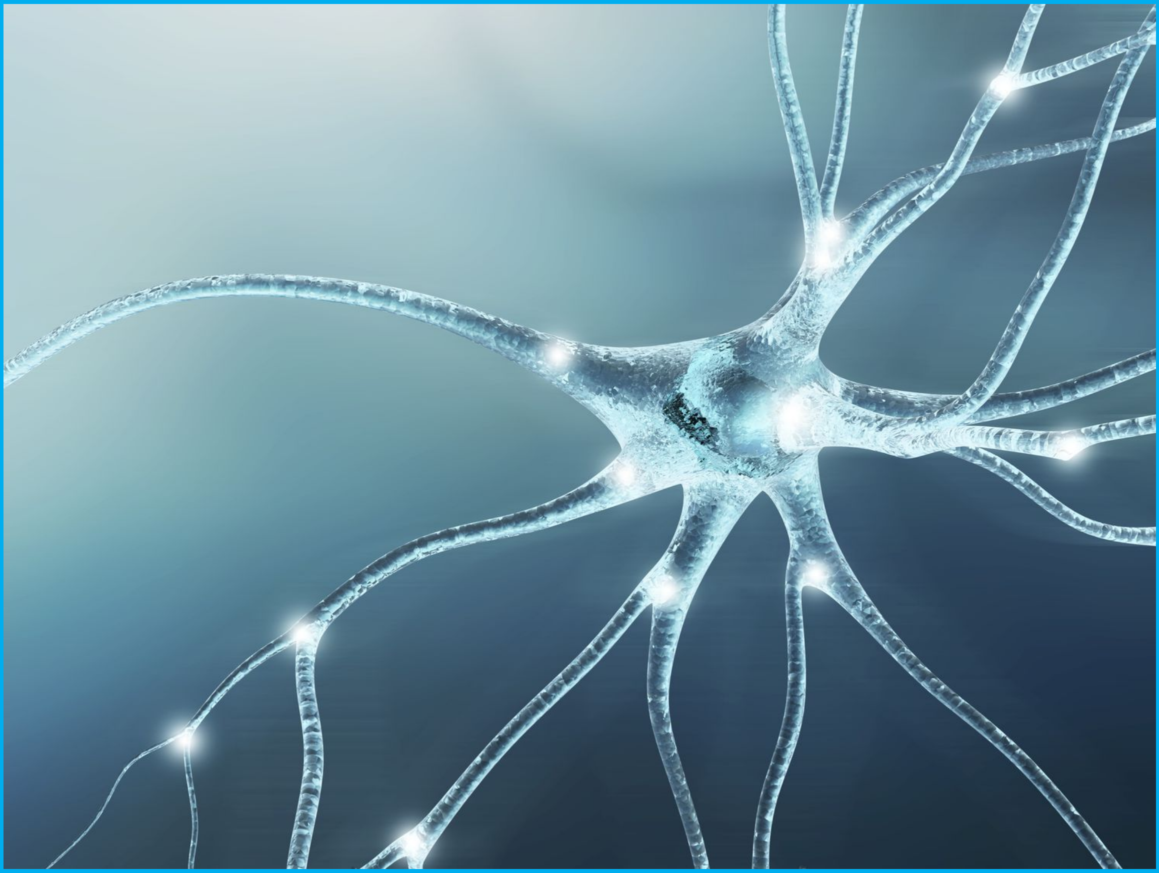
Master Thesis

Optimization algorithm for visual cortex stimulation

I. de los Mozos

Technische Universiteit Delft



Optimization algorithm for visual cortex stimulation

by

Inés de los Mozos Vargas

The work in this thesis was made in the Bioelectronics section
Department of of Microelectronics Faculty of Electrical Engineering,
Mathematics & Computer Science
in partial fulfillment of the requirements for the degree of

Master of Biomedical Engineering, Medical devices track.

at the

TECHNISCHE UNIVERSITEIT DELFT

August 2024

Author . . . .

Inés de los Mozos Vargas
Student number: 5857368
23rd August, 2024

Thesis committee . . . .

Thesis Supervisor: Tiago Costa
Daily supervisor: Francesc Varkevisser
Dr. Matin Jafarian
TU Delft

Abstract

This master thesis investigates the potential of vector field stimulation as an innovative technique for enhancing visual cortex stimulation [1]. An algorithm was designed and implemented to evaluate this concept, utilizing COMSOL simulations to model the generated electric fields. The algorithm was validated using Green functions, which serve as analytical approximations to the Poisson equation solved in COMSOL. The interior point method was employed in the algorithm to minimize visual error [2], a concept introduced in the literature and adjusted to our problem. Results indicate that vector field stimulation improves targeted stimulation precision noticeably compared to traditional bipolar stimulation methods. Additionally, it was observed that incorporating the minimization of input current alongside visual error further enhanced the algorithm's performance. Despite challenges such as numerical errors and the algorithm getting trapped in local minima, the findings highlight the need for future research to improve microstimulation techniques. This report emphasises the promising potential of vector field stimulation and the supporting algorithm in improving visual cortex stimulation.

Keywords: Visual prostheses, COMSOL, Vector field stimulation, Current Steering, Poisson equation, phosphenes, activating function, microelectrode arrays (MEAs), intracortical microstimulation, Finite element modelling (FEM), Visual error (VSE), Interior point method

Acknowledgments

The completion of this master's thesis marks the end of my time as a student in Delft. I am very grateful for the opportunity to study here and for the experiences of the past two years. This project would not have been possible without the help and support of my friends, family, and colleagues. First, I would like to thank my supervisors, Tiago and Francesc, for their guidance throughout this project. A special thanks to Cesc for always being available for questions, feedback, and discussions, and for his consistently positive attitude.

Thank you to Dr. Matin Jafarian for taking the time to read this thesis and participate in my thesis committee. I extend my gratitude to everyone in the bioelectronics group. The excellent environment on the 16th floor made working on this project an even greater experience. I am also deeply thankful to my friends, who have become like a second family over these past two years. Special thanks to Sofia, Juanfran, Alba, Marta, Nerea, Bea, Maria, Nikita, Kush, Joao, Arun, Monica, Rebecca and Oscar for making this experience memorable. A special thanks to Adri for his unconditional support and advice. Lastly, I would like to thank my family—mum, dad, and Palo—for always being there, supporting, and believing in me.

Inés de los Mozos Vargas
Delft, 2024

Contents

List of Figures **iv**

List of Tables **xi**

1 Introduction **1**

1.1 Background 1

1.2 Problem statement 1

1.3 Chapter organization 2

2 Background and theory **3**

2.1 Visual prostheses 3

2.2 Microstimulation of visual cortex 5

2.2.1 Target regions of cortical visual prostheses 5

2.2.2 Intracortical vs surface microstimulation 6

2.2.3 Current mode vs voltage mode 7

2.2.4 Phosphene generation 8

2.3 Microstimulation: Fundamental concepts and Safety 10

2.3.1 Microelectrode arrays (MEAs) 10

2.3.2 Electrode tissue interface 10

2.3.3 Safety thresholds 11

2.3.3.1 Tissue damage threshold: 13

2.3.3.2 Electrode damage threshold: 14

2.3.4 Activating Function 14

2.3.5 Current Steering 16

2.4 Problem statement 18

3 Prior work 19

3.1 Electric field shaping 19

3.1.1 Current steering 19

3.1.2 Vector field shaping 22

3.1.3 Proposed Implementation 23

3.2 State-of-the-art methods to optimize electrical stimulation processes . 24

3.2.1 Calculate individual electrical contributions of each electrode using FEM 24

3.2.1.1 Origin of Poisson equation 25

3.2.2 Optimize objective function 27

3.2.3 Analysis of the resulting electric field 32

3.2.4 Conclusion 32

3.3 Problem statement and research question 33

3.4 Project plan 34

4 Methods 35

4.1 Introduction 35

4.2 Problem formulation 35

4.3 Data collection 37

4.3.1 Model description 37

4.3.2 Model validation 43

4.3.3 Analytical input: Green´s Functions 44

4.4 Objective function 46

4.5 Optimization algorithm 50

4.5.0.1 Interior Point Methods 51

5 Results 53

5.1 Generated Electric Potential Fields 53

5.1.1 Visualization of the Potential Fields 54

5.2 Activating Function Response: 55

5.2.1 Visualization of the AF response 55

5.3 Algorithm Convergence study 56

5.4 Solution analysis 57

5.4.1 Study case 1: using data extracted from Green Function approximation 60

5.4.2 Study case 2: using data extracted from COMSOL at a cut plane placed 100 μm away from the sources 64

5.4.3 Study case 3: using data extracted from COMSOL at a cut plane placed 1 μm away from the sources 67

5.4.4 Study case 4: minimization of injected current sources 72

5.4.5 Study case 5: Multi-target VSE minimization 74

5.4.6 Study case 6: using exponential objective function 77

6 Discussion and conclusions 81

6.1 Discussion 81

6.2 Future developments 82

6.3 Conclusion 85

Bibliography 86

A Green Functions code for potential field analytical input 94

B Problem Functions 97

C Electric potential distributions for individual electrodes in the MEA, in a 2D cut plane at 100 μm above the tips of the electrodes from the 3D COMSOL model 102

D Electric potential distributions for individual electrodes in the MEA, in a 2D cut plane at 1μm above the tips of the electrodes from the 3D COMSOL model 104

E Electric potential distributions for individual electrodes in the MEA, extracted using Green functions 106

F Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from Green functions108

G Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 100µm 111

H Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 1µm 114

I Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 100µm and intensity minimization 117

J Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from Green Function and exponential objective function 120

List of Figures

2.1 Human Visual pathway [12]. 4

2.2 Possible stimulation sites for a visual prosthesis [23]. 4

2.3 Cortical visual prostheses system [8]. Where the following components can clearly be distinguished: an image-capturing device, the glasses (a,d), an intra-cortical microstimulation array (e), and an image-processing device, the computer (c). 5

2.4 a) Layers of V1 , b) Regions of Visual cortex (Figure adapted from [30, 31]). 6

2.5 Action potential graphical representation [37]. 9

2.6 Utah array [45] 11

2.7 Electrode double layer [34]. 12

2.8 Biphasic stimulating pulse, cathodic-first [46]. 12

2.9 Cable model of the axon of the neuron. **a)** Experimental setup used for deriving the cable equation. It assumes that the total stimulus current (I_i), which has an axial flow inside the axon, decreases with distance since part continually crosses the membrane to return as a current (I_o) outside the axon. It also considers the axon to have an infinite length. **b)** Circuit model of an axon. Figure adapted from [56, 57]. 16

2.10 Current steering using electrode pairs. The electric fields generated by the stimulation are represented by a continuous line for electrode 5 (E5), a dotted line for electrode 6 (E6), and a dashed line for simultaneous stimulation of both electrodes. **A)** Both electrodes are stimulated with the same amount of current. **B)** E5 is stimulated with 30 % of the total current, while E6 is stimulated with 70 % of the current [61]. 17

3.1 The figure demonstrates the Vector Field Shaping technique from [1]. On the left, the top images show the electrodes activated after differential stimulation, while the bottom images represent the resulting activation field. On the right, the images illustrate the activation pattern achieved with Vector Field Shaping, where more electrodes are activated, leading to a more precise activation field. 23

3.2 a) Set-up example where the electrode is in the coordinate system (X,Y,Z). This electrode is located on the surface (grey plane), and the centre of the coordinate system corresponds to the centre of the electrode. b) Schematic representation of the electrode (red) and horizontally oriented axon (green) on (X,Z) plane (Y=0). Activating function representation for **c)** horizontal and **d)** vertically oriented fibers, where red represents the areas of depolarization and blue areas of hyperpolarization [55]. 28

3.3 Visual interpretation of the described visual error in [7]. Where red represents the excited locations (j) and the green represents the target locations (i) 30

4.1 Problem formulation flow chart 37

4.2 3D Geometry of the model with the defined measures 39

4.3 Material properties of the model, the highlighted element in each figure represents the section of the model to which each specific material is assigned. 40

4.4 Boundary conditions, the highlighted element in each figure represents the section of the model to which each specific BC is assigned. 42

4.5 Mesh defined for the simulations 43

4.6 Cut plane at 1 μm over the microelectrodes tips. 43

4.7 Electrode location visualization. Where the region of study is 2000 um. 46

4.8 VSE cost functions against the current density introduced at certain pixels, pink and dark blue. The orange pixel is the original target, and the dark blue and pink are the undesirably activated pixels. 49

4.9 Visualization of an example problem at its 3rd iteration: a) Solved by Simplex method, b) Solved by interior point methods. Image source [90]. 51

5.1 Visualization of the potential fields for a unit current into electrode 1 of the MEA calculated using (a) the analytical approach with Green's function at a distance of 100 μm from the source, (b) FEM at a 2D cut plane 1 μm above the tip of the electrodes, and (c) FEM at a 2D cut plane 100 μm above the tip of the electrodes. 54

5.2 Visualization of (a) the electric potential field for electrode 1, calculated using Green´s functions at a cut-plane 100 μm away from the source. (b) Its corresponding AF response. (c) 1D plot of extracellular potential represented in (a). (d) 1D plot of the activating function. . 56

5.3 Visualization of the convergence behaviour of the algorithm over 50 runs for target point [5,7]. The main plot displays the overall convergence trend where most runs appear to converge to a similar point. However, as shown in the inset, a closer examination reveals that not all runs converge to the global minimum; some get trapped in local minima. The inset magnifies this region, illustrating the variations among runs that are not visible in the broader view. 57

5.4 Visualization of the convergence behaviour of the algorithm over 50 runs for target point [8,4]. The main plot displays the overall convergence trend where most runs appear to converge to a similar point. However, as shown in the inset, a closer examination reveals that not all runs converge to the global minimum; some get trapped in local minima. The inset magnifies this region, illustrating the variations among runs that are not visible in the broader view. 58

5.5 Visualization of symmetry operations applied to original solutions from triangle 59

5.6 Visualization of resulting activation pattern when attempting to activate point (8,8). In the figure, the relative position between the target and the electrodes can be distinguished. 61

5.7 Representation of how the (a) VSE and (b) the percentage of wrongly activated pixels distribute along the overall Green Functions dataset . 62

5.8 Distinction between the targets considered close to electrodes and the ones considered far. 63

5.9 Distinction between the areas inside and outside the electrode region. 64

5.10 Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close and (b) far from electrodes. And between points (c) inside the electrode area and outside of it. Green Functions dataset 65

5.11 Visualization of resulting activation pattern when attempting to activate point (8,8) using COMSOL data as an input. In the figure, the relative position between the target and the electrodes can be distinguished. 66

5.12 Representation of how the VSE and the percentage of wrongly activated pixels distribute along the overall dataset extracted from COMSOL at 100 μm cut plane 67

5.13 Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close, (b) far from electrodes, (c) inside and (d) outside electrode area for COMSOL dataset extracted at 100µm cut plane. 68

5.14 Visualization of resulting activation pattern when attempting to activate point (8,8) using COMSOL data at 1µm cut plane as an input. In the figure, the relative position between the target and the electrodes can be distinguished. 69

5.15 Representation of how the VSE and the percentage of wrongly activated pixels distribute along the overall dataset of data extracted from COMSOL at a 1µm cut plane. 70

5.16 Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close, (b) far from electrodes, (c) inside electrode area and (d) outside electrode area, for the COMSOL dataset extracted at 1µm cut plane. 71

5.17 Representation of how the VSE distributes along the overall dataset distinguishing between points close (a and c) and far from electrodes (b and d) for COMSOL dataset extracted at 1µm cut plane (a and b) and 100µm cut plane (c and d) when intensity minimization is also performed. 74

5.18 Representation of how the VSE distributes along the overall dataset distinguishing between points inside (a and c) and outside the electrode area (b and d) for COMSOL dataset extracted at 1µm cut plane (a and b) and 100µm cut plane (c and d) when intensity minimization is also performed. 75

5.19 Multitarget stimulation optimal solutions visualization. 76

5.20 Comparisson between activating pattern for target points [9,7] and [5,7] when using (a) bipolar stimulation (b) vector field stimulation. 78

5.21 Visualization of histograms comparing the VSE with Green Function data, (a) Exponential vs (b) non Exponential Objective function . . . 79

5.22 Visualization of activated patterns for target points [9,1] and [7,3], with (a and c) Exponential objective function and (b and d) with original objective function 80

6.1 Activation patterns generated using different stimulation methods. (a) Activation pattern generated using bipolar stimulation in this study. (b) Activation pattern generated using vector field stimulation in this study. (c) Activation pattern generated using bipolar stimulation as presented in [1]. (d) Activation pattern generated using vector field stimulation as presented in [1]. 83

C.1 Electric potential field for electrode 1 103

C.2 Electric potential field for electrode 2 103

C.3 Electric potential field for electrode 3 103

C.4 Electric potential field for electrode 4 103

C.5 Electric potential field for electrode 5 103

C.6 Electric potential field for electrode 6 103

C.7 Electric potential field for electrode 7 103

C.8 Electric potential field for electrode 8 103

C.9 Electric potential field for electrode 9 103

C.10 Generated electric potential fields for each electrode.Data extracted from a 2D cut plane at 100μm above the tips of the electrodes from the 3D COMSOL model 103

D.1 Electric potential field for electrode 1 105

D.2 Electric potential field for electrode 2 105

D.3 Electric potential field for electrode 3 105

D.4 Electric potential field for electrode 4 105

D.5 Electric potential field for electrode 5 105

D.6 Electric potential field for electrode 6 105
D.7 Electric potential field for electrode 7 105
D.8 Electric potential field for electrode 8 105
D.9 Electric potential field for electrode 9 105
D.10 Generated electric potential fields for each electrode. Data extracted from a 2D cut plane at 1μm above the tips of the electrodes from the 3D COMSOL model 105

E.1 Electric potential field for electrode 1 107
E.2 Electric potential field for electrode 2 107
E.3 Electric potential field for electrode 3 107
E.4 Electric potential field for electrode 4 107
E.5 Electric potential field for electrode 5 107
E.6 Electric potential field for electrode 6 107
E.7 Electric potential field for electrode 7 107
E.8 Electric potential field for electrode 8 107
E.9 Electric potential field for electrode 9 107
E.10 Generated electric potential fields for each electrode. 107

F.2 Activated patterns algorithm output for Green function input data for every pixel in Fig. 5.5a. 110

G.2 Activated patterns algorithm output for COMSOL 100μm input data for every pixel in Fig. 5.5a. 113

H.2 Activated patterns algorithm output for COMSOL 1μm input data for every pixel in Fig. 5.5a. 116

I.2 Activated patterns algorithm output for COMSOL 100μm input data for every pixel in Fig. 5.5a including intensity minimization. 119

J.2 Activated patterns algorithm output for Green Function input data for every pixel in Fig. 5.5a including exponential objective function. . . . 122

List of Tables

2.1 Electrode damage thresholds in charge per phase [46]. 14

3.1 Compilation of optimization problem formulations across the literature 30

5.1 Comparison of algorithm performance metrics across Study Cases 1, 2, and 3. 71

5.2 Comparison of algorithm performance for the different study types with and without intensity minimization 73

Chapter 1

Introduction

1.1 Background

Blindness is a condition that affects millions of people around the world [3, 4, 5]. The World Health Organization estimated in 2019 that there were around 1 billion blind and moderately to severely visually impaired people worldwide [6]. All indications are that this number is only going to go up in the upcoming decades due to the ageing of the population [5]. This illness has a big negative impact on its patient's lives, hence, several investigations have placed blindness at the same level as cancer, as the second most feared condition to suffer from [7]. Therefore, there is a clear need for the development of a treatment or a device that could enable these people to have a much better quality of life. This is where visual prostheses come into play. These devices are designed to restore vision using electrical stimulation [8]. Throughout this project, the focus is going to be on a type of visual aid known as cortical visual prostheses.

1.2 Problem statement

Cortical visual prostheses aim to restore vision by implanting electrodes into the patient's cortex and electrically stimulating its neurons to artificially evoke sight. The visual cortex is organized in such a way that, electrically stimulating in a 2D grid of neurons directly evokes the same shape of light in the visual field. Hence, the higher the spatial density of electrodes, ideally matching the density of neurons, the higher the quality of visual restoration. However, since the human cortex has a surface area of around 1000 cm^2 and contains approximately on the order of 4–6 billion neurons [9], covering the entire visual field with single neuron resolution would require 4 to 6 billion electrodes. Implanting such a high number of electrodes poses severe challenges in terms of surgical risks and power-consumption for the stimulation electronics, which can lead to heat increase and tissue damage. Therefore, there is a critical need to enhance the spatial resolution of stimulation without increasing the number of electrodes. This project seeks to investigate a novel technique known as Vector Field Stimulation and to develop an optimization algorithm to advance this approach, aiming to achieve high spatial resolution while minimizing the need for additional electrodes.

1.3 Chapter organization

This thesis will be outlined as follows: First, Chapter 2 presents the background knowledge and theory related to visual prostheses and visual cortex microstimulation. Chapter 3 compiles a literature review, providing an overview of current optimization methods for stimulation and state-of-the-art region of interest stimulation techniques. Also, in Chapter 3, a novel method for stimulation, vector field stimulation, is introduced.

Chapter 4 details the methods used to simulate the generated electric fields, as well as the development and implementation of the optimization algorithm. The results of the algorithm and its comparison to bipolar stimulation are presented in Chapter 5. Finally, Chapter 6 discusses the results and provides future recommendations, concluding with Section 6.3.

Chapter 2

Background and theory

2.1 Visual prostheses

Visual prostheses are created to restore sight in patients who suffer a loss of their visual function Fig. 2.3. The three main tasks these devices must perform are: obtaining information about the environment in which the user is located, processing this information, and finally transmitting it to the patient [10].

The visual pathway is the route in which visual stimuli are transmitted from the eyes to the brain for their interpretation. Photoreceptor cells, that are found in the retina, capture the light that enters through the eyes and convert it to electrical signals. These signals are sent along the optic nerve to the lateral geniculate nucleus (LGN) where the first processing occurs, next, information is transmitted to the primary visual cortex in the occipital lobe of the brain where the information is reprocessed and interpreted to get a perception of the visual input (see Fig. 2.1) [11]. To restore lost visual function, several sites in the visual system can be stimulated, some of the regions studied throughout the literature are the subretinal, suprachoroidal, and epiretinal region of the eyes (for retinal prostheses); the optic nerve, the lateral geniculate nucleus (LGN) and the visual cortex (see Fig. 2.2) [13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. The focus of this research is on cortical visual prostheses since the visual cortex has some advantages over the other mentioned sites. Their design can be supported by already existing accurate mathematical models that describe the visual cortex [7, 24].

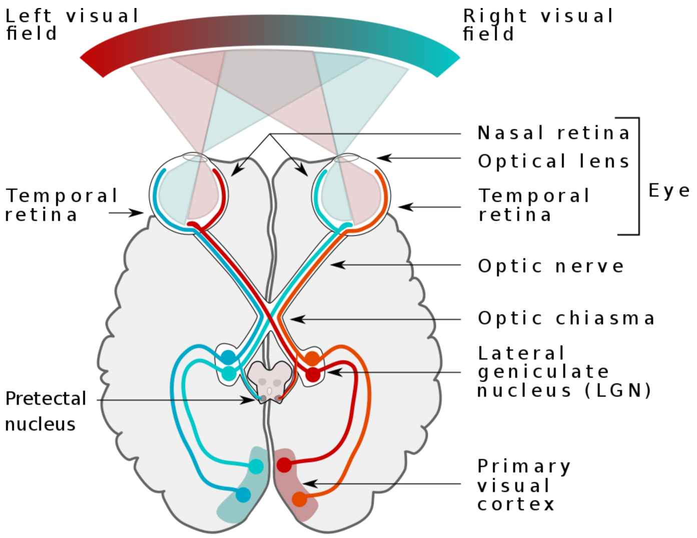


Figure 2.1: Human Visual pathway [12].

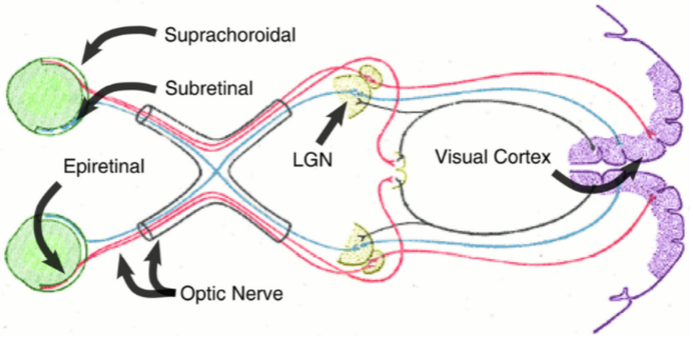


Figure 2.2: Possible stimulation sites for a visual prosthesis [23].

Moreover, they can succeed when other stimulation sites do not. For instance, in cases where the retina is damaged or there is another problem along the visual pathway, visual cortical prostheses would be effective since they are placed higher up the visual pathway [25]. Some examples of conditions that cannot be treated by prostheses placed in the early stages of the visual pathway, such as the retina or optic nerve, are glaucoma and diabetic retinopathy (DR) [25]. DR is a secondary effect of diabetes mellitus and is a microvascular complication that leads to progressive retinal damage. On the other hand, glaucoma is related to the circulation of fluids in the eye which can be caused by either a tumor, inflammation, or damage in the retina caused by trauma and causes irreversible vision loss by degeneration of retinal ganglion cells [25].

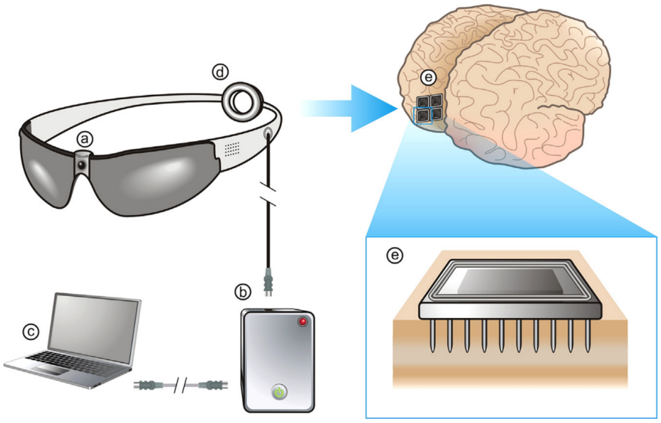


Figure 2.3: Cortical visual prostheses system [8]. Where the following components can clearly be distinguished: an image-capturing device, the glasses (a,d), an intra-cortical microstimulation array (e), and an image-processing device, the computer (c).

2.2 Microstimulation of visual cortex

2.2.1 Target regions of cortical visual prostheses

Cortical visual prostheses work by electrically stimulating neurons in the visual cortex to artificially evoke sight. Small light dots are generated by stimulating the visual cortex, known as phosphenes. These phosphenes can be plotted onto a map of the visual field, generating an intuition of the relationship between certain areas of the visual cortex and the phosphenes it generates. By combining several phosphenes, it is possible to create an artificial pixelized image [3][24].

The visual cortex forms part of the cerebral cortex and oversees visual perception and cognition [24], it is divided into five regions (V1-V5) (see Fig. 2.4b) and as you move from one region to the next the information becomes more abstract [10]. Region V1, also known as the primary visual cortex, acts as the end point of the LGN (see Fig. 2.1) projections and, therefore, is a key region in the visual pathway [25]. V1 is vertically divided into 6 layers. Layers 2 and 3 are known as supragranular layers, indicating their involvement in higher-order processing, they contribute to the integration of information from other cortical areas. On the other hand, layer 4 is the input/granular layer, that receives the direct input from the LGN [5]. This makes it

an ideal target for stimulations since its neurons are likely to evoke visual percepts. Finally, layers 5 and 6 are known as the infragranular which are in charge of sending information to other cortical areas, they are involved in the control and coordination of motor responses based on visual stimuli [26][27] (see Fig. 2.4a).
Based on the literature [28] layers 4 and 5 presented the highest neuronal stability after electrode implantation, making them more suitable for long-term stimulation scenarios like this one. Also, in the literature, the V1 region has been reported as a region able to evoke higher-resolution images due to its larger magnification factor in comparison to retinal prostheses [3]. Therefore, intracortical visual prostheses aim to target layer 4 of V1 [25, 24, 26, 29].

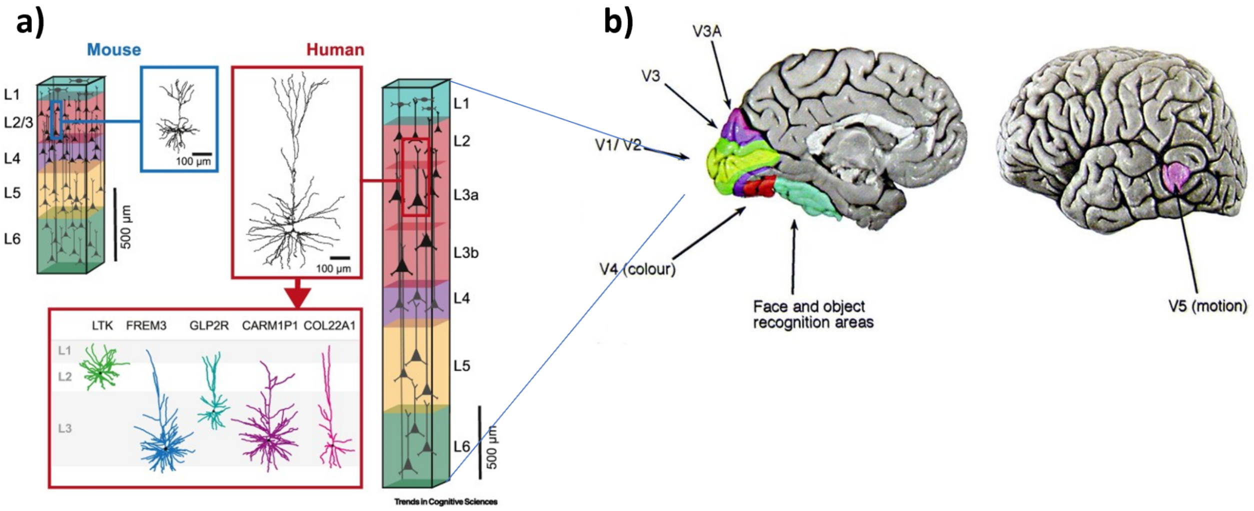


Figure 2.4: a) Layers of V1 , b) Regions of Visual cortex
(Figure adapted from [30, 31]).

2.2.2 Intracortical vs surface microstimulation

There are two main approaches to stimulate the target area, layer 4 of V1. Cortical surface stimulation and intracortical stimulation [24]. In the first approach, electrodes are placed right on the surface of the brain where electrical stimulation is applied. On the other hand, with intracortical stimulation, penetrating electrodes are implanted right in the target region layer 4.

Although the intracortical approach is more invasive, it would be the method used in the implementation of this project. This is because the stimulation amplitudes needed to evoke phosphenes are much lower, and it would avoid some problems such as cross-talk between other electrodes characteristic of the surface stimulation technique. Furthermore, Meikle et al. [32] claimed that the smallest phosphenes were produced by the electrodes that were implanted into the cortex. This phosphene type is desirable for achieving high-resolution images [32]. Moreover, Foroushani and his group (2018) probed that only a current from 1.9–25 µA was needed for evoking phosphenes when using intracortical microelectrodes that were placed in a 2mm depth of the human V1 [3]. They also claimed that from the first intracortical visual prostheses, it was observed that a current 100 times lower than the one required for surface electrodes was needed to evoke phosphenes. Furthermore, it is possible to implant hundreds of electrodes on a very small surface of the cortex [5]. With all the listed advantages, this application favors intracortical stimulation over surface stimulation.

2.2.3 Current mode vs voltage mode

Electrical stimulation in the cortex can be applied mainly using two methods: voltage mode stimulation (VMS) and current mode stimulation (CMS). This project will be developed on the basis that the method of stimulation used will be current mode stimulation. This choice of methodology is because it has proven to be safer since it allows for the control of the amount of charge delivered to the patient [23].

Otherwise, if too much charge were to be delivered there would be an accumulation of it in the electrode-tissue interface that could lead to non-reversible oxidation reactions, harming that way the patient and the electrode [25].

To understand how this method works, it would be useful to look at the generalized form of Ohm´s law, which describes the relationship between voltage and current in the following way:

$$I = V/Z. \tag{2.1}$$

This states that the current is equal to the voltage divided by the impedance (Z). The impedance is non-linear and accounts for the opposition that both the electrodes and the tissue make to the electrical current flux [33]. In the tissue, this opposition combines the resistance generated by the intra and extra-cellular fluid, and the membrane's capacitive reactance [33]. In CMS, a constant current source is applied to the tissue, so in this case, the user can control the current which is kept constant, and the voltage is regulated based on changes in tissue impedance [34]. Unlike in VMS, the amount of current delivered to the tissue is independent of the impedance [35], which is a much safer option for both the tissue and the electrodes used. However, this comes at the cost of power efficiency since there are power conversion losses [24]. Therefore, energy consumption must be taken into account at every step of the development of this project, and efforts must be made to ensure its optimization.

2.2.4 Phosphene generation

To induce phosphenes, the activation of neurons is essential. To achieve this, the injected currents must be able to induce the opening of the voltage-gated sodium channels within the cell membranes of the V1 neurons. If these currents can depolarize the membranes over their resting potential of -70mV, an influx of sodium occurs, amplifying the depolarization. This process initiates an action potential, which subsequently propagates along the neurons, enabling communication with other neurons [36]. In Fig. 2.5 a schematic representation of these action potentials can be studied. The nervous system is composed of several types of neurons, each with its unique functions. The activation threshold that needs to be achieved to produce action potentials changes among these different types. Several studies have shown that the predominant neuron type in V1 that mediates phosphene induction is pyramidal neu-

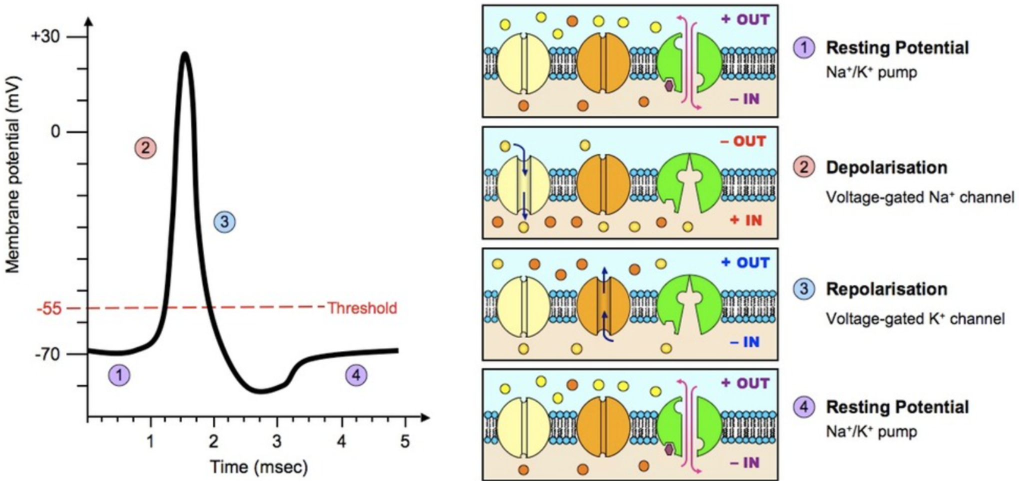


Figure 2.5: Action potential graphical representation [37].

rons [3], which have an activation threshold from -60mV to -55mV [38]. So, to evoke phosphenes the applied stimulus should surpass the targeted neuron activation threshold. Another relevant term that should also be considered is phosphene threshold, which refers to the minimum stimulus required to evoke the perception of phosphenes in a patient. Several factors can influence this term, including the depth in which the electrodes are implanted, the materials and impedance of the electrodes, the spacing between different electrodes, and the established parameters for the applied stimulation [3, 39].

Therefore, the strength of the applied stimuli, its amplitude, the pulse polarity (cathodic-first or anodic-first), the pulse duration, the frequency, the inter-train interval, and the train length of the applied stimulus have a direct impact on the phosphene threshold.

The most optimal values reported in the literature for some of these parameters have been frequencies from 100-200 Hz which have been stated as the most effective in humans [40, 3], and pulse durations between 0.01 and 1 ms [40, 3]. Regarding the pulse polarity, for intracortical electrodes, cathodic-first pulses have been proven to be more effective in generating phosphenes when compared to anodic-first pulses [40]. For the amplitude, several values have been reported, as they tend to vary based on the user and the specific type of electrodes used [3]. For example, [3] reported that with amplitudes 1.9–25, μA phosphenes were evoked using intracortical microelectrodes that were placed 2 mm inside the V1 region.

Moreover, [3] asserted that introducing brief interruptions during prolonged stimulation trains enhances the duration over which phosphenes are perceived.

Additionally, some other assumptions that can be concluded from their study are that a decrease in phosphene threshold could be triggered by increasing any of these parameters: electrode depth, distance between electrodes, frequency, pulse duration, train length, or the number of stimulation electrodes. However, the number of times the stimulation is applied will cause the phosphene threshold to increase, making it harder to evoke phosphenes in the patient.

2.3 Microstimulation: Fundamental concepts and Safety

2.3.1 Microelectrode arrays (MEAs)

An accurate way to deliver precise electric microstimulation to specific neurons in the V1 is by using MEAs. Generally, the microelectrodes used in microstimulation have surface areas of less than 2000µm^2 [35].

A common MEA used in long-term human clinical studies, which has been approved by the Food and Drug Administration (FDA) is the Utah array [41, 42]. It is made of a silicon-based matrix that encases a grid of penetrating microelectrodes. The arrays range in configuration from 2x2 to 12x12 microelectrodes per array [43]. These electrodes are spaced by 400 µm and have a 1.5mm length (see Fig. 2.6). The most common materials used for stimulation electrodes are platinum and iridium oxide (IrOx) [35], which are used to coat the microelectrodes tips, to enhance their ability to deliver current [5]. For stimulation applications, IrOx presents high charge storage capacity and lower impedance in comparison to other electrode materials [44].

2.3.2 Electrode tissue interface

The interaction between intracortical electrodes and the neural tissue of V1 is known as the electrode-tissue interface. Many electrochemical reactions take place in this

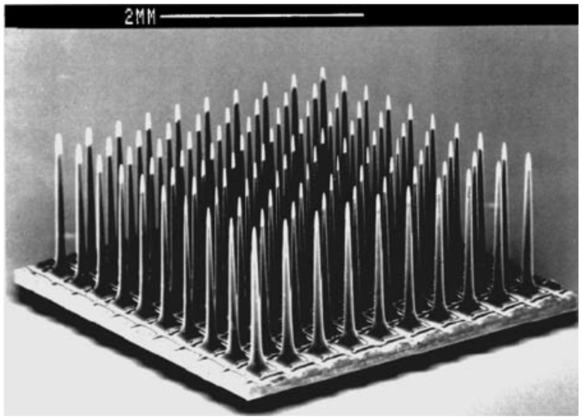


Figure 2.6: Utah array [45]

region that determine the efficiency and safety of the microstimulation [35]. The electrode potential applied to the intracortical electrodes influences the behavior of the water molecules in the tissue, generating the electrode double layer.

When an electrical potential is applied to the intracortical electrodes embedded in V1, the water molecules in the nervous tissue orient themselves in response to the potential and get attached to the electrode surface forming a layer classified as Helmotz layer in the literature [8], surrounding it is the diffuse layer which is formed by charged ions in the tissue that are attracted to the electrode by the polarity of the electric field that was created by the applied stimulation. This layer is more movable and is also surrounded by water molecules (see Fig. 2.7). This double layer can be electrically considered as a charged capacitor and can influence the effectiveness of the stimulation applied [8]. High current densities could destroy this bilayer, which would lead to a dielectric breakdown. At this point, the capacitance of the electrodes is lost and Faradaic reactions can occur. These reactions can cause the microelectrodes to reduce or oxidize and could lead to tissue damage [8].
Therefore, to ensure safety, the Faradaic effect should be limited and charge balance should be guaranteed during microstimulation.

2.3.3 Safety thresholds

Two main threshold limits should be taken into account when applying microstimulation, one is the tissue damage threshold, and the other one is the electrode damage

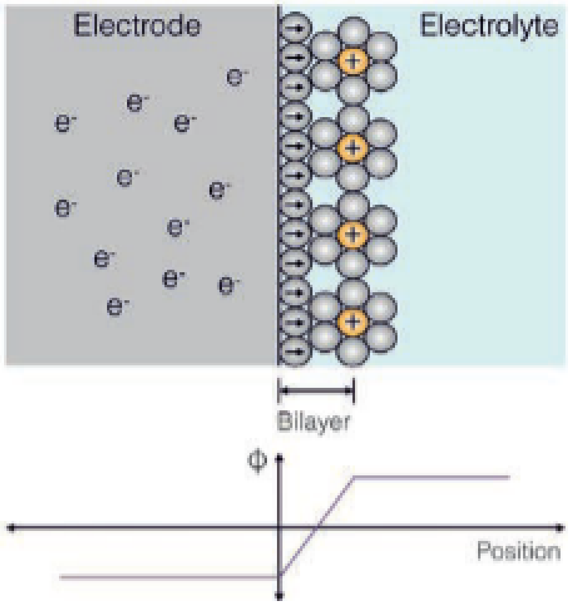


Figure 2.7: Electrode double layer [34].

threshold.

An important parameter to establish these threshold values is the type of stimulation pulses used. Typically, for cortical microstimulation the type of stimulation pulses used are biphasic with a leading cathodic phase, with the charge in the leading phase equal to the charge in the trailing phase (see Fig. 2.8). Two parameters based on

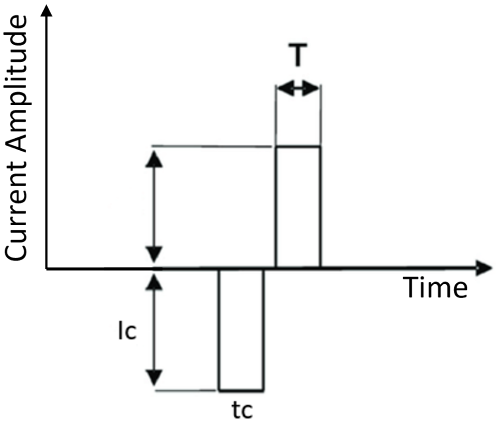


Figure 2.8: Biphasic stimulating pulse, cathodic-first [46].

these pulses that will affect determining the tissue and electrode damage thresholds are:

- **Charge per phase:** which is defined as the current times the pulse width:

$$I_c \times t_c = q_c \frac{\mu C}{\text{ph}} \tag{2.2}$$

- **Charge density :** which is the charge per phase divided by the geometric surface area of the electrode used.

2.3.3.1 Tissue damage threshold:

This threshold establishes the limit in which safe stimulation is delivered to the patient, it indicates the amount of current that can be injected without causing tissue damage [47].

The most reported value in literature for a safe limit on the charge density of a stimulation pulse is 30 μC/cm^2 [48, 46]. This limit was reported by Shannon with the Shannon equation Eq. (2.3) [49], where D is charge density, Q is charge per phase, and k is the adjustable parameter. Shannon summarized the observations from a study by McCreery et al [50] where they evaluated the tissue damage observed in the brains of animal models after electrical stimulation.

$$\log(D) = k - \log(Q) \tag{2.3}$$

Moreover, most safety values given in the literature are given for big area electrodes, also referred to as macro electrodes, in the order of $10^{-3} cm^2$, mostly used in spinal cord stimulation or Deep Brain Stimulation (DBS). However, the study case of this project is based on a much smaller target area where microelectrodes are needed which are in the order of $10^{-4} cm^2$.

Cogan and their research group also studied the same for cases where penetrating microelectrodes were used [46]. They observed that the tissue damage thresholds for the microelectrode cases do not follow the Shannon line anymore, and in almost every instance the injected current density was well above 30 μC/cm^2 [47]. Based on their observations, the tissue damage threshold for long-term microstimulation with microelectrodes embedded in the cortex should be 4nC/ph [47]. However, the tissue damage threshold is not only dependent on the charge per phase but also on the frequency of the pulses and the duty cycle of the stimulation [47].

Therefore, these conclusions should be taken into account when defining the tissue damage threshold of this project.

2.3.3.2 Electrode damage threshold:

This threshold is established for the electrodes themselves since the amount of charge density injected can also damage the electrode. Surpassing this threshold could cause the electrode to degrade or malfunction [51]. As mentioned in Section 2.3.2, for electrode safety, water electrolysis should be avoided. Table 2.1 shows the charge per phase threshold for the most common materials used in microelectrode manufacturing [46].

For 2000 μcm^2 Electrode	**Charge per phase (nC/ph)**
Platinum	2
Iridium Oxide	60
Titanium Nitride	20

Table 2.1: Electrode damage thresholds in charge per phase [46].

2.3.4 Activating Function

The activating function (AF) is a mathematical approximation that computes the electric-field gradient along the axons of the neurons. It is useful to assess where neurons will be activated, giving an intuition of what areas are receiving the strongest excitation. This is possible because this function helps estimate the generation of action potentials in the targeted neuron. Consequently, it helps evaluate the spatial extent of neuronal activation.

The AF is proportional to the inward and outward currents along a neuron and its deviation from the resting potential [52]. It pinpoints the sites of elevated hyperpolarisation and depolarization caused by the electric field that is acting on the neuron. The AF is based on the cable theory model of the axons of neurons. This model assumes that neurons can be represented as core conductors with electric circuits that approximate the behavior of the membrane see (Fig. 2.9a). The derivation of the cable equation comes from applying Kirchhoff's law and Ohm´s law over the model [53, 54]. According to this theory, the transmembrane voltage dynamics of the different axonal segments can be described by the following equation [55]:

$$\frac{C_m dV_i}{dt} = \left[\frac{d}{4\rho_i}\left(\frac{V_{i-1} - 2V_i + V_{i+1}}{\Delta x^2} + \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{\Delta x^2}\right) - \sum I_n^{ion}\right] \tag{2.4}$$

Where d is the diameter of the axon which normally is reported to be between 1 to 10 μm; V_{i-1},V_i and V_{i+1} are the transmembrane voltages of the neighboring axonal compartments; C_m is the membranes capacitance per unit square that has a value of 1 μF/cm; ρ_i is the resistivity of the axoplasm which is 300 Ω·cm; ϕ_{i-1}, ϕ_i and ϕ_{i+1} refer to the extracellular potentials in the vicinity of axonal compartments, $\sum I_n^{ion}$ is the sum of all intrinsic ionic currents and x is the direction of the axonal fiber. The cable model is visualized in Fig. 2.9b. To predict the membrane response from the cable model Fig. 2.9 it is assumed that at the beginning of the stimulation, the capacitive currents across the membrane are predominant over the intracellular currents. Based on this approximation the initial polarization of the membrane is proportional to the source term of the cable equation, this is called activating function [58, 59]. In its generalized form, the activating function is defined as the second spatial derivative of the potential multiplied by a constant Eq. (2.5).

$$AF = \frac{d}{4\rho_i} * \frac{\partial^2 V}{\partial x^2} \tag{2.5}$$

When analyzing this function, it should be noted that a positive value indicates depolarization of the membrane, while a negative value indicates hyperpolarisation

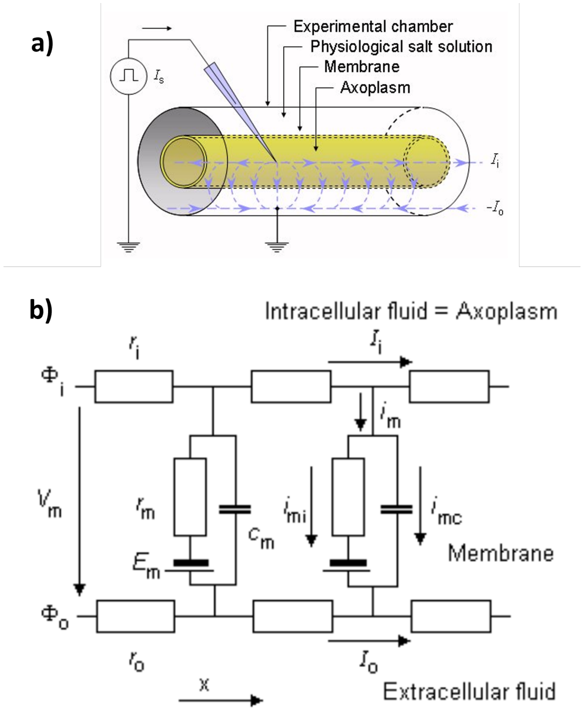


Figure 2.9: Cable model of the axon of the neuron. **a)** Experimental setup used for deriving the cable equation. It assumes that the total stimulus current (I$_i$), which has an axial flow inside the axon, decreases with distance since part continually crosses the membrane to return as a current (I$_o$) outside the axon. It also considers the axon to have an infinite length. **b)** Circuit model of an axon. Figure adapted from [56, 57].

[60].

2.3.5 Current Steering

Current steering is a technique that manipulates the distribution of electrical potential around different electrode sites. This method involves delivering current simultaneously to adjacent electrodes, allowing stimulation to be directed to a point in between the contacts. This is achieved by adjusting the proportion of current delivered to each electrode pair [61].

Normally interactions between electrical fields are aimed to be avoided since they can produce the activation of undesirable locations and produce unwanted responses in the body such as seizures [62]. However, this technique uses these interactions to its advantage. By overlapping individual electrode fields, current steering can focus or redirect the current to excite different tissue sites. This way allows to increase the number of stimulating channels beyond the fixed number of physical electrode contacts [63, 26, 61, 32, 62]. This concept is illustrated in Fig. 2.10.

It has been experimentally incorporated into different devices, such as retinal, optic nerve, cochlear, cortical, and facial muscle implants [32].

Overall, this technique offers a great opportunity for enhancing the performance of neurostimulation devices by improving the range of stimulation possibilities they can achieve. Therefore, its implementation will be considered in the course of this project.

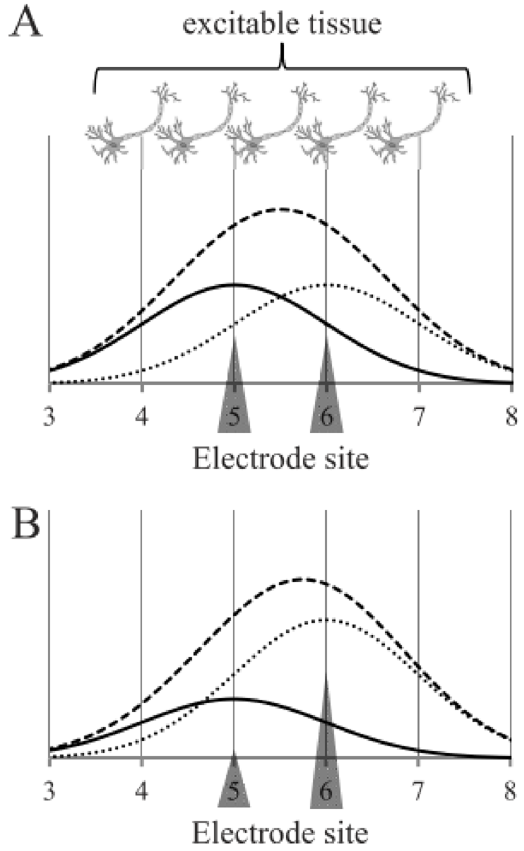


Figure 2.10: Current steering using electrode pairs. The electric fields generated by the stimulation are represented by a continuous line for electrode 5 (E5), a dotted line for electrode 6 (E6), and a dashed line for simultaneous stimulation of both electrodes. **A)** Both electrodes are stimulated with the same amount of current. **B)** E5 is stimulated with 30 % of the total current, while E6 is stimulated with 70 % of the current [61].

2.4 Problem statement

The literature shows that location, proximity, and number of electrodes placed in the brain dictate the resolution of artificial vision [32]. Moreover, several studies have reported that to restore visual sight around 1500-2000 phosphenes would be needed [32]. Therefore, knowing that conventionally each physical electrode evokes one phosphene [32], it can be concluded that thousands of electrodes are required to restore normal vision. However, implanting that number of electrodes will harm the brain [64]. The more electrodes implanted, the greater the tissue damage. Therefore, there is a need for stimulating methods that would improve the resolution of the visual cortical implants without increasing the number of electrodes that would be implanted into the brain. This project aims to address this issue and try to find a way to enhance microstimulation in the visual cortex.

Chapter 3

Prior work

3.1 Electric field shaping

This section will delve into electric field shaping methods, what techniques are currently used in clinical practice, in what areas of the body, and how they can improve the stimulation of the visual cortex.

3.1.1 Current steering

The predominant field shaping technique mentioned within the literature is current steering, which concept was introduced in 2.3.5. This study aims to investigate its potential to enhance visual prosthesis performance.

For example, Firszt et al. in their study probed that additional pitch precepts could be generated using current steering in cochlear implants through concurrent stimulation of contiguous electrodes [63]. Moreover, the study by [26] indicated that the current steering effect is lost when electrodes are placed in different layers of the cortex. However, when they were placed in a single layer or parallel to the cortical surface, there was a clear current steering effect.

Also, from their single-layer measurements, they observed a positive relationship between the current level, shift, and spread of the stimulating electrodes. They also concluded that the lack of a current steering effect in the multi-layer approach was due to the spread of neural activity throughout the cortical column and not to experimental errors [26].

In addition to its application in cochlear implants, [61] studied the feasibility of incorporating this technique in retinal prostheses. They wanted to redirect the current to excite different tissues. To do so, they split the current between the electrodes by Eq. (3.1).

$$I_a = R \times I_t \tag{3.1}$$

Where I_a resembled the current on the first electrode of the pair, I_t was the total current, and R is the current ratio from 0 to 1 which determines the proportion of current sent to the first electrode. Therefore, the current of the second electrode follows:

$$I_b = I_t - I_a. \tag{3.2}$$

They generated a cortical spatial map for each ratio by plotting the spike rate across all their different recording channels. Then they plotted all centroids for each current ratio of an electrode pair on a single cortical map to evaluate how centroids shifted based on the ratio applied.

Their results showed that it was possible to relocate the centroid of cortical spike activation with current steering within the retina using numerous electrode pairs with different retinal distances. They observed that greater adjustments in the current ratio and larger distances (for both virtual and physical electrodes) corresponded to more substantial shifts in the cortex centroid. While they couldn't determine the limit of electrode pitch for active current steering, they successfully demonstrated the feasibility of generating additional virtual electrodes between two physical electrodes. Moreover, they suggested that applying current steering in two dimensions (2D) could enhance this process further.

In line with our goal, [32] investigated the feasibility of using current steering to enhance the resolution of artificial vision provided by intracortical prosthesis while minimizing the need for additional physical electrodes in the brain. In their study, they used a 64-channel, four-shank microelectrode array implanted in the visual cortex of rats. They aimed to study how the position of neuronal activation could be steered by varying the charge distribution between two stimulating electrodes separated by either 300μm, 400 μm and 500 μm.

To assess current steering viability across different electrode array configurations, they explored the relationship between the spread of neural activity, the pitch between electrodes, the peak shift of neural activity, and the stimulation amplitude. Their results suggested that achieving precise phosphenes requires selecting the lowest current amplitude that reliably shifts neural activity peaks without reaching saturation. They concluded that 6 μA was the ideal current for inducing minimal neural activity spread with a stable peak shift across various electrode separation distances. Additionally, they proposed using retinotopic maps of the visual cortex to predict the spatial location of evoked phosphenes based on the current steering response [65].

Moreover, [32] compared two models that predicted current steering response based on single electrode responses. They concluded that a weighted model, scaling cortical response to each electrode when individually stimulated, was more accurate than a linear additive model. The cumulative model is formulated as Eq. (3.3).

$$d \times R_{E_1} + (d - 1) \times R_{E_2} = R_{E_1 x d + E_{2 \times (1-d)}{}_1} \tag{3.3}$$

Where d is the current ratio of the first electrode, R is the evoked response, and I the current. Finally, their results suggested that current steering could be used to increase the resolution of intracortical electrode arrays without altering the number of physical electrodes, hence, power consumption, tissue damage, and potential heat dispersion issues would be diminished.

Expanding the scope of other papers that focused on increasing the number of elicited phosphenes with "virtual electrodes" generated by simultaneous stimulation of electrode pairs, [62] proved that virtual electrodes could be generated using 2D current steering with up to 6 electrodes.

Moreover, [66] implemented current steering into a retinal prosthesis, and they demonstrated how using different return electrode geometries could modify the overall current density delivered. Their results suggested that the number and orientation of the return electrodes could have an important role in the patient's visual perception. This concept was further supported by [67], who conducted experiments to steer the injected current of retinal prostheses by employing various combinations of return electrodes. Through theoretical analysis and experimental validation using immersed electrodes, they demonstrated the feasibility and effectiveness of adjusting current distribution within the retina.

In conclusion, current steering has proven to be an interesting research topic for many stimulation devices. It has been successfully implemented in many research scenarios from DBS, cochlear implants, and retinal prosthesis to even sacral nerve electrostimulation to enhance their performance [68]. Throughout the literature, it has introduced additional pitch percepts in cochlear implants and even improved resolution in retinal prostheses. Furthermore, apart from current steering, [23] introduced similar techniques that can be included in the field shaping families, such as current focusing, virtual electrode stimulation, and phased array stimulation. The main difference [23] states between them is their main purpose, current steering focuses on controlling the direction and spread of the stimulation; virtual-channel stimulation aims to create a stimulus perception in between the primary sources of stimulation (create a virtual electrode) and current focusing tries to limit the extent of activation and concentrate the activation at the target location [23].

3.1.2 Vector field shaping

Vector Field stimulation is a novel technique that aims to achieve region of interest target stimulation [1]. It involves simultaneous current stimulation across multiple electrodes, hence multichannel stimulation. Unlike traditional current steering methods, which focus on controlling individual electrode stimulation levels, VFS seeks to regulate the overall electric field distribution generated by adjusting each electrode's contribution, see Fig. 3.1.

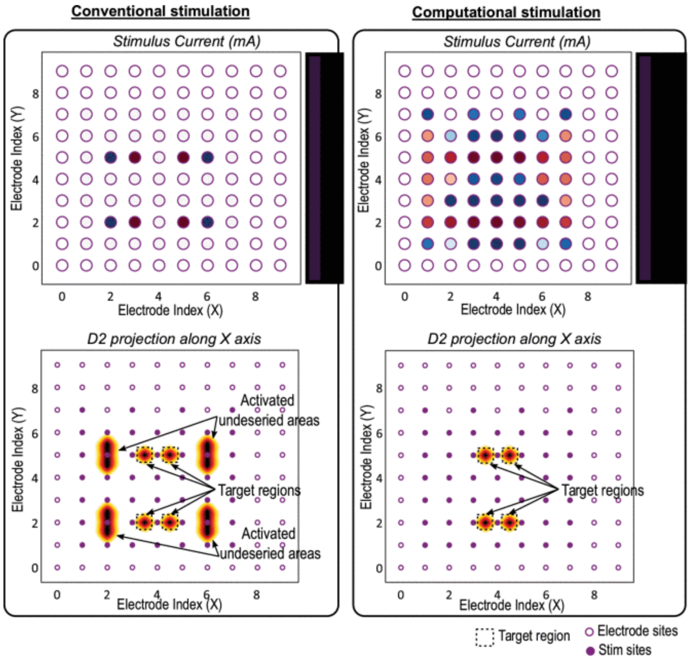


Figure 3.1: The figure demonstrates the Vector Field Shaping technique from [1]. On the left, the top images show the electrodes activated after differential stimulation, while the bottom images represent the resulting activation field. On the right, the images illustrate the activation pattern achieved with Vector Field Shaping, where more electrodes are activated, leading to a more precise activation field.

3.1.3 Proposed Implementation

Previous literature often uses paired electrodes for targeted stimulation [23, 63, 32, 65]. In contrast, our approach aims to independently control current delivery to each electrode to achieve a specific electric field configuration, thereby eliminating the need for electrode pairing constraints. The primary objective is to develop an optimization algorithm based on VFS that calculates the necessary current for each MEA electrode to generate a predefined potential field. By focusing on manipulating the overall electric field rather than individual channel currents, VFS introduces a novel perspective on current steering concepts.

Therefore the following section will explore different optimization techniques previously used in microstimulation.

3.2 State-of-the-art methods to optimize electrical stimulation processes

Throughout the years, many research groups have tried to optimize the problem of electrical microstimulation of neurons. While this project primarily focuses on intracortical microstimulation, this study of literature will also delve into additional techniques.
Based on the reviewed literature a common setup for the optimization process consists of 3 steps:

1. Calculate individual electrical contributions of each electrode using FEM
2. Optimize objective function
3. Analyse the resulting electric field

3.2.1 Calculate individual electrical contributions of each electrode using FEM

One of the first approaches studies take to optimize electrical stimulation is to calculate the electric potential field generated in their region of study by applying unitary currents to their electrodes. They do this by solving Poisson equation Eq. (3.4) using Finite Element Analysis (FEA) [60, 48, 1, 69, 2, 70, 7, 71, 72]. Where V is the resulting electric potential, J_p is the primary current density, and σ is the conductivity of the medium.

$$\nabla^2 V = \frac{\nabla J_p}{\sigma} \tag{3.4}$$

The purpose is to create a current density transfer matrix, that applying the superposition principle will serve as a basis to approximate all possible combinations of active contacts, exploiting the advantage of the system's linearity.

3.2.1.1 Origin of Poisson equation

Maxwell's equations describe the behaviour of electric and magnetic fields generated by neuronal activity or external stimulation in a continuous medium [59]:

$$\nabla \cdot \mathbf{D} = \rho \tag{3.5}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{3.6}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{3.7}$$

$$\nabla \times \mathbf{H} = \mathbf{J}_p + \sigma \mathbf{E} + \frac{\partial \mathbf{D}}{\partial t} \tag{3.8}$$

Where **E** represents the electric field, **D** is the displacement field, **H** the magnetic field, **B** the magnetic induction, and ρ is the volume charge density. Furthermore, **B**= μ **H**, where μ represents the magnetic permeability.
Also, the scalar potential V and the vector potential **A** can be related to the electric field **E** through:

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t} \tag{3.9}$$

Making Eq. (3.8):

$$\nabla \times \mathbf{H} = \mathbf{J}_p + \sigma(-\nabla V - \frac{\partial \mathbf{A}}{\partial t}) + \frac{\partial \mathbf{D}}{\partial t} \tag{3.10}$$

The speed at which the electromagnetic waves described by these equations propagate in a vacuum is the speed of light ($c = 3 * 10^8 m/s$). Bioelectric sources, such as action potentials, typically operate at frequencies below 1 kHz. At these low frequencies, the dynamic propagation effects of electromagnetic waves can be neglected. Instead, the fields are dominated by resistive and capacitive properties of the biological medium, leading to a much slower effective propagation speed compared to the speed of light [59]. In this case, all time-varying quantities in these equations would have a small effect on the overall electromagnetic field distribution. Therefore, a quasi-static approximation can be made, where these time-varying quantities are neglected.
Based on this assumption, and by taking the divergence of Eq. (3.10) we obtain Pois-

son´s equation Eq. (3.4).

When simulating the behavior of electrical stimulation, it is also necessary to define appropriate boundary conditions (BC). These conditions will ensure the simulation accurately represents the real world [59]. The choice of these boundary conditions differs between studies. Dirichlet and Neumann are the most used ones in stimulation applications [60, 48, 1, 2, 7].

- **Dirichlet Boundary condition**

  When electrodes are used to apply electrical stimulation, each electrode gets a voltage V_0 assigned to it relative to the ground, which conventionally is set to 0 V. Therefore, Dirichlet BC dictates the value of the potential field V at the boundaries [59]:

  $$V = 0 \quad \text{on the ground electrode} \tag{3.11}$$

  $$V = V_0 \quad \text{on the stimulating electrode} \tag{3.12}$$

  Although this BC is commonly used [60, 48, 1], it does not account for the potential drop that occurs in the electrode-tissue interface introduced in the Section 2.3.2.

- **Non-homogenous Neumann Boundary Condition**

  Unlike Dirichlet, this condition does account for the drop in the electrode-tissue interface. It is used to model the flow of charge between the electrode and tissue and is defined in the following way:

  $$\sigma \nabla V \cdot \mathbf{n} = \frac{I}{S_{elec}} \tag{3.13}$$

  Where σ is the conductivity of the tissue, $\mathbf{n}$ is the normal vector to the boundary at each point of the electrode surface, I is the injected current, and S_{elec} is the electrode surface area. It states that the normal component of the electrical current density at the electrode boundary is equal to the injected current per unit area [2, 7].

3.2.2 Optimize objective function

The objective function that must be optimized depends on the electrical stimulation application. The most repeated objective in the literature was to maximize the current density within the targeted regions while minimizing it across the rest of the domain [60, 48, 1, 2, 70, 72].

As with the objective function, the way the optimization problem is formulated strongly depends on the application for which the electrical stimulation is meant: spinal cord stimulation [60], deep brain stimulation (DBS) [48, 72], cervical stimulation in the spinal cord [1], electrical brain stimulation (EBS) [2], electrical stimulation of visual cortex [70, 7] or electrical stimulation of the retina [69].

A recurring approach in several studies was to use the activating function, previously introduced in Section 2.3.4. The main idea is to maximize this function in the regions of interest (ROI). Most investigations study this function in the context of 1 dimension (1D) [60, 72]. This approach represents the relationship between changes in the membrane potential at one point along the axon, caused by the delivered stimulation, and the likelihood of generating an action potential at another point along the same linear structure. Eq. (2.5) represents this relationship, which can approximate activation along a 1D fiber. Fig. 3.2 visualizes this concept, it presents the 1D activation produced by a surface electrode, for a fiber horizontal to the electrode (Fig. 3.2 c)), and a fiber vertical to the electrode (Fig. 3.2 d)). In [48, 1] they deviate from 1D and study the activating function in a 3D context. They used a generalization of the activation function to approximate the activation of neurons in all directions. They used a Hessian matrix Eq. (3.14), a mathematical structure formed by second-order partial derivatives of the electric potential. This method enabled them to approximate neural activation in a 3D space.

$$H = \begin{bmatrix} \frac{\partial^2 V_e}{\partial x^2} & \frac{\partial^2 V_e}{\partial x \partial y} & \frac{\partial^2 V_e}{\partial x \partial z} \\ \frac{\partial^2 V_e}{\partial x \partial y} & \frac{\partial^2 V_e}{\partial y^2} & \frac{\partial^2 V_e}{\partial y \partial z} \\ \frac{\partial^2 V_e}{\partial x \partial z} & \frac{\partial^2 V_e}{\partial y \partial z} & \frac{\partial^2 V_e}{\partial z^2} \end{bmatrix} \tag{3.14}$$

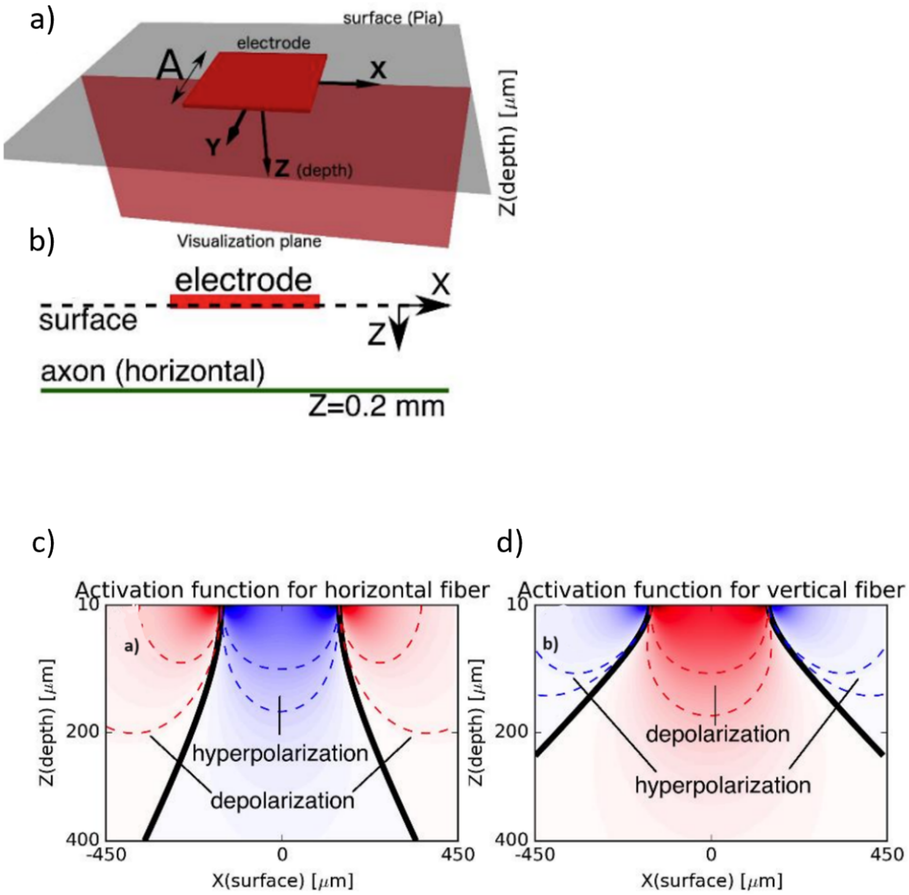


Figure 3.2: a) Set-up example where the electrode is in the coordinate system (X,Y,Z). This electrode is located on the surface (grey plane), and the centre of the coordinate system corresponds to the centre of the electrode. b) Schematic representation of the electrode (red) and horizontally oriented axon (green) on (X,Z) plane (Y=0). Activating function representation for **c)** horizontal and **d)** vertically oriented fibers, where red represents the areas of depolarization and blue areas of hyperpolarization [55].

This 3D approximation can be of great value to this project since it will allow us to study how the neurons interact and it will also help us understand how the electrical signals are propagating in our tissue.

Apart from AF optimization, [7] introduced another interesting objective function, visual error minimization. They introduced this concept in the context of visual cortex stimulation. This error was quantified in pixels, it accounts for the error generated by exciting location (j) when trying to excite location (i) by the following equation:

$$\Delta_{ji} = \frac{\sqrt{[x_i - x_j]^2 + [y_i - y_j]^2}}{P_{sz}} \tag{3.15}$$

Where Δ_{ji} is the visual error in pixels, j is the location excited by mistake, i is the targeted location, and P_{sz} is the desired pixel size which is defined as:

$$d_{neighbj} = \min \sqrt{[x_{visk} - x_{visj}]^2 + [y_{visk} - y_{visj}]^2}_{k=1..Nloc,k\neq j} \tag{3.16}$$

$$P_{sz} = \frac{1}{N_{loc}} \sum_{j=1}^{N_{loc}} d_{neighbj} \tag{3.17}$$

Where N_{loc} is the number of brain locations in the region of study (ROI). The desired pixel size represents the size and spacing between pixels and is obtained with the average distance between nearest neighbor phosphenes ($d_{neighbj}$). So if the pixels are large and very spaced, the resolution of the visual field will be low, each pixel will represent a larger area of the visual field, so there will be fewer pixels to capture the details of the image. Based on Eq. (3.15), larger pixel sizes correspond to a reduced impact of individual pixels on visual error.

On the other hand, if the pixels are small and close to each other, the resolution of the visual field will be high, where each phosphene will represent a finer detail.

By normalizing the distance between points by the pixel size in Eq. (3.15) visual error can be compared between different grid sizes (resolutions). Furthermore, to compute the total visual error, they sum all the errors obtained from all of the unwanted regions that had been excited:

$$\Delta_{\mathrm{j}} = \min\left(\Delta_{\mathrm{ji}}\right), \mathrm{i} = 1...\mathrm{N_{on}} \ and\ j = 1 \ldots N_{UDE} \tag{3.18}$$

$$\Delta_{tot} = \sum_{j=1}^{N_{UDE}} \Delta_j \tag{3.19}$$

Where N_{on} is the set of target locations (locations we would like to excite) while

N_{UDE} represents the number of locations that should have remained unexcited but were excited by mistake.

See Fig. 3.3 for a visual interpretation of this concept. Although the optimization algorithm will strongly depend on how the optimization problem is formulated, most studies reported in this literature use convex optimization. The following table compiles all the different objective functions used in the commented literature, their constraints, and the optimization approach they used to find a solution:

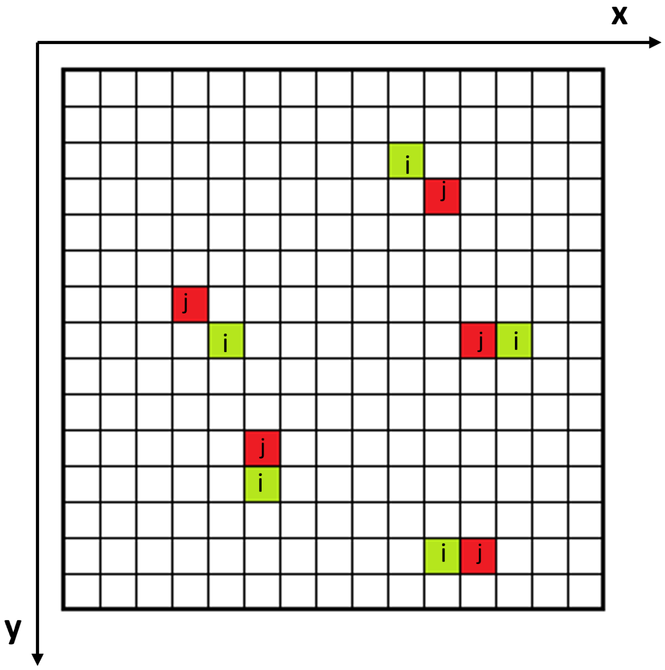


Figure 3.3: Visual interpretation of the described visual error in [7]. Where red represents the excited locations (j) and the green represents the target locations (i)

Table 3.1: Compilation of optimization problem formulations across the literature

Boundary conditions	Objective function	Constraints	Optimization problem	Reference
Dirichlet	$\min_{x} \frac{\iiint_{\Omega-\Omega^T} \sigma_x \left\|\nabla f_x^2\right\|^2 \, dx\, dy\, dz}{\iiint_{\Omega^T} \sigma_x \left\|\nabla f_x^2\right\|^2 \, dx\, dy\, dz}$ (3.20)	$I = \mathrm{YB}_w \ \alpha \geq 0$ (3.21)	Quadratic optimization	[60]
Dirichlet	$max \ d_\Omega\left(C\right) = \sum_{i=1}^{n} \frac{C_i}{\|\Omega\|} \int_\Omega \sum_{k=1}^{3} \lambda_k \left(H_i\left(x\right)\right) dx$ (3.22) where Hi is defined as 3.14	$e_\Omega\left(x,c\right) = \lambda_{\max}\left(\sum_{i=1}^{n} c_i H_i\left(x\right)\right) \leq \alpha \quad for \ every \ x \ in \ \Omega,$ (3.23) $0 \leq \ c \leq \ c_{max} \ where \ c_{max} \ was \ 30 \ \mu Ccm^{-2},$ (3.24) $e_{\Omega_\mathrm{T}}\left(x,c\right) \leq \alpha_T for \ every \ x \ in \ thalamus \ region \ \Omega_\mathrm{T}$ (3.25) $g_{\Omega_\mathrm{IC}}\left(x,c\right) \leq \alpha_{IC} \ for \ every \ x \ in \ IC \ region \ \Omega_\mathrm{IC}$ (3.26)	Linear convex optimization	[48]
Dirichlet	Not defined (N.D) Based on [48]	N.D Based on [48]	Convex optimization	[1]
Neumann	$min \frac{1}{N_{on}} \sum_i d_i + \frac{1}{N_{off}} \left(\sum_j d_{uj} + \sum_j d_{l_j}\right)$ (3.27)	$\forall \ locations \ i \ \epsilon \ E: \ T_i \ b + L \ d_i \geq \alpha_{exc},$ (3.28) $\forall \ j \ \epsilon \ O: \ T_j b - L d_{u_j} \leq \alpha_{off}, T_i b + L d_{l_j} \geq -\alpha_{off},$ (3.29) $-I_{max} \leq \ r_k \leq \ I_{max} \ \forall \ considered \ locations \ k,$ (3.30) $-b_{max} \leq \ b_p \leq b_{max} \ \forall \ electrodes \ p,$ (3.31) $-b_{max} \leq -\sum_P \ b_p \leq \ b_{max}$ (3.32)	Mixed integer linear	[2]
Neumann	Based on [2]	[2]	Convex optimization	[7]

3.2.3 Analysis of the resulting electric field

After optimizing electrical stimulation, some studies validated their results. For instance, [48] used NEURON software to validate the neural activation results obtained with their optimization algorithm. NEURON is a simulation software that allows the modeling of individual neurons and networks of neurons [73].

Furthermore, although in [70] is used to obtain the input to their algorithm, Benson´s retinotopic template could offer the possibility to validate the algorithm's predicted activation patterns with experimental data obtained from retinotopic studies. This template associates V1 locations to the visual field [74].

3.2.4 Conclusion

From all these studies there are some valuable insights we could use for our project, for instance, most of the studies, calculated the potential generated by different electrodes by solving the Poisson equation with FEM. Furthermore, in numerous cases, it was assumed that the region of study was homogenous and isotropic. Although this does not resemble the reality of our medium of interest, it would be a good first assumption to simplify our model.

Moreover, most of the problems defined in the literature were convex optimization problems, which established the current state-of-the-art to solve this kind of problem, optimization of electrical stimulation of neurons. Although all cases differ in the problem definition of the model they studied, their selected objective function and constraints could serve as a baseline or inspiration to define our model. Therefore, a good first step is to decide our objective function and desired constraints to formulate the optimization problem properly. Moreover, once our objective function is clear, we could use the problem formulations compiled in this literature as inspiration to develop our framework and as benchmarks to compare with.

Other observations made from this review, are that the safety current limits should be carefully selected, as well as the neuron excitation threshold, which are often calculated using HH models of the neurons.

Moreover, from these studies we can obtain some benchmark values to compare and validate our results, such as the visual error introduced in Abouelseoud et al. papers [2, 70, 7] or the activation thresholds obtained for pyramidal cell's activation in other studies. Furthermore, the results obtained in some of this paper could be used to compare the feasibility of the proposed algorithm.

3.3 Problem statement and research question

The goal of this project is to develop an optimization algorithm that would be able to enhance V1 microstimulation while adhering to all safety constraints required for a cortical visual prosthesis. This includes limiting the amount of current delivered to the brain, evading non-Faradaic reactions, avoiding or minimizing stimulation of undesired regions to ensure that there are no undesired reactions from the microstimulation delivered, and maximizing activation of all desired locations. The algorithm will be based on vector field stimulation, we want to be able to output the amount of current that is necessary to be delivered by each electrode in the MEA based on a desired electric field without the necessity of having the electrodes paired, as the recently introduced in Section 3.1.2. The main objective is to enhance the way microstimulation of the primary visual cortex is performed right now. The research question that this approach aims to solve is:

How does vector field stimulation affect visual error and power efficiency in a cortical visual prosthesis?

- How would these parameters be affected if we limit the number of electrodes that are allowed to be activated at a time?
- What happens if the electrodes within the MEA are grouped, and only one electrode in each group can be activated simultaneously?
- In this approach, the visual cortex will be represented as a grid. Can this

technique increase the percentage of possible activated points in the grid when compared to bipolar stimulation?

3.4 Project plan

Based on the literature discussed in this chapter, this project aims to demonstrate the advantages of the novel technique discussed, Vector Field stimulation (Section 3.1.2), over the methods currently in use. To do so, the questions discussed in the last section are to be solved. The plan is to follow the steps discussed in Section 3.2:

1. Calculate individual electrical contributions of each electrode in the MEA using FEM.

2. Optimize the objective function of our problem, which will be defined based on Vector Field stimulation. It was also discussed in this chapter, that most studies that tried to optimize microstimulation used the one-dimensional activating function to estimate the distribution of neural activation. However, in this problem formulation, I would like to use a 3D approximation approach since it will allow us to study how the neurons interact and help us understand how electrical signals propagate in our tissue.

3. Analyse the resulting electric field to prove the feasibility of this new technique.

The end goal of this thesis is to develop an optimization algorithm that validates the potential of this novel technique for use in cortical visual prostheses.

Chapter 4

Methods

4.1 Introduction

Based on the findings made in Chapter 2 and Chapter 3, the goal of this project is to develop an optimization algorithm that enhances V1 microstimulation while adhering to all safety constraints required for a cortical visual prosthesis. This includes limiting the amount of current delivered to the brain, evading non-Faradaic reactions, avoiding or minimizing stimulation of undesired regions, and maximizing activation of all desired locations. Additionally, this project aims to address the research question discussed in Section 3.3 regarding the effectiveness of the optimization algorithm under different constraints.

A robust and comprehensive problem formulation is essential to achieve these objectives. This chapter outlines the approach to designing, validating, and implementing the optimization algorithm.

4.2 Problem formulation

To solve the problem, and based on the information gathered in the literature Chapter 3, the main objective was determined to be minimization of the visual error. This error, as introduced in Section 3.2.2, is quantified in pixels and accounts for the discrepancy between the generated phosphenes and the desired locations.

The optimization algorithm must comply with several constraints to ensure safety in a cortical visual prosthesis. These constraints are:

- **Activation Constraint:** The points activated in the original pattern must also be activated in the resulting pattern.
- **Current balance constraint:** This constraint ensures that the sum of the currents delivered to all electrodes over a complete stimulation cycle is zero. This ensures that there is no net charge accumulation across the electrode array as a whole. However, this does not prevent individual electrodes from becoming polarized within a single cycle. To prevent charge accumulation on individual electrodes, it is necessary to manage the currents in the time domain, ensuring that any positive current applied to an electrode is followed by an equivalent negative current within the same cycle. Since our work focuses on static simulations, we do not address this dynamic current balance.
- **Maximum Injected Current Constraint:** This limits the current that can be injected into each electrode, set to a maximum of 5 mA per electrode based on safety restrictions outlined in [2].

Therefore, a general formulation of the problem would be Eq. (4.1).

$$
\begin{aligned}
\text{minimize} \quad & VSE(Pres, Pobj) \\
\text{subject to} \quad & \sum_{i=1}^{n} I_i \ = 0 \quad \text{(Current Balance Constraint)} \\
& I_i \leq 5\,\text{mA} \quad \forall i \quad \text{(Maximum Injected Current Constraint)} \\
& \forall\,\text{pos} \in \text{positions\_toactivate},\ T_{\text{res}}[pos] \geq \text{threshold} \quad \text{(Activation Constraint)}
\end{aligned}
\tag{4.1}
$$

VSE represents the visual error function which has as arguments the objective pattern P_{obj}, and the achieved activation pattern P_{res} and n is the total number of electrodes. Additionally, $Tres$ represents the transfer matrix, which will be introduced later.

To fully define the objective function and the optimization process, several key concepts need to be introduced. The following sections will outline the flow of the problem, starting from data collection, which serves as input to the algorithm, and proceeding to the selection and description of the optimization algorithm. Fig. 4.1 provides an overview of the problem flow, which will be progressively detailed in the next sections.

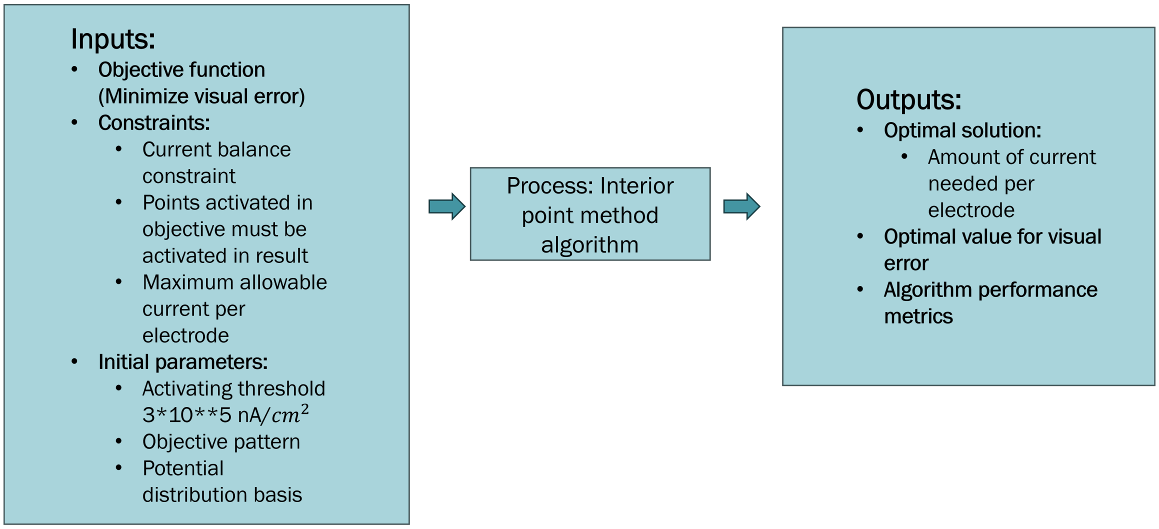

Inputs:
• Objective function (Minimize visual error)
• Constraints:
• Current balance constraint
• Points activated in objective must be activated in result
• Maximum allowable current per electrode
• Initial parameters:
• Activating threshold 3*10**5 nA/cm^2
• Objective pattern
• Potential distribution basis
Process: Interior point method algorithm
Outputs:
• Optimal solution:
• Amount of current needed per electrode
• Optimal value for visual error
• Algorithm performance metrics

Figure 4.1: Problem formulation flow chart

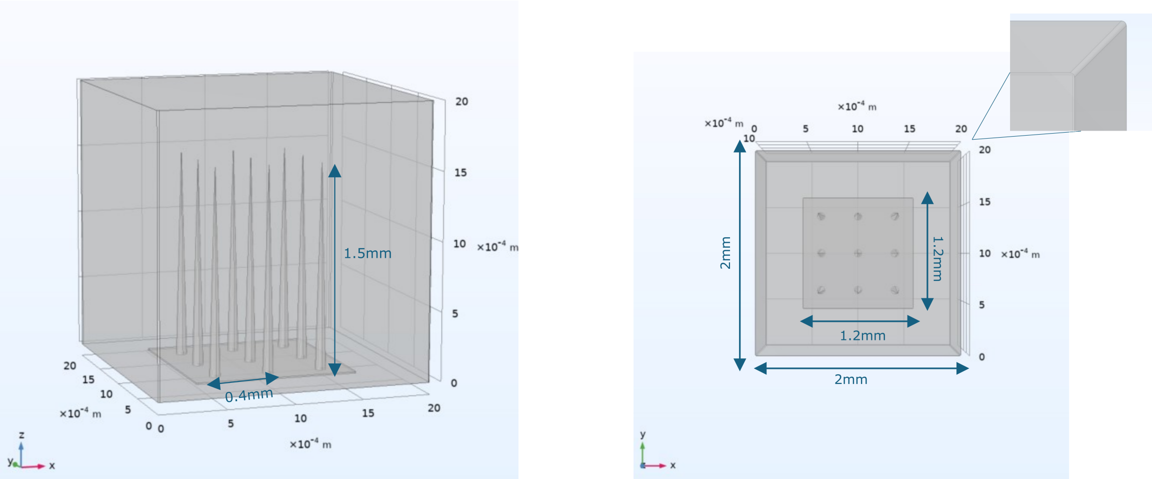
4.3 Data collection

As described in Section 2.3.1, MEAs are an accurate way to deliver precise microstimulation in the visual cortex. As an input to our algorithm, we need a potential field distribution basis, the electric potential fields generated by each electrode in the MEA when a unit current is injected into them.

4.3.1 Model description

To calculate the individual contribution of each electrode, the MEA was modelled using FEM in the COMSOL Multiphysics software [75]. In COMSOL, the Poisson equation Eq. (3.4) is solved for each element. This equation serves as a quasi-static approximation of the Maxwell equations, as detailed in Section 3.2.1.1. This assumption is valid only if the considered signals are less than 100MHz [76, 77, 7], in those cases the propagation medium, the biological tissue, can be considered as memory-less, the signal at any moment depends only on the input at that moment, not on any previous inputs. This feature makes the use of a stationary model an appropriate choice, as the system's behaviour is stable and predictable over time based on present inputs alone. Therefore, when setting up the simulation study, a stationary 3D analysis was selected using the AC/DC interface which is based on Maxwell´s equation and specifically designed to simulate the behaviour of electrical and magnetic fields [75]. In the first step, the geometry of the model is defined, see Fig. 4.2. This study is based on a version of the UTAH array [78] that has been FDA-approved for human use since 2004. This is the Neuroport Array model PN 4383 designed by Blackrock Neurotech [79]. For computational efficiency reasons (faster processing times, reduced memory usage), instead of the original 10x10 grid, the model designed in this project consists of a 3x3 array of 1.5 mm long microelectrodes which are separated from each other by an electrode pitch of 400 μm. Each electrode had a cone-like shape, with a base radius of 40 μm. The dimensions of our simplified MEA are 1.2mm x 1.2mm. The MEA was embedded in a cube that simulates the brain, of dimensions 2000 μm x 2000 μm x 2000 μm. Furthermore, it is important to mention that all sharp corners were smoothed using fillets of 10 μm, to avoid singularities in the numerical solutions. Several studies have highlighted the problems of having sharp corners in a computational mesh such as generating singularities in the numerical solution and causing abrupt changes in gradients [80, 81, 82].

In the next step, material properties are assigned to each component of the model. For that, the material properties of each element of the MEA were specified as displayed in Fig. 4.3. The electrodes are defined to have an exposed tip length of 50 μm, where each tip is coated with Platinum (Pt), which has an electric conductivity of 8.9e6 [S/m] [83] (see Fig. 4.3a). The rest of the electrode and the base of the MEA were given the properties of Polymide, with a conductivity of 1e-12 [S/m] [83] (see Fig. 4.3b), an electrical insulating material with properties similar to the ones used in Blackrocks Microsystems model [79]. The cube in which the MEA is embedded is considered to be made of grey matter, which has a conductivity of 0.587 [S/m]



(a) 3D Geometry of the model

(b) 3D Geometry of the model top view

Figure 4.2: 3D Geometry of the model with the defined measures

[84] (see Fig. 4.3c). After setting up the material properties, the following boundary conditions (BC) were defined:

- **Current conservation:** This BC was applied in all boundaries of the model, to ensure that the total current that enters the system equals the total current that leaves it, which is established by Kirchoff´s Current Law [85]:

$$\nabla \cdot \mathbf{J} = Q_{j,v} \quad \text{on all highlighted structures in Fig. 4.4a.} \tag{4.2}$$

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e \quad \text{on all highlighted structures in Fig. 4.4a} \tag{4.3}$$

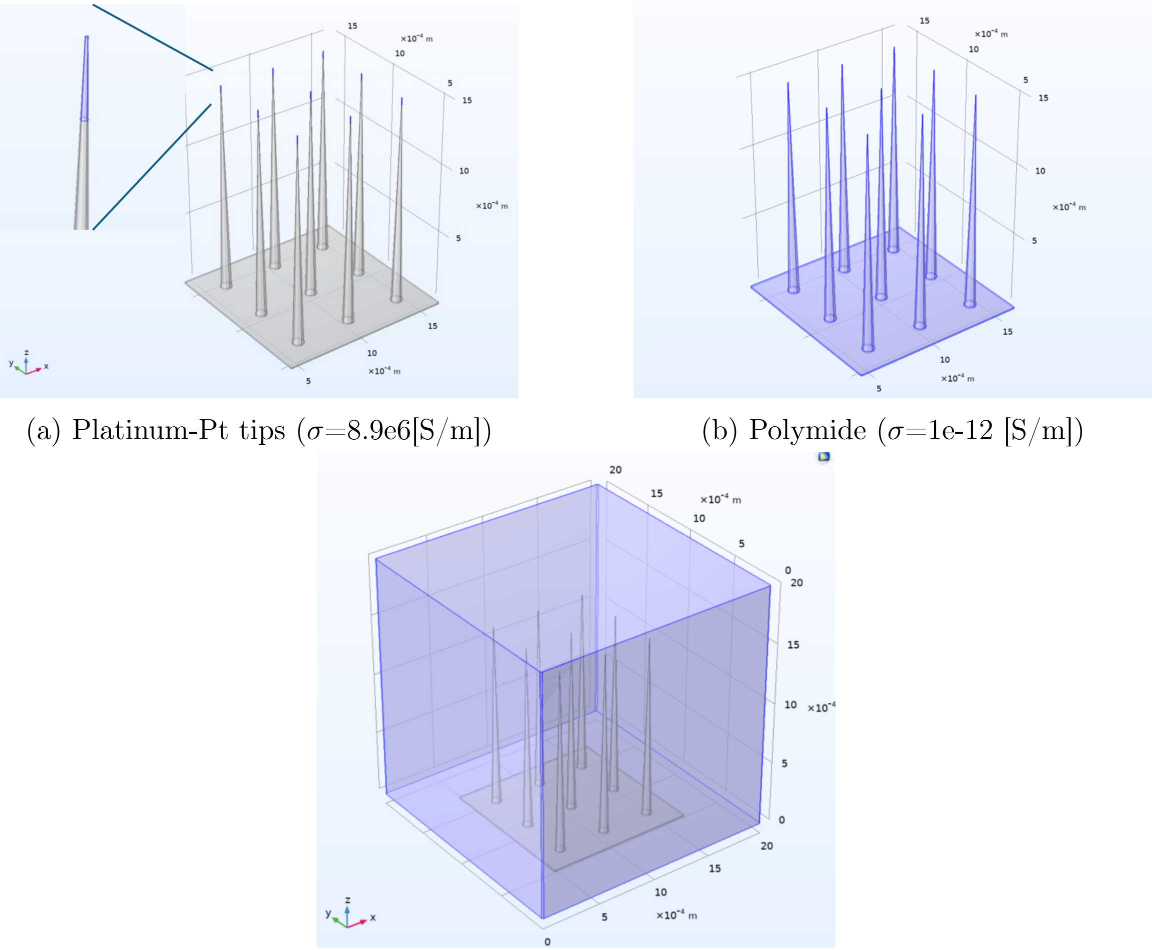
$$\mathbf{E} = -\nabla V \quad \text{on all highlighted structures in Fig. 4.4a} \tag{4.4}$$

  Where **J** is the current density vector, $Q_{j,v}$ is the volumetric charge density, σ is the electrical conductivity of the material, **E** is the electric field vector and V is the electric potential.

- **Electric Insulation:** This BC is also known as the homogeneous Neumann boundary condition, and it is used to simulate every insulating element of the model properly:

$$\mathbf{n} \cdot \mathbf{J} = 0 \quad \text{on all highlighted structures in Fig. 4.4a.} \tag{4.5}$$

  Where **n** is a normal vector to the insulating boundary and **J** is the current



(a) Platinum-Pt tips (σ=8.9e6[S/m])

(b) Polymide (σ=1e-12 [S/m])

(c) Grey matter (σ=0.587 [S/m])

Figure 4.3: Material properties of the model, the highlighted element in each figure represents the section of the model to which each specific material is assigned.

density vector.

- **Dirichlet Boundary Conditions:** With this BC the value of the potential field is set at every boundary. The ground in this model was defined in the outer boundary of the box-like tissue, setting there:

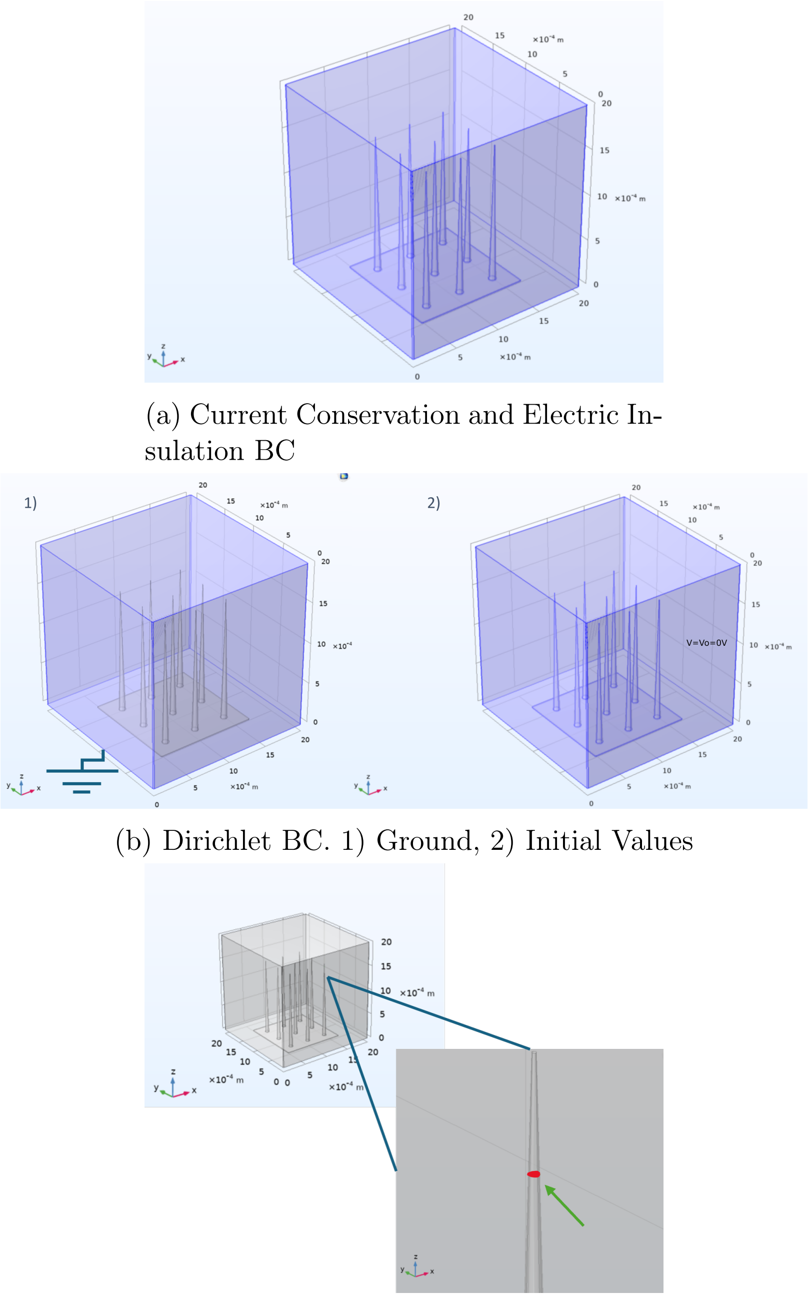
$$V = 0 \quad \text{on all highlighted structures in Fig. 4.4b} \tag{4.6}$$

This condition ensures that the outer boundary acts as a reference potential (ground). On the other hand, in the rest of the boundaries V was set to V_0 which was originally initialized as 0V Fig. 4.4b. This BC sets the starting conditions for the simulation. While V_0 starts at 0 V, it allows flexibility for adjustments if different potential values are required for other simulations or scenarios.

- **Current terminal:** This BC condition is applied in any surface in which the current is going to be injected. In the project, this BC will be changed for every simulation into the electrode we want to activate at each time:

$$\int_{\Omega} \mathbf{J} \cdot n \, \mathrm{d}S = I_0 \quad \text{on all highlighted structures in Fig. 4.4c} \tag{4.7}$$

To numerically solve the problem, COMSOL discretizes the domain into elements. For the discretization of the electric potential, quadratic Lagrange elements were selected. These elements approximate the solution using piecewise second-degree polynomial functions [59]. This choice is crucial as it offers superior accuracy and smoothness compared to linear elements. Quadratic elements are particularly effective in capturing sharp gradients and maintaining solution continuity across the domain [59, 86]. Finally, to ensure the accuracy of the MEA simulation, a tetrahedral mesh with "extremely fine" resolution was defined, as depicted in Fig. 4.5. The mesh had a maximum element size of $4 * 10^{-}5$ m and a minimum element size of $4 * 10^{-}7$ m. In static simulations, using a fine mesh is important for accurately resolving spatial variations in the electric field intensity around the microelectrodes. It is also important to accurately model the interface between the microelectrodes and the cortex, as explained in Section 2.3.2. Nine different simulations with this model were conducted, each activating a different electrode. To extract data from the 3D model, a cut plane was created at 1 µm above the tips of the electrodes (see Fig. 4.6), capturing the electric potential field distribution in that 2D plane. To mitigate numerical errors originating from the discretization inherent in COMSOL's Finite Element Method (FEM), a smoothing operation was applied during data extraction. For each electrode, the potential values were saved in matrices named T1 through T9. These electric potential results were then used as input for the optimization algorithm, which will be discussed further in this chapter.



(a) Current Conservation and Electric Insulation BC

(b) Dirichlet BC. 1) Ground, 2) Initial Values

(c) Current terminal

Figure 4.4: Boundary conditions, the highlighted element in each figure represents the section of the model to which each specific BC is assigned.

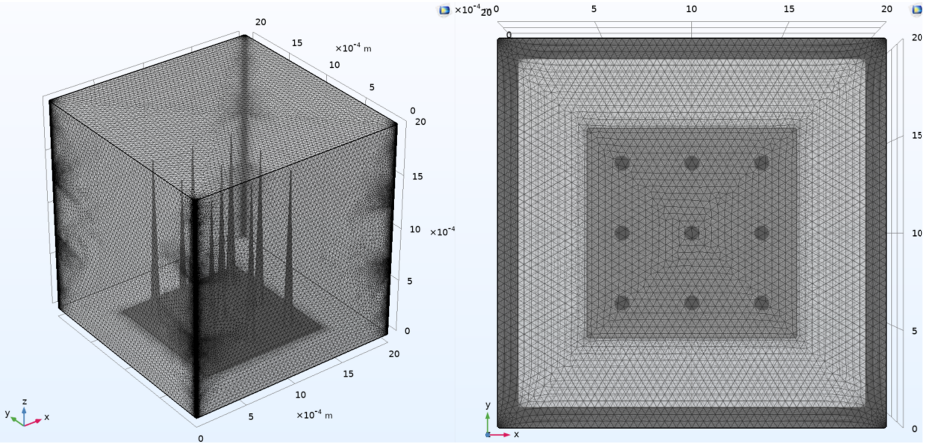


Figure 4.5: Mesh defined for the simulations

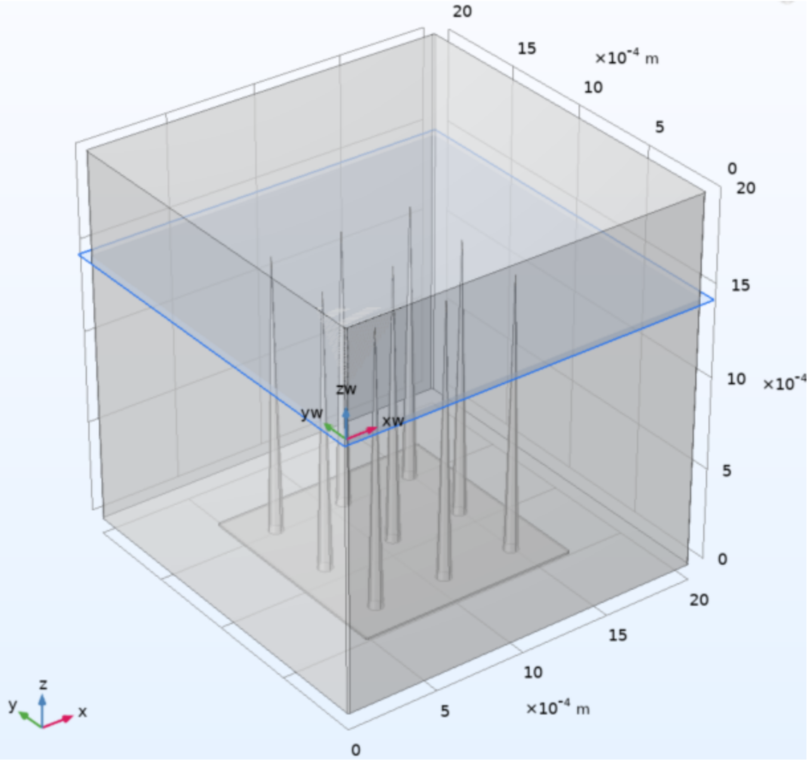


Figure 4.6: Cut plane at 1 µm over the microelectrodes tips.

4.3.2 Model validation

To ensure simulation accuracy and reliability, several validation steps were performed. First, the overall injected current was evaluated by doing a surface integration of the normal current on the terminal in each simulation to confirm that a 1 A current was being accurately injected. The normal component of the current obtained from this operation was 0.9998 A, indicating a 0.2 mA error. This deviation may be due to the limitations of the surface integration method in COMSOL, which is known to have some inaccuracies.

Next, as discussed in Section 3.2.1, the system is linear due to its governance by the Poisson equation. Therefore, superposition can be used. This principle states that the net response caused by two or more sources is the sum of the responses that would have been caused by each stimulus individually. This principle is defined by two properties [87]:

$$\mathcal{F}(V_1 + V_2) = \mathcal{F}(V_1) + \mathcal{F}(V_2) \quad \text{Additivity property} \tag{4.8}$$

$$\mathcal{F}(aV) = a\mathcal{F}(V) \quad \text{Homogeneity property} \tag{4.9}$$

To validate this principle, several test cases were performed. For example, the potential field generated by the first and third electrodes was calculated individually in COMSOL, and the electric potential values from these simulations were saved into two separate matrices. These matrices were then summed and compared with a separate simulation where both the first and third electrodes were activated simultaneously. The electric potential values from this combined activation were also saved in matrix format. The results showed that the values from the summed matrix and the direct simulation were the same, thus confirming the validity of the superposition principle.

4.3.3 Analytical input: Green´s Functions

FEM simulations usually generate numerical errors, mainly arising from the discretization step of the FEA [88]. To accurately validate the optimization algorithm developed in this chapter, it was crucial to ensure that the inputs were free of numerical errors. Therefore, an analytical setup was defined. The Poisson equation can be approximated by Green's functions in free space, especially when the point of interest is far from the current source. Although this is not the primary goal of our problem, it establishes a solid foundation for evaluating the algorithm's performance. According to [89], the solution of the Poisson equation in a finite volume with either Neumann or Dirichlet boundary conditions can be obtained using Green's functions. They introduced Eq. (4.10) as the Green function in free space, which gives the potential response at point r due to a unit point charge placed at $\mathbf{r}'$.

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} \tag{4.10}$$

In electrostatics the potential field Φ at a point due to a charge distribution ρ can be expressed by Eq. (4.11) [89].

$$\Phi(\mathbf{r}) = \int G(\mathbf{r}, \mathbf{r}')\rho(\mathbf{r}')\, d^3\mathbf{r}' \tag{4.11}$$

Where $G(\mathbf{r}, \mathbf{r}')$ is the Green's function in free space Eq. (4.10) and ρ $(\mathbf{r}')$ is the charge distribution.

Based on this information a code was implemented in Python (see Appendix Appendix A) to obtain the analytical solution to our problem, the function used to calculate the potential at each point was :

$$\Phi(\mathbf{r}) = \frac{I}{4\pi|\mathbf{r} - \mathbf{r}'| * \sigma} \tag{4.12}$$

Where σ is the conductivity of the medium and I is the current in Amperes injected in each electrode.

Also, in this Python implementation, the electrodes were positioned like they were for the COMSOL simulations. The coordinates of the electrodes as depicted in Fig. 4.7 were used in the analytical solution Appendix A. In this case, data was extracted from a cutplane located 100 micrometres above the current sources. This distance was chosen to ensure the validity of Green's function implementation, which is only accurate at points far from the source.

This method serves as a first step to validate the algorithm without the discretiza-

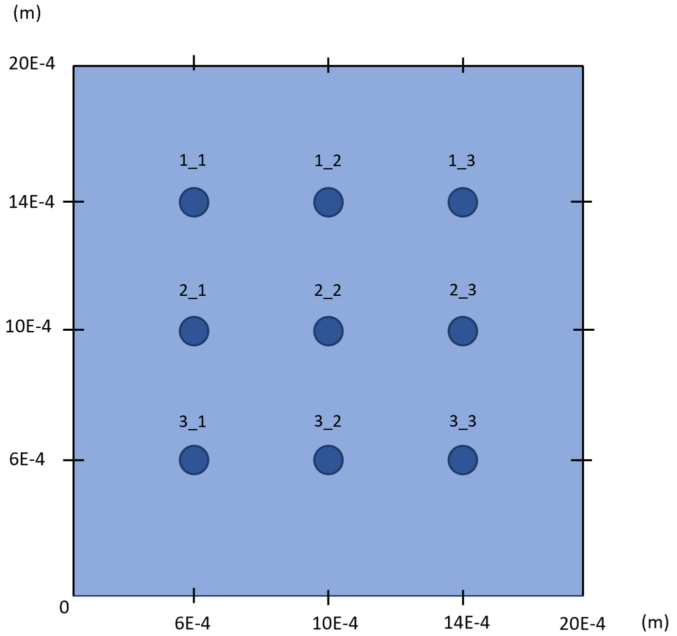


Figure 4.7: Electrode location visualization. Where the region of study is 2000 um.

tion error characteristic of a non-analytical solution. While round-off error, inherent in any computational solution, remains present, it is considerably smaller than the discretization error introduced by the Finite Element Method (FEM). Nevertheless, it is important to note that this data cannot be used to make conclusions about our problem. For scenarios involving complex geometries, such as cone electrodes, and studying interactions at close distances, the simplified Green's function approach may not be sufficient. The local effects of the electrode geometry and the strong interactions near the tips require a more detailed numerical solution. Therefore, FEM remains necessary for these cases to capture the accurate potential distribution and validate the optimization algorithm effectively.

4.4 Objective function

As discussed at the beginning of this chapter, the goal is to minimize the visual error. To calculate the visual error two patterns are needed:

- **Objective pattern (P_{obj}):** This is the pattern of phosphenes that are desired to be activated.

- **Resultant pattern (P_{res}):** This is the activated pattern generated by the currents injected into each microelectrode. These currents are the output of the algorithm at each iteration.

P_{obj} and P_{res} are binary matrices where 1 represents an activated point, and 0 represents an unactivated point.

The user provides P_{obj} as a binary matrix input to the algorithm. Several steps are necessary to achieve P_{res}.

The reference matrices T1 through T9 presented in the previous section Section 4.3 are employed, representing the electric potential fields generated by each electrode when a unitary current has been applied to them. These matrices were previously computed using the Finite Element Method (FEM) in COMSOL Multiphysics by solving the Poisson equation Eq. (3.4) and capturing the potential distribution in a 2D plane.

The activating function (AF) is utilized as a measure for determining neuron activation. It can be calculated from these reference matrices due to its linearity as described in Section 2.3.4. Derived from cable theory, the AF is directional, meaning its effect depends on the neuron's orientation relative to the electric field generated by the stimulation. To account for the unknown orientation of each neuron, the AF is computed as the average response in two orthogonal directions, x and y. This averaging provides a realistic estimate of the overall probability of neuron activation.

The algorithm employs nine decision variables (a1-a9) to control the current injected into each electrode. The current density transfer matrices for the original transfer matrices T_i where i ranges from 1 to 9 are calculated using the following equation:

$$T_i = AF(T_i) = \frac{d}{4\rho}\left[(T_{i-1,j} - 2T_{i,j} + T_{i+1,j}) + (T_{i,j-1} - 2T_{i,j} + T_{i,j+1})\right] \cdot (10^6) \cdot \frac{1}{h^2} \tag{4.13}$$

where $AF(T_i)$ represents the activating function for T_i, d refers to the axon diameter, which is usually given in micrometres (µm) and takes values from 1 to 10 µm, ρ is the resistivity of the axoplasm, given in Ωcm and h is the discretization parameter that defines the length of the axonal compartments, the resolution of our grid in this case.

This function is implemented in the Python code by using the central difference formula as described in [55]. Listing B.1 shows its implementation in Python.

Next, the current density transfer matrix T_{res} is obtained by summing the individual contributions of each electrode:

$$T_{\text{res}} = \sum_{i=1}^{9} a_i T_i \quad \text{where} \quad a_i \text{ are the decision variables} \tag{4.14}$$

T_{res} is then compared to the threshold of $3{*}10^5$ nA$/cm^2$ which has been reported in the literature as the activating threshold value to trigger action potentials in pyramidal neurons [55]. Based on this comparison, a new matrix, P_{res}, is constructed, where a value of 1 is assigned if the element in T_{res} exceeds the threshold, and 0 otherwise.

With both P_{obj} and P_{res} determined, the visual error is calculated using the **visual_error(matrixObj, matrixRes)** function, defined based on the formulation in [2], (see Listing B.8). This function takes as input the two activated patterns P_{obj} and P_{res}, the resulting transfer matrix (T_{res}), and the activating threshold.

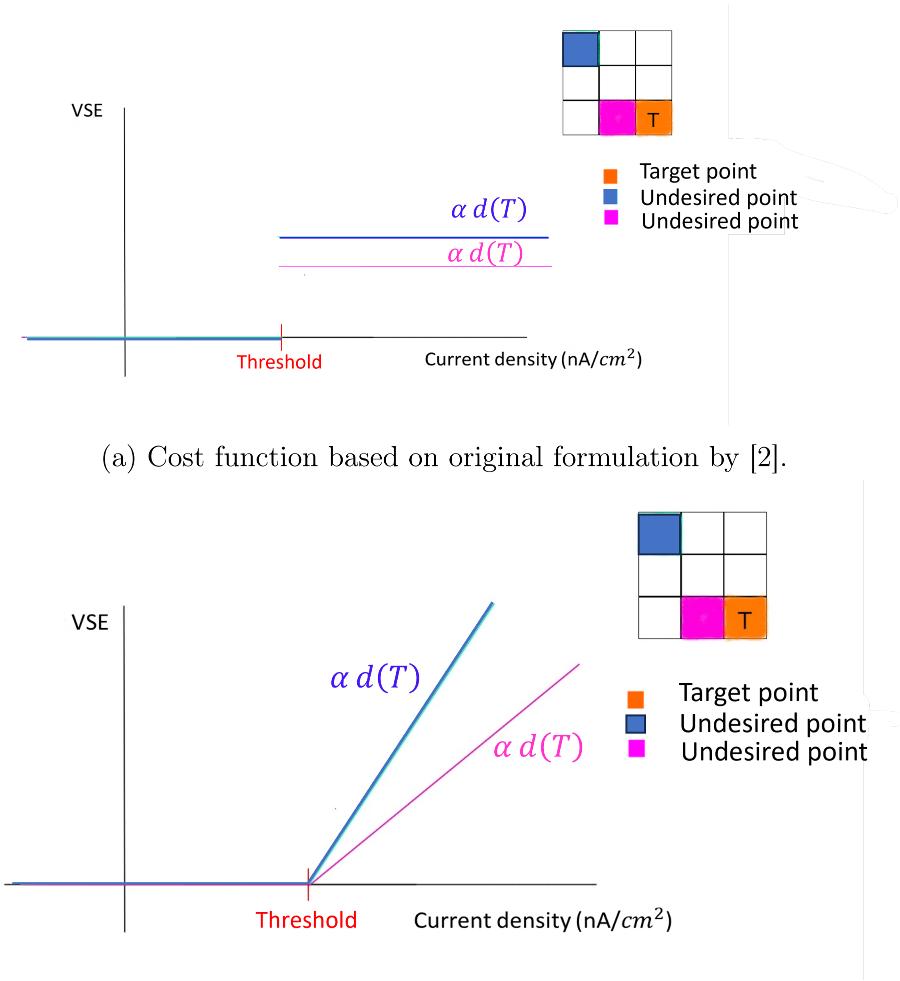
The visual error introduced in [2] is extended here to:

$$\Delta_{\text{j}} = \min\left(\Delta_{\text{ji}}\right) * \left(\text{T}_{\text{res}}[\text{j}] - \text{threshold}\right), \text{i} = 1...\text{N}_{\text{on}} \ and \ j = 1 \ldots N_{UDE} \tag{4.15}$$

$$\Delta_{tot} = \sum_{j=1}^{N_{UDE}} \Delta_j \tag{4.16}$$

The introduction of the term (T_{res}[j]-threshold) in the original formulation Eq. (3.18) ensures that the visual error formulation becomes continuous, which is beneficial for the optimization algorithm. This term measures the deviation of the current density at Tres[element] from the threshold, thereby increasing the error impact as the density deviates further from the threshold. Consequently, the objective function is continuous since it comprises the sum of continuous functions (i.e., the visual error at each pixel).

To illustrate this, Fig. 4.8 provides a comparison between the original and new formulations of the Visual Error (VSE) cost function. The original formulation often causes the objective function to be discrete, meaning the VSE would only change when decision variables varied significantly. In contrast, the new approach ensures the VSE changes in every iteration, resulting in a smoother objective function. Fig. 4.8b illustrates the impact of the distance between the target and the undesired activated point on the VSE. Specifically, when both undesired activated pixels exceed the threshold by the same amount, the pixel closer to the target (represented in pink) has a smaller impact on the VSE compared to the pixel farther from the target (represented in dark blue).



(a) Cost function based on original formulation by [2].

(b) VSE cost function based on the new formulation.

Figure 4.8: VSE cost functions against the current density introduced at certain pixels, pink and dark blue. The orange pixel is the original target, and the dark blue and pink are the undesirably activated pixels.

Additionally, several other functions were created to be used within this VSE function:

- **count_ones(matrixObj):** This function takes P_{obj} as input and counts the total number of locations that should be activated and their corresponding positions (see Appendix Listing B.3).

- **function_positions(matrixObj):**This function takes P_{obj} as input and returns an array of positions for every location in the matrix (see Appendix Listing B.5).
- **undesired_activations(matrixObj,matrixRes):** This function returns the number of undesired activated points and their locations (see Appendix Listing B.4).
- **pixel_size(Nloc,positions):** This function gets as input the total number of brain locations that are to be controlled (Nloc) and the coordinates and returns the pixel size (see Appendix Listing B.7). The calculations follow Eq. (3.17). And also calls **dneigh_function(j, Nloc, position)** that calculates the nearest neighbour phosphene distance Eq. (3.16) defined by [2] (see Appendix Listing B.6).

Consequently, the objective function for the problem has been defined (see Listing B.9). The objective function takes the decision variables a_1-a_9, which are updated by the optimization algorithm in each iteration, and the target activation pattern P_{obj} as inputs. Within the function, the previously mentioned steps are performed to calculate P_{res} in each iteration. Subsequently, the visual error between the newly calculated P_{res} and the input P_{obj} is determined. The output of the objective function is the visual error, which the algorithm aims to minimize iteratively.

This function weights each pixel proportional to the distance between the undesired activated point and the desired activation point. The decision variables in this optimization problem (the currents injected into each microelectrode) are continuous, reflecting the continuous nature of the currents. Given the continuity of the objective function, the following section will discuss the type of algorithm most suitable for this optimization problem.

4.5 Optimization algorithm

The continuous and non-linear characteristics of this objective function, coupled with the presence of constraints (including inequality constraints) and bounded decision variables (Maximum Injected Current Constraint), make interior point methods an ideal choice for solving the defined optimization problem. Additionally, interior point methods are well-suited for problems with a relatively large number of decision variables, such as this problem which involves 9 decision variables [90].

The algorithm is implemented using the trust-region option in the scipy.optimize package in Python [91]. This algorithm is a trust-region method for constrained optimization, capable of switching between two implementations based on the problem definition. Given the inclusion of inequality constraints in this problem, the algorithm employs trust-region interior point methods.

The 'trust-constr' algorithm handles inequality constraints by introducing slack variables and solving a sequence of equality-constrained barrier problems with progressively smaller barrier parameters [91]. The interior point method used in this algorithm is described in [92].

4.5.0.1 Interior Point Methods

These methods work by iteratively improving a feasible solution from within the interior of the feasible region, rather than from the boundary, which is a common approach in other methods like the simplex method [90] (see Fig. 4.9). The algorithm used by 'trust-constr', as described by [92], utilizes a barrier method to decompose the original problem into a sequence of manageable sub-problems. Each sub-problem incorporates a barrier term that penalizes violations of the inequality constraints.

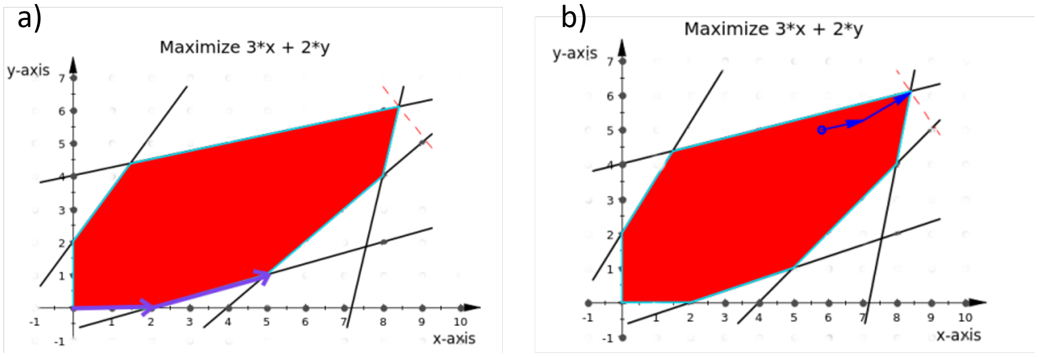


Figure 4.9: Visualization of an example problem at its 3rd iteration: a) Solved by Simplex method, b) Solved by interior point methods. Image source [90].

Consider the original problem formulated as:

$$\begin{aligned} &\min f(x) \\ &\text{subject to} \quad h(x) = 0, \\ &g(x) \leq 0. \end{aligned} \tag{4.17}$$

The corresponding barrier subproblem is formulated as:

$$\min f(x) + \mu \sum_i \ln(s_i) \quad \text{subject to} \quad g(x) + s = 0, \quad s > 0 \tag{4.18}$$

Where μ is the barrier parameter and the s are the slack variables that must be positive. Each of these subproblems is solved by sequential quadratic programming (SQP) and trust region techniques [92]. This leads to solving the subproblem to find an approximate solution (x^+, s^+) that satisfies:

$$E(x^+, s^+; \mu) \leq \epsilon_\mu \tag{4.19}$$

where E measures the optimality conditions of the barrier subproblem. After solving each barrier subproblem, the algorithm updates the barrier parameter to progressively reduce the influence of the barrier term. This is done by:

$$\mu \leftarrow \theta\mu \quad \text{where} \quad \theta \in (0, 1) \tag{4.20}$$

The algorithm iterates until the optimality condition $E(x, s; 0) \leq \epsilon_{TOL}$ is satisfied [92]. The implementation is carried out in Python using the scipy.optimize.minimize function.

Chapter 5

Results

5.1 Generated Electric Potential Fields

As detailed in Section 4.3, the electric potential fields generated by each electrode in the Microelectrode Array (MEA) are used as inputs for the algorithm. These fields were calculated using two different methods:

1. **Using the Green´s Functions**

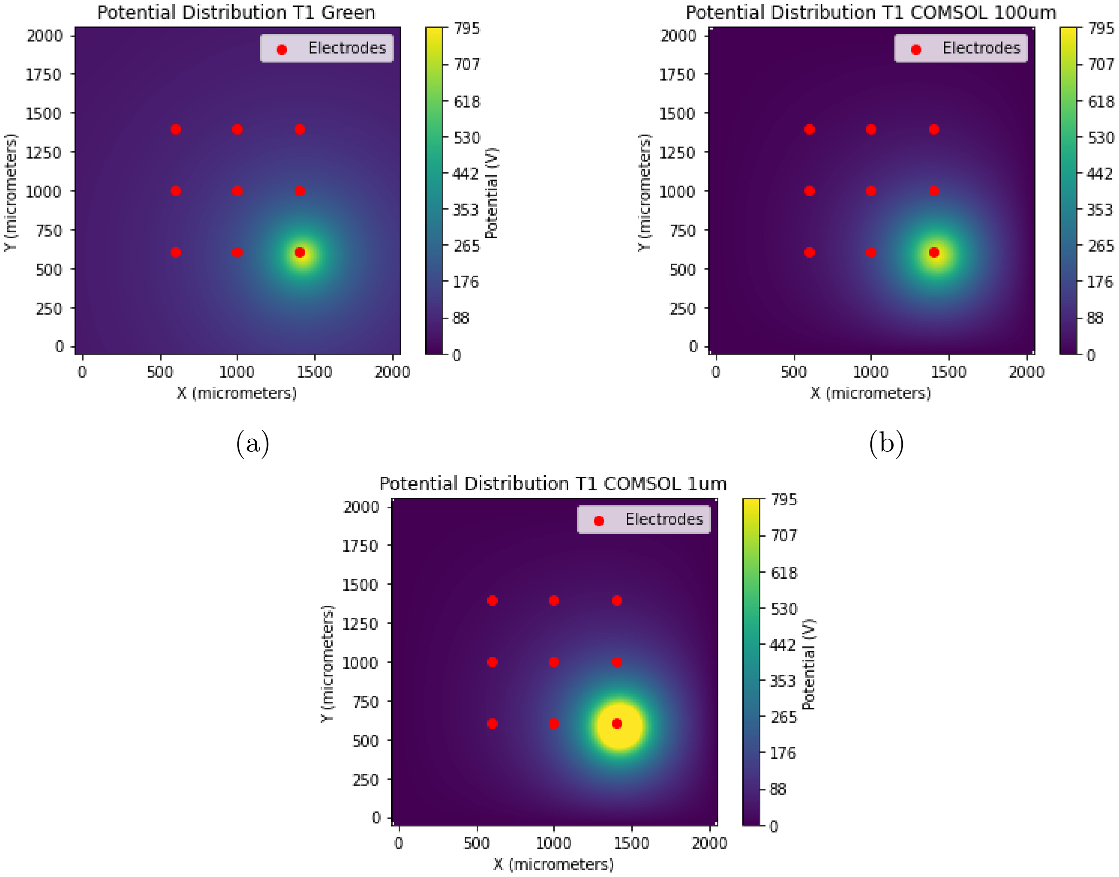
   The electric potential fields were computed using Green´s Functions as described in Section 4.3.3. A 2D cut plane 100µm from the current source was extracted from the analytical solution. This method provides an analytical basis to validate the algorithm performance (see Fig. 5.1a).

2. **Using FEM in COMSOL Multiphysics**

   The electric potential fields were also calculated using FEM in COMSOL Multiphysics software. This approach involves solving the Poisson equation for each element. Two versions of these fields were extracted from the COMSOL 3D model. A 2D cut plane at 100µm from the tip of the electrodes (Fig. 5.1b) and a 2D cut plane at 1µm from the tip of the electrodes (Fig. 5.1c). The 100µm cut plane is used for comparing with the solutions obtained from the analytical input as mentioned in Section 4.3.3.

5.1.1 Visualization of the Potential Fields

The generated potential fields for each method are visualized in Fig. 5.1. The potential distributions generated by the other electrodes can be found in Appendix C, Appendix D and Appendix E. These colour plots depict the electric potential field resulting from a 1 A current injected into a microelectrode. It is crucial to highlight that this simulation uses a current value significantly higher than those typically employed in practical microstimulation applications.



(a)

(b)

(c)

Figure 5.1: Visualization of the potential fields for a unit current into electrode 1 of the MEA calculated using (a) the analytical approach with Green's function at a distance of 100 μm from the source, (b) FEM at a 2D cut plane 1 μm above the tip of the electrodes, and (c) FEM at a 2D cut plane 100 μm above the tip of the electrodes.

In real-world microstimulation scenarios, the currents used usually range from microamperes (μA) to milliamperes (mA). Consequently, the electric potential fields observed in actual applications are expected to be orders of magnitude smaller compared to the plots generated for a 1 A current. For example, with a current of 1 mA, the resulting potential field would be approximately 1000 times less intense than what is depicted for 1 A.

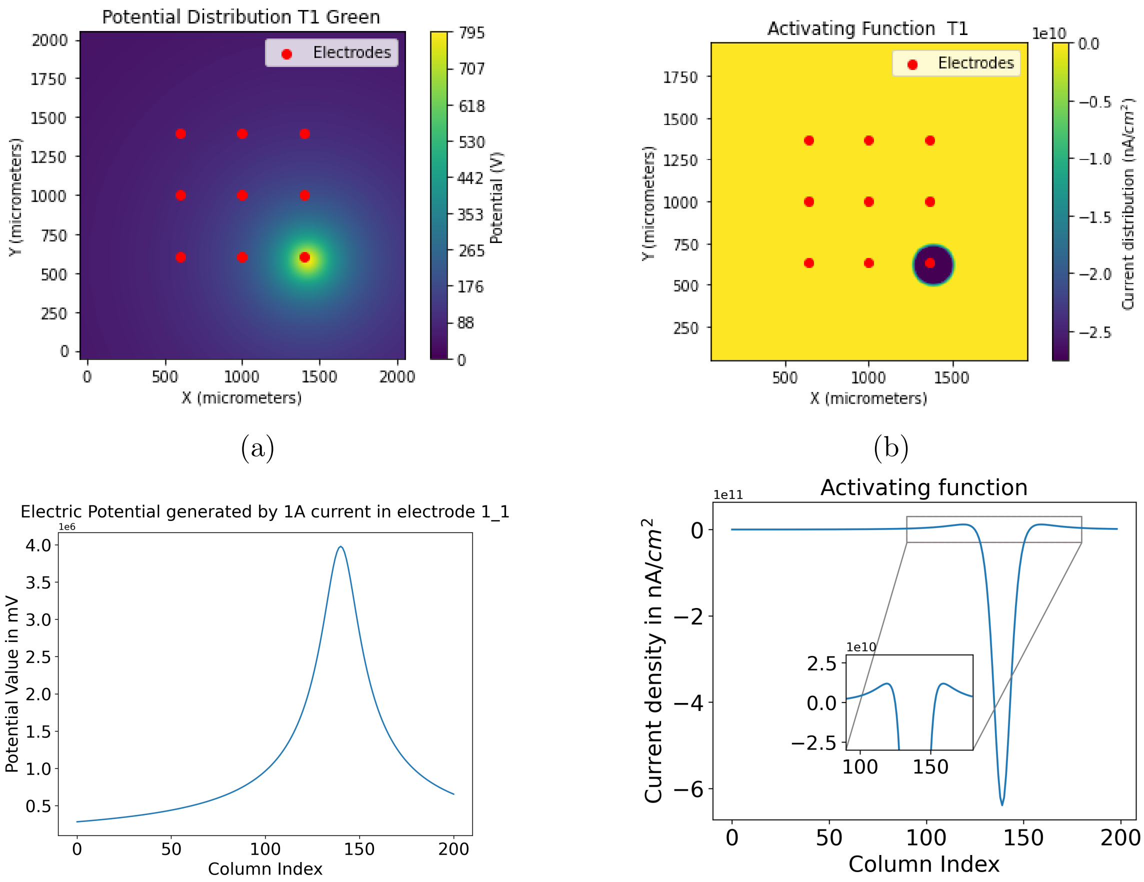
These figures show that the highest potential is located at the site of the current injection, with a gradual decrease in potential as the distance from the electrode increases. This behaviour is consistent across both methods of calculation. Additionally, the figures demonstrate that Green's Function provides a good approximation to the Poisson equation, particularly when the plane of study is far from the current source (100 μm away in this case). This is noticeable in the comparable potential distributions observed in the cut planes taken at this distance (see Fig. 5.1b and Fig. 5.1a). Furthermore, a significantly higher potential near the electrodes can be observed when studying the cut plane taken close to the electrode tips (see Fig. 5.1c). This indicates a more intense electric field near the electrodes, highlighting the differences in potential distribution at varying distances from the current source.

5.2 Activating Function Response:

As mentioned in Section 4.4, to determine neuron activation, the activating function was applied to all potential fields generated by each electrode in the MEA and the current density response was calculated.

5.2.1 Visualization of the AF response

The current density distribution (nA/cm^2) generated by electrode one is shown in Fig. 5.2. The activating function computes the second derivative of the potential in two dimensions. As a result, the new plot Fig. 5.2b presents negative values whereas the original plot showed positive values. Furthermore, in Fig. 5.2d the areas of depolarization (small positive sidelobes) and hyperpolarization (negative peak) distinct from the activating function can be distinguished.



(a)

(b)

(c)

(d)

Figure 5.2: Visualization of (a) the electric potential field for electrode 1, calculated using Green´s functions at a cut-plane 100 µm away from the source. (b) Its corresponding AF response. (c) 1D plot of extracellular potential represented in (a). (d) 1D plot of the activating function.

5.3 Algorithm Convergence study

The convergence of the algorithm was studied for several different target points to assess its robustness. The study involved executing the optimization problem 50 times for each target point, where the only thing that changed was the initial point for the decision values $a_1 - a_9$, which was randomly assigned in every run. This was used to explore the algorithm behaviour across various starting points. The results, depicted in Figs. 5.3 and 5.4, show that the algorithm converges to a point near the same location across multiple runs. However, noticeable deviations from this point

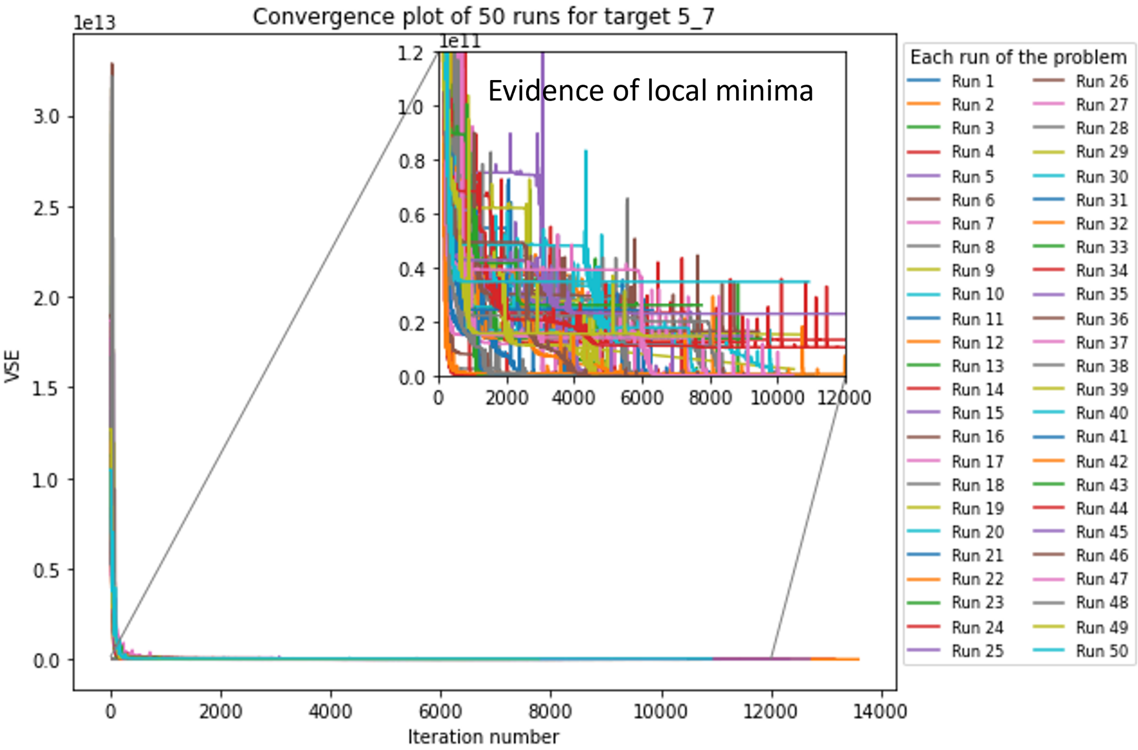


Figure 5.3: Visualization of the convergence behaviour of the algorithm over 50 runs for target point [5,7]. The main plot displays the overall convergence trend where most runs appear to converge to a similar point. However, as shown in the inset, a closer examination reveals that not all runs converge to the global minimum; some get trapped in local minima. The inset magnifies this region, illustrating the variations among runs that are not visible in the broader view.

are apparent when zoomed in. These deviations indicate that the algorithm may be converging to local minima rather than the global optimum.

The methodology used in this study to address this issue involved running the optimization problem 30 times for each target point and selecting the minimum value obtained as the optimal solution. This approach aims to mitigate the impact of local minima by increasing the likelihood of finding a global minimum.

5.4 Solution analysis

The resolution selected for all simulations in this study was 100 μm. This resolution was chosen to adequately capture the interaction of fields between the electrodes

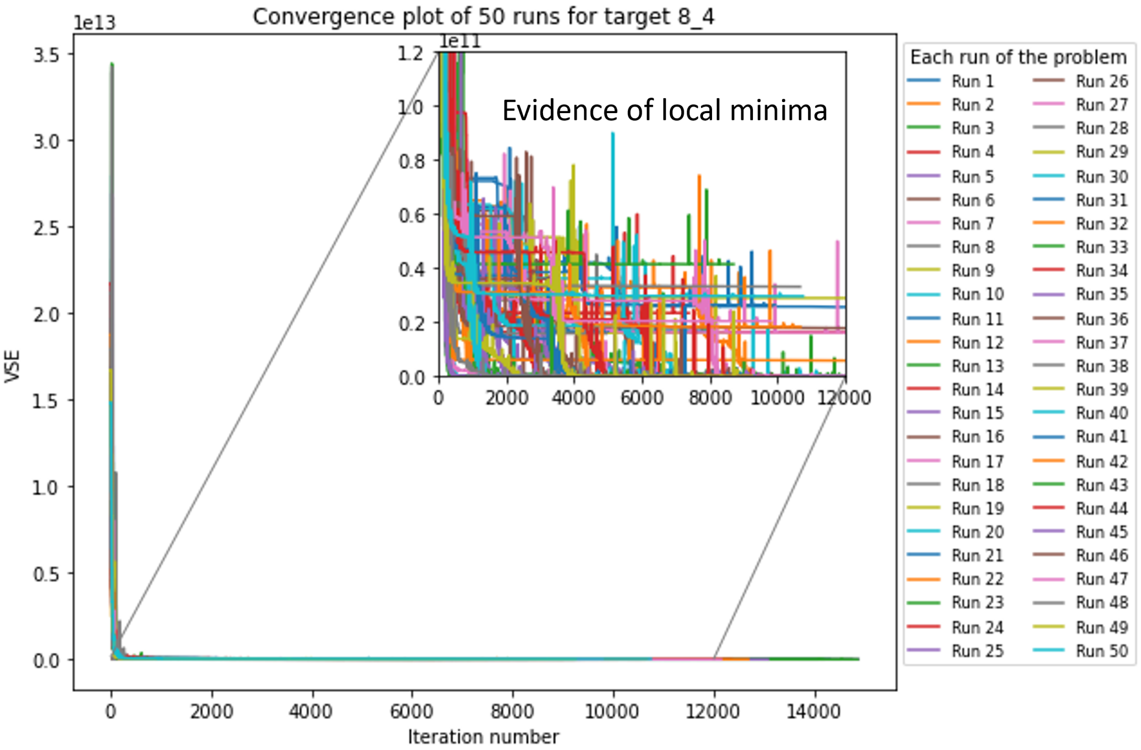
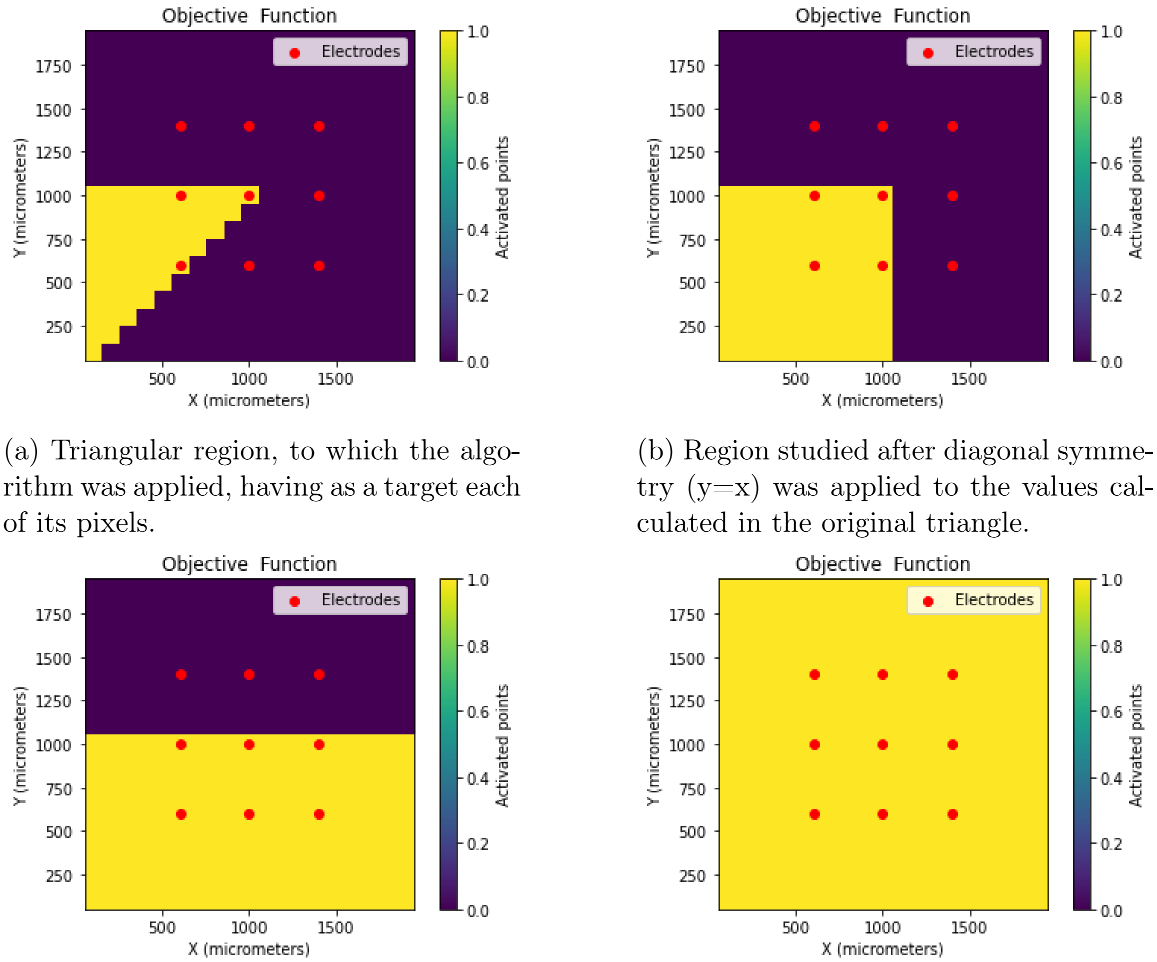


Figure 5.4: Visualization of the convergence behaviour of the algorithm over 50 runs for target point [8,4]. The main plot displays the overall convergence trend where most runs appear to converge to a similar point. However, as shown in the inset, a closer examination reveals that not all runs converge to the global minimum; some get trapped in local minima. The inset magnifies this region, illustrating the variations among runs that are not visible in the broader view.

while managing computational resources efficiently. The study region is initially represented by a 20x20 pixel matrix corresponding to a physical area of 2000µm by 2000 µm. However, after the activating function (AF), which uses the central difference formula to compute spatial derivatives, see Eq. (4.13), the matrix is reduced to 19x19 pixels. This reduction occurs because the central difference formula requires neighbouring pixels on both sides for accurate calculations. Since edge pixels lack sufficient neighbours, the formula cannot be applied at the boundaries, leading to the reduction of the matrix size to avoid errors and ensure accurate results.

In this study, symmetry is used to optimize the computation of optimal solutions across the study area. The algorithm is initially applied to a triangular region defined by the coordinates (0,0), (9,0), and (9,9) within the 19x19 pixel matrix, representing the study area (see Fig. 5.5a). The choice of this triangular region is due to its symmetrical properties.

The optimal solutions obtained for the pixels within this triangle were then extended



(a) Triangular region, to which the algorithm was applied, having as a target each of its pixels.

(b) Region studied after diagonal symmetry (y=x) was applied to the values calculated in the original triangle.

(c) Region studied after vertical symmetry was applied to the values calculated in Fig. 5.5b.

(d) Region studied after horizontal symmetry was applied to the values calculated in Fig. 5.5c.

Figure 5.5: Visualization of symmetry operations applied to original solutions from triangle

by symmetry to cover the entire 19x19 pixel matrix. Thereby optimizing computational efficiency. Three different symmetry operations were applied, first diagonal symmetry, by mirroring the values calculated in the original triangle over the line y=x (see Fig. 5.5b). Then vertical symmetry by further extending the matrix, mirroring the previously calculated values from the bottom left quadrant to the bottom right quadrant (see Fig. 5.5c). Finally, the values from the bottom half were mirrored to the top half, as depicted in Fig. 5.5d.

To assess the performance of the algorithm, 6 different study cases were performed.

1. **Study case 1:** using data extracted from Green Function approximation
2. **Study case 2:** using data extracted from COMSOL at a cut plane placed 100 μm away from the sources.
3. **Study case 3:** using data extracted from COMSOL at a cut plane placed 1 μm away from the sources.
4. **Study case 4:** adding minimization of injected current sources to the objective function.
5. **Study case 5:** multi target VSE minimization.
6. **Study case 6:** explore focality improvement by exponential objective function.

Furthermore, to ensure that the results from different study cases could be directly compared, random initial values for the initial guesses were systematically saved and reused throughout the study. This approach ensures that the results are comparable across different cases.

5.4.1 Study case 1: using data extracted from Green Function approximation

In this case, the input matrices to the algorithm were the ones obtained by using the Green Function approximation previously introduced in Section 4.3.3. For every target point, the algorithm gave as an output the amount of current needed to be injected in each pattern to achieve the minimum visual error. Fig. 5.6 shows an example of a resulting activated pattern given by the optimization output. Where the relative position between the target and the electrodes can clearly be observed. In this case, 6 pixels were wrongly activated when attempting to activate pixel [8,8]. In Appendix F all of the outputs given by the algorithm for the pixels contained in

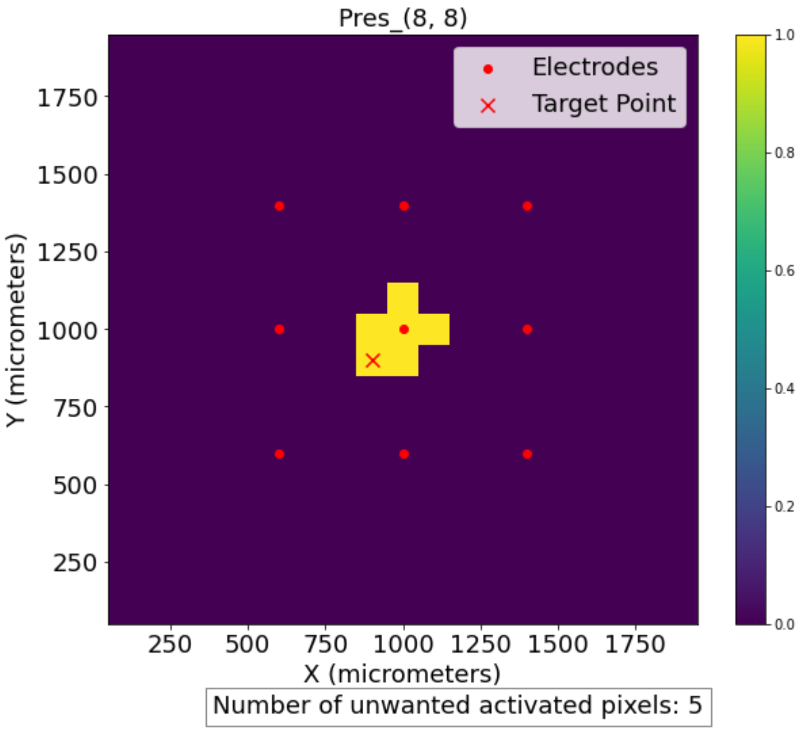
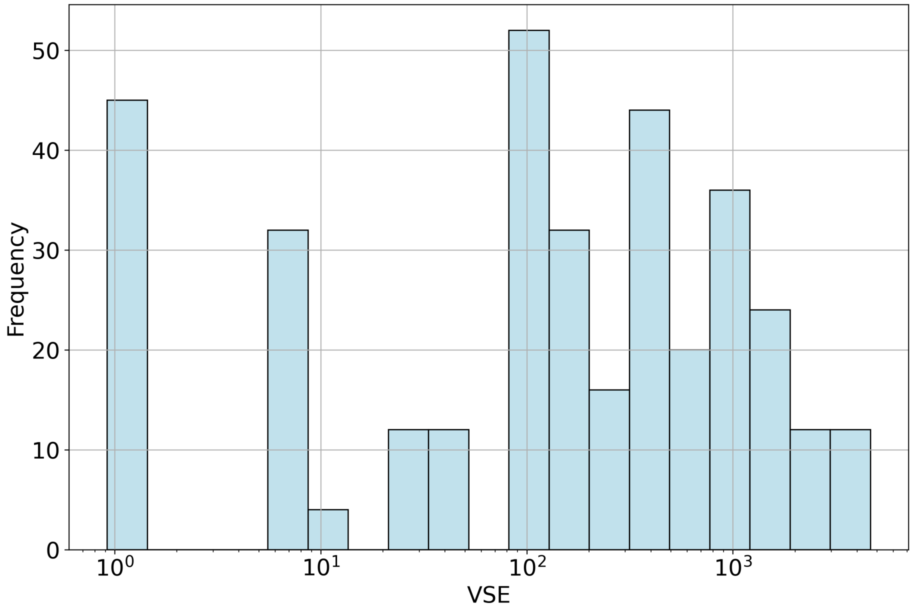


Figure 5.6: Visualization of resulting activation pattern when attempting to activate point (8,8). In the figure, the relative position between the target and the electrodes can be distinguished.

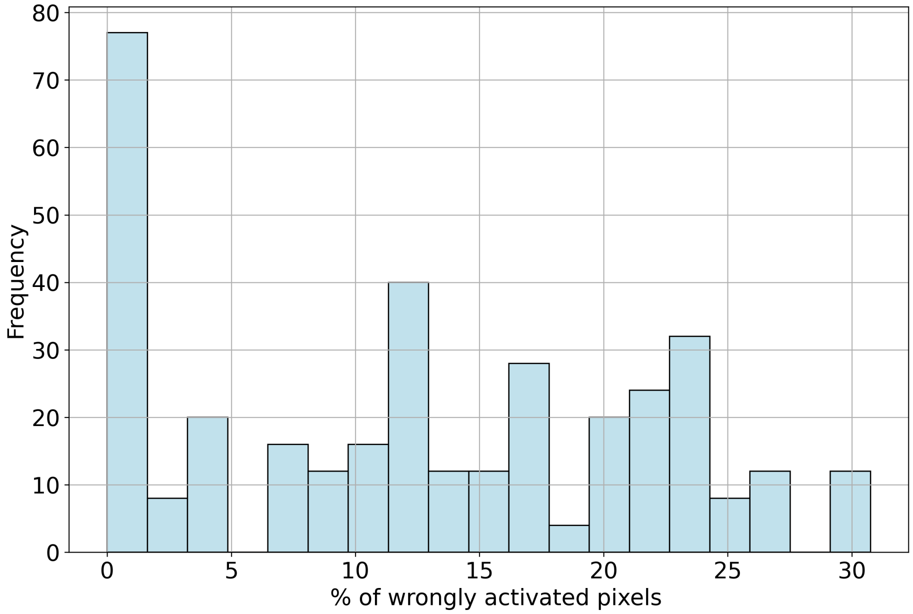
Fig. 5.5a can be observed.

For the overall target data set (Fig. 5.5d), 2.21% of the targets did not converge to a solution. For the remaining dataset, the average number of undesirably activated pixels was 46.35 ±32.25, with a minimum of 0 for target points corresponding to the electrodes and a maximum of 111 undesirably activated pixels. The mean percentage of wrongly activated pixels was 12.84 ±8.94 % for the overall dataset. The average execution time for each run was 16 min and 30 seconds ± 9 minutes and 27 seconds, and the average number of iterations required to achieve an optimal solution was 557.11 ± 312.13. The average VSE was 537.7e6± 791.7. All VSE values along the report were scaled down by 10^6 for better visualization.

Fig. 5.7a and Fig. 5.7b visualize the distribution of the VSE and the percentage of wrongly activated pixels across the dataset. As VSE is not inherently intuitive, the percentage of wrongly activated pixels was also calculated for clarity. This percentage was determined by dividing the number of undesirable activated pixels by the



(a) Green function dataset



(b) Green function dataset

Figure 5.7: Representation of how the (a) VSE and (b) the percentage of wrongly activated pixels distribute along the overall Green Functions dataset

total number of pixels (361) and multiplying by 100. Additionally, an analysis was conducted to differentiate the pixels based on their proximity to the electrodes, categorized as either close (within 300 µm) or far. This distinction is visualized in Fig. 5.8. The separated data is presented in Fig. 5.10a and Fig. 5.10b. Furthermore, the points within the electrode region were distinguished from those outside the electrode area, as defined in Fig. 5.9.

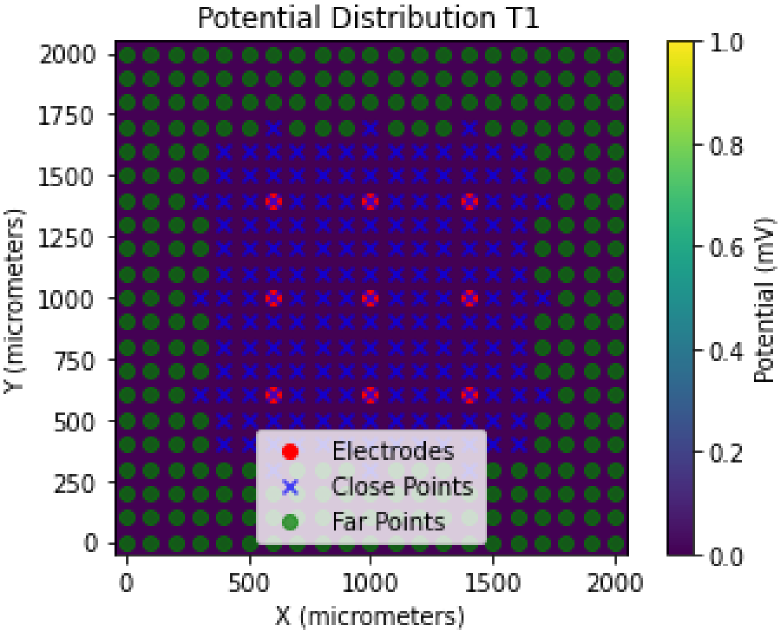


Figure 5.8: Distinction between the targets considered close to electrodes and the ones considered far.

This distinction is useful since if the MEA used in this setup were to be scaled up, the results observed within the electrode area could be expected to be extrapolated to the larger MEA.

In these images, it can be observed how the targeted pixels closer to the electrodes presented a smaller VSE than those that were farther away. The same can be said when looking at the percentage of wrongly activated pixel plots.

In a similar manner, Fig. 5.10c and Fig. 5.10d show that the target points that were within the electrode area generated a smaller percentage of wrongly activated pixels, than the targets that were outside this area.

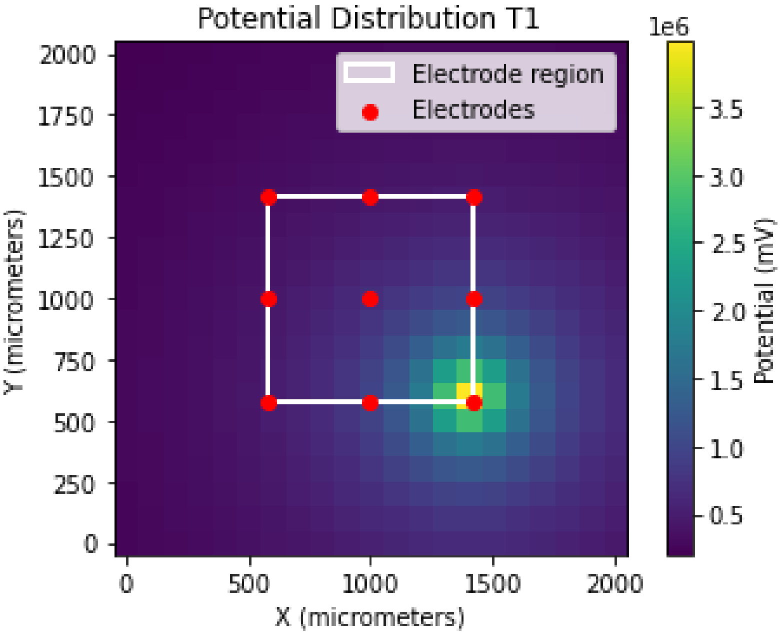
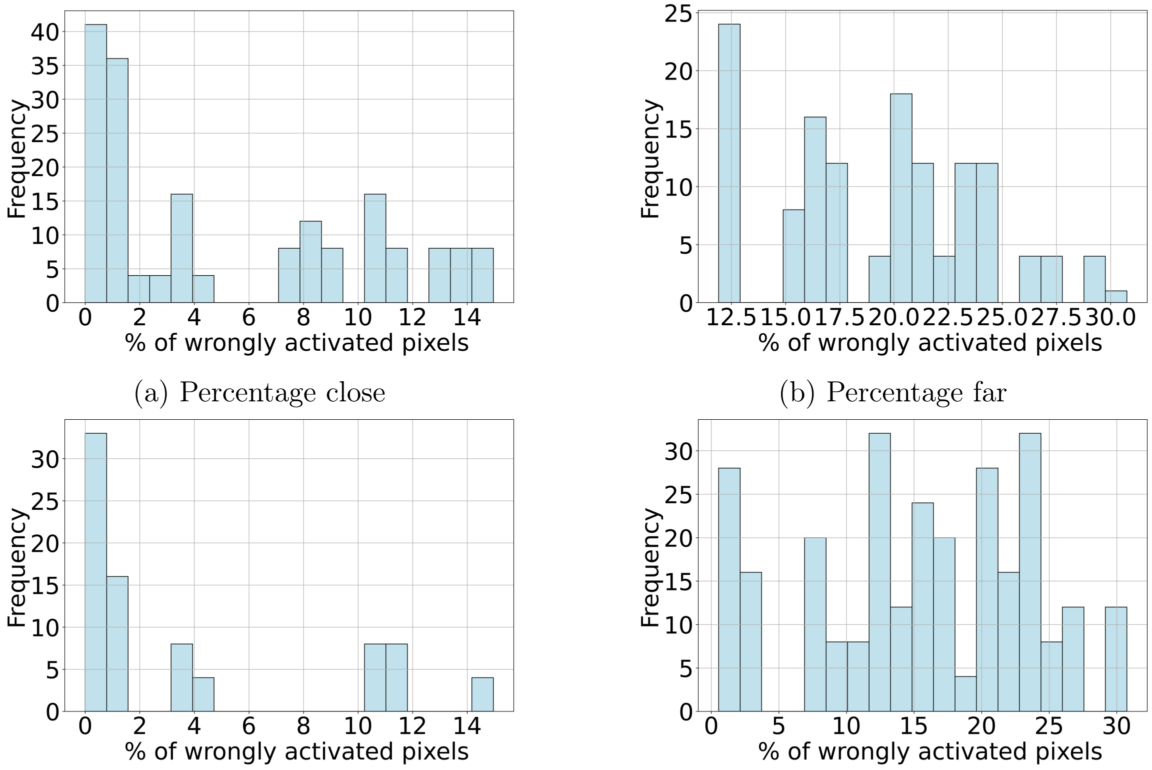


Figure 5.9: Distinction between the areas inside and outside the electrode region.

5.4.2 Study case 2: using data extracted from COMSOL at a cut plane placed 100 µm away from the sources

As introduced in Section 4.3.3, the Green Functions approximation only holds if the points of the study are placed far away from the sources. Therefore, to compare the results obtained from these approximations presented in Section 5.4.1 with the results obtained in COMSOL, the data was extracted at a cut plane placed 100 µm away from the current sources, the electrode tips. This data was used to obtain the input matrices to the algorithm. Furthermore,

Fig. 5.11 shows an example using COMSOL data as input for the same target point as shown in the previous section. Although the input data for the algorithm came from different sources, the activated pattern given as an output to the target point [8,8] looks the same in both cases, proving this way that the Green Functions are a good approximation for COMSOL solutions when the study points are far from the source.



(a) Percentage close

(b) Percentage far

(c) Percentage inside

(d) Percentage outside

Figure 5.10: Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close and (b) far from electrodes. And between points (c) inside the electrode area and outside of it. Green Functions dataset

All of the outputs given by the algorithm for the pixels contained in Fig. 5.5a can be observed in Appendix G. In this case, 12.8 % of the total targets did not converge to an optimal solution. For the rest of the results, the average number of undesirably activated pixels was 52.24 ± 34.04, with a minimum of 0 for target points corresponding to the electrodes and a maximum of 128 undesirably activated pixels for the overall dataset. The mean percentage of wrongly activated pixels was 14.47 ±9.43 %. The average execution time for each run was 21 min and 04 seconds ± 14 minutes and 11 seconds, and the average number of iterations required to achieve an optimal solution was 734.41 ± 512.67. The average scaled VSE was 820 ± 1965

Fig. 5.12a and Fig. 5.12b visualize the distribution of the VSE and the percentage of wrongly activated pixels across the dataset. As in Section 5.4.1 the percentage of wrongly activated pixels was also calculated for clarity over the VSE which is not that

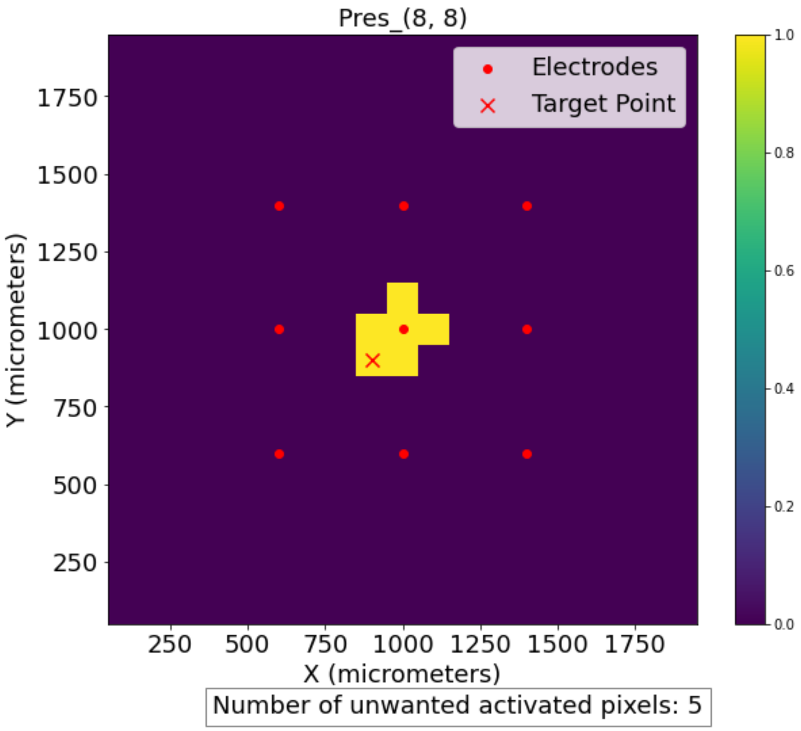
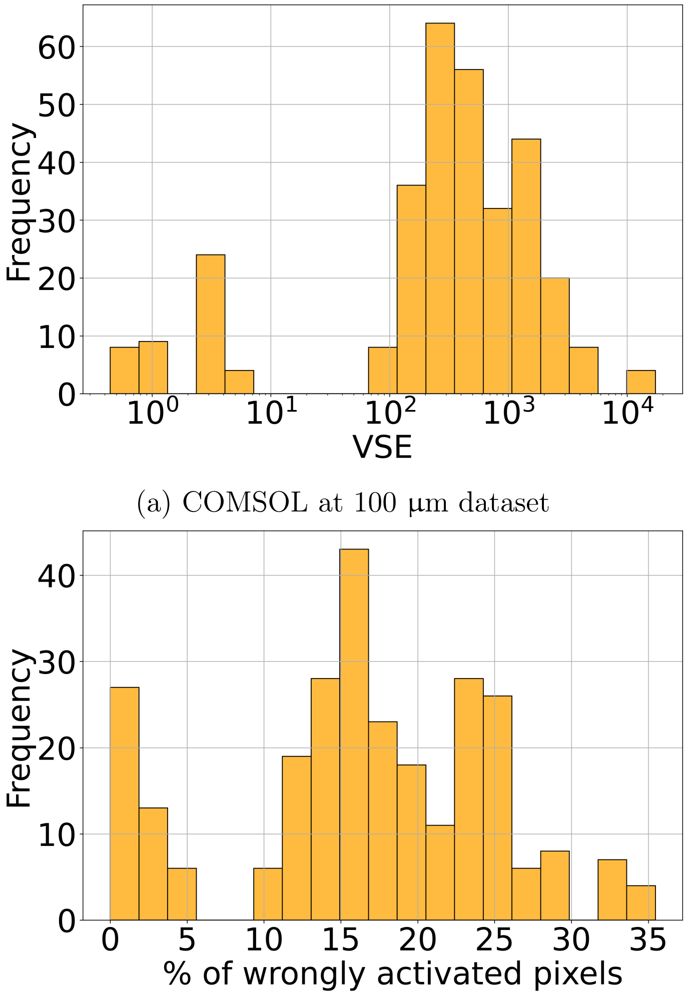


Figure 5.11: Visualization of resulting activation pattern when attempting to activate point (8,8) using COMSOL data as an input. In the figure, the relative position between the target and the electrodes can be distinguished.

intuitive. Additionally, an analysis was conducted to differentiate the pixels based on their proximity to the electrodes, categorized as either close (within 300 µm) or far. This data separation is visualized in Fig. 5.13a and Fig. 5.13b. In this case, the algorithm presented a poorer performance when compared with Section 5.4.1, characterized by a higher VSE rate, higher percentage of wrongly activated pixels, longer execution times and an increased number of iterations needed to reach an optimal solution. The observed trend that target pixels farther from the electrodes generate a higher VSE and therefore a bigger percentage of wrongly activated pixels remains consistent also in this case. By comparing the results from both cases, Section 5.4.1 and Section 5.4.2, it is observed that the results from the Green function approximation are similar to those from the Finite Element Method (FEM) across various metrics, such as the average number of undesirably activated pixels, the percentage of wrongly activated pixels, execution time, and the number of iterations. This similarity indicates that the Green function provides a sufficiently accurate approximation of



(a) COMSOL at 100 μm dataset

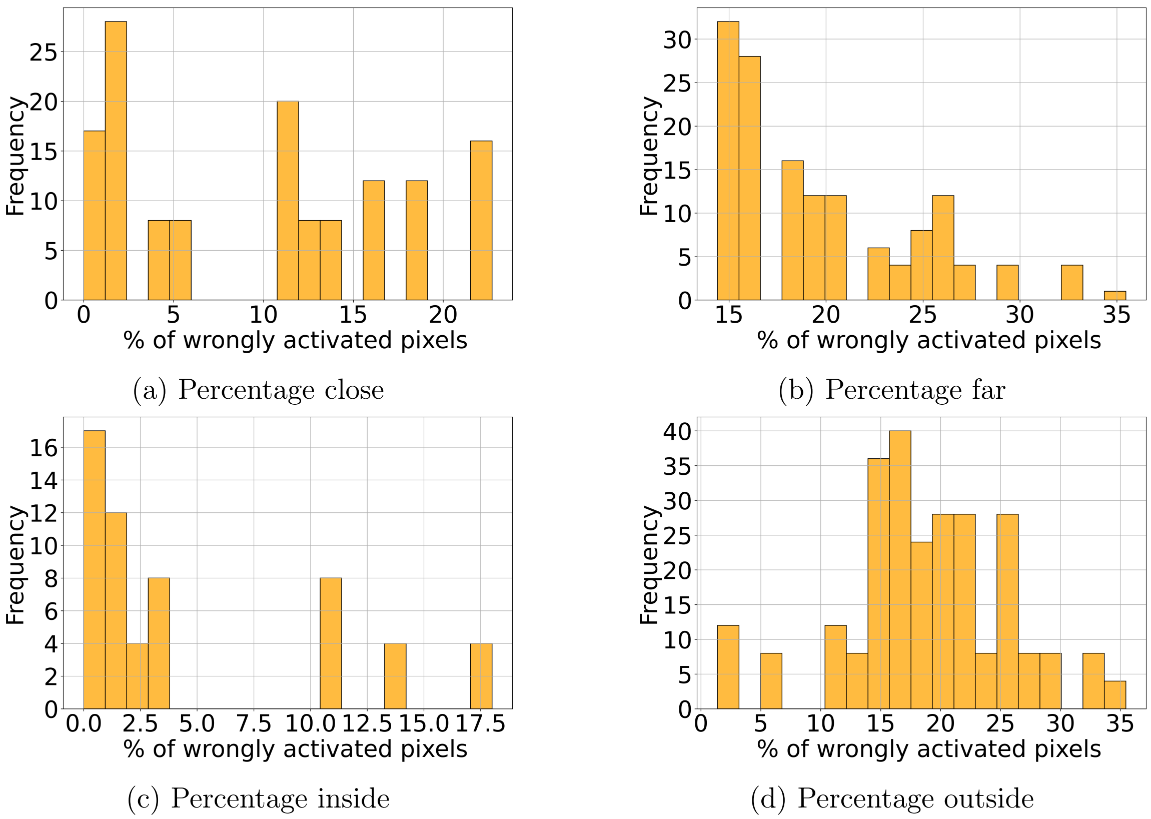
(b) COMSOL at 100 μm dataset

Figure 5.12: Representation of how the VSE and the percentage of wrongly activated pixels distribute along the overall dataset extracted from COMSOL at 100 μm cut plane .

the data extracted from COMSOL. This approximation allowed for a robust evaluation of the algorithm's performance without the numerical errors typically associated with the Finite Element Method (FEM).

5.4.3 Study case 3: using data extracted from COMSOL at a cut plane placed 1 μm away from the sources

In this project, the interactions of the electric field at close distance to the electrode tips were of particular interest. While previous cases provided valuable insights into



(a) Percentage close

(b) Percentage far

(c) Percentage inside

(d) Percentage outside

Figure 5.13: Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close, (b) far from electrodes, (c) inside and (d) outside electrode area for COMSOL dataset extracted at 100µm cut plane.

the algorithm's performance, an additional study was necessary to focus on the electric potential fields in closer proximity to the electrodes. In this study, data was extracted from a 1 µm cut plane adjacent to the electrode tips to create the input matrices for the algorithm. The corresponding example, consistent with those presented in earlier cases, where the target pixel is [8,8] is illustrated in Fig. 5.14. In this case, the resulting activating pattern differs from the one obtained in the previous cases. This is because the algorithm may have been stuck in a local minimum. A problem discussed in Section 5.3 and that we can expect for any point in the study. All of the outputs given by the algorithm for the pixels contained in Fig. 5.5a can be observed in Appendix H.

In this study case, 26.59% of the targets did not converge to a solution. For the targets that did converge, the average number of undesirably activated pixels was 56.93 ±

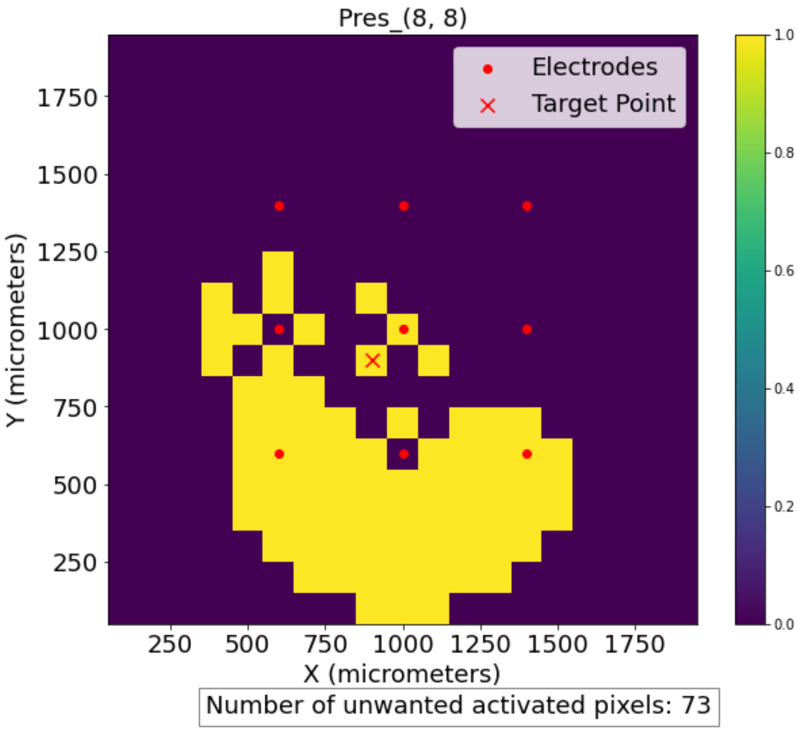
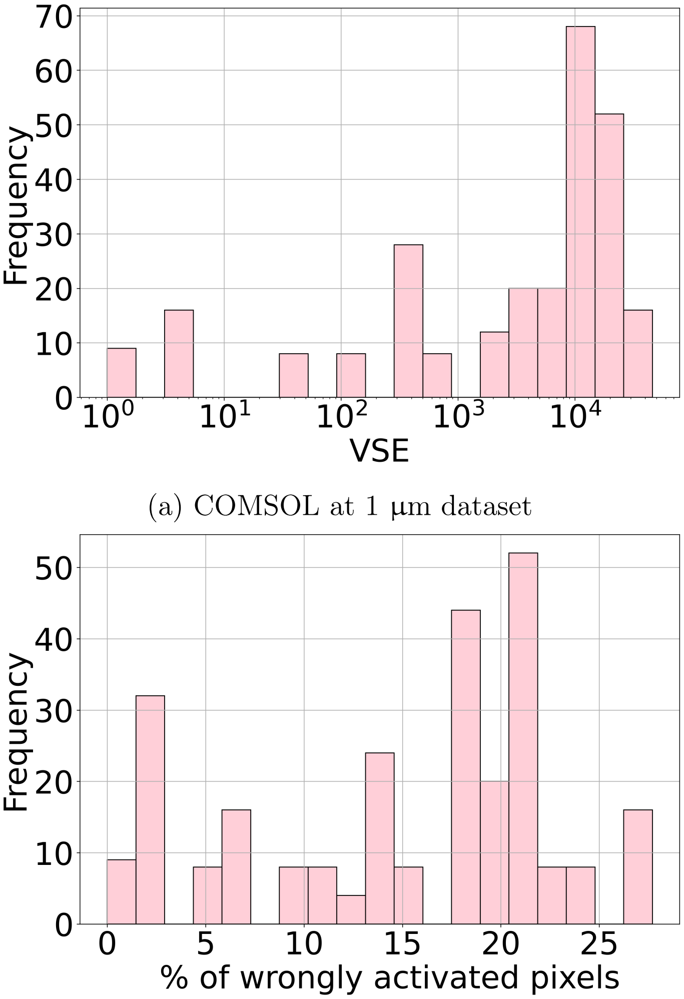


Figure 5.14: Visualization of resulting activation pattern when attempting to activate point (8,8) using COMSOL data at 1µm cut plane as an input. In the figure, the relative position between the target and the electrodes can be distinguished.

28.22, with a minimum of 0 for target points corresponding to the electrodes and a maximum of 117 undesirably activated pixels for the overall dataset. The mean percentage of wrongly activated pixels was 15.77 ±7.81 %. The average execution time for each run was 25 min and 16 seconds ± 14 minutes, and the average number of iterations required to achieve an optimal solution was 1042.828 ± 672.16. The average VSE was 10416 ± 10002.

Fig. 5.20a and Fig. 5.20b visualize the distribution of the VSE and the percentage of wrongly activated pixels across the dataset. Again, an analysis was conducted to differentiate the pixels based on their proximity to the electrodes, categorized as either close (within 300 µm) or far. And to differentiate between targets inside the electrode area and outside. These separations can be visualized in Fig. 5.16. The results in this study showed worse performance compared to the other two cases. There is an increase in the mean number of undesirably activated pixels, in the percentage of wrongly activated pixels, in execution time, and in a number of iterations



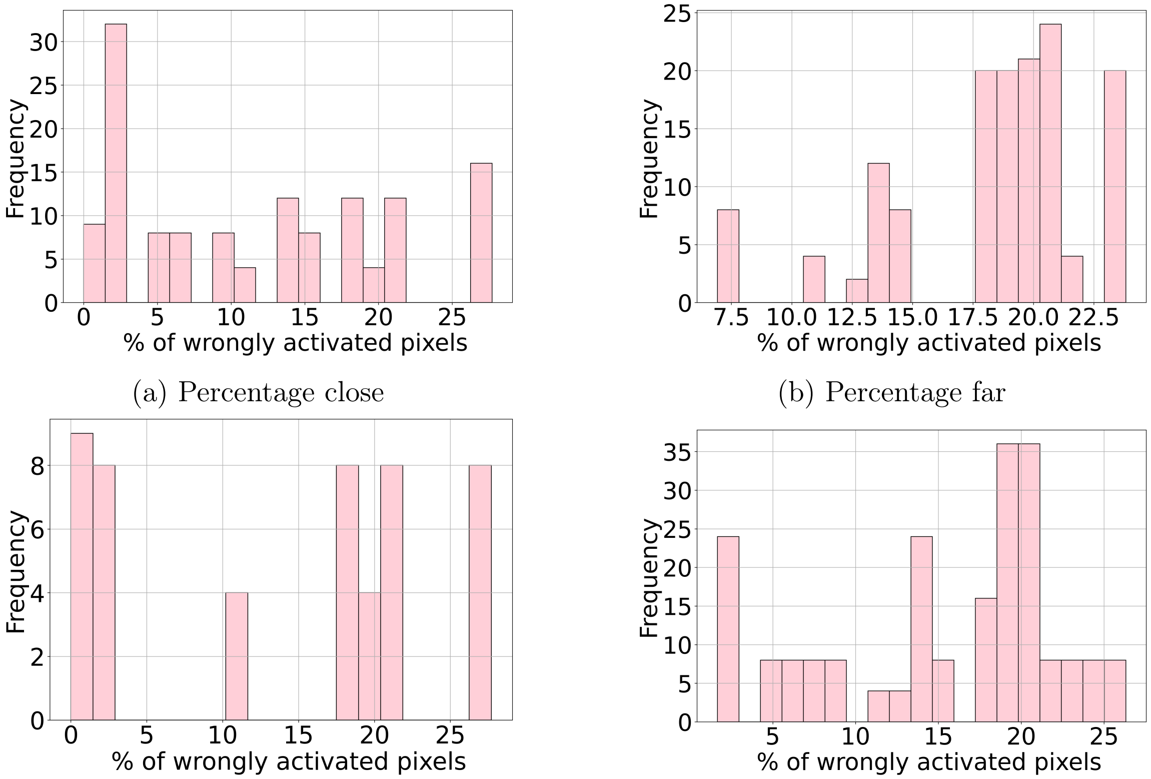
(a) COMSOL at 1 μm dataset

(b) COMSOL at 1μm dataset

Figure 5.15: Representation of how the VSE and the percentage of wrongly activated pixels distribute along the overall dataset of data extracted from COMSOL at a 1μm cut plane.

which suggests that using input data that was closer to the electrode tips made it more challenging for the algorithm to converge. Furthermore, there is an increase in the VSE that can be visualized when comparing the 3 study cases, as displayed in Table 5.1.

Consistent with the observation made in the two previous study cases Fig. 5.16 shows that the target points far from the electrode are the ones that achieved VSE with higher orders of magnitude.



(a) Percentage close

(b) Percentage far

(c) Percentage inside

(d) Percentage outside

Figure 5.16: Representation of how the percentage of wrongly activated pixels distribute along the overall dataset distinguishing between points (a) close, (b) far from electrodes, (c) inside electrode area and (d) outside electrode area, for the COMSOL dataset extracted at 1μm cut plane.

Table 5.1: Comparison of algorithm performance metrics across Study Cases 1, 2, and 3.

Study case	1	2	3
VSE scaled (avg±sd)	537.7e6± 791.7	820 ± 1965	10416 ± 10002
Execution Time (avg ± sd) (mm:ss)	16:30± 9:27	21:04±14:11	25:16±14:00
N iter (avg ±sd)	557.11 ± 312.13	734.41 ± 512.67	1042.828 ± 672.16
% of non convergence	2.21%	12.8 %	26.59%
Und Activated pixels (avg±sd)	46.35 ±32.25	52.24 ± 34.04	56.93 ± 28.22
% wrongly activated pixels (avg±sd)	12.84 ±8.94 %	14.47 ±9.43 %.	15.77 ±7.81 %.

5.4.4 Study case 4: minimization of injected current sources

Another point studied in this project was minimizing the currents injected into the electrodes, in addition to the original objective of minimizing visual error. This study aimed to investigate how limiting the number of electrodes activated at a time would impact both visual error and power efficiency Section 3.3. To address this question, the original objective function described in Eq. (4.1) was modified to:

$$\begin{aligned} \text{minimize} \quad & VSE(Pres, Pobj) + \quad currentFactor \\ \text{where} \quad & currentFactor = \sum_{i=1}^{9} a_i^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2 + a_6^2 + a_7^2 + a_8^2 + a_9^2 \end{aligned} \tag{5.1}$$

The currentFactor uses the square of the currents to account for both positive and negative values, avoiding the potential discontinuities that could arise from using absolute values in the objective function.

From prior study cases, it was observed that the scales of the currentFactor and the VSE differed significantly. The VSE values varied from as low as 1 to as high as 10^9, while the currentFactor ranged from 10^-7 to 10^-9. Due to these significant differences in magnitude, both components were normalized to ensure they contributed equally to the final objective function. Specifically, the VSE values were divided by their maximum observed value, 10^9, resulting in a normalized range of 0 to 1. The currentFactor was normalized using the formula:

$$\text{currentFactor} = \frac{\sum_{i=1}^{9} a_i^2 - 10^{-9}}{10^{-7} - 10^{-9}} \tag{5.2}$$

The aim of this study was not only to minimize visual errors but also to reduce the amount of current injected into the patient's brain for each target.

To assess the performance of the algorithm with this new objective function two analyses were performed, one using the data extracted from COMSOL at a cut plane 100μm from the electrode tips and one at a cut plane 1μm and its results were compared with the previously introduced cases Section 5.4.2 and Section 5.4.3. The following table compares the performance of the algorithm for both types of input data with

Table 5.2: Comparison of algorithm performance for the different study types with and without intensity minimization

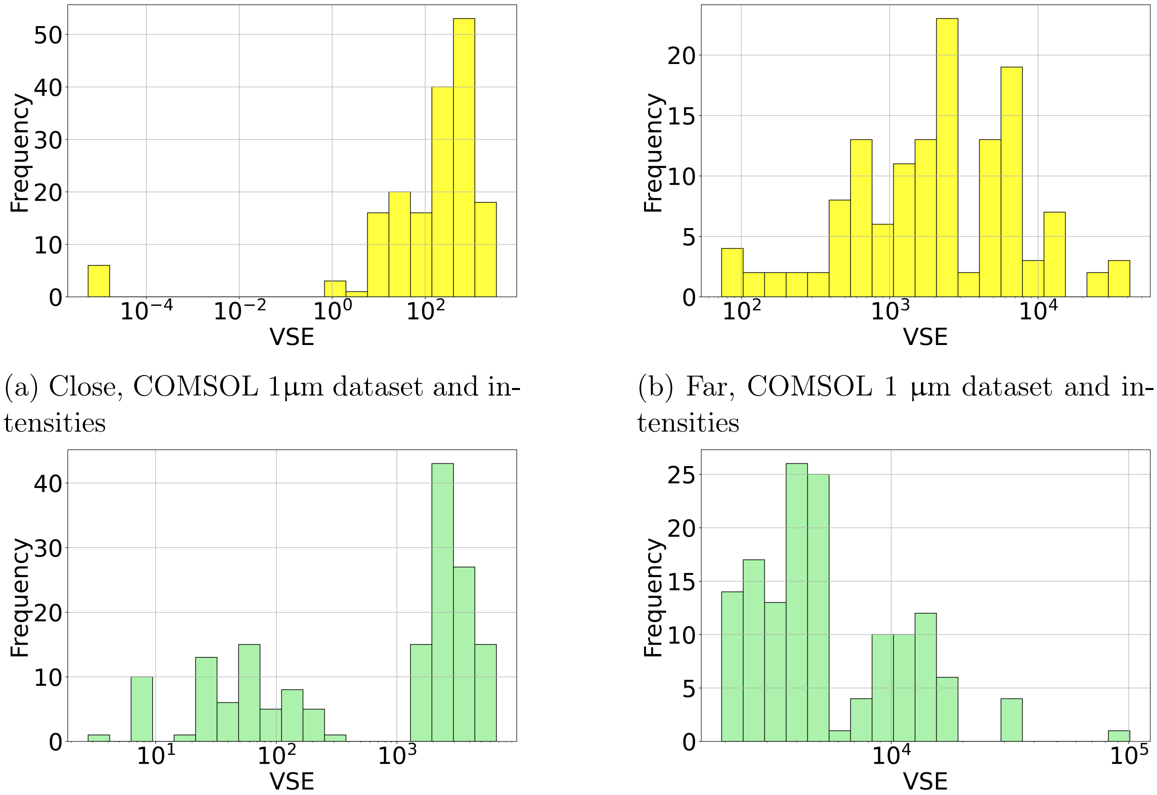
Study case	COMSOL 100µm	COMSOL 100µm and currentFactor	COMSOL 1µm	COMSOL 1µm and currentFactor
VSE scaled (avg ±sd)	820 ± 1965	6342 ±13115	10416 ± 10002	3562627338 ±154
Execution Time (avg ± sd) (mm:ss)	21 :04 ± 14 :11	11:51 ± 12:52	25:16 seconds ± 14 :00	15:06 ± 13:38
N iter (avg±sd)	734.41 ± 512.67	310.35 ± 381.092	1042.828 ± 672.16	371.85 ± 368.13
% of non convergence	12.8 %	4.43%	26.59%	4.43%
Und Activated pixels (avg±sd)	52.24 ± 34.04	56.55 ± 37.35	56.93 ± 28.22	38.94 ± 25.78
% wrongly activated pixels (avg±sd)	14.47 ±9.43 %.	15.66 ± 10.34%	15.77 ±7.81 %.	10.79 ± 7.14%
Current Factor (avg)	0.015	0.010	0.073	0.004

the performance obtained before, see Table 5.2.

From this comparison, it can be concluded that this approach made the algorithm converge much faster as evidenced by the reduced average execution time and fewer iterations. However, the values given for the visual error were higher. Furthermore, for the case with COMSOL cutplane at 100µm, the average percentage of wrongly activated pixels was higher for the intensity minimization case. Contrarily, for the COMSOL cut plane at 1µm it was smaller in the intensity case.

The output current values across all cases are compared, as shown in the "Current Factor" row in Table 5.2. This row represents the average sum of squared current values. The data indicate that in cases where intensity minimization was optimized, the sum of currents was lower than in the other cases. Specifically,in the COMSOL 100µm case the current factor was reduced by 31% at the cost of an increase of the VSE by a factor of 7.73. In the case of COMSOL 1µm the current factor was reduced by 94% along with a 65% reduction in the VSE.

As in previous study cases, an analysis was conducted to differentiate the pixels based on their proximity to the electrodes, categorized as either close (within 300 µm) or far. This distinction is visualized in Fig. 5.17. As observed in previous cases, target



(a) Close, COMSOL 1µm dataset and intensities

(b) Far, COMSOL 1 µm dataset and intensities

(c) Close, COMSOL 100µm dataset and intensities

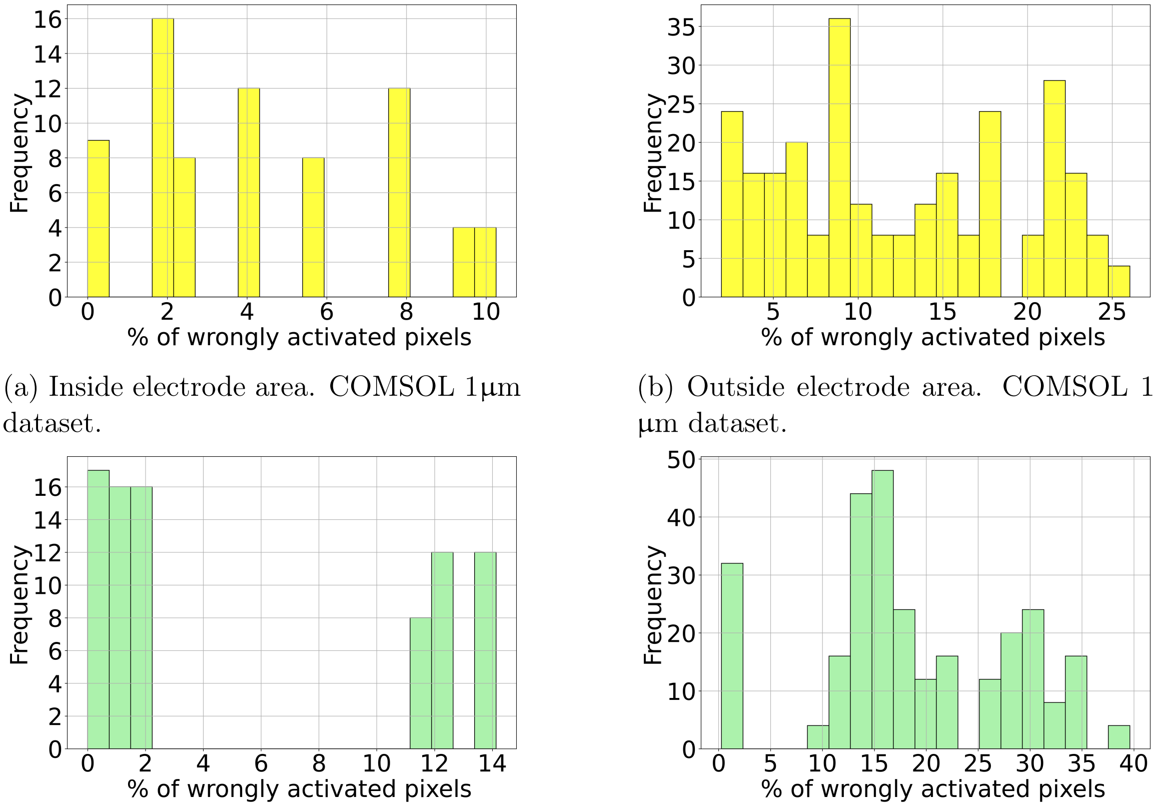
(d) Far, COMSOL 100µm dataset and intensities

Figure 5.17: Representation of how the VSE distributes along the overall dataset distinguishing between points close (a and c) and far from electrodes (b and d) for COMSOL dataset extracted at 1µm cut plane (a and b) and 100µm cut plane (c and d) when intensity minimization is also performed.

points closer to the electrodes exhibited smaller values of VSE. Fig. 5.9 distinguishes between targets in the electrode area and those outside, as visualized in Fig. 5.18. For both datasets the 1µm and the one from the 100 µm cut plane, target points within the electrode area demonstrated better performance. Specifically, they presented a lower percentage of wrongly activated pixels compared to the targets outside the electrode area.

5.4.5 Study case 5: Multi-target VSE minimization

As outlined in Section 3.3, this thesis aimed to explore the potential of Vector Field Stimulation (VFS) and assess its impact on improving spatial resolution. Building on the performance assessments from previous studies, a new case study was conducted.



(a) Inside electrode area. COMSOL 1μm dataset.

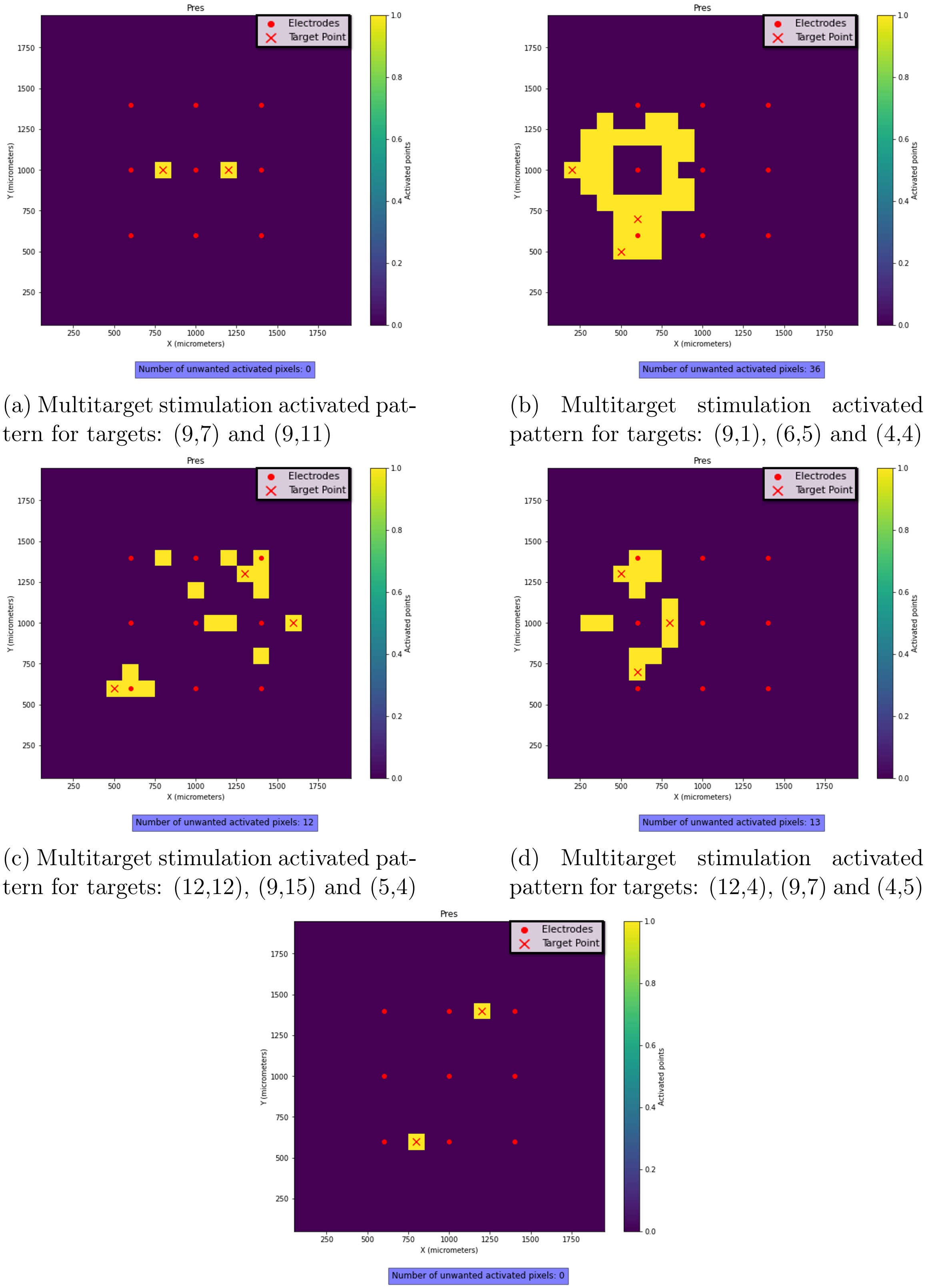
(b) Outside electrode area. COMSOL 1 μm dataset.

(c) Inside electrode area. COMSOL 100μm dataset.

(d) Outside electrode area. COMSOL 100μm dataset.

Figure 5.18: Representation of how the VSE distributes along the overall dataset distinguishing between points inside (a and c) and outside the electrode area (b and d) for COMSOL dataset extracted at 1μm cut plane (a and b) and 100μm cut plane (c and d) when intensity minimization is also performed.

In this study, multiple target points were simultaneously considered. Given the vast number of possible combinations for multitarget stimulation, several pairs and groups of targets were randomly selected and used as inputs. Each scenario was executed 50 times with different random initial guesses, and the optimal solution was determined based on the minimum Visual Sensory Error (VSE) across all runs. The following figures illustrate the activation patterns generated by the algorithm for the various multitarget scenarios, see Fig. 5.19: It can be observed that two of the multitarget cases achieved a VSE of 0 and that all the other cases activated points that were close to the desired targets. This demonstrates that the algorithm performs well in delivering accurate stimulation, with several scenarios achieving optimal precision and others coming very close to the desired outcomes.



(a) Multitarget stimulation activated pattern for targets: (9,7) and (9,11)

(b) Multitarget stimulation activated pattern for targets: (9,1), (6,5) and (4,4)

(c) Multitarget stimulation activated pattern for targets: (12,12), (9,15) and (5,4)

(d) Multitarget stimulation activated pattern for targets: (12,4), (9,7) and (4,5)

(e) Multitarget stimulation activated pattern for targets: (5,7) and (13,11)

Figure 5.19: Multitarget stimulation optimal solutions visualization.

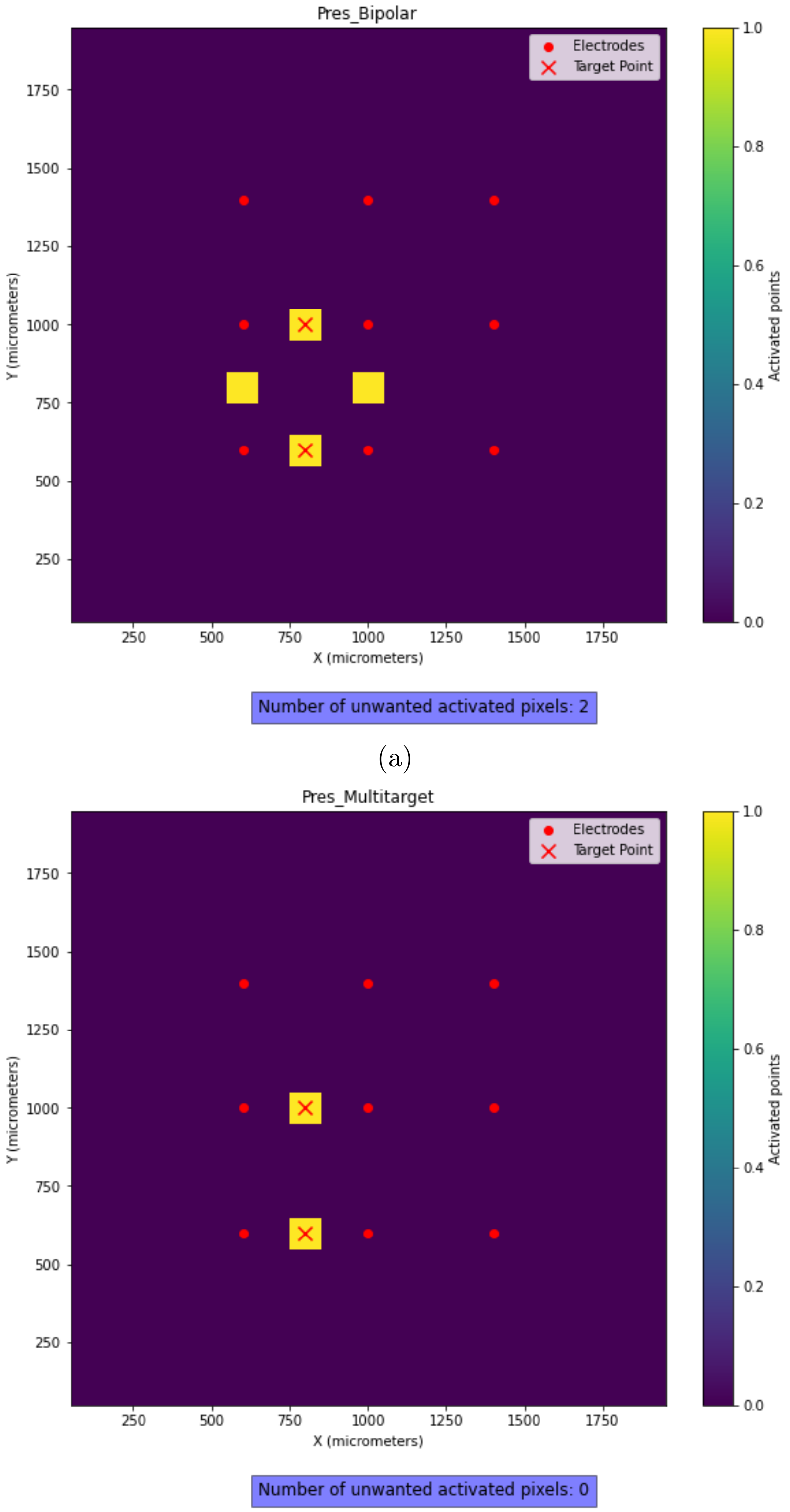
Furthermore, a comparison between this technique and bipolar stimulation was performed as shown in Fig. 5.20. In the bipolar case, the only available electrodes were electrodes $2_1, 2_2, 3_1$ and 3_2, which were paired. The result from comparing bipolar stimulation, commonly used in clinical practice, with vector field stimulation reveals that the percentage of wrongly activated pixels increased for the bipolar case. In contrast, for the multichannel vector field stimulation the VSE for achieving this activation pattern was 0.

5.4.6 Study case 6: using exponential objective function

Another study was performed to enhance the focality of this approach. Since, from the results, it was found that the newly introduced objective function, described in Eq. (4.15), does not consistently ensure that the point furthest from the target will yield the highest VSE value. This is because the VSE also depends on the potential field generated at that point. Specifically, if two points have the same potential value but different distances from the target, the point farther from the target would generally contribute more to the VSE. However, this is not always the case if the potential values differ. To address this limitation, a new objective function was explored where the VSE increases exponentially with the distance from the point to the target Eq. (5.3).

$$\Delta_\mathrm{j} = \exp\left(\min\left(\Delta_\mathrm{ji}\right)\right) \cdot \left(\mathrm{T_{res}[j]} - \mathrm{threshold}\right), \; \mathrm{i} = 1 \ldots \mathrm{N_{on}} \text{ and } \mathrm{j} = 1 \ldots \mathrm{N_{UDE}} \tag{5.3}$$

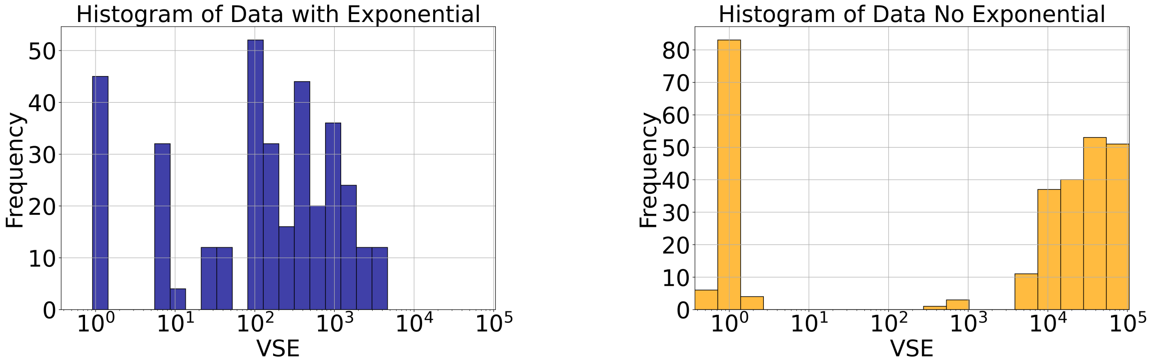
Where the distance from the target to the activated pixels is defined by $\min(\Delta_{ji})$. Fig. 5.21 illustrates the difference in VSE between the results obtained using the exponential objective function and those obtained using the non-exponential function. It is important to note that the VSE for the exponential results was recalculated using the original formulation with the output decision variables, ensuring that the VSE measures were comparable between the two cases. From these plots, it can be observed that the visual error with the exponential objective function reaches values approximately two orders of magnitude higher than those with the original approach.



(a)

(b)

Figure 5.20: Comparisson between activating pattern for target points [9,7] and [5,7] when using (a) bipolar stimulation (b) vector field stimulation.

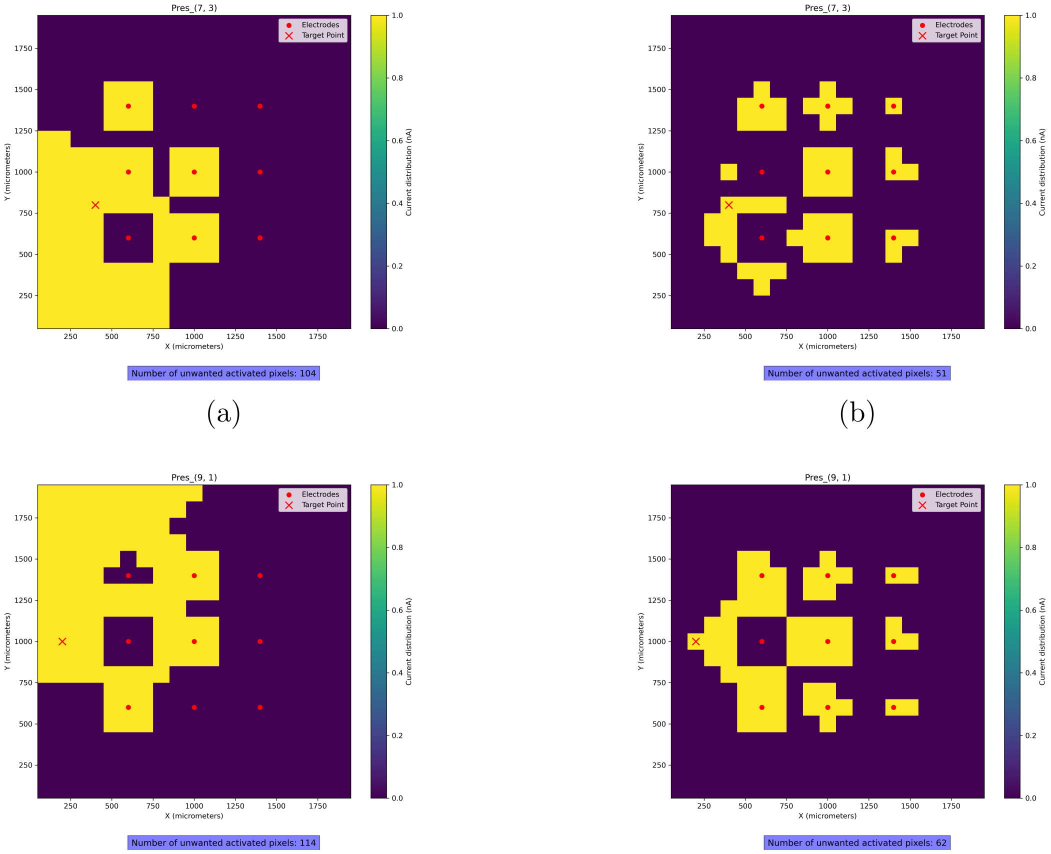


(a)

(b)

Figure 5.21: Visualization of histograms comparing the VSE with Green Function data, (a) Exponential vs (b) non Exponential Objective function

However, as shown in Fig. 5.22, despite the higher VSE and the activation of more unwanted pixels, these pixels are closer to the target, indicating that the exponential approach is more focal. This focality was assessed by calculating the average distance between the target and the undesirably activated points. For the target [7,3], the average distance to the target was 7.72 for the non-exponential case and 5.96 for the exponential case. In the target [9,1] case, the average distance was 6.72 for the non-exponential approach and 6.18 for the exponential approach. This indicates better focality with the exponential approach, as it produced a smaller average distance in both cases.



(a)

(b)

(c)

(d)

Figure 5.22: Visualization of activated patterns for target points [9,1] and [7,3], with (a and c) Exponential objective function and (b and d) with original objective function

Chapter 6

Discussion and conclusions

6.1 Discussion

The primary goal of this project was to investigate the potential of Vector Field Stimulation (VFS) to improve microstimulation delivery in the visual cortex. To achieve this, an optimization algorithm was developed and applied to a 3x3 Microelectrode Array (MEA) modelled in COMSOL. This algorithm determines the optimal current for each electrode in the MEA to generate a desired electric field while minimizing visual error. The research question addressed was: How does vector field stimulation affect visual error and the precision of target stimulation in a cortical visual prosthesis?

The algorithm's input, comprising the individual electrical contributions of each electrode in the MEA, was derived from a 3D model created in COMSOL. An analytical approach using Green's Functions was employed to validate the algorithm's performance without discretization errors. These functions approximate the Poisson equation solved by the Finite Element Method (FEM) in COMSOL, applicable when the points of interest are distant from the current source.

The problem setup and algorithm were implemented in Python, utilizing the interior point method. This method is particularly suited for this problem because it can handle continuous, non-linear objective functions with constraints.

This project introduces several innovations compared to previous studies referenced in the literature. For example, the visual error equation (VSE) presented in [2], was extended by introducing a term that ensures the objective function remains continuous. This term quantifies the deviation of the current density at the activated point from the threshold.

Additionally, an analysis was conducted to explore the effects of minimizing the amount of current delivered while achieving a specific target. The results showed improved performance, including a reduced number of iterations needed to reach an optimal solution, shorter execution times, and a lower percentage of non-convergence. Incorporating the minimization of current intensities into the objective function enhanced its sensitivity, leading to better convergence. By increasing the sensitivity of the objective function, the algorithm improved in getting trapped in local minima.

Moreover, an analysis comparing bipolar stimulation and vector field stimulation was performed. The results were promising for vector field stimulation, as it achieved better precision in target stimulation. These findings align with those presented in [1], where a similar outcome was visualized Fig. 6.1. Despite the brevity of the conference paper, both studies highlight the potential of this technique in enhancing target stimulation precision. Furthermore, additional studies were conducted to explore methods for enhancing focality. The proposed exponential method achieved better focality. However, it also resulted in a higher visual error (VSE) Fig. 5.21. This highlights the need to carefully evaluate whether the priority should be on improving focality or reducing visual error, as finding an optimal balance between these two factors is crucial for achieving the best overall performance.

6.2 Future developments

The proposed algorithm works well and has achieved promising results. However, there is still some room for improvement. One critical next step is addressing the issue of local minima, which can impede the algorithm's ability to find the global optimum. Currently, the approach involves multi-start techniques, where the algorithm is run multiple times from different starting points, and the best

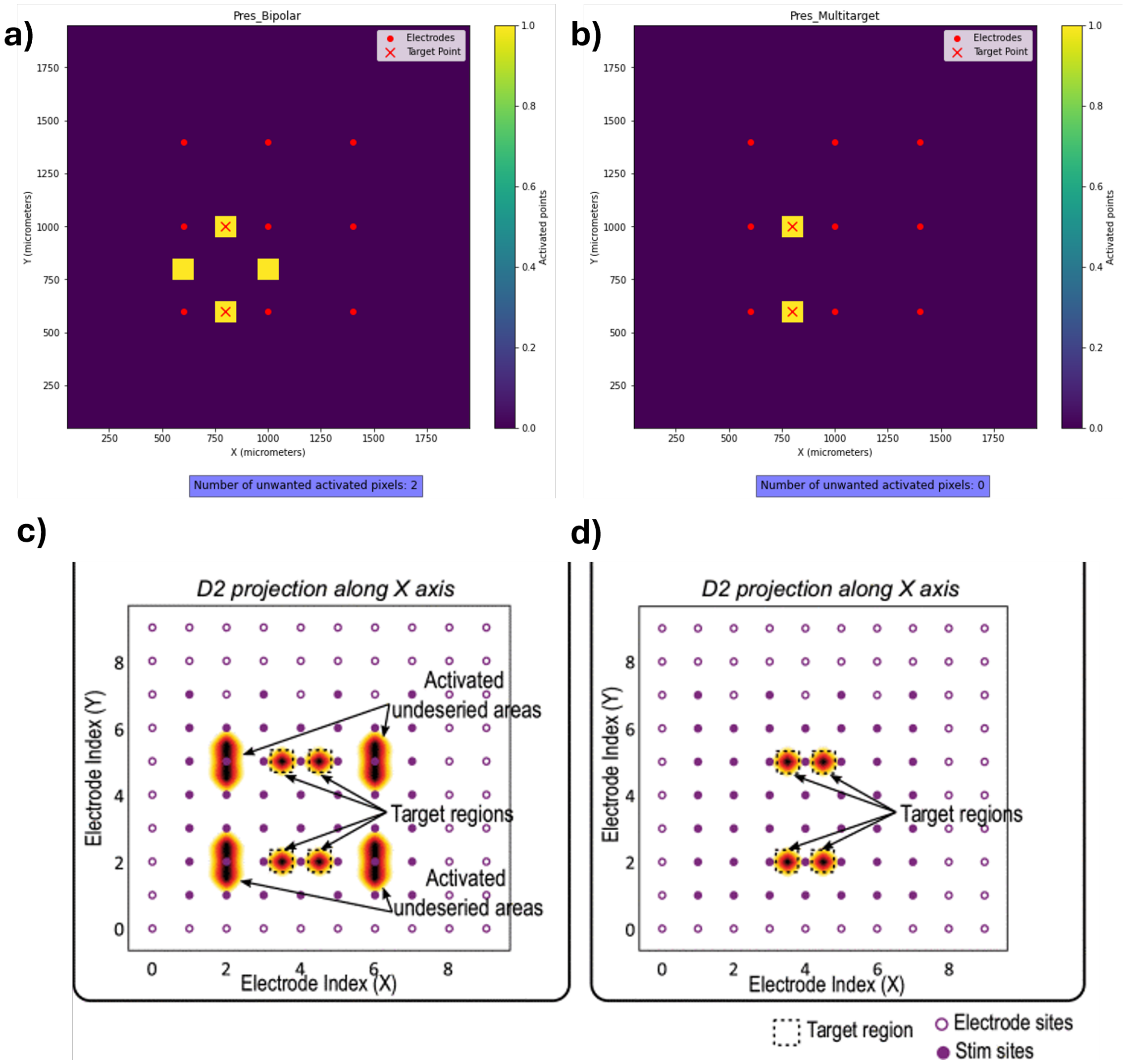


Figure 6.1: Activation patterns generated using different stimulation methods. (a) Activation pattern generated using bipolar stimulation in this study. (b) Activation pattern generated using vector field stimulation in this study. (c) Activation pattern generated using bipolar stimulation as presented in [1]. (d) Activation pattern generated using vector field stimulation as presented in [1].

result is selected from these runs. Additionally, each target is processed through several iterations to find the minimum value. Another improvement suggested and tried in this study is the minimization of current intensities, which has increased the sensitivity of the objective function and improved convergence. Despite these advancements, there is still room for improvement. Exploring additional advanced global optimization techniques would be a big next step.

Another significant development would be transitioning from 2D to 3D. As illustrated in Figures Fig. 5.1b and Fig. 5.1c, potential field distributions vary significantly across different levels. A 3D model would provide a more comprehensive understanding of how electric fields interact in three-dimensional space, leading to more precise and effective stimulation patterns.

Additionally, numerical errors observed in the COMSOL simulations indicate the need for improved simulation accuracy. Future efforts should focus on refining the mesh to achieve higher resolution and developing post-processing filtering methods to correct or minimize these errors. Implementing advanced filtering techniques, such as smoothing algorithms or error correction filters, could further enhance the reliability of the simulation results.

Moreover, as discussed in the report, the activating function effect is influenced by the neuron's orientation relative to the electric field generated by stimulation. Currently, this project computes the average between the x and y directions due to the unknown orientation of the neurons. To enhance the precision of this stimulation approach, incorporating neuron orientation with respect to the electric field at each pixel would be a possible next step. By optimizing the stimulation to the exact orientation of each neuron, the unintended activation of adjacent neurons that are not aligned with the electric field could be minimized. This can reduce cross-talk, where stimulation intended for one neuron accidentally affects others, a problem that is also very common in retinal prosthesis [93].

Finally, once the Biolab in the bioelectronics department is fully operational and the algorithm is optimized in 3D, it can be validated by testing on cell cultures using a MEA. This approach will help assess its performance and accuracy in a controlled and biologically relevant setting.

6.3 Conclusion

In conclusion, this report has explored the potential of vector field stimulation as a novel technique for enhancing visual cortex stimulation. Additionally, an algorithm was designed and implemented to test this concept. The findings suggest that vector field stimulation, facilitated by the newly developed algorithm, can significantly improve the precision of targeted stimulation. These results are crucial for improving visual prostheses, which aim to restore sight to the blind. While the study faced limitations such as numerical errors in the COMSOL simulations and the algorithm getting trapped in local minima, it highlights the need for further research into improving current methods of visual cortex microstimulation. Future studies should focus on addressing numerical errors, transitioning from 2D to 3D study models, and solving the local minima problem. Overall, this report highlights the promising potential of vector field stimulation in improving existing stimulation techniques.

Bibliography

[1] Arindam Mandal, Diego Peña, Rajesh Pamula, Karam Khateeb, Logan Murphy, Azadeh Yazdan-Shahmorad, Steve Perlmutter, Forrest Pape, Jacques C Rudell, and Visvesh Sathe. A 46-channel vector stimulator with 50mv worst-case common-mode artifact for low-latency adaptive closed-loop neuromodulation. In *2021 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–2. IEEE, 2021.

[2] Gehan Abouelseoud, Yasmine Abouelseoud, Amin Shoukry, Nour Ismail, and Jaidaa Mekky. A mixed integer linear programming approach to electrical stimulation optimization problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):527–537, 2018.

[3] Armin Najarpour Foroushani, Christopher C Pack, and Mohamad Sawan. Cortical visual prostheses: from microstimulation to functional percept. *Journal of neural engineering*, 15(2):021005, 2018.

[4] Lalit Dandona and Rakhi Dandona. What is the global burden of visual impairment? *BMC medicine*, 4:1–10, 2006.

[5] Léo Pio-Lopez, Romanos Poulkouras, and Damien Depannemaecker. Visual cortical prosthesis: an electrical perspective. *Journal of Medical Engineering & Technology*, 45(5):394–407, 2021.

[6] World Health Organization et al. World report on vision. geneva, switzerland: World health organization; 2019.

[7] Gehan Abouelseoud, Yasmine Abouelseoud, Amin Shoukry, Nour Ismail, and Jaidaa Mekky. A mixed integer linear programming framework for improving cortical vision prosthesis designs. *Biomedical Signal Processing and Control*, 80:104253, 2023.

[8] Andrej Kral, Felix Aplin, and Hannes Maier. *Prostheses for the Brain: Introduction to Neuroprosthetics*. Academic Press, 2021.

[9] Brian A Wandell, Serge O Dumoulin, and Alyssa A Brewer. Visual cortex in humans. *Encyclopedia of neuroscience*, 10:251–257, 2009.

[10] Eduardo Fernandez and Richard Normann. Introduction to visual prostheses. 2016.

[11] Perkins School for the Blind. The visual pathway from the eye to the brain, 2024. Accessed on: 25th January 2024.

[12] Wikipedia contributors. Human visual pathway, N/A. Accessed on: [January 2024].

[13] Joseph F Rizzo, John Wyatt, John Loewenstein, Shawn Kelly, and Doug Shire. Methods and perceptual thresholds for short-term electrical stimulation of human retina with microelectrode arrays. *Investigative ophthalmology & visual science*, 44(12):5355–5361, 2003.

[14] Eberhart Zrenner, Karl Ulrich Bartz-Schmidt, Heval Benav, Dorothea Besch, Anna Bruckmann, Veit-Peter Gabel, Florian Gekeler, Udo Greppmaier, Alex Harscher, Steffen Kibbel, et al. Subretinal electronic chips allow blind patients to read letters and combine them to words. *Proceedings of the Royal Society B: Biological Sciences*, 278(1711):1489–1497, 2011.

[15] Mark S Humayun, James D Weiland, Gildo Y Fujii, Robert Greenberg, Richard Williamson, Jim Little, Brian Mech, Valerie Cimmarusti, Gretchen Van Boemel, Gislin Dagnelie, et al. Visual perception in a blind subject with a chronic microelectronic retinal prosthesis. *Vision research*, 43(24):2573–2581, 2003.

[16] John S Pezaris and R Clay Reid. Demonstration of artificial visual percepts generated through thalamic microstimulation. *Proceedings of the National Academy of Sciences*, 104(18):7670–7675, 2007.

[17] Edwin M Maynard. Visual prostheses. *Annual review of biomedical engineering*, 3(1):145–168, 2001.

[18] Claude Veraart, Christian Raftopoulos, J Thomas Mortimer, Jean Delbeke, Delphine Pins, Geraldine Michaux, Annick Vanlierde, Simone Parrini, and Marie-Chantal Wanet-Defalque. Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode. *Brain research*, 813(1):181–186, 1998.

[19] Hirokazu Sakaguchi, Motohiro Kamei, Takashi Fujikado, Eiji Yonezawa, Motoki Ozawa, Carmen Cecilia-Gonzalez, Orlando Ustariz-Gonzalez, Hugo Quiroz-Mercado, and Yasuo Tano. Artificial vision by direct optic nerve electrode (av-done) implantation in a blind patient with retinitis pigmentosa. *Journal of Artificial Organs*, 12:206–209, 2009.

[20] Må E Brelén, F Duret, Benoît Gérard, Jean Delbeke, and Claude Veraart. Creating a meaningful visual perception in blind volunteers by optic nerve stimulation. *Journal of neural engineering*, 2(1):S22, 2005.

[21] Edward J Tehovnik, Warren M Slocum, Stelios M Smirnakis, and Andreas S Tolias. Microstimulation of visual cortex to restore vision. *Progress in brain research*, 175:347–375, 2009.

[22] Richard A Normann, Bradley A Greger, Paul House, Samuel F Romero, Francisco Pelayo, and Eduardo Fernandez. Toward the development of a cortically based visual neuroprosthesis. *Journal of neural engineering*, 6(3):035001, 2009.

[23] Calvin Eiber. *Field Shaping in Retinal Prosthesis: Current Focusing, Current Steering, and Simultaneous Stimulation*. PhD thesis, UNSW Sydney, 2015.

[24] Francesc Varkevisser. A charge controlled switched-voltage mode neurostimulator: For effective, safe and efficient multi-channel intracortical visual stimulation. 2020.

[25] Andrej Kral, Felix Aplin, and Hannes Maier. In *Prostheses for the Brain: Introduction to Neuroprosthetics*, chapter 12. Academic Press, 2021.

[26] Sabrina J Meikle, Maureen A Hagan, Nicholas SC Price, and Yan T Wong. Cortical layering disrupts multi-electrode current steering. *Journal of Neural Engineering*, 2023.

[27] Julien Vezoli, Yujie Hou, and Henry Kennedy. The evolving concept of cortical hierarchy. *The Cerebral Cortex and Thalamus*, page 393, 2023.

[28] Morgan E Urdaneta, Nicolas G Kunigk, Jesus D Peñaloza-Aponte, Seth Currlin, Ian G Malone, Shelley I Fried, and Kevin J Otto. Layer-dependent stability of intracortical recordings and neuronal cell loss. *Frontiers in Neuroscience*, 17:503, 2023.

[29] Edward J Tehovnik and Warren M Slocum. Phosphene induction by microstimulation of macaque v1. *Brain research reviews*, 53(2):337–343, 2007.

[30] Gabriele Peters. Aesthetic primitives of images for visualization. In *2007 11th International Conference Information Visualization (IV'07)*, pages 316–325. IEEE, 2007.

[31] AA Galakhova, S Hunt, R Wilbers, DB Heyer, CPJ de Kock, HD Mansvelder, and NA Goriounova. Evolution of cortical neurons supporting human cognition. *Trends in Cognitive Sciences*, 26(11):909–922, 2022.

[32] Sabrina J Meikle, Maureen A Hagan, Nicholas SC Price, and Yan T Wong. Intracortical current steering shifts the location of evoked neural activity. *Journal of Neural Engineering*, 19(3):035003, 2022.

[33] VJ Bolfe, SI Ribas, MIL Montebelo, and RRJ Guirro. Electrical impedance behavior of biological tissues during transcutaneous electrical stimulation. *Brazilian Journal of Physical Therapy*, 11:153–159, 2007.

[34] Andrej Kral, Felix Aplin, and Hannes Maier. In *Prostheses for the Brain: Introduction to Neuroprosthetics*, chapter 5. Academic Press, 2021.

[35] Andrej Kral, Felix Aplin, and Hannes Maier. In *Prostheses for the Brain: Introduction to Neuroprosthetics*, chapter 6. Academic Press, 2021.

[36] Andrej Kral, Felix Aplin, and Hannes Maier. In *Prostheses for the Brain: Introduction to Neuroprosthetics*, chapter 4. Academic Press, 2021.

[37] Jesse Musokota Mumba, Freddy Kasandji Kabambi, and Christian Tshebeletso Ngaka. Pharmacology of local anaesthetics and commonly used recipes in clinical practice. In *Current topics in anesthesiology*. IntechOpen, 2017.

[38] Desdemona Fricker, Jos AH Verheugen, and Richard Miles. Cell-attached measurements of the firing threshold of rat hippocampal neurones. *The Journal of physiology*, 517(3):791–804, 1999.

[39] Steven F Ronner and Bothwell G Lee. Excitation of visual cortex neurons by local intracortical microstimulation. *Experimental neurology*, 81(2):376–395, 1983.

[40] Edward M Schmidt, Martin J Bak, F Terry Hambrecht, Conrad V Kufta, DK O'rourke, and Prashanth Vallabhanath. Feasibility of a visual prosthesis for the blind based on intracortical micro stimulation of the visual cortex. *Brain*, 119(2):507–522, 1996.

[41] Adrien B Rapeaux and Timothy G Constandinou. Implantable brain machine interfaces: first-in-human studies, technology challenges and trends. *Current opinion in biotechnology*, 72:102–111, 2021.

[42] Samuel C Colachis, Collin F Dunlap, Nicholas V Annetta, Sanjay M Tamrakar, Marcia A Bockbrader, and David A Friedenberg. Long-term intracortical microelectrode array performance in a human: a 5 year retrospective analysis. *Journal of Neural Engineering*, 18(4):0460d7, 2021.

[43] Utah Array. https://blackrockneurotech.com/products/utah-array/. Accessed: Feb 11, 2024.

[44] Stefan Eick, Jens Wallys, Boris Hofmann, André Van Ooyen, Uwe Schnakenberg, Sven Ingebrandt, and Andreas Offenhäusser. Iridium oxide microelectrode arrays for in vitro stimulation of individual rat neurons from dissociated cultures. *Frontiers in neuroengineering*, 2:856, 2009.

[45] Seung-Jae Kim, Sandeep C Manyam, David J Warren, and Richard A Normann. Electrophysiological mapping of cat primary auditory cortex with multielectrode arrays. *Annals of biomedical engineering*, 34:300–309, 2006.

[46] Cogan Stuart F. Limitations and opportunities with neural stimulation electrodes. Oral presentation, 2010. Presented at Case Western Reserve University, Cleveland, Ohio.

[47] Stuart F Cogan, Kip A Ludwig, Cristin G Welle, and Pavel Takmakov. Tissue damage thresholds during therapeutic electrical stimulation. *Journal of neural engineering*, 13(2):021001, 2016.

[48] Daria Nesterovich Anderson, Braxton Osting, Johannes Vorwerk, Alan D Dorval, and Christopher R Butson. Optimized programming algorithm for cylindrical and directional deep brain stimulation electrodes. *Journal of neural engineering*, 15(2):026005, 2018.

[49] Robert V Shannon. A model of safe levels for electrical stimulation. *IEEE Transactions on biomedical engineering*, 39(4):424–426, 1992.

[50] Douglas B McCreery, William F Agnew, Ted GH Yuen, and Leo Bullara. Charge density and charge per phase as cofactors in neural injury induced by electrical stimulation. *IEEE transactions on Biomedical Engineering*, 37(10):996–1001, 1990.

[51] Sandeep Negi, Rajmohan Bhandari, Loren Rieth, Rick Van Wagenen, and Florian Solzbacher. Neural electrode degradation from continuous electrical stimulation: comparison of sputtered and activated iridium oxide. *Journal of neuroscience methods*, 186(1):8–17, 2010.

[52] Andrej Kral, Felix Aplin, and Hannes Maier. In *Prostheses for the Brain: Introduction to Neuroprosthetics*, chapter 7. Academic Press, 2021.

[53] I Winkler, S Denham, C Escera, D Jaeger, and R Jung. Encyclopedia of computational neuroscience. 2015.

[54] Christof Koch. Cable theory in neurons with active, linearized membranes. *Biological cybernetics*, 50(1):15–33, 1984.

[55] Maxim Komarov, Paola Malerba, Ryan Golden, Paul Nunez, Eric Halgren, and Maxim Bazhenov. Selective recruitment of cortical neurons by electrical stimulation. *PLoS computational biology*, 15(8):e1007277, 2019.

[56] Vasiliki Giagka. Bioelectricity – et4130 lecture 2, February 15 2023. Lecture presented in ET4130 course.

[57] Dr. Jaakko Malmivuo. BEM - Biological Effects of Electromagnetic Fields. https://www.bem.fi/book/03/03.htm, 1993. Accessed: March 11, 2024.

[58] Frank Rattay. Analysis of models for external stimulation of axons. *IEEE transactions on biomedical engineering*, (10):974–977, 1986.

[59] Sébastien Joucla and Blaise Yvert. Modeling extracellular electrical neural stimulation: from basic understanding to mea-based applications. *Journal of Physiology-Paris*, 106(3-4):146–158, 2012.

[60] Lakshmi Narayan Mishra, Gaurav Kulkarni, and Mandar Gadgil. A novel current steering method for targeted spinal cord stimulation. *Frontiers in Pain Research*, 4:1028368, 2023.

[61] Gerald Dumm, James B Fallon, Chris E Williams, and Mohit N Shivdasani. Virtual electrodes by current steering in retinal prostheses. *Investigative ophthalmology & visual science*, 55(12):8077–8085, 2014.

[62] Thomas C Spencer, James B Fallon, and Mohit N Shivdasani. Creating virtual electrodes with 2d current steering. *Journal of neural engineering*, 15(3):035002, 2018.

[63] Jill B Firszt, Dawn Burton Koch, Mark Downing, and Leonid Litvak. Current steering creates additional pitch percepts in adult cochlear implant recipients. *Otology & Neurotology*, 28(5):629–636, 2007.

[64] Jeffrey V Rosenfeld, Yan T Wong, Edwin Yan, Julian Szlawski, Anand Mohan, Jonathan CM Clark, Marcello Rosa, and Arthur Lowery. Tissue response to a chronically implantable wireless intracortical visual prosthesis (gennaris array). *Journal of neural engineering*, 17(4):046001, 2020.

[65] Michael S Beauchamp, Denise Oswalt, Ping Sun, Brett L Foster, John F Magnotti, Soroush Niketeghad, Nader Pouratian, William H Bosking, and Daniel Yoshor. Dynamic stimulation of visual cortex produces form vision in sighted and blind humans. *Cell*, 181(4):774–783, 2020.

[66] Craig O Savage, Filiz Isabell Kiral-Kornek, Bahman Tahayori, and David B Grayden. Can electric current steering be used to control perception of a retinal prosthesis patient. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3013–3016. IEEE, 2012.

[67] NL Opie, Nigel H Lovell, Gregg J Suaning, Philip Preston, and Socrates Dokos. Current steering for high resolution retinal implants. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2760–2763. IEEE, 2013.

[68] FJO Rodrigues, P Mendes, Marian Bartek, and BAZ Mimoun. A steering electrode array for selective stimulation of sacral nerve roots. *Proceedings ICT. OPEN: Micro Technology and Micro Devices (SAFE 2011), Veldhoven, Nov. 14-15 2011, 1-4*, 2011.

[69] Zhijie Charles Chen, Bing-Yi Wang, and Daniel Palanker. Real-time optimization of the current steering for visual prosthesis. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 592–596. IEEE, 2021.

[70] Gehan Abouelseoud, Yasmine Abouelseoud, Amin Shoukry, Nour Ismail, and Jaidaa Mekky. On the use of time division multiplexing to improve electrical

brain stimulation focality. *Biomedical Signal Processing and Control*, 61:102048, 2020.

[71] Cameron C McIntyre, Susumu Mori, David L Sherman, Nitish V Thakor, and Jerrold L Vitek. Electric field and stimulating influence generated by deep brain stimulation of the subthalamic nucleus. *Clinical neurophysiology*, 115(3):589–595, 2004.

[72] HCF Martens, E Toader, MMJ Decré, DJ Anderson, R Vetter, DR Kipke, Kenneth B Baker, Matthew D Johnson, and Jerrold L Vitek. Spatial steering of deep brain stimulation volumes using a novel lead design. *Clinical neurophysiology*, 122(3):558–566, 2011.

[73] NEURON documentation. https://nrn.readthedocs.io/en/8.2.4/. Accessed: March 24, 2024.

[74] Noah C Benson, Omar H Butt, Ritobrato Datta, Petya D Radoeva, David H Brainard, and Geoffrey K Aguirre. The retinotopic organization of striate cortex is well predicted by surface topology. *Current Biology*, 22(21):2081–2085, 2012.

[75] Ac/dc module user's guide, Jun 2023.

[76] Gislin Dagnelie. *Visual prosthetics: physiology, bioengineering, rehabilitation*. Springer Science & Business Media, 2011.

[77] Moritz Dannhauer, Dana Brooks, Don Tucker, and Rob MacLeod. A pipeline for the simulation of transcranial direct current stimulation for realistic human head models using scirun/biomesh3d. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5486–5489. IEEE, 2012.

[78] Kelly E Jones, Patrick K Campbell, and Richard A Normann. A glass/silicon composite intracortical electrode array. *Annals of biomedical engineering*, 20:423–437, 1992.

[79] May 2024.

[80] Jan Liu, Oemer Atmaca, Toni J Ly, and Peter P Pott. Numerical sensitivity analysis of microelectrodes for multi-local im-pedance measurements on needles. In *Mikro-Nano-Integration; 9. GMM-Workshop*, pages 1–5. VDE, 2022.

[81] Yiqian Mao, Qiwei Zhan, Qingtao Sun, Dezhi Wang, and Qing Huo Liu. Mesh-splitting impedance transition boundary condition for accurate modeling of thin structures. *IEEE Transactions on Antennas and Propagation*, 2023.

[82] Henrik Wallén and Juhani Kataja. Some computational aspects of too sharp edges. In *CEM'11 Computational Electromagnetics International Workshop*, pages 70–74. IEEE, 2011.

[83] Nicole A Pelot, Brandon J Thio, and Warren M Grill. Modeling current sources for neural stimulation in comsol. *Frontiers in computational neuroscience*, 12:40, 2018.

[84] Eric Michel, Daniel Hernandez, and Soo Yeol Lee. Electrical conductivity and permittivity maps of brain tissues derived from water content based on t1-weighted acquisition. *Magnetic resonance in medicine*, 77(3):1094–1103, 2017.

[85] Gaofeng Zhou, Yannian Wang, and Lujun Cui. Biomedical sensor, device and measurement systems. *Advances in bioengineering*, 177, 2015.

[86] Understanding and changing the element order. https://www.comsol.com/support/learning-center/article/Understanding-and-Changing-the-Element-Order-47301. Accessed: 2024-07-1.

[87] Valerie Illingworth and J. Cullerne. *The Penguin Dictionary of Physics*. Penguin Books Ltd, August 6 2009. English Paperback.

[88] Kishore Rama Rao, John Nehrbass, and Robert Lee. Discretization errors in finite methods: Issues and possible solutions. *Computer methods in applied mechanics and engineering*, 169(3-4):219–236, 1999.

[89] Evgeny Tsymbal. Electrostatics. https://unlcms.unl.edu/cas/physics/tsymbal/teaching/EM-913/section2-Electrostatics.pdf, 2024. Lecture Notes.

[90] Naveen Balaji. Linear programming: Simplex and interior point methods, 2023. Roll No: 170420.

[91] SciPy Community. scipy.optimize.minimize — scipy v1.9.1 manual, 2024. Accessed: 2024-07-02.

[92] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[93] Alejandro Barriga-Rivera, Lilach Bareket, Josef Goding, Ulises A Aregueta-Robles, and Gregg J Suaning. Visual prosthesis: interfacing stimulating electrodes with retinal neurons to restore vision. *Frontiers in neuroscience*, 11:620, 2017.

Appendix A

Green Functions code for potential field analytical input

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters
L_visualize = 2000 # 2000 micrometers
resolution = 10 # 10 micrometers
N_visualize = int((L_visualize / resolution)+1) # number of grid points

# Electrode positions (in micrometers)
r_0 = np.array([
    [600, 600, 0],
    [600, 1000, 0],
    [600, 1400, 0],
    [1000, 1400, 0],
    [1000, 1000, 0],
    [1000, 600, 0],
    [1400, 600, 0],
    [1400, 1000, 0],
    [1400, 1400, 0]
])

```

```
# Current injections (in amperes)
I = np.array([0, 0, 1, 0, 0, 0, 0, 0, 0])

# Define the grid
x = np.linspace(0, L_visualize, N_visualize)
y = np.linspace(0, L_visualize, N_visualize)
z = np.linspace(0, L_visualize, N_visualize)
X, Y, Z = np.meshgrid(x, y, z, indexing='ij')

# Conductivity parameters
sigma_tissue = 0.2E-6 # in S/micrometer (SI would be 0.2 S/m for tissue)
sigma_electrode = 8.9 # in S/microm (SI would be 8.9E6 S/m for electrode)

# Initialize potential array
phi_val = np.zeros((N_visualize, N_visualize, N_visualize))

# Apply ground boundary condition for the cube boundary
phi_val[:, :, -1] = 0 # Set potential to zero at the bottom boundary of the cube

# Function to calculate the potential
def f_phi(r_0, r, I, sigma_tissue, sigma_electrode):
    distance = np.linalg.norm(r - r_0)
    conductivity = sigma_electrode if distance < 1e-3 else sigma_tissue
    if distance == 0:
        return 0
    else:
        return I / (4 * np.pi * conductivity * distance)

# Calculate the potential at each point in the grid
for i in range(N_visualize):
    for j in range(N_visualize):
        for k in range(N_visualize):
            r = np.array([x[i], y[j], z[k]])
            for l in range(r_0.shape[0]):
                phi_val[i, j, k] += f_phi(r_0[l], r, I[l], sigma_tissue,
                    sigma_electrode)
```

```

# Convert potential values to millivolts
phi_val *= 1e3 # Conversion from volts to millivolts

# Save the potential field for further analysis or comparison with COMSOL
np.save('potential_field.npy', phi_val)

# Extract the potential values at z = 100 micrometer
z_index = np.argmin(np.abs(z - 100)) # Find index closest to z = 100 $\upmu$m

# Plot the potential slice
plt.figure()
plt.imshow(phi_val[:, :, z_index], extent=(0, L_visualize, 0, L_visualize), origin='lower', cmap='jet')
plt.colorbar(label='Potential (mV)')
plt.xlabel('x ($\upmu$m)')
plt.ylabel('y ($\upmu$m)')
plt.title('2D Slice of Electric Potential at z = 100 $\upmu$m')
plt.axis('equal')
plt.show()

# Extract the slice at z = 100 micrometer
phi_slice = phi_val[:, :, z_index]

# Prepare data for saving
row_indices, col_indices = np.meshgrid(np.arange(N_visualize), np.arange(N_visualize), indexing='ij')
data_to_save = np.column_stack((row_indices.ravel(), col_indices.ravel(), phi_slice.ravel()))

# Save data to a text file
np.savetxt('potential_slice_z100um_electrode3_3.txt', data_to_save, fmt='%d %d %.6e')
```

Appendix B

Problem Functions

```
h = resolution*(10 ** -4)# Now my resolution is in micrometers I have to give it in cm to match units
def second_derivative(Telectrode):
    # I add -2 to account for the edges
    AFe=np.zeros((Telectrode.shape[0]-2, Telectrode.shape[1]-2))
    # To compute second derivative for each (x, y) coordinate
    for i in range(1, Telectrode.shape[0]-1):
        for j in range(1, Telectrode.shape[1]-1):
            d= 1*(10**-4)# axon diameter in cm
            rho= 300 #ohm*centimeter
            # nA/cm2 as an otput
            AFe[i - 1, j - 1] = (d / (4 * rho)) * ((Telectrode[i - 1, j] - 2 * Telectrode[i, j] + Telectrode[i + 1, j]) +
                                                (Telectrode[i, j - 1] - 2 * Telectrode[i, j] + Telectrode[i, j + 1])) * (10 ** 6) / (h ** 2)
    return AFe
```

Listing B.1: Activating Function

```
def activatedpattern(Matrixin, threshold):
    Pres = np.where(np.round(Matrixin) >= threshold, 1, 0)
    return Pres
```

Listing B.2: Activated pattern generation

```python
#I will use this function to get the number of desired activated locations and at what positions in the matrix are they
def count_ones(matrix):
    # Use numpy's vectorized operations to find positions of ones
    ones_mask = matrix == 1
    # Count the number of ones
    ones_count = np.sum(ones_mask)
    # Find the positions of ones and non-ones
    positionstoActivate = np.argwhere(ones_mask)
    positionsNOTActivate = np.argwhere(~ones_mask)
    # Convert the positions to list of tuples
    positionstoActivate = list(map(tuple, positionstoActivate))
    positionsNOTActivate = list(map(tuple, positionsNOTActivate))

    return ones_count, positionstoActivate, positionsNOTActivate
```

Listing B.3: Count activated points in input

```python
#To get the number of undesired activated positions and their location so I can compute the visual error
def undesired_activations(matrixObj,matrixRes):
    matrixObj = np.array(matrixObj)
    matrixRes = np.array(matrixRes)
    # Create a boolean mask where elements are True if they are different
    diff_mask = matrixObj != matrixRes#For broadcating suggested by Si-yang
    Nun = np.sum(diff_mask)
    undisired_activated_positions=[]
    # Find the indices where the mask is True
    undisired_activated_positions = np.argwhere(diff_mask)
    undisired_activated_positions = list(map(tuple, undisired_activated_positions))
    return Nun,undisired_activated_positions
```

Listing B.4: Count number of unactivated positions

```python
def function_positions(matrix):
    indices = np.indices(matrix.shape)
    # Reshape the indices array to have pairs of coordinates
    positions = np.stack(indices, axis=-1).reshape(-1, 2)
    # Convert to a list of tuples
    positions = list(map(tuple, positions))
    return positions
```

Listing B.5: Gives the coordinates of every element in the input matrix

```
def dneigh_function(j, Nloc, position):
    dneighp=[]
    for k in range(Nloc):
        if k!=j:
            dneighp.append(math.sqrt((position[k][0]-position[j][0])**2+
            (position[k][1]-position[j][1])**2))
    dneigh=np.amin(dneighp)
    return dneigh
```

Listing B.6: To calculate the nearest neighbor phosphene distance defined by Abouseoul 2018

```
def pixel_size(Nloc,positions):
    dneight=0
    for j in range(Nloc):
      dneigh= dneigh_function(j, Nloc, positions)
      dneight+=dneigh
    Psize=(1/Nloc)*dneight
    return Psize
```

Listing B.7: To calculate the Pixel size for the visual error formulation

```
def visual_error(matrixObj, matrixRes, Tres, thres):
    TvisualError=0
    Nact,positions_toactivate,Undesiredpositions =count_ones(matrixObj) # returns the number of locations that should be excited and their corresponding position
    positions=function_positions(matrixObj) # returns an array of positions for every location
    #Now I want to calculate the number of undesired activated locations
    # Calculate pixel-wise absolute differences and count non-zero values
    Nun,undesired_activated_positions=undesired_activations(matrixObj,matrixRes)
    Nloc=matrixObj.shape[0]*matrixObj.shape[1]#number of brain locations that are to be controlled
    Psz=pixel_size(Nloc,positions)
    if matrixObj.shape != matrixRes.shape:
        raise ValueError("Matrices must have the same shape.")

    for j in range(Nun):
        visperror=[]#equivalent to the delta ji in the paper equation 3
        for i in range(Nact):
            visperror.append(math.sqrt((positions_toactivate[i][0]-
            undesired_activated_positions[j][0])**2+
            (positions_toactivate[i][1]-
            undesired_activated_positions[j][1])**2)/Psz)
            element=undesired_activated_positions[j]
            visualErrorPixel=np.amin(visperror)*(Tres[element]-thres)
        TvisualError+=visualErrorPixel

    return TvisualError
```

Listing B.8: Visual Error

```
def objective_function (variables,Pobj):
    a1, a2, a3, a4, a5, a6, a7, a8, a9 = variables
    #Total transfer matrix
    Tres= a1*T1+a2*T2+a3*T3+a4*T4+a5*T5+a6*T6+a7*T7+a8*T8+a9*T9
    #Create a new matrix in which I will add a one at every point were the AF surpasses the activating threshold while add a 0 at every point where the the AF is below non excited threshold
    TharsOn=3*(10**5) #nA/cm2
    Pres= activatedpattern(Tres,ThersOn)
    # Once I get the activated patterned matrix try to get visual error
    VisualError = visual_error(Pobj, Pres,Tres,ThersOn)
    print("Visual error:", VisualError)
    return VisualError
```

Listing B.9: Objective function

Appendix C

Electric potential distributions for individual electrodes in the MEA, in a 2D cut plane at 100 μm above the tips of the electrodes from the 3D COMSOL model

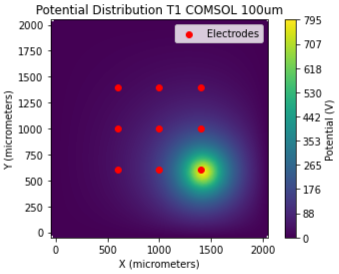


Figure C.1: Electric potential field for electrode 1

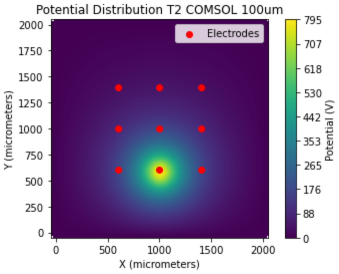


Figure C.2: Electric potential field for electrode 2

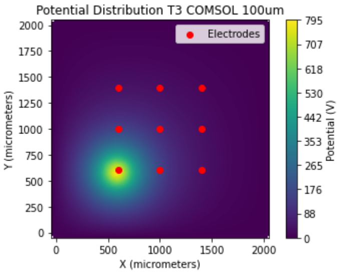


Figure C.3: Electric potential field for electrode 3

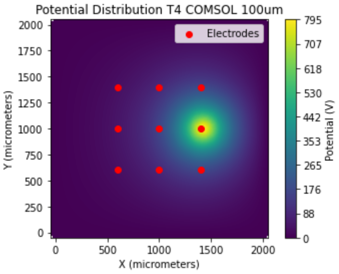


Figure C.4: Electric potential field for electrode 4

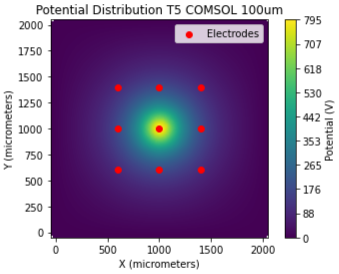


Figure C.5: Electric potential field for electrode 5

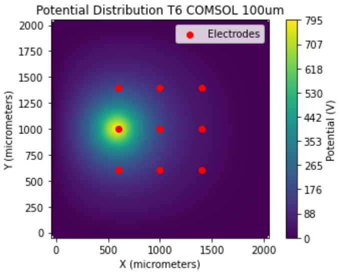


Figure C.6: Electric potential field for electrode 6

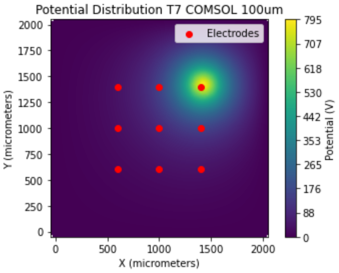


Figure C.7: Electric potential field for electrode 7

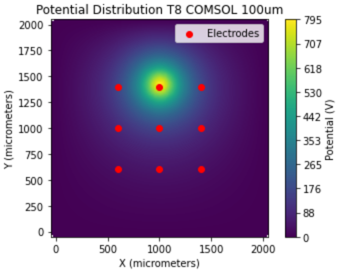


Figure C.8: Electric potential field for electrode 8

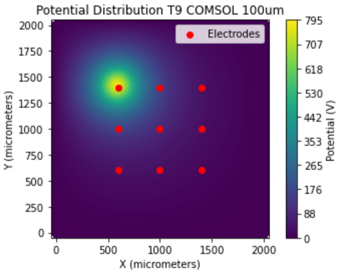


Figure C.9: Electric potential field for electrode 9

Figure C.10: Generated electric potential fields for each electrode.Data extracted from a 2D cut plane at 100µm above the tips of the electrodes from the 3D COMSOL model

Appendix D

Electric potential distributions for individual electrodes in the MEA, in a 2D cut plane at 1μm above the tips of the electrodes from the 3D COMSOL model

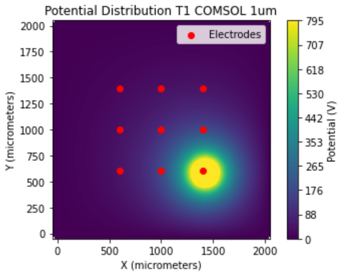


Figure D.1: Electric potential field for electrode 1

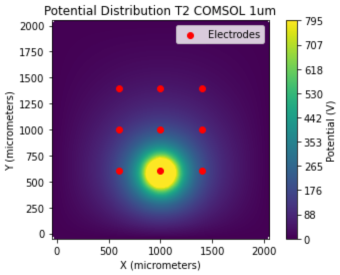


Figure D.2: Electric potential field for electrode 2

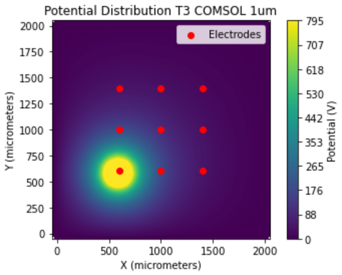


Figure D.3: Electric potential field for electrode 3

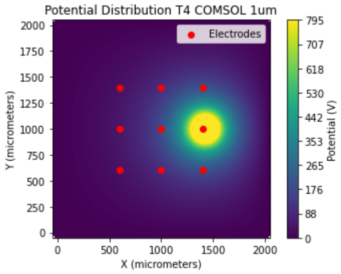


Figure D.4: Electric potential field for electrode 4

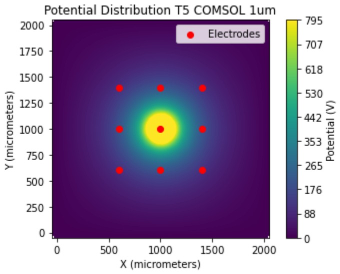


Figure D.5: Electric potential field for electrode 5

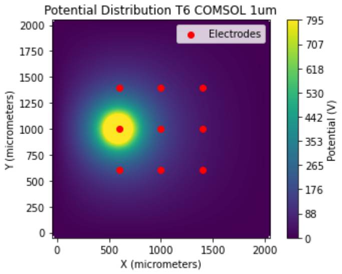


Figure D.6: Electric potential field for electrode 6

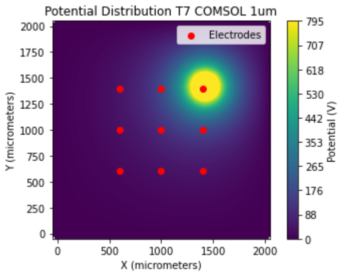


Figure D.7: Electric potential field for electrode 7

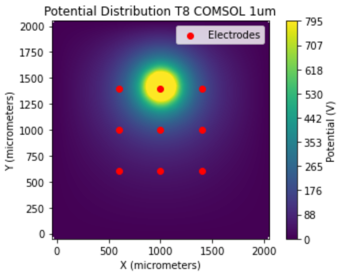


Figure D.8: Electric potential field for electrode 8

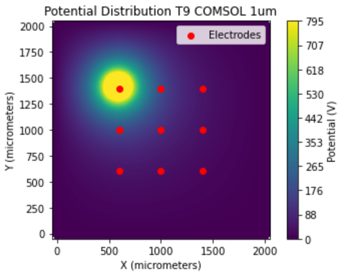


Figure D.9: Electric potential field for electrode 9

Figure D.10: Generated electric potential fields for each electrode. Data extracted from a 2D cut plane at 1μm above the tips of the electrodes from the 3D COMSOL model

Appendix E

Electric potential distributions for individual electrodes in the MEA, extracted using Green functions

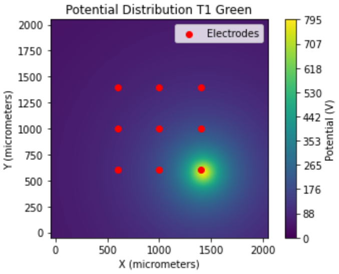


Figure E.1: Electric potential field for electrode 1

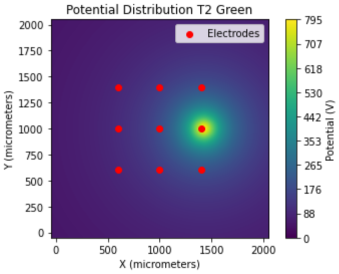


Figure E.2: Electric potential field for electrode 2

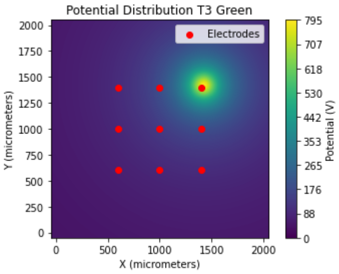


Figure E.3: Electric potential field for electrode 3

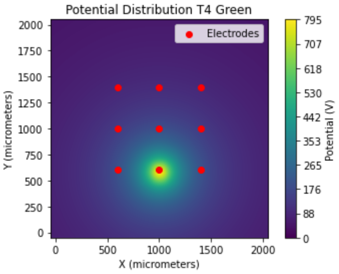


Figure E.4: Electric potential field for electrode 4

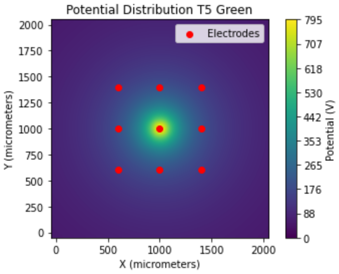


Figure E.5: Electric potential field for electrode 5

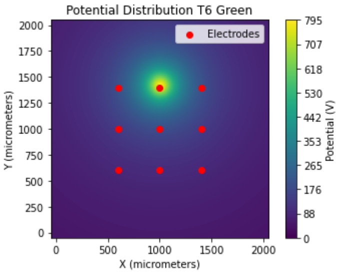


Figure E.6: Electric potential field for electrode 6

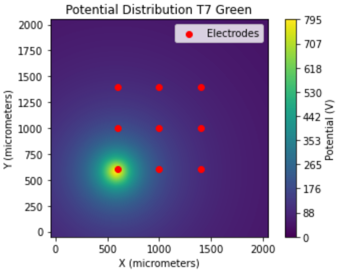


Figure E.7: Electric potential field for electrode 7

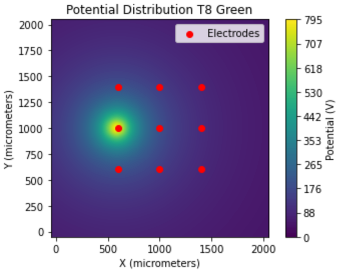


Figure E.8: Electric potential field for electrode 8

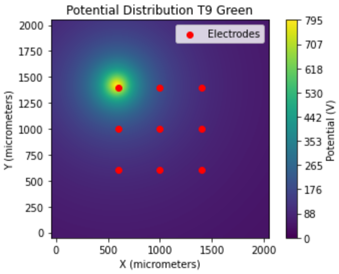
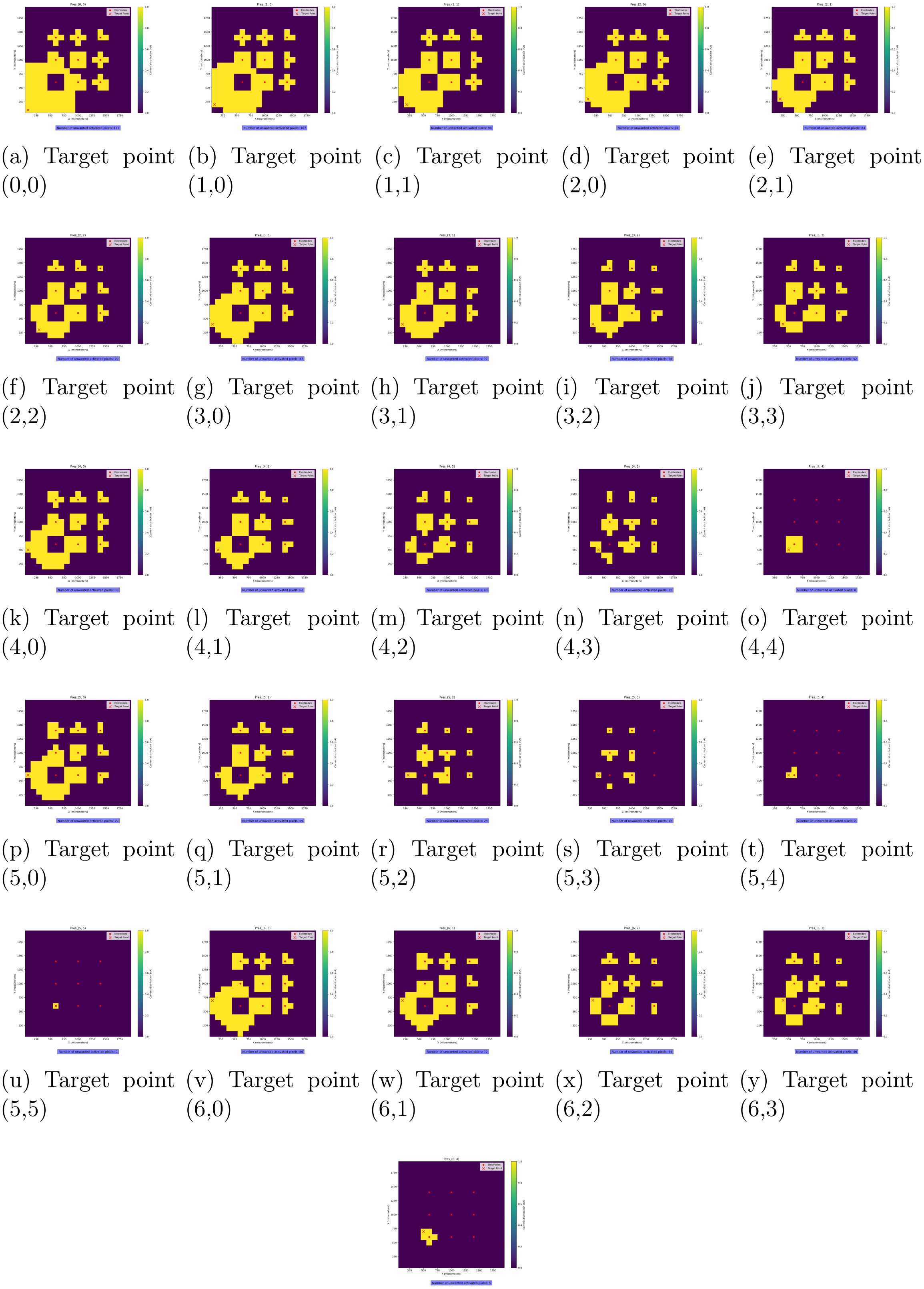


Figure E.9: Electric potential field for electrode 9

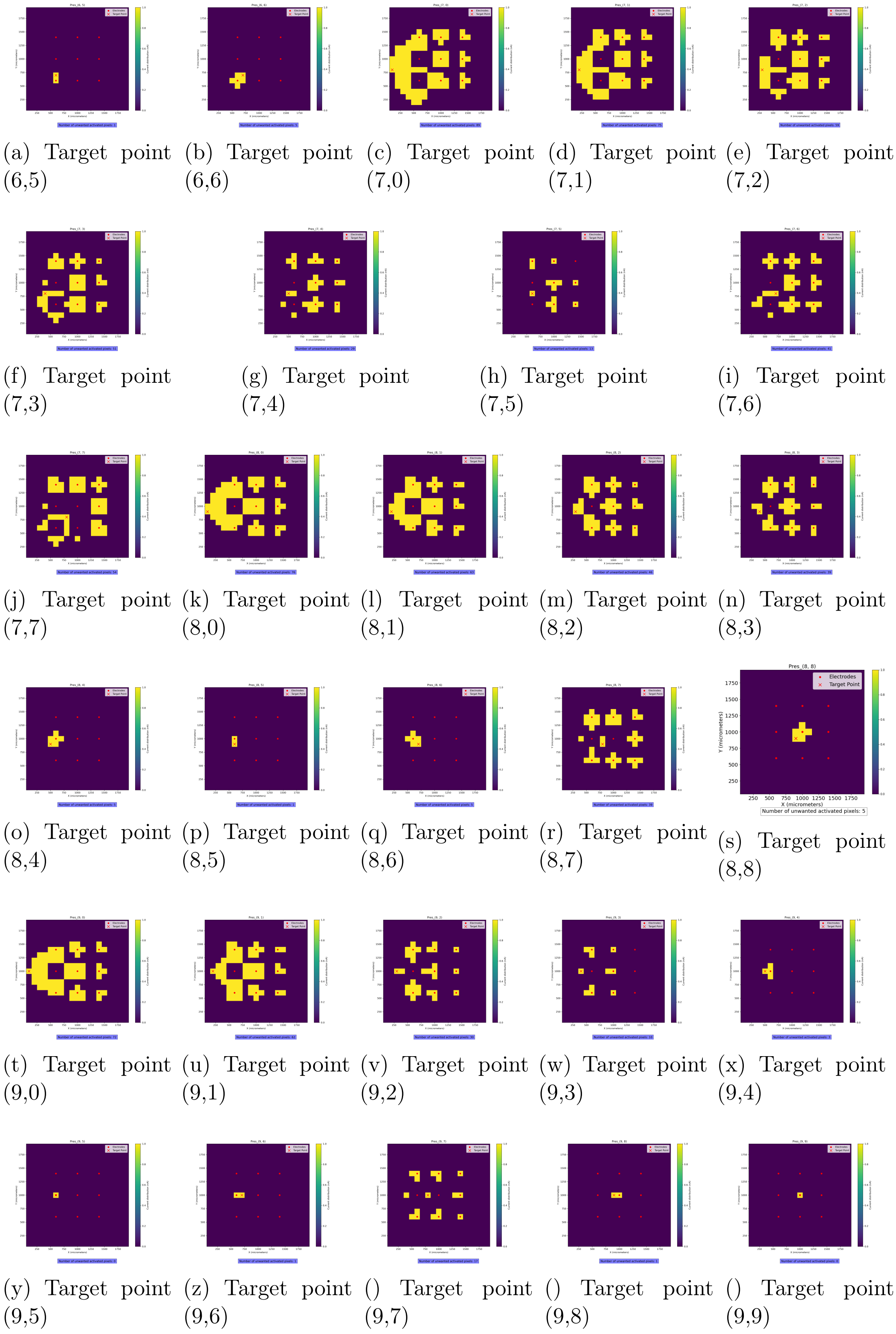
Figure E.10: Generated electric potential fields for each electrode.

Appendix F

Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from Green functions



(a) Target point (0,0)
(b) Target point (1,0)
(c) Target point (1,1)
(d) Target point (2,0)
(e) Target point (2,1)
(f) Target point (2,2)
(g) Target point (3,0)
(h) Target point (3,1)
(i) Target point (3,2)
(j) Target point (3,3)
(k) Target point (4,0)
(l) Target point (4,1)
(m) Target point (4,2)
(n) Target point (4,3)
(o) Target point (4,4)
(p) Target point (5,0)
(q) Target point (5,1)
(r) Target point (5,2)
(s) Target point (5,3)
(t) Target point (5,4)
(u) Target point (5,5)
(v) Target point (6,0)
(w) Target point (6,1)
(x) Target point (6,2)
(y) Target point (6,3)
(z) Target point (6,4)



(a) Target point (6,5)

(b) Target point (6,6)

(c) Target point (7,0)

(d) Target point (7,1)

(e) Target point (7,2)

(f) Target point (7,3)

(g) Target point (7,4)

(h) Target point (7,5)

(i) Target point (7,6)

(j) Target point (7,7)

(k) Target point (8,0)

(l) Target point (8,1)

(m) Target point (8,2)

(n) Target point (8,3)

(o) Target point (8,4)

(p) Target point (8,5)

(q) Target point (8,6)

(r) Target point (8,7)

(s) Target point (8,8)

(t) Target point (9,0)

(u) Target point (9,1)

(v) Target point (9,2)

(w) Target point (9,3)

(x) Target point (9,4)

(y) Target point (9,5)

(z) Target point (9,6)

() Target point (9,7)

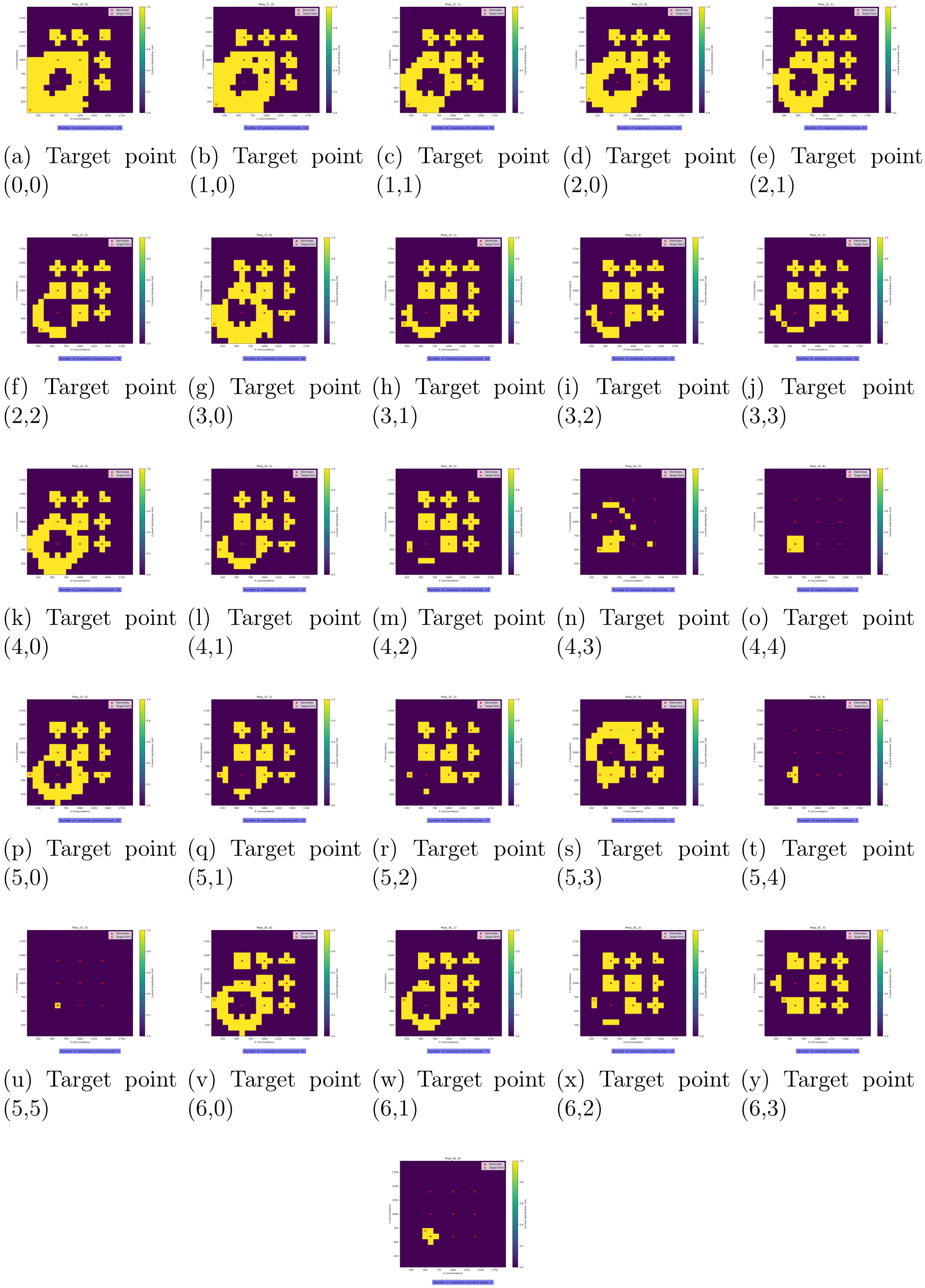
() Target point (9,8)

() Target point (9,9)

Figure F.2: Activated patterns algorithm output for Green function input data for every pixel in Fig. 5.5a.

Appendix G

Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 100μm



(a) Target point (0,0)

(b) Target point (1,0)

(c) Target point (1,1)

(d) Target point (2,0)

(e) Target point (2,1)

(f) Target point (2,2)

(g) Target point (3,0)

(h) Target point (3,1)

(i) Target point (3,2)

(j) Target point (3,3)

(k) Target point (4,0)

(l) Target point (4,1)

(m) Target point (4,2)

(n) Target point (4,3)

(o) Target point (4,4)

(p) Target point (5,0)

(q) Target point (5,1)

(r) Target point (5,2)

(s) Target point (5,3)

(t) Target point (5,4)

(u) Target point (5,5)

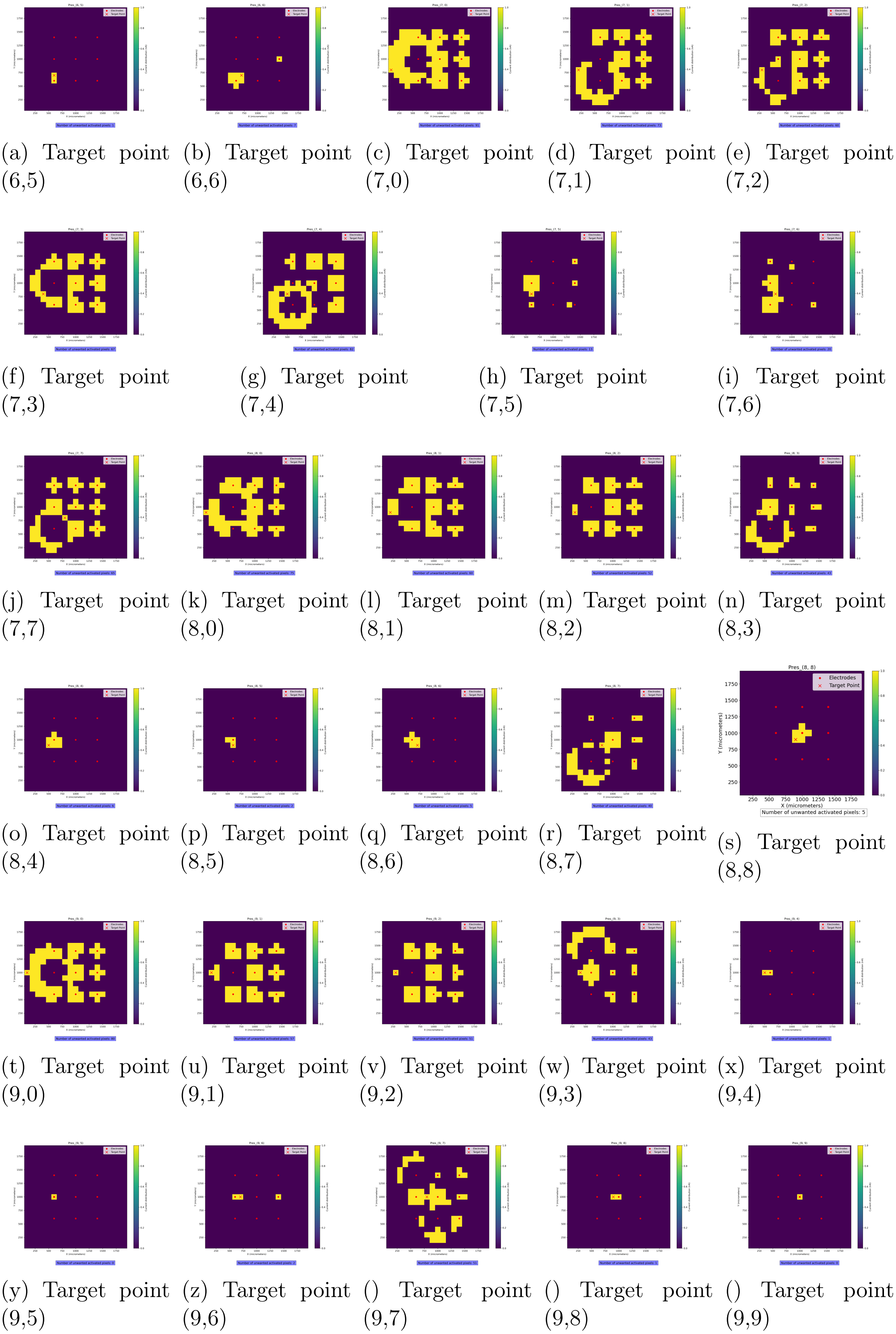
(v) Target point (6,0)

(w) Target point (6,1)

(x) Target point (6,2)

(y) Target point (6,3)

(z) Target point (6,4)



(a) Target point (6,5)

(b) Target point (6,6)

(c) Target point (7,0)

(d) Target point (7,1)

(e) Target point (7,2)

(f) Target point (7,3)

(g) Target point (7,4)

(h) Target point (7,5)

(i) Target point (7,6)

(j) Target point (7,7)

(k) Target point (8,0)

(l) Target point (8,1)

(m) Target point (8,2)

(n) Target point (8,3)

(o) Target point (8,4)

(p) Target point (8,5)

(q) Target point (8,6)

(r) Target point (8,7)

(s) Target point (8,8)

(t) Target point (9,0)

(u) Target point (9,1)

(v) Target point (9,2)

(w) Target point (9,3)

(x) Target point (9,4)

(y) Target point (9,5)

(z) Target point (9,6)

() Target point (9,7)

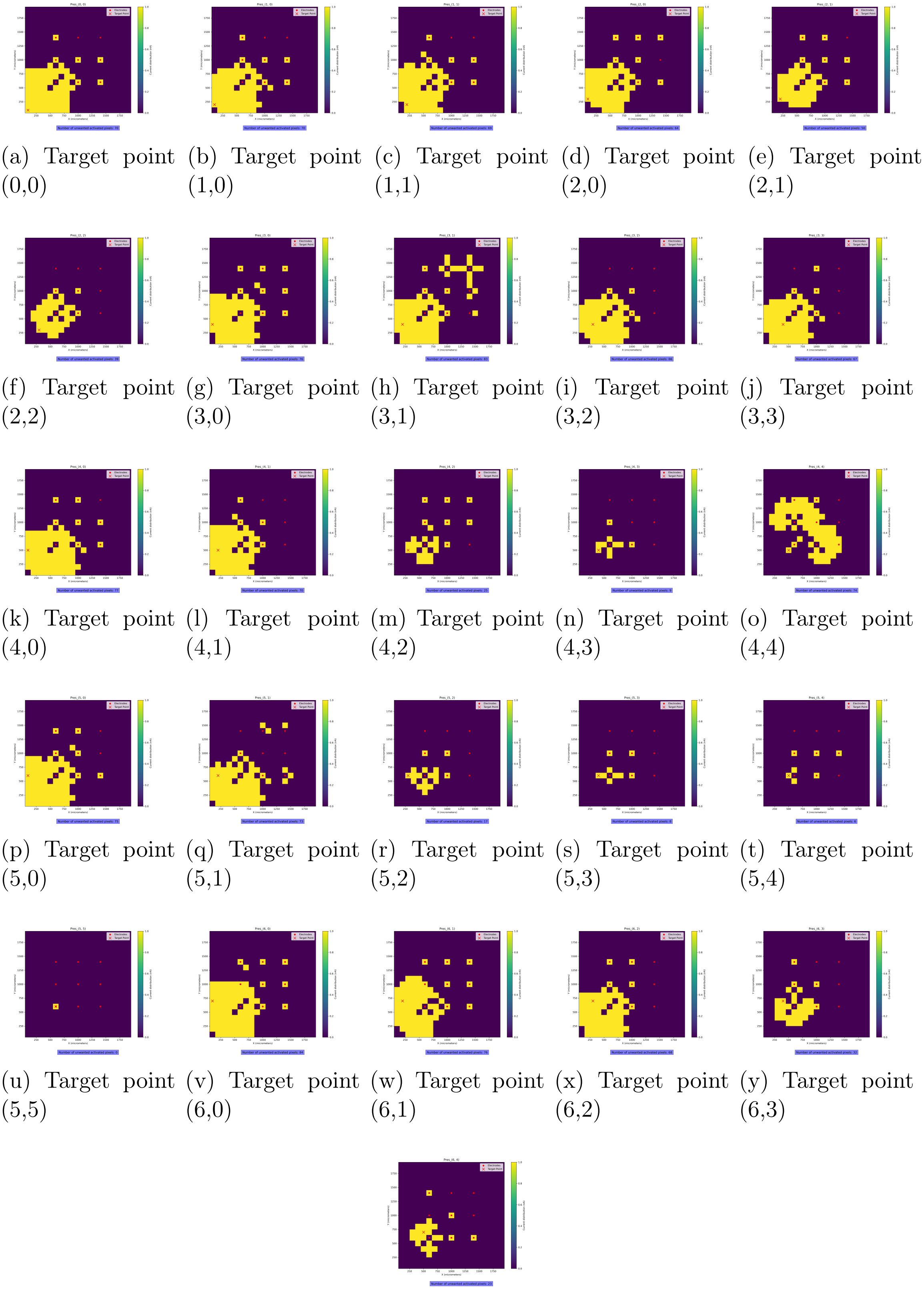
() Target point (9,8)

() Target point (9,9)

Figure G.2: Activated patterns algorithm output for COMSOL 100μm input data for every pixel in Fig. 5.5a.

Appendix H

Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 1μm



(a) Target point (0,0)

(b) Target point (1,0)

(c) Target point (1,1)

(d) Target point (2,0)

(e) Target point (2,1)

(f) Target point (2,2)

(g) Target point (3,0)

(h) Target point (3,1)

(i) Target point (3,2)

(j) Target point (3,3)

(k) Target point (4,0)

(l) Target point (4,1)

(m) Target point (4,2)

(n) Target point (4,3)

(o) Target point (4,4)

(p) Target point (5,0)

(q) Target point (5,1)

(r) Target point (5,2)

(s) Target point (5,3)

(t) Target point (5,4)

(u) Target point (5,5)

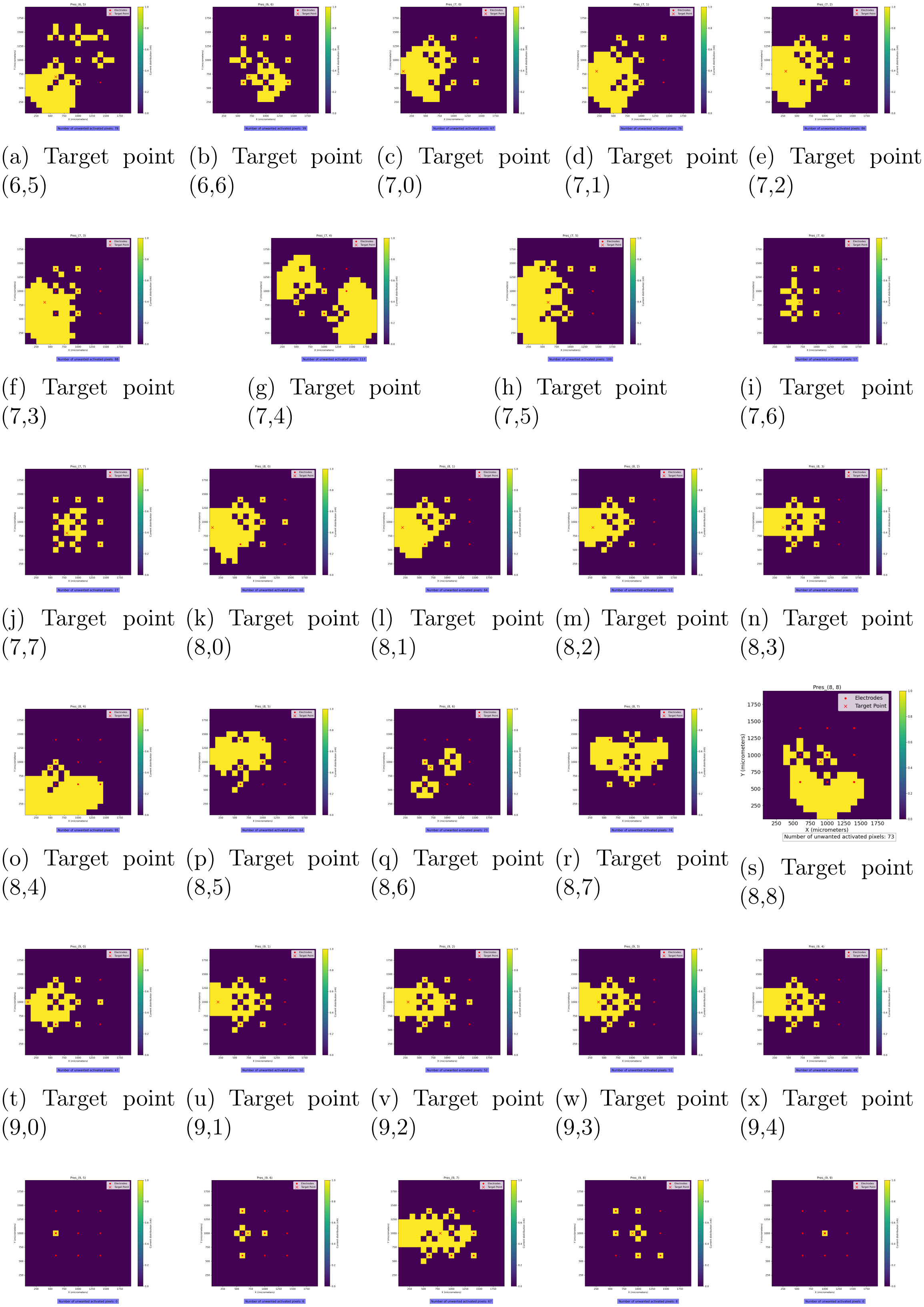
(v) Target point (6,0)

(w) Target point (6,1)

(x) Target point (6,2)

(y) Target point (6,3)

(z) Target point (6,4)



(a) Target point (6,5) (b) Target point (6,6) (c) Target point (7,0) (d) Target point (7,1) (e) Target point (7,2)

(f) Target point (7,3) (g) Target point (7,4) (h) Target point (7,5) (i) Target point (7,6)

(j) Target point (7,7) (k) Target point (8,0) (l) Target point (8,1) (m) Target point (8,2) (n) Target point (8,3)

(o) Target point (8,4) (p) Target point (8,5) (q) Target point (8,6) (r) Target point (8,7) (s) Target point (8,8)

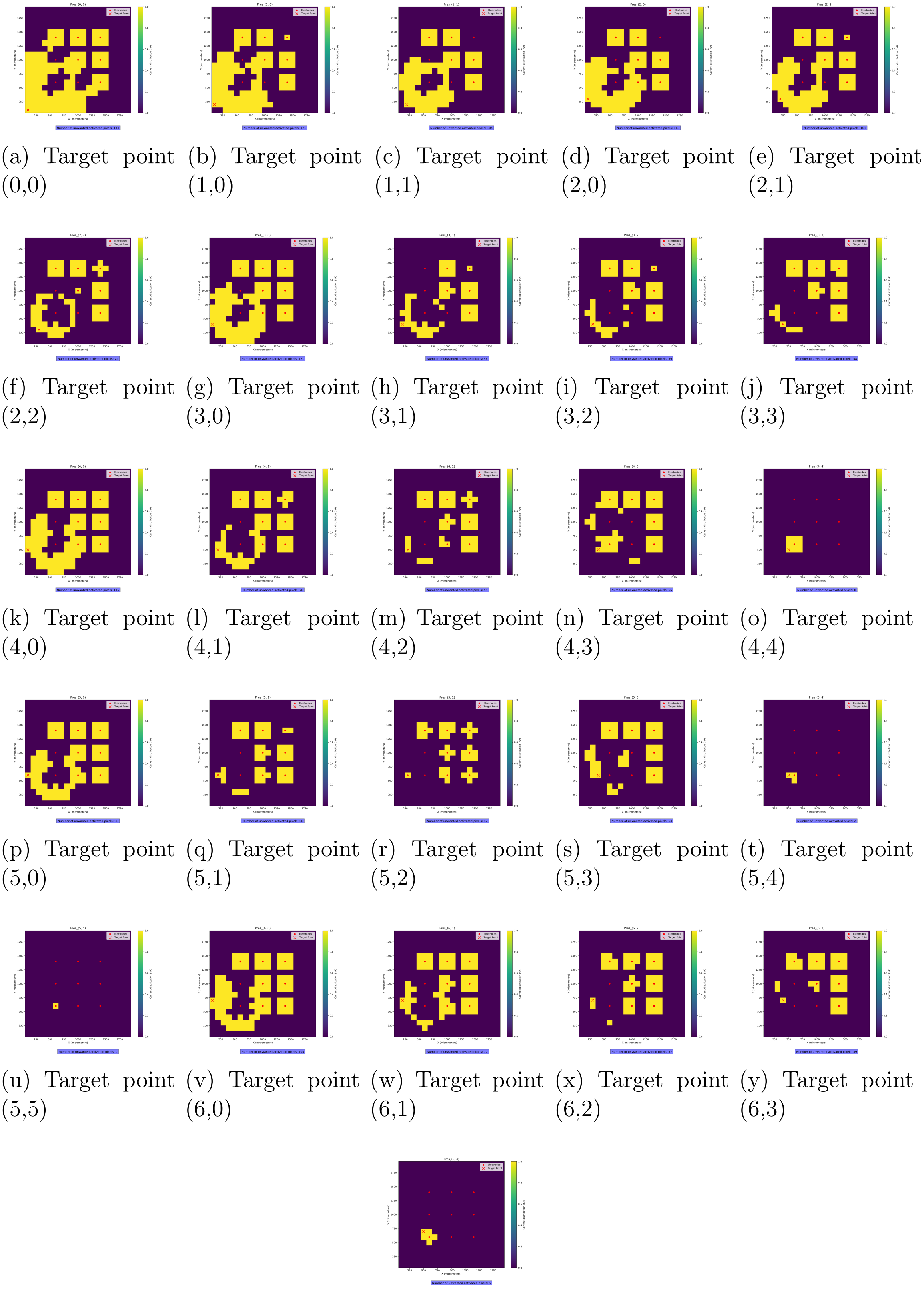
(t) Target point (9,0) (u) Target point (9,1) (v) Target point (9,2) (w) Target point (9,3) (x) Target point (9,4)

(y) Target point (9,5) (z) Target point (9,6) () Target point (9,7) () Target point (9,8) () Target point (9,9)

Figure H.2: Activated patterns algorithm output for COMSOL 1µm input data for every pixel in Fig. 5.5a.

Appendix I

Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from COMSOL 100μm and intensity minimization



(a) Target point (0,0)

(b) Target point (1,0)

(c) Target point (1,1)

(d) Target point (2,0)

(e) Target point (2,1)

(f) Target point (2,2)

(g) Target point (3,0)

(h) Target point (3,1)

(i) Target point (3,2)

(j) Target point (3,3)

(k) Target point (4,0)

(l) Target point (4,1)

(m) Target point (4,2)

(n) Target point (4,3)

(o) Target point (4,4)

(p) Target point (5,0)

(q) Target point (5,1)

(r) Target point (5,2)

(s) Target point (5,3)

(t) Target point (5,4)

(u) Target point (5,5)

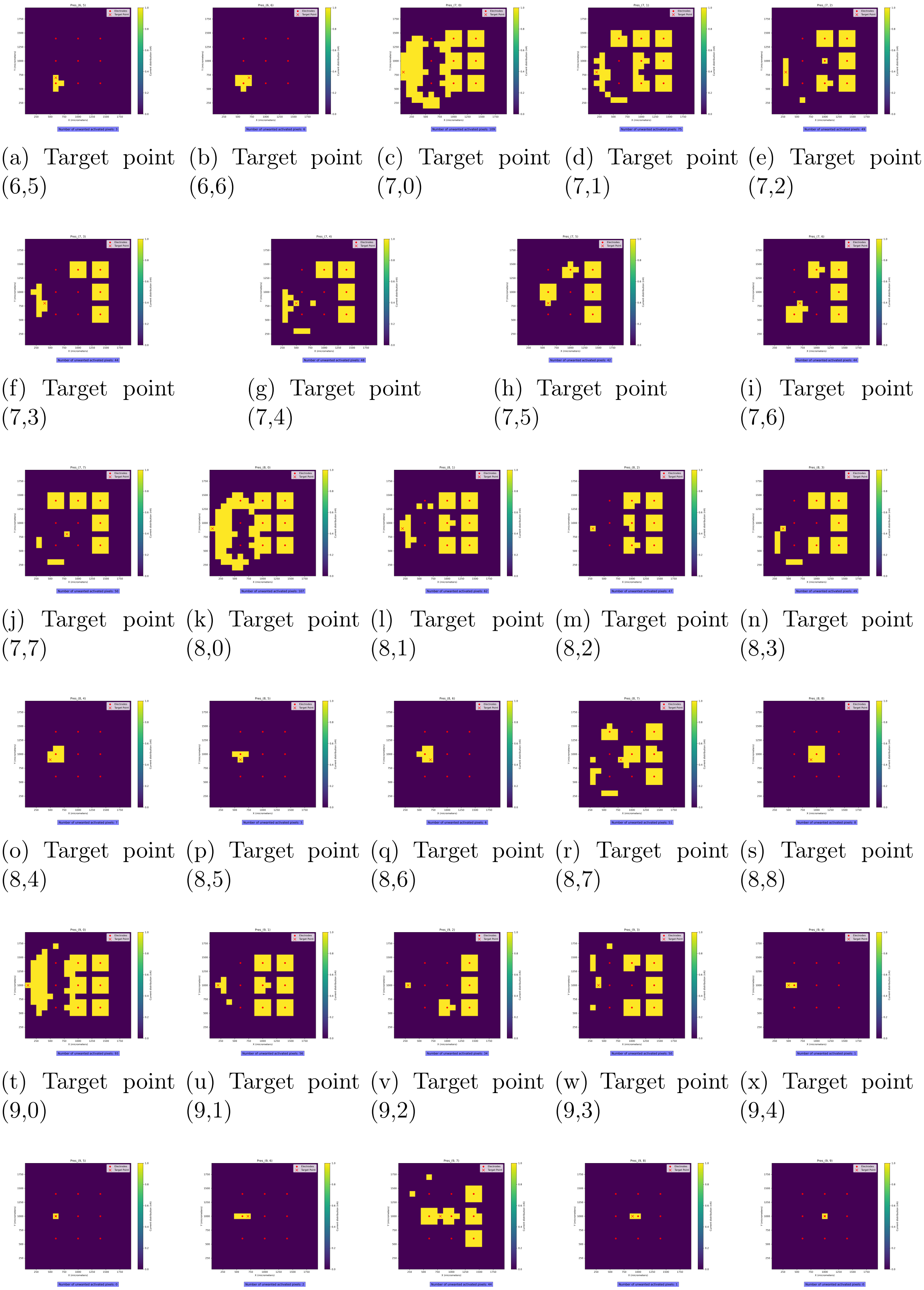
(v) Target point (6,0)

(w) Target point (6,1)

(x) Target point (6,2)

(y) Target point (6,3)

(z) Target point (6,4)



(a) Target point (6,5)

(b) Target point (6,6)

(c) Target point (7,0)

(d) Target point (7,1)

(e) Target point (7,2)

(f) Target point (7,3)

(g) Target point (7,4)

(h) Target point (7,5)

(i) Target point (7,6)

(j) Target point (7,7)

(k) Target point (8,0)

(l) Target point (8,1)

(m) Target point (8,2)

(n) Target point (8,3)

(o) Target point (8,4)

(p) Target point (8,5)

(q) Target point (8,6)

(r) Target point (8,7)

(s) Target point (8,8)

(t) Target point (9,0)

(u) Target point (9,1)

(v) Target point (9,2)

(w) Target point (9,3)

(x) Target point (9,4)

(y) Target point (9,5)

(z) Target point (9,6)

() Target point (9,7)

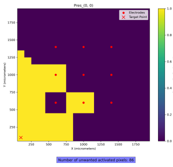
() Target point (9,8)

() Target point (9,9)

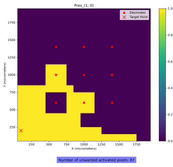
Figure I.2: Activated patterns algorithm output for COMSOL 100μm input data for every pixel in Fig. 5.5a including intensity minimization.

Appendix J

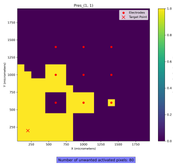
Output activated patterns for target point contained in original triangle introduced in Fig. 5.5a with input data from Green Function and exponential objective function



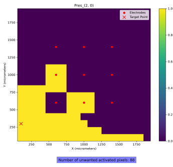
(a) Target point (0,0)



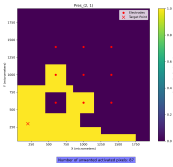
(b) Target point (1,0)



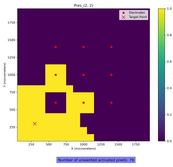
(c) Target point (1,1)



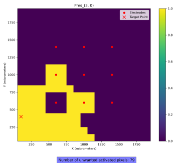
(d) Target point (2,0)



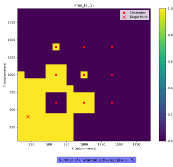
(e) Target point (2,1)



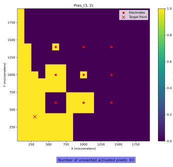
(f) Target point (2,2)



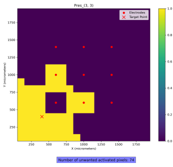
(g) Target point (3,0)



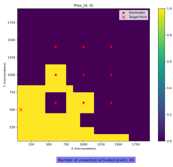
(h) Target point (3,1)



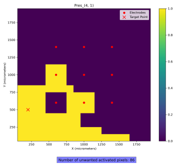
(i) Target point (3,2)



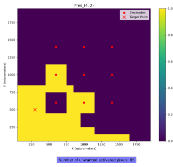
(j) Target point (3,3)



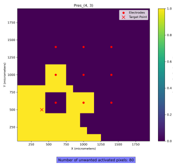
(k) Target point (4,0)



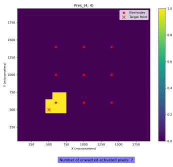
(l) Target point (4,1)



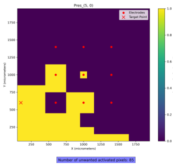
(m) Target point (4,2)



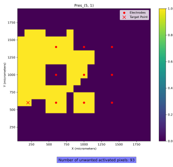
(n) Target point (4,3)



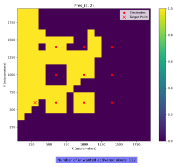
(o) Target point (4,4)



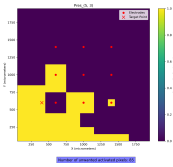
(p) Target point (5,0)



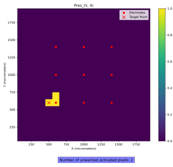
(q) Target point (5,1)



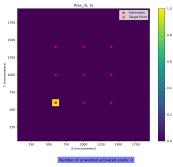
(r) Target point (5,2)



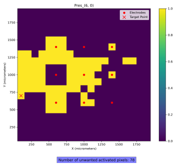
(s) Target point (5,3)



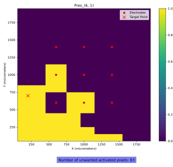
(t) Target point (5,4)



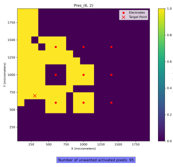
(u) Target point (5,5)



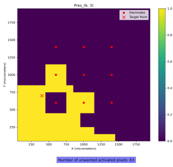
(v) Target point (6,0)



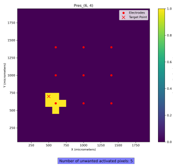
(w) Target point (6,1)



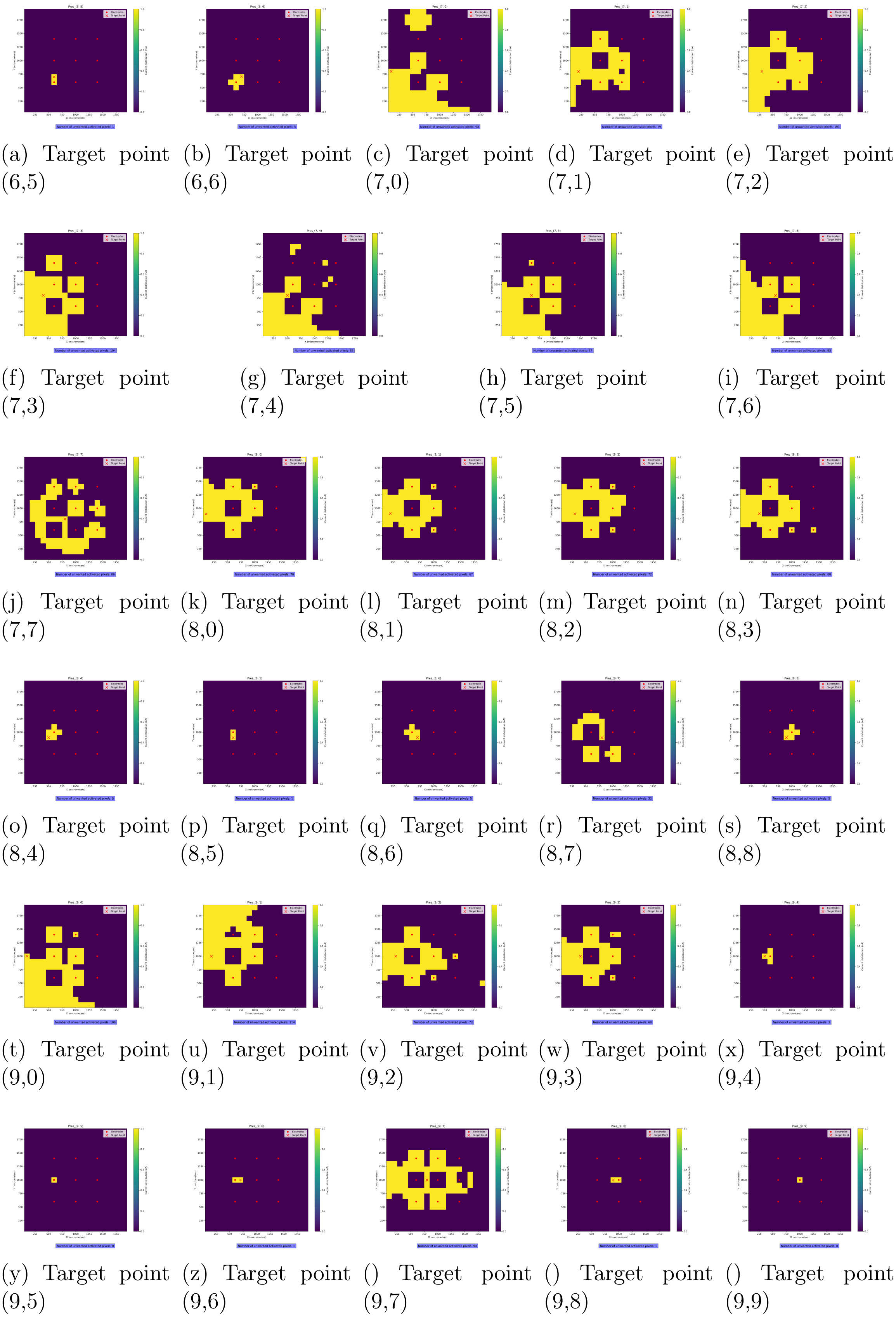
(x) Target point (6,2)



(y) Target point (6,3)



(z) Target point (6,4)



(a) Target point (6,5) (b) Target point (6,6) (c) Target point (7,0) (d) Target point (7,1) (e) Target point (7,2)

(f) Target point (7,3) (g) Target point (7,4) (h) Target point (7,5) (i) Target point (7,6)

(j) Target point (7,7) (k) Target point (8,0) (l) Target point (8,1) (m) Target point (8,2) (n) Target point (8,3)

(o) Target point (8,4) (p) Target point (8,5) (q) Target point (8,6) (r) Target point (8,7) (s) Target point (8,8)

(t) Target point (9,0) (u) Target point (9,1) (v) Target point (9,2) (w) Target point (9,3) (x) Target point (9,4)

(y) Target point (9,5) (z) Target point (9,6) () Target point (9,7) () Target point (9,8) () Target point (9,9)

Figure J.2: Activated patterns algorithm output for Green Function input data for every pixel in Fig. 5.5a including exponential objective function.