Stefan Bernert, BSc

Functional & Physiological Characterization of Neonatal Rat Brain Organotypic Slices with Multi Electrode Arrays

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

Ass.Prof.Dipl.-Ing.Dr.techn. Theresa Rienmüller

Institute for Health Care Engineering
Head: Univ.-Prof.Dipl.-Ing.Dr.techn. Christian Baumgartner

Graz, January 2020

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

____________________ ________________________________

Date Signature

Thanks

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Frau Prof. Theresa Rienmüller, die meine Masterarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die Zeit, egal ob am Institut oder während ihres Auslandsaufenthaltes, möchte ich mich herzlich bedanken.

Ich bedanke mich bei meinen Studienkollegen Sara und Daniel, die mir mit Hilfsbereitschaft und Ideen zur Seite standen und maßgeblich dazu beigetragen haben, dass diese Masterarbeit fertig geworden ist.

Ebenfalls möchte ich mich bei Josie bedanken, die nicht nur meine Masterarbeit, sondern auch schon meine Bachelorarbeit korrekturgelesen hat. Sowie bei Gert, der mir in vielen Fragen immer wieder Ratschläge gab.

Außerdem möchte ich David und Helmut für das Korrekturlesen meiner Masterarbeit danken.

Abschließend möchte ich mich bei meiner Mutter Irmgard, meiner Oma und Peter bedanken, die mir mein Studium durch ihre Unterstützung ermöglicht haben und stets für mich da waren. Aber auch bei meinem Vater Kurt und meiner Schwester Daniela, die den Weg leider nicht bis zum Ende mitgehen konnten.

Stefan Bernert

Abstract

Organotypic brain slice cultures can be used to study electrophysiological and cellular processes of the brain in vitro closely mimicking in-vivo like situations. The focus of this work is on the examination of the hippocampus, a major component of the human brain that plays an important role in the consolidation of information from short-term to long-term memory. Two different types of high density multi electrode arrays (MEA) with either 60 or 4096 electrodes are used to examine electrophysiological processes of rat hippocampal organotypic slices and dissociated neuronal cultures. The recorded neuronal activity is visualized as heatmap and subsequently overlayed with microscopic images of the respective neuronal culture. The developed program offers the unique possibility to combine the physiological structure and the neuronal activity of the samples, although the visibility of the dissociated neurons could be further increased using fluorescence microscopy. A further validation of the results using different samples will be required.

Keywords: CMOS-MEA, Organotypic Slices, Dissociated Neurons, Mapping, Overlay

Zusammenfassung

Organotypische Hirnschnittkulturen können für Untersuchungen elektrophysiologischer und zellulärer Prozesse des Gehirnes verwendet werden. Der Fokus dieser Arbeit liegt auf der Untersuchung des Hippocampus, einem Hauptbestandteil des menschlichen Gehirns, der eine wichtige Rolle bei der Verarbeitung von Informationen vom Kurzzeit- zum Langzeitgedächtnis spielt. Zwei verschiedene Arten von Multi Elektroden Arrays (MEA) mit 60 oder 4096 Elektroden werden verwendet, um elektrophysiologische Prozesse von Hippocampus Schnitten der Ratte und dissoziierten neuronalen Kulturen zu untersuchen. Die aufgezeichnete neuronale Aktivität wird als Heatmap visualisiert und anschließend mit mikroskopischen Bildern der jeweiligen neuronalen Kultur überlagert. Das entwickelte Programm bietet die einzigartige Möglichkeit, die physiologische Struktur und die neuronale Aktivität der Proben zu kombinieren, die Sichtbarkeit der dissoziierten Neuronen könnte weiter erhöht werden mithilfe der Fluoreszenzmikroskopie. Eine weitere Validierung der Ergebnisse anhand verschiedener Proben ist erforderlich.

Stichwörter: CMOS-MEA, organotypischen Schnitte, dissoziierte Neuronen, Mapping, Überlappung

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the fundamentals applied throughout the thesis, provides an overview and an conveys understanding which is necessary to follow the presented methods and results. For further information on this topic please consult the list of references.

1.1 Rat Brain Hippocampus

The hippocampus got its name due to the similarity to the shape of a sea horse. The structure is one of the best characterized brain tissues. Anatomical and physiological investigations can be done perfectly due to the well structured neurons located in this area, which is responsible for some major cognitive functions. Studies assert that the hippocampus is mainly involved in aspects of learning or memory. The structure displayed in Figure 1.1, can be divided into four regions (CA1) to (CA4) called Cornu Ammon or Ammon's horn and their neuronal pathways. (CA2) is so unclear and small that it is often ignored, this also applies to (CA4). Dentate Gyrus (DG), Subiculum (S) and Enthorhinal Cortex (EC) are the other important hioppocampal regions. The Entorhinal cortex generates the main input for the Hippocampus and Dentate Gyrus, some other regions are also important but not large enough for projections. The signals enters the region by the perforant path (pp), the (DG) and runs up along the mossy fibre (mf) and Schaffer collaterals (sc) and can leave the tissue by alveus (a) or fimbria (f). [25]; [2]

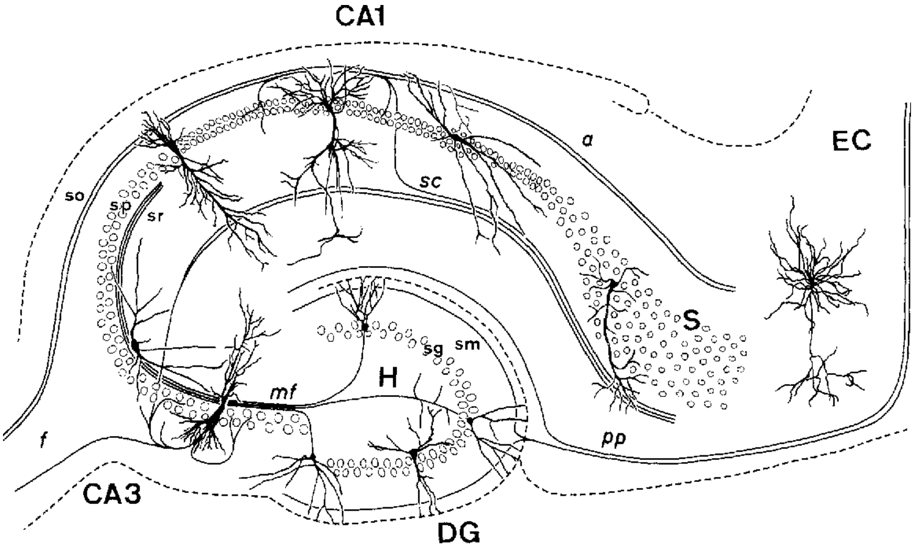


Figure 1.1: Hippocampus structure with the main regions and the pathways between them. (DG) Dentate Gyrus; (H) Hilus; (CA1-3) Cornu Ammon; (EC) Enthorhinal Cortex; (S) Subiculum; (so) Stratum Oriens; (sp) Stratum Pyramidale; (sm) Stratum Moleculare; (sr) Stratum Radiatum; (sg) Stratum Granulosum. [2]

The basic concept of the hippocampus region is described in Figure 1.2. The closed-loop concept is taken into account when the major regions are observed, the projection system connects one major output back to an input. The smallest circuit needs at least three synapses to complete the loop, the (ECII) projects to (CA1), from this part to (ECVI) and back to (ECII). The projections are described very well, the input starts at the Entorhinal Cortex. This region sends the signal mainly to Dentate Gyrus, but it is also connected to (CA3) and (CA1). The path to (CA3) is very pronounced and there are many more path synapses, than possible inputs. Dentate Gyrus is the only region which sends the projections only to one other region, all other parts are sending fibers to two or more regions. (CA3) sends signals mainly to (CA1) or Subiculum, similarly, (CA1) projects towards Subiculum and the Entorhinal complex. Observations show that the result is a series of overlapping loops and not only a single one. [25]

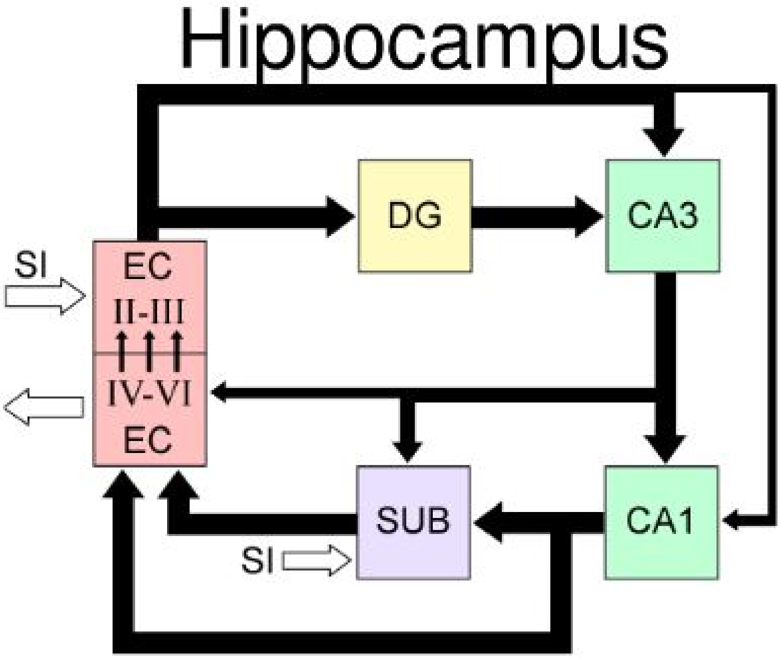


Figure 1.2: Hippocampus structure diagram, connections between the areas and information propagation, thicker lines show higher projections from signals to this synapses. (DG) Dentate Gyrus; (CA1-3) Cornu Ammon; (EC) Enthorhinal Cortex; (Sub) Subiculum; (SI) Sensory Inputs; [8]

1.1.1 Hippocampus Slice Preparation

Brains of five day old prenatal rats were used for this experiment. For multielectrode array measurements parts of the hippocampus or visual cortex might be used, both can transversely be cut in slices and provide an intact neurons system. This procedure ensures that spontaneous and stimulated activities can be measured. The brain slices have a thickness of ∼400 µm, this results in a transparent and stable structure. After cutting the organotypic slices, the probes are left to rest on a petri dish. They are enclosed in culture medium and placed in an incubator. This is also important for the following measurements, because the tissue needs some time to recover after a trauma. Although acute rat brain slices could be used, quite some time after the slices are cut, the best time for measurements is after five to six days in cultures with medium, the slices remain viable for up to two weeks, but at the end the structure dissolves.

Two possible culturing methods are currently used: Stoppini and Gähwiler cultures [7], [26]. In this thesis the approach presented by Stoppini is used and the organotypic slices are cultured on a membrane. The main focus lies on hippocampal slices. In Figure 1.3 the preparation of hippocampal and cerebellum slices is shown. In (A) the brain was dissolved from the head, afterwards the cerebellum is cut out, on the left side the remaining brain is separated through the middle. Then the midbrain is removed to uncover the hippocampus in (C). By removing the bordering part in (D) the hippocampus will be exposed. In (E) the brain is put onto the tissue chopper displayed in (1). In the presented case the device creates slices with a thickness of ∼400 µm, like shown in steps (2) and (3). The slices are separated in (F) and placed onto the petri dishes, surrounded with medium. On the right, cerebellum slice preperation is shown, starting with the cut out of the inferior colliculus in (B) at the brain stem. When only the cerebellum remains, it is clipped into slices by the tissue chopper in (C). The slices get cut along the dotted lines in (D) and the same procedure is used afterwards in (F). [27]

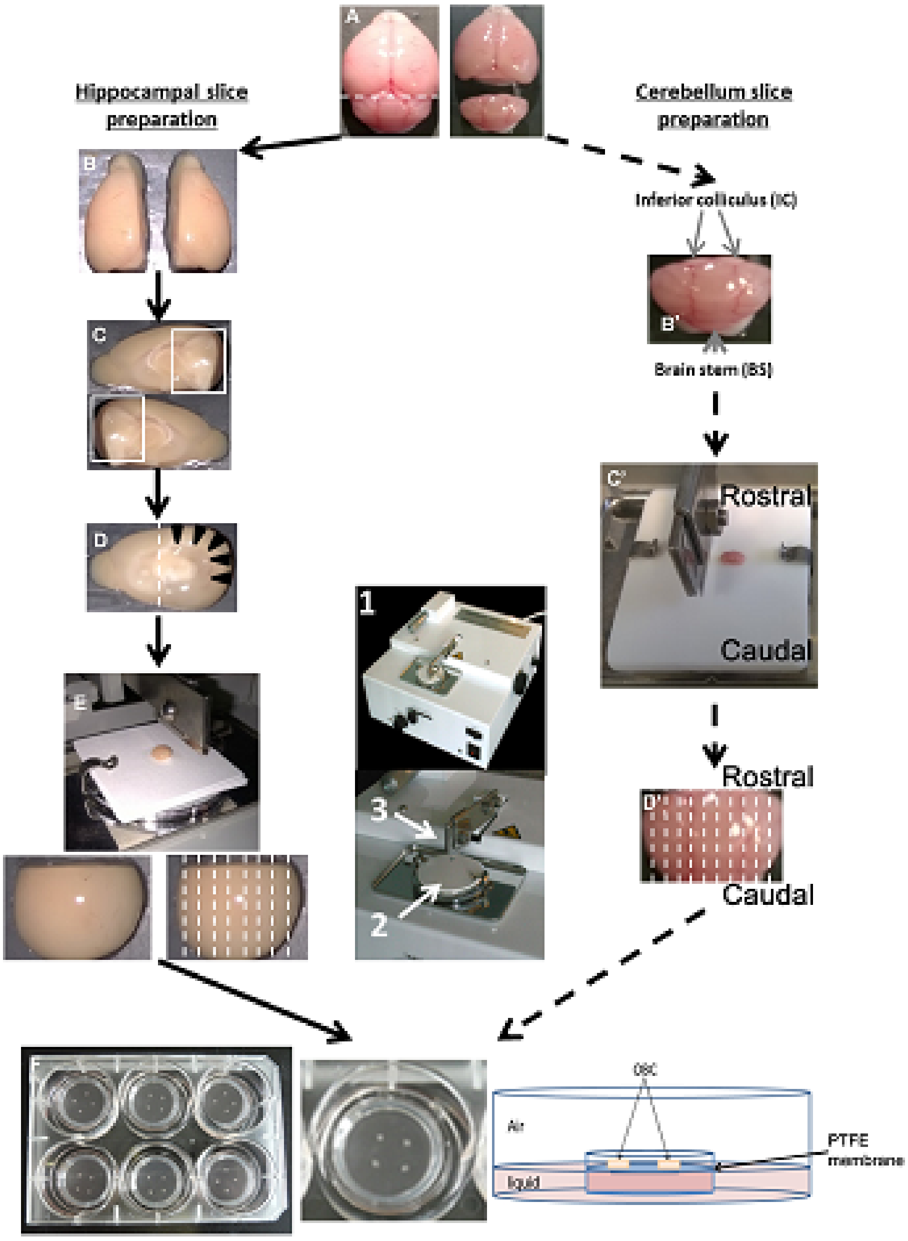


Figure 1.3: Slice preparation of hippocampus and cerebellum, from the intact brain to the slices on the petri dish. [27]

1.2 Dissociated Neurons

The brains of the rats are removed after they are euthanized with carbon dioxide. The hippocampus is dissociated in 5 ml of 0.05 % trypsin-ethylenediaminetetraacetic. This procedure takes at least 15 minutes with a constant temperature of 37 °C. Additional dissociation is achieved by using a fire-polished Pasteur pipette. Afterwards, the neurons are seeded onto a coated CMOS-MEA chip [15], the used concentration may vary from 25 000 cells/cm^2 up to 100 000 cells/cm^2. In Figure 1.4 dissociated neurons are illustrated. Lower density of neurons results in better pronounced organic connections, the space between the neurons is increased and the chance of forming neuron clusters is reduced. After 24 hours half of the medium on the membrane has to be exchanged. Following almost 13 days of constant conditions at 35.5 °C and 5 % CO_2 in the incubator and an air humidity of 95 % enough connections are formed to measure them. During the measurements a perfusion with a buffer solution is not required, because the neurons are surrounded with medium. On the one hand this is a benefit during the recording of signals, on the other hand neurons can be exhausted every time the old medium is pipetted and replaced by a new one. [6]; [20]; [15]; [3];

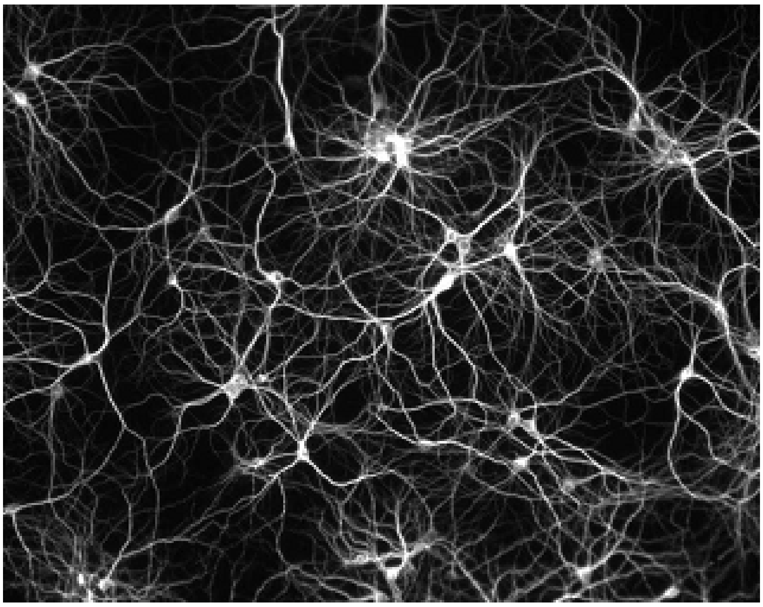


Figure 1.4: Dissociated culture extracted from hippocampal neurons. [22]

1.3 Buffer Solution

The solution is known as artificial cerebrospinal fluid (ACSF). It is necessary to use a buffer solution during the measurements otherwise the brain slices will dehydrate and die. The fluid is almost pH neutral with a pH value of 7.4, which corresponds to the pH value of brain fluid. There are several versions of ACSF, varying by only a few components and concentrations. In the following tables, some solutions are presented. Before glucose is added, the solution can be used for almost a month. Afterwards, the fluid must be consumed in one or two days. [17]

Essential for brains is the chosen solution, which depends on the age of the rats or mice. For young animals standard and HEPES ACSF is the correct choice and for aged rats choline ACSF. [9]

Table 1.1 shows only an example of one buffer solution. In Table 3.1 the used ACSF is displayed as well as one Table .1 in the appendix. The Solution has to be enriched with carbogen gas for 15 minutes, the pH values of all presented solutions are almost identical.

Chemical	**M.W.**	**Conc. [mM]**	**g/1L 10x Stock**	**g/2L 10x Stock**	**g/500mL ACSF**	**g/1L ACSF**
NaCl	58.44	136	79.48	158.95		
KCl	74.55	2.5	1.86	3.72		
$CaCl_2 \cdot 2H_2O$	147.01	2	2.94	5.88		
$MgCl_2 \cdot 6H_2O$	203.30	1.3	2.64	5.28		
HEPES	238.31	10			1.19	2.38
$Glucose \cdot H_2O$	198.17	10			0.99	1.98

Table 1.1: ACSF Buffer Solution with HEPES. [28])

1.4 Microscope

On the basis of the MEA chip, which doesn't allow light to shine through, a microscope which uses reflected-light was chosen. In this case an Olympus BX51 in Figure 1.5 was developed for electro-physiological applications and features prevention against vibrations. The device includes a video port for live tracking data on a computer and to saving images. [23]

Figure 1.5: Olympus reflected-light microscope. [24]

1.5 Camera

The used camera should employ CMOS technology to generate high resolution images, leading to a reduced time between perfusion and exposure. The chips also feature a relatively dark array in which neurons cannot be seen well. To make them visible fluorescence would be needed. In fact the camera should also support this and have a high sensitivity at the same wavelength as the emitted material. The sensitivity should match the emitted light from the microscope. [11]

1.6 Fluorescence

Fluorescence is based on the impact of absorbed and emitted light on organic or inorganic material and is a well-established physical phenomenon. The process of light emission and absorption is almost concurrent due to a brief delay among absorption and emission, in a duration of less than a microsecond. The phenomenon is called phosphorescence, when the emission of light exists longer, after the last excitation light has been deleted. The fundamental concept of fluorescence microscopy is to excite the sample with a specific wavelength, and to separate the fainter emitted light which comes from the fluorescence afterwards. The emitted light can be detected with a camera adjusted to the wavelength. In Figure 1.6 the physical concept behind fluorescence is shown. Light forces the specimens to enter an excited state (S_1). Electrons will lose their vibration energy after relaxing back to the ground state. This process results in fluorescence, which causes a wavelength shift (stokes shift), which causes a wavelength shift (stokes shift) different to the excitation wavelength. During the relaxation from (S_1) to the ground state (S_0), the fluorescent light is emitted. [5]; [10]; [31]

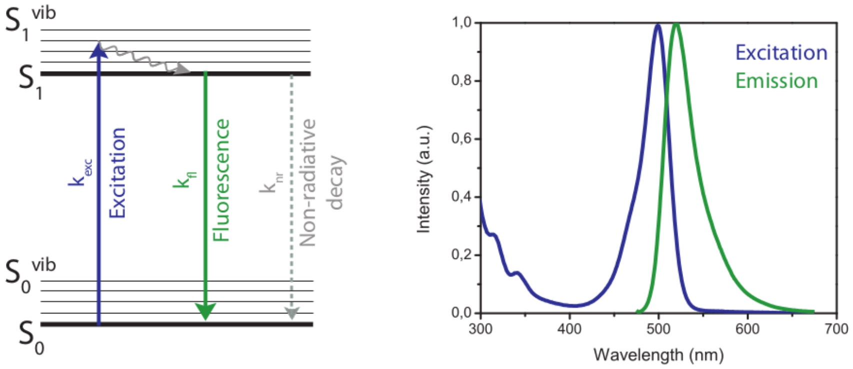


Figure 1.6: Fluorescence description with stokes shift for wavelength [4]

1.7 MEA-System

The system shown in Figure 1.7 consists of the headstage, a chip with electrodes to trigger and measure the action potentials, an interface board to send the received data via USB 3.0 to the computer and software which displays the measured signals in real time with 4225 sensing electrodes and 1024 stimulation points. The data mining from the chip is relatively large, almost 250 MByte per second, therefore the provided software is needed to create, a stimulus and measure subsequent potentials. The data rate of the headstage is between 18 kHz to 25 kHz per channel and has a resolution of 14 bit. The created file has Hierarchical Data Format (HDF5). [16] The main device includes electronic for data filtering and amplification, before the signals are transferred to the computer. In addition, an external reference electrode has to be placed in the bath solution depending on the chip actually. For some CMOS chips it is not necessary. Furthermore, a heating plate is integrated to achieve physiological conditions to ensure the well-being of the brain slices. For this reason the buffer solution is warmed up as well before reaching the chip, the striven temperature is around 35 °C. Temperature is controlled and adjusted with a separate device, the Temperature Controller TC01/TC02. The heating plate in the headstage is controlled with this device and it also heats the buffer solution and therefore both outputs can be adjusted. [16]; [19]

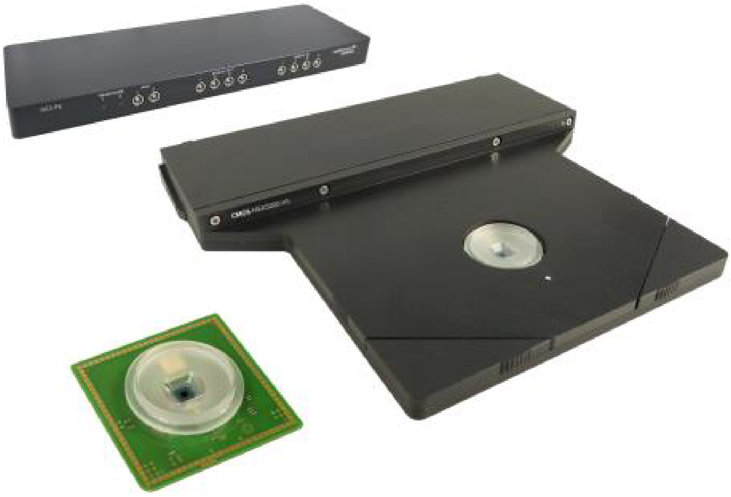


Figure 1.7: MEA-5000 system with headstage, interface board and MEA chip. [16]

The previous system before CMOS-MEA technology had a maximum of electrode channels for detection in the range of 60, 120 and 256. The MEA-2100 system consists of the same components. The chips has to be inserted in the headstage, which has an amplifier, analog digital converter and stimulus generator. The second part is the interface board with a signal processor. Sampling frequency is up to 50 kHz per channel, resolution is 16 bit for in and out channels and eight channels are available as input. The software also delivers a HDF5 file for further investigations. Therefore, another software must be used to stimulate and measure. The generated impulses are rectangular monophasic or biphasic and may be single or pulse trains. [18]

Other manufacturers integrate the main device and interface board into one system, as shown in Figure 1.8 . The temperature control and prepossessing is also installed. The device has only 4096 electrodes, but here the number of sensing and stimulation electrodes are the same. Sampling frequency is 20 kHz with a 12 bit resolution. [1]

Figure 1.8: BioCAM-X MEA-system, all in one device. [1]

There are also CMOS arrays which use optical stimulation and record electrically. The chip has a light-emitting diode (LED) array which allows spatiotemporal manipulation of brain slices for neuronal activities. The array consists of a transparent glass substrate which eases the stimulation of the tissue with LEDs. The electrical signals are recorded with photodiodes. [21]

1.7.1 Multielectrode Arrays

The first multielectrode arrays (MEA) were developed in the 1970s. The systems Figure 1.9 allow multi site sensing and activity tracking of neuronal electrical signals in vitro. Microelectrodes are separated by $\sim$400 µm and have a size of $\sim$30 µm in diameter and there are 60, 128 or 256 electrodes on one array. MEAs are divided into passive and active devices based on the fabrication technology. Passive chips belong to the old generation. The active MEAs or complementary metal oxide semiconductor, CMOS-MEA were developed in the last 20 years, the devices are fabricated in semiconductor foundries. The active chips have to be powered up before they can be used, as well as powered down to prevent damage. [13]

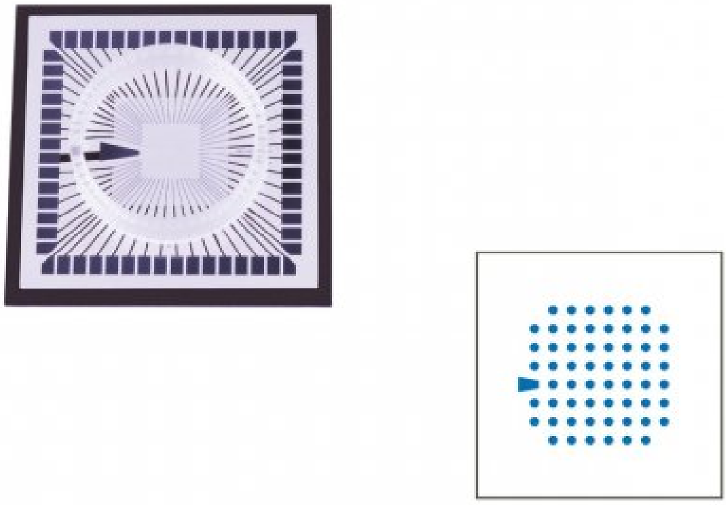


Figure 1.9: MEA chip with 60 electrodes, on the left side the chip on the right side the schematic electrodes. [29]

1.7.2 CMOS-Multielectrode Arrays

The chips have a small electrode separation between two electrodes up to 10 µm for a high resolution. The cell size is between 10 µm to 20 µm. The CMOS-MEAs in Figure 1.10 are separated in two major types. The first type are full-imaging MEAs. Here the stimulatory electrodes have the same number as the channels for simultaneous readout. The other one uses a switch-matrix where the electrodes are sequentially used. The number of electrodes reaches 16000 or more in some chips. It allows the recording of a bandwidth of signals such as slow field potentials up to single action potentials. This technology is light sensitive, hence the sensor needs constant light settings, which can be provided by covering the chip with a petri dish encased in aluminium. [30]

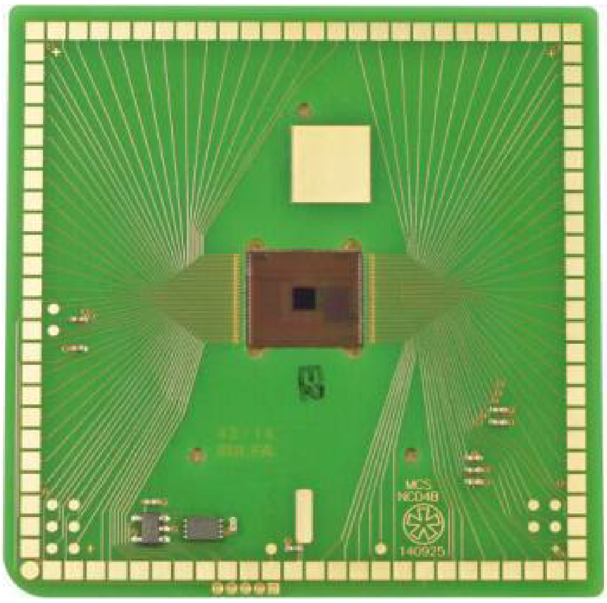


Figure 1.10: CMOS chip with 4225 recording and 1024 stimulation electrodes. [29]

1.8 Software

Generally, two types of software are used to stimulate and measure the hippocampus slice and to edit and transform the signals and pictures. Camera software is used for taking photos of the brain slices. Main software is the CMOS-MEA control which is provided by the system to control and handle it, the other one was Matlab for image acquisition and processing and also for the overlaying of both data sets. [29]

1.8.1 Matlab

Matlab is a software to solve mathematical problems and produce graphic representation of the conclusions. The program achieves this by numerical calculation with the help of matrices. In comparison to other programs, Matlab primarily uses numerical and not symbolic solutions to solve problems. Expansion Simulink adds an internal program, which can be used for time-controlled or event-oriented simulations. Toolboxes allow the user to work on specific problems like signal or image processing. Other extensions are Computational Finance or High Frequency and Mixed Signal. In this work, toolboxes used for image processing and visual representation are demonstrated. [14]

2 Tasks

In this thesis, we first aimed for electrophysiological data of hippocampal slices acquired using MEAs with the location of the neurons on the organotypic slices. Pictures are taken with a microscope, data will be processed and both, the image and the measured signals are mapped. The overlay will be done with a mathematical program, including video and picture processing. An overview of the steps is displayed in Figure 2.1. The recorded signals should undergo pre-processing where the spikes are filtered and increased occurrence is detected. When the data are combined it should be possible to detect single neurons after an action potential, as the resolution of the CMOS-MEA is high enough. The same measurement can be done using the 60 electrode MEA chip, and tracking wave propagation should be possible. Using this device, the single neuron mapping is not feasible as the resolution is not high enough due to the space between the electrodes. Activities can then be assigned to specific regions.

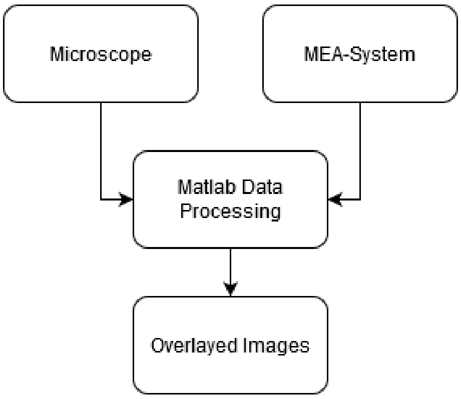


Figure 2.1: Flowchart describing the principle steps from data acquisition to outcome.

The second part of this thesis is the same procedure as described using hippocampal slices, however, with dissociated neurons after they have formed organic connections and can propagate action potentials. Also in this case all processes will be done, except the marking of regions. The workflow remains the same as shown in the overview flowchart in Figure 2.1. The neurons should also be enriched with fluorescent substances to see them better under microscope. Limitation from light source is that only reflected-light can be used for CMOS chips.

3 Methods

This part is divided into two big steps. The first one is related to measurements with the MEA-system and recording pictures with a microscope. The collected data could be from organotypic slices or dissociated neurons. The other one includes image processing and programming in Matlab, as well as compiling heatmaps via codes and saving them.

3.1 Preparation

Before the rat brain slices or the dissociated neurons can be used for the measurements on the MEA-system they need to be prepared. This is different for slices and neurons. Also the nutritional support during the measurements is different for both. Figure 3.1 demonstrates the principle interaction between the devices during measurement of hippocampal slices. For dissociated neurons the perfusion system can be omitted.

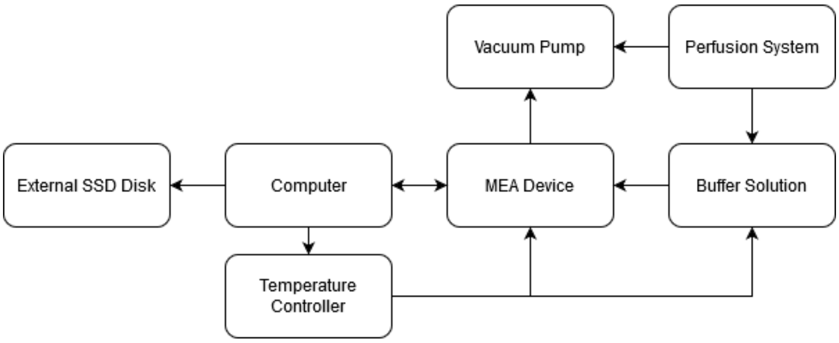


Figure 3.1: Measurement setup arrows display how the devices interact with each other

3.1.1 Hippocampal Brain Slices

Hippocampal brain slices are placed on a membrane where they stayed for approximately two weeks. During the incubation time the solution surroundings the tissue slices needs to be replaced every three to four days. Otherwise brain slices will die and cannot be used anymore. The incubator containing the slices requires a permanent temperature of 37 °C and 5 % CO_2. Before the brain slice is taken from the membrane and placed on the MEA chip, the surface has to be cleaned and the slices will be transferred with the help of a scalpel and tweezer. Only one slice is positioned on the chip at the same time.

3.1.2 Dissociated Neurons

An advantage of dissociated neurons is that they can be seeded directly onto the MEA chips after PEI coating. Furthermore, a continuous perfusion of the dissociated neurons is not necessary during measurements. When neurons are seeded on the chip, they have to be incubated for approximately two weeks at 37 °C and 5 % CO_2. This is necessary to let the connections grow between the neurons. If the chip is taken too early for measurements, the connections are not fully developed and it is not possible to record wave propagations.

3.2 Buffer Solution

The buffer solution had a pH close to the physiological value, shown in Table 3.1, and flew over the slices during the measurements. After running over the tissue, a pump vacuumed the liquid away. The liquid was filled in a 50 ml tube which had to be filled during the whole measurement. Otherwise, air would get into the inlet tube which could cause noise and lead to a dry out of the slice.

Chemical	**M.W.**	**Conc. [mM]**	**g/1L 10x Stock**	**g/2L 10x Stock**	**g/500mL ACSF**	**g/1L ACSF**
NaCl	58.44	125	73.05	146.10		
KCl	74.55	3.5	2.60	5.21		
$NaH_2PO_4 \cdot 2H_2O$	156.01	1.2	1.87	3.74		
$CaCl_2 \cdot 2H_2O$	147.01	2.4	3.52	7.05		
$MgCl_2 \cdot 6H_2O$	203.30	1.3	2.64	5.28		
$NaHCO_3$	84.01	26			1.09	2.18
Glucose$\cdot H_2O$	198.17	25			2.47	4.95

Table 3.1: ACSF Buffer Solution with a pH 7.35. [17]

3.3 Measurement Setup

For measurements, the MEA control software was needed [16] to record the signals and set the trigger impulses. The amount of acquired data was very high, although the measurements were recorded only for a few seconds. Artefacts or noise have to be avoided. The reference electrode helps with this, as does an optimal setup for the perfusion including the temperature control for both, the buffer and the chip, the vacuum pump and the wire which fixes the slices to the chip to avoid its swimming off. In Figure 3.2, the setup for the recording is shown: in red the headstage, green is the temperature controller for adjustment of the buffer solution as well as the headstage and blue the perfusion system which controls the inflow and the drain.

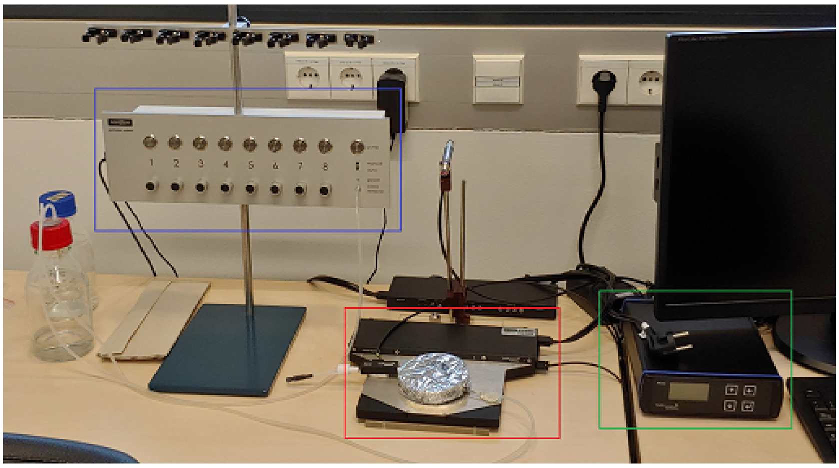


Figure 3.2: Measurement setup with the important devices showing headstage in red, temperature controller in green, perfusion and vacuum system in blue

In Figure 3.3, the arrangement for a hippocampal slice recording is displayed. On the left side is the tempered influx of the buffer solution, on the right side the drain of the vacuum pump and at the bottom the reference electrode, which is connected to the solution. In the middle of the chip, the slice is placed and fixed with a platinum wire for biocompatibility. When neurons are recorded, the influx and drain can be omitted, as well as the fixation. Another type of chip can also be used, where the reference electrode is already implemented into the array and no external is needed.

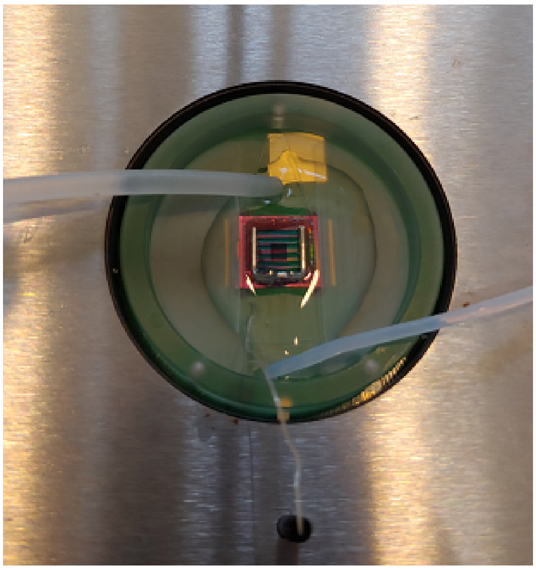


Figure 3.3: Chip during recording, on the left side top the inflow, right bottom drain, in the middle bottom the reference electrode within in the solution. The slice is fixed with a platinum wire.

3.3.1 CMOS-MEA Control

The control software in Figure 3.4 will be used for online recording. It also includes the stimulation via electrodes. This part shows real-time activity and gives an overview of the chip. The raw data can be visualized and zoomed into with the control panel. Recording to an SSD hard disk assists in saving space on the disk while live-observing and this can also be done for spike sorting. In the control section, the storage can be chosen. This applies for general parameters. Interesting regions can be selected to reduce data and, in the overview, the whole chip is displayed. Other functions like single view, indicate only one electrode, while in the panel detail view the chosen electrodes of the overview panel will be displayed. For both, the resolution can be changed and in single view also the time scale. The software can also create a stimulus and send it to the electrodes. Therefore, the stimulation area has to be marked, which can be done in the stimulation panel. The generator can only create stimulations with a resolution of 10 µs and 105 µV, shorter and lower is not possible. The impulse must have a

positive voltage, to prevent damage on the chip. The stimulation window can be configured for a single pulse or pulse sequence. [16]
Temperature control is achieved with an additional program where the heating plate of the headstage and the inflow of the buffer solution can be controlled and regulated. The default parameters for the solution start at a medium flow. The temperature is displayed over time and can be adjusted for both channels separately. [19]

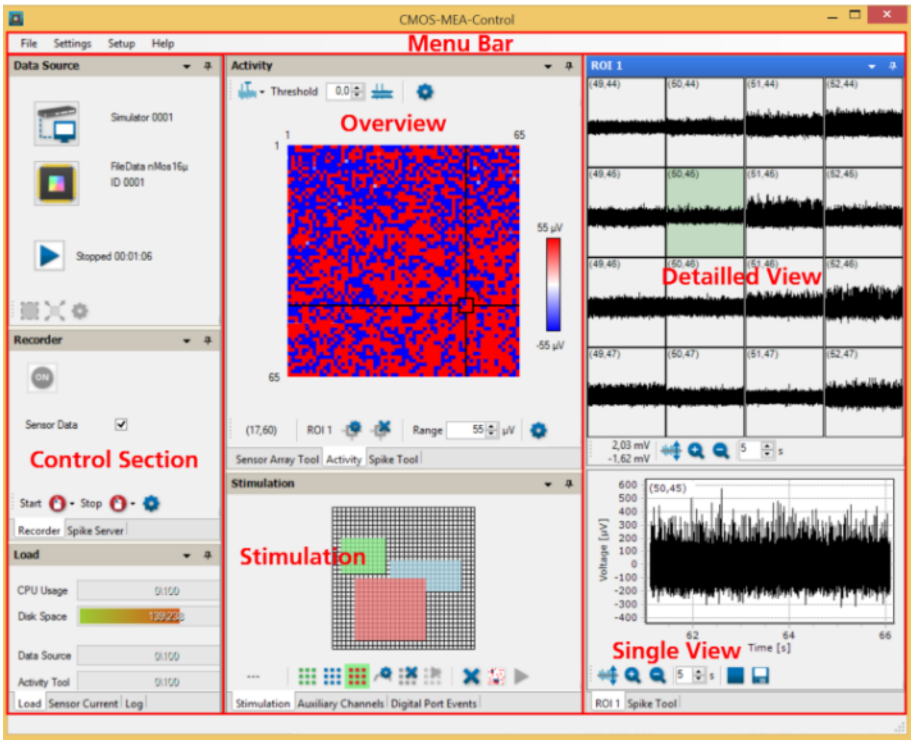


Figure 3.4: CMOS-MEA control software overview. Control section on the left side shows general information, the overview in the middle displays the chip array and the stimulation to trigger an impulse. The right side detailed view visualizes a small number of electrodes and the single view displays only one electrode. [16]

3.3.2 Spike Sorting

The spike sorting algorithm from Multi Channel Systems [16], was created to identify neural units from recorded signals. This can be done for CMOS-MEA or 60 electrode MEA. Spike sorting will be done completely automatically due to the high number of channels. The assumption is that a sensor record several signals, resulting in a mixture of noise signals and zero or more source signals from neurons. The raw data combines a mixture of overlapping signals. The program will separate those signals, ideally one signal belongs to a neuron. The main steps of spike sorting are to divide the chip in regions of 100 to 200 electrodes and for each region of interest the sources will be analyzed separately. Every detected event shows activity in several sensors at the same time. Each event forms a region of interest, these regions are ultimately overlapping and merged together. The outcome of this sorting algorithm is illustrated in Figure 3.5. More details to the spike sorting can be found here. [16]
Further details to the algorithm can also be found in this paper, along with the accurate equation and the implementation. [12]

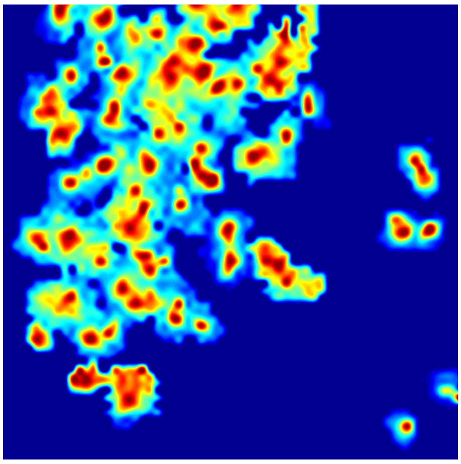


Figure 3.5: Chip array with the map after spike sorting, the colored points are regions of interest. [16]

3.3.3 Processing Recorded Data

The recorded data has to be investigated with the MEA-tools software, which also includes the spike sorting code. Neuronal activities are identified by probability of spikes measured with the MEA-system.

The software of the CMOS-MEA device creates a video where the activity on the chip is shown. Due to this video not being exact enough, the decision was made to use the raw data and create a heatmap with all data points via Matlab, as shown in Figure 3.6. This will reduce the chance of missing events. The first step was to create a matrix, which has the same number of rows and columns as the chip. An advantage towards the 60 electrode system is that the data file already has the correct format. The files are in HDF5 size which can be handled via Matlab. The limited memory space on a computer made a file segmentation necessary, heatmaps were created for every data frame, a data frame is the measured signals over the chip at the same time and a pseudo-coloration as seen in the CMOS-MEA tool, was generated.

The 60 electrode MEA device displays the results as signals in channels over time. To avoid this problem, a code was created in Matlab, displayed in Figure 3.6, allowing to use the data points of all electrodes and to visualize them in a heatmap. First step was to arrange the data and input them into the matrix. Afterwards the average of 1000 to 10 000 data points was calculated for each electrode and saved in a matrix with the same order as the MEA chip. Then a heatmap was generated, with a colorbar like the CMOS-MEA tool. This could be repeated till all data were imported.

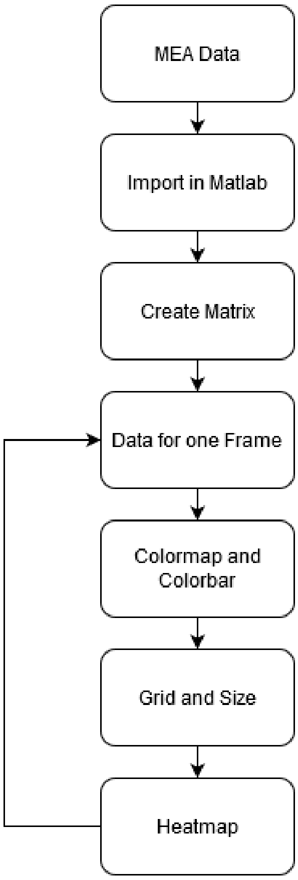


Figure 3.6: Flowchart describes the code to create the heatmap in Matlab, for CMOS-MEA and 60 electrode MEA files For the 60 electrode chip device small adjustments were done.

3.4 Image Recording

The image was recorded with a CMOS camera connected to a microscope. The light which was applied for the pictures was reflected, as no transmitted light passes through the chip. This would be possible for the 60 electrode chip. On the microscope was a mount to hold the chips in place as shown in Figure 3.7. However, it is difficult to place the chip well on the microscope to achieve a 90° angle and hence, a nice image. The microscope was an

Olympus BX51 with reflected-light. The plan was to change the focus for the pictures to several planes of the brain slices. Two photos should be recorded, one focused on the slice structure and the other one on the chip array. It will also work with only one picture, as long as the structure and the edges visible in the image. This adjustments were needed to determine the edges and corners of the array, to find out when rotations in Matlab were necessary and which parts in the image could be cut away. The thickness of the slices are ∼400 µm. For dissociated neurons only one was needed, as under the neurons the chip was visible as well as the edges of the array.

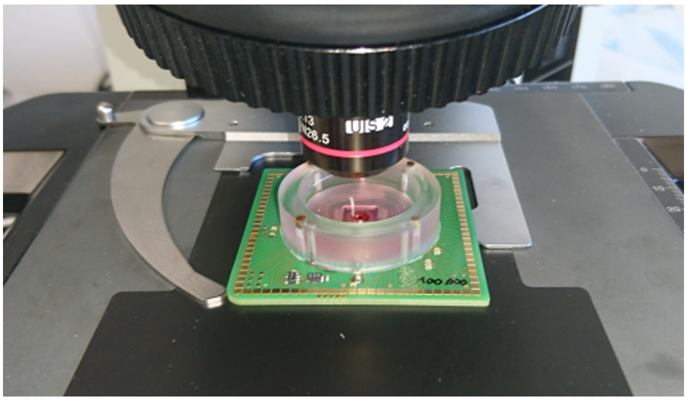


Figure 3.7: Recording images from hippocampal slices or neurons by microscope from Olympus, mounting to fixture the chips during the procedure.

The camera was from Jenoptik (GRYPHAX Progres Subra), as shown in Figure 3.8, using CMOS technology to receive high resolution images with a rapid focus change, to reduce the exposure time. This is important for the membrane where the slices are placed. After the perfusion is stopped, the membrane will dry out, alters the shape of the organotypic slices irreversible, results in dehydration it was not possible to see different structures as the slice becomes transparent. The Camera can detect wavelengths over the visible spectrum. The best sensitivity is between 430 nm and 480 nm and between 530 nm and 580 nm. The highest relative spectral sensitivity is with a wavelength around 550 nm. These values should be compatible to the microscope used. The dissociated neurons are hardly visible on the CMOS-MEA chip resulting from the electrode's dark surface. Fluorescence would help to see the neurons more clearly when the image is used for the overlaying. [11]

Figure 3.8: CMOS camera connected to the microscope. [11]

3.5 Image Processing

The next sub-headlines explain the steps which were done with Matlab. In Figure 3.9, all single steps for overlaying neurons are displayed. In fact, from the detected signals received from the array a heatmap was created in Matlab, where a loop run every iteration through one data line. In this way it was also easy to define which parts of the recording were interesting, and the recorded data can be watched for a few seconds or the hole record. The images were then saved into a file. In the next sub headlines the details to the single steps are described. The flowchart is for CMOS-MEA and 60 electrode MEA, is the same for slices and neurons.

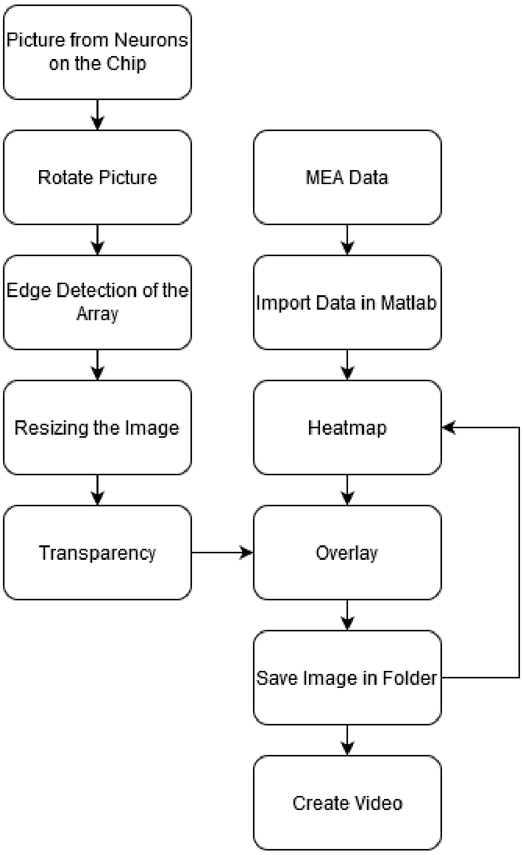


Figure 3.9: Flowchart describing all steps of the code for dissociated neurons in Matlab, from import to the final part the video.

In Figure 3.10, a flowchart is displayed to demonstrate the code for rat hippocampal slices, further details are described in the next sub headlines. This code also creates the heatmap in Matlab for every iteration and after overlaying the image is saved into a folder. The last step was the creation of the video.

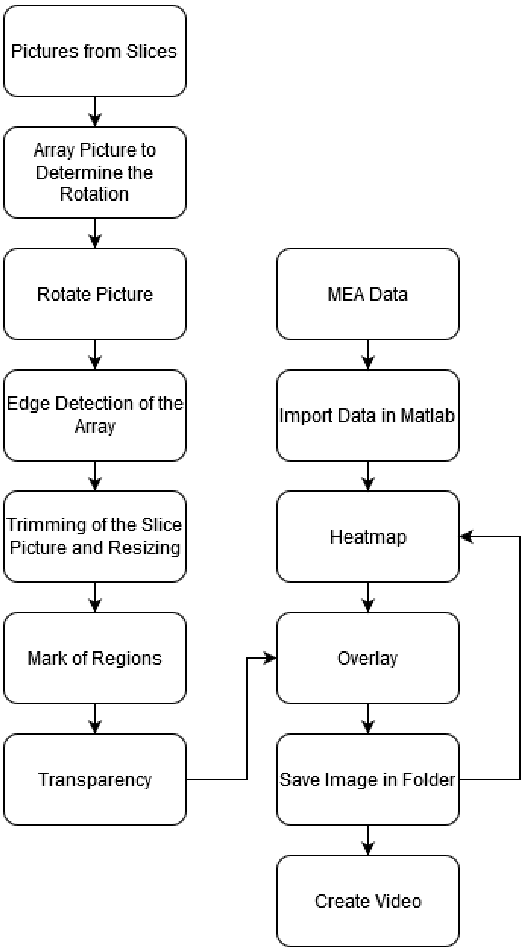


Figure 3.10: Flowchart describing the steps of the code for hippocampus slices in Matlab.

The microscope image was loaded into Matlab. Then the angle of the MEA chip array was checked and rotated as long as the edge of the chip array lie parallel to the upper image border, the rotation was done with a bilinear filter. The plan was to record at least two pictures from the brain slices, but it will work also with one picture, in this work only one picture was taken and then modified. With two pictures the handle will be, the second picture with focus on the chip array was investigated to find out the rotation parameters and the edges of the array. This informations are used on the slices structure image. Afterwards, a rectangle with starting values in one corner was selected and placed over the chip array, the parts beyond the square where removed. This procedure was the same for both samples. Only the chip array with parts of the brain or neurons remains on the pictures.

3.5.1 Image Resizing

The recorded microscope images were extended to the size of the MEA heatmap, which was again done for both, dissociated neurons and organotypic slices. For the slices only one picture is needed, one where the structure from the hippocampus can be seen. The size of the heatmap was set in the program and could be adapted.
Generation of the heatmap also showed the number of electrodes in a grid besides the square. Hence, it was easy to find out which electrodes should be investigated in more detail. On the side, a colorbar shows the maximum and minimum voltage which was measured in the record, which correlates with the data points shown in the heatmap. To see the depolarisation and generally the membrane potential over the tissue or neurons an optical scala was included, where red has the highest positive potential in the recorded file, white represent 0 V and blue is the highest negative potential.

3.5.2 Marking of Regions

For the hippocampal slices an additional option was possible to mark important regions. This option was implemented in Matlab, where the name of the region was written in the image like (CA1), (CA3) or (DG). Therefore, a command was used to implement the text in the picture. Color as well as the size and style of the font can be chosen. X/Y - coordinates are adjusted to shift the text to the appropriate place in the picture.

3.5.3 Overlaying

To generate an overlay, it is necessary to exactly cut the picture, as overlaying only makes sense when the pictures have the same angle and size. The microscope images are changed in transparency depending on the brightness of the image for the overlaying.

3.5.4 Mapping

The mapping was the last step and it was planned to do it, with the spike sorting algorithm file. Neuronal spikes in the recorded file will be detected with this software, the overlay will then show in which region of the hippocampus or the dissociated neurons the specific neuron is placed.

3.5.5 Video Creation

After the last overlayed heatmap is saved into a folder, all pictures are merged into one video, where the length of this video corresponds to the number of heatmaps generated, this number can be selected in the GUI.

4 Results

This chapter displays the results from the measurements with the MEA-system, the saved pictures form the microscope and the overlay combination of both, which is the output of the Matlab code.

4.1 Microscope Images

The sections has the same name as later the heatmaps and overlays, to define the context between these headlines.

4.1.1 Microscope Measurement 1

The raw data acquired by the microscope are shown in Figure 4.1. These images are used for the latter overlay. The pictures are named according to the names of the heatmaps and overlays.

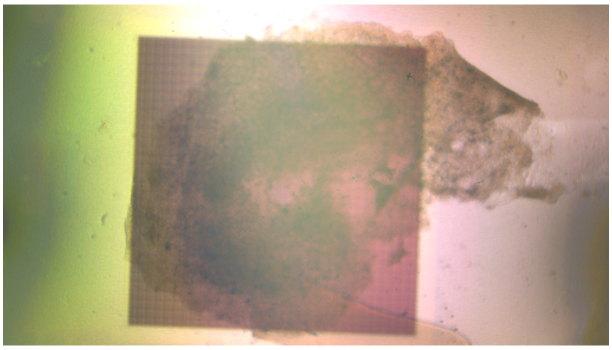


Figure 4.1: Microscope image of a hippocampal slice, without image processing, from 2019.03.19

4.1.2 Microscope Measurement 2

The microscope image in Figure 4.2 shows a hippocampal slice on the chip. This picture was used for measurement 2 in the adjustment and latter in the heatmap and overlay.

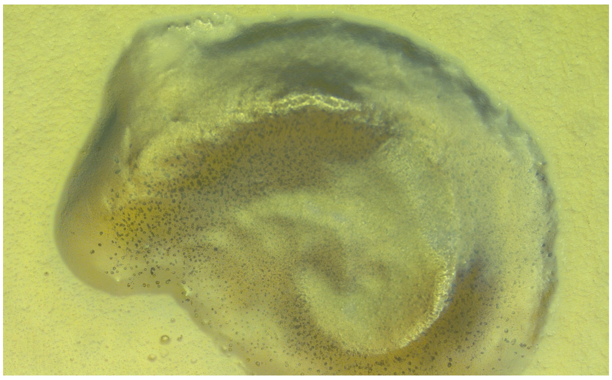


Figure 4.2: Microscope image of a hippocampal slice, without image processing, from 2019.03.20

4.1.3 Microscopy of Cultured Neurons

Figure 4.3 shows the image of dissociated neurons cultured on a CMOS-MEA chip. To increase the visibility of the neurons, the red channel of the original PNG image was removed. For this case fluorescence was planned to make them more visible.

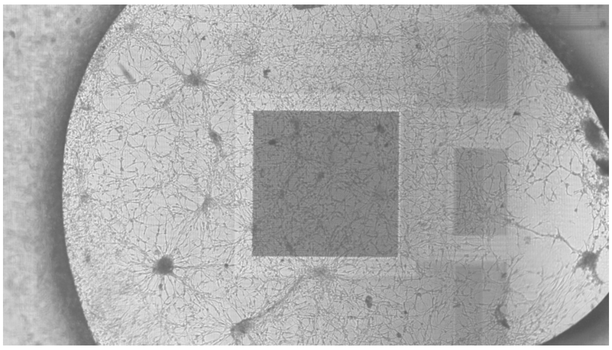


Figure 4.3: Microscope image of dissociated neurons which are cultured on the chip, the red channel in the image was removed, from 2020.02.11

4.1.4 Microscope 60 Electrode MEA

The image in Figure 4.4 show the microscope picture of the 60 electrode MEA, with dissociated neurons. The image was then modified in headline image adjustment. Furthermore, it was used for the 60 electrode MEA overlay which is shown in the next headlines.

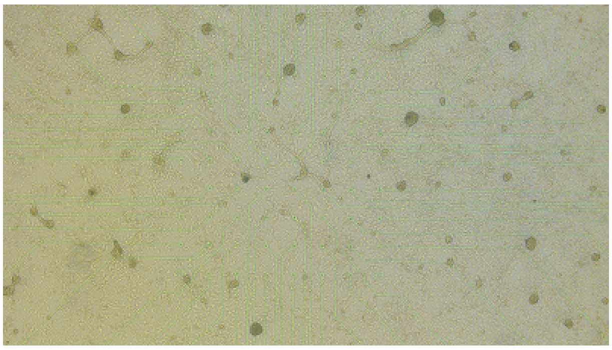


Figure 4.4: Microscope image from dissociated neurons, without image processing, coated on the 60 electrode MEA, from 2020.02.11

4.2 MEA-Results

The results from the CMOS and 60 electrode MEA can be opened in Matlab and the data converted into the heatmap see section 3.3.3.

4.2.1 Spike Sorting with Software

The MEA software can process the measured data and sort them to a number of interesting regions, followed by spike recognition. In Figure 4.5, the regions of interest are shown, from measurement 2.

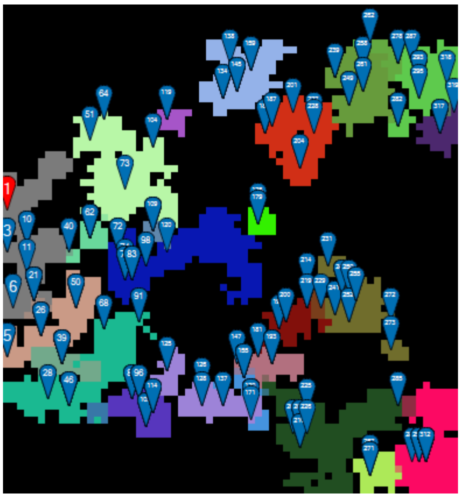


Figure 4.5: Outcome of the spike sorting tool, the picture is divided in regions of interest, automatically done by the program, shown in results as measurements 2

After the regions of interest are calculated the software assigns the spikes in this area. In Figure 4.6, the map is shown afterwards, the red points show areas where spikes were measured.

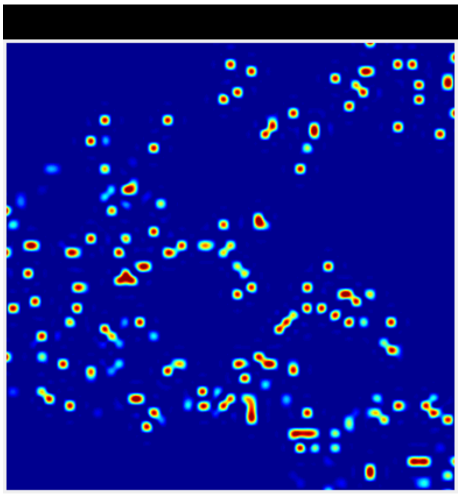


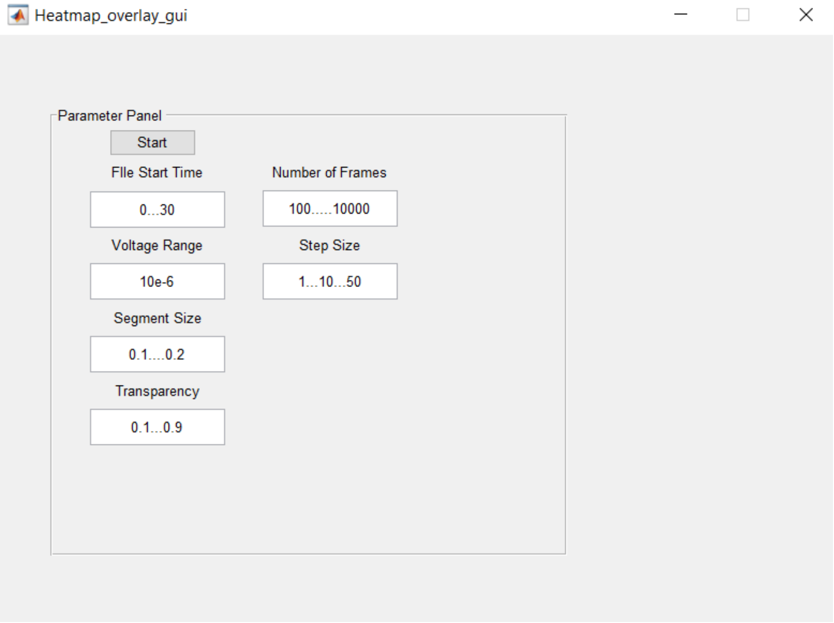
Figure 4.6: Spike sorting will generate this map where the highest number of spikes are shown in red, in results as measurements 2

4.3 Program

All parts of the program were implemented in Matlab R2018a. The following subsections correspond to the individual steps of the process, where the last result is the overlayed video.

4.3.1 Code GUI

For a better handling a guide was implemented in Matlab shown in Figure 4.7, where some parameters can be changed. The exact matching of the microscope images is semi automatically, by investigating the pixels in the picture. The x/y pixel values from the image has to be written in the code. The GUI includes a *Start Button* and other numerical parameters, such as the *File Start Time*, defining where the sequence starts in the file, the *Segment Size*, defining how many data lines are read from the file, and the *Voltage Range*, depending on the measured signals, which can vary between data. The *Transparency* of the image overlay can be adjusted and depends on the intensity of the picture. *Number of Frames* defines how many data lines will be plotted and merged to the video, and the *Step Size* allows fine adjustment,defining whether every data line should be read or, for example, only data lines every second. Some example values are shown in the Figure.



FIle Start Time: 0...30

Number of Frames: 100.....10000

Voltage Range: 10e-6

Step Size: 1...10...50

Segment Size: 0.1....0.2

Transparency: 0.1...0.9

Figure 4.7: Matlab GUI to set parameters for the heatmap and overlay

4.3.2 Image Adjustment

The images were rotated and trimmed, and bilinear filtering was used to get an accurate image. Also a resizing was already done in this pictures. Figure 4.8 shows data recorded from measurement 1.

Figure 4.9 shows the image after rotation and cutting. The filter was the same as in the other pictures and the image was already resized and used in measurement 2.

Cultured neurons coated on the CMOS-MEA chip are shown in Figure 4.10. After resizing and rotation, neurons and cell clusters became visible on the image.

In Figure 4.11, the adjustment for the microscope image is shown.

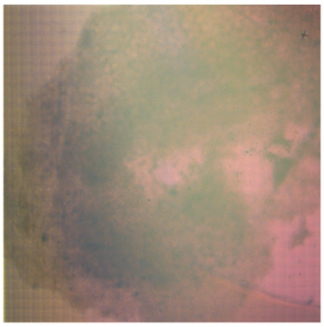


Figure 4.8: Image after rotating, trimming and resizing, shown in results as measurement 1

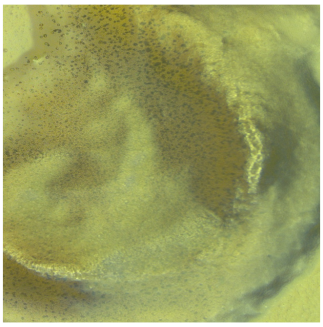


Figure 4.9: Image after rotation, also the trimmed and resized, shown in results as measurement 2

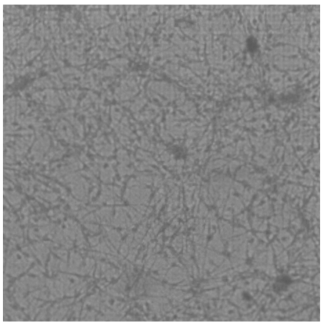


Figure 4.10: Image after rotating, trimming and resizing, without red channel, used in cultured neurons.

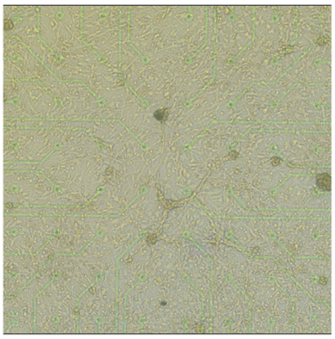


Figure 4.11: Image show the resized MEA array with dissociated neurons on it.

4.3.3 Marking of Regions

After the images had the proper size, an overlay was added to mark important regions. This part was only relevant for organotypic slices. Important areas are Dentate Gyrus DG and Cornu Ammon CA3, see section 1.1. Figure 4.12 shows the results for measurement 1. For measurement 2 the region was marked. In Figure 4.13, the result is shown.

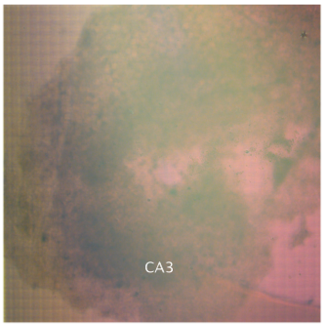


Figure 4.12: Region was marked in the image, shown in results as measurements 1

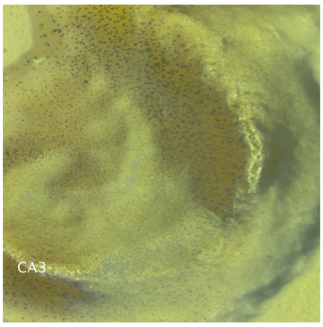


Figure 4.13: Region was marked in the image, shown in results as measurements 2

4.3.4 CMOS-MEA Heatmap

This section displays the results which were measured with the CMOS-MEA system and subsequently used for the Matlab analysis, split into hippocampus slices and cultured neurons equipped with numbers. Here, only some examples are represented, the other parts can be found in the Appendix.

4.3.4.1 Slices Measurement 1

In Figure 4.14, the heatmaps over 22.5 ms are shown, where the step size between the images is 2 ms, the starting time from the data set is 20 seconds. The file date is 2019.03.19 when the recording was done. Stimulus was applied in the left corner, shown as square.

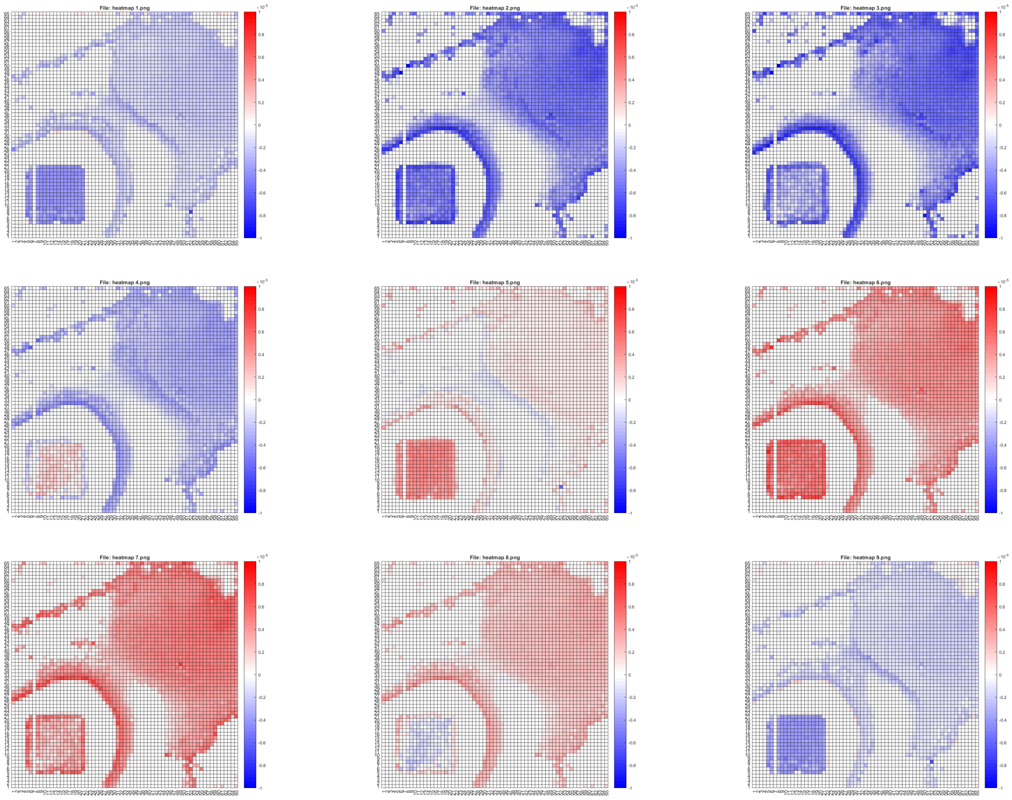


Figure 4.14: Heatmaps were created with the MEA data, start time was 20 seconds and the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.4.2 Slices Measurement 2

The shown heatmap belongs to the headline 4.2.1, where the spike sorting was done. In this setup no voltage was applied and only spontaneous action potentials were measured. Figure 4.15, shows the heatmap of the MEA data points.

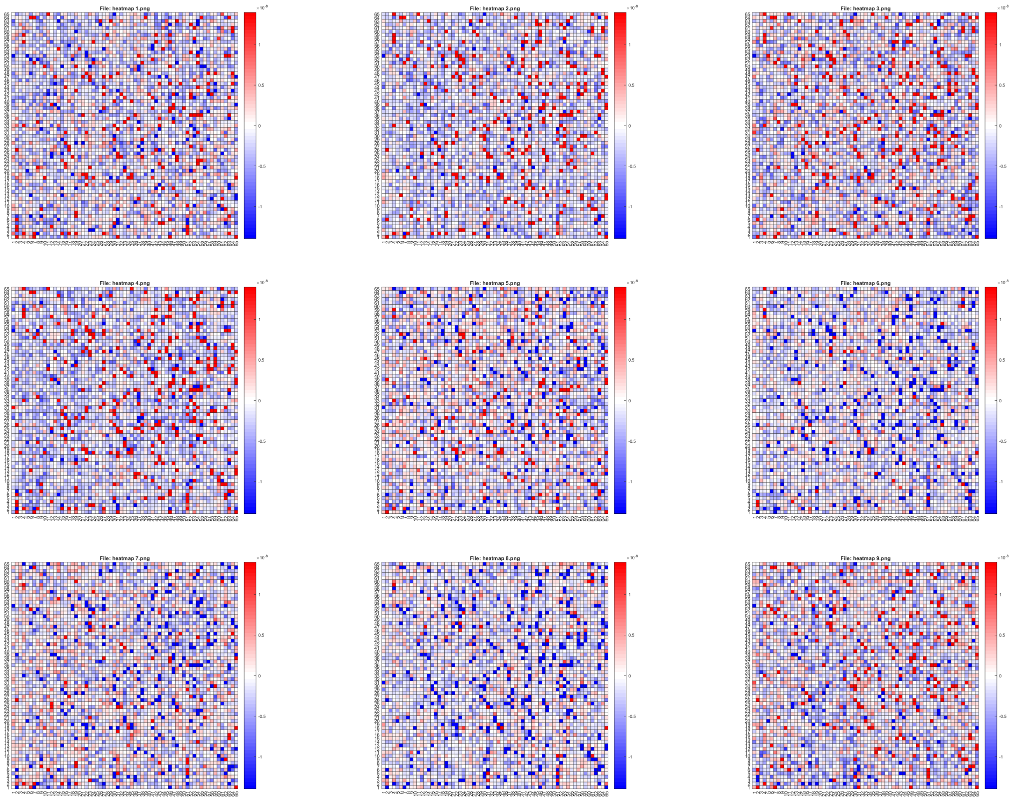


Figure 4.15: Heatmaps were created with the MEA data points. Spontaneous action potentials were measured and no voltage was applied. Start time was 5 seconds and the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.4.3 Cultured Neurons

In this section, dissociated neurons were measured, shown in Figure 4.16. The start time in the data set was 20 seconds and the frame rate was 2 ms.

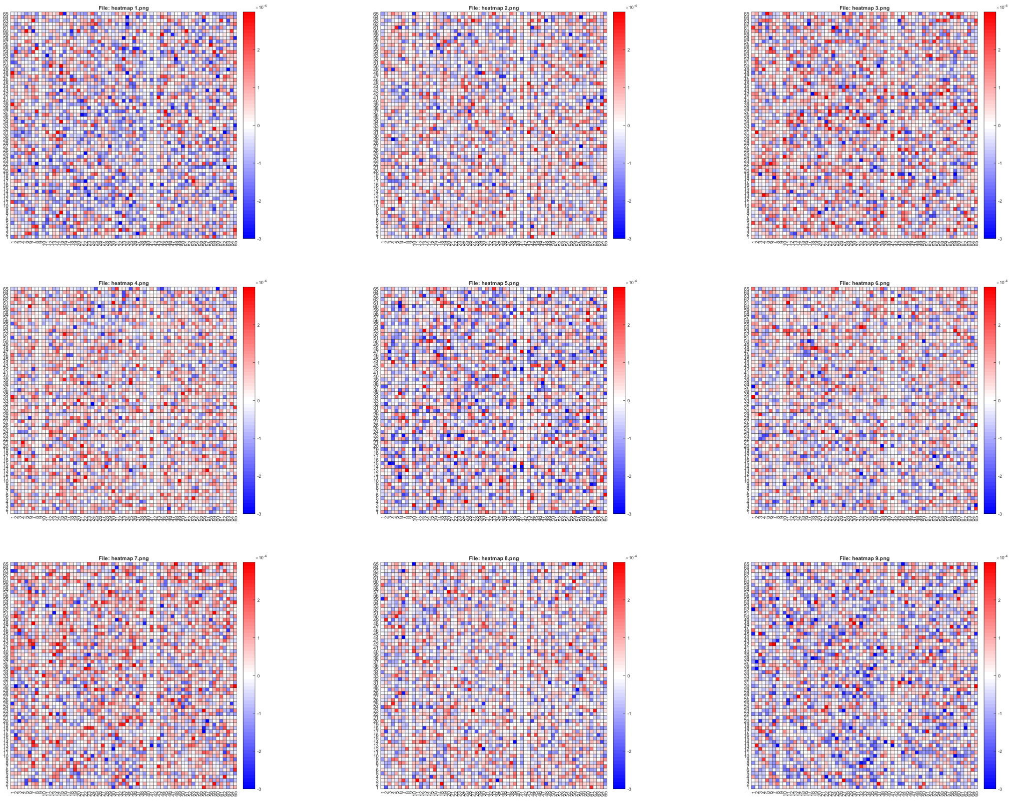


Figure 4.16: Heatmaps were created with the MEA data points with dissociated neurons on it. Start time was 20 seconds and the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.5 60 Electrode MEA Heatmap

For the 60 electrode MEA the measurement was done only with dissociated neurons. The heatmap shown in Figure 4.17 was created with the average over a number of data, to reduce the noise. It would also be possible to use single data lines. In all four corners the value is not existing and white, as there are no electrodes in this region. The starting time was 5 seconds in the file.

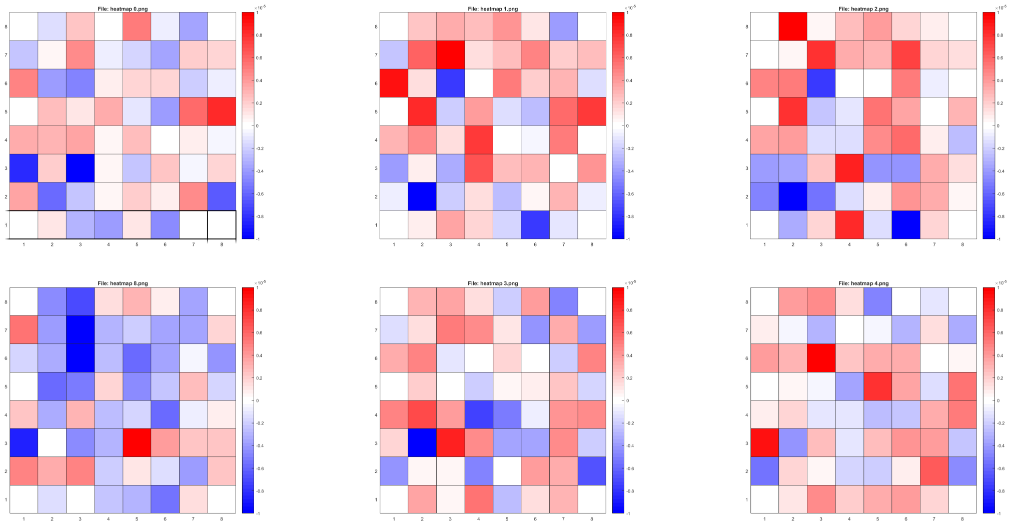


Figure 4.17: Heatmap created with the 60 electrode MEA data, start time is 5 seconds the frames display the average of a number of data. The first frame is shown top left, the last frame is at the bottom right, in the corners the value is not existing because there is no electrode on the array.

4.3.6 Overlaying

The next sub headlines represent the outcome after the overlaying was done, divided in the individual measurements. The CMOS-system was used for slice measurement 1 and 2 also for the cultured neurons, the 60 electrode system was only used once.

4.3.6.1 Slice Measurement 1

In Figure 4.18, the heatmap from Figure 4.14 is shown with the slice overlay, the hole image series has a duration of 2 ms.

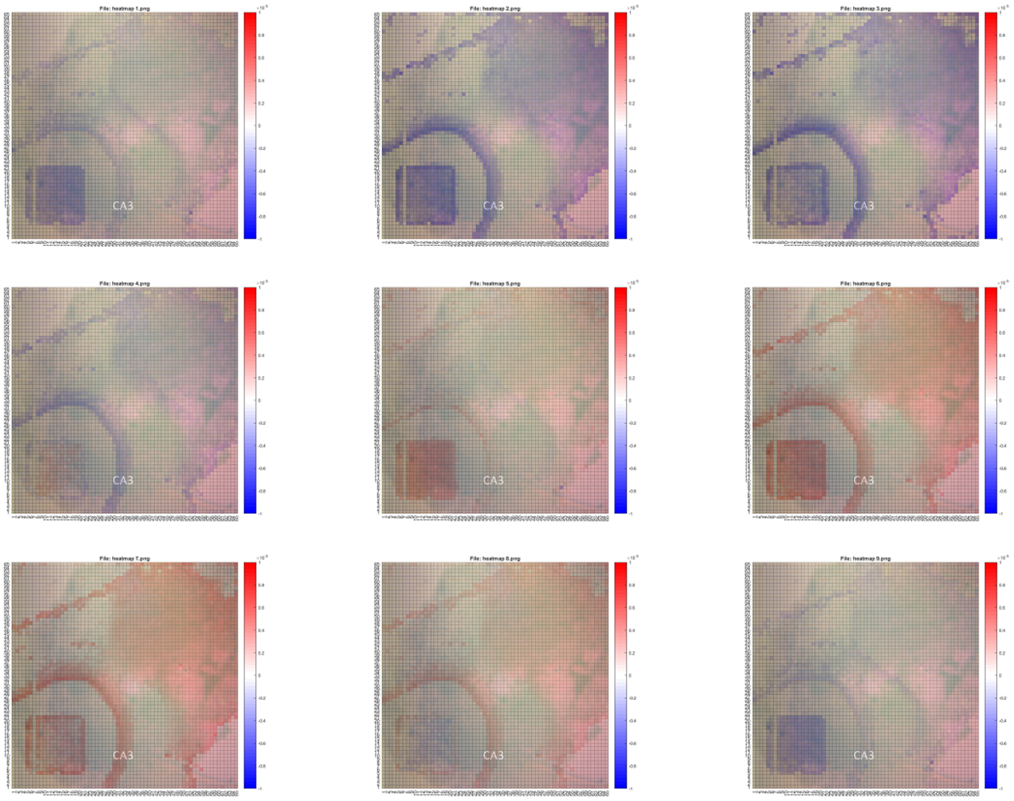


Figure 4.18: Overlay with the heatmap from Figure 4.14 and Figure 4.12, start time is 20 seconds the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.6.2 Slice Measurement 2

Figure 4.19, shows the overlay of the heatmap and the hippocampal slice. In this frame series no action potential was applied, the measured spontaneous action potentials can be seen in the spike sorting 4.2.1.

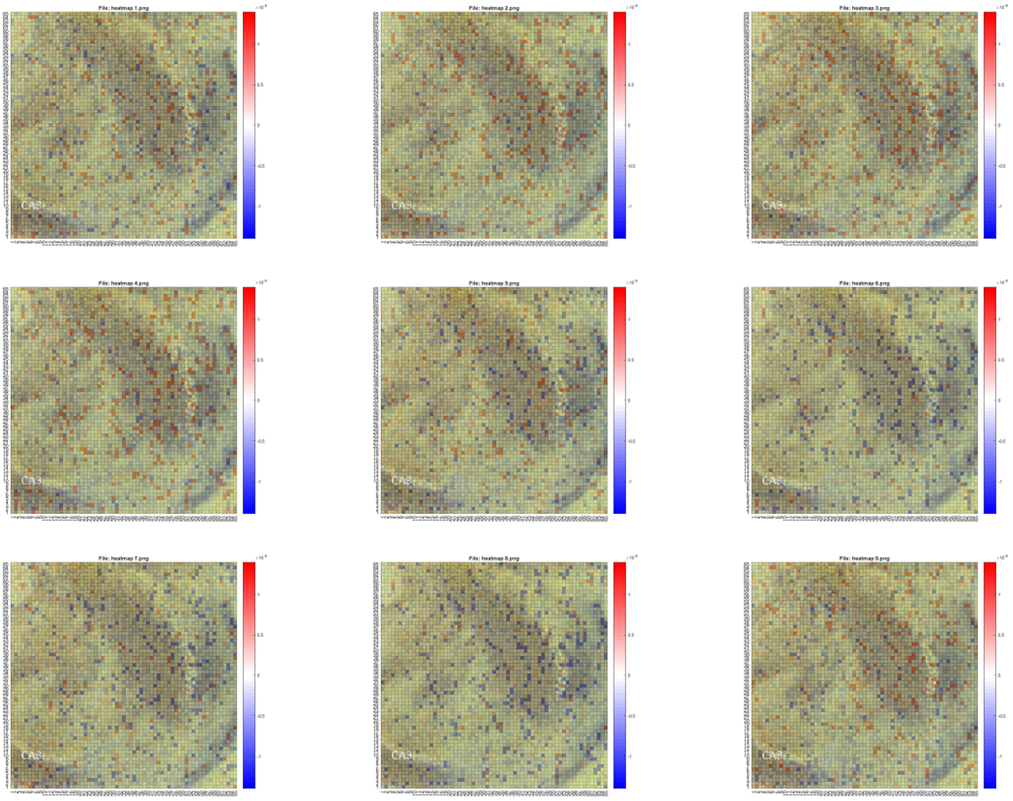


Figure 4.19: Overlay with the heatmap from Figure 4.15 and Figure 4.13, start time is 5 seconds the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.6.3 Cultured Neurons

The overlay shows the results of the heatmap and the dissociated neurons picture after adjustment. Figure 4.20 shows the results. Neurons should be fluorescently labelled to make them more visible on the dark surface. Start time was 5 seconds and the frame rate was 2 ms.

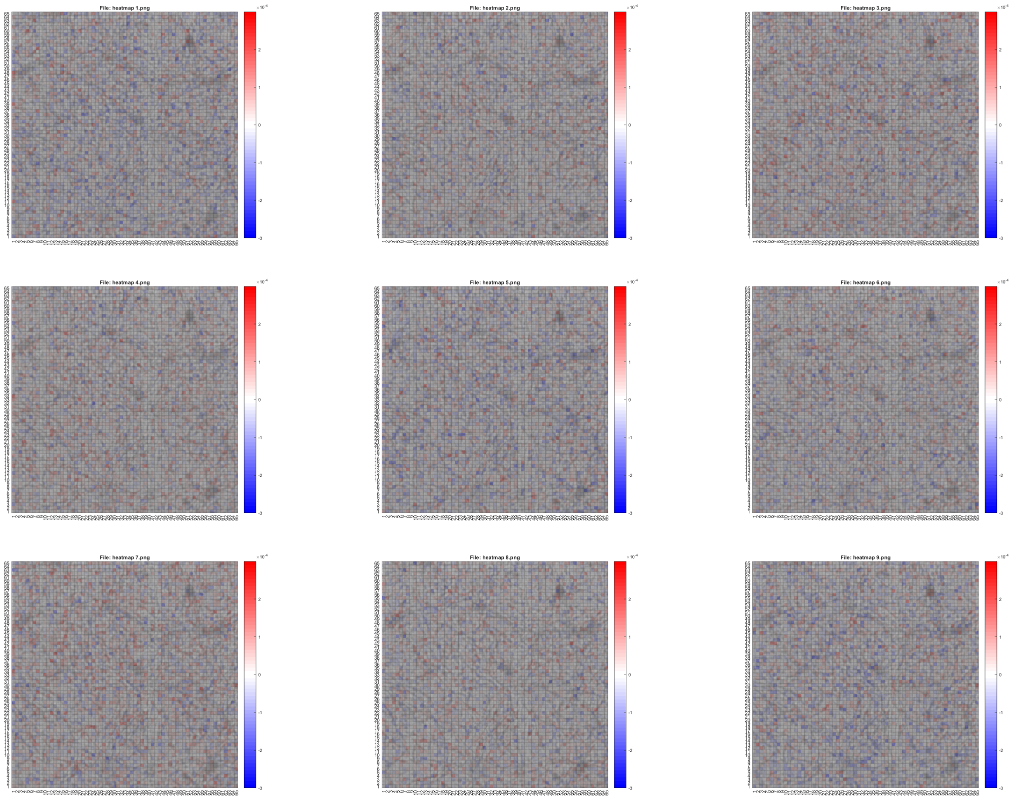


Figure 4.20: Overlay with the heatmap from Figure 4.16 and Figure 4.10, start time is 5 seconds the frame rate was 2 ms. The first frame is shown top left, the last frame is at the bottom right.

4.3.6.4 60 Electrode MEA

The results of the overlay of the 60 electrode MEA is shown in Figure 4.21. Dissociated neurons were cultured on the chip. Only six images are displayed, corners are empty no electrodes present there.

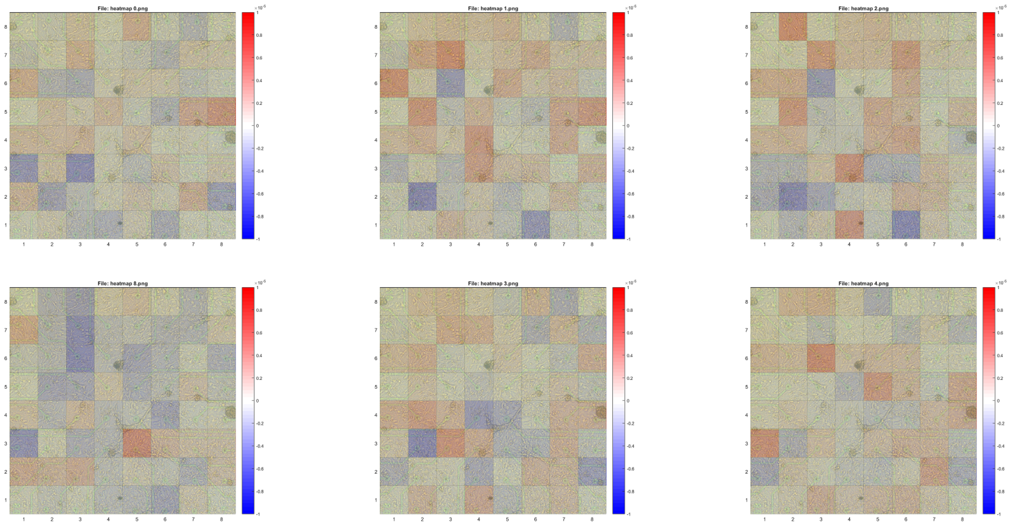


Figure 4.21: Overlay with the heatmap from Figure 4.17 and Figure 4.11, start time is 5 seconds the frame is a average over quite a few data lines. The first frame is shown top left, the last frame is at the bottom right.

4.3.7 Video Output

The video shows the series of images from the headline overlay, for a hole data set. It is possible to generate the video output for CMOS-MEA and 60 electrode MEA.

5 Discussion

This chapter debates the problems and occurrences, that arose during the thesis, but also mentions positive outcomes and successful parts. Sections are divided in measurement, image recording, a program part which is related to the Matlab code and the outcome after merging all data, which is split into the measurements equipped with the numbers of the results. The results were discussed and interpreted.

5.1 MEA Measurement

Recording data with the system was quite difficult and it represented a challenge to measure action potentials at all, which could either be triggered or spontaneously occurring, in hippocampal slices or dissociated neurons. Furthermore, the setup was complicated to use for slices, as the perfusion over the slice and the vacuum had to be coordinated to achieve a laminar flow, where not too much or too little fluid remained in the array. The adjustment of the tubes was hard, as besides, the chip had to be covered with metal to reduce noise. Air bubbles in the tube spreading over the array, can easily be misinterpreted as spikes. Mounting the slices was another dare to fix them on the chip with enough pressure to keep them in place and not to destroy them.

The dissociated neurons had the advantage that no perfusion was needed. However, the challenge here was that they needed two weeks of growth before enough organic connections developed. During this time, they needed to stick and when medium was changed, one had to be careful to not absorb the neurons. The coating was also important, as otherwise they the neurons did not remain on the array and no connection was established. Despite coating, sometimes no connections between cells and surface were formed.

5.2 Image Recording

Hippocampal slices are vulnerable to dehydration. As the buffer solution, which is provided during the measurement, had to be removed almost completely to focus the slice and the array, this represented a risk factor of drying the hippocampal slices. This fact makes the recording highly time dependent and required a sophisticated process to reduce the duration. An additional problem for the slices was that at least two pictures were required one with focus on the structure and the other one with focus on the array. When the image of the array was used to detect the chip surface, the location was taken to the second picture and used for it. For detailed determination, it would have been necessary to know the exact distance between the objective lens and the array, as well as the distance between the slice, the array and the objective.

For dissociated neurons the image capturing was not a time problem as cells were constantly covered by nutrient solution and the dehydration factor was not given. It was rather a temperature problem, when the neurons were put out of the incubator, as a relatively constant temperature was needed to ensure their survival. During measurements or preparations, there was a safety membrane which was sealing the array. This membrane had to be removed for the image recording, as otherwise it would not have been possible to focus the MEA chip. This is leading into another problem as long as the safety membrane is removed, the possibility for contamination is available.

5.3 Code

The code in Matlab started straight forward. First, there was the read in of the microscope images, followed by testing, whether a rotation was needed. The rotation was done with bilinear filtering, and cropping the image to the same size as the input. The cutting was done by analysing the picture and the pixels. By doing this, the knowledge whether a rotation was needed or not was delivered for both measurements.

Afterwards, the heatmap was created to visualize the data from the CMOS-MEA system and the 60 electrode MEA-system. The data points were saved into a matrix and then read via a loop. The created heatmaps were afterwards merged with the microscope image for the overlay, which was done for every picture, followed by the generation of a video. Such video could be generated for every measured data line or with an arbitrary number of data. In case too many data frames are required, the code would need to be optimized for time and memory, this will be the next step for time efficiency. Also the read in from the MEA data could have been done segment wise and maybe without using too much memory space merged to a video.

5.4 Outcome of the Code

The overlay of the program looks nice, but it is not clear how good it will fit with real data. This would be the next step and then it would be possible that some adaptions for the heatmap is needed. When the MEA-system measures correct values, then it will show the accuracy of the program and where weaknesses are. When the measurements show good results, the spike sorting file can be investigated how good it fit, with the Matlab code.

5.4.1 Organotypic Slice Measurement 1

The heatmap and the overlay fitted really good, however, a problem was the captured slice image. The brightness of the image is low also the quality, changing the transparency makes it difficult to see the slice. One solution to overcome these limitations would be to change the image into gray scale. Nonetheless, still the better way is to record pictures carefully and focus only parts of it. The results show that we were not able to record spontaneous spiking activity or response to stimuli, but the acquired data were still useful to test the whole procedure. For the first example, a stimulus was applied in the bottom left corner it is also possible to see a wave propagation over the chip, but this result could not be correct as the slice was on the wrong position. The shown propagation could be from the solution on the chip.

5.4.2 Organotypic Slice Measurement 2

In this experiment the spike sorting algorithm was used to detect the spontaneous spikes, where the created region of interest showed interesting results. The map of spikes was a number of action potentials that was measured all over the chip. The problem was, we measured signals but it was not clear if the spikes were background noise or triggered local field potentials. During sampling no stimulus was applied to induce an action potential. The slice image showed really good structures, however, the focus was not changed to see the array. Rotation and trimming were not easy, as a second picture was needed.

5.4.3 Dissociated Neurons

On the dark surface of the chip array it is difficult to see the neurons. For that reason, it was planned to use fluorescence to make the neurons visible and would be the next steps. The white vertical stripes in the heatmaps were caused by bad connections between the chip and the headstage. When the chip was placed in the headstage, gaps between the electrode needles and contact points could occur, or the chip connection was soiled. The problem could easily be solved and resulted in a still functional chip. In this setup, no voltage to trigger an action potential was applied. Rather, it was only a search for spontaneous action potentials activity. The organic connections were formed, which could be seen on the microscope image. Unfortunately, when the acquired data were investigated it was not possible to see spontaneous action activity in the data points.

5.4.4 60 Electrode MEA

The electrode array of the 60 electrode device was different, because of the increased space between the electrodes. This means, that measured signals were always a combination of several neurons. The exact edges were also not indefinable, as in the CMOS arrays. The results unveiled, that it was not possible to measure a wave propagation across the chip, which could be caused by the fact, that not enough organic connections have been formed. Another reason could be that the impulses were to weak. The chip is missing electrodes in all corners, which had to be considered when the matrix was created in Matlab. Maybe the signals would be easier to measure with slices, but also here it will not be possible to do neuron mapping. In this measurement also no stimulus was applied.

6 Conclusion

First of all, the measurements have to deliver useful results, to proof the Matlab code and the outcome. For now, it is not clear whether the coating forms isolation between the slices or neurons and the chip, hence, signal may improve with different coating the chip prior to cell seeding. For slices, there could be more than one problem which cause the failing of the measurement. The slices were good-looking when they were taken from the incubator, however, no useful signals were measured when voltage was applied on certain areas of the chip. Also waiting for spontaneous action potentials was not successful. Interference may be caused by the perfusion above the chip as well as by the fixation, although the reference electrode and the metal cover removed most of the noise.

Also the microscopy could be improved. First, the fixation of the chip on the microscope table could be changed, allowing an easier placement of the chip to yield a better overlay. Second, the objective could be changed. Currently, one magnification is too small while the other is too big. For the 60 electrode MEA system it is also possible to use transmission light. The focus of the different planes of the slices and array is important to get sharp structures and edges. For neurons, fluorescence staining seemed interesting and was planned to do, but due to time troubles it was not possible. By image acquisition of fluorescence labelled neurons the overlay might become more clear.

On the one hand, the MEA software for spike sorting is a really nice tool and assisted a lot to identify interesting parts of the measurements. On the other hand, the algorithm will only be usable with valuable results.

The Matlab code could be expanded in this way, also that a single electrode in the heatmap can be selected and the signal path is shown. The only problem is the size of the file, which could be avoided by reading the file segment wise. Images with sharp edges can be used with the edge detection tool and the rotation and cutting will be done automatically. For neurons the fluorescence will open new possibilities, to see the neurons on the dark surface of the CMOS array. When all the steps are done the single neuron mapping should be possible, especially for slices. Dissociated neurons can be assigned to the electrodes on the chip.

Appendix

Possible buffer solution for hippocampal slices, it is not ideal for rats, more for mice suitable.

Chemical	**M.W.**	**Conc. [mM]**	**g/1L 10x Stock**	**g/2L 10x Stock**	**g/500mL ACSF**	**g/1L ACSF**
NaCl	58.44	124	72.50	145.00		
KCl	74.55	2.5	1.86	3.72		
$NaH_2PO_4\cdot 2H_2O$	156.01	1.2	1.87	3.74		
$MgSO_4\cdot 7H_2O$	246.47	2	4.92	9.84		
$CaCl_2\cdot 2H_2O$	147.01	2	2.94	5.88		
$NaHCO_3$	84.01	24			1.01	2.02
HEPES	238.31	5			0.60	1.19
$Glucose\cdot H_2O$	198.17	12.5			1.24	2.48

Table .1: ACSF Buffer Solution with HEPES for young mice. [9]

Matlab code to create the heatmap and overlay from results. This code can be used for hippocampal slices and dissociated neurons, only a small adjustment in the code had to be done in section marking of regions. This code provides a GUI for better handling, it is shown in the results.

```
function varargout = Heatmap_overlay_gui(varargin)
% HEATMAP_OVERLAY_GUI MATLAB code for Heatmap_overlay_gui.fig
%      HEATMAP_OVERLAY_GUI, by itself, creates a new HEATMAP_OVERLAY_GUI orraises the existing
%      singleton*.
%
%      H = HEATMAP_OVERLAY_GUI returns the handle to a new HEATMAP_OVERLAY_GUI or the handle to
%      the existing singleton*.
%
%      HEATMAP_OVERLAY_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in HEATMAP_OVERLAY_GUI.M with the given input arguments.
%
%      HEATMAP_OVERLAY_GUI('Property','Value',...) creates a new HEATMAP_OVERLAY_GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Heatmap_overlay_gui_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Heatmap_overlay_gui_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Heatmap_overlay_gui

% Last Modified by GUIDE v2.5 31-Mar-2020 09:55:20

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
                      @Heatmap_overlay_gui_OpeningFcn, ...
                   'gui_OutputFcn',
                      @Heatmap_overlay_gui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code — DO NOT EDIT


% ——— Executes just before Heatmap_overlay_gui is made visible.
function Heatmap_overlay_gui_OpeningFcn(hObject, eventdata,
   handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved — to be defined in a future version of
   MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Heatmap_overlay_gui (see
   VARARGIN)

% Choose default command line output for Heatmap_overlay_gui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Heatmap_overlay_gui wait for user response (see
   UIRESUME)
% uiwait(handles.figure1);


% ——— Outputs from this function are returned to the command line.
```

```
function varargout = Heatmap_overlay_gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% ——— Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
TStart = str2double(get(handles.StartTime, ’string’));
TSeg = str2double(get(handles.SegTime, ’string’));
VRange = str2double(get(handles.Vrange, ’string’));
loop_number = str2double(get(handles.Loop_number, ’string’));
sample_space = str2double(get(handles.sample_space, ’string’));
trans = str2double(get(handles.Trans, ’string’));;
heatmap_overlay(TStart, TSeg, VRange, loop_number, sample_space, trans);






function StartTime_Callback(hObject, eventdata, handles)
% hObject    handle to StartTime (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,’String’) returns contents of StartTime as text
%        str2double(get(hObject,’String’)) returns contents of StartTime as a double


% —— Executes during object creation, after setting all properties.
function StartTime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to StartTime (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,’BackgroundColor’), get(0,’defaultUicontrolBackgroundColor’))
    set(hObject,’BackgroundColor’,’white’);
end



function SegTime_Callback(hObject, eventdata, handles)
% hObject    handle to SegTime (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,’String’) returns contents of SegTime as text
%        str2double(get(hObject,’String’)) returns contents of SegTime as a double


% —— Executes during object creation, after setting all properties.
function SegTime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SegTime (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,’BackgroundColor’), get(0,’defaultUicontrolBackgroundColor’))
    set(hObject,’BackgroundColor’,’white’);
end
```

```



function Loop_number_Callback(hObject, eventdata, handles)
% hObject    handle to Loop_number (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Loop_number as text
%        str2double(get(hObject,'String')) returns contents of Loop_number as a double


% ——— Executes during object creation, after setting all properties.
function Loop_number_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Loop_number (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function sample_space_Callback(hObject, eventdata, handles)
% hObject    handle to sample_space (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sample_space as text
%        str2double(get(hObject,'String')) returns contents of sample_space as a double

% --- Executes during object creation, after setting all properties.
function sample_space_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sample_space (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in Transperancy.
function Transperancy_Callback(hObject, eventdata, handles)
% hObject    handle to Transperancy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
pop_strings = get(pop_menu, 'String');
pop_val = get(pop_menu, 'Value');
result = pop_strings{pop_val};
% Hints: contents = cellstr(get(hObject,'String')) returns Transperancy contents as cell array
%        contents{get(hObject,'Value')} returns selected item from Transperancy


% --- Executes during object creation, after setting all properties.
function Transperancy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Transperancy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,’BackgroundColor’), get(0,’defaultUicontrolBackgroundColor’))
    set(hObject,’BackgroundColor’,’white’);
end



function Vrange_Callback(hObject, eventdata, handles)
% hObject    handle to Vrange (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,’String’) returns contents of Vrange as text
%        str2double(get(hObject,’String’)) returns contents of Vrange as a double


% ——— Executes during object creation, after setting all properties.
function Vrange_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vrange (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,’BackgroundColor’), get(0,’defaultUicontrolBackgroundColor’))
    set(hObject,’BackgroundColor’,’white’);
end

function heatmap_overlay(TStart, TSeg, VRange, loop_number, sample_space, trans)
tic % tic — toc commands to measure elapsed time

%% Possible Parameter Definitions
%TStart = 5;      % Start time of observed data segment
%TSeg = 0.2;      % Length of data segment

% Displayed Voltage Range for Heatmap
```

```matlab
%VRange = 10e-6;

%loop_number = 501; %definition for number of heatmaps
%sample_space = 10;  %space between data lines, every 10th is taken
%trans = 0.3;
%% Constants
TScale = 1e-6;      % Time scale of data [ s ]

imScale = 10;    % Scale Factor for Image Size (later Multiplied by Number of Electrodes)

%coordinates for marking of region x and y values
mark_x = 280;
mark_y = 510;

%% Read File

fileName = '2019.03.19-11.09.59-Rec_fixed.cmcr';  % Define file name
file = ['data/',fileName];                     % Add file path

% Read Info about Sensor Data from h5 File Structure
info_sdata = h5info(file,'/Acquisition/Sensor Data/SensorData 1 1');
N_electrodes = info_sdata.Dataspace.Size(1);    % Number of electrodes per row/column
DataSize = info_sdata.Dataspace.Size(3);        % Number of datapoints per electrode

% Read General Info from h5 File Structure
info = h5info(file,'/Acquisition');
fSample = info.Groups(1).Datasets(1).ChunkSize(1);  % Sample rate
T = double(info.Attributes(9).Value) * TScale;      % Recording time (converted to [s])

% Read Sensor Meta Data
metaData = h5read(file,'/Acquisition/Sensor Data/SensorMeta');
% Get conversion factors for individual electrodes
convFactor(1,:) = sscanf(metaData.ConversionFactors{1},'%f');
convFactor = reshape(convFactor,[N_electrodes,N_electrodes]);
% Get voltage scaling factor
VScale = 10^double(metaData.Exponent);

```

```
% Get Indices for Start and End of the predefined Time Window
segSize = TSeg * fSample + 1;
startInd = TStart * fSample + 1;

% Store Sensor Data of all Electrodes in the predefined Time Window in a 3-Dimensional Array
    % X and Y Coordinates are switched compared to depiction in CMOS-MEA-Tools!
data = h5read(file,'/Acquisition/Sensor Data/SensorData 1 1', ...
       [1 1 startInd],[N_electrodes N_electrodes segSize]);


%% Convert Data

% Define Time Variable
t = TStart + (0 : 1/fSample : TSeg);

% Switch X and Y Coordinates of Data and Adjust Values to [V] through
% Multiplication with the Voltage Scale and the Individual Conversion Factors
for i = 1 : N_electrodes
    for j = 1 : N_electrodes
        sig(i,j,:) = double(data(j,i,:)) * convFactor(j,i) * VScale;
    end
end

%% Folder Path
%folder to save heatmap, when active
folder='C:\Users\Irmgard\Documents\MATLAB\MEA\Heatmap';
% Load a Custom Colormap from File
load('MyColormap.mat','mycmap');

%folder overlay, for overlay images and video
folder_overlay='C:\Users\Irmgard\Documents\MATLAB\MEA\Overlay';

%% Image Loading and Adjustment
 MicroscopeImage = imread('2019.03.19-11.09.59-Rec_picture.jpg');

%command to rotate the image, the picture has to be investigated to see
%when the array is parallel
HorizontalAdjust = imrotate(MicroscopeImage, 2.5,'bilinear');
```

```
 %figure;
 %imshow(HorizontalAdjust)
%to trimm the picture, the dimension is adjusted to the array
cropImage = imcrop(HorizontalAdjust,[439 171 1343-439 1067-171]);
%resize to get the same dimension like the heatmap
resizeImage = imresize(cropImage,N_electrodes*imScale*[1 1]);
 %figure;
% imshow(resizeImage)
%% Marking of Region
% insert text on finishImage for marking regions, when used in making
% transparent the input finishImage has to be replaced with MarkingImage
mark_region = 'CA3';
%x/y coordination defined in the top, also size of font and color
MarkingImage = insertText(resizeImage,[mark_x mark_y],mark_region, ...
     'FontSize', 30, 'BoxOpacity', 0, 'TextColor', 'white');

     % figure;
     %  imshow(MarkingImage)
%% Create Heatmap, every Dataline and Overlay with Image
n = 1; %counter for the number of heatmaps

for r = 1 : sample_space : loop_number

% Open New Figure and Adjust its Size
f = figure('visible', 'off');

%define the size, when changed the image has to be investigated to get the
%new pixels size and the coordinates for the overlay
set(gcf,'Position',[500, 100, 700, 600],'Color','White')
%define the a invisible plot, over the heatmap
axes('Position', [0, 0.97, 1, 0.05]) ;
set(gca, 'Color', 'none', 'XColor', 'none', 'YColor', 'none') ;

% Adjust Size and Position of the Heatmap
axes('position',[0.115 0.11  0.83 0.83]);

% Plot Heatmap
h = heatmap(sig(:,:,r));

h.Colormap = mycmap;                % Load custom colormap
```

```
h.ColorLimits = [-VRange VRange];    % Adjust voltage range
h.FontSize = 8;                      % Adjust Font Size

% Adjust Heatmap Orientation (electrode (1,1) in bottom left corner)
h.YDisplayData = N_electrodes + 1 - (1:N_electrodes);

% Adjust Heatmap Shape so it is always Square
h.Units = 'centimeters';
h.Position(3:4) = min(h.Position(3:4))*[1,1];

% File Name in Plot Title, saving heatmap and change to png
h.Title = (['File: heatmap ', num2str(n), '.png']);
%h =  figure;

ax = axes;
ax.Color = 'none';
ax.XTick = [];
ax.YTick = [];

% Adjust Position and Size of Axes to have same Dimensions as Heatmap
ax.Position = [0.115,0.11,0.711,0.857];

 %% Transparent and Showing in one Picture

%alphadata change the transparent form the microscope image when slices are
%used the command marking of regions has to be activated and MarkingImage
%instead of finishImage
im = image(MarkingImage,'AlphaData', trans);
%shift brain image in x/y direction, for an accurat overlay, when heatmap
%size is changed the values has to be adapted
 xdata = get(im,'XData');
 ydata = get(im,'YData');
 set(im,'XData',xdata+1,'YData',ydata+20);
 %switching transparent
axis off

n = n+1;
 %saving image in file, for each loop iteration
 imageName = sprintf('Overlay#%d.png', r); % Create filename.
 folderName_over = fullfile(folder_overlay, imageName); %folder path
 saveas(f,folderName_over);  %save image in folder
end


%% Create Video, Images from Overlay
%name the video
name_video = 'Overlay';
vfilename = [name_video '.avi'];  %video format
         video = VideoWriter(fullfile(folder_overlay,vfilename));%create the video object
         video.FrameRate = 11;
         open(video);                 % open the file for writing
         for x = 1 : sample_space : loop_number
           filename = [name_video, sprintf('#%d',x) '.png'];  %chose images for video
           I = imread(fullfile(folder_overlay,filename)); % read the next image
           writeVideo(video,I);      % add the image to the video file
         end
         close(video);
%%
toc % Display elapsed time since program was started



function Trans_Callback(hObject, eventdata, handles)
% hObject    handle to Trans (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Trans as text
%        str2double(get(hObject,'String')) returns contents of Trans as a double


% ——— Executes during object creation, after setting all properties.
function Trans_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Trans (see GCBO)
% eventdata  reserved — to be defined in a future version of
```

```
    MATLAB
% handles    empty — handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,’BackgroundColor’), get(0,’
    defaultUicontrolBackgroundColor’))
    set(hObject,’BackgroundColor’,’white’);
end
```

Matlab code for the 60 electrode MEA, it is also usable for slices and cultured neurons. The adjustment had to be done in the section marking of region.

```
%%Colormap 60 electrode MEA

close all
clear all
tic % tic - toc commands to measure elapsed time


TStart = 5;      % Start time of observed data segment
TSeg = 0.2;      % Length of data segment


% Displayed Voltage Range for Heatmap
VRange = 10e-6;

loop_number = 5; %definition for number of heatmaps
sample_space = 1;  %space between data lines, every 10th is taken
trans = 0.8;

%% Heatmap
%folder to save heatmaps
folder='C:\Users\Irmgard\Documents\MATLAB\MEA\Heatmap_60';
% Load a Custom Colormap from File
load('MyColormap.mat','mycmap');
%folder save overlay and video
folder_overlay='C:\Users\Irmgard\Documents\MATLAB\MEA\Overlay_60';

%% Constants
TScale = 1e-6;       % Time scale of data [ s ]

%% Constants
N = 60;              % Total number of electrodes
refElectrode = 15;   % Position of reference electrode


%% File Name
file01 = '2020-02-11T09-39-45McsRecording.h5';

%% Data Import
info01 = h5info(file01);
T01 = double(info01.Groups.Groups.Attributes(4).Value * 1e-6);    %
```

```
    Measurement duration in [s]

rawData = h5read(file01,'/Data/Recording_0/AnalogStream/Stream_0/
    ChannelData');
temp = h5read(file01,'/Data/Recording_0/AnalogStream/Stream_0/
    InfoChannel');
rawDataLabels = str2double(temp.Label);
rawDataBits = temp.ADCBits(1);



fSample01 = length(rawData) / T01;
t01 = 0 : 1/fSample01 : T01 - 1/fSample01;
%VScale = 10^double(rawData.Exponent);

%% Put Data in Right Order
data01 = zeros(size(rawData));
temp = rawDataLabels;
temp(find(isnan(rawDataLabels))) = refElectrode;

for i = 1:60    % Put electrode data in right order
    [a,j] = min(temp);
    data01(:,i) = double(rawData(:,j)) / double(2^rawDataBits);
    temp(j) = 100;

end


%% slice image loading and editing
 MicroscopeImage = imread('MEA_60electrode_2020_02_11.jpg');
%figure;
%imshow(MicroscopeImage)
%brain_rotate = imrotate(brain_slice, 270, 'loose');  %rotate the
    slice image to get the correct overlay
HorizontalAdjust = imrotate(MicroscopeImage, 0,'bilinear'); %
    should be in the last version not needed because the crope
%          %command will cut out everything outside the chip array
%figure;
%imshow(HorizontalAdjust)

cropImage = imcrop(HorizontalAdjust,[544 195 1221-544 879-195]);
resizeImage = imresize(cropImage,[763 761]);

finishImage = imrotate(resizeImage, 270, 'bilinear', 'crop');
```

```
figure;
imshow(finishImage);

%% Marking of Region
% insert text on finishImage for marking regions, when used in making
% transparent the input finishImage has to be replaced with MarkingImage
% mark_region = 'CA3';
% MarkingImage = insertText(finishImage,[450 530],mark_region,...
%      'FontSize', 30, 'BoxOpacity', 0, 'TextColor', 'white');
%    figure;
%     imshow(brain_image)


%%
  % n create 6 heatmaps over time
   for n = 0:9
      column = 1;
      z_mea = 1;    %counter for the columns of the data signals
      count=1;      %internal counter for the missing electrodes 1,8,57,64
      while column < 9

      row = 1;
      while row < 9
          %pushing the values from the file in the matrix
          if count == 1 || count == 8 || count == 57 || count == 64
              array(row,column) = NaN;  %no values for the edges, no electrodes in the 60 MEA
          else
          sig(row,column) = mean(data01(10001+1000*n:10000+1000*(n+1),z_mea)); %take mean and abs from 10000 data points
          z_mea = z_mea+1;
          end
          row = row +1;
          count = count+1;
      end
      column = column +1;

      end
      f = figure('visible', 'off');
      %define the size, when changed the image has to be
         investigated to get the
      %new pixels size and the coordinates for the overlay
       set(gcf,'Position',[500, 100, 700, 600],'Color','White')
       axes('Position', [0, 0.97, 1, 0.05]) ;
       set(gca, 'Color', 'none', 'XColor', 'none', 'YColor', 'none
          ') ;

       % Adjust Size and Position of the Heatmap
       axes('position',[0.115 0.11  0.83 0.83]);

       % Plot Heatmap
       h = heatmap(sig(:,:),'CellLabelColor','none');
       h.Colormap = mycmap;                 % Load custom colormap
       h.ColorLimits = [-VRange VRange];    % Adjust voltage range
       h.FontSize = 8;                      % Adjust Font Size
       % Adjust Heatmap Orientation (electrode (1,1) in bottom
          left corner)
       h.YDisplayData = 8 + 1 - (1:8);

       % Adjust Heatmap Shape so it is always Square
       h.Units = 'centimeters';
       h.Position(3:4) = min(h.Position(3:4))*[1,1];
       h.Title = (['File: heatmap ', num2str(n), '.png']);

       ax = axes;
       ax.Color = 'none';
       ax.XTick = [];
       ax.YTick = [];

       % Adjust Position and Size of Axes to have same Dimensions
          as Heatmap
       ax.Position = [0.114,0.11,0.711,0.852];

       %% making transparent and showing in 1 picture

        %alphadata change the transparent form the microscope
           image when slices are
        %used the command marking of regions has to be activated
           and MarkingImage
        %instead of finishImage
        im = image(finishImage,'AlphaData', trans);
        %shift brain image in x/y direction, for an accurat
           overlay, when heatmap
        %size is changed the values has to be adapted
         xdata = get(im,'XData');
         ydata = get(im,'YData');
         set(im,'XData',xdata+1,'YData',ydata+20);

         axis off

         n = n+1;
         %saving image in file, for each loop iteration
         imageName = sprintf('Overlay#%d.png', n); % Create filename.
         folderName_over = fullfile(folder_overlay, imageName); % folder path
         saveas(f,folderName_over);  %save image in folder

   end

%% make video, taking images from overlay and create the video
%
name_video = 'Overlay';
vfilename = [name_video '.avi'];
         video = VideoWriter(fullfile(folder_overlay,vfilename));% create the video object
         video.FrameRate = 10;
         open(video);                % open the file for writing
         for x = 1 : n
           filename = [name_video, sprintf('#%d',x) '.png'];
           I = imread(fullfile(folder_overlay,filename)); % read the next image
           writeVideo(video,I);      % add the image to the video file
         end
         close(video);

toc % Display elapsed time since program was started
```

Reference

[1] 3Brain. "BioCAM-X-userguide." In: 45 (2019). URL: https://www.3brain.com/ (visited on 09/25/2019) (cit. on pp. 12, 13).

[2] Manfred Caeser and Ad Aertsen. "Morphological organization of rat hippocampal slice cultures." In: *The Journal of Comparative Neurology* 307.1 (May 1, 1991), pp. 87–106. ISSN: 0021-9967, 1096-9861. DOI: 10.1002/cne.903070109. URL: http://doi.wiley.com/10.1002/cne.903070109 (visited on 01/14/2020) (cit. on pp. 1, 2).

[3] S Feldt et al. "Functional clustering in hippocampal cultures: relating network structure and dynamics." In: *Physical Biology* 7.4 (Oct. 26, 2010), p. 046004. ISSN: 1478-3975. DOI: 10.1088/1478-3975/7/4/046004. URL: http://stacks.iop.org/1478-3975/7/i=4/a=046004?key=crossref.392b2cdcc8172213c894d7923f80c4fa (visited on 11/25/2019) (cit. on p. 6).

[4] *Fluorescence | Scientific Volume Imaging*. URL: https://svi.nl/Fluorescence (visited on 01/13/2020) (cit. on p. 10).

[5] *Fluorescence Microscopy - Basic Concepts in Fluorescence | Olympus Life Science*. URL: https://www.olympus-lifescience.com/en/microscope-resource/primer/techniques/fluorescence/fluorescenceintro/ (visited on 01/13/2020) (cit. on p. 10).

[6] A.A. Gladkov et al. "Functional Connectivity of Neural Network in Dissociated Hippocampal Culture Grown on Microelectrode Array." In: *Sovremennye tehnologii v medicine* 9.2 (June 2017), p. 61. ISSN: 20764243. DOI: 10.17691/stm2017.9.2.07. URL: http://www.stm-journal.ru/en/numbers/2017/2/1340 (visited on 01/14/2020) (cit. on p. 6).

[7] B. H. Gähwiler. "Organotypic monolayer cultures of nervous tissue." In: *Journal of Neuroscience Methods* 4.4 (Dec. 1, 1981), pp. 329–342. ISSN: 0165-0270. DOI: 10.1016/0165-0270(81)90003-0. URL: http://www.sciencedirect.com/science/article/pii/0165027081900030 (visited on 03/02/2020) (cit. on p. 4).

[8] *Hippocampus Basic Circuit*. URL: http://neurones.co.uk/ (visited on 01/14/2020) (cit. on p. 3).

[9] *Improved Adult Brain Slice Methods*. jonathan. URL: https://www.brainslicemethods.com (visited on 01/14/2020) (cit. on pp. 7, 58).

[10] *Introduction to Fluorescence Microscopy*. Nikon's MicroscopyU. URL: https://www.microscopyu.com/techniques/fluorescence/ (visited on 01/13/2020) (cit. on p. 10).

[11] Jenoptik. "Subra technical data sheet." In: 2 (). URL: https://www.jenoptik.de/produkte/kameras-kameramodule/mikroskopkameras/progres-gryphax-usb-30-kamera (visited on 12/2019) (cit. on pp. 9, 29).

[12] Christian Leibig, Thomas Wachtler, and Günther Zeck. "Unsupervised neural spike sorting for high-density microelectrode arrays with convolutive independent component analysis." In: *Journal of Neuroscience Methods* 271 (Sept. 2016), pp. 1–13. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2016.06.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0165027016301273 (visited on 01/22/2020) (cit. on p. 25).

[13] Mario Malerba et al. "Fabrication of Multielectrode Arrays for Neurobiology Applications." In: *Cell-Based Microarrays: Methods and Protocols*. Ed. by Peter Ertl and Mario Rothbauer. Methods in Molecular Biology. New York, NY: Springer New York, 2018, pp. 147–157. ISBN: 978-1-4939-7792-5. DOI: 10.1007/978-1-4939-7792-5_12. URL: https://doi.org/10.1007/978-1-4939-7792-5_12 (visited on 09/02/2019) (cit. on p. 13).

[14] *MathWorks – Entwickler von MATLAB und Simulink*. URL: https://de.mathworks.com/ (visited on 01/20/2020) (cit. on p. 16).

[15] Multi Channel Systems MCS. "CMOS-Applications Cortical Neuron Culture Protocol 1." In: (2018). URL: https://www.multichannelsystems.com/ (visited on 10/24/2018) (cit. on p. 6).

[16] Multi Channel Systems MCS. "CMOS-MEA5000-System-Manual." In: 163 (2019). URL: https://www.multichannelsystems.com/ (visited on 05/09/2019) (cit. on pp. 11, 12, 21, 24, 25).

[17] Multi Channel Systems MCS. "MEA-Application Note Organotypic Cultures of Hippocampal Slice Preparation." In: 28 (2013). DOI: 10.21769/BioProtoc.2605. URL: https://www.multichannelsystems.com (visited on 11/26/2013) (cit. on pp. 7, 21).

[18] Multi Channel Systems MCS. *MEA2100-System Manual*. URL: https://www.multichannelsystems.com (visited on 2020) (cit. on p. 12).

[19] Multi Channel Systems MCS. "TC01-TC02$_{M}anual_{R}evG$." In: 22 (2019). URL: https://www.multichannelsystems.com/ (visited on 2018) (cit. on pp. 11, 24).

[20] Beatrice Miccoli et al. "High-Density Electrical Recording and Impedance Imaging With a Multi-Modal CMOS Multi-Electrode Array Chip." In: *Frontiers in Neuroscience* 13 (June 25, 2019), p. 641. ISSN: 1662-453X. DOI: 10.3389/fnins.2019.00641. URL: https://www.frontiersin.org/article/10.3389/fnins.2019.00641/full (visited on 11/25/2019) (cit. on p. 6).

[21] Arata Nakajima et al. "CMOS image sensor integrated with micro-LED and multielectrode arrays for the patterned photostimulation and multichannel recording of neuronal tissue." In: *Optics Express* 20.6 (Mar. 12, 2012), p. 6097. ISSN: 1094-4087. DOI: 10.1364/OE.20.006097. URL: https://www.osapublishing.org/oe/abstract.cfm?uri=oe-20-6-6097 (visited on 08/26/2019) (cit. on p. 13).

[22] *Neuroscience Gallery \ ConnCAD.com*. URL: http://www.conncad.com/gallery/gallery/view/cultured%20hippocampal%20neurons.jpg (visited on 01/14/2020) (cit. on p. 7).

[23] Olympus. "Manual Fixed Stage Upright Mircoscope BX51WI/BX61WI." In: 16 (2016). URL: https://www.olympus-lifescience.com (visited on 06/2016) (cit. on p. 8).

[24] *Olympus BX51-P Transmitted and Reflected Polarized Light Microscope, Olympus BX51P, BX53-P*. URL: https://www.spachoptics.com/product-p/olympus-bx51p-tr.htm (visited on 01/22/2020) (cit. on p. 9).

[25] Gordon M. Shepherd. *The Synaptic Organization of the Brain*. Oxford University Press, USA, 1990. 740 pp. ISBN: 0-19-506255-8 (cit. on pp. 1, 2).

[26] L. Stoppini, P. A. Buchs, and D. Muller. "A simple method for organotypic cultures of nervous tissue." In: *Journal of Neuroscience Methods* (Apr. 1, 1991). DOI: 10.1016/0165-0270(91)90128-M (cit. on p. 4).

[27] Jeremy Welsch et al. "Organotypic Brain Cultures: A Framework for Studying CNS Infection by Neurotropic Viruses and Screening Antiviral Drugs." In: *BIO-PROTOCOL* 7.22 (2017). ISSN: 2331-8325. DOI: 10.21769/BioProtoc.2605. URL: https://bio-protocol.org/e2605 (visited on 01/22/2020) (cit. on pp. 4, 5).

[28] Wordpress. *acsf-recipes*. URL: https://clampthis.files.wordpress.com/2011/10/acsf-recipes.pdf (visited on 01/20/2020) (cit. on p. 8).

[29] *www.multichannelsystems.com | Innovations in Electrophysiology*. URL: https://www.multichannelsystems.com/ (visited on 01/20/2020) (cit. on pp. 14, 15).

[30] Günther Zeck et al. "Electrical Imaging: Investigating Cellular Function at High Resolution." In: *Advanced Biosystems* 1.11 (Nov. 2017), p. 1700107. ISSN: 23667478. DOI: 10.1002/adbi.201700107. URL: http://doi.wiley.com/10.1002/adbi.201700107 (visited on 08/26/2019) (cit. on p. 14).

[31] *ZEISS Microscopy Online Campus | Microscopy Basics | Fluorescence Microscopy*. URL: http://zeiss-campus.magnet.fsu.edu/articles/basics/fluorescence.html (visited on 01/13/2020) (cit. on p. 10).