

# **The *Electromagnetic Sphere Automaton* – Navigating Ferromagnetic Spheres to Display Binary Images**

Thomas Preindl



**MASTERARBEIT**

eingereicht am  
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2018

© Copyright 2018 Thomas Preindl

This work is published under the conditions of the Creative Commons License *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0)—see <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

# Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 14, 2018

Thomas Preindl

# Contents

<b>Declaration</b>	iii
<b>Acknowledgements</b>	vii
<b>Abstract</b>	viii
<b>Kurzfassung</b>	ix
<b>1 Introduction</b>	1
<b>2 Related Work</b>	3
2.1 Tangible Media Group . . . . .	3
2.2 Daniel Rozin . . . . .	3
2.3 SnOil . . . . .	4
2.4 The Mechanical Turk . . . . .	5
2.5 Robots as pixels . . . . .	6
<b>3 Rearrangement</b>	7
3.1 Glossary of important terms and mathematical notation . . . . .	8
3.2 Problem statement . . . . .	9
3.2.1 Movement constraints . . . . .	9
3.3 Role assignment . . . . .	9
3.3.1 Hungarian algorithm . . . . .	11
3.4 Motion planning . . . . .	11
3.5 Algorithmic description . . . . .	12
3.5.1 Heuristic Distance . . . . .	12
3.5.2 Progression combinatorics . . . . .	13
3.6 Performance . . . . .	13
3.7 Further considerations . . . . .	15
3.7.1 Multiple sphere types . . . . .	15
3.7.2 Feed and retrieval . . . . .	16
<b>4 Position Detection</b>	20
4.1 External camera . . . . .	20
4.2 Photoresistors . . . . .	20
4.3 Force-sensitive resistors . . . . .	20

4.4	Capacitive sensing . . . . .	21
4.5	Hall sensors . . . . .	21
4.6	Current sensing . . . . .	22
<b>5</b>	<b>Simulations and Experiments</b>	<b>23</b>
5.1	Sphere parameters . . . . .	23
5.2	Magnet parameters . . . . .	24
5.3	Sphere detection . . . . .	25
<b>6</b>	<b>Software</b>	<b>27</b>
6.1	Path finding . . . . .	27
6.2	Control logic . . . . .	28
6.3	Hardware driver . . . . .	28
6.4	Web application . . . . .	28
6.5	Sequential control system . . . . .	28
6.6	Third-party software . . . . .	29
<b>7</b>	<b>Hardware</b>	<b>31</b>
7.1	Prototypes . . . . .	31
7.2	Multi-channel variable voltage driver	33
7.2.1	12CH DMX512 LED Decoder . . . . .	35
7.2.2	16-Channel Magnet Driver . . . . .	35
7.2.3	PCA9685 PWM driver . . . . .	35
7.3	Microcontroller . . . . .	36
7.4	Computer . . . . .	36
7.5	Analog-to-digital converter . . . . .	36
7.6	Electromagnets . . . . .	36
7.7	Power supply . . . . .	37
7.8	Temperature sensor . . . . .	37
7.9	Position detection . . . . .	38
7.9.1	Hall sensors . . . . .	38
7.9.2	Force sensitive resistors . . . . .	38
7.10	Steel spheres . . . . .	38
7.11	Display surface . . . . .	39
7.12	Hazards . . . . .	39
7.13	Mechanical Design . . . . .	40
7.14	Mechanical Problems . . . . .	40
<b>8</b>	<b>Applications</b>	<b>42</b>
8.1	Proposed interaction modes . . . . .	42
8.2	Symbolic display . . . . .	42
8.3	Timer . . . . .	43
8.4	Game concepts . . . . .	43
8.4.1	Simon . . . . .	43
8.4.2	Last Man Standing . . . . .	44

Contents	vi
<b>9 Closing Remarks</b>	45
9.1 Conclusion . . . . .	45
9.2 Future development . . . . .	45
<b>A DVD Contents</b>	47
<b>References</b>	49
Literature . . . . .	49
Online sources . . . . .	50

# Acknowledgements

In the making of this thesis many people came to be inseparable from its final form and I owe them much for enabling me to hold this finished piece in my hands. Foremost I would like to thank my advisor Wilhelm Burger, who, from the first moment on, supported my idea. He not only gave priceless advice on every aspect of this extensive endeavour but also provided me with valuable tools to build the machine. I also like to give a loud shout out to Philip Stadler, with whom I had countless discussions on our work. His feedback and encouragement kept me pursuing a distant goal. Another big thanks to Simon Hofer my good friend and flatmate who never questioned me working through the nights, even when I went to sleep when he got out of bed. A heartfelt thanks goes to Stephanie Cedeño, whose short visit to Linz was enough to engage a vigorous friendship. A big thanks to her absolutely necessary input when it came the theoretic context behind this project. I would like to thank all my dear friends who helped me not getting lost in my thesis and showed me the colourful world around it. A special thanks to the Media Interaction Lab for supplying me the hardware for the position sensing. And last but not least I want to thank my family who supports me unconditionally in everything I do.

# Abstract

Predominant display technology focuses on high refresh rates and high resolution, thus trying to maximize throughput of information. Their design follows a mindset of efficiency and the desire to make the screen itself unnoticed, only presenting the digital information it is fed with. This thesis proposes a display which counteracts this tendency by creating an interface which focuses on slowness and minimalism, opening room for contemplation and in this way contradicting the conventional premise of a display.

The proposed display uses ferromagnetic spheres as pixels to display two-dimensional bitmaps. The spheres are moved by controlling a grid of individually addressable stationary electromagnets. The thesis evaluates the feasibility of such a device and discusses the related challenges; choosing the optimal setup of electromagnets, controlling the power of the electromagnets individually, detecting the position of the spheres and navigating them using a multi-agent path finding algorithm.

An implementation of the electromagnetic sphere display demonstrates and validates its functional principle. The thesis and the associated machine should augment the common idea of a display by resetting its customary traits and rebuilding it with a renewed focus on slowness and the visual, tactile and auditory experience of wandering pixels.

# Kurzfassung

Vorherrschende Bildschirmtechnologie fokussiert sich auf hohe Bildwiederholraten und hohe Auflösung, um so den Durchsatz an Information zu maximieren. Ihre Gestaltung folgt dem Gebot der Effizienz und dem Wunsch den Bildschirm selbst unbemerkt zu halten, um nur die digitale Information zu vermitteln, mit welcher der Bildschirm gespeist wird. Diese Arbeit stellt einen Bildschirm vor welcher dieser Tendenz entgegenwirkt, indem er sich auf Langsamkeit und Minimalismus fokussiert, dabei Raum für Reflexion schafft und so der konventionellen Prämisse eines Bildschirms widerspricht.

Der hier entworfene Bildschirm nützt ferromagnetische Kugeln als Pixel um zweidimensionale Bitmaps darzustellen. Die Kugeln werden durch ein Raster an stationären Elektromagneten bewegt, welche einzeln und in variabler Intensität angesteuert werden. Diese Arbeit evaluiert die Machbarkeit eines solchen Geräts und behandelt die dazugehörigen Herausforderungen; die Wahl der optimalen Elektromagneten, die variable Ansteuerung der einzelnen Elektromagneten, die Positionserkennung der Kugeln und die Navigation dieser durch einen Multiagenten-Wegfindungsalgorithmus.

Eine Implementierung des elektromagnetischen Kugel-Bildschirms demonstriert und validiert sein Funktionsprinzip. Die Arbeit und die dazugehörige Maschine sollen die übliche Vorstellung eines Bildschirms erweitern, indem sie seine gewohnten Eigenschaften mit einem Fokus auf Langsamkeit und die visuelle, taktile und auditive Erfahrung von wandernden Kugeln neu bestimmt.

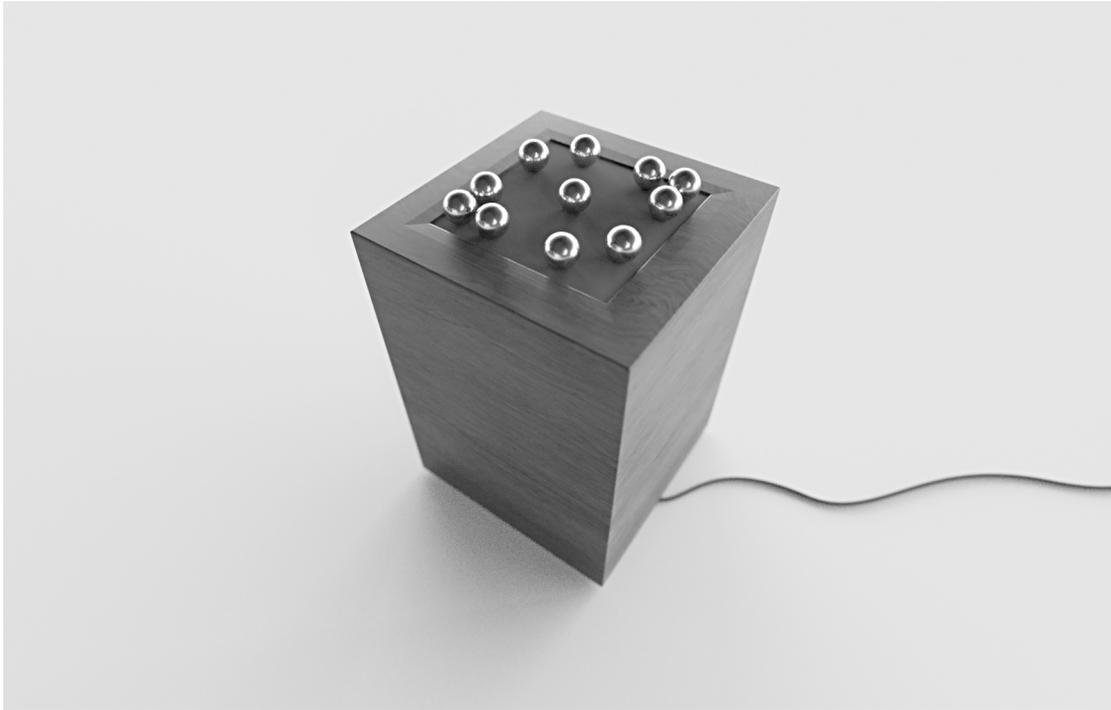
# Chapter 1

## Introduction

In the urban society displays are found everywhere, saturating people with information. They claim their space in living rooms, bars and pockets, serving as ever present messengers. While being omnipresent, the common display itself is not designed to be the focus of attention, rather the displayed information should lead the monologue. The usual LCD screen is black, with a preferably unnoticeable thin frame. A modern display sets out to be a window into another world, yet the window itself should not be seen. The design reveals the desire to obliterate the traces of the distributing medium only showing information it wants to present.

The electromagnetic sphere display (see Fig. 1.1) on the other hand aims to focus the attention of the observer on itself rather than the information it presents. It guides the gaze of the beholder along the saccadic movement of the spheres and the sound they make when jumping into a new position. The information the steel spheres are presenting has to be decoded by the observer and cannot be apprehended immediately. Expanding the possibilities of a common display the pixels represented by the steel spheres can be manipulated and put into different positions. Compared to a device with a touchscreen, the proposed display reacts with a physical movement rather than a purely visual. The electromagnetic sphere display is able to react to interactions by rearranging the spheres into a new configuration and enter into a dialogue with the user. The possibility of interaction offers a wide range of applications to explore.

The machine immediately raises the question of why such a device should be built. Intuitively it seems inconvenient to display information by placing steel spheres on a board. If the spheres only represent dots on a grid, faster, smaller and more rigid machines can easily be thought of, doing the very same thing. This assumes that the sole purpose of a display is to represent information in a direct manner, without the display interfering with the information transmission through its technical limitations or its appearance. Marshal McLuhan famously states, what has become a proverb by now, that “the medium is the message” [8]. Taking this statement literally it is always valid to create a new medium. Through the long-winded process of changing the presented information the machine puts its emphasis on the process of depiction. The machine purposefully increases the distance between presentation and the presented, opening up a cleft into which the observer can step to think about what is shown, who shows it and why it is shown. The machine deliberately states that it is depicting something, that it is not a window to another world but rather a reduced representation of it. Similarly, the



**Figure 1.1:** A concept rendering of the proposed machine.

German playwright Berthold Brecht used the “*Verfremdungseffekt*” (distancing effect) to enable the audience to take on a position of critical distance. It is a concept by which any illusion of a play being real should be destroyed. The goal is to let the audience in for interpretation instead of identification, just as the concept of the electromagnetic sphere display wants the observer to carefully interpret the information that is shown in its abstract form.

Besides from serving the purpose of conveying information, the machine is also a kinetic sculpture and can be incorporated into the realm of *rolling ball sculptures*.

This thesis is organized as follows. Chapter 2 discusses work that relates to the proposed machine, influenced its development or fuelled the idea. Following the central part of this thesis, Chapter 3, describes the path finding algorithm to navigate the spheres on the board. Chapter 4 presents different methods for detecting the sphere positions. Some of those methods are tested in the following Chapter 5, where simulations and experiments of essential technical aspects of the machine are described. The details of the underlying software are explained in Chapter 6 while the following Chapter 7 describes the used hardware parts. Hypothetical applications for the machine are described in Chapter 8. Finally the closing Chapter 9 reflects upon the encountered problems, the final result and possible further development.

## Chapter 2

# Related Work

Displays have become an irreplaceable part of modern life. As such, we are very familiar with the concept of a display. Yet there are numerous displays which defy their common image. Following a handful of objects and concepts which meet this requirement are selected and examined.

### 2.1 Tangible Media Group

In the realm of physical imaging devices, the *Tangible Media Group*, led by Hiroshi Ishii, is one of the most profiled and productive research groups. They seek to give digital information a dynamic physical form. Following a concept called *Radical Atoms* they aim to create interfaces which directly map bits to the physical space, something they call a tangible user interface [2]. They compare the interaction with conventional screens as like looking through a pool of water while the information we interact with is down below. A metaphor of an iceberg in the water is used to symbolize the information we want to interact with. Through tangible user interfaces the iceberg should be lifted out of the water, so we can directly interact with it. At the moment we only can see the tip of the iceberg, but in the future we may directly interact with a physical manifestation of all digital information [2]:

[...] as if the iceberg had risen from the depths to reveal its sunken mass.

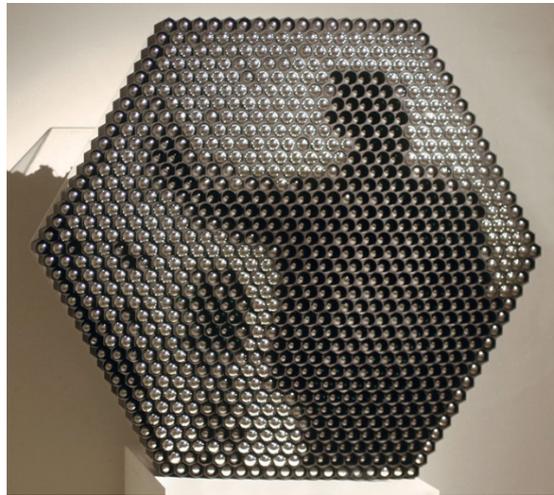
An example of their interfaces is the Dynamic Shape Display *InForm* [4] which uses a grid of actuated pistons to generate 2.5D shapes (see Fig. 2.1). The electromagnetic sphere display similarly creates a tangibles image, yet the displayed image is dominated by the shape of the spheres.

### 2.2 Daniel Rozin

Display devices which focus on the experience of sound and movement of the display are built by the Israeli-American artist Daniel Rozin [14]. He approaches the concept of a display in a way, in which pixels are represented by rotating or expanding and retracting objects. These displays, called *mechanical mirrors*, are equipped with a camera to mirror the observer. An example of his work can be seen in Fig. 2.2. The electromagnetic sphere



**Figure 2.1:** Dynamic Shape Display inFORM can render 3D content physically. Image source [18].

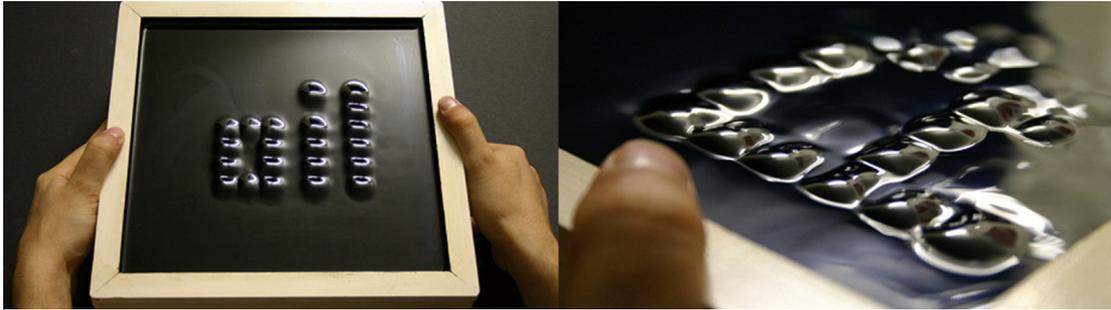


**Figure 2.2:** The art installation *Shiny Balls Mirror* (2003) by Daniel Rozin. Image source [14].

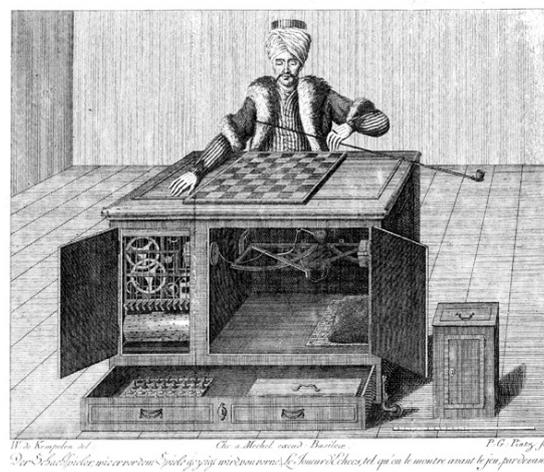
display similarly draws part of its charm from the sound it makes piloting the spheres with the humming magnets.

### 2.3 SnOil

One of the related works in terms of technology is *SnOil* [16], a physical display using ferrofluid (a fluid reactive to magnetic fields) and a matrix of electromagnets to display low resolution images. A basin is filled with ferrofluid under which the grid of electromagnets is placed. The electromagnets generate bumps in the overlying fluid. Those bumps act as pixels to display bitmaps. In contrast to the electromagnetic sphere display it can activate an arbitrary number of pixels simultaneously, making it more versatile.



**Figure 2.3:** The SnOil display works by deforming a ferrofluid with electromagnets. For interaction it uses a tilt sensor. Image source [16].

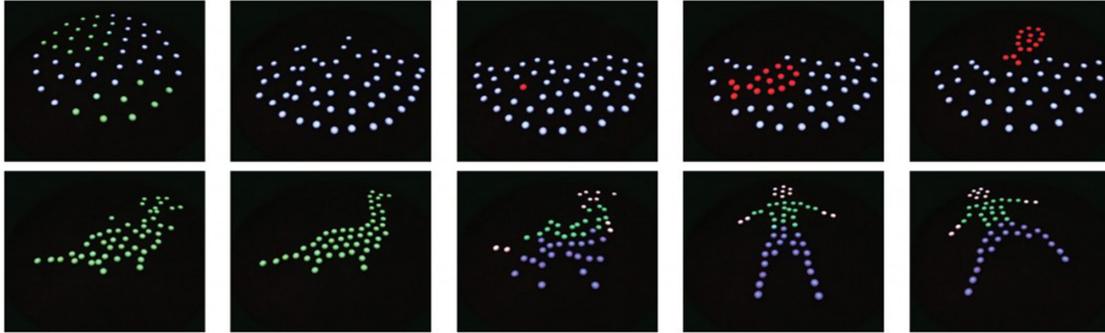


**Figure 2.4:** A copper engraving of the *Mechanical Turk*, showing the open cabinets and working parts. Image source [13]

## 2.4 The Mechanical Turk

An initial idea for the casing of the proposed display was a solemn wooden box, only the power cord suggesting it being an electronic device. This notion came from the original association to the *Mechanical Turk* as seen in Fig. 2.4, a fake chess-playing machine constructed in the late 18th century. The mechanical turk should convince the audience of being an autonomous chess playing machine, while a skilled chess player operated the machine from within, hidden behind complex yet misleading mechanical moving parts.

The ferromagnetic sphere display yet, should deliver the promise not held by the *Mechanical Turk*. It should be a machine autonomously reacting to interactions. Through the use of electromagnetism, which since ever holds a spellbinding fascination, the machine seeks to intrigue the observer. The use of wood avoids the association with electronic devices, rather drawing from the realm of antiquities and mechanical automatons.



**Figure 2.5:** A display that uses a robot swarm to create representational images and animations, with applications in entertainment. Image source [1]

## 2.5 Robots as pixels

The paper *Image and animation display with multiple mobile robots* explores the possibility of using physical robots as mobile pixels to display information [1]. They present a method to use a robot swarm for creating representational images and animations with applications in entertainment. Each robot thereby is equipped with a RGB LED for controllable colour (see Fig. 2.5). They see the benefits of this technology in the fact that the display is not bounded by a rectangle but by the surface the robots are moving on. The general concept is similar to the electromagnetic sphere display in a way that it has a fixed number of pixels which can be used and at the same time has to be moved physically.

## Chapter 3

# Rearrangement

To rearrange one arrangement of spheres to any other, a sequence of moves has to be found so that the spheres reach their determined destination. This problem of rearrangement falls into the research field of *multi-agent path finding* (MAPF), which treats the problem of finding paths for a set of agents in a graph, which the agents navigate simultaneously [10]. In this case we look at an *anonymous* MAPF, meaning that all agents are interchangeable and can be navigated to any target. The mobile agents in this particular application are the magnetically moved steel spheres and as they are indistinguishable to one another they can be attributed as being anonymous.

It can be said that this path finding problem, just as any MAPF problem, is a generalization of the sliding tile puzzle (also known as 15-puzzle) which is NP-complete [9]. Since the search space of the MAPF problem has a much higher branching factor than the sliding tile puzzle, as multiple agents can be moved at the same time, a well-informed search method is particularly important. MAPF problems are in general hard to solve due the combinatorial explosion of the search space. To make the problem more manageable, the process of multi-agent path finding is split into two sub-problems; *role assignment* (see Section 3.3) and *motion planning* (see Section 3.4).

Role assignment is the problem of assigning each agent a target position so that the maximum distance between any agent and target is minimized. By finding an assignment the *anonymous* MAPF problem is transformed to a MAPF problem.

In motion planning a sequence of moves is calculated so that every sphere reaches its prior determined target position. This is done by exploring a subset of all possible move sequences reduced by the movement constraints and an estimation function that tells us which branch of move sequences brings us the closest to the target arrangement. A straightforward method for finding a path is to traverse the search tree of possible consecutive arrangements using a global A\* approach, which is a common approach in solving MAPF problems [11]. In this particular application a standard A\* method is infeasible due the exceedingly large search space (see Section 3.6) and a heuristic function that leads to too many expansions. The branching factor of the search space can be estimated with  $\mathcal{O}(3^n)$ , with  $n$  being the number of agents, considering three possible next states for each agent when they only move towards their target. A plot of the branching factor can be seen in Fig. 3.4.

Multiple, not interchangeable sphere types (e.g., having different colours) are not considered in the following approach. This extended problem is examined in Section 3.7.1.

### 3.1 Glossary of important terms and mathematical notation

Before describing the problem a few terms have to be defined:

#### Position

A position represents the placement of a sphere on the display grid. Positions are referred to as

$$\mathbf{a}_i = (x_i, y_i), \quad (3.1)$$

where  $x_i, y_i \in \mathbb{N}$ ,  $0 \leq x_i < w$  and  $0 \leq y_i < h$ . The width of the display grid is denoted as  $w$  and the height as  $h$ .

#### Arrangement

An arrangement

$$A = (\mathbf{a}_0, \dots, \mathbf{a}_{n-1}) \quad (3.2)$$

is a ordered set of positions within the size of the display grid. It represents the placement of a set of spheres on the board. The starting configuration is called *initial arrangement* and the defined positioning which has to be reached *target arrangement*. An arrangement can also be represented as a  $w \times h$  matrix

$$\mathbf{A} = \begin{bmatrix} p_{0,0} & \cdots & p_{w-1,0} \\ \vdots & \ddots & \vdots \\ p_{0,h-1} & \cdots & p_{w-1,h-1} \end{bmatrix}, \quad (3.3)$$

with  $p_{j,k} \in [-1, 0, \dots, n-1]$ . If  $p_{j,k} = -1$  there is no agent occupying the position  $(j, k)$ . When  $p_{j,k} \geq 0$ ,  $p_{j,k}$  denotes the index in  $A$  of the agent occupying that position. When referred to as *anonymous* arrangement the spheres are interchangeable and have no assigned index. In this case  $p_{j,k} = 0$  marks an empty position and  $p_{j,k} = 1$  an occupied positions. Therefore an anonymous arrangement can be represented by a binary matrix.

#### Movement

A movement is the displacement of a single sphere to a neighbouring position. In this case movements can only target horizontally or vertically neighbouring positions. A movement always has a preceding position associated with it. Therefore, compared to an initial position  $\mathbf{a}_i$ , it is defined as

$$\mathbf{a}_i^* = \mathbf{a}_i + \mathbf{\Delta}, \quad (3.4)$$

where  $\mathbf{\Delta} \in \{\mathbf{0}, \mathbf{e}_x, \mathbf{e}_y, -\mathbf{e}_x, -\mathbf{e}_y\}$ , with  $\mathbf{e}_x$  and  $\mathbf{e}_y$  denoting the unit vectors in a Cartesian coordinate system of two dimensions.

#### Progression

A progression (also called consecutive arrangement) is an arrangement  $A^*$  that follows a preceding arrangement  $A$ . An allowed progression holds  $n$  movements that can be executed in parallel without contradicting any constraints. A progression is in its representation identical to an arrangement, with the contextual difference that it always has a preceding arrangement, so that

$$A^* = (\mathbf{a}_0^*, \dots, \mathbf{a}_{n-1}^*). \quad (3.5)$$

## 3.2 Problem statement

As initial arrangement we assume an arbitrarily positioned set of spheres (hereafter called *agents*)  $A_0$  on the board. The target arrangement is defined as  $T$ , while no two agents nor targets occupy the same position in their respective arrangement. The set of targets can originate from a bitmap image having the same resolution as the board. We also require the number of targets to be equal to the number of agents,  $|A_0| = |T| = n$ . In an implementation, if the number of agents in both arrangements is not equal, the redundant agents of the target arrangement can be dropped or agents from the start arrangements can be added to the target arrangement until the number of agents is equal. A feed and retrieval mechanism to change the number of agents in case of  $|A_0| \neq |T|$  is considered in Section 3.7.2. The desired algorithm should result in a sequence of consecutive arrangements

$$\mathcal{P} = (A_0, A_1, A_2, \dots, T), \quad (3.6)$$

of which no arrangement violates any constraints (see Section 3.2.1) in respect to its predecessor.

### 3.2.1 Movement constraints

When an agent is moved, the movement entails a set of constraints which prohibit certain movements parallel to the executed move. Thus when a sphere is attracted by a neighbouring magnet, all other spheres surrounding that magnet must be held in place by their underlying magnet. Additionally when a sphere is moved, all positions neighbouring this sphere can not be a target for other spheres, as the underlying magnet would also attract the currently moving sphere. A comprehensive visualisation of the constraints can be seen in Fig. 3.1. The positions towards no other agent can be moved *relative to the agent* of the moving agent are defined as  $C$ . The positions from which no other agent can be moved away from *relative to the target position* are defined as  $D$ . Both sets contain the same relative positions

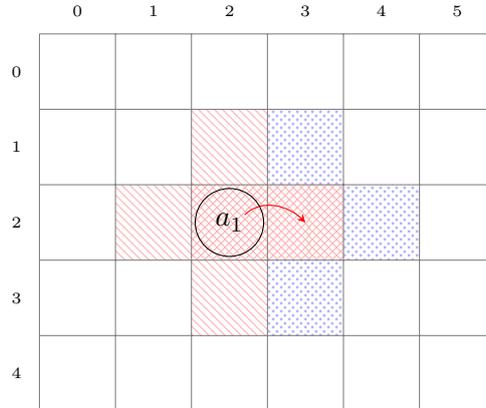
$$D = C = \{\mathbf{0}, \mathbf{e}_x, \mathbf{e}_y, -\mathbf{e}_x, -\mathbf{e}_y\}. \quad (3.7)$$

## 3.3 Role assignment

The first problem to solve is the assignment of each position in the initial arrangement to a position in the target arrangement so that the maximum distance between each initial and target position is minimal (also called the *makespan*) while also avoiding collisions. We model the problem as weighted complete bipartite graph

$$G = (A, T, E) \quad (3.8)$$

with  $n$  agent vertices  $A$ ,  $n$  target vertices  $T$  and the edges  $E$  being weighted by the distance between agent and target (as seen in Fig. 3.2). Instead of using the Euclidean distance as edge weights (as proposed by McAlpine et al. in [7]) the Manhattan distance is used, as spheres can not be moved vertically. The Manhattan distance of two points is



**Figure 3.1:** Visualisation of the constraints which result from a single move. Red dashed cells designate positions  $C$  towards no other agent can be moved, parallel to the current move. Blue dotted cells designate the positions  $D$  from which no other agent can be moved away from, parallel to the current move. In cross-hatched cells neither can happen.

defined as the sum of the absolute differences of their Cartesian coordinates. This leads to a edge weight of

$$w_{ij} = \|a_i - t_j\|_1. \quad (3.9)$$

It is assumed that no two agents and no two targets occupy the same position. A perfect matching with minimal makespan is searched. A perfect matching is a set of edges of which none share a common vertex and every vertex is part of exactly one edge.

To solve this problem a modified variant of the SCRAM (Scalable Collision-Avoiding Role Assignment with Minimal-Makespan) algorithm is applied [7]. SCRAM was developed mostly with teams of homogeneous robots in mind which have to distribute subtasks efficiently. SCRAM describes two general functions which provide a collision avoiding minimum makespan role assignment: *Minimum Maximum Distance Recursive* (MMDR) and *Minimum Maximal Distance + Minimum Sum Distance<sup>2</sup>* (MMD+MSD<sup>2</sup>). Both solve the *linear bottleneck assignment problem* (LBAP) in polynomial time [7]. This is done by transforming the LBAP into the *linear assignment problem* (LAP), through modification of the edge weights. The edge weights of all edges  $E$  are modified so that the weight of any edge  $e$  is greater than the sum of weights of all edges with weight values less than of  $e$ . The LAP is then solved by the Hungarian algorithm (see Section 3.3.1). The algorithmic description can be seen in Algorithm 3.1. MMDR builds upon the Hungarian algorithm, also known as the Kuhn–Munkres algorithm, which is a combinatorial optimization algorithm to optimally solve an assignment problem. In this case we want to assign each sphere on the board a target position so that the makespan is minimized while avoiding collisions. MMDR has the property of being dynamically consistent, meaning that the assignment does not change while the agents approach their targets. The dynamic consistency as well as the method of collision avoidance of SCRAM assumes that the agents move towards their targets in straight lines at equal speed, which is not the case with the moving steel spheres. The spheres approach their targets only in discrete horizontal or vertical steps. In certain conditions they approach their targets at equal speed, performing steps at the same time, yet they also move

alternately if the constraints prohibit a simultaneous movement. Therefore the methods of SCRAM do not apply perfectly on the problem at hand. Nevertheless it is assumed that the assignment is close enough to the optimum to be used in this application. If an assignment method would be used that considered the unequal speed of agents which results from the movement constraints and therefore is implicated in the actual motion planning, the role assignment and motion planning would have to be intertwined again, which as a result would enormously increase the search space.

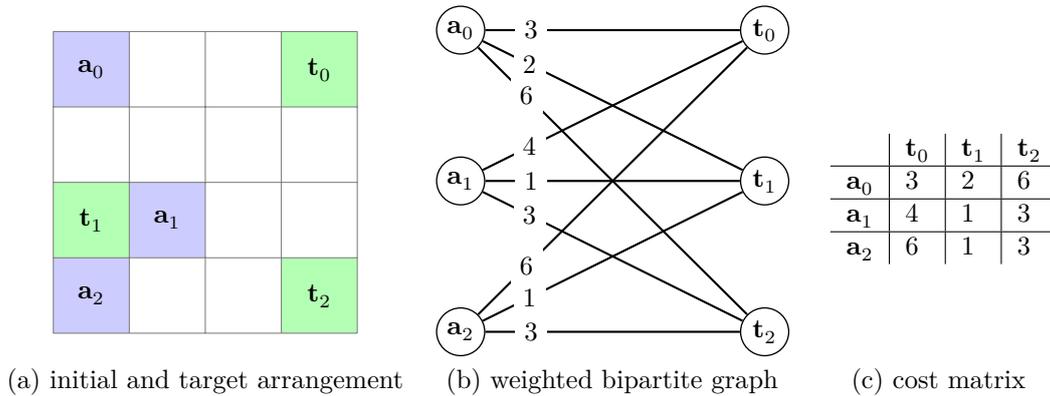
### 3.3.1 Hungarian algorithm

Each sphere in the start arrangement should be assigned a target position in the target arrangement so that the *maximum distance* between any sphere and its target is minimal. The generalized form of this combinatorial optimization problem is commonly defined as *linear bottleneck assignment problem*. There is a more common problem which is called the *linear assignment problem*. The *linear assignment problem* requires every agent to be assigned to a role so that the *sum of costs* is minimized. In this case the cost is the distance from an agent to its target. The Hungarian algorithm solves the *linear assignment problem* in polynomial time [7]. The Hungarian algorithm on its own would offer a solution to the assignment problem at hand, yet it would only minimize the sum of all distances but not minimize the maximum distance. The maximum distance would ultimately determine the performance of a found assignment, as a new arrangement is only complete if the last agent reaches its target and the last agent is expected to be the one most distant from its target. The problem can be explained more informally through the following example. The linear assignment problem asks for the best assignment for a group of people which have to be assigned to a set of jobs, where the possible assignments are ranked by the sum of their rated skill level [3]. A solution to the problem can easily be transformed to minimize the summed score, which is needed for getting the assignment with the minimum distance. A comprehensive description of the Hungarian algorithm can be found in [3] as the full algorithm is too extensive to be covered here at the full length.

## 3.4 Motion planning

In the previous step a position for each sphere to navigate to has been determined. Secondly a sequence of arrangements has to be found of which none violates any constraints when compared to its preceding arrangement. The last element of the sequence has to be the target arrangement  $T$  (see Equation 3.6).

The method *Increasing Cost Tree Search* (ICTS) described by Sharon et. al. solves the MAPF problem optimally [10]. ICTS seeks to minimize the sum of all distances travelled, which is not optimal for the problem at hand, as here it is sought to minimize the maximum distance travelled. ICTS is part of a class of methods called the *global search approach*. Sharon et. al. state that this usually very efficient method is less efficient than A\* in very dense environments. Considering the problem of this thesis the plane on which the search is conducted is expected to be very dense, meaning that there are potentially many emerging collisions. Therefore this method is likely to be not optimal for this specific problem.



**Figure 3.2:** An initial arrangement  $A_0 = (a_0, a_1, a_2)$  and a target arrangement  $T = (t_0, t_1, t_2)$  is shown in (a). The same problem modelled as a relationship of nodes represented by a weighted bipartite graph (b) and as a cost matrix (c), both holding the same information.

In this case a variant of  $A^*$  is used which accepts a non-optimal solution. After finding a solution  $A^*$  would continue to open nodes, until it has assured that the optimal solution has been found. The following algorithm does stop with the first found solution.

### 3.5 Algorithmic description

A search tree is built of which the nodes are arrangements. The root node is the initial arrangement  $A_0$ . When a node is expanded, all possible next arrangements reachable by a single progression are generated. These are reduced by checking each consecutive arrangement for consistency with the defined constraints. A heuristic function is defined to enable us executing an informed search method, determining which node to expand next.

Initially an assignment is found through Algorithm 3.1. Then the principles of the  $A^*$  search method are applied in Algorithm 3.2 resulting a solution. The expansion of nodes is done in Algorithm 3.3, while the legality of expanded nodes is checked in Algorithm 3.4.

#### 3.5.1 Heuristic Distance

In finding a solution for problems with a search spaces too big to be examined in its entirety, heuristics are essential. As stated in [5], heuristics are applied in one of two situations. Firstly if a problem has no exact solution and heuristics help to choose a solution which has a higher probability of being correct. Secondly, as it is the case here, if the problem has an exact solution but the calculation costs are far out of reach. Basically a heuristic is an educated guess on what path towards the solution is the best. The path planning problem stated here has an optimal solution, but finding it exhaustively by trying all possible solutions is infeasible considering the technology we have at hand. We define a heuristic for this path finding problem as the maximum Manhattan distance

between any assignment of initial and target position of an arrangement as

$$h(A, T) = \max_{i \in n} (\|\mathbf{a}_i - \mathbf{t}_i\|_1). \quad (3.10)$$

In this way the cost to reach the target is never overestimated and the heuristic is therefore *admissible*. An admissible heuristic ensures that we can find the optimal route.

For this application yet another heuristic is used. It is the sum of the Manhattan distances between each pair of initial and target position

$$h(A, T) = \sum_{n=0}^{n-1} \|\mathbf{a}_i - \mathbf{t}_i\|_1. \quad (3.11)$$

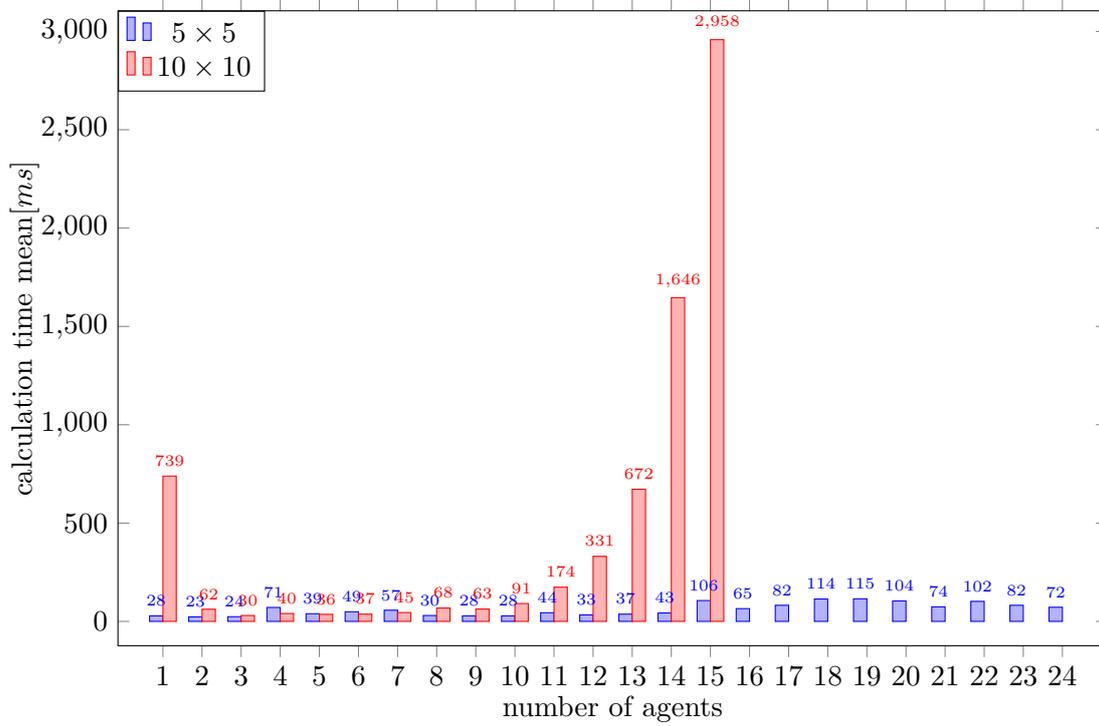
Even though this heuristic is not admissible, possibly leading to no optimal solution, it is used because it is more fine grained. The heuristic in Equation 3.10 often leads to no change in the heuristic value, giving no information if the arrangement has moved closer to its target. This furthermore leads to many additional expansions, costing computation time.

### 3.5.2 Progression combinatorics

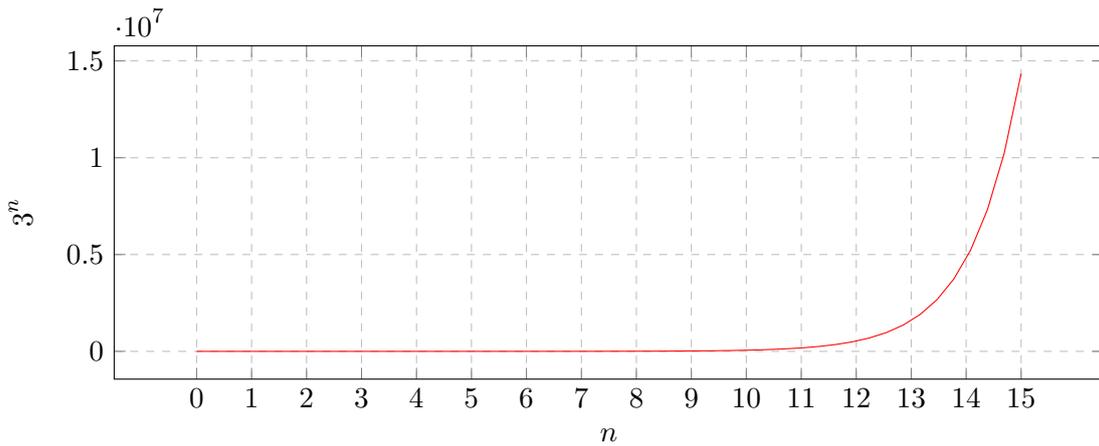
What makes the described problem increasingly complex is the fact that multiple spheres can be moved at once. If we considered only one single movement per time step the branching factor would be  $2n$ , as we would consider a maximum of 2 directions per agent which bring the agent nearer to its target. In our case we have a branching factor of  $3^n$ , considering 3 possible next states per agent in every combination with all other agents (the third state being the agent keeping its position). Even though the whole state space is greatly reduced by the given constraints, it is generally still too big to be explored by brute force, i.e. through a breadth-first search. We need to find possible progressions succeeding an arrangement. In those every agent can do one of five movements  $\{up, down, left, right, none\}$ . This leads to a maximum number of possible successive progressions  $5^n - 1$ , with  $n$  being the number of agents. The number is reduced by the board size and movement constraints. Through the high number of possible progressions it is not practical to find all progressions exhaustively. We limit the number of movements to those which decrease the Manhattan distance  $d$  or keep it equal, while *none* being the only move which does not change  $d$ . The maximum number of possible successive progression then still is  $3^n - 1$ .

## 3.6 Performance

To confirm the applicability of the proposed algorithm a performance test is conducted. The test series are performed on a board with size  $5 \times 5$  and  $10 \times 10$ . On the  $5 \times 5$  board all possible numbers of agents were agents still have the possibility to move are tested (1 to 24). The tests predicate that the algorithm is perfectly applicable for a board size of  $5 \times 5$ , as the maximum mean calculation time for path finding is 115 milliseconds (see Fig. 3.3). Yet when using a  $10 \times 10$  board the algorithm soon becomes impractical when approaching a 3 seconds mean when calculating a path for 15 agents. This is due the high number of agents having many possible movement direction as a result of having a large grid.



**Figure 3.3:** Results for the calculation time mean for finding the path between 500 randomized pairs of arrangements on the board sizes  $5^2$  and  $10^2$ . The high calculation time for one agent stems from a software bug which leads to an extensive but unnecessary exploration of nodes when only considering a single agent.



**Figure 3.4:** Plot of the branching factor  $b = \mathcal{O}(3^n)$  for  $n$  agents. The growth is also reflected in the calculation time seen in Fig. 3.3. When the number of nodes exceeds  $10^6$  it is considered as infeasible, as the profiling results in Fig. 3.3 suggest.

---

**Algorithm 3.1:** Finds a minimum makespan role assignment. This function is the MMDR  $O(n^5)$  polynomial time implementation as described by McAlpine et al. [7]. It rearranges target positions  $T$  so that their index corresponds with the indices of their assigned agents in  $A$ . The weight of the edges used to sort them is defined as the Manhattan distance  $\|\overline{\mathbf{a}_i, \mathbf{t}_j}\|_1$  between the edge vertices. The Hungarian algorithm used in this algorithm returns a perfect matching which minimizes the summed distances. An implementation of it is considered too extensive to be treated here. As it is a widely used algorithm it shall be referred to a comprehensive explanation in [3].

---

```

1: procedure RoleAssignment( $A, T$ )
   Input:  $A = (\mathbf{a}_0, \dots, \mathbf{a}_{n-1}), T = (\mathbf{t}_0, \dots, \mathbf{t}_{n-1})$  where  $\mathbf{a}_i = (x_i, y_i), \mathbf{t}_j = (x_j, y_j)$ 
   Returns  $T' = (\mathbf{t}'_0, \dots, \mathbf{t}'_{n-1})$  where  $T'$  is a permutation of  $T$ 
2:    $Edges \leftarrow \{\overline{\mathbf{a}_0, \mathbf{t}_0}, \overline{\mathbf{a}_0, \mathbf{t}_1}, \dots, \overline{\mathbf{a}_{m-1}, \mathbf{t}_{m-1}}\}$ 
3:    $edgesSorted \leftarrow \text{SortAscendingDistance}(Edges)$ 
4:    $lastDistance \leftarrow -1$ 
5:    $rank \leftarrow 0$ 
6:    $currentIndex \leftarrow 0$ 
7:   for  $e \in edgesSorted$  do
8:     if  $\|e\| > lastDistance$  then
9:        $rank \leftarrow currentIndex$ 
10:       $lastDistance \leftarrow \|e\|$ 
11:       $\|e\| \leftarrow 2^{rank}$ 
12:       $currentIndex \leftarrow currentIndex + 1$ 
13:   return HungarianAlg( $edgesSorted$ ) ▷ returns a set of edges
14: end procedure

```

---

```

15: procedure HungarianAlg( $A, T$ )
16:   ... ▷ for a full implementation see [3].
17:   return B ▷ B is a permutation of A
18: end procedure

```

---

## 3.7 Further considerations

### 3.7.1 Multiple sphere types

When considering multiple, non-interchangeable types of spheres on the board, the anonymous MAPF expands to a non-anonymous MAPF problem. Since the role assignment procedure (see Section 3.3) transforms the path finding problem to a MAPF anyway, it may seem that the path planning procedure must not be modified to solve a problem with non-interchangeable spheres. Yet the path planning implementation only considers agents moving towards their target position. When having multiple sphere types spheres have to get out of the way to let certain spheres pass, which leads to spheres having to move away from their targets. Moving away from targets is not considered in the described algorithm and therefore applying it on non-anonymous agents can lead to dead ends. An algorithm addressing the problem of path finding for teams of agents is described by Hang Ma and Sven Koenig [6]. It could be adopted for the

---

**Algorithm 3.2:** Finds sequence of arrangements satisfying the given constraints to reach target arrangement  $A'$  starting with initial arrangement  $A$ . The mapping  $V : A \rightarrow v$ , where  $v$  is the depth of its corresponding node in the search tree, contains all visited nodes.  $w$  denotes the width of the board and  $h$  its height.

---

```

1: procedure FindPath( $A, T$ )
   Input:  $A = (\mathbf{a}_0, \dots, \mathbf{a}_{n-1}), T = (\mathbf{t}_0, \dots, \mathbf{t}_{n-1})$  where  $\mathbf{a}_i = (x_i, y_i)$  and  $\mathbf{t}_j = (x_j, y_j)$  where  $0 \leq x_i, x_j < w, 0 \leq y_i, y_j < h$ .
   Returns  $\mathcal{P} = (A_0, \dots, A_{k-1})$ , a sequence of arrangements (the found path) where  $A_0 \hat{=} A, A_{k-1} \hat{=} T$ .
2:  $T' \leftarrow$  RoleAssignment( $A, T$ ) ▷ returns permutation of  $T$ 
3:  $c \leftarrow \langle \text{data} = A, \text{depth} = 0, \text{children} = \emptyset \rangle$ 
4:  $L \leftarrow \emptyset$  ▷ list of open nodes
5:  $V \leftarrow \emptyset$  ▷ map of visited nodes
6: while  $c.\text{data} \neq T'$  do
7:    $c.\text{children} \leftarrow$  GenerateProgressions( $c.\text{data}, T', V, c.\text{depth}$ )
8:   for  $c' \in c.\text{children}$  do
9:      $c'.\text{heuristic} \leftarrow$  Heuristic( $c'.\text{data}, T'$ ) ▷ see equation 3.11
10:     $V \leftarrow V \cup \langle c', c.\text{depth} + 1 \rangle$ 
11:     $L \leftarrow L \cup c'$ 
12:     $c \leftarrow \text{argmin}_{c' \in L} (c'.\text{heuristic} + c'.\text{depth})$ 
13:     $L \leftarrow L \setminus c$  ▷ remove  $c$  form  $L$ 
14:     $T' \leftarrow$  RoleAssignment( $c, T$ ) ▷ returns permutation of  $T$ 
15:   return  $c.\text{data}$ 
16: end procedure

```

---

problem at hand.

### 3.7.2 Feed and retrieval

The initiating vision of the machine resulted from the idea of having a Go board which autonomously plays Go against a human opponent. Apart from needing multiple sphere types, spheres have to be transported onto and off the display grid. This could be done in multiple ways. The user could be ordered to add or remove certain spheres, what I consider as a not very elegant solution. Another possibility could be a rail around the board on which spheres could be parked. An additional magnet on the side serving as a exit port could move spheres onto the rail. The rail is slanted so the spheres roll towards an entry port. The entry port has a controllable barrier that can hinder or enable spheres to enter the board. Two types of spheres would also require two separate rails.

---

**Algorithm 3.3:** Given an arrangement  $A$  and a target arrangement  $T$  this function returns all allowed successive arrangements following  $A$  where  $d_i \geq d_i^*$ , with  $d_i$  being the distance between an agent  $a_i$  and its target position  $t_i$  and  $d_i^*$  being the distance between the new position  $a_i^*$  and  $t_i$ . Arrangements which have been visited at shallower depths are excluded by checking against a visited map  $V$  which holds a depth value  $v$  for each key  $B$  which is a visited arrangement.  $t$  denotes the search tree depth of the current node.

---

1: **procedure** `GenerateProgressions`( $A, T, V, t$ )  
    **Input:**  $A = (\mathbf{a}_0, \dots, \mathbf{a}_{n-1})$ ,  $T = (\mathbf{t}_0, \dots, \mathbf{t}_{n-1})$  and  $V : \mathbf{A} \rightarrow K$  where  $\mathbf{A}$  is an anonymous arrangement (see equation 3.3) and  $v$  the depth of the arrangements corresponding node in the search tree.  
    **Returns**  $\mathcal{P} = (A_0, \dots, A_{m-1})$  which is a sequence of arrangements which each form progressions with its predecessor.

At first all possible movements where  $d_i \geq d_i^*$  are generated.

2: **for**  $i \leftarrow 0, \dots, n-1$  **do** ▷ iterate over all  $n$  agents  
    3:  $M_i \leftarrow (\mathbf{a}_i)$  ▷  $M_i$  holds possible successor positions of agent  $\mathbf{a}_i$ .  
    4: **if**  $x_i < x'_i$  **then**  
       5:  $M_i \leftarrow M_i \cup (\mathbf{a}_i + (1, 0))$   
    6: **else if**  $x_i > x'_i$  **then**  
       7:  $M_i \leftarrow M_i \cup (\mathbf{a}_i - (1, 0))$   
    8: **if**  $y_i < y'_i$  **then**  
       9:  $M_i \leftarrow M_i \cup (\mathbf{a}_i + (0, 1))$   
    10: **else if**  $y_i > y'_i$  **then**  
       11:  $M_i \leftarrow M_i \cup (\mathbf{a}_i - (0, 1))$

Now all combinations of all generated movements are created.

12:  $q \leftarrow (q_0, \dots, q_{n-1}) = (0, \dots, 0)$  ▷ indices for iterating over all movements  
    13:  $f \leftarrow \text{false}$  ▷ carryover flag  $f$   
    14: **while**  $\neg f$  **do**  
       15:  $f \leftarrow \text{true}$   
       16:  $A^* \leftarrow ()$   
       17: **for**  $i \leftarrow 0, \dots, n-1$  **do**  
           18:  $A^* \leftarrow A^* \cup M_i(q_i)$   
           19: **if**  $f$  **then**  
               20:  $q_i \leftarrow q_i + 1$   
               21:  $f \leftarrow \text{false}$   
               22: **if**  $q_i = |M_i|$  **then**  
                   23:  $q_i \leftarrow 0$   
                   24:  $f \leftarrow \text{true}$  ▷ set carryover flag  
       25: **if** `IsLegal`( $A, A^*, V, t$ ) **then**  
           26:  $\mathcal{P} \leftarrow \mathcal{P} \cup A^*$   
    27: **return**  $\mathcal{P}$   
 28: **end procedure**

---

---

**Algorithm 3.4:** Ensures a arrangement  $A^*$  is legal, considering its predecessor  $A$ , the defined constraints  $B$  and  $D$  (see Section 3.2.1), the already visited arrangements  $V$  and the current search tree depth  $t$ .

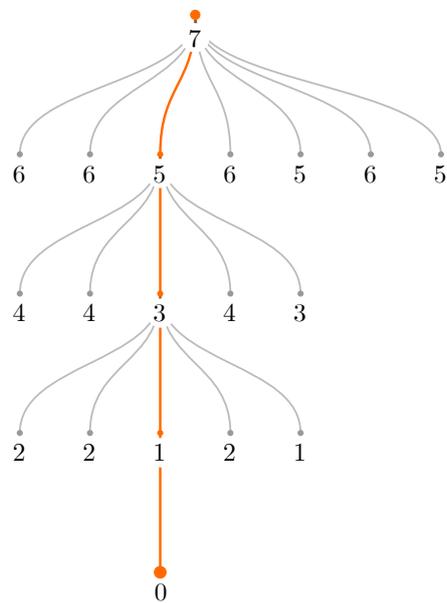
---

```

1: procedure IsLegal( $A, A^*, V, t$ )
   Input:  $A := (\mathbf{a}_0, \dots, \mathbf{a}_{n-1})$  where  $\mathbf{a}_i := (x_i, y_i)$ ,  $A^* := (\mathbf{a}_0^*, \dots, \mathbf{a}_{n-1}^*)$  where
    $\mathbf{a}_i^* := (x_i^*, y_i^*)$ ,  $V : A \rightarrow v$  where  $A$  is an arrangement and  $v$  the depth of its
   corresponding node in the search tree.  $t$  is the depth of  $A^*$  in the search tree.
   Returns {true, false}
2:  $C \leftarrow D \leftarrow \{(0, 0), (0, 1), (1, 0), (-1, 0), (0, -1)\}$ 
3: for  $i \leftarrow 0, \dots, n-1$  do
4:   for  $j \leftarrow i+1, \dots, n-1$  do
5:     if  $\mathbf{a}_i^* = \mathbf{a}_j^*$  then
6:       return false ▷ two agents occupy the same position
7:   if  $A^* \subset V$  and  $V(A^*) < t$  then
8:     return false ▷ arrangement was already visited
9:    $\mathbf{A}^B \leftarrow [-1]_{w \times h}$ 
10:   $\mathbf{A}^D \leftarrow [-1]_{w \times h}$ 
11:  for  $i \leftarrow 0, \dots, n-1$  do ▷ populate constraint map
12:    if  $\mathbf{a}_i \neq \mathbf{a}_i^*$  then
13:      for all  $(d_x, d_y) \in B$  do
14:         $(x, y) \leftarrow (x_i + d_x, y_i + d_y)$ 
15:        if  $0 \leq x < w \wedge 0 \leq y < h$  then
16:           $\mathbf{A}_{x,y}^B \leftarrow i$ 
17:         $(x, y) \leftarrow (x_i^* + d_x, y_i^* + d_y)$ 
18:        if  $0 \leq x < w \wedge 0 \leq y < h$  then
19:           $\mathbf{A}_{x,y}^D \leftarrow i$ 
20:  for  $i \leftarrow 0, \dots, n-1$  do ▷ check against constraint map
21:    if  $\mathbf{a}_i \neq \mathbf{a}_i^*$  then
22:       $(x, y) \leftarrow (x_i, y_i)$ 
23:      if  $\mathbf{A}_{x,y}^D \neq i$  then
24:        return false
25:       $(x, y) \leftarrow (x_i^*, y_i^*)$ 
26:      if  $\mathbf{A}_{x,y}^B \neq i$  then
27:        return false
28:  return true
29: end procedure

```

---



**Figure 3.5:** A search tree as generated by the described algorithm using the arrangements in Fig. 3.2 as inputs. The numbers at each node describe the heuristic cost to reach  $T$ . The orange edges represent the found path.

## Chapter 4

# Position Detection

When moving the spheres on the board their position has to be known so that the correct magnets can be activated. Through user interaction and uncertainty if the designated move has been executed as intended, the screen state can not be determined only by inference. Multiple approaches have been thought of to solve the problem of sensing the sphere positions.

### 4.1 External camera

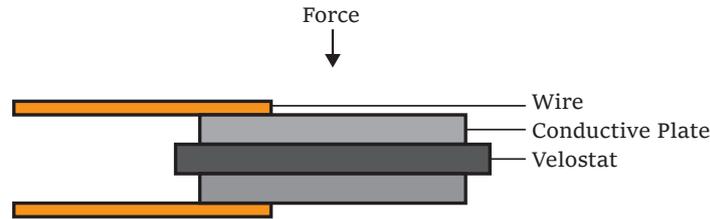
An external camera could provide information about the position of the spheres. A feature detection algorithm thereby determines the state of the board. This approach is dependent on external factors as having enough available light and no interfering objects in the line of sight. This approach is not favoured because it prevents the machine from being an enclosed machine and limits its operability.

### 4.2 Photoresistors

Photoresistors could be placed on top of each magnet. Light shining through a small hole can then inform about the presence of a sphere. This approach is also dependent on external light. Infrared light sources could be placed besides the sensors to provide the needed light. External light could then be a disturbing factor and could lead to detection errors. The height of the sensors could impose a problem as the distance between magnet and sphere should be held minimal.

### 4.3 Force-sensitive resistors

A force-sensitive resistor (FSR) is a material whose resistance changes when pressure is applied. Force-sensitive resistors could measure the weight the spheres are exerting on each magnet. FSR technology is often used in touch-sensitive electronics. FSR sensors can be manufactured very easily by layering a conductive plate, a force sensitive material like velostat, and another conductive plate (see Fig. 4.1). For this machine the sensors can be arranged in a grid layout. To minimize the number of needed wires, the lower conductive sensor areas can be connected horizontally and the upper areas vertically.



**Figure 4.1:** The structure of an FSR sensor with velostat as the force sensitive material. The higher the force which acts upon the material, the lower is the resistance which is measured between the two wires.

A multiplexer can then drive the column of the sensor which should be measured with a positive voltage while another multiplexer links the respective row with an analog-digital-converter. This sensor setup is visualized in Fig. 4.2. The insulators and their copper layers can be manufactured as thin printed circuit boards. A big advantage of this method is that the sensor is very thin and therefore keeps the distance between sphere and magnet minimal. Flexible circuit boards are available down to a thickness of 0.065 mm, enabling a very thin sensor<sup>1</sup>. As this sensor is not dependent on sensing a magnetic field, other non-magnetic objects could exert force on the sensors. This fact can be viewed as favourable, as the user can interact with the machine as with a touchscreen. A downside of this approach is that it can lead to false positives when a board position is encircled by spheres on each side, so that the stiffness of the overlaying board will transfer the weight also on the centred position.

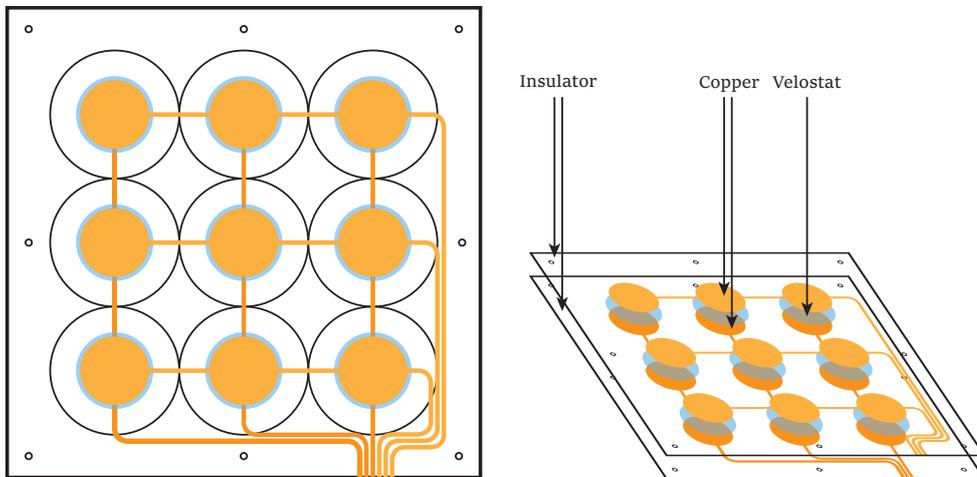
#### 4.4 Capacitive sensing

Capacitive sensing is another popular technology to implement touch-sensitive electronics. In their most basic configuration capacitive sensors are even easier to manufacture as FSRs, as no additional material is needed that is not available on a common printed circuit board anyway. A capacitor basically consists of two conductors separated by an insulating material (i.e., the dielectric) [17]. Therefore a capacitor can be simply be created by using a multi-layered PCB. A capacitive sensor usually does not measure the charge stored in the capacitor, nor capacitance but the change in capacitance over time. This can be a problem when static spheres should be detected.

#### 4.5 Hall sensors

Placing Hall sensors (magnetic field sensors) on top of each magnet to determine if a sphere is present. A Hall effect sensor is a transducer that varies its output voltage in response to a magnetic field. Its linear sourcing output voltage is set by the supply voltage and varies in proportion to the strength of the magnetic field. The concept is that a magnetic metal sphere changes a present electromagnetic field. In contrast to the

<sup>1</sup><http://www.pcbuniverse.com/pcbu-flex-pcb.php>



**Figure 4.2:** A sensor with a grid layout. Multiplexers are used to apply a positive voltage to one column at a time. For every column the voltage on every row is measured.

FSR method mentioned above, this can only detect objects that influence the magnetic field. In combination with the FSR the machine could discern non-magnetic objects from magnetic ones, determining if a user touched the board or if a sphere has been placed.

An experiment to test this effect can be seen in Fig. 5.3. The approach using magnetic field sensors is promising because it is not dependent on external factors as enough available light. Additionally the sensors are not visible from outside. A SMD variant of the sensor should be chosen because of its small height to achieve a very small distance between electromagnet and sphere.

## 4.6 Current sensing

A changing magnetic field induces a current in a conductor. Knowing that an attracted sphere changes the electromagnetic field, we can assume that a change in current on the power line of the electromagnet can be measured. Through this an attracted sphere could be detected without the need of any additional sensors on top of the magnets. A downside of this method is that static spheres, which are already occupying the magnet, can not be detected.

## Chapter 5

# Simulations and Experiments

The initial idea of the machine demanded a quick and cost effective way of determining a rough estimation of its feasibility. Another reason for conducting simulations and experiments was to approximate the optimization problem arising from the core principle. When wanting to minimize the effort needed to move a sphere to a new position many parameters have to be optimized:

- the size of the electromagnet and its windings,
- the voltage with which the electromagnets are excited,
- the size and wall thickness of the hollow steel spheres and
- the thickness and surface material of the top plate.

The interdependence of many of these parameters makes it exceptionally difficult to determine an optimum. The most critical moment is when a sphere should be set in motion. Once the sphere is in motion, the electromagnet has no difficulties to move it towards its position. A voltage spike in the beginning could be enough to set the sphere in motion, so that subsequently a lower voltage level could drive the magnet so that the generated heat and the needed current is held at a minimum.

### 5.1 Sphere parameters

Using hollow metal spheres for the display is expected to have two advantages compared to solid spheres. Firstly, a lower weight of the sphere means that less magnetic force has to be exerted to attract it. The second expected advantage could be that the different inner structure and mass of the sphere could result in different readings on the Hall effect sensor. This means that the sensors could not only detect if a sphere lies on top of a magnet but could also distinguish different types of spheres, which in turn can have distinct colors.

A simulation is conducted in which the differences in force acting upon a solid and a hollow steel sphere are compared. The simulation determined how much the force that acts upon the sphere lessens in proportion to the lowered mass. The simulation setup can be seen in Fig. 5.1. The results of the simulation can be seen in Table 5.1.

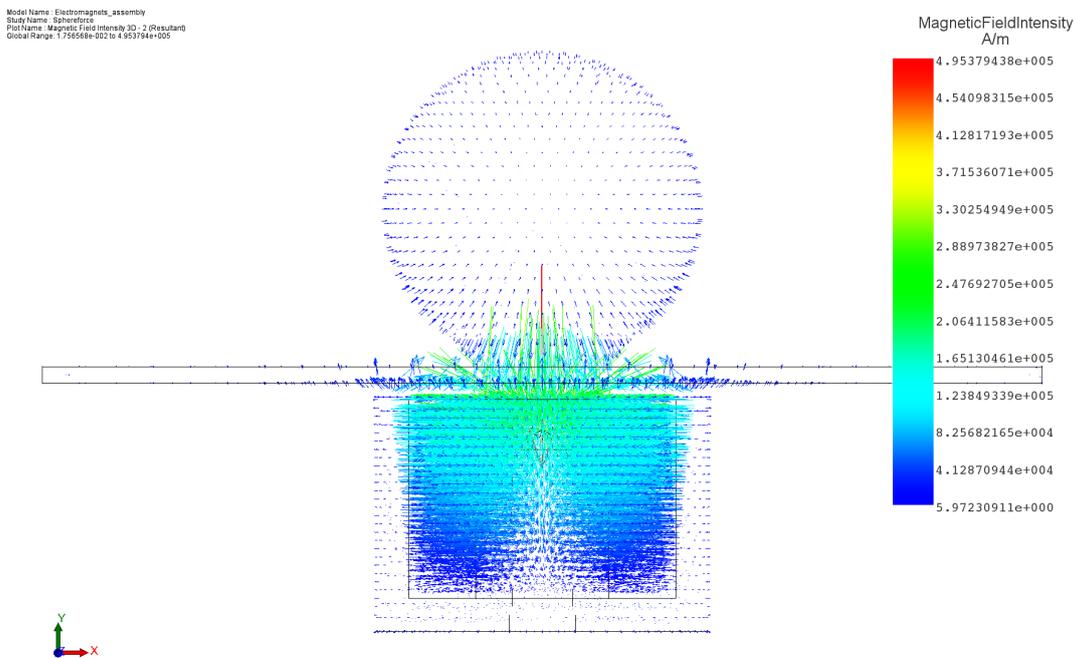
### Simulation parameters

Solid/hollow sphere diameter:	19 mm
Hollow sphere wall thickness:	1 mm
Plate thickness:	1 mm
Coil voltage:	24 V
Coil resistance:	49 $\Omega$
Coil windings:	1600

### Simulation results

Force solid sphere	5.571 N
Force hollow sphere	5.412 N

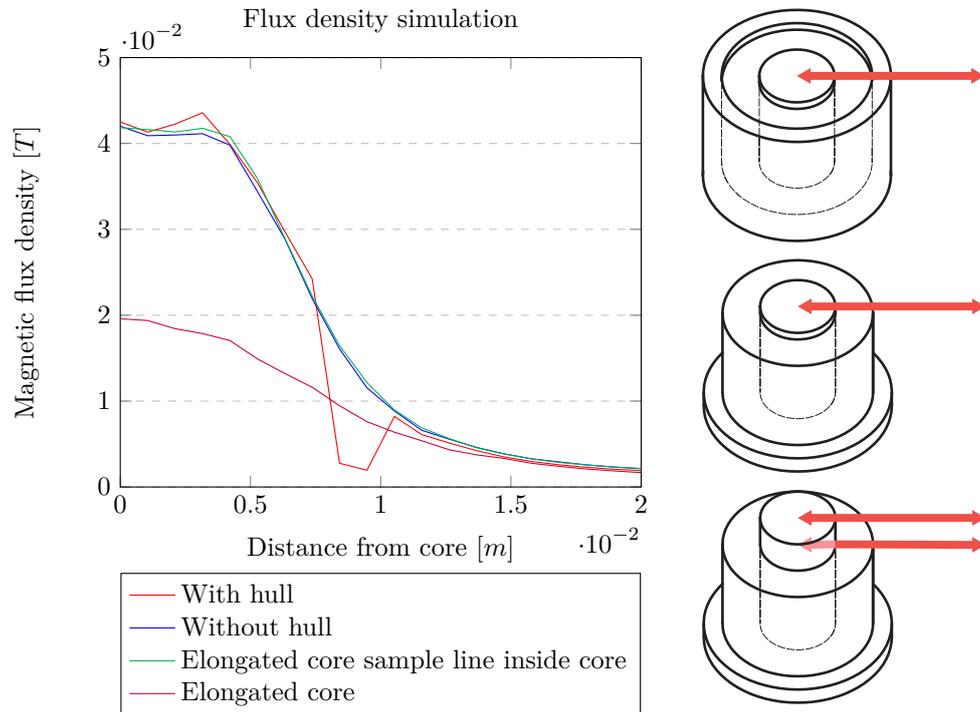
**Table 5.1:** The simulation parameters and results. It is apparent that the attraction force differs only slightly comparing solid and hollow spheres.



**Figure 5.1:** Magnetic field intensity of the solid sphere simulation. Plot generated with EMS [15].

## 5.2 Magnet parameters

The magnetic field of the electromagnets has to be shaped in a way so that horizontally neighbouring spheres can be easily attracted. Through electromagnetic simulations the influence of the electromagnets shape on the shape of the magnetic field has been examined. The results of the simulations can be seen in Fig. 5.2. It is apparent that neither the removal of the hull nor the elongation of the core of the electromagnet change the

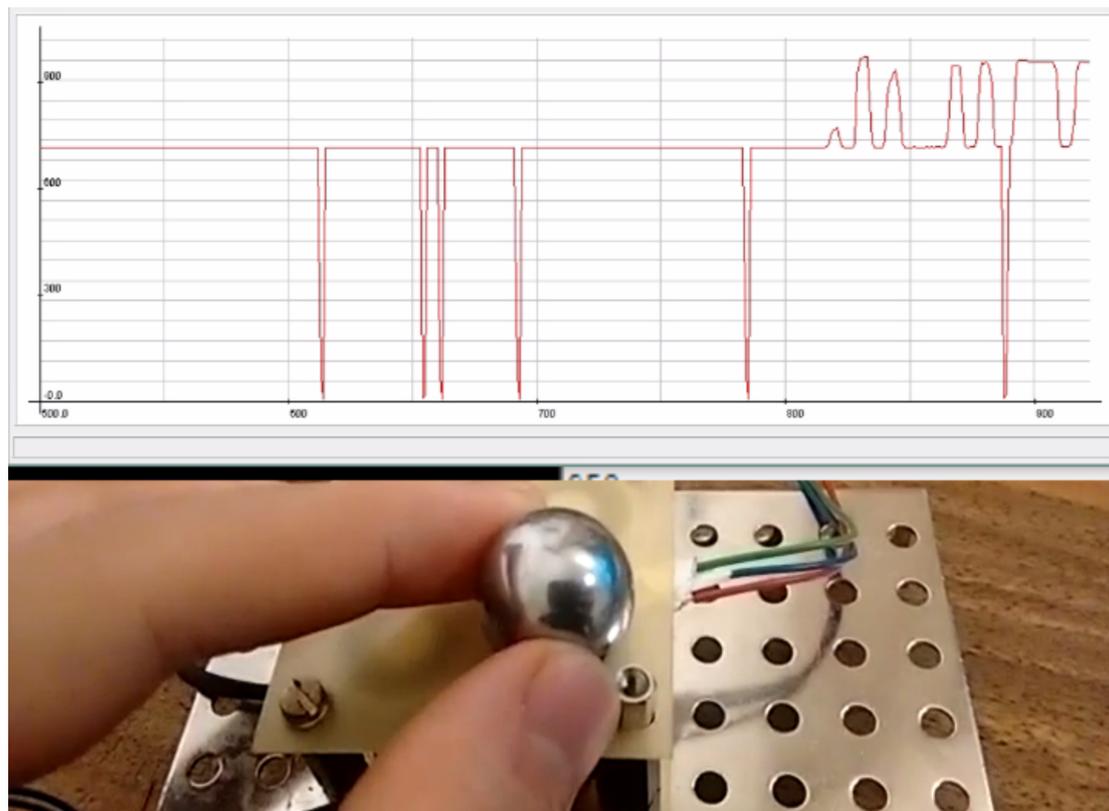


**Figure 5.2:** A simulation of an electromagnet using a coil with 5000 turns, with a measured resistance of  $49 \Omega$  powered by 12 V. The red line represents the line on which the simulation data is sampled (on top of the core). The amount of magnetic flux is expected to be equivalent to the force with which it acts upon a potential ferromagnetic object.

shape of the simulated electromagnetic field in a manner so that it is beneficial for the here desired attraction of a planarly displaced steel sphere.

### 5.3 Sphere detection

Three methods to detect the position of the spheres have been tested. The influence of a steel sphere on the electromagnetic field measured by a Hall sensor proved to be feasible. The test setup can be seen in Fig. 5.3. Also a custom force sensitive resistor (FSR) has been tested, which could be extended to a matrix. It consists of a non-conductive base material (a 0.4 mm epoxy plate), a conductive copper foil and the conductive material velostat which changes its resistance dependent on the pressure which is acts upon it. Fig. 4.1 shows the structure of an FSR sensor.



**Figure 5.3:** Testing the influence of a hollow steel sphere on the magnetic field measured by a Hall sensor placed at the center of a electromagnet which is powered by 7V. The sharp signal drops are errors in the visualisation and do not reflect the collected sensor data.

## Chapter 6

# Software

This project demanded multiple software elements: the path finding algorithm, the control logic for converting the found path to a series of magnet intensity curves, the magnet driver logic for interfacing with the pulse width modulation (PWM) driver, the exchangeable application logic and additionally a web application for demonstrating the path finding algorithm.

### 6.1 Path finding

The path finding algorithm was written in C++ for a number of reasons: It is both suited for embedded systems and larger systems. If the algorithm would have a small footprint it could be ported to run on a microcontroller. The speed of this compiled language allows an efficient implementation. Additionally if the application is programmed in another language, there exist C++ bindings for almost all popular languages to call the path finding software from there. The path finding implementation in C++ used following libraries:

- *Google Test*<sup>1</sup>, a C++ test framework for unit testing,
- *Easylogging++*<sup>2</sup>, a single header efficient logging library and
- *JSON for modern C++*<sup>3</sup>, a library for serializing the path finding solutions for transferring them to other languages.

In retrospect, I would rather write the path finding algorithm in Python for faster iterations and more compact and readable code. If there had been major performance issues, the relevant parts could have been implemented in Cython. The Cython language is a superset of the Python language that additionally supports calling C functions and declaring C types on variables and class attributes.<sup>4</sup>

---

<sup>1</sup><https://github.com/google/googletest>

<sup>2</sup><https://github.com/muflihun/easyloggingpp>

<sup>3</sup><https://github.com/nlohmann/json>

<sup>4</sup><http://cython.org/>

## 6.2 Control logic

The user should be able to input an image whereupon the spheres are rearranged so that they represent the input image. The input image is required to be a two-dimensional bicoloured black-and-white bitmap with the same resolution as the display. Additionally the number of black, or optionally white pixels, of the image has to be equal to the number of spheres on the display. The provided API provides the possibility to input a set of positions instead of providing an image. This method is nearer to the specific handling of the information in the path finding algorithm. The image approach can be added for convenience.

## 6.3 Hardware driver

The signal processing and control of the individual electromagnets in the first two prototypes was done on the microcontroller *Arduino UNO*. The software for the *Arduino UNO* was written using the *Arduino IDE*, which uses a set of C/C++ functions.

## 6.4 Web application

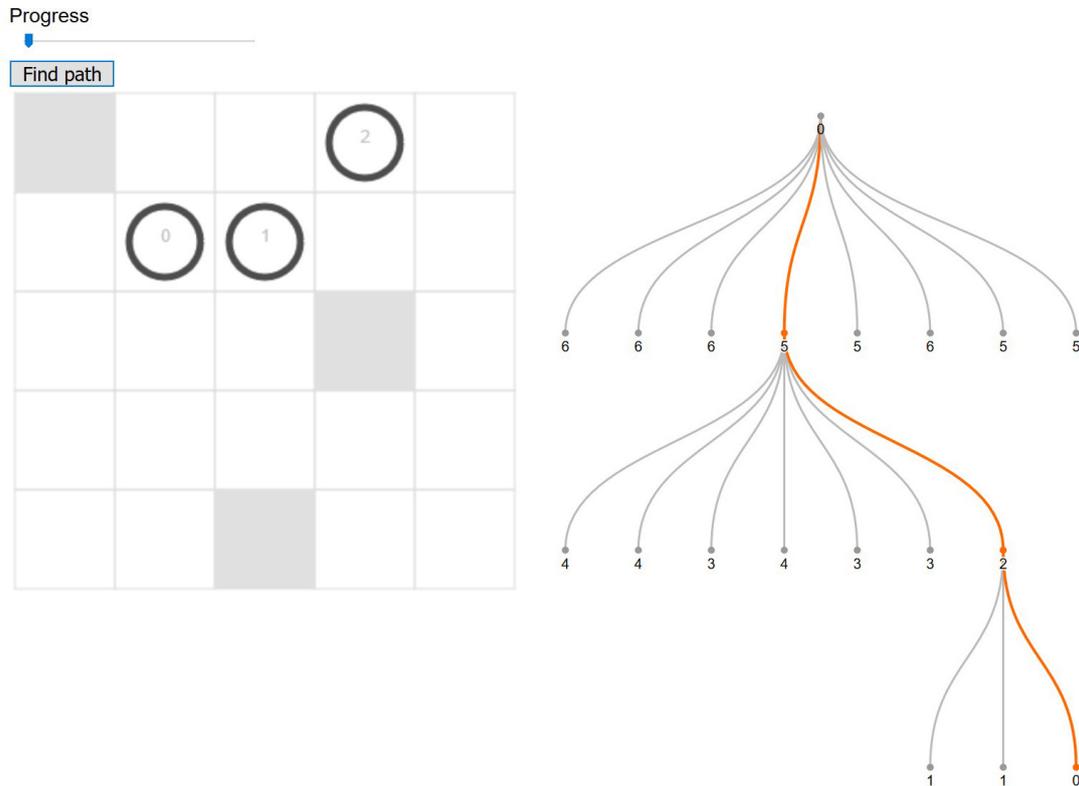
For testing the algorithm in a convenient manner a web application was implemented (see Fig. 6.1). It runs in a *Node.js* environment using *Express.js* for serving the website. Using *node-gyp* the application is extended with a native C++ module which contains the path finding implementation. The visualisation of the search tree is done using the data visualisation library *D3.js*.

## 6.5 Sequential control system

Given a valid progression, an intensity curve for each electromagnet has to be determined to execute that progression. Those assigned intensity curves are chosen from a set of 5 predefined curves

- PULL,
- RELEASE,
- HOLD,
- RHOLD (Release Hold) and
- OFF.

The definition of the listed intensity curves is shown visually in Fig. 6.2 and in numbers in Program 6.1. To determine which magnet should perform which intensity curve, the start arrangement  $A$  and the target arrangement  $A'$  are needed as inputs.  $A'$  is a valid progression from  $A$ , meaning it does not violate any constraints as defined in Section 3.2.1. If  $a_i \neq a'_i$ , it means that the sphere at position  $a_i$  should be moved to position  $a'_i$ . So the magnet at position  $a'_i$  is assigned the PULL curve and the magnet at position  $a_i$  is assigned the RELEASE curve. If  $a_i = a'_i$  it means that the sphere at that position should stay in place, so the HOLD curve is assigned. If the holding magnet has a horizontally or vertically neighbouring magnet which releases a sphere, meaning  $a_i$



**Figure 6.1:** A screenshot of the web application which was implemented to test and visualize the path finding algorithm. The left side shows the board with the agents depicted as circles and their targets drawn as grey rectangles. The tree on the right shows the search tree that was generated in the process of finding the path which is marked in orange. By clicking on the nodes of the search tree the arrangement which it represents is shown on the left. The numbers beneath the nodes denote their heuristic distance to the target arrangement.

where  $a_i \neq a'_i$ , the RHOLD (release hold) curve is assigned to the holding magnet. In this way the holding magnet releases the sphere for a few seconds so that the moving sphere has time to gain momentum so is less affected by the holding magnet. All other magnets which had received none assignment are assigned the OFF curve. After the assignment all curves are executed simultaneously, powering the magnets according to the intensity at the point in time on the curve.

## 6.6 Third-party software

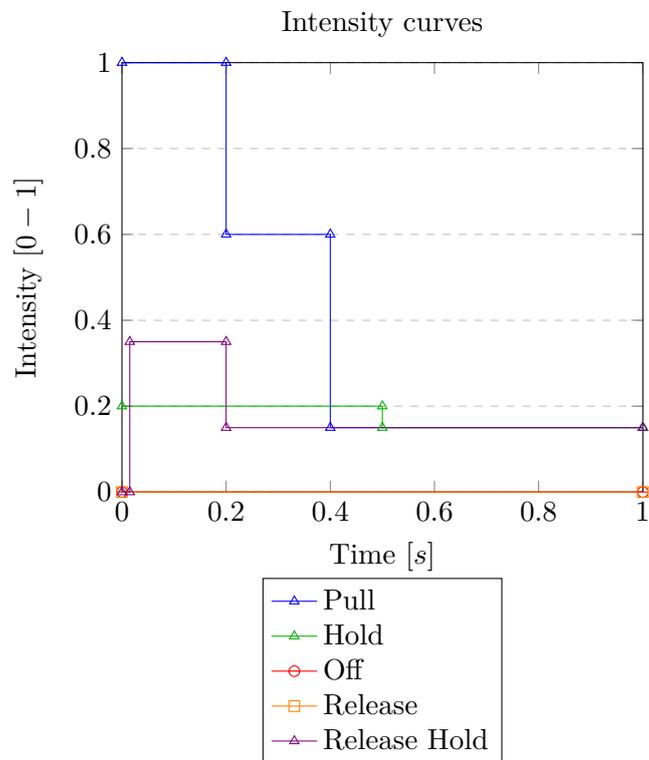
For constructing the mounting plate of the final prototype the CAD software *Fusion 360* was the tool of choice. The electro-magnetic simulations were conducted using *EMS* by *EMWorks* for *Autodesk Inventor Professional 2016*, Student Version. The schematic and the PCB of the magnet driver board have been designed with *EAGLE*, version 9.1.1, education licence. The concept renderings were created using the open source

```

1  PULL  = Curve([[1,0.1],[0.6,0.1],[0.15,0]])
2  HOLD  = Curve([[0.2,0.2],[0.15,0]])
3  OFF   = Curve([[0,0]])
4  RELEASE = Curve([[0,0]])
5  RHOLD = Curve([[0,0.015],[0.35,0.1],[0.15,0]])

```

**Program 6.1:** Definition of the intensity curves in Python code as part of the file `marvlet.py`. Each curve is defined by a sequence of tuples. The first value of shown tuples is the intensity and the second is the duration that intensity is held in seconds. A plot of the intensity curves can be seen in Fig. 6.2.



**Figure 6.2:** A plot of the intensity curves for the electromagnets defined in Program 6.1.

3D computer graphics toolset *Blender*, version 2.79b. As version control system a git repository hosted on GitHub was used. The software itself was written using the IDE's *Visual Studio 2017* (C++, JavaScript), *Visual Studio Code* (Python 3) and the *Arduino* IDE.

## Chapter 7

# Hardware

This chapter describes all the significant hardware elements that were used in building the prototypes and conducting the experiments.

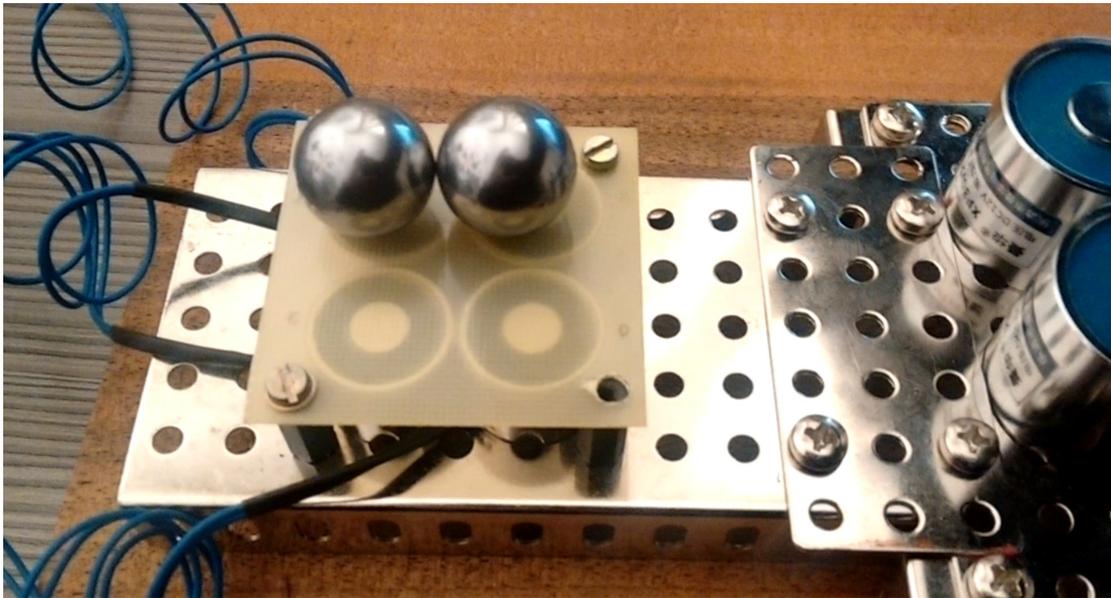
### 7.1 Prototypes

Apart from various small test setups three comprehensive prototypes have been built.

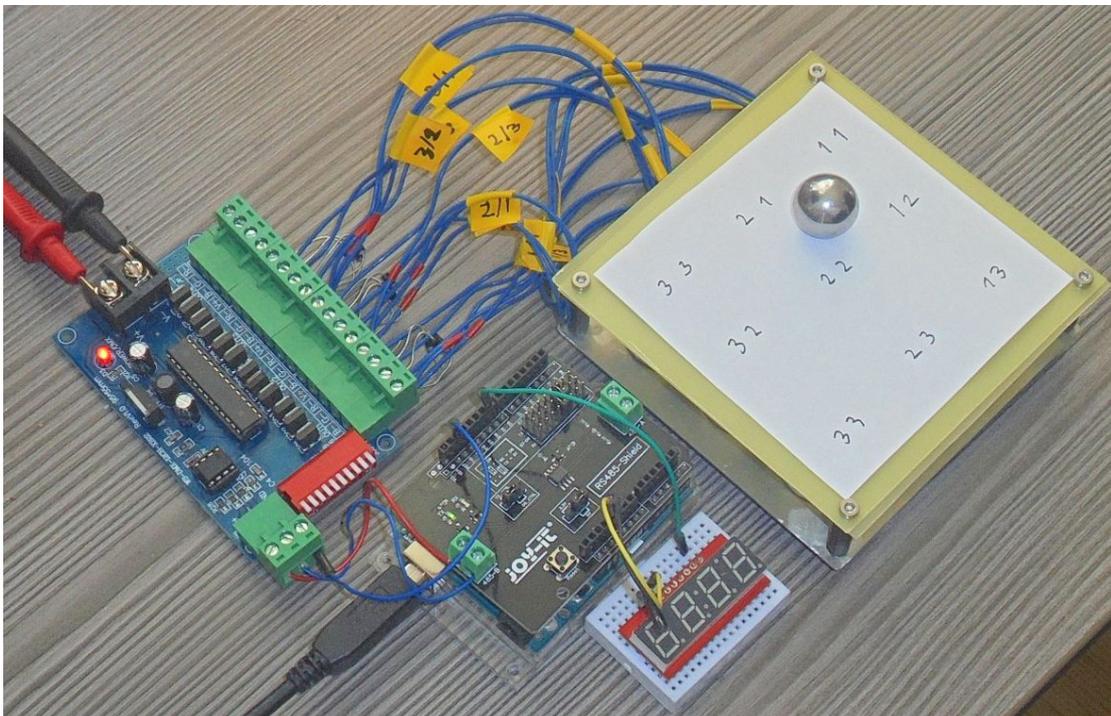
The first consisted of a  $2 \times 2$  grid of electromagnets of type P20/15 (20 mm diameter, 15mm height) which were controlled by the *Dual H-Bridge Motor Drivers*, an *Arduino UNO* and a serial connection to a laptop. The electromagnets were powered by 24V through a laboratory power supply. The first prototype confirmed that the basic movement principle was functional in a  $2 \times 2$  grid. It was tested with solid and hollow spheres. At times the spheres stuck together, preventing a successful next progression. This led to the conclusion that the distance between the spheres should be increased, by using bigger magnets. An image of one version of the first prototype can be seen in Fig. 7.1.

The second prototype consisted of a  $3 \times 3$  grid of electromagnets of type P30/25 and the 12-channel DMX driver (see Fig. 7.2). The second prototype was also powered by a laboratory power supply, successfully testing voltages up to 30V with it. Still occasionally a magnet was not strong enough to set a sphere into motion, even when it was pulling on a hollow steel sphere. This led to the conclusion that the final prototype should use an even higher voltage to power the magnets.

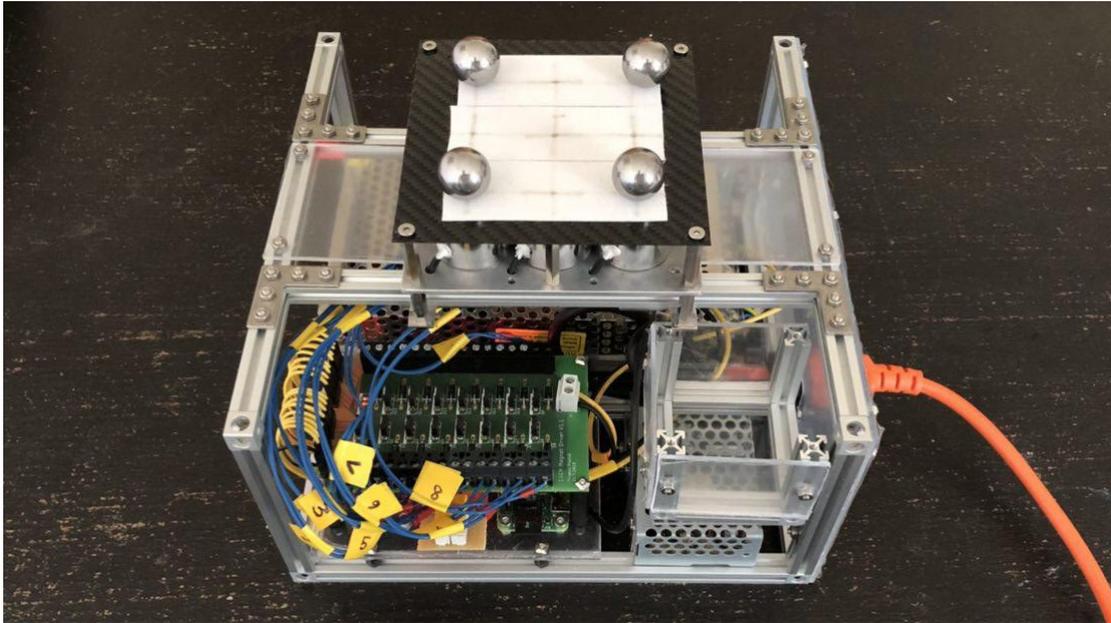
The third and final prototype (see Fig. 7.3) should be an autonomous machine, not dependent on external power supply. It was composed of the single board computer Raspberry Pi, the digital-analog converter *ADS1115*, a PWM controller board based on the *PCA6985* and a custom magnet driver board which could switch the provided 48V using PWM signals. The whole setup was powered by two fixed power supplies, providing 48V for the magnets and 5V for the logic circuits. A diagram of the whole setup is shown in Fig. 7.4.



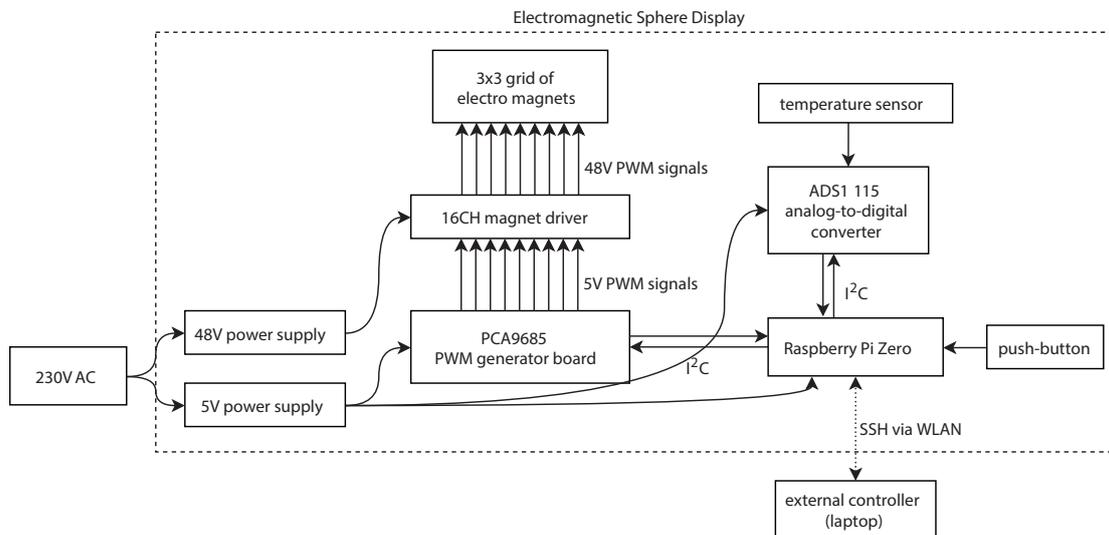
**Figure 7.1:** The first prototype used two Dual H-Bridge Motor controllers to power the electromagnets.



**Figure 7.2:** The second prototype utilizing the 12-channel LED driver and 9 electromagnets of type P30/25. The 7-segment display informs about the activated electromagnets.



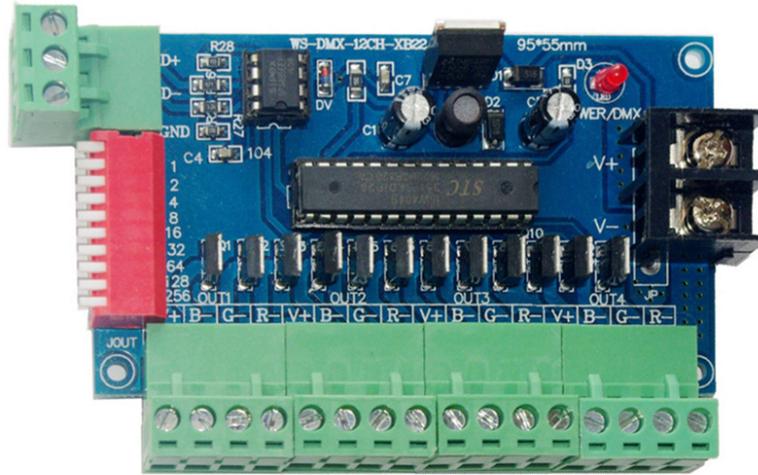
**Figure 7.3:** The final prototype powering a  $3 \times 3$  grid with a 48 V power supply. The control logic runs on a Raspberry Pi Zero single-board computer.



**Figure 7.4:** System diagram of the final prototype.

## 7.2 Multi-channel variable voltage driver

For the first prototypes to elude having to design a circuit to control variable voltage levels for each electromagnet individually an existing controller board was utilized. On the market a plethora of circuits switching high-voltage and high-current loads on multiple channels are available. They are designed for a multitude of different applications



**Figure 7.5:** The 12CH DMX512 LED Decoder used for controlling the electromagnets in the second prototype. Image source see footnote.

and have different capabilities. For initial experiments two *L298N Dual H-Bridge Motor Controllers*<sup>1</sup> have been used. This circuit board was designed for controlling the speed of two DC motors and enables, due to its H-bridge, a voltage to be applied across the load in both directions. With a logic level of 5 V it can control a voltage up to 35 V with a current flow up to 2 A. This was ideal for initial tests, as it was not determined that switching the magnet polarity would not help to improve the operation principle of the sphere display. For the  $3 \times 3$  prototypes 9 magnets had to be controlled, so the H-Bridge Controller was deemed to be space inefficient and impracticable. A single board that could source variable voltages for more than 9 loads with  $\geq 24$  V including providing current of  $\geq 1$  A per channel was needed. Also a serial interface to control all the electromagnets should be in place, to reduce the needed wiring and output pins.

Some popular commercial multi-channel drivers allow the control of multiple loads via relays, not enabling variable voltage levels<sup>2</sup>. Apart from variable voltage control the amount of available channels is also crucial. Many modules provide a channel count with a base of two (1, 2, 4, 8, 16, 32, ...) yet a number of channels with a power of two (1, 4, 9, 16, 25, 36, ...) is needed, when quadratic displays are built. The best options holding the aforementioned parameters including a reasonable price seemed to be two multi-channel DMX LED controllers. They are advertised as *12CH DMX512 LED Decoder*<sup>3</sup> (see Fig. 7.5) and *32CH DMX512 LED Decoder*<sup>4</sup>. The 12 channel variant has been used in the second prototype and will be described in more detail in the following section.

<sup>1</sup><http://www.robotpark.com/L298N-Dual-H-Bridge-Motor-Driver>

<sup>2</sup><https://www.sainsmart.com/products/16-channel-12v-relay-module>

<sup>3</sup><https://de.aliexpress.com/item/Free-shipping-DC5V-24V-12CH-DMX512-LED-Decoder-12channels-DMX-3P-Constant-voltage-controller-dimmer-For/32804400797.html>

<sup>4</sup><https://de.aliexpress.com/item/Digital-tube-Display-36CH-DMX512-Decoder-controller-DC5-36V-MAX-3A-XRL-3pin-controller-RGB-controller/32795465597.html>

### 7.2.1 12CH DMX512 LED Decoder

The driver used to control the electromagnets, according to its modest documentation, is dimensioned for controlling voltages up to 24 V and 1 A per channel. The limiting factors are the MOSFETs used to drive the individual loads. The voltage regulator which steps down the supply voltage to 5 V for the logic circuit is dimensioned for supply voltages up to 45 V<sup>5</sup>. The MOSFETs of type *HU60N03* are specified with  $V_{DSS} = 30$  V. This suggests that the actual maximum supply voltage is 30 V. The logic part of the circuit mainly consists of a differential bus transceiver to handle the DMX signal and a microcontroller to interpret the signal and control the MOSFETs.

The driver board has no flyback diodes to protect the circuit from the spikes resulting from switching inductive loads as the electromagnets. Considering the pulse-width modulation many spikes are expected in a short time due the high frequency switching. The flyback diodes in the second prototype were connected via the screw terminals. This seemed not an acceptable solution for the final prototype as it increased the danger of shorting the circuit. Also the ultimately used voltage of 48 V can not be supplied by this controller board. Additionally the 12-channel DMX driver sets all outputs high for a few seconds on start-up, and as a consequence the total current draw would exceed the maximum rating of the power supply used. Furthermore the driver is controlled via the DMX standard, which also required an additional circuit for communication. Finally the flyback diodes connected to the screw terminal provided a very fragile connection.

### 7.2.2 16-Channel Magnet Driver

Due the drawbacks and limitations of the 12-channel DMX driver (see Section 7.2.1) it was necessary to go for a different solution for the final prototype. As a result a 16-channel magnet driver board was designed. It features 16 n-channel MOSFETs of the type *NTD3055L104* [19]. Those transistors are designed for high speed switching applications in power supplies, converters and power motor controls and bridge circuits [19]. They are logic level MOSFETs, meaning that their gate-to-source threshold voltage is  $\leq 5$  V. In this case their threshold voltage is quoted as  $V_{GS(th)} = 1.6 V_{DC}$  [19]. They are specified for a maximum drain-source voltage of 60 V, therefore perfectly suitable for the needed 48 V to drive the magnets. Additionally they can take a continuous drain current of 12 A, which is also more than enough. A single channel is designed as open drain, so the transistor output (also drain) works as a current sink. An open drain terminal is connected to ground when a high voltage is applied to the gate, yet presents a high impedance when a low voltage is applied to the gate [20]. In this case the gate is connected to the PWM output of the *PCA9685* PWM driver. The full circuit can be seen in Fig. 7.8.

### 7.2.3 PCA9685 PWM driver

To drive all the channels of the driver board with PWM signals, the *PCA9685* PWM controller board is used. The *PCA9685* is I<sup>2</sup>C bus controlled 16-channel LED controller [21]. It can be easily be expanded by daisy chaining multiple devices. When

---

<sup>5</sup><https://www.onsemi.com/pub/Collateral/LM2575-D.PDF>

building a  $5 \times 5$  display only two controller boards are needed (providing a total of 32 channels).

### 7.3 Microcontroller

The single-board microcontroller *Arduino UNO* was used to interface with the DMX controller. This intermediate controller is used to stay independent from the top-level software that runs the application logic and determines which magnets to power. When the more complex top-level software crashes, the microcontroller software still runs smoothly and can turn off active magnets to prevent overheating. Additionally the *Arduino UNO* has an integrated analog-to-digital converter, which is needed for retrieving the positional sensor data and the temperature sensor.

The *Arduino UNO* alone is not capable of retrieving and sending data according to the DMX protocol. The hardware terms of DMX are based on the *RS485* standard which uses a balanced signal pair connection. To handle the signalling a *RS485 Shield* for *Arduino UNO* is used<sup>6</sup>.

### 7.4 Computer

For running the path finding algorithm and the application logic a microcontroller is not sufficient. For this purpose the single board computer *Raspberry Pi* is used. It is running the Linux Distribution *Raspbian Stretch Lite* which is based on *Debian*, a freely distributed operating system. The one board computer also has an  $I^2C$  interface, which is needed to communicate with the later mentioned PWM board and the temperature sensor.

### 7.5 Analog-to-digital converter

As an ADC to read the temperature sensor the a board based on the *ADS1115* is suggested<sup>7</sup>. It provides 4 inputs and samples with a bit depth of 16-bit. As it is interfaced with  $I^2C$  it can be easily be appended to the existing network used to communicate with the PWM controller.

### 7.6 Electromagnets

There is a wide range of different electromagnets available<sup>8</sup>. In Fig. 7.6 the types of magnets that have been used in experiments are listed. All the mentioned magnets are specified for 12 V. To reach the needed magnetic field strength to attract a neighbouring sphere which was determined by experiment, the maximum rating is exceeded by a factor of 4, reaching 48 V. This transgression of limits yields a increased heat generation,

<sup>6</sup><http://anleitung.joy-it.net/?goods=rs485-shield>

<sup>7</sup><https://www.adafruit.com/product/815>

<sup>8</sup>[https://de.aliexpress.com/store/product/FREE-SHIPPING-12V-DC-11LB-5kg-P25-20-Electric-Lifting-Lift-Magnet-Electromagnet-Solenoid/1724883\\_32550011225.html](https://de.aliexpress.com/store/product/FREE-SHIPPING-12V-DC-11LB-5kg-P25-20-Electric-Lifting-Lift-Magnet-Electromagnet-Solenoid/1724883_32550011225.html)

Type	Diameter	Height	Force (at 12 V)	Resistance
P20/15	20 mm	12 mm	2.5 kg	50 $\Omega$
P20/25	20 mm	25 mm	3.0 kg	49 $\Omega$
P25/40	25 mm	40 mm	8.0 kg	44 $\Omega$
P30/25	30 mm	25 mm	12.0 kg	30 $\Omega$

**Figure 7.6:** The different electro magnet types and their properties used for the prototypes.



**Figure 7.7:** The power supply *S-240-48* with a constant voltage of 48 V and a maximum current rating of 5 A. Image source [12]

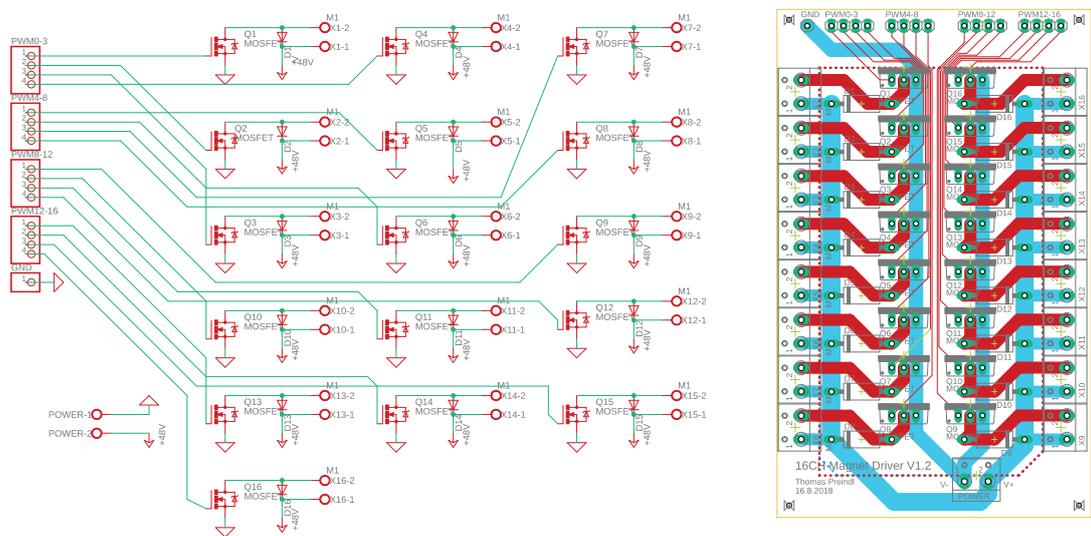
which is considered in software ensuring short full-on times. Also a temperature sensor is installed, enabling emergency shut down if the heat is too high.

## 7.7 Power supply

The initial prototypes have been powered by a variable power supply for laboratory use. The final prototype uses the switched-mode power supply *S-240-48* by *IeGeek* with a constant voltage of 48 V (see Fig. 7.7).

## 7.8 Temperature sensor

To prevent the machine from overheating a temperature sensor was built in to inform the software if the machine should pause. The used sensor is the thermistor *TS-NTC*. It is a resistor whose resistance is dependent on temperature. The sensor is mounted onto the mounting plate on which the electromagnets are attached. The sensor is glued to a ring cable lug which is mounted underneath the central electromagnet. Its resistance is measured utilizing the thermistor as voltage divider whereby the voltage is measured by the Analog-to-Digital converter *ADS1115*. It was chosen because it is I<sup>2</sup>C compatible and can easily be appended to the existing communication channel.



**Figure 7.8:** The schematic and PCB layout of the magnet driver board. It is designed so that it can be connected to the PCA9685 PWM driver.

## 7.9 Position detection

Different approaches for detecting the positions of the spheres have been tested. Measuring the current of the electromagnets, measuring the electromagnetic field and determining the exerted force onto the electromagnets. More details on the tested methods can be found in chapter 4.

### 7.9.1 Hall sensors

The tests were conducted with the hall sensor *A1324 LUA-T*<sup>9</sup> from Allegro MicroSystems. For the final sensor assembly a SMD hall sensor like the *SS39ET* by Honeywell would be used due to its small height of 1.4 mm [22].

### 7.9.2 Force sensitive resistors

The *Kitronyx Snowboard* (a circuit board with integrated force and capacitive touch sensing controllers) was used for evaluating the utility of a force sensing resistor (FSR) matrix for sensing the sphere positions.

## 7.10 Steel spheres

Many different kinds of steel spheres have been acquired. *Kugelpompe*<sup>10</sup> has been a valuable retailer of solid steel spheres. Yet experiments have exposed solid spheres to be

<sup>9</sup><https://www.arrow.de/products/a1324lua-t/allegro-microsystems>

<sup>10</sup><https://www.kugelpompe.at/>

suboptimal for this application. The sourcing of hollow steel spheres has turned out to be exceptionally difficult. Only in China distributors were found, of which almost all sell only big quantities which are manufactured on demand. The successful approach was to request samples, despite the high combined cost of the product and including shipping. The manufacturer which provided the project with ferromagnetic hollow steel spheres with sizes 16 mm and 21 mm was *Zhejiang Shangyu XinXin Steel Ball Co. Ltd.*<sup>11</sup>.

### 7.11 Display surface

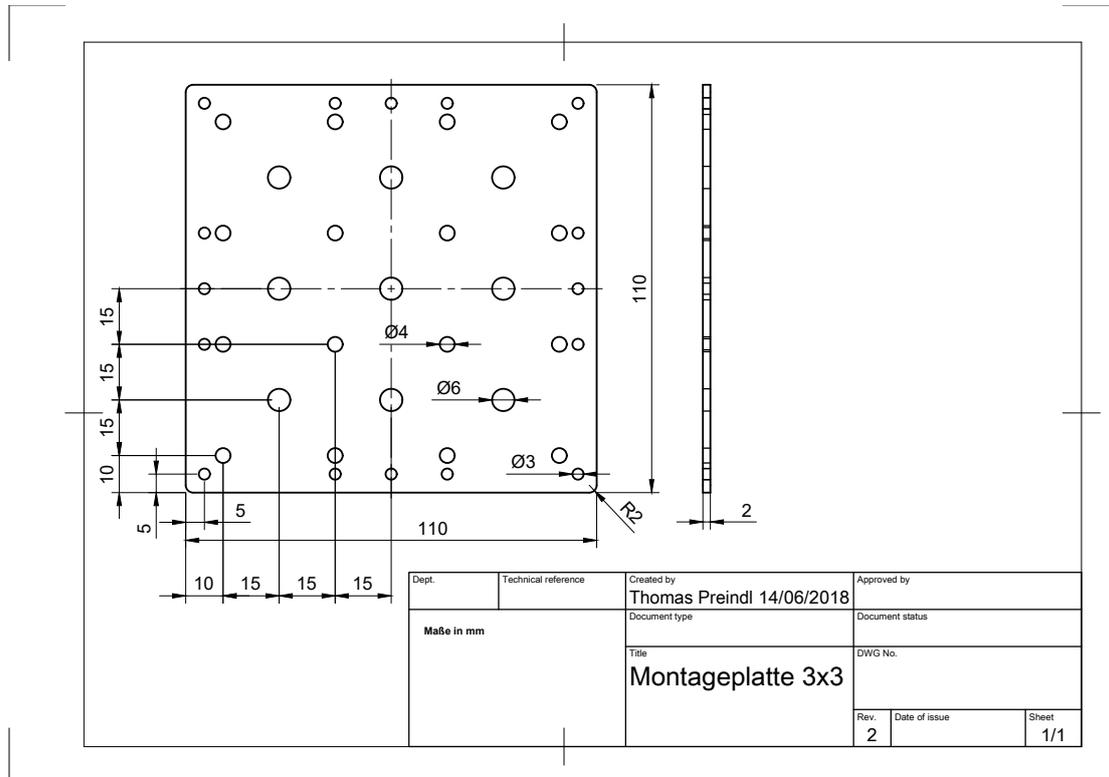
The top plate mounted on top of the electromagnets is needed to give the spheres a flat surface to roll on, while also offering a housing for a sensing method to detect the positions of the spheres. The surface character of the top plate is crucial to the behaviour of the machine. A very rigid top plate reduces friction and therefore the amount of force that is needed to set a sphere into motion. On the downside a moving sphere needs a long time to settle having reached its target position. If the friction is very low, the spheres oscillate back and forth like a bouncing ball until they finally reach a halting position where the electromagnet can be switched off. If the magnet is switched off too early, the sphere, due its inertia, is shot away from the board, eventually hurting living organisms. Another downside of a rigid top plate is that once the magnet is switched off, the spheres tend to roll of the board due to a eventually slightly tilted table the machine stands on. To mitigate those problems, while taking a loss in the needed force to set a sphere into motion, a soft display surface is used. For the final prototype a thin layer of woven linen fabric is glued on top of a rigid carbon fibre plate. In experiments this setup has proven to serve its purpose perfectly.

### 7.12 Hazards

The possibility of a crashing operating system still poses a hazard. The PWM controller board keeps the last PWM state as long as power is supplied, without regard to the state of the I<sup>2</sup>C master device. If the last state of a channel is full-on and the operating system of the I<sup>2</sup>C master device is crashing or rebooting without executing a shutdown procedure to turn off all channels, the *PCA9685* PWM controller will still hold the channel full-on until it is told else. This can lead to overheating and the subsequent destruction of an electromagnet or in the worst case the whole machine. Also it poses a fire hazard to the environment due the generation of heat. This problem can be prevented by using a microcontroller between the computer which runs the application logic and the PWM controller. The microcontroller could monitor the duration a magnet is turned on and also monitor the heat via a temperature sensor. If a parameter exceeds a certain limit, it can counteract by turning off the magnets and pause the progression until the magnets have cooled down.

---

<sup>11</sup><http://xxballs.com/>



**Figure 7.9:** Construction drawing of the mounting plate for the electromagnets.

### 7.13 Mechanical Design

For the final prototype a sturdy non-magnetic aluminium plate for mounting the electromagnets was designed and manufactured (see Fig. 7.9). It has nine 6 mm holes through which the magnets are held by M6 hexagon socket cylinder bolts. The other holes serve either as cable feed-through or are needed for distance bolts on top of which the top plate is mounted.

A robust frame was constructed using 10 mm aluminum profiles. They are marketed by the name *MakerBeam* and sold as kits including stainless steel bracket to connect the profiles.

### 7.14 Mechanical Problems

Certain progressions lead to spheres attracting each other and consequently sticking together, not arriving at the determined position. When this happens it is hard for the machine to separate them again. Without an implemented sensing method there is no way for the machine to know when such an unwanted attraction happened. To prevent those incidents the intensity curves (as mentioned in Chapter 6) have to be fine-tuned. For that the process of unwanted attraction can be recorded as a slow-motion video and subsequently analysed. It is advised to also record a screen on which the current

state of the magnets is displayed, or else the current strength of the electromagnets can only be estimated by watching the moving spheres. Certain problematic arrangements have been repeatedly tested using different intensity curves. The test have shown that the unwanted attraction only seldom happens and can not always be repeated. This is presumably influenced by factors as the varying magnet strength resulting from the generated heat which changes the current flow, magnetized spheres, eventually a non-planar standing ground of the machine or external factors as wind.

## Chapter 8

# Applications

Despite the narrow constraints in speed and resolution the magnetic sphere display allows a multitude of applications. In this chapter a selection of hypothetical utilizations is outlined. Depending on the resolution and the implemented position sensing method different applications are possible. If a feed and retrieval mechanism is implemented, further applications arise. In most applications a sound system as an additional channel of communication is useful or necessary.

### 8.1 Proposed interaction modes

When operating the machine the user can only interact by moving spheres or touching the displays surface. If the position sensing is implemented via force sensors the user could, depending on the application, interact in following proposed ways:

- Place a sphere on the board.
- Remove a sphere from the board.
- Short tap on an empty position.
- Double tap on an empty position.
- Long press on an empty position.
- Swipe over multiple empty positions.
- Place a token on the display which is not a sphere.

The final prototype, developed as part of this thesis, which does not incorporate a position sensing method, relies on remotely controlling the incorporated computer and via a button dedicated to starting, stopping and resetting the movement procedure.

### 8.2 Symbolic display

Not too long ago display resolution in common applications, like mobile phones, have been so low that one could identify a single pixel with the naked eye. For such devices low resolution pictograms and fonts (see Fig. 8.1) have been designed. The steel sphere display with a proposed resolution of  $5 \times 5$  is capable of displaying one letter at a single point in time. With the current design the display is not able to feed or retrieve spheres, so letters with different numbers of black pixels can not be displayed in succession

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

**Figure 8.1:** A pixelfont using a maximum resolution of  $5 \times 5$  per symbol.

0 1 2 3 4 5 6 7 8 9

**Figure 8.2:** A custom numerical pixelfont created for this application with a resolution of  $5 \times 5$  using a fixed number of 8 black pixels.

without human intervention. For that reason a pixelfont has been designed for the machine, in which all digits have the same amount of black pixels (see Fig. 8.2). With it any number from 0 to 9 can be displayed in human readable form, only by moving 8 available spheres. Despite the low resolution a  $5 \times 5$  sphere display has  $2^{25} \approx 10^7$  possible states using an arbitrary number of spheres. Considering a fixed number of 8 spheres, the number of possible states are  $\binom{25}{8} \approx 10^6$ . If arrangements which are rotations of other arrangements are excluded, the number of unique arrangements is approximately  $10^5$ . It is apparent that even the low resolution of  $5 \times 5$  holds many possible states for displaying information.

### 8.3 Timer

The proposed machine can be used as a timer. The user can place spheres on the board, each sphere representing a minute. Every time when the user places a sphere, the timer restarts using the number of currently placed spheres as the duration in minutes. The spheres then rearrange in a manner that all of them move from one side of the board to another. The number of needed steps is then divided by the timer duration, determining how long the machine waits between each step. When the machine stops moving spheres, the timer has finished. A sound is played signalling the finished timer.

## 8.4 Game concepts

The specific interaction possibilities of the electromagnetic sphere displays open the door for many games, specifically designed for this device. Subsequently a two rough concepts are presented.

### 8.4.1 Simon

Similar to the popular memory game *Simon*, where one has to memorize a sequence of colours, a game could be implemented in which the player has to remember a sequence of arrangements. Initially the user places any number of spheres on the board. The player starts the game with a double tap at any position. The spheres then rearrange into a randomly generated arrangement. After a short pause the next random arrangement is displayed. Every finished arrangement is signalled with a sound. Afterwards the player rearranges the spheres to the first arrangement, then to the second, while confirming

each arrangement with a double tap. The validity of an entered arrangement is notified via a sound after the double tap. If the player makes a mistake, she is corrected by the machine. After each successful rearrangement of the whole sequence by the player, another arrangement is added by the computer. The game does not require a feed and retrieval mechanism.

#### 8.4.2 Last Man Standing

The game starts with an arbitrary number of spheres placed on the display. The spheres start to move in a random pattern. After several moves the movements pause and the player has to remove the sphere which has executed the most movements. The game is lost when the wrong sphere is chosen. A sound communicates if the player was wrong or right. This procedure is repeated until no sphere is left. The player wins if every decision was right and no sphere is left on the board.

## Chapter 9

# Closing Remarks

It was not apparent that the initially proposed idea of moving ferromagnetic spheres by controlling a rigid grid of electromagnets was realizable, yet the basic concept of the machine was proven feasible through simulations, experiments and finally the construction of a functional prototype.

### 9.1 Conclusion

The path to building the final working prototype was far from trivial. There have been multiple obstacles in making it work. Firstly the amount of current needed to reliably attract the spheres from a horizontally displaced electromagnet was underestimated and adjusted after conducting multiple experiments. Using hollow steel spheres was also essential in enabling the proposed method. The exact timing for the intensity curves of the electromagnets was figured out by trial and error. Finding the right components and materials for constructing the machine was also a big time factor. The algorithmic part of the machine was very extensive and proved to be one of the most complex elements of the system. Concluding it can be said that this project spans over a very wide area of topics making it a compelling undertaking in terms of design and engineering — all in all a challenging feat of problem solving. The prototype is the first step towards a machine which defies the common image of a display by trying to slow down communication. It already can fulfil this purpose, yet further development is desirable to fully reach the goal aimed at.

### 9.2 Future development

An apparent improvement would be to increase the resolution of the final prototype. As previously discussed a electromagnetic sphere display with a resolution of  $5 \times 5$  can already display numerical data in human-readable form. A higher resolution also strongly increases the number of possible states. On the downside the needed amount of electromagnets also grows quadratically with the increased resolution, thus increasing the cost, power consumption and heat generation. To make it an interactive device one of the proposed and tested sphere sensing methods has to be implemented. The preferred method as argued in Chapter 4 is based on using force sensitive resistors, favoured for its

space-saving and cost effective properties. Having a way to detect the sphere positions would increase the number potential applications and also help to reset the display after a failed progression. Another area for potential future work is to optimize the intensity curves to minimize the needed time for the spheres to reach their target position and to reduce the amount of failed movements. This could be achieved by analysing slow-motion recordings of the executed progressions and look for critical progressions. To give the machine also a finished appearance a casing should be built, which lets the user focus more on the information the spheres represent. As a result of the enclosed heat generating elements a proper cooling method has to be implemented. To expand the means of interaction an automated feed and retrieval method can be implemented which enables the machine to add or remove spheres from the board. Having multiple sphere types on the display which could be differently coloured or of different size could further increase the number of applications. As a result the path finding algorithm and the sphere sensing method would have to be adapted. A possible feed and retrieval method would then also increase in complexity. The path finding algorithm can be optimized for board resolution bigger than  $5 \times 5$ . This can be done by implementing an iterative path finding algorithm which quickly can offer a solution which then is continuously improved. As a final note the in Chapter 8 suggested applications can be implemented and evaluated by their utility.

# Appendix A

## DVD Contents

Path: /

thesis.pdf . . . . . This document.

Path: /construction

base\_110x110x2mm.pdf Mounting plate as PDF-File.

base\_110x110x2mm.dxf Manufacturing ready DXF-File of the mounting plate.

Path: /pcb

mosfet\_driver\_16ch\_v12.sch The Eagle schematic of the driver board.

mosfet\_driver\_16ch\_v12.brd The Eagle PCB layout of the driver board.

Path: /references

/manuals . . . . . Contains the referenced manuals.

This folder contains all referenced literature that could be obtained digitally.

Path: /simulations

/results . . . . . Contains results as images and in plain text.

All data associated with the conduction of the simulations with EMS.

Path: /software

/marvle . . . . . Contains a Microsoft Visual Studio project.

/marvle/marvle . . . . . Contains an implementation of the path finding algorithm.

/marvle/marvle\_test . . . . . Contains unit tests for the path finding implementation.

/marvle/marvle\_web . . . . . Contains the web application.

/marvlet . . . . . Contains the Python code for the program logic and hardware interfacing.

## Path: /videos

- 2x2\_1\_24V.mp4 . . . . Testing a  $2 \times 2$  grid powered by 24 V with 1 sphere.
- 2x2\_2\_24V.mp4 . . . . Testing a  $2 \times 2$  grid powered by 24 V with 2 spheres.
- 3x3\_2\_24V.mp4 . . . . Testing a  $3 \times 3$  grid powered by 24 V with 2 spheres.
- 3x3\_3\_24V.mp4 . . . . Testing a  $3 \times 3$  grid powered by 24 V with 3 spheres.
- 3x3\_3\_48V.mp4 . . . . Testing a  $3 \times 3$  grid powered by 48 V with 3 spheres.
- 3x3\_4\_48V.mp4 . . . . Testing a  $3 \times 3$  grid powered by 48 V with 4 spheres.
- FSR\_test.mp4 . . . . . Testing a force sensitive resistor.
- surface\_test.mp4 . . . . Testing the effects of different display surfaces.
- hallsensor\_test.mp4 . . Testing the effect of a steel sphere on the magnetic field.
- slowmotion.mp4 . . . . A slowmotion recording of moving spheres.
- magnetcurrent\_24V\_1ohm5.mp4 Measuring the current flow when moving a sphere on the magnet.
- progression\_animation.mp4 A rendered animation of moving spheres.

## Path: /visualisation

- 2017\_marble\_animation\_packed.blend 3D scene of the progression animation.
- 2017\_marble\_v4\_packed.blend 3D scene of the machine rendering.

# References

## Literature

- [1] Javier Alonso-Mora et al. “Image and Animation Display with Multiple Mobile Robots”. *International Journal of Robotics Research* 31.6 (2012), pp. 753–773 (cit. on p. 6).
- [2] Hiroshi Ishii et al. “Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials”. *Interactions* XIX (Feb. 2012), pp. 38–51 (cit. on p. 3).
- [3] H. W. Kuhn. “The Hungarian Method for the Assignment Problem,” *Naval Research Logistics Quarterly* 2 (1955), pp. 83–97 (cit. on pp. 11, 15).
- [4] Daniel Leithinger et al. “inFORM: Dynamic Physical Affordances and Constraints through Shape and Object Actuation”. In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology – UIST*. (St Andrews, United Kingdom). 2013, pp. 417–426 (cit. on p. 3).
- [5] George F. Luger. *Künstliche Intelligenz*. Munich: Pearson Studium, 2001, pp. 147–148 (cit. on p. 12).
- [6] Hang Ma and Sven Koenig. “Optimal Target Assignment and Path Finding for Teams of Agents Categories and Subject Descriptors”. In: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems – AAMAS*. (Singapore). 2016, pp. 1144–1152 (cit. on p. 15).
- [7] Patrick MacAlpine, Eric Price, and Peter Stone. “SCRAM: Scalable Collision-Avoiding Role Assignment with Minimal-Makespan for Formational Positioning”. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. (Austin, Texas, USA). 2015, pp. 2098–2102 (cit. on pp. 9–11, 15).
- [8] Marshall McLuhan. *Understanding Media: The Extensions of Man*. New York: McGraw-Hill, 1964 (cit. on p. 1).
- [9] Daniel Ratner and Manfred Warmuth. “Finding a shortest solution for the  $N \times N$  extension of the 15-puzzle is intractable”. In: *Proceedings of the 4th National Conference on Artificial Intelligence – AAAI*. (Austin, Texas, USA). 1986, pp. 168–172 (cit. on p. 7).
- [10] Guni Sharon et al. “Meta-Agent Conflict-Based Search For Optimal Multi-Agent Path Finding”. In: *Proceedings of the 5th Annual Symposium on Combinatorial Search*. (Niagara Falls, Ontario, Canada). 2012, pp. 97–104 (cit. on pp. 7, 11).

- [11] Guni Sharon et al. “The Increasing Cost Tree Search for Optimal Multi-Agent Pathfinding”. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. (Barcelona, Spain). July 2011, pp. 662–667 (cit. on p. 7).

## Online sources

- [12] *48V Power Supply*. URL: <https://www.ebay.de/itm/Schaltnetzteil-5V-12V-24V-48V-Netzteil-Trafo-DIY-LED-Strip-Power-Supply-Frei-DHL/401485636262> (visited on 09/01/2018) (cit. on p. 37).
- [13] Wikimedia Commons. *The Turk*. 1783. URL: [https://en.wikipedia.org/wiki/The\\_Turk#/media/File:Tuerkischer\\_schachspieler\\_windisch4.jpg](https://en.wikipedia.org/wiki/The_Turk#/media/File:Tuerkischer_schachspieler_windisch4.jpg) (cit. on p. 5).
- [14] *Daniel Rozin Homepage*. 2018. URL: <http://www.smoothware.com/danny/contact.html> (visited on 01/24/2018) (cit. on pp. 3, 4).
- [15] *Electromagnetic Simulation by EMWorks*. URL: <https://www.emworks.com/product/ems> (visited on 09/01/2018) (cit. on p. 24).
- [16] Martin Frey. *Snoil Documentation*. 2018. URL: <http://www.freymartin.de/en/projects/snoil> (visited on 01/24/2018) (cit. on pp. 4, 5).
- [17] *Introduction to Capacitive Touch Sensing*. URL: <https://www.allaboutcircuits.com/technical-articles/introduction-to-capacitive-touch-sensing/> (cit. on p. 21).
- [18] Tangible Media Group / MIT Media Lab. *inFORM*. 2012. URL: <https://tangible.media.mit.edu/project/inform/> (visited on 09/01/2018) (cit. on p. 4).
- [19] *NTD3055L104 Power MOSFET*. Issue 4. ON Semiconductor. URL: <https://www.onsemi.com/pub/Collateral/NTD3055L104-D.PDF> (visited on 01/24/2018) (cit. on p. 35).
- [20] *Open collector*. URL: [https://en.wikipedia.org/wiki/Open\\_collector](https://en.wikipedia.org/wiki/Open_collector) (visited on 09/01/2018) (cit. on p. 35).
- [21] *PCA9685 16-channel, 12-bit PWM Fm+I2C-bus LED controller*. Rev. 4. NXP Semiconductors. Apr. 2015. URL: <https://www.nxp.com/docs/en/data-sheet/PCA9685.pdf> (visited on 01/24/2018) (cit. on p. 35).
- [22] *SS39ET/SS49E/SS59ET Series Linear Hall Effect Sensors*. Issue 4. Honeywell. Feb. 2015. URL: <https://sensing.honeywell.com/honeywell-sensing-ss39et-ss49e-ss59et-product-sheet-005850-3-en.pdf> (visited on 01/24/2018) (cit. on p. 38).