Describing systems for the exploration of tangible and spatial computer interaction

Louis James

May 17, 2020

Acknowledgements

Thanks to my family, Florent, Chudleigh dwellers, Jamie ...

Abstract

Presented here is a specification of experimental approaches to computer interaction based on spatial and tangible methods. The report describes an prototypical implementation of a base system for interaction in space and a theoretical API for such a system and similar systems. This utilises computer vision techniques to analyse a surface to track objects and project onto the space creating a feedback system for interaction. To finish a ethnomethodological framework for evaluation and further development is proposed???

Contents

1 Introduction **5**
- 1.1 Project aims 5

2 Background **6**
- 2.1 Definitions 6
- 2.2 Beginning with the Exocortex 6
- 2.3 A virtual exploration of Dynamicland 7
- 2.4 Paper programs - open source 8
- 2.5 Sage digital research or Moving from implementation to abstraction 8
- 2.6 MIT Prof - tangible media group 9
- 2.7 Nielsen: augmenting ltm and using ai to augment human-i ?????? 9
- 2.8 Tangible bits - Hiroshi Ishii and Brygg Ullmer 9

3 Specification and context **10**

4 Project in depth **12**
- 4.1 Raspberry pi testing 12
- 4.2 Build 12
- 4.3 API 12
- 4.4 12

5 Creative process **14**

6 Debugging and problem solving **15**

7 Evaluation and Conclusions **16**

List of Figures

2.1 RealtalkOS, the operating system of *Dynamicland* 8

3.1 The initial physical schema: *Paperprograms* 11

4.1 System schema 13

Chapter 1

Introduction

1.1 Project aims

- open source project for tangible interaction
- Prototypical
- ethnomethodological frameworks for evaluation

Chapter 2

Background

The motivation for this project stems in part from a feeling of frustration in how working on computers can often be a constricted affair and a pondering over how we might expand the *keyboard-mouse-monitor* model to improve the utility of computers regarding our own perceptive abilities. How might a spatial, haptic and tangible environment for interaction create an improved space for working and thinking with computers as well with our physical health? How might such an environment fundamentally augment our cognitive capabilities; memory and learning as well as creativity itself?

2.1 Definitions

1. Computing
2. keyboard-mouse-monitor model (kmm model) (?)
3. Cognition
4. Exocortex

2.2 Beginning with the Exocortex

I started off looking at *Exocortexes* and other personal archiving systems. Systems that allow the user to externalise thought and memory. This could be via simply storing and organising work and ideas efficiently and methodically or unifying many tasks or different workflows into a singular interface.

Org mode is a good example of such a system. Org mode is a "computing environment for authoring mixed natural and computer language documents" [1]. It is designed for taking notes, producing documents and organising and runs inside of the text editor, Emacs. It has the ability to export to different formats such as HTML, LaTeX and supports "outlining, note-taking, hyperlinks, spreadsheets, TODO lists, project planning, GTD" as well as literate programming, all in plain-text [1]. (it is incidentally what this document is produced with)

Another point of reference when I was looking at externalised 'artificial information-processing systems' was Devine Lu Linvega's Exocortex XXIIV – nataniev. *XXIIV* is a personal archive and log with documentation of Linvega's personal tools and artworks. The site has gone through some changes since I first came across it. Originally a static, javascript and lisp based website with diaries, blog type posts and categorised personal logs. It is now somewhat stripped back in style and has been rewritten in C (C99). The work contains a selection of esoteric programming tools including a synthetic language, games, software all logged using their own *Arvelie calendar* [2].

Both these two systems have their own specific use-cases, *Org-mode*; in academia and science and *XXIIV*; an experimental personal archive. They both utilise the contemporary and prevailing *keyboard-mouse-monitor* paradigm of computer interaction but push the boundaries of cognition in this medium, particularly regarding memory and productivity. These two projects were a birth point in thinking about how software systems can be augment thought and improve learning ability and computer productivity.

2.3 A virtual exploration of Dynamicland

Another original and critical point of reference was *Dynamicland*, a research project in Oakland, USA. The aim of the project is to implement a new more powerful and accessible model of computing.

> In Oakland, we built the first full-scale realization of the vision, inviting thousands of people into our space to collaborate. Together, these artists, scientists, teachers, students, programmers, and non-programmers created hundreds of projects that would have been impossible anywhere else. – Dynamicland.org

Dynamicland is a communal computer where the building is the computer (ENIAC). Programs are embodied in the room on pieces of colour-coded paper. The programs are recognised via the codes and their code, stored in a database is then run, it can also *read* code using OCR but generally the code is there symbolically. Projectors on the ceiling transform the paper and workbenches into whatever the programmer decides. This relatively simple model makes for an exciting new ecosystem for collaborative computing and expressive programming. Victor highlights his ideas for the progression of computing and interaction in a series of talks (available online) and on his website. In his talk "Seeing Spaces" he talks of a new kind of maker-space which allow makers to see across time and possibilities. *Dynamicland* offers a computational medium which allows for full use of the human senses and a more humane representation of thought [3].

DL was the inspiration for the main physical and technical model for this project, an *augmented* workspace either on the floor or a table which is projected onto. A camera/s pointing down onto the projection space is the sensor for detecting interaction, with the projector as the actuator. This base model can be seen in Figures 3.1 and 4.1.

Figure 2.1: RealtalkOS, the operating system of *Dynamicland*

2.4 Paper programs - open source

Looking to find some of the code for *Dynamicland* and a more detailed specification of **DL** I stumbled across *Paper Programs* (*Dynamicland* has an 'open-source model', but it is only open if you can visit it physically as the source code is physically in the space). *Paper Programs* is a browser-based partial clone of *Dynamicland*. This was another starting point for playing around with but I found that I couldn't set it up and have it stable enough to develop on. It also suffers from being quite slow, due to the Computer Vision and graphics being done in the browser (it uses a version of OpenCv compiled to WebAssembly) [4]. While WebAssembly has the scope for doing high-performance computation in the browser but I found there was still a significant lag from detecting papers to projecting back down on to them. Another branch which had implemented blob detection on the GPU I also found to be slow and unstable (Link to pull request), this may have been due to my lighting and camera setup.

2.5 Sage digital research or Moving from implementation to abstraction

Moving from implementation to abstraction
Ethnomethodology
Embodied Cognition
Haptic interfaces

2.6 MIT Prof - tangible media group

`http://tangible.media.mit.edu/projects/`

2.7 Nielsen: augmenting ltm and using ai to augment human-i ??????

Other approaches to
[5]
[6]

2.8 Tangible bits - Hiroshi Ishii and Brygg Ullmer

[7]

Chapter 3

Specification and context

As in my original specification the main aim was to create a system for spatial interaction. Initially I imagined it to work on a table top surface (in the end it was developed on a floor mat due to considerations in my development environment; see Chapter 4). The other principle component was to that interaction would be based on the placement and movement of objects around the work-surface. The position and movements of theses objects would be picked up by a camera and actuated by a projector; both situated above the surface looking down onto it. It could also be setup in a horizontal direction, with, for example, magnetised components keeping the objects to a board. Alongside this a computer keyboard may be used for additional input such as inputting text or selecting something.

My original plan was to use the already made *Paperprograms* system and build on top of this. Due to technical issues with PP I decided to implement the system myself using openFrameworks, a C++ toolkit for experimental application development. I chose this framework as it has straightforward 'out of the box' graphics capabilities as well as numerous Add-ons which include *Opencv* [8] wrappers and GUI libraries as well as an active community of users. This combination in one framework seemed suitable for quick experimentation and prototyping for the project. The physical setup would include a Projector and HD webcam and computer for running the application.

Another design consideration I had in mind was accessibility. From my research into similar projects an aim was to create a similar system that could be open source and easily setup so that others could build on top of the system. This was another reason for using openFrameworks which is cross platform (Windows, OSx, IOS and Linux). This would mean with minor or no modification of the code, it could be run on any all the major desktop platforms. The hardware requirements are also the kind which are either cheaply sourced or available in most educational institutions: one of the target areas that further development was envisioned.

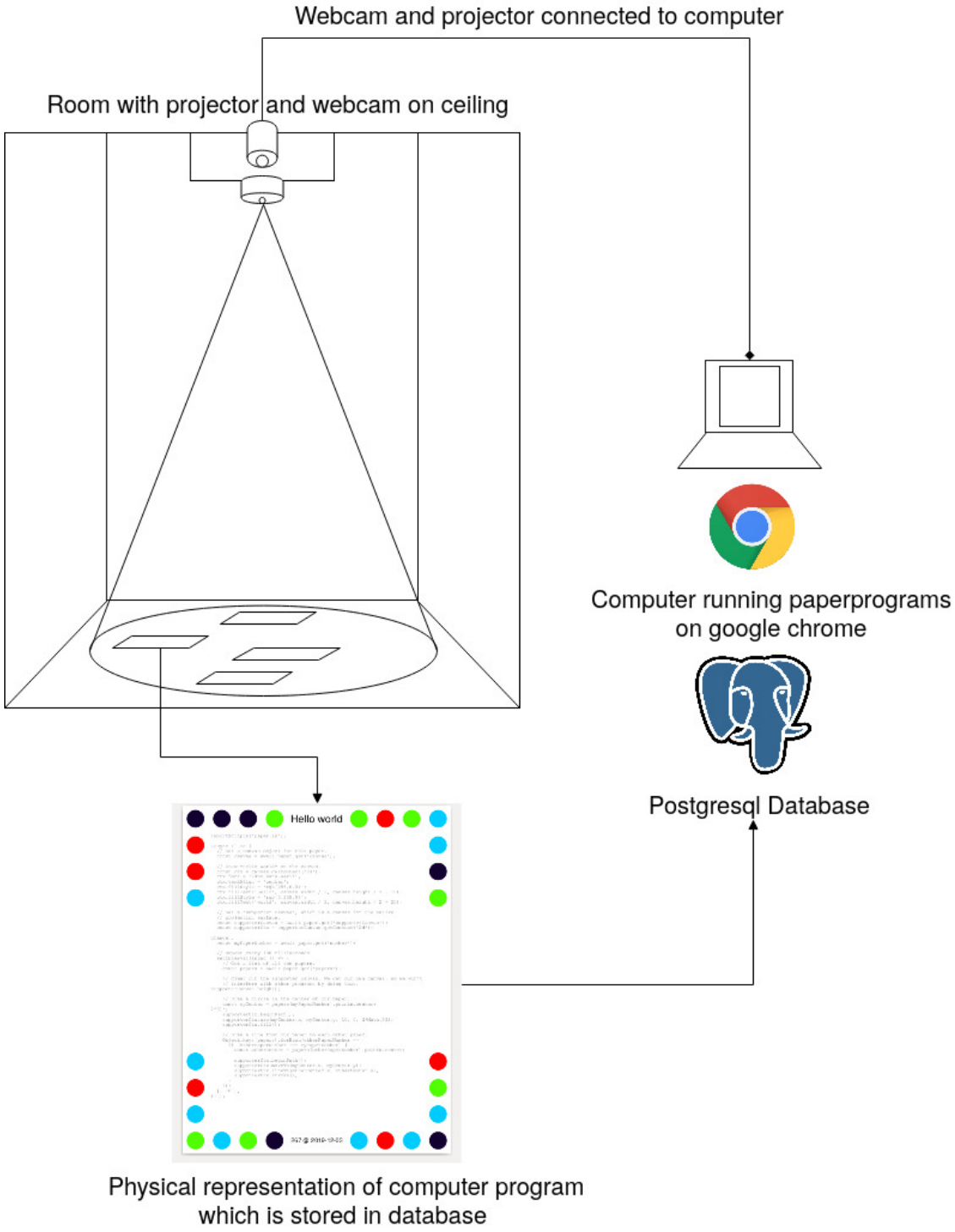

Figure 3.1: The initial physical schema: *Paperprograms*

Chapter 4

Project in depth

See system schema Fig.

4.1 Raspberry pi testing

4.2 Build

4.3 API

4.4

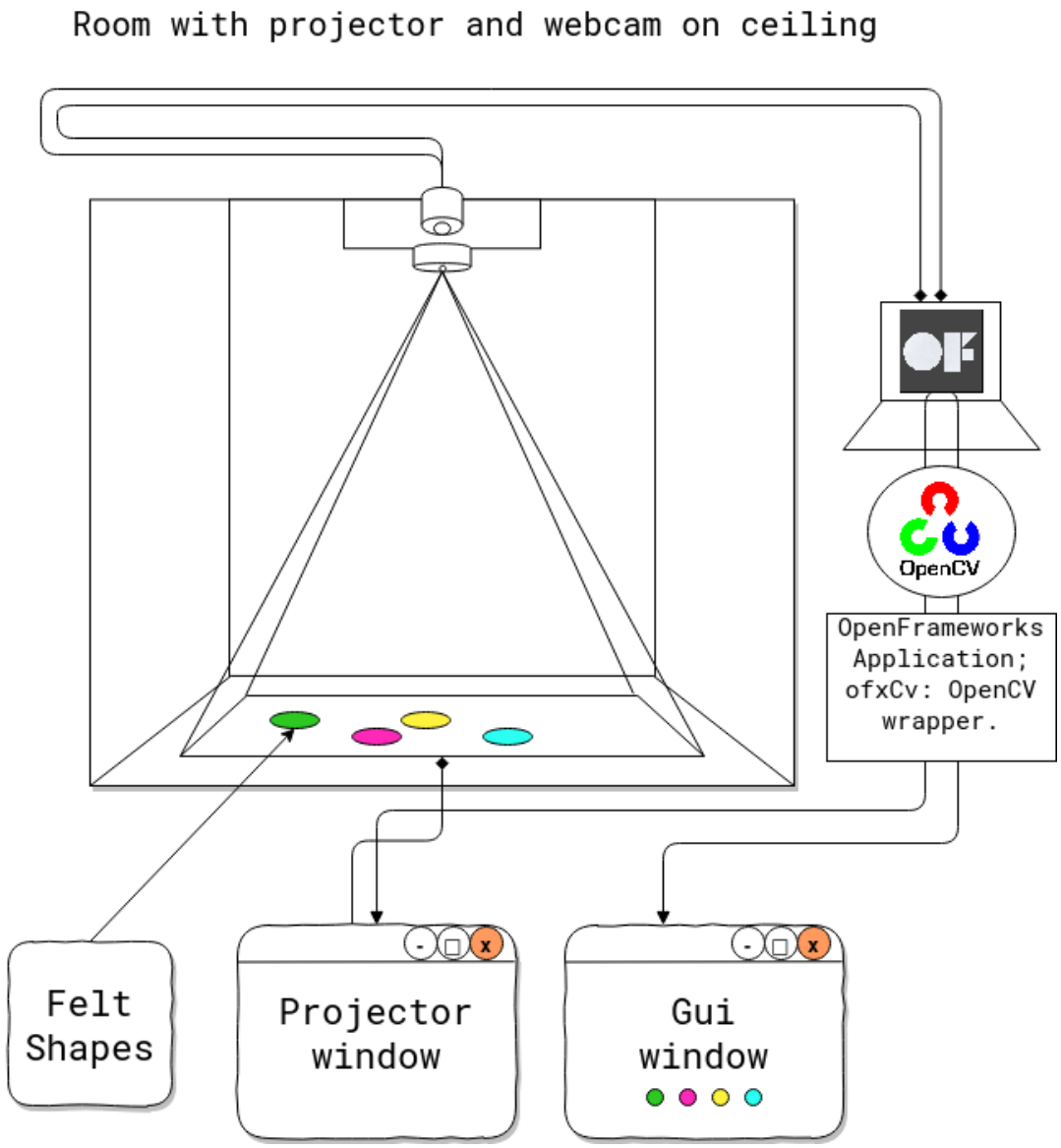

Figure 4.1: System schema

Chapter 5

Creative process

Chapter 6

Debugging and problem solving

Chapter 7

Evaluation and Conclusions

Bibliography

[1] E. Schulte, D. Davison, T. Dye, and C. Dominik, "A multi-language computing environment for literate programming and reproducible research," *Journal of Statistical Software*, vol. 46, pp. 1–24, 1 2012.

[2] D. L. Linvega, "The nataniev ecosystem is a collection of exocortex tools," 2020, 18W04, https://wiki.xxiivv.com/site/about.html.

[3] B. Victor and A. Kay, "Dynamicland," 2014, https://dynamicland.org.

[4] JP, "Paper programs is a browser-based system for running javascript programs on pieces of paper," 2018, https://paperprograms.org/.

[5] M. A. Nielsen, "Augmenting long-term memory," 2018, http://augmentingcognition.com/ltm.html.

[6] S. Carter and M. Nielsen, "Using artificial intelligence to augment human intelligence," *Distill*, 2017. https://distill.pub/2017/aia.

[7] H. Ishii, "Tangible bits: designing the seamless interface between people, bits, and atoms," in *Proceedings. International Symposium on Mixed and Augmented Reality*, pp. 199–199, IEEE, 2002.

[8] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.