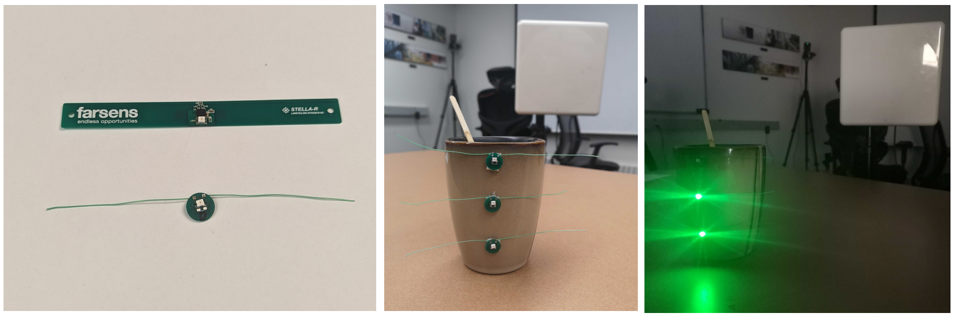


Figure 7.2: RFID Pixel Elements: a) Stellar Evaluation module (top) and custom PCB for the Rocky 100 chip (bottom); b) Pixel elements installed on a coffee cup; c) Two pixels are turned on to indicate the progression of time

The pixel elements were not able to reliably harvest energy to power the LED. The power harvesting and communication range of the pixel were affected by the antenna's performance and interference when two pixels were placed nearby. So, just using simple copper wires as the antenna might have reduced the performance. To reliably communicate and harvest power, the pixels need to be close to the RFID reader antenna. Additionally, the unpredictable write times on the Rocky 100 chip made these pixels unsuitable for high frame rates.

To completely overcome the issue related to electrical energy management, Pixelbrush explored mechanical pixel elements that do not require electrical energy to update their visual state. These pixels are very simple and compact since the mechanism to update is moved to an external device. The minimal design of the pixel comes at the cost of needing to manually brush the pixels to update the display with new information.

---

[1] http://www.rtutech.com/28web/Upload/20181225105747967.pdf

Scalability — To build large display surfaces and enable different applications proposed in this thesis, we needed to build a large number of physical pixels. The PCB design, mechanical fabrication, and assembling of the physical pixels are tedious, expensive, time-consuming, and require intensive manual labour. For Scatterpixels, we designed a custom PCB for the pixel circuitry and fabricated it at a PCB manufacturing facility. The majority of the electronic components were soldered at the fabrication facility itself. But, we assembled the individual pieces in the lab to build the 70 pixels. For each pixel, the microcontroller on the PCB is flashed with an Arduino bootloader firmware and then programmed with a unique ID. The assembly process includes painting the plastic sphere and adding board pins for charging contacts, soldering the battery, RF module, and LED to the PCB, and then putting these components into the plastic sphere.

Pixelbrush uses 3D-printed flip dot pixels, which contain a circular disc with a different colour on each side. The discs are printed as two halves, one in white and the other in black. An NFC chip and a tiny magnet are placed between the discs and glued together. The black-sided disc is printed with axles that need to be sanded, and more sanding is required along the disc edge after gluing them together. Due to the intense manual labour required to fabricate and assemble the pixels, we limited our exploration to small-scale displays. In the future, the pixels can be fabricated in a manufacturing facility to produce them on a large scale.

7.1.2 *Expressive Pixels*

The physical pixels discussed in this thesis are minimal in terms of display element colour. Currently, they are monochrome, but they could be made more expressive using display elements that support RGB colours. For example, the red LED in Scatterpixels can be replaced with an RGB LED or can exploit the RGB capabilities of the LED matrix in PocketView. Using richer colour schemes for the display element can help in creating more aesthetic user experiences and convey information more easily.

7.1.3 *Interactive Pixels*

The display systems discussed in this thesis use a host device (phone or desktop computer) to program the visual state of the pixels. The base station sends signals to the physical pixel to reflect the visual states programmed on the host device. However, it would be more interesting to explore alternative ways of interacting with the pixels directly. For example, Sato et al. [109] designed pixels with a built-in accelerometer to interact directly by tapping on them. To make future pixels even more interactive, additional input sensing capabilities like touch input, mid-air gestures, and speech input could be incorporated to enable direct interactions with the pixel.

7.1.4 *Integration into different construction materials*

The pixel elements designed in the Pixelboard and PixelBrush system are intended for inclusion into construction material (e.g. wood) to form a display panel. But, we have not explored the detailed process of building these panels because of the pixel size and scalability issues discussed previously. Prior work has experimented with integrating displays into construction materials. For example, Alex et al. [94] used PMOLED displays to shine information through construction material (e.g. veneer, glass) and incorporated touch input for interaction. Similarly, Living Wood Technology [58] adds an LED matrix to the backside of a veneer plank and enables input interaction through capacitive sensing. These displays seamlessly integrate digital information into construction materials but still require wires to power and communicate with the pixels.

To overcome these wiring constraints, we explored using independent pixel elements to form display panels that provide flexibility in installation. Although our pixel elements are relatively large and mainly support low-resolution displays, we take a step forward to enable the design of such display panels. We imagine a future where independent pixels are integrated into the construction material during the fabrication stage. This will enable architects and designers to plan construction with integrated display surfaces and allows construction workers to easily customize and install them without requiring technical expertise.

7.1.5 *Use in art installation and live perfomances*

We designed displays that are mainly intended to show information through images, text, or numbers. But, low-resolution displays have been used in art installations and in live stage performances to enhance the audience experience. Our independent physical pixels can also enable these applications.

Jim Campbell [10] designed custom lighting solutions for art installations and public buildings using low-resolution LED matrices, projectors, and digital screens. He implicity explores human perception and expressiveness of low-res displays by mapping them with high-resolution content. The lighting choreographer [27] is a system to show lighting patterns on a performer during a dance performance. It uses wired LED strips powered by a battery and controlled through wireless communication to show different lighting patterns. The performer or choreographer can program lighting patterns, that synchronize with the background music of the performance. The wired LED strips can be replaced with scatterpixels, which offer more flexibility in attaching pixels to different locations on the body without any wiring restrictions.

The operation time of Scatterpixel (5-8 hrs) makes it ideal for live stage performances and has fast enough refresh rates to update the pixels in real time. Scatterpixels can also be placed on the floor in a live theatre to show animations or lighting patterns. These lighting patterns can be synchronized with the screenplay of the performance.

7.2 FUTURE WORK

We propose future directions to explore the space of customizable displays, discuss their usage scenarios, and provide an initial idea of their technical implementation. We also propose future studies required to understand how people perceive and render information on irregularly shaped displays and explore the extended use cases of customizable displays.

7.2.1 *High-Resolution Screen as a Display Element*

The thesis explores physical pixels using one-bit monochrome displays, representing binary visual states. To create higher-resolution customizable displays, the one-bit display element could be replaced with high-resolution screens to form pixel tiles (Figure 7.3a). Such pixel tiles could enable higher-resolution customizable displays compared to the solutions discussed in this thesis, but this comes at the cost of reduced physical reconfigurability. Siftables [84] are battery-powered pixel tiles with colour LCD screen, RF radio, and IR modules to communicate between neighbouring tiles. These tiles do not operate together as a single coordinated display to show information and require frequent recharging for continuous operation.

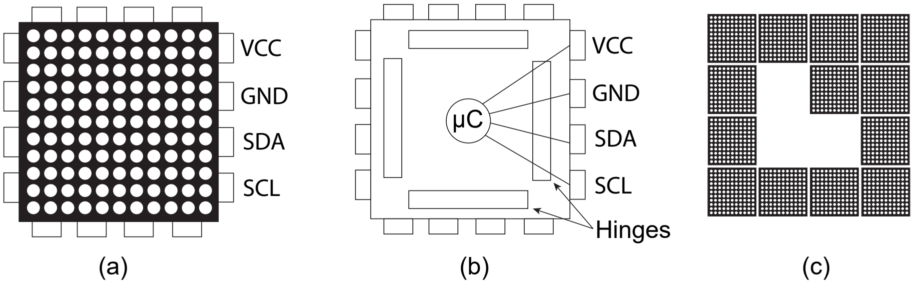


Figure 7.3: Design concept for tiled display screens: (a) front view of a pixel tile using a LED matrix; (b) Back side view of pixel tile with a microcontroller; (c) pixel tiles arranged to form a arbitrary 2D display configuration.

To avoid the need for individual recharging, each tile can interlock with neighbouring tiles using hinges and share a common power and communication link. The tile has a microcontroller that connects to an I2C bus for communication with a base station (Figure 7.3b). Multiple tiles can be interlocked to build 2D or 3D display structures of the desired shape and size (Figure 7.3c). Once the tiles are interlocked in the desired configuration, one tile connects to a base station to receive power and control signals for updating all the tiles with the desired information. The base station receives information from a host device (e.g. phone) to be displayed on the tiles. A one-time spatial registration method finds the locations of the tiles in 3D space by tracking a visual code on each tile. The spatial registration provides a spatial map that allows the tiles to operate together as a single coordinated display.

Based on the use-case of the displays, the pixels can use different display elements, such as passive displays (e.g. E-ink display screen) or active displays (e.g. LED or LCD screen). Active display screens are suitable for content that changes frequently, making them ideal for showing dynamic information. On the other hand, passive displays are better suited for content that changes occasionally. For pixel tiles with LCD or LED screens, the base station receives power from a wall outlet and routes it to the interconnected tiles. They can show dynamic information like email notifications, meeting schedules, or weather updates. For pixel tiles with e-ink screens, they can also be powered by connecting the base station to the wall outlet. Alternatively, when e-ink tiles are used, the base station could be equipped with an NFC system to harvest energy and receive information through NFC signals. When an NFC-compatible phone (host device) comes near the base station, the e-ink tiles are updated with new information. This method of updating tiles will take more time but allows easier installation as it eliminates the need for wires to a wall outlet.

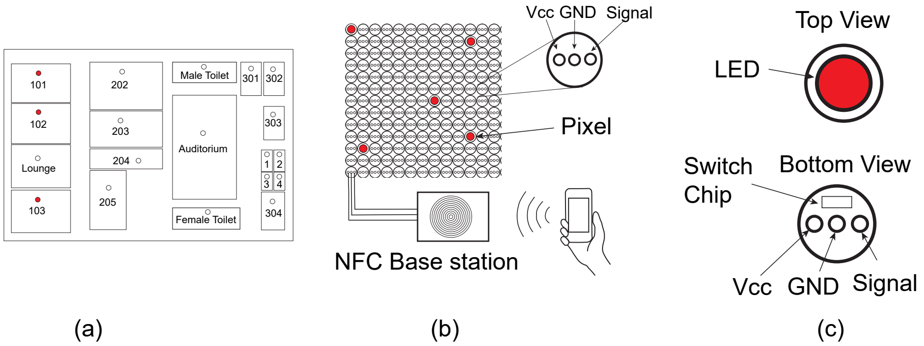
As an example workflow, these tiles can serve as a digital photo frame on a desk, then be customized to a fridge display to show the weekly schedule, and later transform into a large low-resolution TV screen.

7.2.2 *Adding visual interactivity to printed documents*

A printed document shows static information; for example, an indoor map printed on paper shows the locations of rooms and washrooms in a building. Pixels can be integrated into these documents to make them interactive. For example, an indoor map embedded with pixels can highlight a route to a destination room from your current location, or highlight selected rooms (Figure 7.4a). A notice board can light up pixels to indicate upcoming events or highlight events of a specific type (e.g. sports, entertainment, or academic workshops).

To achieve this goal, a display substrate inserted with pixel elements is attached to the back of a printed document. The display substrate is a 3-layered flexible PCB with layers for power, ground, and communication (Figure 7.4b). The substrate is cut to match the shape and size of the physical document. A pixel has a display element and an addressable switch (Figure 7.4c). When the pixels are inserted into the display substrate, they receive power and communicate with the base station through a one-wire interface protocol.

Pixels are inserted only at the locations of interest. For example, in a bus route poster, the pixels are inserted only at bus stations, and the remaining area has no pixels. This reduces the overall power consumption of the display by minimizing the pixels required to show information. After inserting the required pixels, the display substrate is connected to an NFC base station



(a) (b) (c)

Figure 7.4: Design concept to add visual interactivity to printed documents: (a) an indoor map poster is attached with pixels to the backside; (b) display substrate connected to an NFC base station mounted with pixels at the location of interest; (c) Top View and Bottom View of the pixel element

to harvest energy and receive information to control the pixels from an NFC-compatible phone.

To interact with the document, a user taps their phone on the base station to launch an app. The app shows different ways to search for information on the document. For example, enter a room number, search for male washrooms, or use voice to input the query. After inputting the query, the user again taps on the base station to light up the corresponding pixels to show the output of the query. For example, pixels light up to show the room number of interest or highlight the route to reach it. The display requires power only for a short time period when a user interacts with the document and is unpowered otherwise.

7.2.3 *Manually Updatable Display*

In a traditional display system, the pixels are automatically updated without human intervention. However, in certain scenarios where frequent updates are not necessary, a manual approach can be used by leveraging human intervention. Pixelbrush demonstrated this concept by using a movable electromagnetic brush to manually update mechanical pixels. The manual approach can also be extended to electrically powered pixels to minimize energy usage.

In the current implementation of Scatterpixels, we used pixels that remain constantly powered to operate a microcontroller, a radio module, and an LED through a battery. These pixels can be arranged to form a display and controlled wirelessly to show animations. However, for cases where occasional updates are sufficient and show static content, the pixel design can be modified to optimize energy usage in each pixel.

To modify the design of Scatterpixels for manually updatable displays (Figure 7.5a), a reed switch is introduced between the battery and the remaining pixel circuitry, and replaces the LED with a bi-stable display element (E-ink). The reed switch is a contact switch that closes a circuit when a magnetic field

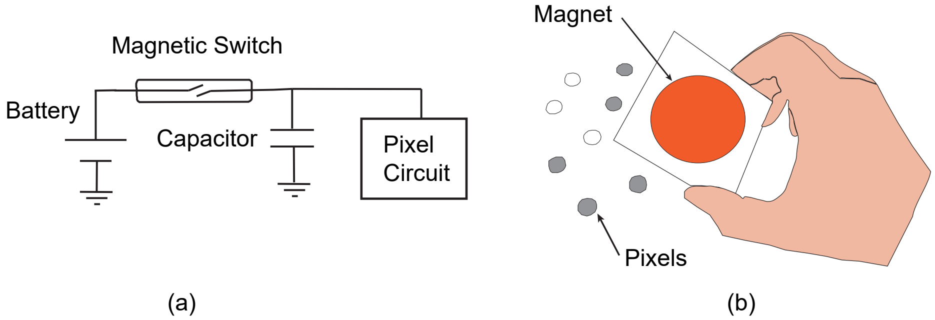


Figure 7.5: Design concept for manually updatable displays: (a) modified pixel circuitry for scatterpixels to enable manually updatable displays; (b) a magnetic handheld brush moves over the pixels to update it.

is present. A handheld device (Figure 7.5b) equipped with a large permanent magnet moves over the pixels closing the reed switch to quickly charge a capacitor. The charged capacitor powers the pixel circuitry for a short duration during which it receives radio signals from the base station and updates the bi-stable display element. The bi-stable display retains its visual state even after the capacitor discharges. The manual process of updating the pixels can significantly increase the battery life of self-powered pixels when the display needs to show static information and only requires occasional updates.

7.2.4 *Wearable Customizable Pixel Displays*

PocketView uses tethered pixels to build displays of different form factors to fit a wide range of pockets (Figure 7.1a). However, the pixels are limited to the pocket area, so it would be interesting to explore adding pixels to other parts of the body. For example, small pixels can attach to personal accessories, like a shirt, shoes, or hat (Figure 7.6), to form a display. The shapes, sizes, resolutions, and locations of these displays can be customized to different parts of the body. The displays would show dynamic information like fitness progress while running, navigation instructions, meeting reminders, or make a fashion statement. Additionally, these pixels would be conveniently placed on clothing only when needed.

Each pixel would contain a magnet, a bi-stable display or LED as a display element, and NFC IC for energy harvesting and communication. An NFC antenna array stitched on the inside of the accessory using conductive threads can provide power and communication, and also locates the pixels. For the system to work with existing clothing, an inner garment stitched with the antenna array can be worn inside. The NFC antenna array is connected to a base station, which can selectively power the individual antennas in the array to spatially register and control the pixels. The base station can be powered by a battery when an LED is used as a display element. Instead of stitching antennas into the clothing, we can design the pixels with a bi-stable display

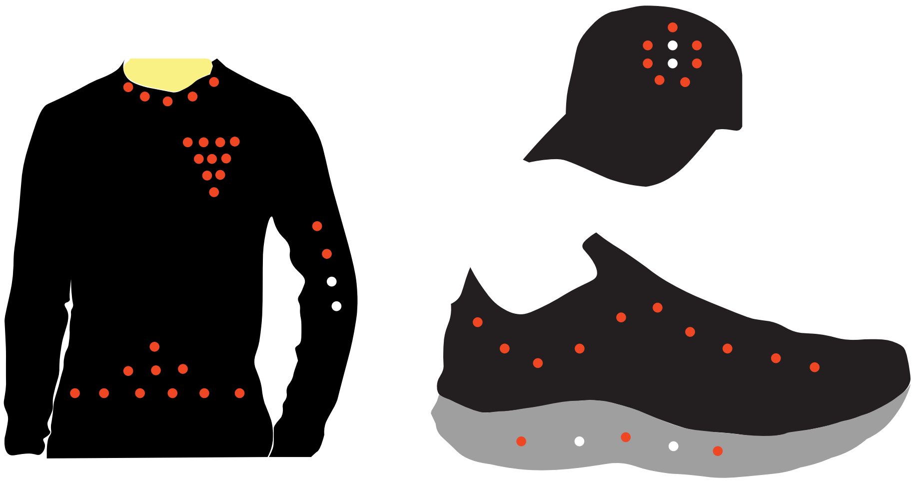


Figure 7.6: Design concept for wearable customizable pixels for shirts, shoes, and cap

element that can be updated manually by brushing a phone over them. Each pixel directly receives power and control signals from a phone through NFC signals, similar to Alterwear [20] but with a simpler communication protocol to control a one-bit display.

7.2.5 *Perceive Information, Render Content, and Extended use cases of Customizable Displays*

Customizable displays may be sparse, irregularly spaced and shaped, and low resolution, so it is critical to find optimal ways to show content in a way that can be easily understood and is graphically sound. The Gestalt principle of "closure" states that humans can compensate for missing pixels in cyclic or recognizable shapes and images [108]. The human eye's spatial resolution determines the distance at which pixels are perceptible, for a given pixel size. Persistence of vision states that humans perceive a light source to be continually present when it is flickered above a certain threshold rate [43]. Using these principles, Sato et al. [108] built optimization algorithms to render content on irregularly shaped displays. A user study was conducted to understand the effectiveness of optimization and found it useful.

Superpixelator [53] rasterizes vector graphics into low-resolution images and reduces artifacts like blips, missing pixels, and extra pixels. Similarly, Timothy et al. [28] also map high-resolution images to pixel art-type low-res images. Prior work on visual perception and rendering has been predominantly studied for displays with constant pixel densities. Future work could systematically explore visual perception and content rendering for customizable displays, that support arbitrary shapes, sizes, and pixel densities.

To run these studies without any technical challenges and hardware limitations, we could create computer simulations to mimic different configurations of displays. The different configurations can be generated by varying the size, shape, resolution, frame rate, active viewing area of the pixel, and colour of the pixel elements. The simulations can easily generate different 2D and 3D display configurations that are rendered on a screen, web page, AR, or VR headset. A study using these simulations would increase our understanding of human perception with these displays, and establish the usefulness and limits of customizable displays. The user study can learn the characteristics and constraints of human perception in this setting and use it to develop models that can predict optimal rendering solutions for arbitrary display configurations. The trained models could suggest optimal transformations or filters that consider human perception.

Other user studies could further explore the design space and applications of customizable displays. These studies could qualitatively examine the use case of customizable displays through workshops and interviews with designers and architects.

8 CONCLUSION

8.1 CONCLUSION

In this thesis, we explore a design space of pixel independence for customizable displays for flexible and easy integration of pixels into the existing infrastructure. We summarize different displays design discussed in the thesis and provide a final word on our ultimate vision.

Summary

We investigated different displays that integrate pixels into the surroundings that include the human body, wall, floor, and ceilings.

In chapter 3, we show how to integrate pixels into pockets in clothing to shine information through it. We used bright LED matrices of different shapes and sizes to shine information through a wide range of pockets.

In Chapter 4, we explored the concept of ad hoc reconfigurable displays that use wireless physical pixels to instantly create displays on a desired surface. These pixels enable different configurations ranging from one-bit displays to 2D displays on a floor, ceiling, or walls.

In Chapter 5, we describe the design of an independent solar-powered pixel element that harvests energy from light to drive the pixel circuitry. A laser base station sends frequency-modulated signals to control an electrochromic display element on the pixel.

In Chapter 6, we describe the design of a mechanical flip dot pixel that can be controlled using a handheld electromagnetic brush. This pixel is suitable for inclusion in displays that need to be updated occasionally.

Final Word

Displays have been commonly used to visualize digital information from devices and gadgets. Traditionally, a display contains a set of pixels precisely arranged within a confined area to maximize fidelity. The pixels are wired to a central control unit to receive power and control signals to update them with the required information. In this thesis, we explored ways to move pixels from a screen and integrate them into the surrounding environment to form ubiquitous displays. To achieve this goal, we explored the space of *customizable displays* that provides flexibility in creating different display layouts and is easy to install. We show display designs enabled by designing and combining pixels to suit different application needs. The pixels discussed in this thesis are relatively big and suitable for forming low-resolution displays, but with advancements in technology, we hope pixels can be made smaller and more minimal. This will enable seamless integration of digital information into the physical world to achieve the goal of pixels being truly ubiquitous and not only constrained within the physical limits of a dedicated computing device.

BIBLIOGRAPHY

[1] Ali Abedi, Mohammad Hossein Mazaheri, Omid Abari, and Tim Brecht. "WiTAG: Rethinking Backscatter Communication for WiFi Networks." In: *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*. HotNets '18. Redmond, WA, USA: Association for Computing Machinery, 2018, 148–154. ISBN: 9781450361200. DOI: 10.1145/3286062.3286084. URL: https://doi.org/10.1145/3286062.3286084.

[2] Karan Ahuja, Sujeath Pareddy, Robert Xiao, Mayank Goel, and Chris Harrison. "LightAnchors: Appropriating Point Lights for Spatially-Anchored Augmented Reality Interfaces." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 189–196. ISBN: 9781450368162. DOI: 10.1145/3332165.3347884. URL: https://doi.org/10.1145/3332165.3347884.

[3] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley. "Multi-robot system for artistic pattern formation." In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 4512–4517. DOI: 10.1109/ICRA.2011.5980269.

[4] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Roland Siegwart, and Paul Beardsley. "Image and animation display with multiple mobile robots." In: *The International Journal of Robotics Research* 31.6 (2012), pp. 753–773. DOI: 10.1177/0278364912442095. eprint: https://doi.org/10.1177/0278364912442095. URL: https://doi.org/10.1177/0278364912442095.

[5] Orkhan Amiraslanov, Paul Lukowicz, and Jingyuan Cheng. "Large Scale, Flexible Electroluminescent Display." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. UbiComp '16. Heidelberg, Germany: Association for Computing Machinery, 2016, 245–248. ISBN: 9781450344623. DOI: 10.1145/2968219.2971372. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2968219.2971372.

[6] David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. "Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modeling." In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., 2000, 393–402. ISBN: 1581132085. DOI: 10.1145/344779.344960. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/344779.344960.

[7] Moritz Behrens, Nina Valkanova, Ava Fatah gen. Schieck, and Duncan P. Brumby. “Smart Citizen Sentiment Dashboard: A Case Study Into Media Architectural Interfaces.” In: *Proceedings of The International Symposium on Pervasive Displays*. PerDis ’14. Copenhagen, Denmark: Association for Computing Machinery, 2014, 19–24. ISBN: 9781450329521. DOI: `10.1145/2611009.2611036`. URL: `https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2611009.2611036`.

[8] Oliver Bimber and Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005.

[9] Philipp Brauner et al. “Towards Accepted Smart Interactive Textiles.” In: *HCI in Business, Government and Organizations. Interacting with Information Systems*. Ed. by Fiona Fui-Hoon Nah and Chuan-Hoo Tan. Cham: Springer International Publishing, 2017, pp. 279–298. ISBN: 978-3-319-58481-2.

[10] Jim Campbell. “http://www.jimcampbell.tv/portfolio/installations/tilted$_{plane}/$.” In.

[11] Angie Chandler, Joe Finney, Carl Lewis, and Alan Dix. “Toward Emergent Technology for Blended Public Displays.” In: *Proceedings of the 11th International Conference on Ubiquitous Computing*. UbiComp ’09. Orlando, Florida, USA: Association for Computing Machinery, 2009, 101–104. ISBN: 9781605584317. DOI: `10.1145/1620545.1620562`. URL: `https://doi.org/10.1145/1620545.1620562`.

[12] Amutha Chinnappan, Chinnappan Baskar, Shikha Baskar, Greeshma Ratheesh, and Seeram Ramakrishna. “An overview of electrospun nanofibers and their application in energy storage, sensors and wearable/flexible electronics.” In: *Journal of Materials Chemistry C* 5.48 (2017), pp. 12657–12673.

[13] Marcelo Coelho, Jamie Zigelbaum, and Joshua Kopin. “Six-Forty by Four-Eighty: The Post-Industrial Design of Computational Materials.” In: *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’11. Funchal, Portugal: Association for Computing Machinery, 2010, 253–256. ISBN: 9781450304788. DOI: `10.1145/1935701.1935752`. URL: `https://doi.org/10.1145/1935701.1935752`.

[14] Ashley Colley, Minna Pakanen, Saara Koskinen, Kirsi Mikkonen, and Jonna Häkkilä. “Smart Handbag as a Wearable Public Display - Exploring Concepts and User Perceptions.” In: *Proceedings of the 7th Augmented Human International Conference 2016*. AH ’16. Geneva, Switzerland: Association for Computing Machinery, 2016. ISBN: 9781450336802. DOI: `10.1145/2875194.2875212`. URL: `https://doi.org/10.1145/2875194.2875212`.

[15] Ashley Colley, Paweł W. Woundefinedniak, Francisco Kiss, and Jonna Häkkilä. “Shoe Integrated Displays: A Prototype Sports Shoe Display and Design Space.” In: *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. NordiCHI ’18. Oslo, Norway: Association for Computing Machinery, 2018, 39–46. ISBN: 9781450364379. DOI: `10.1145/3240167.3240216`. URL: `https://doi.org/10.1145/3240167.3240216`.

[16] Ashley Colley, Paweł W. Woundefinedniak, Francisco Kiss, and Jonna Häkkilä. “Shoe Integrated Displays: A Prototype Sports Shoe Display and Design Space.” In: *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. NordiCHI ’18. Oslo, Norway: Association for Computing Machinery, 2018, 39–46. ISBN: 9781450364379. DOI: `10.1145/3240167.3240216`. URL: `https://doi.org/10.1145/3240167.3240216`.

[17] Cool-Mania. “https://www.cool-mania.com/party/geek-t-shirts/led-message-t-shirt-with-programmable-display.” In.

[18] Laura Devendorf, Joanne Lo, Noura Howell, Jung Lin Lee, Nan-Wei Gong, M Emre Karagozler, Shiho Fukuhara, Ivan Poupyrev, Eric Paulos, and Kimiko Ryokai. “I don’t want to wear a screen: probing perceptions of and possibilities for dynamic displays on clothing.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pp. 6028–6039.

[19] Jan Diehm and Amber Thomas. “Women’s Pockets are Inferior.” In: *The Pudding* (2018). URL: `https://pudding.cool/2018/08/pockets/`.

[20] Christine Dierk, Molly Jane Pearce Nicholas, and Eric Paulos. “AlterWear: Battery-Free Wearable Displays for Opportunistic Interactions.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–11. ISBN: 9781450356206. DOI: `10.1145/3173574.3173794`. URL: `https://doi.org/10.1145/3173574.3173794`.

[21] Christine Dierk, Molly Jane Pearce Nicholas, and Eric Paulos. “AlterWear: battery-free wearable displays for opportunistic interactions.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 220.

[22] Christine Dierk, Tomás Vega Gálvez, and Eric Paulos. “AlterNail: Ambient, Batteryless, Stateful, Dynamic Displays at Your Fingertips.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 6754–6759. ISBN: 9781450346559. DOI: `10.1145/3025453.3025924`. URL: `https://doi.org/10.1145/3025453.3025924`.

[23] Christine Dierk, Tomás Vega Gálvez, and Eric Paulos. “Alternail: Ambient, batteryless, stateful, dynamic displays at your fingertips.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 6754–6759.

[24] David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. "PocketThumb: a Wearable Dual-Sided Touch Interface for Cursor-based Control of Smart-Eyewear." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.2 (2017), p. 9.

[25] Farsens. "http://www.farsens.com/en/." In.

[26] Sam Ferguson, Anthony Rowe, Oliver Bown, Liam Birtles, and Chris Bennewith. "Networked Pixels: Strategies for Building Visual and Auditory Images with Distributed Independent Devices." In: *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. Camp;C '17. Singapore, Singapore: Association for Computing Machinery, 2017, 299–308. ISBN: 9781450344036. DOI: 10.1145/3059454.3059480. URL: https://doi.org/10.1145/3059454.3059480.

[27] Minoru Fujimoto, Fujita Naotaka, Tsutomu Terada, and Masahiko Tsukamoto. "Lighting Choreographer: An LED Control System for Dance Performances." In: *Proceedings of the 13th International Conference on Ubiquitous Computing*. UbiComp '11. Beijing, China: Association for Computing Machinery, 2011, 613–614. ISBN: 9781450306300. DOI: 10.1145/2030112.2030240. URL: https://doi.org/10.1145/2030112.2030240.

[28] Timothy Gerstner, Doug DeCarlo, Marc Alexa, Adam Finkelstein, Yotam Gingold, and Andrew Nealen. "Pixelated Image Abstraction." In: *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. NPAR '12. Annecy, France: Eurographics Association, 2012, 29–36. ISBN: 9783905673906.

[29] Alix Goguey et al. "PickCells: A Physically Reconfigurable Cell-Composed Touchscreen." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–14. ISBN: 9781450359702. DOI: 10.1145/3290605.3300503. URL: https://doi.org/10.1145/3290605.3300503.

[30] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. "BitDrones: Towards Using 3D Nanocopter Displays As Interactive Self-Levitating Programmable Matter." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 770–780. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858519. URL: http://doi.acm.org/10.1145/2858036.2858519.

[31] Matthew G. Gorbet, Maggie Orth, and Hiroshi Ishii. "Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., 1998, 49–56. ISBN: 0201309874. DOI: 10.1145/274644.274652. URL: https://doi.org/10.1145/274644.274652.

[32] Tobias Grosse-Puppendahl, Oskar Bechtold, Lukas Strassel, David Jakob, Andreas Braun, and Arjan Kuijper. "Enhancing Traffic Safety with Wearable Low-Resolution Displays." In: *Proceedings of the 2nd International Workshop on Sensor-Based Activity Recognition and Interaction*. iWOAR '15. Rostock, Germany: Association for Computing Machinery, 2015. ISBN: 9781450334549. DOI: 10.1145/2790044.2790059. URL: https://doi.org/10.1145/2790044.2790059.

[33] Tobias Grosse-Puppendahl, Steve Hodges, Nicholas Chen, John Helmes, Stuart Taylor, James Scott, Josh Fromm, and David Sweeney. "Exploring the Design Space for Energy-Harvesting Situated Displays." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, 41–48. ISBN: 9781450341899. DOI: 10.1145/2984511.2984513. URL: https://doi.org/10.1145/2984511.2984513.

[34] Tobias Grosse-Puppendahl, Steve Hodges, Nicholas Chen, John Helmes, Stuart Taylor, James Scott, Josh Fromm, and David Sweeney. "Exploring the Design Space for Energy-Harvesting Situated Displays." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, 41–48. ISBN: 9781450341899. DOI: 10.1145/2984511.2984513. URL: https://doi.org/10.1145/2984511.2984513.

[35] Noman Haleem, Sayed Ibrahim, Tanveer Hussain, Abdul Jabbar, Mumtaz Hassan Malik, and Zulfiqar Ali Malik. "Determining the light transmission of woven fabrics through different measurement methods and its correlation with air permeability." In: *Journal of Engineered Fibers and Fabrics* 9.4 (2014), p. 155892501400900409.

[36] Ollie Hanton, Zichao Shen, Mike Fraser, and Anne Roudaut. "FabricatINK: Personal Fabrication of Bespoke Displays Using Electronic Ink from Upcycled E Readers." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3501844. URL: https://doi.org/10.1145/3491102.3501844.

[37] Ollie Hanton, Zichao Shen, Mike Fraser, and Anne Roudaut. "FabricatINK: Personal Fabrication of Bespoke Displays Using Electronic Ink from Upcycled E Readers." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3501844. URL: https://doi.org/10.1145/3491102.3501844.

[38] Ollie Hanton, Michael Wessely, Stefanie Mueller, Mike Fraser, and Anne Roudaut. "ProtoSpray: Combining 3D Printing and Spraying to Create Interactive Displays with Arbitrary Shapes." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–13.

ISBN: 9781450367080. DOI: 10.1145/3313831.3376543. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3313831.3376543.

[39] Emmi Harjuniemi, Ashley Colley, Piia Rytilahti, Hong Li, Jesse Forest, and Jonna Häkkilä. "Idle stripes shirt: ambient wearable display for activity tracking." In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, pp. 254–259.

[40] Chris Harrison and Haakon Faste. "Implications of Location and Touch for On-Body Projected Interfaces." In: *Proceedings of the 2014 Conference on Designing Interactive Systems*. DIS '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, 543–552. ISBN: 9781450329026. DOI: 10.1145/2598510.2598587. URL: https://doi.org/10.1145/2598510.2598587.

[41] Chris Harrison, Brian Y. Lim, Aubrey Shick, and Scott E. Hudson. "Where to Locate Wearable Displays? Reaction Time Performance of Visual Alerts from Tip to Toe." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2009, 941–944. ISBN: 9781605582467. URL: https://doi.org/10.1145/1518701.1518845.

[42] Jeremy Hartmann, Yen-ting Yeh, and Daniel Vogel. "AAR: Augmenting a Wearable Augmented Reality Display with an Actuated Head-Mounted Projector." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. New York, New York, USA: ACM Press, 2020, pp. 1–14. ISBN: 9781450375146. DOI: 10.1145/3379337.3415849.

[43] Selig Hecht, Simon Shlaer, and Cornelis D. Verrijp. "INTERMITTENT STIMULATION BY LIGHT : II. THE MEASUREMENT OF CRITICAL FUSION FREQUENCY FOR THE HUMAN EYE." In: *Journal of General Physiology* 17.2 (Nov. 1933), pp. 237–249. ISSN: 0022-1295. DOI: 10.1085/jgp.17.2.237. eprint: https://rupress.org/jgp/article-pdf/17/2/237/1238128/237.pdf. URL: https://doi.org/10.1085/jgp.17.2.237.

[44] Luke Hespanhol, M. Hank Haeusler, Martin Tomitsch, and Gernot Tscherteu. *Media architecture compendium: digital placemaking*. eng. OCLC: 1000491846. Stuttgart: Avedition, 2017. ISBN: 978-3-89986-251-5.

[45] Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. "Phygital Field: An Integrated Field with a Swarm of Physical Robots and Digital Images." In: *SIGGRAPH ASIA 2016 Emerging Technologies*. SA '16. Macau: Association for Computing Machinery, 2016. ISBN: 9781450345392. DOI: 10.1145/2988240.2988242. URL: https://doi.org/10.1145/2988240.2988242.

[46] Marius Hoggenmueller, Alexander Wiethoff, Andrew Vande Moere, and Martin Tomitsch. "A Media Architecture Approach to Designing Shared Displays for Residential Internet-of-Things Devices." In: *Proceedings of the 4th Media Architecture Biennale Conference*. MAB18.

Beijing, China: Association for Computing Machinery, 2018, 106–117. ISBN: 9781450364782. DOI: 10.1145/3284389.3284391. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3284389.3284391.

[47] Marius Hoggenmüller and Alexander Wiethoff. "LightSet: Enabling Urban Prototyping of Interactive Media Façades." In: *Proceedings of the 2014 Conference on Designing Interactive Systems*. DIS '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, 925–934. ISBN: 9781450329026. DOI: 10.1145/2598510.2598551. URL: https://doi.org/10.1145/2598510.2598551.

[48] Cedric Honnet, Hannah Perner-Wilson, Marc Teyssier, Bruno Fruchard, Jürgen Steimle, Ana C Baptista, and Paul Strohmeier. "PolySense: Augmenting Textiles with Electrical Functionality using In-Situ Polymerization." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13.

[49] Meng-Ju Hsieh, Rong-Hao Liang, Da-Yuan Huang, Jheng-You Ke, and Bing-Yu Chen. "RFIBricks: Interactive Building Blocks Based on RFID." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–10. ISBN: 9781450356206. DOI: 10.1145/3173574.3173763. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3173574.3173763.

[50] Scott E Hudson, Chris Harrison, Beverly L Harrison, and Anthony LaMarca. "Whack gestures: inexact and inattentive interaction with mobile devices." In: *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*. ACM. 2010, pp. 109–112.

[51] F. Ichikawa, J. Chipchase, and R. Grignani. "Where's The Phone? A Study of Mobile Phone Location in Public Spaces." In: *2005 2nd Asia Pacific Conference on Mobile Technology, Applications and Systems*. 2005, pp. 1–8. DOI: 10.1109/MTAS.2005.243793.

[52] Manoj Kumar Imrith, Roshan Unmar, and Satyadev Rosunee. "Investigating the relationship between knitted fabric porosity and light permeability." In: *Indian Journal of Materials Science* 2016 (2016).

[53] Tiffany C. Inglis, Daniel Vogel, and Craig S. Kaplan. "Rasterizing and Antialiasing Vector Line Art in the Pixel Art Style." In: *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. NPAR '13. Anaheim, California: Association for Computing Machinery, 2013, 25–32. ISBN: 9781450321983. DOI: 10.1145/2486042.2486044. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2486042.2486044.

[54] "Intel Drone Light shows." In: URL: https://inteldronelightshows.com/.

[55] Walther Jensen, Ashley Colley, Jonna HÃ¤kkilÃ¤, Carlos Pinheiro, and Markus LÃ¶chtefeld. "TransPrint: A Method for Fabricating Flexible Transparent Free-Form Displays." In: *Advances in Human-Computer Interaction* 2019 (2019), p. 1340182. ISSN: 1687-5893. DOI: `10.1155/2019/1340182`. URL: `https://doi.org/10.1155/2019/1340182`.

[56] Walther Jensen, Ashley Colley, and Markus Löchtefeld. "VitaBoot: Footwear with Dynamic Graphical Patterning." In: *Proceedings of the 23rd International Symposium on Wearable Computers*. ISWC '19. London, United Kingdom: Association for Computing Machinery, 2019, 279–283. ISBN: 9781450368704. DOI: `10.1145/3341163.3346937`. URL: `https://doi.org/10.1145/3341163.3346937`.

[57] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. "RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units." In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, 637–644. ISBN: 9781450330695. DOI: `10.1145/2642918.2647383`. URL: `https://doi.org/10.1145/2642918.2647383`.

[58] Lasse Kaila, Henrik Raula, Miika Valtonen, and Karri Palovuori. "Living Wood: A Self-Hiding Calm User Interface." In: *Proceeding of the 16th International Academic MindTrek Conference*. MindTrek '12. Tampere, Finland: Association for Computing Machinery, 2012, 267–274. ISBN: 9781450316378. DOI: `10.1145/2393132.2393191`. URL: `https://doi.org/10.1145/2393132.2393191`.

[59] Thorsten Karrer, Moritz Wittenhagen, Leonhard Lichtschlag, Florian Heller, and Jan Borchers. "Pinstripe: eyes-free continuous input on interactive clothing." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 1313–1322.

[60] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. "MakerWear: A Tangible Approach to Interactive Wearable Creation for Children." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 133–145. ISBN: 9781450346559. DOI: `10.1145/3025453.3025887`. URL: `https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3025453.3025887`.

[61] Majeed Kazemitabaar, Leyla Norooz, Mona Leigh Guha, and Jon E. Froehlich. "MakerShoe: Towards a Wearable e-Textile Construction Kit to Support Creativity, Playful Making, and Self-Expression." In: *Proceedings of the 14th International Conference on Interaction Design and Children*. IDC '15. Boston, Massachusetts: Association for Computing Machinery, 2015, 449–452. ISBN: 9781450335904. DOI: `Makershoe`. URL: `https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2771839.2771883`.

[62] Sho Kimura, Ryo Oguchi, Hideo Tanida, Yasuaki Kakehi, Keita Takahashi, and Takeshi Naemura. "PVLC projector: image projection with imperceptible pixel-level metadata." In: Jan. 2008, p. 135. DOI: 10.1145/1400885.1401030.

[63] Brian T. Kirby, Michael Ashley-Rollman, and Seth Copen Goldstein. "Blinky Blocks: A Physical Ensemble Programming Platform." In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, 1111–1116. ISBN: 9781450302685. DOI: 10.1145/1979742.1979712. URL: https://doi.org/10.1145/1979742.1979712.

[64] Ryo Kishi. "http://ryokishi.org/works/orbs/orbs.html." In.

[65] Konstantin Klamka, Raimund Dachselt, and Jürgen Steimle. "Rapid Iron-On User Interfaces: Hands-on Fabrication of Interactive Textile Prototypes." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14.

[66] Pascal Knierim, Steffen Maurer, Katrin Wolf, and Markus Funk. "Quadcopter-Projected In-Situ Navigation Cues for Improved Location Awareness." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–6. ISBN: 9781450356206. DOI: 10.1145/3173574.3174007. URL: http://dl.acm.org/citation.cfm?doid=3173574.3174007.

[67] Vladan Koncar. "Optical Fiber Fabric Displays." In: *Opt. Photon. News* 16.4 (2005), pp. 40–44. URL: http://www.osa-opn.org/abstract.cfm?URI=opn-16-4-40.

[68] S Suresh Kumar, Vinay Kumar Midha, Alpa Kumari, Upma Gupta, and Shruti Singla. "Transmittance behaviour of curtain fabrics under natural lighting conditions." In: (2018).

[69] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. "Zooids: Building Blocks for Swarm User Interfaces." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, 97–109. ISBN: 9781450341899. DOI: 10.1145/2984511.2984547. URL: https://doi.org/10.1145/2984511.2984547.

[70] Lee-Delisle. "PixelPhones: http://sebleedelisle.com/2011/09/pixelphones-a-huge-display-made-with-smart-phones/." In.

[71] Jinha Lee, Yasuaki Kakehi, and Takeshi Naemura. "Bloxels: Glowing Blocks as Volumetric Pixels." In: *ACM SIGGRAPH 2009 Emerging Technologies*. SIGGRAPH '09. New Orleans, Louisiana: Association for Computing Machinery, 2009. ISBN: 9781605588339. DOI: 10.1145/1597956.1597961. URL: https://doi.org/10.1145/1597956.1597961.

[72] Kenneth Levenberg. "A method for the solution of certain non-linear problems in least squares." In: *Quarterly of Applied Mathematics* 2.2 (1944), pp. 164–168. ISSN: 0033-569X. DOI: 10.1090/qam/10666. URL: http://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/.

[73] Ming Li and Leif Kobbelt. "Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices." In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. MUM '12. Ulm, Germany: Association for Computing Machinery, 2012. ISBN: 9781450318150. DOI: 10.1145/2406367.2406397. URL: https://doi.org/10.1145/2406367.2406397.

[74] J. Lifton, Michael Broxton, and Joseph Paradiso. "Experiences and directions in Pushpin computing." In: vol. 2005. May 2005, pp. 416 –421. ISBN: 0-7803-9201-9. DOI: 10.1109/IPSN.2005.1440959.

[75] Joshua Lifton, Deva Seetharam, Michael Broxton, and Joseph Paradiso. "Pushpin Computing System Overview: A Platform for Distributed, Embedded, Ubiquitous Sensor Networks." In: vol. 2414. Aug. 2002, pp. 605–614. ISBN: 978-3-540-44060-4. DOI: 10.1007/3-540-45866-2_12.

[76] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. "Ambient Backscatter: Wireless Communication out of Thin Air." In: *SIGCOMM Comput. Commun. Rev.* 43.4 (2013), 39–50. ISSN: 0146-4833. DOI: 10.1145/2534169.2486015. URL: https://doi.org/10.1145/2534169.2486015.

[77] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. "Ambient Backscatter: Wireless Communication out of Thin Air." In: *SIGCOMM Comput. Commun. Rev.* 43.4 (2013), 39–50. ISSN: 0146-4833. DOI: 10.1145/2534169.2486015. URL: https://doi.org/10.1145/2534169.2486015.

[78] Chin-Yuan Lu, Han-Wei Hsieh, Rong-Hao Liang, Chi-Jung Lee, Ling-Chien Yang, Mengru Xue, Jr-Ling Guo, Meng-Ju Hsieh, and Bing-Yu Chen. "Combining Touchscreens with Passive Rich-ID Building Blocks to Support Context Construction in Touchscreen Interactions." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450380966. URL: https://doi.org/10.1145/3411764.3445722.

[79] Donald W Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441. ISSN: 0368-4245. DOI: 10.1137/0111030. URL: http://epubs.siam.org/doi/10.1137/0111030.

[80] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. "SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machin-

ery, 2018. ISBN: 9781450356206. DOI: 10.1145/3173574.3173928. URL: https://doi.org/10.1145/3173574.3173928.

[81] Matthew Mauriello, Michael Gubbels, and Jon E Froehlich. “Social fabric fitness: the design and evaluation of wearable E-textile displays to support group running.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 2833–2842.

[82] Yogesh Kumar Meena et al. “PV-Tiles: Towards Closely-Coupled Photovoltaic and Digital Materials for Useful, Beautiful and Sustainable Interactive Surfaces.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376368. URL: https://doi.org/10.1145/3313831.3376368.

[83] M. Merenda, D. Iero, and F. G. D. Corte. “Reconfigurable UHF RFID tag with sensing capabilities.” In: *2019 IEEE International Conference on RFID Technology and Applications (RFID-TA)*. 2019, pp. 333–336. DOI: 10.1109/RFID-TA.2019.8891965.

[84] David Merrill, Jeevan Kalanithi, and Pattie Maes. “Siftables: Towards Sensor Network User Interfaces.” In: *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. TEI ’07. Baton Rouge, Louisiana: Association for Computing Machinery, 2007, 75–78. ISBN: 9781595936196. DOI: 10.1145/1226969.1226984. URL: https://doi.org/10.1145/1226969.1226984.

[85] H. Miawarni and E. Setijadi. “VLC design and experiment using 650 nm LASER transmitter and Parallel Red Diffuse LED as receiver.” In: *2016 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*. 2016, pp. 65–68. DOI: 10.1109/COMNETSAT.2016.7907418.

[86] Jiri Militky, Marie Travnickova, and Vladimir Bajzik. “Air permeability and light transmission of weaves.” In: *International journal of clothing science and technology* (1999).

[87] Pierre Moulon, Pascal Monasse, and Renaud Marlet. “Adaptive Structure from Motion with a Contrario Model Estimation.” In: *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. Springer Berlin Heidelberg, 2012, pp. 257–270. ISBN: 9783642374463. DOI: 10.1007/978-3-642-37447-0_20. URL: http://link.springer.com/10.1007/978-3-642-37447-0{_}20.

[88] Christos Mourouzi, Isabel P. S. Qamar, and Anne Roudaut. “Sweep-Screen: Sweeping Programmable Surfaces to Create Low-Fi Displays Everywhere.” In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI EA ’18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–6. ISBN: 9781450356213. DOI: 10.1145/3170427.3188462. URL: https://doi.org/10.1145/3170427.3188462.

[89] Grace Ngai, Stephen C.F. Chan, Vincent T.Y. Ng, Joey C.Y. Cheung, Sam S.S. Choy, Winnie W.Y. Lau, and Jason T.P. Tse. "I*CATch: A Scalable Plug-n-Play Wearable Computing Framework for Novices and Children." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, 443–452. ISBN: 9781605589299. DOI: 10.1145/1753326.1753393. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/1753326.1753393.

[90] Hiroki Nozaki. "Flying Display: A Movable Display Pairing Projector and Screen in the Air." In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '14. Toronto, Ontario, Canada: ACM, 2014, pp. 909–914. ISBN: 978-1-4503-2474-8. DOI: 10.1145/2559206.2579410. URL: http://doi.acm.org/10.1145/2559206.2579410.

[91] Simon Olberding, Michael Wessely, and Jürgen Steimle. "PrintScreen: Fabricating Highly Customizable Thin-Film Touch-Displays." In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, 281–290. ISBN: 9781450330695. DOI: 10.1145/2642918.2647413. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2642918.2647413.

[92] Ugur Olgun, Chi-Chih Chen, and John L. Volakis. "Efficient ambient WiFi energy harvesting technology and its applications." In: *Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation*. 2012, pp. 1–2. DOI: 10.1109/APS.2012.6349384.

[93] Gary M Olson, Susan A Duffy, and Robert L Mack. "Thinking-Out-Loud as a Method for Studying 11 Real-Time Comprehension Processes." In: *New methods in reading comprehension research* (2018), p. 253.

[94] Alex Olwal and Artem Dementyev. "Hidden Interfaces for Ambient Computing: Enabling Interaction in Everyday Materials through High-Brightness Visuals on Low-Cost Matrix Displays." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517674. URL: https://doi.org/10.1145/3491102.3517674.

[95] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoediauer, Martin Kaltenbrunner, Siegfried Bauer, et al. "RESi: A Highly Flexible, Pressure-Sensitive, Imperceptible Textile Interface Based on Resistive Yarns." In: *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM. 2018, pp. 745–756.

[96] Roshan Lalintha Peiris, Owen Noel Newton Fernando, Chua Su Bee, Adrian David Cheok, Arij Glycin Ganesan, and Prabhash Kumarasinghe. "dMarkers: ubiquitous dynamic makers for augmented reality." In: *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*. ACM. 2011, pp. 217–224.

[97] Roshan Lalintha Peiris, Mili John Tharakan, Adrian David Cheok, and Owen Noel Newton. "AmbiKraf: A Ubiquitous Non-Emissive Color Changing Fabric Display." In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. MindTrek '11. Tampere, Finland: Association for Computing Machinery, 2011, 320–322. ISBN: 9781450308168. DOI: 10.1145/2181037.2181096. URL: https://doi.org/10.1145/2181037.2181096.

[98] Roshan Lalintha Peiris, Mili John Tharakan, Adrian David Cheok, and Owen Noel Newton. "AmbiKraf: a ubiquitous non-emissive color changing fabric display." In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. 2011, pp. 320–322.

[99] Claudio S. Pinhanez. "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces." In: *Proceedings of the 3rd International Conference on Ubiquitous Computing*. UbiComp '01. Atlanta, Georgia, USA: Springer-Verlag, 2001, 315–331. ISBN: 3540426140.

[100] Claudio Pinhanez. "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces." In: *Ubicomp 2001: Ubiquitous Computing*. Ed. by Gregory D. Abowd, Barry Brumitt, and Steven Shafer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 315–331. ISBN: 978-3-540-45427-4.

[101] Pol Pla and Pattie Maes. "Display Blocks: A Set of Cubic Displays for Tangible, Multi-perspective Data Exploration." In: *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. TEI '13. Barcelona, Spain: ACM, 2013, pp. 307–314. ISBN: 978-1-4503-1898-3. DOI: 10.1145/2460625.2460677. URL: http://doi.acm.org/10.1145/2460625.2460677.

[102] Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. "Project Jacquard: Interactive Digital Textiles at Scale." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 4216–4227. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858176. URL: http://doi.acm.org/10.1145/2858036.2858176.

[103] "Premium Handmade LED Clothing." In: (2019). URL: http://led-clothing.com/.

[104] Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. "HuddleLamp: Spatially-Aware Mobile Displays for Ad-Hoc Around-the-Table Collaboration." In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. Dresden, Germany: Association for Computing Machinery, 2014, 45–54. ISBN: 9781450325875. DOI: 10.1145/2669485.2669500. URL: https://doi.org/10.1145/2669485.2669500.

[105] Sami Ronkainen, Jonna Häkkilä, Saana Kaleva, Ashley Colley, and Jukka Linjama. “Tap input as an embedded interaction method for mobile devices.” In: *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM. 2007, pp. 263–270.

[106] M. Rubenstein, C. Ahler, and R. Nagpal. “Kilobot: A low cost scalable robot system for collective behaviors.” In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 3293–3298. DOI: `10.1109/ICRA.2012.6224638`.

[107] T Scott Saponas, Chris Harrison, and Hrvoje Benko. “PocketTouch: through-fabric capacitive touch input.” In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM. 2011, pp. 303–308.

[108] Munehiko Sato, Atsushi Hiyama, Tomohiro Tanikawa, and Michitaka Hirose. “Particle Display System: Virtually Perceivable Pixels with Randomly Distributed Physical Pixels.” In: *Journal of Information Processing* 17 (2009), pp. 280–291. DOI: `10.2197/ipsjjip.17.280`.

[109] Munehiko Sato, Atsushi Hiyama, Tomohiro Tanikawa, and Michitaka Hirose. “Particle Display System — Virtually Perceivable Pixels with Randomly Distributed Physical Pixels —.” In: *JIP* 17 (Dec. 2009), pp. 280–291. DOI: `10.2197/ipsjjip.17.280`.

[110] Munehiko Sato, Yasuhiro Suzuki, Shinya Nishizaka, Yusuke Torigoe, Atsushi Izumihara, Atsushi Hiyama, Kunihiro Nishimura, Tomohiro Tanikawa, and Michitaka Hirose. “Large-Scale Displays for Public Spaces—Constellation of Departure: Presenting the Impression of Airplanes Taking off above an Airport Departure Lounge.” In: *Comput. Entertain.* 9.1 (Apr. 2011). DOI: `10.1145/1953005.1953009`. URL: `https://doi.org/10.1145/1953005.1953009`.

[111] Jürgen Scheible and Markus Funk. “In-situ-displaydrone: Facilitating Co-located Interactive Experiences via a Flying Screen.” In: *Proceedings of the 5th ACM International Symposium on Pervasive Displays*. PerDis ’16. Oulu, Finland: ACM, 2016, pp. 251–252. ISBN: 978-1-4503-4366-4. DOI: `10.1145/2914920.2940334`. URL: `http://doi.acm.org/10.1145/2914920.29403`.

[112] Ryan Schmidt, Eric Penner, and Sheelagh Carpendale. “Reconfigurable displays.” In: *Workshop on Ubiquitous Display Environments*. Citeseer. 2004.

[113] Arne Schmitz, Ming Li, Volker Schönefeld, and Leif Kobbelt. “Ad-Hoc Multi-Displays for Mobile Interactive Applications.” In: *Eurographics*. 2010.

[114] Stefan Schneegass, Sophie Ogando, and Florian Alt. “Using On-Body Displays for Extending the Output of Wearable Devices.” In: *Proceedings of the 5th ACM International Symposium on Pervasive Displays*. PerDis ’16. Oulu, Finland: Association for Computing Machinery, 2016, 67–74. ISBN: 9781450343664. DOI: `10.1145/2914920.2915021`. URL: `https://doi.org/10.1145/2914920.2915021`.

[115] Julia Schwarz, David Klionsky, Chris Harrison, Paul Dietz, and Andrew Wilson. "Phone As a Pixel: Enabling Ad-hoc, Large-scale Displays Using Mobile Devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2235–2238. ISBN: 978-1-4503-1015-4. DOI: `10.1145/2207676.2208378`. URL: `http://doi.acm.org/10.1145/2207676.2208378`.

[116] Susanne Seitinger, Daniel S. Perry, and William J. Mitchell. "Urban Pixels: Painting the City with Light." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, 2009, 839–848. ISBN: 9781605582467. DOI: `10.1145/1518701.1518829`. URL: `https://doi.org/10.1145/1518701.1518829`.

[117] R. Shigeta, T. Sasaki, D. M. Quan, Y. Kawahara, R. J. Vyas, M. M. Tentzeris, and T. Asami. "Ambient RF Energy Harvesting Sensor Device With Capacitor-Leakage-Aware Duty Cycle Control." In: *IEEE Sensors Journal* 13.8 (2013), pp. 2973–2983. ISSN: 1530-437X. DOI: `10.1109/JSEN.2013.2264931`.

[118] I. Slavkov, D. Carrillo-Zapata, N. Carranza, X. Diego, F. Jansson, J. Kaandorp, S. Hauert, and J. Sharpe. "Morphogenesis in robot swarms." In: *Science Robotics* 3.25 (2018). DOI: `10.1126/scirobotics.aau9178`. eprint: `https://robotics.sciencemag.org/content/3/25/eaau9178.full.pdf`. URL: `https://robotics.sciencemag.org/content/3/25/eaau9178`.

[119] Yuta Sugiura, Toby Chong, Wataru Kawai, and Bruce H. Thomas. "Public/Private Interactive Wearable Projection Display." In: *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '18. Tokyo, Japan: Association for Computing Machinery, 2018. ISBN: 9781450360876. DOI: `10.1145/3284398.3284415`. URL: `https://doi.org/10.1145/3284398.3284415`.

[120] Yuta Sugiura, Koki Toda, Takayuki Hoshi, Youichi Kamiyama, Takeo Igarashi, and Masahiko Inami. "Graffiti Fur: Turning Your Carpet into a Computer Display." In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, 149–156. ISBN: 9781450330695. DOI: `10.1145/2642918.2647370`. URL: `https://doi.org/10.1145/2642918.2647370`.

[121] Yuta Sugiura, Koki Toda, Takashi Kikuchi, Takayuki Hoshi, Youichi Kamiyama, Takeo Igarashi, and masahiko Inami. "Grassffiti: Drawing Method to Produce Large-Scale Pictures on Conventional Grass Fields." In: *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '17. Yokohama, Japan: Association for Computing Machinery, 2017, 413–417. ISBN: 9781450346764.

DOI: 10.1145/3024969.3025067. URL: https://doi.org/10.1145/3024969.3025067.

[122] Joanna Szmyt and Zbigniew Mikolajczyk. "Light transmission through decorative knitted fabrics in correlation with their fabric cover." In: *AUTEX Research Journal* 10.2 (2010), pp. 44–48.

[123] Kazuki Takashima, Saul Greenberg, Ehud Sharlin, and Yoshifumi Kitamura. *A shape-shifting wall display that supports individual and group activities*. Tech. rep. U. Calgary, 2015. DOI: 10.11575/PRISM/30644.

[124] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Joshua R. Smith, and Shyamnath Gollakota. "Powering the next Billion devices with wi-fi." In: *Communications of the ACM* 60.3 (2017), 83–91. ISSN: 0001-0782. DOI: 10.1145/3041059. URL: http://dx.doi.org/10.1145/3041059.

[125] B Thomas, Karen Grimmer, J Zucco, and Steve Milanese. "Where does the mouse go? An investigation into the placement of a body-attached touchpad mouse for wearable computers." In: *Personal and Ubiquitous computing* 6.2 (2002), pp. 97–112.

[126] LED Throwies. "http://www.graffitiresearchlab.com/blog/projects/led-throwies/." In.

[127] Zhao Tian, Kevin Wright, and Xia Zhou. "The DarkLight Rises: Visible Light Communication in the Dark." In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. MobiCom '16. New York City, New York: Association for Computing Machinery, 2016, 2–15. ISBN: 9781450342261. DOI: 10.1145/2973750.2973772. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2973750.2973772.

[128] Christian Tonn, Frank Petzold, Oliver Bimber, Anselm Grundhöfer, and Dirk Donath. "Spatial augmented reality for architecture—designing and planning with and within existing buildings." In: *International Journal of Architectural Computing* 6.1 (2008), pp. 41–58.

[129] Radu-Daniel Vatavu. "Smart-Pockets: Body-deictic gestures for fast access to personal data during ambient interactions." In: *International Journal of Human-Computer Studies* 103 (2017), pp. 1–21.

[130] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. "Project Zanzibar: A Portable and Flexible Tangible Interaction Platform." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3174089. URL: https://doi.org/10.1145/3173574.3174089.

[131] Akira Wakita and Midori Shibutani. "Mosaic textile: wearable ambient display with non-emissive color-changing modules." In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. ACM. 2006, p. 48.

[132] Ryoichi Watanabe, Yuichi Itoh, Masatsugu Asai, Yoshifumi Kitamura, Fumio Kishino, and Hideo Kikuchi. "The Soul of ActiveCube: Implementing a Flexible, Multimodal, Three-Dimensional Spatial Tangible Interface." In: *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACE '04. Singapore: Association for Computing Machinery, 2004, 173–180. ISBN: 1581138822. DOI: 10.1145/1067343.1067364. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/1067343.1067364.

[133] Mark Weiser. "The Computer for the 21st Century." In: *SIGMOBILE Mob. Comput. Commun. Rev.* 3.3 (1999), 3–11. ISSN: 1559-1662. DOI: 10.1145/329124.329126. URL: https://doi.org/10.1145/329124.329126.

[134] Robert S Weiss. *Learning from strangers: The art and method of qualitative interview studies*. Simon and Schuster, 1995.

[135] Jason Wiese, T. Scott Saponas, and A.J. Bernheim Brush. "Phoneprioception: Enabling Mobile Phones to Infer Where They Are Kept." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2013, 2157–2166. ISBN: 9781450318990. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2470654.2481296.

[136] Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. "Steerable Augmented Reality with the Beamatron." In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST '12. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2012, 413–422. ISBN: 9781450315807. DOI: 10.1145/2380116.2380169. URL: https://doi.org/10.1145/2380116.2380169.

[137] Jasper de Winkel, Vito Kortbeek, Josiah Hester, and Przemysław Pawełczak. "Battery-Free Game Boy." In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4.3 (Sept. 2020). DOI: 10.1145/3411839. URL: https://doi.org/10.1145/3411839.

[138] Shuai Xu, Qian Ma, Xiao-Fang Yang, and Shu-Dong Wang. "Design and fabrication of a flexible woven smart fabric based highly sensitive sensor for conductive liquid leakage detection." In: *RSC Advances* 7.65 (2017), pp. 41117–41126.

[139] Jackie (Junrui) Yang and James A. Landay. "InfoLED: Augmenting LED Indicator Lights for Device Positioning and Communication." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 175–187. ISBN: 9781450368162. DOI: 10.1145/3332165.3347954. URL: https://doi.org/10.1145/3332165.3347954.

[140] Chungkuk Yoo, Inseok Hwang, Seungwoo Kang, Myung-Chul Kim, Seonghoon Kim, Daeyoung Won, Yu Gu, and Junehwa Song. "Card-Stunt as a Service: Empowering a Massively Packed Crowd for Instant Collective Expressiveness." In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '17. Niagara Falls, New York, USA: Association for Computing Machinery, 2017, 121–135. ISBN: 9781450349284. DOI: `10.1145/3081333.3081357`. URL: `https://doi.org/10.1145/3081333.3081357`.

[141] Y. Zhao, J. R. Smith, and A. Sample. "NFC-WISP: A sensing and computationally enhanced near-field RFID platform." In: *2015 IEEE International Conference on RFID (RFID)*. 2015, pp. 174–181. DOI: `10.1109/RFID.2015.7113089`.

[142] flashionstatement. "https://www.flashionstatement.com/." In.

[143] https://twinkly.com/. "Twinkly." In.

[144] https://ueberall.us/portfolio/dazzle/. "EInk Prism Screens." In.

[145] https://www.powercastco.com/. "Powercast." In.

[146] http://www.farsens.com/en/products/rocky100/. In.