

TECHNISCHE UNIVERSITÄT
Wien

DIPLOMARBEIT

Visualization of EMG Signals of TMR Patients

Eine Arbeit zur Erlangung des akademischen Grades
Diplom-Ingenieur

Unter der Leitung von
Ao.Univ.Prof.Dipl.Ing.Dr.techn. Eugenijus Kaniusas
E354 - Institute of Electrodynamics, Microwave and Circuit Engineering

Eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

ausgeführt von

Patrick Müller, BSc.

Mat. Nr. 0929488
Ratschkygasse 29/18
1120 Wien

Wien, am XX.XX.2013

Unterschrift des Studenten

© Copyright by
Patrick Müller, BSc.
2013

EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig, ohne fremde Hilfe angefertigt worden ist. Inhalte und Passagen, die aus fremden Quellen stammen und direkt oder indirekt übernommen worden sind, wurden als solche kenntlich gemacht. Ferner versichere ich, dass ich keine andere, außer der im Literaturverzeichnis angegebenen Literatur verwendet habe. Diese Versicherung bezieht sich sowohl auf Textinhalte sowie alle enthaltenden Abbildungen, Skizzen und Tabellen.

Wien, am XX.XX.2013

Unterschrift des Studenten

DANKSAGUNG

Zu aller erst möchte ich mich an dieser Stelle bei meinen Eltern für ihre Unterstützung während meines gesamten Studiums bedanken.

Darüber hinaus gilt mein Dank allen, an diesem Projekt beteiligten, Personen bei der Firma "Otto Bock Healthcare GmbH" und meinem Universitätsbetreuer Ao.Univ.Prof.Dipl.Ing.Dr.techn. Eugenijus Kaniusas für die gute Zusammenarbeit.

KURZFASSUNG DER ARBEIT

Visualization of EMG Signals of TMR Patients

von

Patrick Müller, BSc.

Technische Universität, Wien, 2013

Bei "Targeted Muscle Reinnervation" handelt es sich um eine Operationsmethode die bei Patienten mit einem hohen Amputationsgrad Anwendung findet, um Electromyographie-Signale zur Ansteuerung moderner Prothesen zu generieren. Ein sehr großes Problem bei der postoperativen Versorgung dieser Patienten ist das Auffinden, der für die Prothesenansteuerung am besten geeigneten Signale. Zurzeit versuchen Therapeuten durch händisches Rotieren und Umplatzieren der Elektrode das best mögliche Signal zu finden. Diese Vorgehensweise ist sehr zeitaufwändig und mühselig. Darüber hinaus kann auch nicht gewährleistet werden, dass tatsächlich das beste Signal gefunden wurde.

Ziel dieser Arbeit ist es eine Software zu entwickeln, welche durch eine geeignete Auswertung von Electromyographie-Signalen einer Elektroden-Matrix, die exakte Position des besten Electromyographie-Signals zur Prothesenansteuerung findet. Dazu wurden zwei Ansätze, bikubische Interpolation und Template Matching, verfolgt und auf ihre Eignung getestet. Bei der bikubischen Interpolation handelt es sich um eine Flächen-Interpolationsmethode, bei der die Datenwerte der Elektroden-Matrix als Stützpunkte dienen, um die Potentialverteilung in den Zwischenbereichen zu berechnen. Template Matching hingegen, verwendet Formen und Größen bereits bekannter Hotspots, um diese mit den Daten der Matrix zu vergleichen und Gemeinsamkeiten zu finden. Zum Auffinden dieser Gemeinsamkeiten wurden zwei verschiedene Methoden, die normalisierte Kreuzkorrelation und die Summe der absoluten Differenzen, untersucht. Dazu wurden reale

Daten, welche von Patiententests stammen, aber auch simulierte Daten, welche standardisierte und gleichbleibende Bedingungen schaffen, verwendet. Dabei hat sich gezeigt, dass Template Matching unter Verwendung der Kreuzkorrelation die besten Resultate liefert, was einerseits an der hohen Genauigkeit der Lokalisation und andererseits an der Unempfindlichkeit dieser Methode gegenüber Störungen liegt.

Um das Auffinden der Hotspots am Patienten zu erleichtern, wurde zusätzlich eine geeignete visuelle Darstellung der Ergebnisse entwickelt. Dabei werden die berechneten Positionen der besten Signale auf ein Foto des Patienten projiziert. Zur Datenerfassung wurde ein System aufgebaut, welches in der Lage ist 32 Electromyographie-Signale gleichzeitig aufzuzeichnen. Darüberhinaus wurden auch noch verschiedene Konzepte zum Aufbau der Elektroden-Matrix konstruiert und auf ihre Eignung getestet.

Das Resultat dieser Arbeit ist ein Software-Tool, welches in Kombination mit einer geeigneten Elektroden-Matrix zu einer einfachen und schnellen Lokalisation von Electromyographie-Signalen an der Hautoberfläche von "Targeted Muscle Reinnervation"-Patienten eingesetzt wird. Das erleichtert einerseits die Arbeit des Therapeuten und schafft eine angenehmere Vorgehensweise für den Patienten, andererseits kann auch sichergestellt werden, dass immer das best mögliche Signal gefunden wird.

ABSTRACT OF THE THESIS

Visualization of EMG Signals of TMR Patients

by

Patrick Müller, BSc.

Technische Universität, Wien, 2013

”Targeted Muscle Reinnervation” is a surgical method, which is applied to patients with a high level of amputation, in order to generate electromyography signals to control modern prosthesis. A big problem in the post-operative care of these patients is to find the exact position of the best signal, so called hotspots, for prosthesis control. Currently a therapist manually rotates and repositions an electrode in order to find this signal. Unfortunately this procedure is very slow and cumbersome and additionally one can not say, whether the most intensive signal was found or just another good one.

The aim of this work is to develop a novel approach, which uses a software-assisted interpretation of electromyography signals, derived from an electrode array, in order to determine the best electrode positions for prosthesis control. For this purpose two different approaches, bicubic interpolation and template matching, were developed and tested. The former uses a two dimensional interpolation method to compute the potential field distribution in between the electrodes, by using the data values as nodes for its calculations. The basic idea behind template matching is to make use of shape and size of a known hotspot and compare this template to the data derived from the electrodes in order to find common features. Therefore two different methods, normalized cross correlation and sum of absolute differences, were used. For this purpose real data, derived from patient tests, but also simulated data, which produces standard and constant conditions, was used. In doing so it turned out, that template matching using normalized cross

correlation showed the best results. The reason for that is the localization's high accuracy and the method's ruggedness to disturbances.

Additionally, an appropriate visualization of the results was designed, which makes the task of finding the hotspot at the patient's skin much easier. Therefore the determined hotspot position is projected to an image of the patient. Furthermore a data acquisition system was established, which is able to record up to 32 electromyography signals simultaneously. In addition to that, a lot of different electrode array build-ups were constructed and tested as well.

The basic outcome of this work is a software tool, which in combination with an electrode array is used to locate hotspots at the patient's skin in an easy and fast manner. On the one hand it simplifies the therapist's work and makes the process more pleasant for the patient and on the other hand it ensures that always the best signal will be found.

TABLE OF CONTENTS

1	Introduction	6
1.1	Motivation and Problem Definition	6
1.2	State of the Art	7
1.2.1	Targeted Muscle Reinnervation	7
1.2.2	Electromyography	10
1.2.3	Current Procedure in Therapy	15
2	Methodology	17
2.1	Build-up of the Electrode Array	17
2.1.1	Analog Digital Converters	17
2.1.2	Electrodes	18
2.1.3	Electrode Array (1. Approach)	19
2.1.4	Electrode Array (2. Approach)	21
2.1.5	Electrode Array (3. Approach)	22
2.1.6	Electrode Array (4. Approach)	23
2.2	Data Acquisition and Signal Conditioning	24
2.2.1	Data Acquisition	24
2.2.2	Signal Conditioning	26
2.3	Simulation of sEMG Signals	27
2.4	Hotspot Localization	29
2.4.1	Bicubic Interpolation	30
2.4.2	Template Matching	35
2.4.3	Dealing with faulty electrodes	41
2.5	Visualization	41
2.5.1	Visualization of sEMG Signals	41
2.5.2	Visualization of the Hotspot	43
2.6	Overview of the Software	46
3	Results	48
3.1	Hotspot Localization	48
3.1.1	Bicubic Interpolation	48
3.1.2	Template Matching	53

3.1.3	Final Decision	55
3.2	Test Using Simulated Data	58
3.2.1	Results of Sequence 1	58
3.2.2	Results of Sequence 2	59
3.3	Patient Tests - Real Data	60
3.3.1	Test Set-up and Preparations	60
3.3.2	Test Procedure	60
3.3.3	Evaluation of the Results	62
4	Conclusion	64
4.1	Basic Findings	64
4.2	Strengths and Limitations	65
4.3	Final Outcome	65
A	Software - Basic Structure And Important Methods	66
	References	80

LIST OF ABBREVIATIONS AND SYMBOLS

ACh	Acetylcholine
ADC	Analog Digital Converters
ADP	Adenosine Diphosphate
AP	Action Potential
ATP	Adenosine Triphosphate
ARV	Average Rectified Value
BI	Bicubic Interpolation
CNS	Central Nervous System
CSV	Comma Separated Values
DAQ	Data Acquisition
DC	Direct Current
DOF	Degree of Freedom
ECG	Electrocardiography
EMG	Electromyography
FEA	Finite Element Analysis
FT	Fast Twitch
GUI	Graphical User Interface
HS	Hotspot
ID	Identity
MRI	Magnetic Resonance Imaging
MVC	Maximum Voluntary Contraction
NCC	Normalized Cross Correlation
P	Phosphate
PNS	Peripheral Nervous System
RGB	Red Green Blue

RIC	Rehabilitation Institute of Chicago
RMS	Root Mean Square
RT	Real Time
sEMG	Surface Electromyography
SAD	Sum of Absolute Differences
ST	Slow Twitch
TM	Template Matching
TMR	Targeted Muscle Reinnervation
XNA	Not acronymed. It is a set of tools with a managed runtime environment. a_{ij}
a_{ij}	16 coefficients for bicubic interpolation.
C	Matrix, which describes all data points of the data set.
$D(t)$	Activation matrix in generativ model for sEMG.
DS	Data set matrix.
E	Electrodes matrix for template matching.
$F(t)$	Force function in generativ model for sEMG.
f	Function value of a corner node of the unit square.
f'_x	Derivative in x-direction of a corner node of the unit square.
f'_y	Derivative in y-direction of a corner node of the unit square.
f'_{xy}	Cross derivative in xy-direction of a corner node of the unit square.
$G(t)$	Mixture matrix in generativ model for sEMG.
$GV_{x,y}$	Gray values for visualization.
I	Interpolated data matrix.
$IV_{x,y}$	Interpolation values for visualization.
K	Set of the electrode value's positions for template matching.
P	Solution matrix for template matching.

$p(x, y)$	Two dimensional interpolation surface.
$p'_x(x, y)$	Derivative in x-direction of the interpolation surface.
$p'_y(x, y)$	Derivative in y-direction of the interpolation surface.
$p'_{xy}(x, y)$	Cross derivative in xy-direction of the interpolation surface.
S	Synergy matrix in generativ model for sEMG.
SAD_{score}	SAD score for validation.
T	Template matrix for template matching.
$Y(t)$	Inner EMG signal in generativ model for sEMG.
$Z(t)$	Overall sEMG signal in generativ model for sEMG.
$X_{\mathbf{cond}}$	Conditioned signal.
$X_{\mathbf{raw}}$	Raw signal.
$X_{\mathbf{scaled}}$	Scaled signal.
$X_{\mathbf{unscaled}}$	Unscaled Signal.
θ	Rotational angle of a Gaussian distribution.
μ	Expected value of a Gaussian distribution.
σ_x	Variance in x-direction of a Gaussian distribution.
σ_y	Variance in y-direction of a Gaussian distribution.

CHAPTER 1

Introduction

To get an easier introduction to this thesis, this chapter provides the reader with some basics about physiology and some technical background to the topics related to this work. In addition to that, the motivation behind this work is mentioned. Subsequently, the subject area and problem definition is explained and the second subchapter describes the state of the art of all work related topics.

1.1 Motivation and Problem Definition

Unfortunately, there are not any reliable statistical data about amputees, neither in Austria nor any other comparable country in central Europe. Nevertheless, statistics from the United States of America show that about 0.5% of the population live with limb loss [1][2]. Possible causes for that are usually trauma like industrial accidents, motor vehicle collisions, or war injuries but also diseases like congenital deformity, vascular diseases, or cancer.

The motivation behind this work but also every other work related to prosthetics is to provide people with physical disabilities due to limb loss with a higher quality of living. Since this thesis only deals with amputations of the upper extremity, the focus of motivation is on this area in particular. In addition to the huge functional impairment an amputee has to deal with, there is also an emotional impact. Especially the loss of an arm or a hand will have a profound social impact as well. Causes for this are deficits of skills in gesture and social interactions with other people such as shaking hands. Supplementary, for amputees it is important to attract as little attention as possible. Therefore, the prosthesis, which approximate the human body part better and better, contributes to a higher quality of living.

The motivation behind this thesis in particular is to improve the supplying of Targeted Muscle Reinnervation (TMR) patients in order to save the patient's and the therapist's time on the one hand, and to allow a better control of the prosthesis on the other hand. A huge challenge in supplying TMR patients is finding the most intense electromyography (EMG) signal on the patient's skin. Consecutively, the most intense EMG signal is called a hotspot (HS). Although the rough position of the hotspot is known from surgery, due to the unpredictable growth of the innervating nerve, one can not tell the exact position. Therefore, the therapist has to manually reposition and rotate the electrode in order to find the

hotspot. This procedure is very time consuming and cumbersome and in addition to that one can not tell whether in fact the most intense signal was found. Thus, this work deals with finding an easier and faster approach to find hotspots on the TMR patient's skin.

To realize this project there were some limitations and general requirements respectively. First of all, the goal of this work is to find a solution for the software part of the hotspot localization problem. Nevertheless, the hardware, as a part of this project, has to be taken into consideration as well. Therefore, the electrode array is supposed to be built up of a maximum of 32 electrodes by using standardized Otto Bock raw signal electrodes. To develop a straightforward and more flexible build-up the electrode array shall be used on a multitude of patients, but not only one specific patient. To sample the electrode values two 16 channel analog digital converters have to be used and the software has to be written in C#.

1.2 State of the Art

This chapter provides a short overview of the current state of the art of all work related topics. First of all basics about TMR will be described. The second subchapter deals with the physiology behind EMG and the technical devices necessary to sample EMG signals are explained as well. At the end the current procedure in finding hotspots on the patient's skin and the disadvantages related to this procedure will be mentioned.

1.2.1 Targeted Muscle Reinnervation

TMR is a surgical method to reroute nerves, which were responsible for the signal transfer to the muscles of the sound limb and which remain in the residual limb of amputees, to intact muscles in order to get EMG signals for prosthesis control. The following subchapters provide some basic information about how muscles are controlled and about the development of TMR. In addition to that the application to patients with amputations of the upper extremities and the surgical procedure will be explained in detail.

1.2.1.1 Basics About Controlling Muscles

To get a deeper insight how TMR works one has to know some basics about how muscles are controlled by nerves. The initial point for a muscle contraction is the central nervous system (CNS) composed of the brain and the spinal cord. The motor command is emitted by the motor cortex in the brain and forwarded by the spinal cord to the spinal nerve. The spinal nerve is the conjunction between the CNS and the peripheral nervous system (PNS) and changes over to the peripheral

nerve, which innervates a muscle and consequently initiates the contraction [3][4]. Figure 1.1 shows the nerve's path from the brain to the muscle. The detailed process within the muscle which causes a muscle contraction will be described in chapter 1.2.2.2.

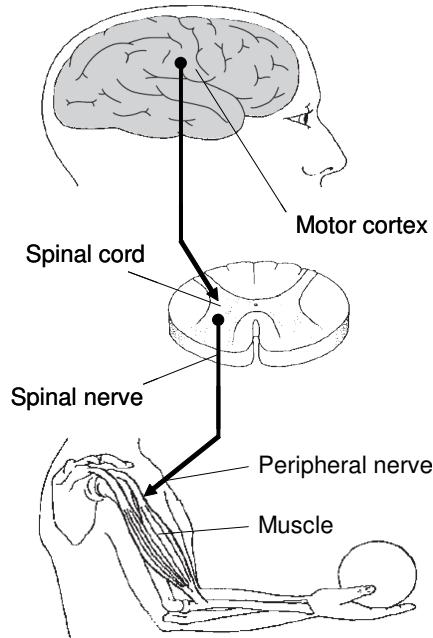


Figure 1.1: Path of a motor command from the motor cortex, via the spinal cord, the spinal and peripheral nerve to the muscle (modified from [3]).

For the principle of TMR it is very important that in case of an abscission of the limb and the nerve respectively this chain of command from the motor cortex to the end of the nerve still works. Accordingly, the nerve's signal is still available, although the nerve's targeted muscle does not exist anymore [5].

1.2.1.2 The History of TMR

First of all the reason for developing TMR has to be explained. A huge challenge in supplying patients with a high level of amputation like transhumeral amputations and shoulder disarticulations is finding EMG signal sources for prosthesis control. Due to the less remaining muscles one can not find enough signals and therefore is not able to provide sufficient control for modern prosthesis with a high amount of degrees of freedom (DOFs) [6]. One approach is to utilize a remaining agonist antagonist muscle pair in order to get a bidirectional movement, like elbow flexion and extension. To generate more movements from a single muscle pair co-contractions of the muscles are used to switch from one mode to another. Thus an amputee is able to perform elbow, rotational hand, and grasping movements by using just two EMG signals [7] [8]. Nevertheless, not being able to execute those

movements at the same time is a huge disadvantage, which makes the prosthesis handling slow and cumbersome [9] [10]. Therefore the idea of TMR arose in order to be able to control more DOFs at the same time.

TMR has been developed by Dr. Todd Kuiken and his team at „Rehabilitation Institute of Chicago“ (RIC). As previously mentioned the idea behind TMR is to reroute the nerves in the residual limb, which still provide signals [5], to a biological amplifier (a muscle) in order to be able to record the emerging EMG signals [11] [12]. This rerouting of the nerves is necessary because it is very costly to detect nerve signals straight and without any biological amplification and in addition to that a straight nerve signal measurement has to be invasive. In order to reinnervate targeted muscles with the nerves a surgery is required. Therefore some necessary preparation on the targeted muscles has to be done (see more details in chapter 1.2.1.3). Right now this procedure has been used on patients with a high amputation level (transhumeral amputation and shoulder disarticulation), but in the future maybe patients with lower amputation levels would benefit from a TMR surgery as well. So the aim of TMR is to gain more EMG signals for prosthesis control in order to be able to execute multiply movements simultaneously.

1.2.1.3 Operative Procedure

The first step is to define which muscles are used as new targeted muscles. Depending on the level of amputation different muscles are used. In addition to that the location of the muscle and the muscle’s function affect the selection as well. Therefore on the one hand muscles as close to the amputation as possible are chosen, but on the other hand the muscles have to have as little function as possible in order to not have any impact on other movements. In case of patients with shoulder disarticulations musculus pectoralis major is used, whereas residual upper arm muscles are chosen in case of transhumeral amputees [13].

The first step during surgery itself is denervating the targeted muscle by cutting the naturally innervating nerve (Figure 1.2 (a)). To prevent this nerve from reinnervating the muscle again it is ligated at its end. Big muscles like musculus pectoralis major are split in segments in order to get more independently biological amplifiers from a single muscle. In addition to that muscles are repositioned and subcutaneous fat is removed to lower the fat’s damping effect and thus increase the EMG signal’s intensity. After this preparations are made the nerves from the residual limb, naturally responsible for controlling arm and hand movements, are rerouted to the targeted muscles to reinnervate them (Figure 1.2 (b)). Due to the nerve’s slow growth and thus a very slow process of growing together with the muscle its not possible to detect any muscle signals immediately after surgery. Three months after surgery muscle twitches can already be seen and another three months later the patient is able to deliberately control reinnervated targeted muscle [13].

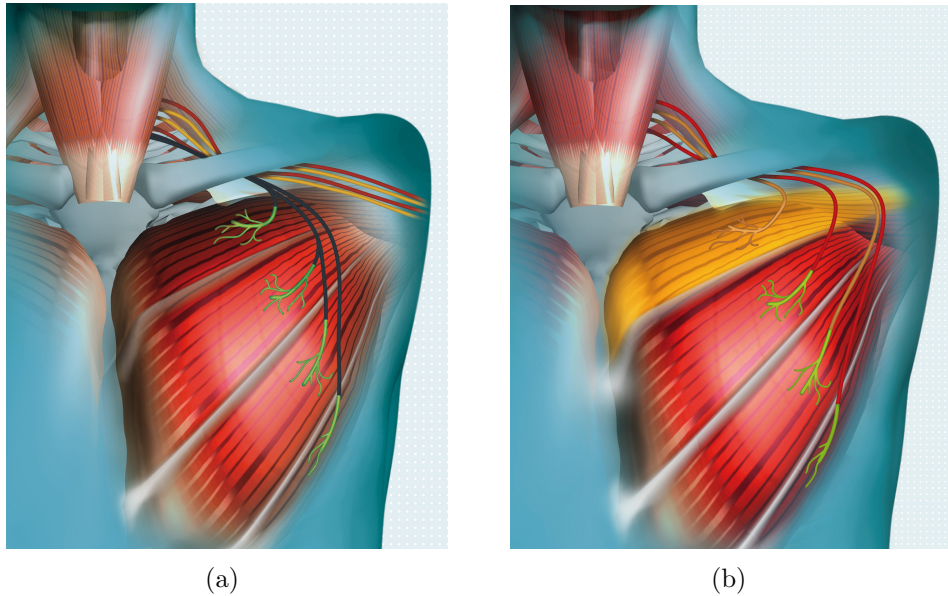


Figure 1.2: TMR - operative procedure: (a) The targeted muscle is denervated by cutting its naturally innervating nerve. (b) The targeted muscle is segmented and reinnervated by the remaining nerves from the residual limb (Source: „Otto Bock Healthcare Products GmbH“).

1.2.2 Electromyography

„Electromyography“ (EMG) is a technique for detecting and recording the electrical muscle activity. To be able to measure EMG signals correctly and interpret the results in a proper way respectively one has to understand how EMG signals are generated and which arrangements have to be taken in order to get the best measurement. Therefore, the following subchapters provide some basics about the anatomic construction of muscles, the sequence of processes taking place in muscle contractions, and the generation and recording of EMG signals.

1.2.2.1 Anatomical Construction of Muscles

Muscles are subdivided in smooth and striated muscles. Additionally, the latter is further subdivided to skeletal and cardiac muscles. Smooth muscles contract without conscious thought and are responsible for the control of organs. In contrast skeletal muscles contract deliberately and they are used to move the body. The connection between muscle and bone and therefore the transmission of force is accomplished by tendons. Cardiac muscles are a special kind of muscles because they are striated but not deliberately contractible [14][15]. Due to the importance of skeletal muscles for this master thesis the following paragraphs will focus on their composition and functionality exclusively. The figure below (Figure 1.3) shows the structure of a muscle and its constituents.

The basic unit of a muscle is called sarcomere. A detailed description of all

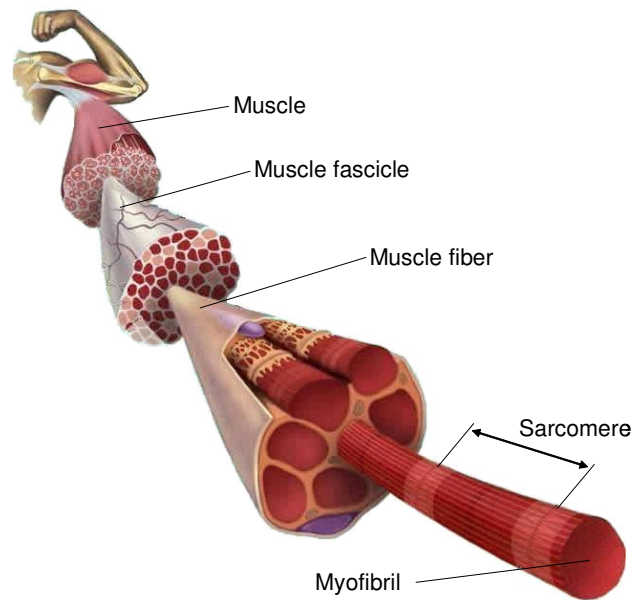


Figure 1.3: Muscle structure from myofibrils, composed of sarcomeres, to muscle fibers to muscle fascicles (modified from [16]).

processes within a sarcomere, which lead to a contraction of the muscle will be explained in the next subchapter (Chapter 1.2.2.2). Myofibrils are composed of repeating sections of sarcomeres and a certain amount of myofibrils in parallel arrangement is called a muscle fiber. Among the myofibrils muscle fibers are also composed of organelles and endomysium. According to the composition of organelles, one distinguishes between slow twitch (ST) and fast twitch (FS) muscle fibers. ST fibers are characterized by a sluggish contraction but a high fatigue resistance, whereas FT fibers contract fast, but due to the high power consumption fatigue quite fast as well. Depending on the size and the functionality of a muscle its muscle fiber's length is between a few millimeters up to a few centimeters. The biggest functional unit a skeletal muscle is composed of is called a muscle fascicle. It is build up of a bundle of muscle fibers surrounded by perimysium, which is connective tissue [14][15].

As already mentioned in chapter 1.2.1.1, the control of muscles is carried out by peripheral nerves. The nerve cells of those peripheral nerves are called motor neuron. The connection between the nerve's axon and the muscle fiber and therefore the signal transmission is accomplished by the neuromuscular junction [15][4]. The figure below (Figure 1.4) illustrates the sequence of processes needed to transmit signals from the nerve to the muscle fiber.

The action potential (AP) moves along the axon and once it reaches the neuromuscular junction calcium channels open and thus acetylcholine (ACh) vesicles are actuated towards the synaptic cleft. When they reach the presynaptic membrane they start to merge into that membrane and release the ACh to the synaptic cleft. Subsequently the ACh particles bind to the muscle fiber's postsynaptic membrane

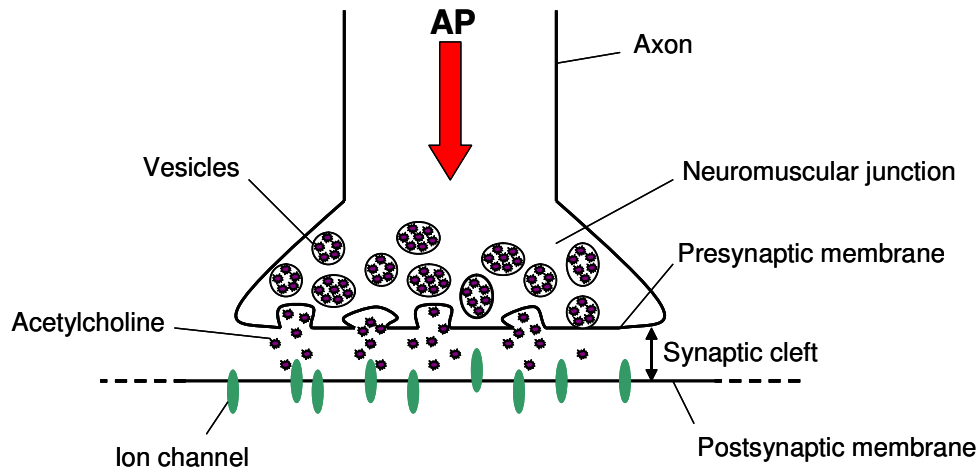


Figure 1.4: Signal transmission from the axon of a peripheral nerve to the muscle via the synaptic cleft.

in order to open ion channels. Thus sodium and calcium ions flow into the muscle cell, which leads to a depolarization causing an AP along the muscle fiber [17][4]. The AP's contribution to the muscle contraction will be explained in the following subchapter (chapter 1.2.2.2).

According to the muscle's functionality a single motor neuron innervates a different amount of muscle fibers. In this context a term named motor unit is introduced, which describes the smallest functional unit in muscle activity [18]. Per definition a motor unit is a single motor neuron and the group of all muscle fibers innervated by it. The amount of muscle fibers innervated by a single motor neuron varies a lot. In muscles which have to perform precise and small movements, such as muscles of orbit, a lot of small motor units, innervating 100 to 300 muscle fibers, are recruited. Rough movements, like thigh muscles perform, are carried out by using less but bigger motor units with up to 2000 muscle fibers [19].

1.2.2.2 Muscle Contraction

For a better understanding of the biochemical processes leading to a muscle contraction one has to know the composition of a sarcomere. Figure 1.5 illustrates the essential parts of a sarcomere.

A sarcomere is bounded by two neighboring Z-lines, with actin filaments attached to them. In between those actin filaments myosin filaments are arranged, which are connected to the Z-lines via the protein titin leading to a centering of the sarcomere [15]. The myosin filament itself is composed of a tail and a head domain (see figure 1.6). Figure 1.5 (a) illustrates the sarcomere's resting state, whereas in figure 1.5 (b) the contraction can be seen causing a change in the sarcomere's length. This process of muscle contraction is depicted by the sliding

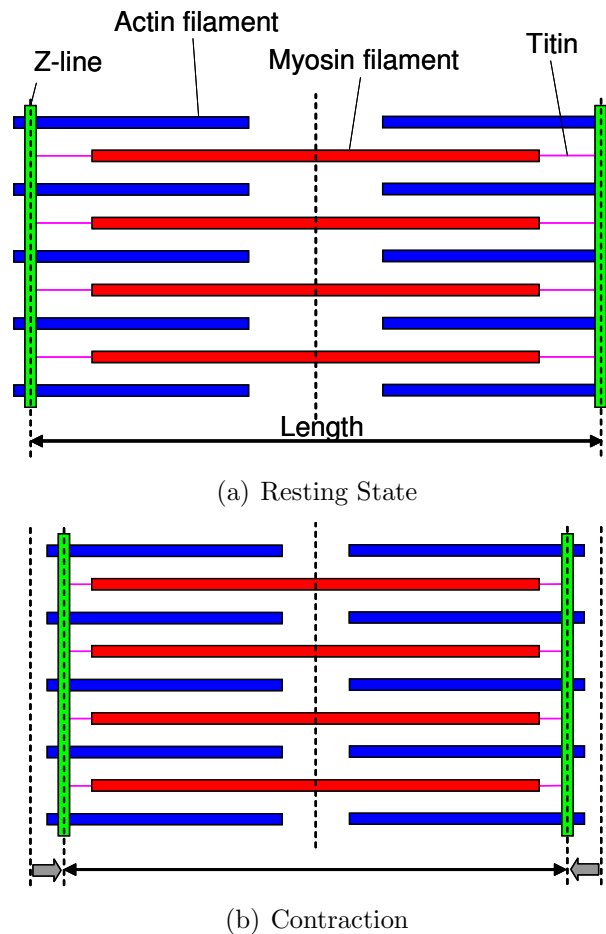


Figure 1.5: Sarcomere: (a) The sarcomere's resting state. (b) The sarcomere's state during contraction. The contraction is caused due to a displacement of the actin filament relative to the myosin filament.

filament theory (Figure 1.6) developed by Andrew F. Huxley and Hugh E. Huxley. Although this theory is not proved completely, it describes the contraction processes the best. It says that an electrical excitation in terms of an AP traveling along the muscle fiber causes Ca^{2+} -ions flowing into the sarcomeres and provoking two major incidents. On the one hand it promotes the myosin head's enzymatic activities leading to a fragmentation of adenosine triphosphate (ATP) to adenosine diphosphate (ADP) and phosphate (P). On the other hand Ca^{2+} -ions bind to troponin of the tropomyosin filaments causing a movement of those filaments in order to unblock the actin's binding sites. Consequently, the myosin head binds to the actin filament resulting in a release of the P and the ADP shortly afterwards. Thus the already pre-stressed myosin head buckles sideways leading to a displacement of the actin filament relative to the myosin filament causing a change in the sarcomere's length. This change in length is in the range of 7 to 10nm per sarcomere. Subsequently the Ca^{2+} -concentration decreases and a new ATP binds

to the myosin head. Thus it detaches from the actin filament and the tropomyosin filaments move back blocking the actin again [20] [21].

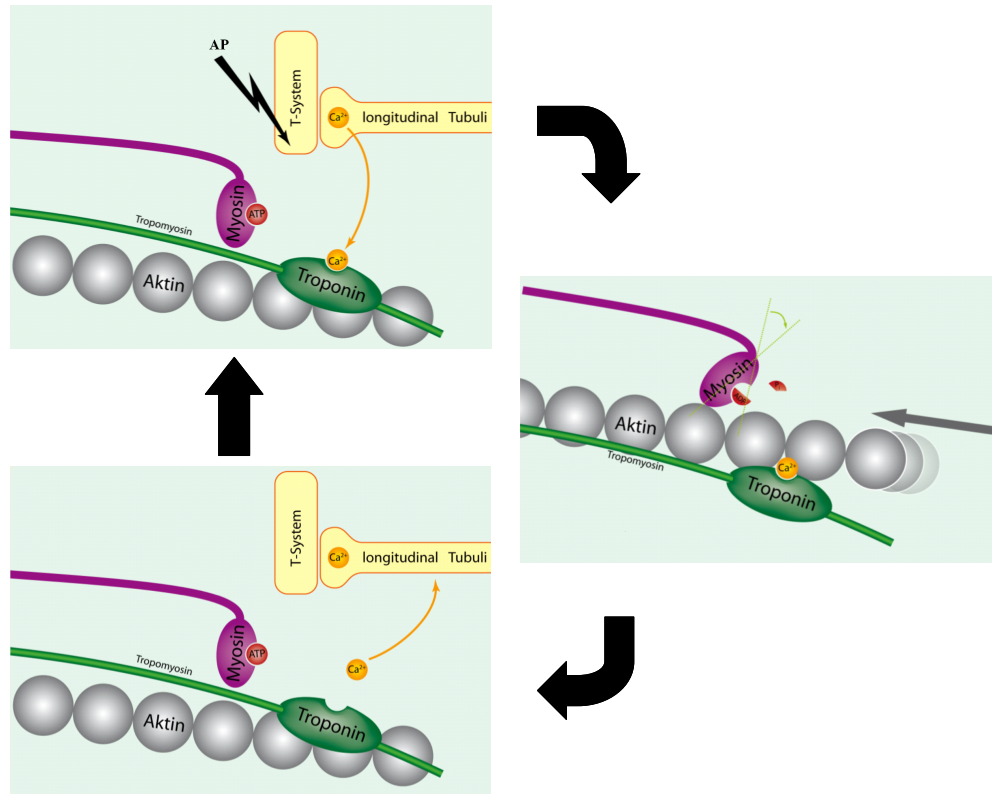


Figure 1.6: Sliding Filament Theory (modified from [22]): An AP traveling along the muscle fiber causes Ca^{2+} -ions flowing into the sarcomeres, causing chemical reactions, leading to a change in the sarcomere's length.

1.2.2.3 Generation of sEMG Signals

As already mentioned in the beginning, EMG is a method to detect electrical muscle activities. To be more specific, surface EMG (sEMG), which is used on this project, measures the potential field on the skin (Figure 1.7), which is provoked by the propagation of the AP along the single muscle fibers within the muscle. In this regard both the surrounding tissue structures and the body geometry affect the potential field distribution on the skin and hence the EMG signal. Therefore the tissue can be seen as a volume conductor featuring spatial and temporal low pass filter behavior, causing high frequencies being damped [23]. Due to multiple muscle fibers contributing to a single muscle contraction the sEMG signal is a summation of more than one potential field. Hence, the contribution of every muscle fiber can be superimposed according to Helmholtz's theorem. Rosenfalck [24] was the first to use the superposition principle in order to estimate sEMG signals.

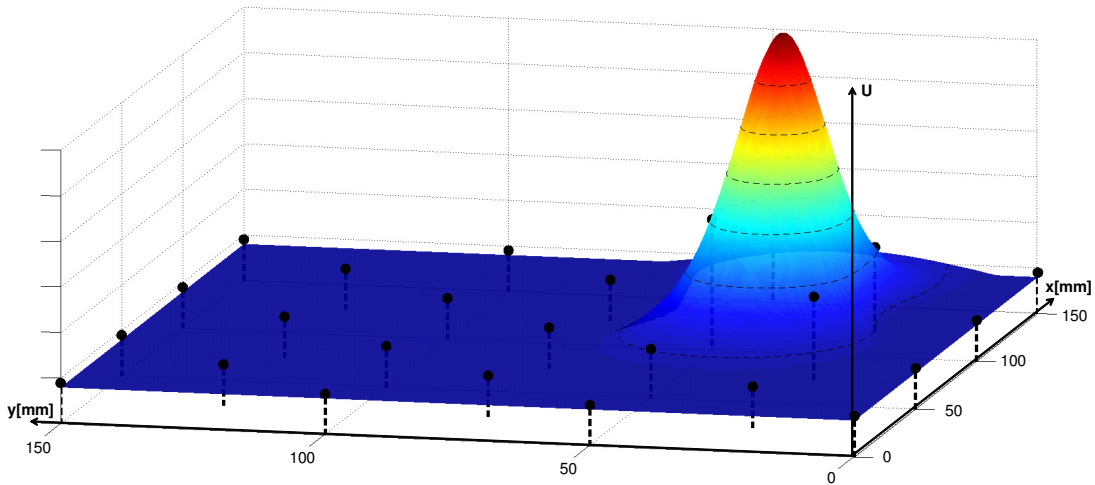


Figure 1.7: Simulated three dimensional potential field distribution. The x- and y-axis determine the position on an $150\text{mm} \times 150\text{mm}$ field and the z-axis displays the strength of the potential field. The black dots denote different electrodes, which detect different amplitudes of the potential field.

1.2.2.4 Detection of EMG Signals

There are two different approaches in detecting EMG signals. The first method uses needle electrodes, which are pierced into the muscle like a syringe. Advantages of this approach are the accurate positioning of the electrode, the low crosstalk from other muscles and the low damping of the signal caused by surrounding tissue. However there are some profound disadvantages as well. To insert the electrode into the muscle a doctor is needed, which makes it quite expensive and time-consuming. In addition to that compared to the second approach this invasive medical procedure is quite unpleasant for the patient. For this reason surface electrodes are used. They are attached to the skin surface and their output is the sum of the potential fields caused by each muscle fiber [23] [25], which makes them suitable for any application with a need of a global EMG signal (the sEMG signal). To reduce noise, i.e. 50Hz electromagnetic fields, bipolar electrodes with an electrode distance of about 2cm are used [26]. The disadvantage of this method is the crosstalk of other muscles in the detected sEMG signal, but the advantages of being less complex and more comfortable preponderate.

1.2.3 Current Procedure in Therapy

This chapter describes the current procedure in therapy in order to find hotspots at the TMR patient's skin. According to the healing process, which varies from one patient to the other, approximately three months after TMR surgery one can start intensive therapeutic treatment. The first step is to find those electrode positions at the patient's skin capable of providing the most intensive sEMG signals in order

to get a proper prosthesis control. Therefore a device is used, which is connected to an electrode, whose signal is transmitted via the device to a computer. At the monitor the therapist is able to see the electrode signal and therefore he can decide whether its suitable for prosthesis control. As a starting point the electrode is put at the approximate position of the hotspot, known from the clinical data about the surgery. As already mentioned this position is an estimation, since it is not possible to predict the nerve's growth during the healing process. Subsequently, the patient initiates the movement associated with the current electrode position and the sEMG signal is recorded. In the next step the electrode is repositioned and rotated and the patient performs the same movement again. Comparing the two sEMG signals with each other the therapist can tell which one fulfills the requirements the most and chose it. This procedure is repeated until the best signal is found. Since most of the TMR patients use six sEMG signals this hotspot finding process has to be done six times, which makes it for both the patient and the therapist very time-consuming and cumbersome. In addition to that one can not tell whether the most intensive signal was found or just an other good one. This is a huge disadvantage because the better and more differentiated sEMG signals are, the more suitable they are for prosthesis control. After all hotspots are located the actual training of prosthesis movements starts. Since this part of therapy is not relevant to this work it is not explained further.

CHAPTER 2

Methodology

This chapter deals with the theory and the execution of the applied methodology. The actual results of the work, which arose from this methodology, are shown in chapter 3. The first subchapter describes the build-up of the electrode array and all its components in detail. Subchapter 2.2 deals with the data acquisition and the conditioning of the sEMG signals. Afterwards, a detailed explanation of the simulation of sEMG signals will be given. In addition to that, the following section describes different approaches in order to localize hotspots within the electrode array. At the end, methods used for visualization will be mentioned and the last subchapter shows how to use the outcome of this work in therapy.

2.1 Build-up of the Electrode Array

Although the hardware-based build-up of the electrode array actually was not part of this project, it was inevitable to work on that issue as well, because the further development of hotspot finding algorithms strongly depends on reliable measurements. In the course of this work different approaches in building an electrode array were used, because profound issues from earlier versions had to be eliminated in order to get better results in localizing hotspots. This chapter provides information about the different build-ups of the electrode array and the reasons for the adaptations will be mentioned. Since those versions only differ from each other by the amount of electrodes and their mounting to the patient's body, the first subchapter deals with the analog digital converters (ADC) responsible for data acquisition. Subsequently, the electrode's technical details will be described. Later on the different approaches will be described in detail.

2.1.1 Analog Digital Converters

In order to detect the sEMG signals, two USB-1608G ADCs by „MC Measurement Computing“ are used. Each of them features 16 analog input channels, which leads to a maximum of 32 electrodes in common. The ADC's resolution is 16 bit and the sampling rate per channel is set to 1000 samples per second. In order to acquire data from each channel at the same time or rather to keep the time between those measured values as low as possible, arrangements have to be made. This is implemented by sampling in „burst mode“ (Figure 2.1). For this purpose the data

is not sampled at evenly spaced intervals, but at the maximum A/D rate. In case of the used ADCs, the lowest possible sample period is $4\mu\text{s}$ [27]. In addition to that, both ADCs on the one hand have to start the data acquisition at the same time and on the other hand have to proceed their sampling at exactly the same rate. The former part of this synchronization is accomplished by using a digital trigger signal in order to start the acquisition. The latter is achieved by using the first ADC's clock rate as a common clock rate.

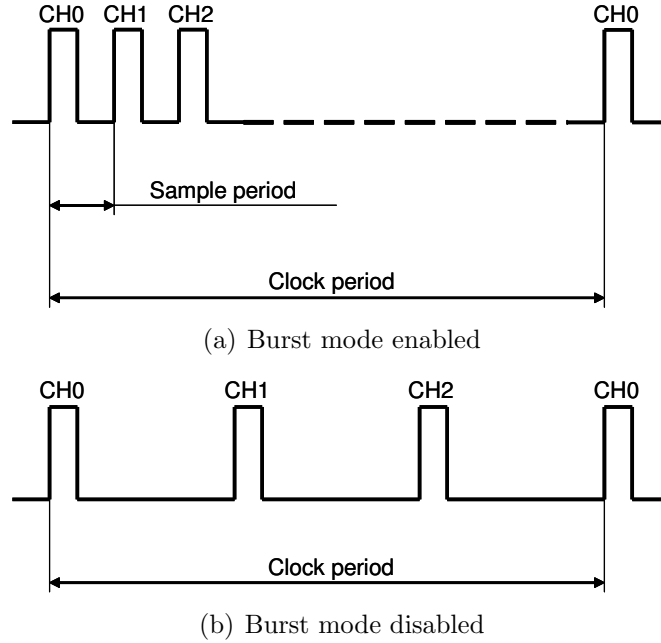


Figure 2.1: Burst mode (modified from [27]). (a) With burst mode enabled the sample period (time between each sample) is set to its lowest possible value ($4\mu\text{s}$). (b) If disabled, the sample period is much higher, because evenly spaced intervals are used.

2.1.2 Electrodes

As previously mentioned bipolar MyoBock raw signal electrodes by Otto Bock are used in this project (Figure 2.2). Compared to other similar electrodes they are more sensitive and therefore even very low sEMG signals can be detected. At its rear side one can adjust its amplification in a logarithmic manner. In addition to that, the high frequency shielding and filtering and the high common mode rejection in the low frequency range ($> 100\text{ dB}$ at 50 Hz) make it less interference-prone to noise, like electromagnetic interference. Furthermore an electronic power-line frequency filter is already implemented for additional interference protection against signals with power-line frequency. To compensate skin resistance differences the electrode's input resistance is kept high. The frequency bandwidth ranges from 90 to 450 Hz , the sensitivity range from $2,000$ to $100,000$ fold and the electrodes are

27mm×18mm×9.5mm in size [28].



Figure 2.2: MyoBock raw signal electrodes used in this project (from [28]).

Since it is very important for the computation of hotspot positions to use electrodes with identical adjustments, which provides the same output on equal input conditions, a calibration was necessary. Therefore a frequency generator was used in order to generate a constant signal and level equal conditions. This signal was applied to the electrodes and subsequently the amplification was adjusted until all of them provided the same output voltage on the same input conditions.

2.1.3 Electrode Array (1. Approach)

In the beginning of the project, the limitation of building an electrode array, usable on a multitude of patients, was neglected because initially the most important matter was getting data as soon as possible, in order to start the development of hotspot finding algorithms. Therefore the first approach used a total of 19 electrodes (figure 2.3 (a)).

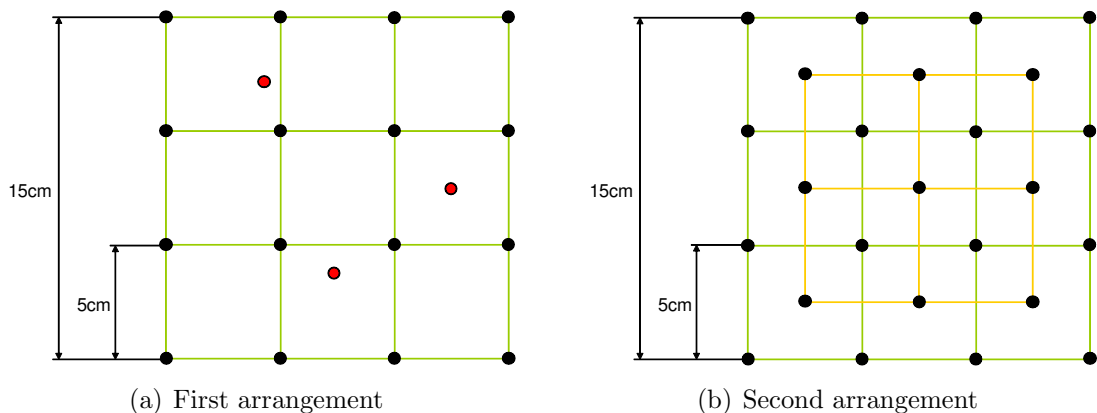
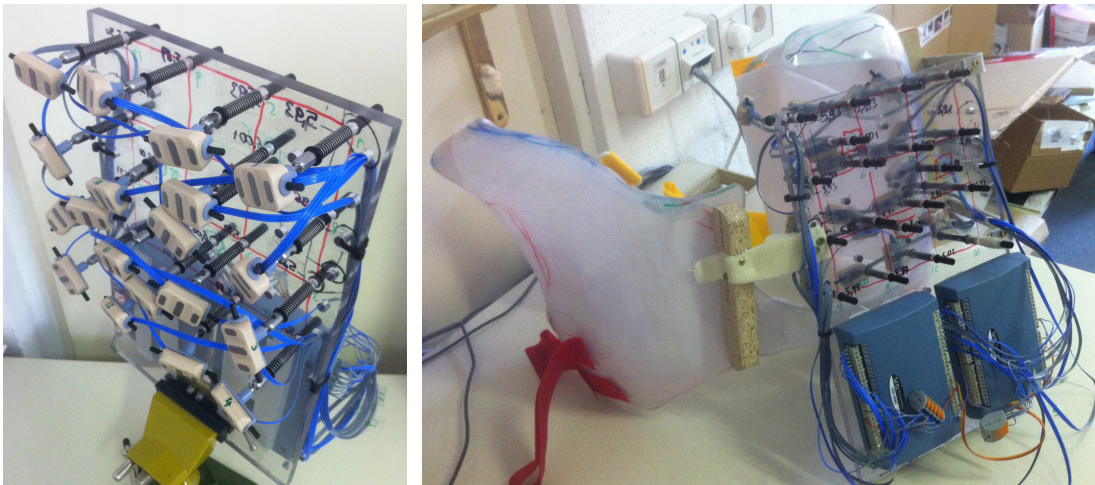


Figure 2.3: Electrode arrangements: (a) The arrangement for the first build-up using a total of 19 electrodes. (b) The arrangement, which was used for all further build-ups. It comprises a total of 25 electrodes.

16 of the 19 electrodes were arranged on a regular grid (4x4 electrodes) with an inter electrode distance of 5cm. Since the patient already used a prosthesis and therefore the hotspots were already known, three electrodes were used to sample the actual hotspot positions on the patient's chest in order to be able to check the accuracy of the computations derived from the 16 other electrodes. All 19 electrodes were mounted on a quadratic acrylic glass plate by using spacers of different length in order to make contact between the electrodes and the skin. To get an equal contact pressure and to avoid liftoffs springs were used. In addition to that a small spherical joint was used in order to make the electrode rotatable and hence more adjustable (figure 2.4 (a)). Since this prototype was used for one particular patient only, a „ThermoLyn[®] PETG“ brace of the patient's chest was made and the acrylic glass plate was mounted to it (figure 2.4 (b)). As one can see on the picture as well, the ADCs were mounted on the brace too in order to get a compact build-up.



(a) Electrode Array

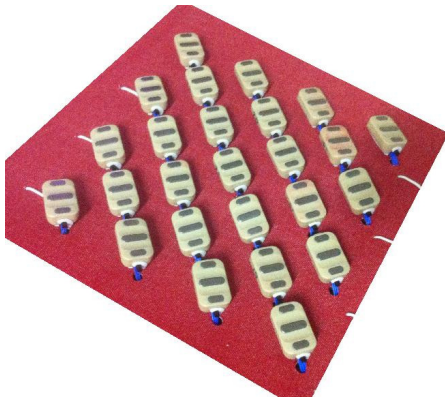
(b) Brace

Figure 2.4: First approach of the electrode array build-up. (a) Electrodes mounted on an acrylic glass plate. (b) Glass plate mounted to a brace.

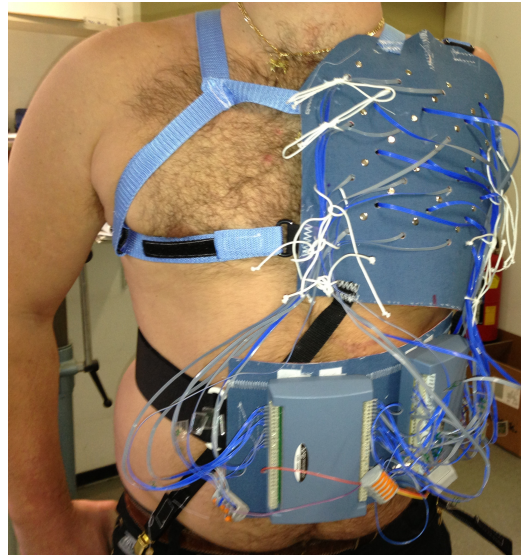
After a few measurements it became clear, that this kind of build-up was not suitable for this work. The first reason for that was the low amount of electrodes used and the quite big inter electrode distance which led to bad sampling behavior. In addition to that, the springs attached to the spacers did not work as expected and therefore electrode liftoffs occurred, which made the measurement data hard to process. Hence another build-up was developed to get rid of these problems and therefore improve the measurement.

2.1.4 Electrode Array (2. Approach)

The second approach used a total of 25 electrodes. To get the setup as flexible as possible, and hence usable on a multitude of patients, the electrodes were mounted on a quadratic piece of textile (Figure 2.5 (a)), which was attached to the patients chest using straps (Figure 2.5 (b)). More precisely, small sockets were riveted to the textile and the electrodes were put into these sockets. Since the rivets were fastened quite loosely the electrodes were rotatable again. In this case the ADCs were mounted on a separate textile, which was worn like a belt bag. A huge advantage of this kind of build-up compared to the first approach is that it is not just applicable to a particular patient, because the textile fits to the skin surface of every patient and it is not of issue whether the patient's right or left limb is amputated.



(a) Electrode Array



(b) Build-up

Figure 2.5: Second approach of the electrode array build-up. (a) Quadratic piece of textile with 25 electrodes attached to it. (b) Whole build-up, which attaches the textile to the chest using straps.

16 electrodes were once again arranged on a regular 4x4 grid with an inter electrode distance of 5cm, but in addition to that, nine electrodes were placed in the middle of the intermediate area between those electrodes, which led to a much better sampling behavior compared to the first approach. Figure 2.3 (b) illustrates the described electrode arrangement. Although measurements of this electrode array build-up were much better than those from the first approach, some features with great need of improvement still occurred. Due to the formation of indentations and elevations on the skin during strong contractions the electrodes still lifted off, like in the previous approach, and therefore the measurements were

distorted. Hence some other improvements had to be made, which led to a third approach.

2.1.5 Electrode Array (3. Approach)

Since the second approach's sampling behavior was good, the third electrode array build-up uses the same electrode arrangement (see figure 2.3 (b)). In order to get constant contact pressure from every electrode to the skin and therefore prevent the electrodes from lifting off, an inflatable pad was used. This pad consists of 25 chambers (one chamber for each electrode), which are connected to each other, leading to an equal pressure distribution within the pad. Since it is not possible to attach the electrodes to the pad directly and to do any sewing respectively, at each side of the pad two additional layers had to be added. Figure 2.6 (a) shows a cross-sectional sketch of the build-up illustrating the different layers.

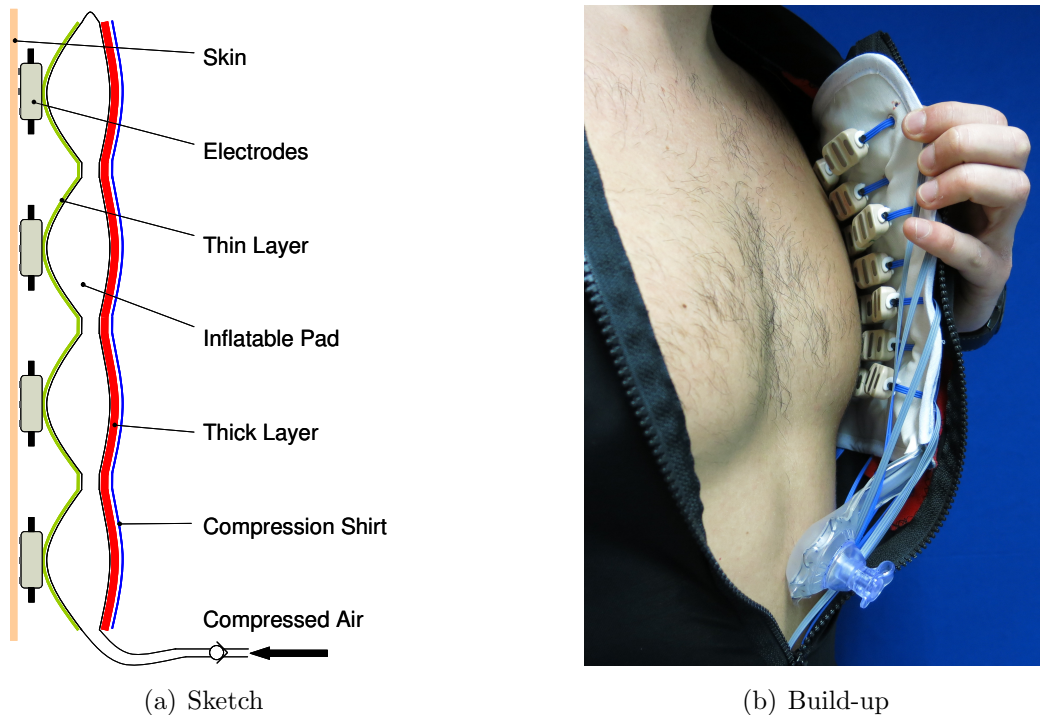


Figure 2.6: Third approach of the electrode array build-up. (a) Cross-sectional sketch of the inflatable pad featuring all of its components and layers. (b) The pad, which is attached to the patient's chest using a compression shirt.

The innermost (proximal) layer, carrying the electrodes in the same way the build-up of the second approach did, was composed of a bidirectionally stretchable and quite thin textile, whereas the outermost (distal) layer was a quite thick and non-stretchable textile with a hook and loop fastener at its surface. When inflated, this kind of composition led to a high expansion of the pad in the direction of the

body, due to the lower resistance of the thin and stretchable layer compared to the thick and non-stretchable one. At the borders, the three layers were sewed up with each other and in order to get small ducts for the electrode cables tubular rivets were used. Additionally the figure shows the junction for the compressed air with a back pressure valve on its disposal. Using the hook and loop fastener, the electrode array was attached to the inner surface of a compression shirt, leading to an inherently good contact pressure which could be adjusted by inflating the pad. Figure 2.6 (b) shows the whole build-up on a chest.

Again compared to its precursor this build-up improved the measurements. But against all expectations electrode lift-offs still occurred, when strong contractions appeared, which made another rethinking of the build-up necessary.

2.1.6 Electrode Array (4. Approach)

Due to the unusable measurement results of the previously mentioned approaches, the limitations made beforehand had to be alleviated. Therefore the idea of building a patient-independent electrode array had to be left behind. Instead of that, the fourth build-up uses a patient specific socket (made of laminating resin) with all electrodes incorporated. Again the same electrode arrangement (see figure 2.3 (b)) as used before was utilized. Since the electrodes were not rotatable, their orientation (angle) had to be determined in advance. Therefore the electrodes were aligned in direction of the muscle fibers of *musculus pectoralis major*. Figure 2.7 shows the build-up of the fourth approach.



Figure 2.7: Fourth approach of the electrode array build-up, using a patient specific socket with all electrodes incorporated.

The measurement results derived from this approach were much better than any other measurements previously made. Nevertheless electrode lift-offs occasionally arose again, but since they occurred quite infrequently and the hotspot finding algorithms are able to deal with a few lift-offs, the measurements were suitable for testing these algorithms. Since this build-up is still in need of improvement, chapter 4 provides the reader with some ideas of how to deal with this issue.

2.2 Data Acquisition and Signal Conditioning

As previously mentioned, the sEMG signals were detected by electrodes which are connected to two ADCs, in further consequence of this chapter also called devices. This chapter deals with the programmatically necessary measures to collect the acquired data from the ADCs and to transfer it to the computer. Additionally, the signal conditioning necessary to make the raw signals derived from the electrodes evaluable is also described.

2.2.1 Data Acquisition

The first step is to initialize the devices by assigning the electrodes to the correct device and start the self calibration process. This initialization is performed on every restart of the software in order to ensure both devices are connected to the computer and work properly.

Following this, the devices are ready to acquire data and to make it available to the computer. Therefore, if the start button is pressed, firstly all callbacks, which are discussed later, are enabled. After that, the configuration of the trigger and all scan settings are done, which include the following items:

- Use channel 0 - 15 of every device
- Input range from -5V to 5V
- Sampling rate is set to 1kHz
- Enable continuous scan
- Enable burst mode
- Enable block mode
- Use internal pace on device 1 (acting as a master)
- Use external pace on device 2 (acting as a slave)

Continuous scan means, that once started, both devices keep on scanning (acquiring) data until a stop command is sent to them. The reason for using burst mode and the internal pace on devices 1 and the external pace on device 2 is explained in section 2.1.1. In order to reduce the traffic between the devices and the computer, block mode is enabled. Therefore, the devices collect data on their internal memory and once a certain amount of samples (100 samples per channel) is available a callback is invoked. Subsequently, the trigger signal is generated in order to start the scan on both devices simultaneously. When started, the devices will acquire data block by block and the invoked callback method transfers the data to the devices. Finally, if the stop button is pressed, the devices will stop their scan procedure and all callbacks will be disabled. Figure 2.8 shows the described scan procedure graphically.

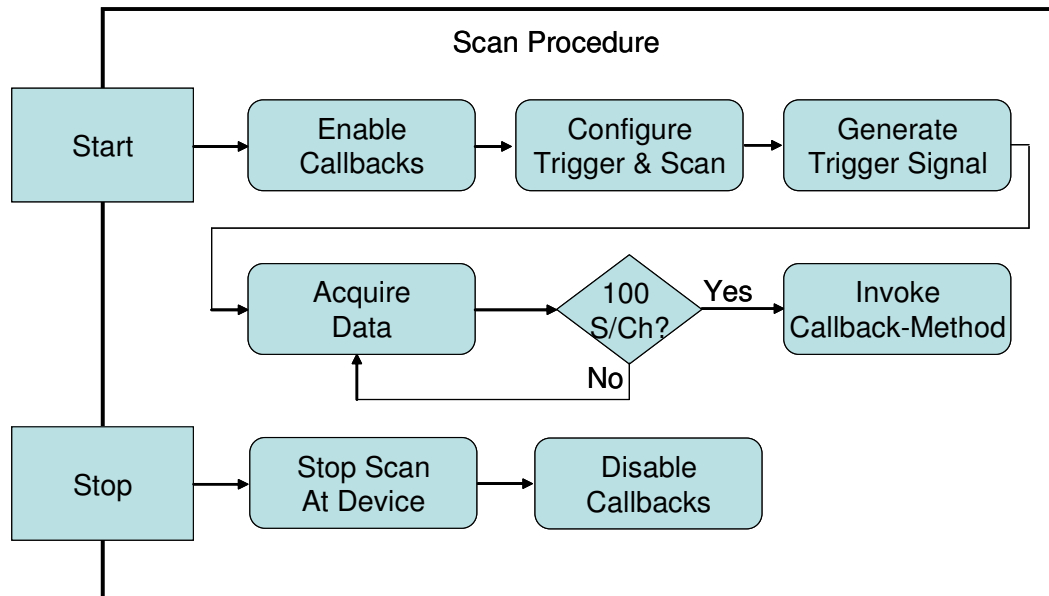


Figure 2.8: Flow chart of the scan procedure showing all tasks being performed while acquiring data from the ADCs.

The previously mentioned callbacks are methods within the software responsible for performing certain actions in case of certain events. Every device features three callbacks:

- „Read Scan Data“
- „Scan Complete“
- „Scan Error“

The most important is „Read Scan Data“, as it is responsible for transferring the data from the device to the computer. As explained before, it is invoked if

another 100 samples per channel are acquired and ready to be taken from the device. In this case those 100 samples are copied to a data array containing the whole measurement. The only purpose of „Scan Complete“ and „Scan Error“ is to stop the scan in case of a completion of the scan or if an error occurs.

2.2.2 Signal Conditioning

The first step in signal conditioning is to scale the raw signal obtained from the electrodes, which is a 16 bit signal in the range of 0 to 65535 to double values between -5V and 5V. Therefore the following equation is used:

$$X_{\text{scaled}} = \frac{X_{\text{unscaled}} - 32768}{32768} \times 5$$

After that, the DC offset, as one can see in figure 2.9(a), created by the electrode, has to be removed from the signal in order to be able to apply further signal conditioning like rectification. Due to its stochastic behavior, the mean of a sEMG signal describes its offset and hence is used to eliminate this offset. Therefore the mean of the whole signal is calculated and subtracted from the data.

Considering the offset compensated raw data, one can roughly tell the strength of the contraction of the muscle, like small, medium or strong contraction. Nevertheless, it is very challenging to get a precise classification from the raw data in order to get values in the range of 0-100% of the maximum voluntary contraction (MVC), which are more processible in terms of programming. Therefore the signal has to be conditioned.

In general the easiest way to get the signal’s amplitude parameters is to calculate the mean of every data point by using a certain amount of its neighboring data points. Since sEMG signals are stochastic the mean would always be 0 and therefore this approach is not suitable for signal conditioning of sEMG signals. A more advanced method, called average rectified value (ARV), rectifies the signal and calculates the mean afterwards. Therefore a moving average is calculated by including neighboring data points to the calculation of every single point. Another very similar approach is called linear envelope, which integrates the signal by applying a low-pass filter (4-10Hz cutoff) to the rectified signal. This kind of conditioning is used if a high correlation between the signal and the force generated by the muscle is required [29]. Since a hotspot is the most intensive (most powerful) sEMG signal detectable on the skin, in case of this project the property of utmost interest is the power of the signal. Therefore another approach, called root mean square (RMS), provides the best output [30]. It is determined by the following equation:

$$X_{\text{cond}}(t_0) = \sqrt{\frac{1}{N+1} \sum_{t=t_0-\frac{N}{2}}^{t_0+\frac{N}{2}} X_{\text{raw}}^2(t)} \quad (2.1)$$

where N is the amount of neighboring data points used to calculate the mean and t_0 determines the position of the sample on time axis. In order to work properly N has to be even and $N \in \mathbb{N}$. Figure 2.9(b) shows the conditioned signal after applying an offset compensation and RMS. The x-axis of the graphs show the time [s] and the y axis represent the voltage [mV].

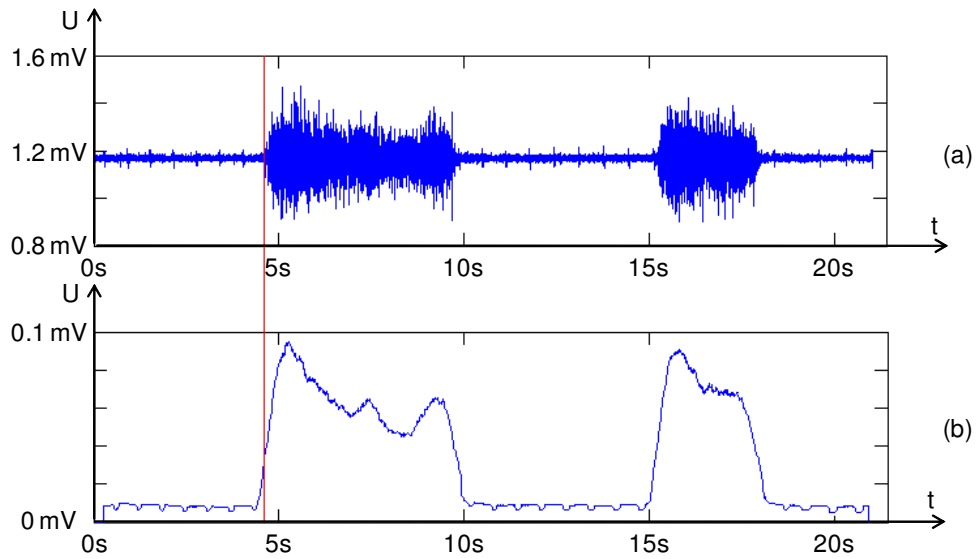


Figure 2.9: Signal conditioning. (a) Raw signal with an offset of approximately 1.17mV. (b) Conditioned signal, by using offset compensation and RMS.

As one can see from equation 2.1, similarly to AVR the moving average is used, which uses a window of a certain width (determined by N) in order to calculate the mean. The wider the window is the smoother the conditioned signal will get but also the bigger the phase shift between raw signal and conditioned signal will be (see red line in figure 2.9) [31]. In practice windows from 50 to 100ms are used. The example shown on the graph used a quite large window width of 500 samples and since a sampling rate of 1kHz was used this corresponds with 500ms.

2.3 Simulation of sEMG Signals

Since simulated sEMG signals are used in this work as well, this chapter describes some basics about how sEMG signals can be generated artificially. A big advantage

of using simulated signals is to be able to debug developed algorithms under the same conditions step by step. Since there are not many TMR patient available another huge advantage is avoiding very time-consuming patient live testings by using simulated data.

Jiang et al introduced a generative model of sEMG, which can be used to simulate EMG signals on the skin's surface [32]. A schematic diagram of this model is shown in figure 2.10. The model assumes the existence of some control information at the motor cortex and the spinal nerve respectively in terms of a time-varying force function $F(t)$. Using a synergy matrix S , indicating the extent to which different muscles participate to the movement, one gets the activation matrix

$$D(t) = SF(t) = [d_1(t), d_2(t), \dots, d_M(t)]^T,$$

where $d_M(t)$ is the activation function of the M^{th} muscle. Furthermore the overall sEMG signal $Z(t)$ is a summation of all inner EMG signals $y_M(t)$ affected by the nonlinear mixture matrix $G(t)$. In fact, $G(t)$ is the response function of the volume conductor of the considered part of the body. As described previously, the volume conductor and hence $G(t)$ depends on both the body's geometry and the tissue structure.

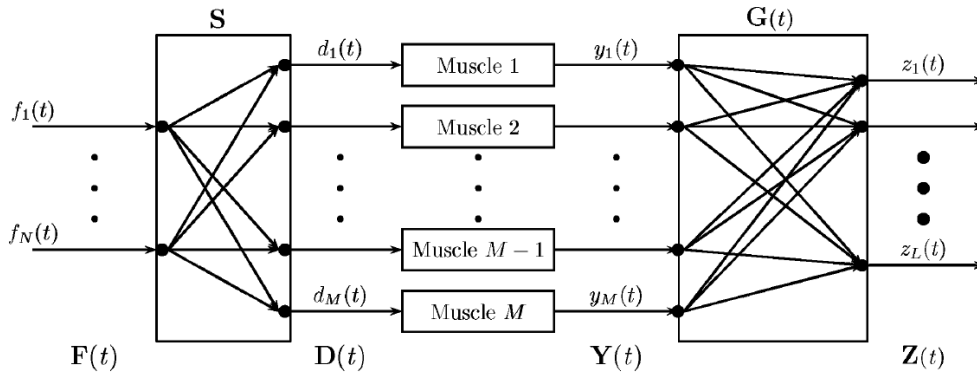


Figure 2.10: Generative model for sEMG (from [32]) describing the generation of EMG signals on the skin surface ($Z(t)$) from the information at the motor cortex ($F(t)$).

Honerer [33] presented a new approach to an approximate modeling of a three dimensional generic EMG model using finite element analysis (FEA) in order to compute the mixture matrix $G(t)$ and hence the resulting sEMG signal. Since there was a close collaboration with Mr. Honerer in the course of this work, all simulated sEMG signals were generated in particular for its purpose. Therefore in a first step magnetic resonance imaging (MRI) data, containing all necessary spatial tissue arrangements is segmented to its constituents (like muscles, bones, fat or skin), which can be seen in figure 2.11 (a). From this set of segmented images a three

dimensional model of the considered part of the body can be generated (figure 2.11 (b)). Subsequently, a muscle fiber is placed in the three dimensional model and an action potential along the fiber is generated. Therefore a mathematical description of the intracellular action potential by Rosenfalck is used. Hence the sEMG signal $Z(t)$ can be calculated by superimposing the contribution of every muscle fiber.

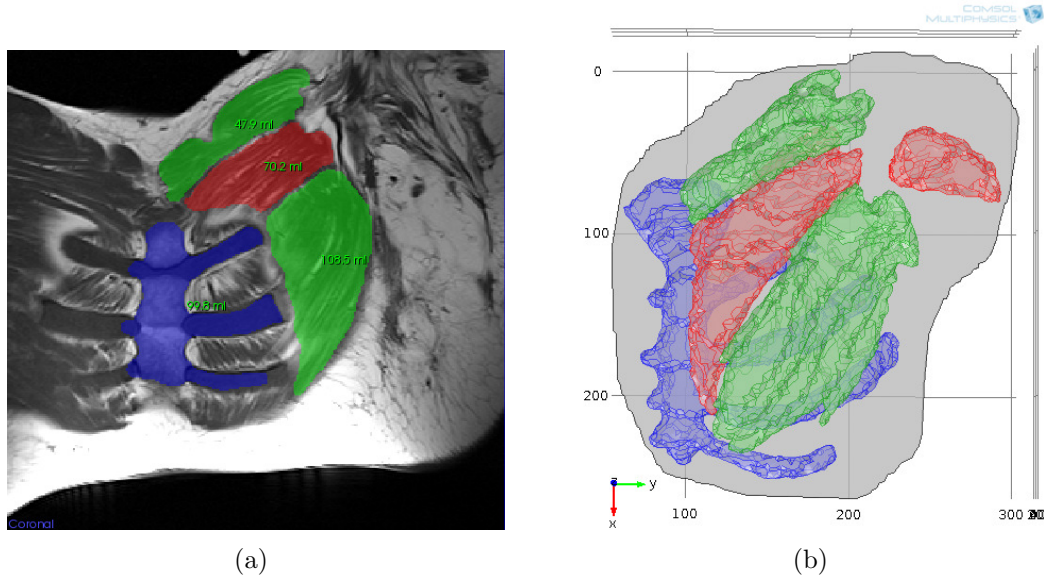


Figure 2.11: Three dimensional generic EMG model using FEA. (a) Segmented magnetic resonance image of the torso of a TMR patient in the coronal plane. One can see the sternum and the ribs, and the three segmented parts of the musculus pectoralis major. (b) Three dimensional model of the chest of a TMR patient, which is generated from the set of segmented images.

2.4 Hotspot Localization

Since in the beginning of this work it was not clear whether a specific approach for hotspot localization would be suitable for the project's purpose, two approaches had been tested. This chapter focuses on these different approaches and explains them in detail. In addition to that, every subchapter provides information about how they were implemented in the source code. Due to lift-offs it is not guaranteed that all electrodes provide proper signal all the time. Therefore the last subchapter will deal with the handling of faulty electrodes. The results and a comparison of the two approaches and hence the choice which had been made in order to get the best hotspot localization is presented in chapter 3.1.

2.4.1 Bicubic Interpolation

One approach to find hotspots is using an interpolation method to compute the potential field distribution in between the electrodes, by using their measured data values as nodes for the calculations. Since there are a lot of different interpolation methods the first step was to investigate the requirements for this work. Since the hotspots mostly do not reside directly below electrodes, but the maximum in the potential field will be located in between the electrodes it is important to use an interpolation method capable of generating interpolated values which can exceed or fall below their neighboring nodes. In addition to that a smooth surface was also desired, because it imitates a biological behavior the best. Therefore a method called bicubic interpolation was used, which yields a smoother surface than corresponding surfaces obtained by bilinear interpolation or nearest-neighbor interpolation and in addition to that the results have fewer interpolation artifacts. The following subchapters provide the reader with the mathematical basics of bicubic interpolation and subsequently the implementation to the project will be described.

2.4.1.1 Mathematical Basics of Bicubic Interpolation

Bicubic interpolation is applied on sets of data points arranged on a regular grid forming so called unit squares, whose length of every side equals 1 (Figure 2.12 (a)).

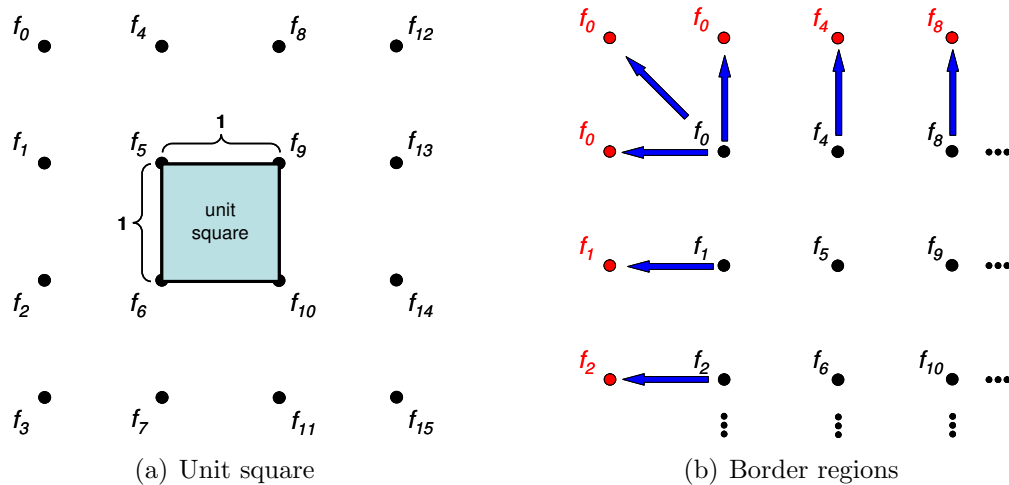


Figure 2.12: Basics of bicubic interpolation. (a) Grid of 16 data points ($f_0 - f_{15}$) formed by unit squares, whose length of every side equals 1. (b) The extension of the border regions by their own values, in order to be able to calculate the difference quotients.

Basically, it uses the unit square's function values f of the four corner nodes. But in addition to that the derivative in x -direction f'_x , the derivative in y -direction

f'_y and the cross derivative in xy -direction f'_{xy} of every node are required. Since most of the time those derivatives are unknown, they are typically approximated from the function values of neighboring nodes. Therefore the central difference quotient is used. In general the derivative of a function f at position x can be approximated by

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x},$$

where the two terms in the numerator of the fraction are the neighboring nodes of $f(x)$ and Δx is the distance to this nodes. Since unit squares are used Δx always equals one. Therefore the derivatives for the purpose of the two dimensional grid of data points can be determined quite easily. For instance if applied to the unit square in figure 2.12 (a) the following derivative in x -direction for f_5 will be derived:

$$f'_{5x} = \frac{f_9 - f_1}{2}.$$

Correspondingly the derivative in y -direction for f_5 is

$$f'_{5y} = \frac{f_6 - f_4}{2},$$

and for the cross derivative in xy -direction f'_{xy} one gets

$$f'_{5xy} = \frac{\frac{f_{10} - f_8}{2} - \frac{f_2 - f_0}{2}}{2} = \frac{f_{10} - f_8 - f_2 + f_0}{4}.$$

This equations work fine if every node of the unit square has a neighboring node on its disposal, but in the border regions of the interpolating surface this is not the case. Nevertheless one needs all derivatives of every node. Therefore the border regions are extended by their own values, which can be seen in figure 2.12 (b). This approximation is possible because nothing in the equations of bicubic interpolation requires the derivatives to be 100% precise [34].

Since all required values are known, or rather can be approximated, the interpolation surface within a single unit square can be written as an polynomial

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j, \quad (2.2)$$

where $p(x, y)$ is the interpolated value at position x and y of the unit square, which are in the range of 0 to 1. To solve the interpolation the 16 coefficients a_{ij} have to be determined. Therefore 16 equations have to be found, which can be achieved by matching $p(x, y)$ with $f(x, y)$, $f'_x(x, y)$, $f'_y(x, y)$ and $f'_{xy}(x, y)$ respectively.

Matching $p(x, y)$ with $f(x, y)$ yields the first four equations:

$$f(0, 0) = p(0, 0) = a_{00}, \quad (2.3)$$

$$f(1, 0) = p(1, 0) = a_{00} + a_{10} + a_{20} + a_{30}, \quad (2.4)$$

$$f(0, 1) = p(0, 1) = a_{00} + a_{01} + a_{02} + a_{03}, \quad (2.5)$$

$$f(1, 1) = p(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}. \quad (2.6)$$

Matching the derivatives of $p(x, y)$ with $f'_x(x, y)$ and $f'_y(x, y)$ by using the following identities

$$p'_x(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} i x^{i-1} y^j,$$

$$p'_y(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i j y^{j-1}$$

leads to eight more equations:

$$f'_x(0, 0) = p'_x(0, 0) = a_{10}, \quad (2.7)$$

$$f'_x(1, 0) = p'_x(1, 0) = a_{10} + 2a_{20} + 3a_{30}, \quad (2.8)$$

$$f'_x(0, 1) = p'_x(0, 1) = a_{10} + a_{11} + a_{12} + a_{13}, \quad (2.9)$$

$$f'_x(1, 1) = p'_x(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} i, \quad (2.10)$$

$$f'_y(0, 0) = p'_y(0, 0) = a_{01}, \quad (2.11)$$

$$f'_y(1, 0) = p'_y(1, 0) = a_{01} + a_{11} + a_{21} + a_{31}, \quad (2.12)$$

$$f'_y(0, 1) = p'_y(0, 1) = a_{01} + 2a_{02} + 3a_{03}, \quad (2.13)$$

$$f'_y(1, 1) = p'_y(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} j. \quad (2.14)$$

In order to get the last four missing equations, the cross derivative of $p(x, y)$

$$p'_{xy}(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} i x^{i-1} j y^{j-1}$$

is matched with the cross derivative $f'_{xy}(x, y)$ to yield

$$f'_{xy}(0, 0) = p'_{xy}(0, 0) = a_{11}, \quad (2.15)$$

$$f'_{xy}(1, 0) = p'_{xy}(1, 0) = a_{11} + 2a_{21} + 3a_{31}, \quad (2.16)$$

$$f'_{xy}(0, 1) = p'_{xy}(0, 1) = a_{11} + 2a_{12} + 3a_{13}, \quad (2.17)$$

$$f'_{xy}(1, 1) = p'_{xy}(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} i j. \quad (2.18)$$

To get an easy way to determine all coefficients from a 16-part set of function values and derivatives from a unit square, one formulates a linear equation

$$A\alpha = x, \quad (2.19)$$

where the unknown coefficients a_{ij} are grouped together in a vector α and the same is done for the function values and derivatives of the nodes (grouped together in vector x), leading to the following matrix multiplication:

$$\begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & 0 \\ 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 9 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f'_x(0,0) \\ f'_x(1,0) \\ f'_x(0,1) \\ f'_x(1,1) \\ f'_y(0,0) \\ f'_y(1,0) \\ f'_y(0,1) \\ f'_y(1,1) \\ f'_{xy}(0,0) \\ f'_{xy}(1,0) \\ f'_{xy}(0,1) \\ f'_{xy}(1,1) \end{bmatrix}.$$

The values of the 16×16 matrix A are derived from the 16 equations mentioned above (2.3 - 2.18). Rearranging equation 2.19 leads to

$$\alpha = A^{-1}x.$$

Therefore in order to calculate the unknown coefficients, which are necessary to get the interpolated values, the inverse matrix A^{-1} is used.

2.4.1.2 Implementation to the Project

All calculations previously mentioned interpolate values within a single unit square, but since 25 values are derived from the electrodes, more than one unit square has to be computed. In order to get a result for the whole surface, it is important to know that due to the surface's continuity and the derivative's continuity at the borders of the unit square a bicubic interpolation on a regular grid can be accomplished by patching together all results derived from every single unit square. In summary, the task of getting an overall surface interpolation is to determine the unknown coefficients a_{ij} for every unit square, calculate all values between the nodes by using equation 2.2 and subsequently patch the results from every unit square together.

Reconsidering the electrode arrangement of the latest electrode array (figure 2.3 (b)), one will easily notice that it is not that obvious to find unit squares to be able to cover the whole grid. One possibility is to use rotated unit squares in order to find rectangles with equal side lengths, but this approach is not suitable because the triangular border regions can not be covered this way (Figure 2.13 (a)). Therefore bigger unit squares are used in order to interpolate single values (Red dots in figure 2.13 (b)). By using these values as auxiliary nodes of the grid quite small unit squares can be applied, which cover all areas of the grid (Figure 2.13 (c)). This is the best possible approach in order to include all values derived from the electrodes to the same extent.

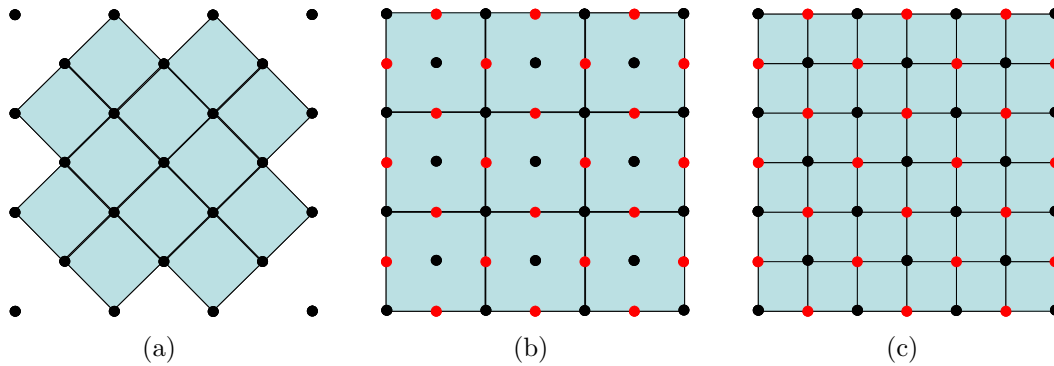


Figure 2.13: Usage of unit squares for interpolation. (a) An approach where rotated unit squares are used, which is not suitable because of the triangular border regions. (b) Bigger unit squares are applied in order to interpolate single values (red dots). (c) Smaller unit squares are applied to the refined grid.

2.4.1.3 Implementation to the Source Code

Figure 2.14 shows the basic concept of the method responsible for bicubic interpolation within the software. Its input parameters are the node values derived from the electrodes and the resolution, which determines the amount of interpolated points within every unit square. Its return value is an array containing all interpolation data. Inside this method a matrix is defined, which represents the inverse matrix A^{-1} explained in chapter 2.4.1.1. The node values are used to get the derivatives $f'_x(x, y)$, $f'_y(x, y)$ and $f'_{xy}(x, y)$. Since in the course of the project the electrode arrangement changed and therefore the amount of unit squares changed as well, there is a need to determine this amount from the node values too. Subsequently, the 16 coefficients of every unit square can be calculated, which leads to a $16 \times X$ matrix, where X is the amount of unit squares. From this matrix and the resolution, a two dimensional interpolation array can be deduced by using equation 2.2. The size of this array is defined by the amount of unit squares and the resolution. For instance reconsidering the electrode arrangement of the latest electrode array (Figure 2.3 (b)) would lead to the use of 36 unit squares (6×6)

as one can see in figure 2.13 (c). If an interpolation with a resolution of 100 is applied to this electrode setup the array would be 600×600 in size.

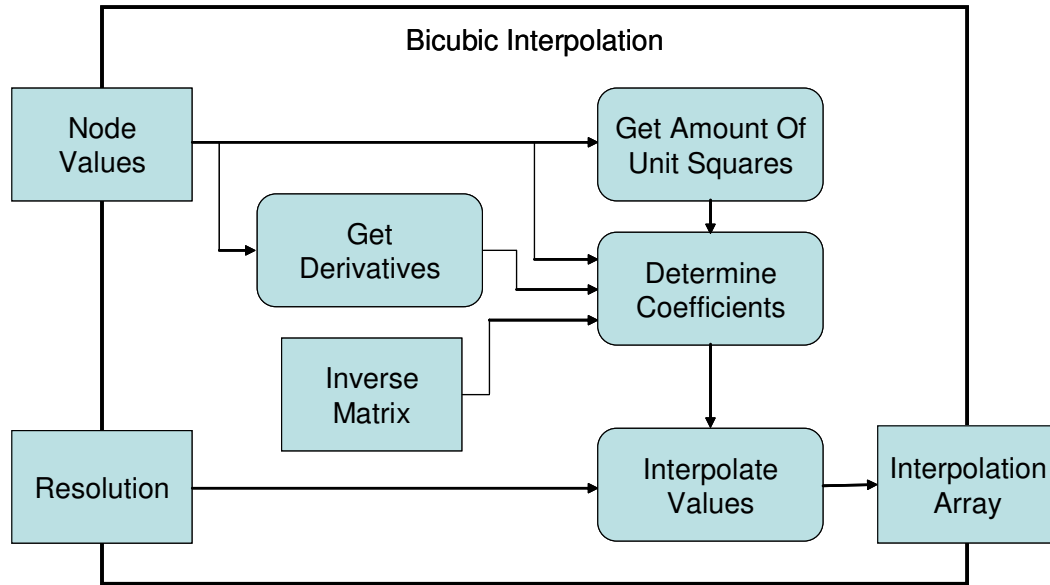


Figure 2.14: Bicubic interpolation - implementation to the source code. This flow chart shows how the interpolation array is derived from the node values and the resolution.

2.4.2 Template Matching

Another approach used in this work is called template matching. The basic idea is to make use of shape and size of a known hotspot (called template) and compare this template to the data derived from the electrodes [35]. Thus the alignment's extent of those two datasets is used to find hotspots. This approach uses electrode values at a discrete time and by applying it to several measurements in time, the motion of the hotspot can be captured. This chapter describes the generation of templates, the basic procedure of template matching and two different methods in order to calculate the alignment of the template and the electrode values. Subsequently, the implementation to the source code will be explained as well.

2.4.2.1 Template

For template matching previous knowledge of the shape and size of hotspots is required. From this knowledge templates can be generated, which in further consequence are used to find patterns in the data recorded [35]. In order to get this knowledge, data from high density electrode recordings, which show the actual hotspot on the patients skin due to its high resolution, was analyzed. From this data one can tell that a hotspot can be approximated by a two dimensional Gaussian distribution with a certain expected value, i.e. the amplitude of the center of

the hotspot, and with the variances in each direction, which determine the shape of the hotspot. But from the high density recordings one can also tell that the ratio from the variance in x -direction to the variance in y -direction stays constant. In addition to that in accordance with the direction of the muscle fiber the hotspot can exist in any rotation. In summary, a hotspot can be described by using the following parameters of a two dimensional Gaussian distribution:

1. Expected value μ (Amplitude in the center)
2. Variance in x -direction σ_x and variance in y -direction σ_y
3. Angle of rotation θ

To determine every value at position x and y of a two dimensional Gaussian distribution the following equation is used [36]:

$$f(x, y) = \mu \times \exp(-(a(x - x_0)^2 + 2b(x - x_0)(y - y_0) + c(y - y_0)^2)),$$

where x_0 and y_0 are the coordinates of the center of the blob and where the parameters a , b and c can be expressed as:

$$\begin{aligned} a &= \frac{\cos^2 \theta}{2\sigma_x^2} + \frac{\sin^2 \theta}{2\sigma_y^2}, \\ b &= -\frac{\sin 2\theta}{4\sigma_x^2} + \frac{\sin 2\theta}{4\sigma_y^2}, \\ c &= \frac{\sin^2 \theta}{2\sigma_x^2} + \frac{\cos^2 \theta}{2\sigma_y^2}. \end{aligned}$$

Exemplarily, figure 2.15 shows three different hotspot templates generated by using the equations above and varying parameters x_0 , y_0 , μ , σ_x , σ_y , and θ according to table 2.1.

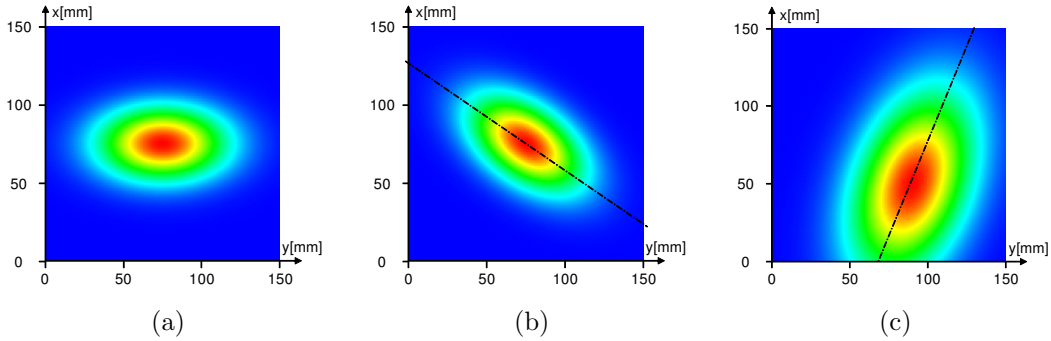


Figure 2.15: Three different templates, generated by using distinct parameters, which can be seen from table 2.1.

Data Set	Position	μ	σ_x	σ_y	θ
(a)	(75/75)	10	16	28	0°
(b)	(75/75)	10	16	28	35°
(c)	(50/90)	10	24	42	110°

Table 2.1: Parameters of the three example templates (within this table, all length specifications are in mm).

2.4.2.2 Basic Procedure

In order to determine the hotspot's position, template matching uses a scan procedure, which shifts the center of a hotspot, i.e. a uniquely generated template, over the recorded electrode values. This scan is necessitated irrespective of the calculation method used afterwards. The scan starts by aligning the center of the template to the top left corner of the electrode array. Subsequently every scan point is approached column by column and row by row until the center of the template reaches the bottom right corner of the electrode array. Figure 2.16 shows the scan procedure graphically.

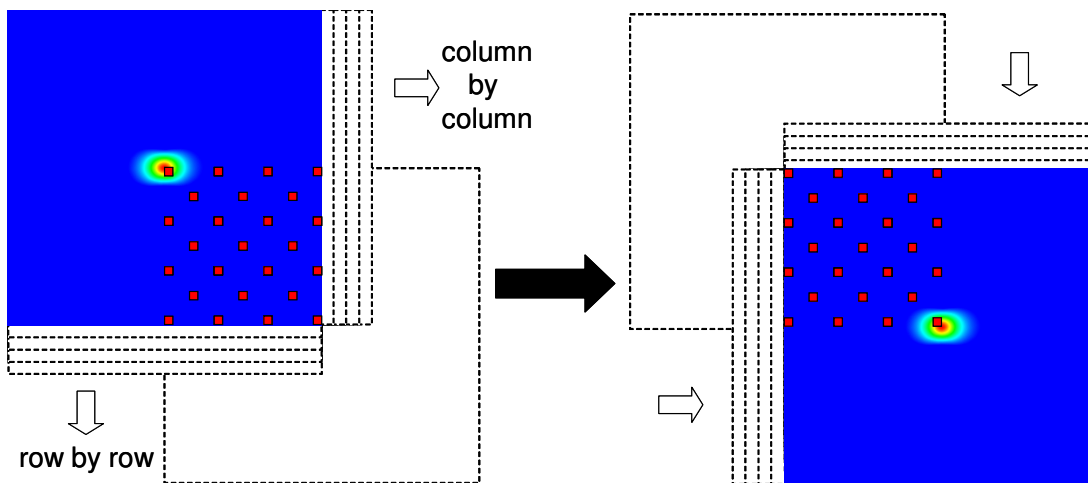


Figure 2.16: Template matching - scan procedure: The template is shifted column by column and row by row over the electrode array (every red dot illustrates an electrode).

The accuracy of the scan is determined by its steps and was therefore chosen in order to get an appropriate resolution for the localization of the hotspot, but to keep the computation time as low as possible. Therefore the distance between every scan point was set to 1mm. Since the electrodes span an area of $150\text{mm} \times 150\text{mm}$, 151×151 scan points are required. The calculation of each scan point on the one hand requires the electrode values at a discrete time and on the other hand the corresponding template values, i.e. the values aligned to the electrode positions, which change for every scan point. For further considerations

we introduce the following definitions: Let

$$P \in \mathbb{R}^{151 \times 151}$$

be the solution matrix containing the values for every single scan point and

$$E \in \mathbb{R}^{7 \times 7}$$

be the electrodes matrix. The information in E consists of 25 values, which are written to the corresponding positions of the matrix. To make it more vivid, figure 2.17 is a graphical representation of matrix E , where gray squares denote the 25 electrode values (E01 - E25), whereas white squares represent the inter electrode areas without any values. Since template matching is applied to electrode values at a discrete time, matrix E stays constant during the whole scan procedure.

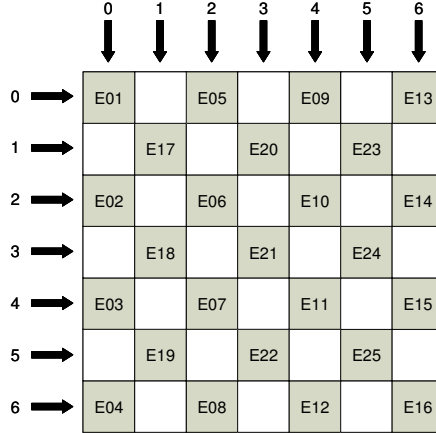


Figure 2.17: Positions of the electrode values (E01 - E25) within matrix E .

Analogously we let

$$T(x, y) \in \mathbb{R}^{151 \times 151}$$

be the matrix containing the template information, which is required to perform the calculation for every single scan point, where $x \in \{0, 1, 2, \dots, 150\}$ and $y \in \{0, 1, 2, \dots, 150\}$ describe the scan point's position. In further consequence we let

$$K := \{(0, 0), (2, 0), (4, 0), \dots, (6, 6)\}$$

be the set of the electrode values's positions within matrix E and the template information's positions within matrix T respectively, then every scan point of P is derived from a function of T and E by

$$P_{x,y} := \sum_{(a,b) \in K} f(E_{a,b}, T(x, y)_{a,b}). \quad (2.20)$$

As previously mentioned, the calculation of $f(E_{a,b}, T(x, y)_{a,b})$ depends on the method used, which is explained in detail in the next two subchapters.

Since shape, size and rotation of a hotspot vary from movement to movement and from patient to patient respectively, different templates have to be used in order to get the best result. Therefore the size of the template is changed in four steps, by modifying σ_x and σ_y and in addition to that the rotation of the template is changed as well. Since the rotation is modified in six steps for every size, a total of 24 templates is used. Depending on the calculation method used, a decision about the most probable position, size and rotation of the hotspot can be made by searching all matrices P , derived from each template, for the minimum (sum of absolute differences) or maximum (normalized cross correlation) value. Since the search for the hotspot is performed in real time it is important to reduce the computational cost to a minimum. Therefore the scan procedure is divided into a rough and a precise scan. The rough scan extends the distance from one scan point to another to 10mm, which makes only 16×16 scan points for the whole area of $150\text{mm} \times 150\text{mm}$. In addition to that, the detection of the size and the rotation of the hotspot is performed within the rough scan as well. Subsequently, the precise scan uses the these findings (best match, size, and rotation) and refines the scan in the region of the best match by reducing the distance between the scan points to 1mm. Compared to the basic procedure without a partition to a rough and precise scan, this approach reduces the computational cost by a factor of 80.

2.4.2.3 Sum of Absolute Differences

One approach to calculate the alignment's extent of the template and the electrode values for each scan point is to sum up the differences between them. Refining equation 2.20 the calculation of every scan point can now be denoted as

$$P_{x,y} = \sum_{(a,b) \in K} |E_{a,b} - T(x, y)_{a,b}|.$$

If the sum of absolute differences of a particular scan point is low, i.e. the recorded electrode values resemble the template values, the likelihood for this scan point of being a hotspot is high and vice versa. Therefore matrix P , containing the sum of differences of each scan point, is searched for its minimum value in order to find the hotspot.

2.4.2.4 Normalized Cross Correlation

Another approach to identify similarities between a template and the electrode values is to use the normalized cross correlation method. It computes the probability of a match by performing a discrete 2D correlation of the two datasets [37].

Therefore equation 2.20 is refined to:

$$P_{x,y} = \frac{\sum_{(a,b) \in K} [E_{a,b} - \bar{E}] [T(x,y)_{a,b} - \bar{T}]}{\sqrt{\sum_{(a,b) \in K} [E_{a,b} - \bar{E}]^2 \sum_{(a,b) \in K} [T(x,y)_{a,b} - \bar{T}]^2}},$$

where \bar{E} is the mean of E , which can be written as

$$\bar{E} = \frac{1}{|K|} \sum_{(a,b) \in K} E_{a,b}$$

and \bar{T} is the mean of the template, which depends on the size of the hotspot used to generate the template. If the correlation coefficient of a particular scan point is high the likelihood for this scan point of being a hotspot is high and vice versa. So in this case matrix P has to be searched for the maximum value in order to find the most probable hotspot position.

2.4.2.5 Implementation to the Source Code

Figure 2.18 illustrates all necessary steps within the software. As an input the electrode values are passed to the method, which are used to create the electrode matrix E . After that the scan is performed step by step and $P_{x,y}$ is determined for every scan point by using E and T . If the scan is completed the hotspot position is searched in P and provided as an output.

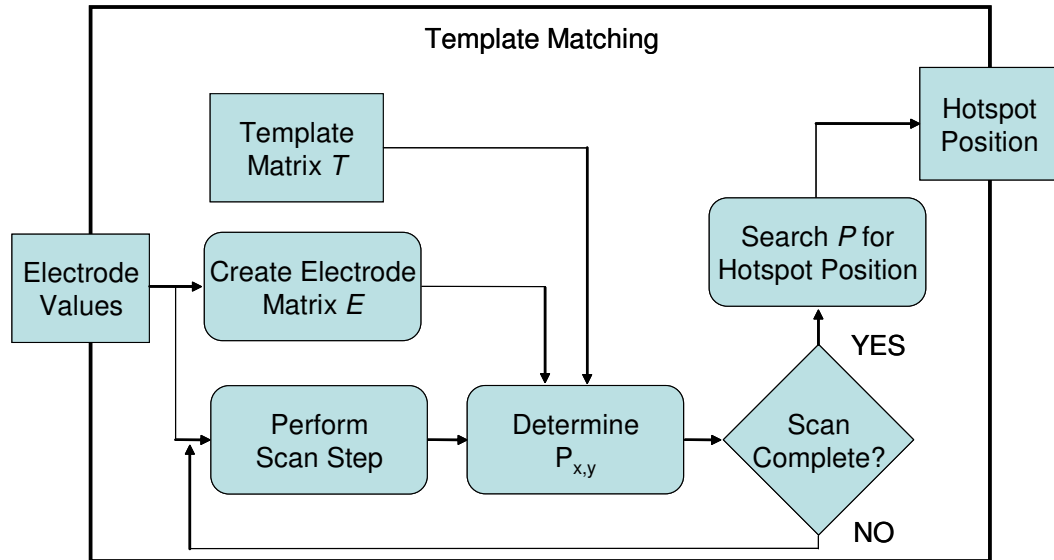


Figure 2.18: Template matching - implementation to the source code. This flow chart shows all necessary steps in order to get the hotspot position from the electrode values.

2.4.3 Dealing with faulty electrodes

As previously mentioned electrode lift-offs cause faulty electrode signals, which distort the localization of hotspots. To deal with this problem, at first it is important to identify whether an electrode works properly or not. For this purpose the user analyses the electrode signals and based on this analysis he tags the affected electrodes. Afterwards, how to treat the absence of electrodes depends on the hotspot localization method. Whereas bicubic interpolation relies on all electrode values and therefore can not just skip faulty electrodes, template matching is a more robust approach, which can deal with the absence of some electrodes. Subsequently, there are two ways of dealing with faulty electrodes:

- Skip faulty electrodes (suitable for template matching)
- Estimate faulty electrodes (suitable for bicubic interpolation and template matching)

The first approach can be achieved by reducing the amount of values used for the summation within equation 2.20 to the amount of properly working electrodes. The second one estimates all faulty electrodes, by taking all properly working neighboring electrodes into account. After that the estimated values are treated as properly working electrodes. The comparison of the two approaches and hence the choice which one to use, is presented in chapter 3.1.

2.5 Visualization

The visualization is an important part of this work, because it converts all computations mentioned above into information comprehensible for the user. Therefore it should provide as much information as possible on a display as simple as possible, in order not to overstrain the user with useless information. The following subchapters deal with the visualization of the sEMG signals on one hand and on the other hand the illustration of the hotspot position itself is discussed as well.

2.5.1 Visualization of sEMG Signals

The visualization of all sEMG signals on a graph is not directly relevant to the projects purpose of visualizing hotspots. But since from its readings one can tell whether all electrodes work properly, it is necessary to visualize them as well. Therefore several settings can be made within the software:

- Signal conditioning: display of the raw signal or the conditioned signal (RMS).
- Zoom: the zoom into the signal can be varied (variation in x -axis).

- Time axis: different time values (5s, 10s, 25s, 50s) can be selected (variation in y -axis).
- Signal representation: signal moves from the left to the right or it appears continuously from the right.

Since the ADCs provide 1000 samples per second on each channel, which would require at least 1000 pixels in direction of the time axis in order to visualize one second, the visualization of the raw signal has to use a simplified description of the signal. Therefore a compression method has to be applied, which ensures the smallest possible loss of information. Depending on the time axis settings, the signal is split into small blocks and from each of which the minimum and maximum values are taken. Hence every block can be represented by one pixel in time axis by plotting a line between minimum and maximum. This procedure can be seen in figure 2.19, which illustrates a block of 30 samples, its minimum and maximum values and the consequent line, representing the block on the graph. Joined together these lines result in the representation of the whole signal, which can be seen in figure 2.9 (a).

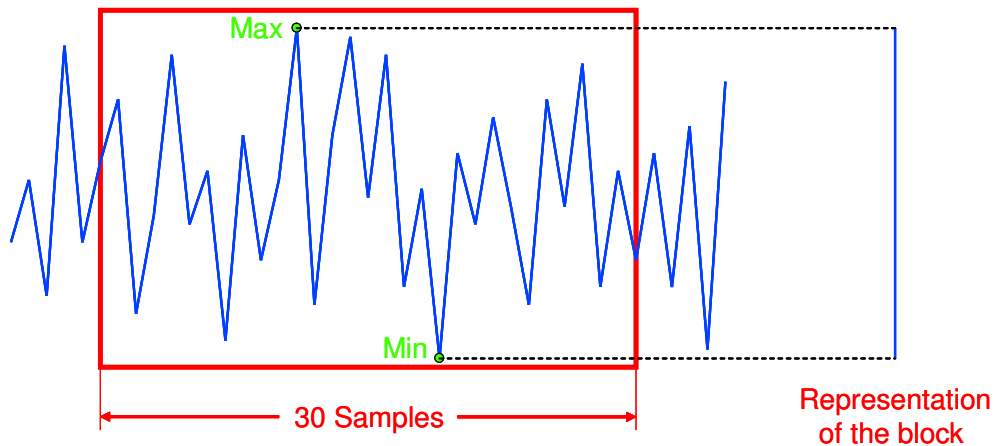


Figure 2.19: Compression of the raw signal. The red rectangle comprises 30 sample values, which have to be compressed. Therefore this block is represented by a line from its minimum to maximum value.

Since the conditioned signal already contains information about a certain range of the signal (see therefore chapter 2.2.2), the visualization of it is much easier. Depending on the time axis settings, a different amount of conditioned values of the signal is used. For small scales on the time axis, more values per second are required than for big scales. In the end on the graph these values are linked together using straight lines.

2.5.2 Visualization of the Hotspot

Since two different approaches, which provide disparate outputs, were used to find the hotspot, the occasion of two different ways of visualization arose as well. On one hand, bicubic interpolation provides a two dimensional array of the whole interpolated potential field and on the other hand template matching supplies the hotspots coordinates without information about the potential field distribution.

2.5.2.1 Visualization of Bicubic Interpolated Data

One approach of a graphical representation of individual values, contained in a two dimensional array, is called a heat map (Figure 2.23 (a)). It represents every value as color by using a specific color coding, which expands a two dimensional diagram by an additional dimension. To make the different values as distinguishable from each other as possible a color coding is used, which utilizes a broad spectral range from blue via green through to red. In addition to that, this kind of color coding supports the natural human desire of construing blue to cold (i.e. low values) and red to hot (i.e. high values). Figure 2.20 (a) shows the path the color coding takes within the three dimensional RGB color model, which adds the three additive primary colors red, green, and blue in various ways in order to generate a broad array of colors. Figure 2.20 (b) illustrates specific colors of intermediate steps at regular intervals.

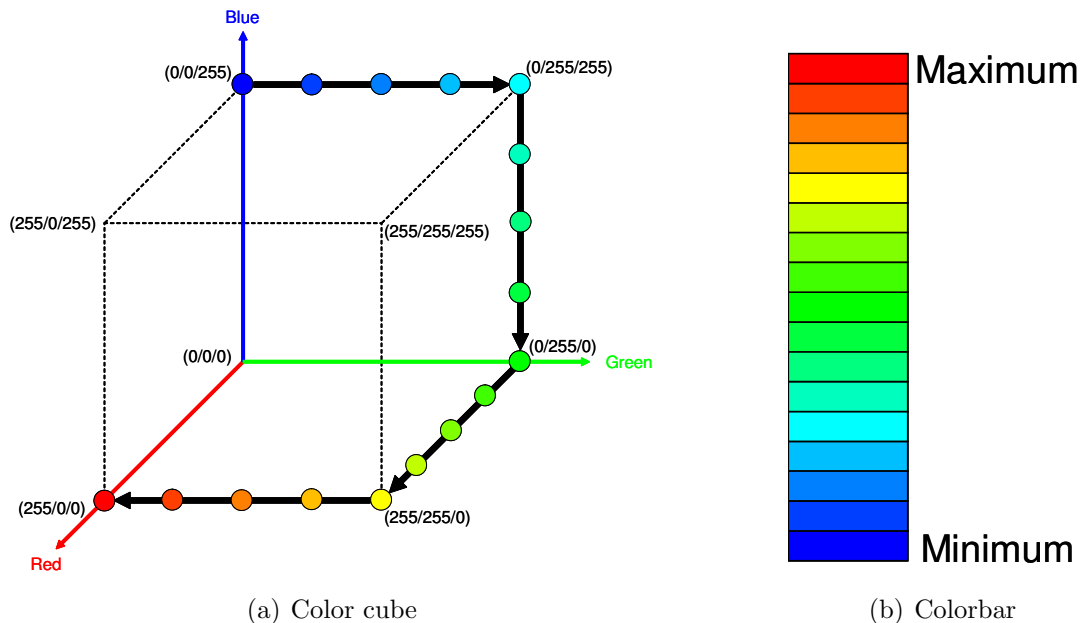


Figure 2.20: Color coding. (a) The path of the color coding in the three dimensional RGB color model from blue (0/0/255) via green (0/255/0) through to red (255/0/0). (b) The resulting colorbar of the color coding from blue (low values) to red (high values).

The implementation to the source code involves several steps. First of all the array derived from the interpolation is searched for its maximum and its minimum, in order to be able to normalize every value to a gray value in the range of 0 to 255 by applying following equation:

$$GV_{x,y} = \frac{IV_{x,y} - \min}{\max - \min} \times 255,$$

where $GV_{x,y}$ is the gray value and $IV_{x,y}$ is the interpolation value at position (x, y) . Subsequently this gray value is converted to a RGB value according to the path of the color coding in the three dimensional RGB model, which results in a total of 256 possible colors from blue to red. Ensuing pixel by pixel is drawn to the graphical user interface (GUI). Figure 2.21 illustrates all necessary steps within the software. Additionally, the electrode positions and a grid is drawn on the heat map as well, in order to provide the user with some aid to orientation. To encapsulate the essential information, the interpolated potential field is only displayed at periods of movement.

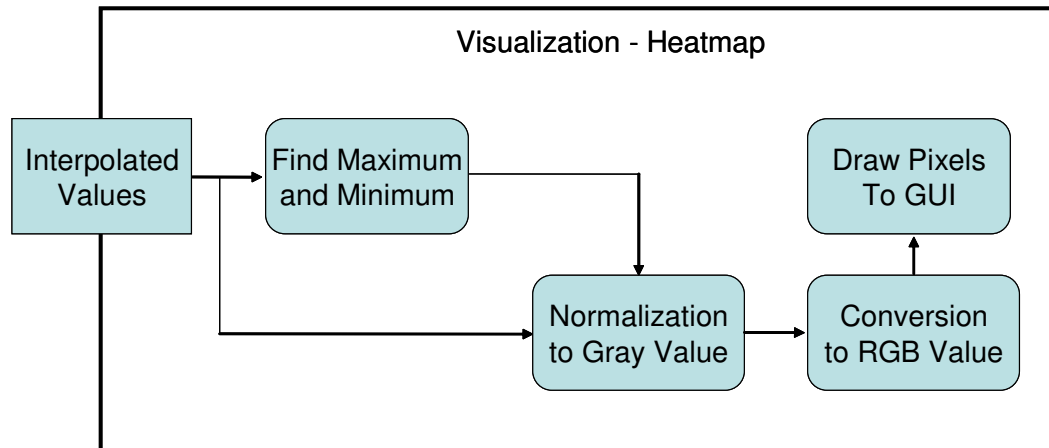


Figure 2.21: Visualization - implementation to the source code. This flow chart shows all steps necessary to visualize the results from bicubic interpolation on a heat map

2.5.2.2 Visualization of Results Derived from Template Matching

The visualization of the hotspot positions derived from template matching uses the current hotspot position on one hand and on the other hand the path of previous hotspots is also taken into account (Figure 2.23 (b)). Again, the essential information is encapsulated. Therefore, hotspots detected between movements are displayed in red and do not contribute to the hotspot path. The essential information itself, i.e. the hotspots detected while the patient performs the given movements, is divided into the single movements (e.g. flexion and extension) by classing them with different colors. To make the path of previous hotspots

distinguishable from the current hotspot, brighter colors are used. Again, the visualization contains some aid to orientation, i.e. electrode positions, a grid and information about faulty electrodes, to increase the ease of use.

Additionally, another approach can be used in order to get a more vivid impression of the hotspot's position on the patient's skin. Therefore an image of the patient is added to the software and the previously mentioned illustration is projected to the corresponding positions onto the image (figure 2.23 (c)). For this purpose the original coordinates have to be translated to new coordinates on the image, which is achieved by vector algebra.

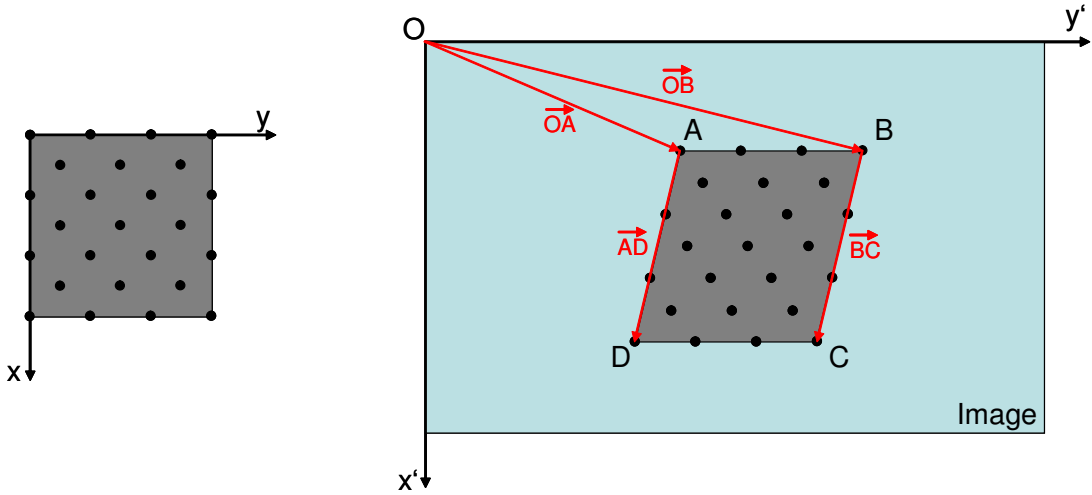


Figure 2.22: Projection onto an image. Vector algebra is used in order to translate the original coordinates x and y to the new coordinates x' and y' .

Therefore we let x and y be the original coordinates in the range of 0 to 150 and x' and y' be the new coordinates within the image. If O is the origin of the image and A , B , C , and D are the four corner points of the electrode array in the image (figure 2.22), then the coordinates (x, y) can be translated to (x', y') by using the following equation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \vec{OA} + \frac{x}{150} \vec{AD} + \left\{ \frac{y}{150} \left[(\vec{OB} + \frac{x}{150} \vec{BC}) - (\vec{OA} + \frac{x}{150} \vec{AD}) \right] \right\}.$$

In order to obtain the four corner points, after loading the image, the user is asked to manually teach them to the software by clicking on the corresponding locations of the image. Since an image, containing the exact positions of the electrode array, can only be taken after the measurements, this kind of visualization can not be applied on currently acquiring data and is therefore only used on recorded data in order to give the user the possibility to review it in a more descriptive way. In summary, figure 2.23 shows the different outputs of the three visualization approaches described above at the same point of time.

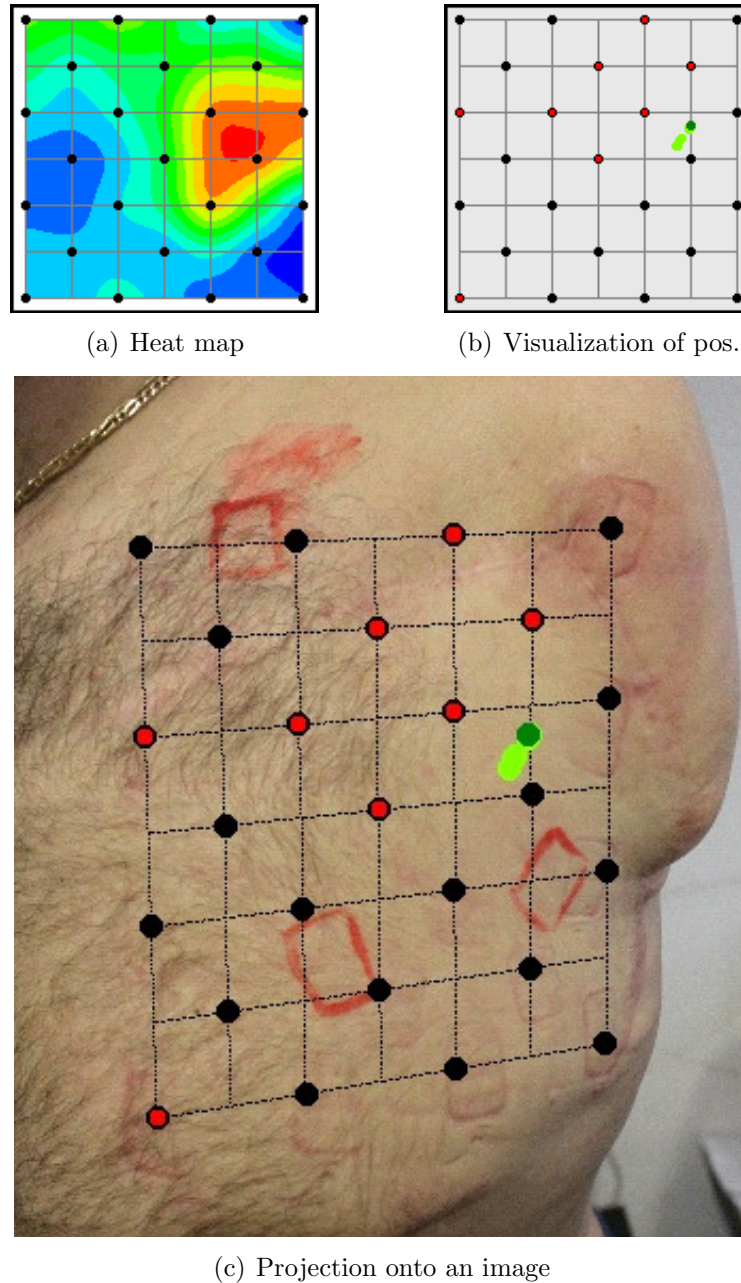


Figure 2.23: Visualization results. (a) Heat map, showing the computed potential field of a bicubic interpolation. (b) Visualization of a hotspot (dark green) and its path (green) derived from template matching. Faulty electrodes, e.g. electrodes which lift off, are tagged red. (c) Projection of the same outcome onto an image of the patient, by applying vector algebra.

2.6 Overview of the Software

To get an impression of how the different parts of the software work together, this chapter briefly gives an overview of the whole software developed within this

project. A detailed description of the most important classes and functions can be seen in appendix A. Figure 2.24 shows a flow chart comprising the four main tasks of the software.

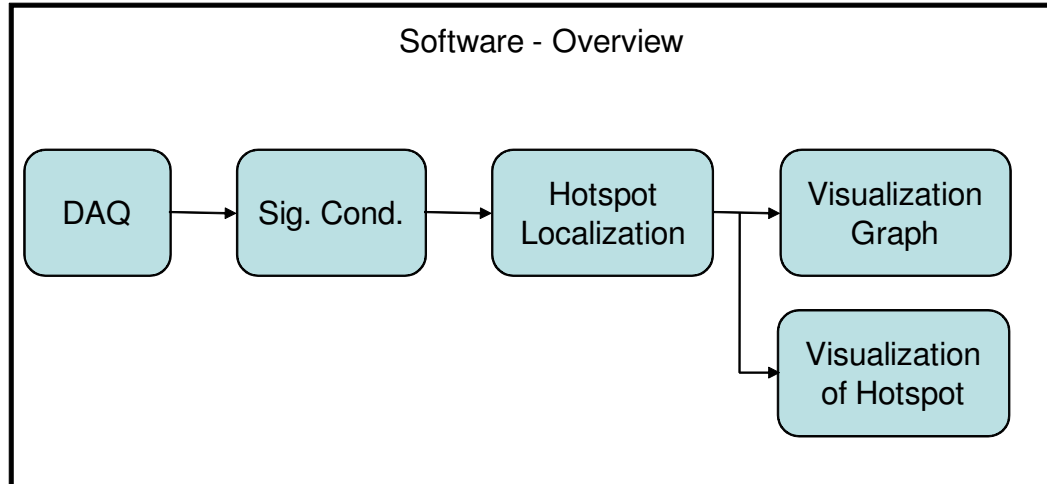


Figure 2.24: Software overview. This flow chart shows the four main tasks of the software, which are the data acquisition (DAQ), the signal conditioning, the hotspot localization and the visualization of the results.

Since this sequence of operations starts with the data acquisition, it will be processed every 100ms. The reason for that is the callback rate and the sampling rate of the ADCs, which deliver data blocks of 100 values using a sampling rate of 1kHz leading to a call of the function every 100ms. After acquiring the data, signal conditioning is applied in order to get interpretable values. Those values are passed on to the localization method. As mentioned above two different localization methods were implemented and therefore depending on the settings made, bicubic interpolation or template matching is used. The last part is the visualization of the results, which again is divided into two subparts: the visualization of the sEMG signals on graphs and the visualization of the actual hotspot location.

CHAPTER 3

Results

After describing the methodology in the previous chapter, this chapter approaches the results derived from this methods. The first subchapter provides information about the choice, which has been made concerning the localization method. Subsequently, the patient test is explained. Therefore the preparations, the test itself and hence the results of the hotspot localization of real data are shown and elucidated.

3.1 Hotspot Localization

Since two different approaches were tried out in order to find the hotspot, the advantages and disadvantages of them had to be figured out, in order to find the one approach which fits the project's purpose the best. Therefore, they were tested using equal datasets and comparing the results to each other. Additionally, their behavior to different electrode arrangements and faulty electrodes respectively was evaluated. In the first place the following two subchapters deal with the ruggedness of each method itself and subsequently the last subchapter compares them with each other and elucidates the final decision. Since the purpose of this project was locating the hotspot, the score for validation was the distance between the actual hotspot and the hotspot position derived from the localization method. As previously mentioned the debugging of the two methods was accomplished using simulated data, because in this way the actual hotspot position is already known and the conditions stay constant without any disturbance (e.g. noise).

3.1.1 Bicubic Interpolation

Bicubic interpolation was tested on four different electrode arrangements (see figure 3.1). The one comprising 16 electrodes was used in the beginning of the project and the final version of the electrode array uses 25 electrodes. In addition to that, two more electrode arrangements, comprising 49 and 121 electrodes respectively, were tested, in order to figure out whether significant improvements can be achieved by this means. Nevertheless, due to the currently used electrode's physical dimensions, the last two arrangements are not feasible in this project. The black dots within the figures represent the electrode positions and the red dots show the different positions of the hotspots of dataset (a) to (h) (see table 3.1), which

where used in order to test bicubic interpolation. As one can see, positions close to electrodes as well as positions far away from them exist for every electrode arrangement.

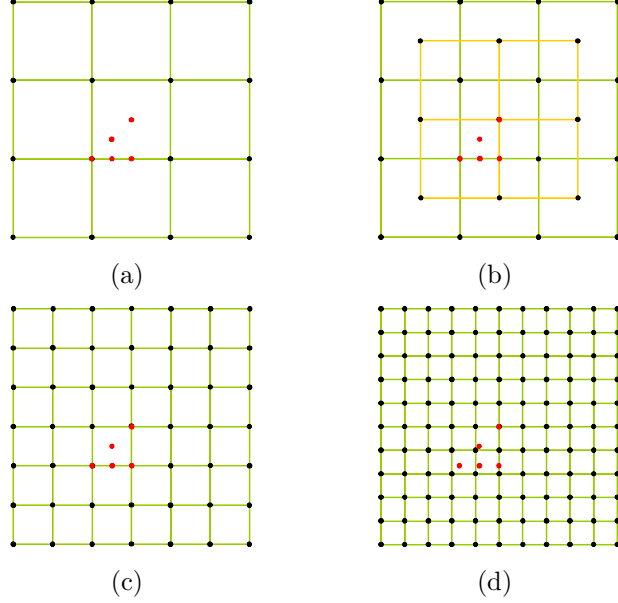


Figure 3.1: Electrode arrangements used for testing bicubic interpolation. (a) 16 electrodes (inter electrode distance = 5cm) (b) 25 electrodes (inter electrode distance = 3.54cm) (c) 49 electrodes (inter electrode distance = 2.5cm) (d) 121 electrodes (inter electrode distance = 1cm)

In addition to the previously mentioned distance between the actual hotspot and the computed hotspot as a score for validation to compare bicubic interpolation and template matching, another score is introduced for the test of bicubic interpolation, in order to refine the comparability of the different electrode arrangements. It describes the difference between the actual data set (i.e. simulated potential field) and the interpolated data set. Therefore the sum of absolute differences (SAD) is used, from which the score can be calculated by

$$SADscore = \sum_{(x,y) \in C} |DS_{x,y} - I_{x,y}|,$$

where

$$C := \{(0, 0), (1, 0), (2, 0), \dots, (150, 150)\}$$

describes all data points of the 151×151 data sets. Matrix DS is the actual data set and matrix I the interpolated data. $SADscore$ can assign values from 0 to ∞ , whereby good interpolation results are represented by values comparatively close to 0.

Since the outcome of the localization method is heavily dependent on the size and the position of the hotspot, a wide range of different data sets was used, in order to get representative results. The most critical hotspot to locate is small and situated exactly between two electrodes. Therefore the datasets used, vary in size (focus on small hotspots), rotation, and position (focus on position between electrodes) of the hotspot. Figure 3.2 shows the eight simulated hotspots, characterized by the different parameters of a two dimensional gaussian distribution, which can be seen from table 3.1.

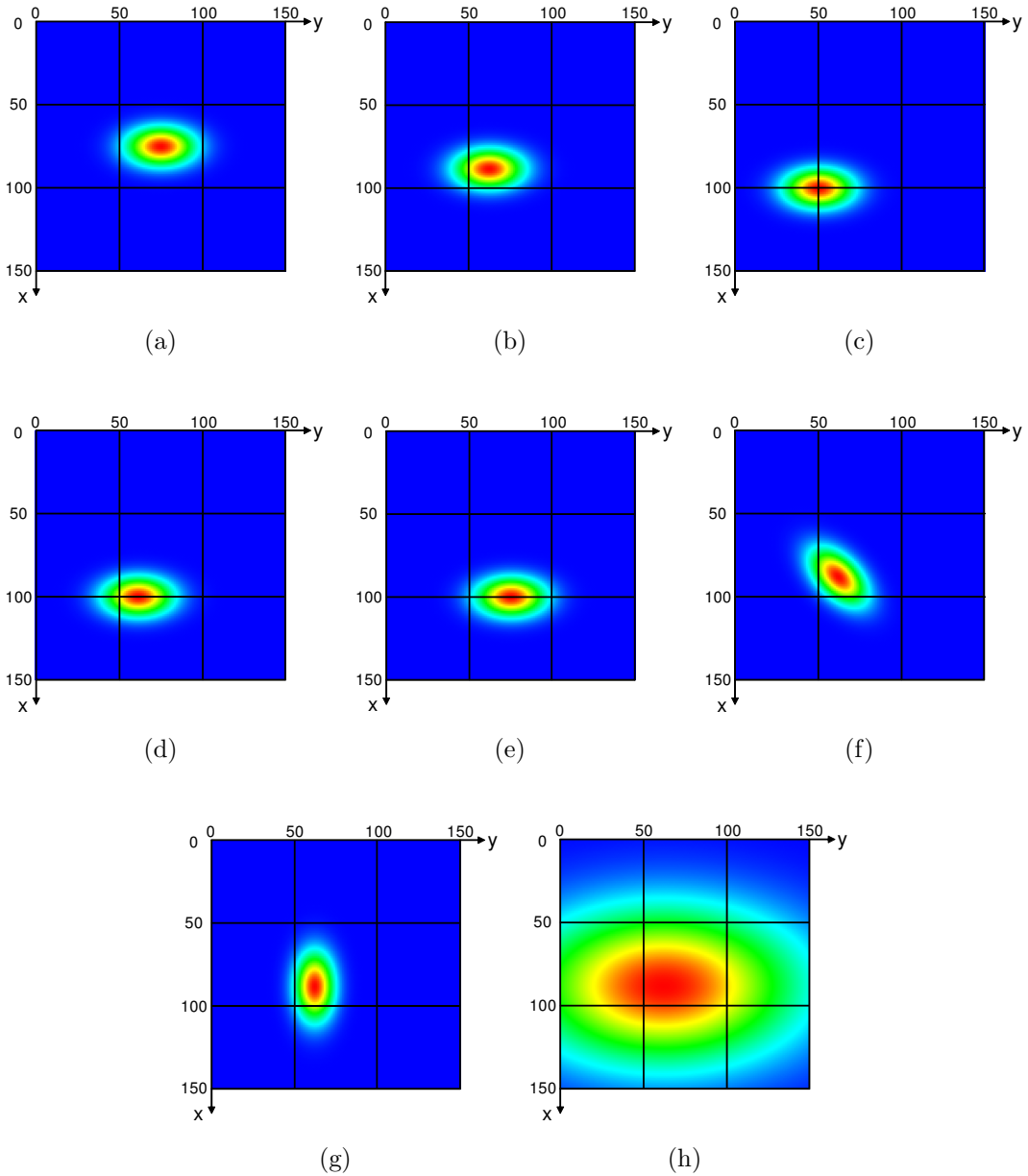


Figure 3.2: Datasets for testing bicubic interpolation. The parameters of each dataset can be seen in table 3.1.

Data Set	Position	σ_x	σ_y	θ
(a)	(75/75)	8	14	0°
(b)	(88/62)	8	14	0°
(c)	(100/50)	8	14	0°
(d)	(100/62)	8	14	0°
(e)	(100/75)	8	14	0°
(f)	(88/62)	8	14	45°
(g)	(88/62)	8	14	90°
(h)	(88/62)	32	56	0°

Table 3.1: Datasets used for testing bicubic interpolation.

Table 3.2 shows the results (*SADscore* and difference between actual hotspot position and computed hotspot position) of applying bicubic interpolation to the data sets declared in table 3.1, using different electrode arrangements. This scores for validation confirm the previously claimed statement, that the interpolation result strongly depends on the position and the size of the hotspot. This can be easily seen by comparing the difference between the actual hotspot position and the computed hotspot position of dataset (b) and (h) with each other. Both of them represent a hotspot at position (88,62) and a rotational angle θ of 0°, but they differ in size. One can easily see that the computed hotspot for bigger hotspots is much more accurate than that for smaller hotspots. Additionally it can be seen that on average both the *SADscore* and the difference improve, if the amount of electrodes is increased.

Furthermore, if the differences between the actual hotspot and the computed hotspot are taken into considerations, one can spot tremendous outliers, resulting in a localization problem. This problem arises in case of an occurrence of a hotspot of small size and a position far from electrodes. The reason for it is the fact, that the interpolation can not detect a connectedness between the two neighboring electrode values and therefore computes two small peaks instead of one single peak. Hence the result is distorted and the correct hotspot position can not be assigned anymore. Exemplarily, this behavior occurs using 25 electrodes and applying bicubic interpolation to data set (e), resulting in an outlier in the y -direction (difference of 23mm). To make it more vivid figure 3.3 shows both the data set and the interpolated potential field on a heat map. The most crucial point for this problem is the amount of electrodes used. The more electrodes are used the less outliers occur and hence the less is the likelihood of incorrect localization results.

To sum up, it can be said that even though using 49 electrodes achieves an average difference of less than 3mm in both directions, due to its outliers up to 13mm it is not suitable for hotspot localization. Since their behavior is much worse, the same can be said for 16 and 25 electrodes respectively. Using 121 electrodes would provide good results, but as mentioned above it is not feasible in practice,

Electrode Arrang.	Data Set					SAD Score	HS (X/Y)	Diff. (X/Y)
	Label	Pos.	σ_x	σ_y	θ			
16 Electr.	(a)	(75/75)	8	14	0°	7017	(43/66)	(32/9)
	(b)	(88/62)	8	14	0°	8473	(99/50)	(11/12)
	(c)	(100/50)	8	14	0°	24625	(100/50)	(0/0)
	(d)	(100/62)	8	14	0°	16686	(100/50)	(0/12)
	(e)	(100/75)	8	14	0°	11319	(99/71)	(1/4)
	(f)	(88/62)	8	14	45°	7221	(97/50)	(9/12)
	(g)	(88/62)	8	14	90°	8473	(99/50)	(11/12)
	(h)	(88/62)	32	56	0°	10551	(96/57)	(8/5)
Average						11796		(9/8.3)
25 Electr.	(a)	(75/75)	8	14	0°	10169	(75/75)	(0/0)
	(b)	(88/62)	8	14	0°	6843	(97/52)	(9/10)
	(c)	(100/50)	8	14	0°	9560	(100/50)	(0/0)
	(d)	(100/62)	8	14	0°	8257	(100/50)	(0/12)
	(e)	(100/75)	8	14	0°	7549	(99/52)	(1/23)
	(f)	(88/62)	8	14	45°	6747	(97/51)	(9/11)
	(g)	(88/62)	8	14	90°	6844	(97/52)	(9/10)
	(h)	(88/62)	32	56	0°	3950	(89/59)	(1/3)
Average						7490		(3.6/8.6)
49 Electr.	(a)	(75/75)	8	14	0°	3501	(75/75)	(0/0)
	(b)	(88/62)	8	14	0°	5602	(92/60)	(4/2)
	(c)	(100/50)	8	14	0°	3444	(100/50)	(0/0)
	(d)	(100/62)	8	14	0°	4405	(100/60)	(0/2)
	(e)	(100/75)	8	14	0°	3500	(100/75)	(0/0)
	(f)	(88/62)	8	14	45°	5090	(75/50)	(13/12)
	(g)	(88/62)	8	14	90°	5602	(88/57)	(0/5)
	(h)	(88/62)	32	56	0°	1546	(88/61)	(0/1)
Average						4086		(2.1/2.8)
121 Electr.	(a)	(75/75)	8	14	0°	465	(75/75)	(0/0)
	(b)	(88/62)	8	14	0°	1001	(90/61)	(2/1)
	(c)	(100/50)	8	14	0°	1958	(103/48)	(3/2)
	(d)	(100/62)	8	14	0°	1957	(103/61)	(3/1)
	(e)	(100/75)	8	14	0°	1957	(103/75)	(3/0)
	(f)	(88/62)	8	14	45°	1284	(89/61)	(1/1)
	(g)	(88/62)	8	14	90°	1001	(89/60)	(1/2)
	(h)	(88/62)	32	56	0°	847	(88/60)	(0/2)
Average						1309		(1.6/1.1)

Table 3.2: Results of testing bicubic interpolation (within this table, all length specifications are in *mm*).

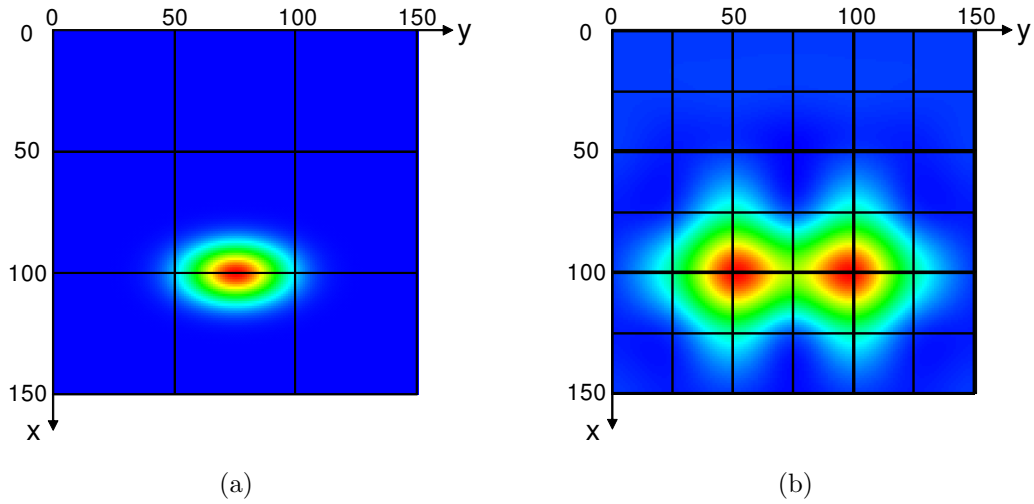


Figure 3.3: Localization problem. (a) Dataset of a hotspot located far from the electrodes (dataset (e) in table 3.1) (b) Bicubic interpolated potential field resulting in two peaks, which leads to a wrong hotspot position.

due to the electrode’s physical dimensions.

3.1.2 Template Matching

Before comparing it to bicubic interpolation, template matching itself had to be tested previously, in order to find out whether sum of absolute differences or normalized cross correlation is the first choice. Therefore other simulated hotspots than the previously used were utilized, because in this case the focus had to be on different rotational angles and different sizes of the hotspot, instead of varying positions. Therefore as a first step the position is kept constant at $x = 88$ and $y = 62$, which corresponds to a position exactly between two electrodes. Additionally the size is kept constant as well, whereas the rotation is varied in six steps. Figure 3.4 shows all datasets used for testing template matching.

Table 3.3 illustrates the localization results if sum of absolute differences (SAD) and normalized cross correlation (NCC) respectively is applied to the mentioned data sets. From this table one can see, that the SAD shows extraordinarily good results, whereas the NCC performs poorer but does also feature good outcomes. The average difference between the actual hotspot position and the computed hotspot position is very low for both approaches. This issue was expectable, because the template matching algorithm uses exactly the same steps in variation of angle of the hotspot (see chapter 2.4.2). Therefore the approaches had to be tested on further datasets, which use varying sizes and in addition to that the shape of the hotspots is varied as well (by altering the ratio σ_x to σ_y). Furthermore the rotational angle is set to a value between the algorithm’s steps, i.e. 15° . Table 3.4 shows the localization results of these datasets. From this table one can see,

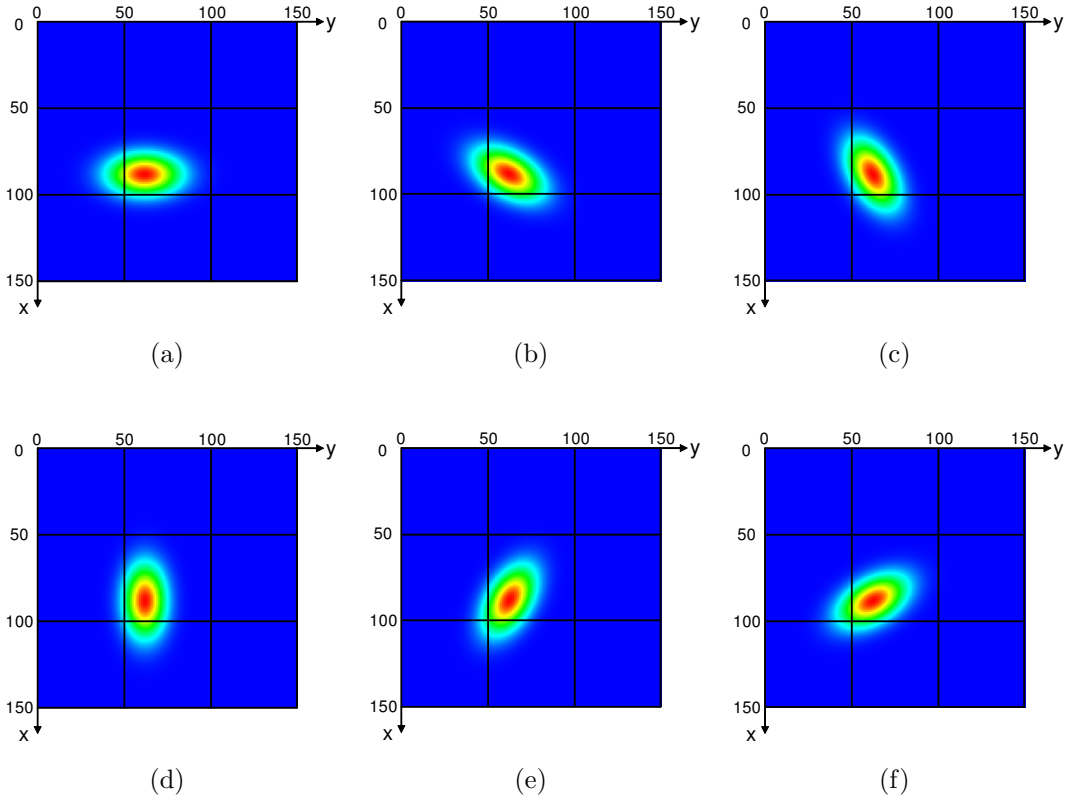


Figure 3.4: Datasets used for testing template matching. They are featuring the same hotspot size but different rotational angles in order to test template matching.

that NCC is a more rugged method than SAD, which faces tremendous outliers up to 17mm. In contrast to the testing of bicubic interpolation, an enhancement of the amount of electrodes used, was not necessary, because at the one hand the results of a 25-electrodes array are fair enough and on the other hand using more electrodes is not possible in practice anyway.

Data Set					SAD HS	Diff.	NCC HS	Diff.
Label	Pos.	σ_x	σ_y	θ	(X/Y)	(X/Y)	(X/Y)	(X/Y)
(a)	(88/62)	8	14	0°	(88/62)	(0/0)	(87/58)	(1/4)
(b)	(88/62)	8	14	30°	(88/62)	(0/0)	(88/59)	(0/3)
(c)	(88/62)	8	14	60°	(88/62)	(0/0)	(91/62)	(3/0)
(d)	(88/62)	8	14	90°	(88/62)	(0/0)	(92/63)	(4/1)
(e)	(88/62)	8	14	120°	(88/62)	(0/0)	(88/62)	(0/0)
(f)	(88/62)	8	14	150°	(88/62)	(0/0)	(88/62)	(0/0)
Average						(0/0)		(1.3/1.3)

Table 3.3: Results of testing template matching, using constant hotspot sizes (within this table, all length specifications are in mm).

Data Set					SAD HS	Diff.	NCC HS	Diff.
Label	Pos.	σ_x	σ_y	θ	(X/Y)	(X/Y)	(X/Y)	(X/Y)
(a)	(88/62)	8	14	15°	(84/48)	(4/14)	(90/65)	(2/3)
(b)	(88/62)	10	17.5	15°	(85/54)	(3/8)	(89/61)	(1/1)
(c)	(88/62)	12	21	15°	(88/62)	(0/0)	(88/61)	(0/1)
(d)	(88/62)	14	24.5	15°	(85/59)	(3/3)	(88/61)	(0/1)
(e)	(88/62)	16	28	15°	(88/61)	(0/1)	(88/62)	(0/0)
(f)	(88/62)	8	17.5	15°	(85/52)	(3/10)	(89/60)	(1/2)
(g)	(88/62)	8	21	15°	(85/45)	(3/17)	(88/61)	(0/1)
(h)	(88/62)	12	24.5	15°	(88/62)	(0/0)	(88/61)	(0/1)
(i)	(88/62)	12	28	15°	(88/62)	(0/0)	(88/62)	(0/0)
Average						(1.8/5.9)		(0.4/1.1)

Table 3.4: Results of testing template matching, using varying hotspot sizes (within this table, all length specifications are in mm).

To sum up, the testing of the two different template matching approaches determined NCC to be the first choice. The reason for that is that over all it provides steadier results than SAD and due to its normalization it is more rugged to unstable data, which occurs due to noise or other artifacts (e.g. movements or electrocardiography (ECG) signals).

3.1.3 Final Decision

Since the usage of more than 25 electrodes is not feasible, the final decision focuses on the results of bicubic interpolation using the second electrode arrangement (figure 3.1 (b)) and compares them to the results from template matching using NCC, which was only tested on the second arrangement anyway. To get comparable results both bicubic interpolation and template matching were tested using the same datasets. Therefore the datasets from table 3.1 and those from table 3.4 were used again, which led to the results, shown in table 3.5. From this results one can say, that template matching is a much better approach than bicubic interpolation, because the average difference between the actual hotspot position and the one computed, is way lower and in addition to that template matching does not feature any outliers.

3.1.3.1 Performance to Faulty Electrodes

Since from the beginning the different build-ups of the electrode array caused electrode lift-offs, the algorithm’s response to that had to be tested as well, before the final decision could be made. Therefore faulty electrodes were artificially generated within simulated datasets by setting their values to 0. After that bicubic interpolation and template matching were tested by estimating faulty electrode

Data Set					BI HS	BI Diff.	TM HS	TM Diff.
Label	Pos.	σ_x	σ_y	θ	(X/Y)	(X/Y)	(X/Y)	(X/Y)
(a)	(75/75)	8	14	0°	(75/75)	(0/0)	(78/72)	(3/3)
(b)	(88/62)	8	14	0°	(97/52)	(9/10)	(89/63)	(1/1)
(c)	(100/50)	8	14	0°	(100/50)	(0/0)	(97/49)	(3/1)
(d)	(100/62)	8	14	0°	(100/50)	(0/12)	(100/61)	(0/1)
(e)	(100/75)	8	14	0°	(99/52)	(1/23)	(100/75)	(0/0)
(f)	(88/62)	8	14	45°	(97/51)	(9/11)	(86/60)	(2/2)
(g)	(88/62)	8	14	90°	(97/52)	(9/10)	(87/61)	(1/1)
(h)	(88/62)	32	56	0°	(89/59)	(1/3)	(88/63)	(0/1)
(i)	(88/62)	8	14	15°	(96/52)	(8/10)	(90/65)	(2/3)
(j)	(88/62)	10	17.5	15°	(96/52)	(8/10)	(89/61)	(1/1)
(k)	(88/62)	12	21	15°	(95/53)	(7/9)	(88/61)	(0/1)
(l)	(88/62)	14	24.5	15°	(94/54)	(6/8)	(88/61)	(0/1)
(m)	(88/62)	16	28	15°	(93/55)	(5/7)	(88/62)	(0/0)
(n)	(88/62)	8	17.5	15°	(95/51)	(7/11)	(89/60)	(1/2)
(o)	(88/62)	8	21	15°	(92/50)	(4/12)	(88/61)	(0/1)
(p)	(88/62)	12	24.5	15°	(94/53)	(6/9)	(88/61)	(0/1)
(q)	(88/62)	12	28	15°	(93/53)	(6/9)	(88/62)	(0/0)
Average						(5.1/9.1)		(0.8/1.2)

Table 3.5: Results of the comparison of bicubic interpolation and template matching (within this table, all length specifications are in mm).

values and in addition to that template matching was also tested by skipping faulty electrode values (see chapter 2.4.3). For this purpose two prominent datasets from those previously used, which caused a big difference between actual and computed hotspot (dataset (a) and (i) from table 3.5), were used again, in order to get the worst case. On each dataset different electrodes (single electrodes, but also a multitude of electrodes) were considered as faulty. Therefore figure 3.5 features all electrodes of the latest arrangement and their labeling and in addition to that the hotspot position of the two datasets are shown as well.

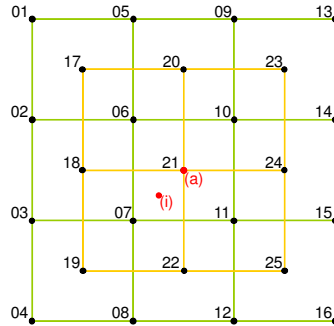


Figure 3.5: Electrode labeling and hotspot positions (dataset (a) and (i))

Table 3.6 shows the results of the tests using both estimated and skipped values, where column “Faulty Electrodes” states which electrodes were faulty. From the average differences between the actual hotspot and the one computed, one can easily see, that template matching performs much better than bicubic interpolation. Additionally one can tell, that using estimated values results in a slightly better result than skipping faulty electrode values. Furthermore this test revealed a dependency of the result on the location of the faulty electrode and the amount of faulty electrodes. Generally one can say, that the bigger the distance between actual hotspot and faulty electrode is, the less is its impact to the result. On the other hand, the result gets worse, the more electrodes are faulty.

Data Set	Faulty Electr.	Estimate Values		Skip Values	
		BI Diff.	TM Diff.	BI Diff.	TM Diff.
(a)	21	(25/25)	(0/0)	-	(0/0)
(a)	7	(0/0)	(3/2)	-	(6/9)
(a)	22	(0/0)	(4/0)	-	(11/0)
(a)	8	(0/0)	(3/4)	-	(6/0)
(a)	21+7	(25/25)	(4/4)	-	(2/2)
(a)	21+22	(25/25)	(1/0)	-	(0/0)
(a)	19+22	(0/0)	(1/0)	-	(12/0)
(a)	21+6+7	(25/25)	(0/13)	-	(0/7)
(a)	5+17+20	(0/0)	(3/0)	-	(12/0)
(i)	21	(11/12)	(3/7)	-	(4/11)
(i)	7	(14/12)	(8/2)	-	(5/13)
(i)	22	(10/10)	(5/1)	-	(1/5)
(i)	8	(9/11)	(2/3)	-	(1/5)
(i)	21+7	(15/38)	(7/20)	-	(1/3)
(i)	21+22	(11/12)	(12/3)	-	(4/10)
(i)	19+22	(10/10)	(9/2)	-	(1/5)
(i)	21+6+7	(15/38)	(4/8)	-	(3/3)
(i)	5+17+20	(9/11)	(2/3)	-	(1/5)
Average		(11.3/14.1)	(3.9/4)	-	(3.9/4.3)

Table 3.6: Results of testing bicubic interpolation and template matching (using NCC) at the presence of faulty electrodes (within this table, all length specifications are in mm). The parameters of dataset (a) and (i) can be seen from table 3.5.

To sum up, the prime selection for an electrode arrangement composed of 25 electrodes is template matching using normalized cross correlation. The main reason for that is the better behavior on normal conditions, but also its performance to faulty electrodes. In case of the latter estimating the missing values should be preferred to skipping them.

3.2 Test Using Simulated Data

Whereas the previous chapter focused on the detection of the best algorithm using non-temporal simulated data, this chapter describes the localization results of template matching using NCC on simulated data, which changes over time. Therefore two different sequences were made. The first one focuses on the movement of the hotspot in a wide range of the whole field and the second one focuses on varying sizes and rotational angles of the hotspot. In reality a hotspot would not take paths like shown in this simulations, but they are perfectly adapted for testing purposes. The following two subchapters describe the results derived by applying the algorithm to these datasets.

3.2.1 Results of Sequence 1

As mentioned above, the first sequence focuses on a wide range of the hotspot's movement. In this case the rotational angle of the hotspot was kept at 0° and its size was small (i.e. $\sigma_x = 16\text{mm}$ and $\sigma_y = 28\text{mm}$). The duration of the sequence was set to 22 seconds, which corresponds to 22.000 samples. Figure 3.6 (a) shows the path the hotspot takes. It starts at position (0/0), moves on to (150/0), to (0/25), to (150/25), to (0,75), and so on, until it reaches the end at position (150, 150). From figure 3.6 (b) one can see the localization result computed by the software and visualized on the GUI.

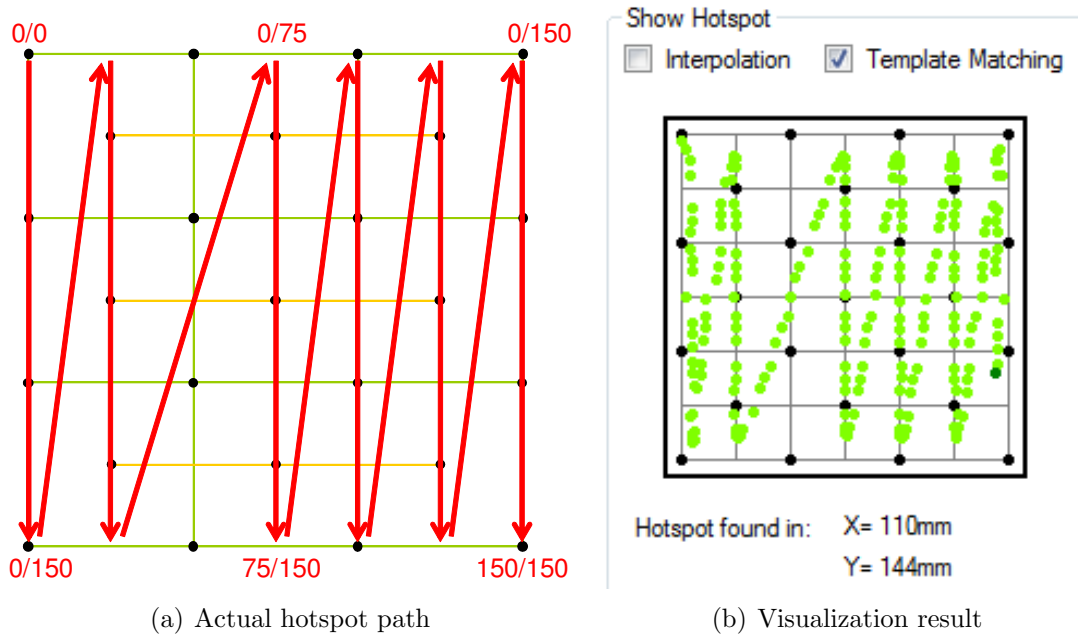


Figure 3.6: Sequence 1. (a) The red arrows illustrate the path the hotspot takes within the simulated data sequence. (b) Result of the localization algorithm, where light green dots indicate the path of the hotspot and the dark green dot is the current hotspot position.

The bright green colored dots indicate the path the hotspot took and the dark green colored dot shows the current hotspot position at the time the screenshot was taken. Additionally the exact current hotspot position can be seen from the labels below the visualization. From this result one can tell, that the border regions of the field cause bigger inaccuracies than the center region. The reason for that is the smaller amount of usable values at the border compared to the center. This behavior is inevitable and since the localization accuracy is still good, it can be neglected. The deviation of the computed path to the actual path (straight line) arises from the small inaccuracy of the localization algorithm. As already shown in the previous chapter the average difference between the actual hotspot and the one computed is about 1mm and hence can be neglected.

3.2.2 Results of Sequence 2

The focus of the second sequence was on a varying size and a varying rotational angle, while the hotspot is moving. Therefore a sequence was generated in which the hotspot moves along a given path (see figure 3.7 (a)). While moving, the hotspot constantly changes its size and its rotational angle. The angle was varied in the range of 0° to 180° and the size ranges from a $\sigma_x = 16\text{mm}$ to a $\sigma_x = 32\text{mm}$ and from a $\sigma_y = 28\text{mm}$ to a $\sigma_y = 56\text{mm}$. Again figure 3.7 (b) shows the localization result, which has been visualized on the GUI.

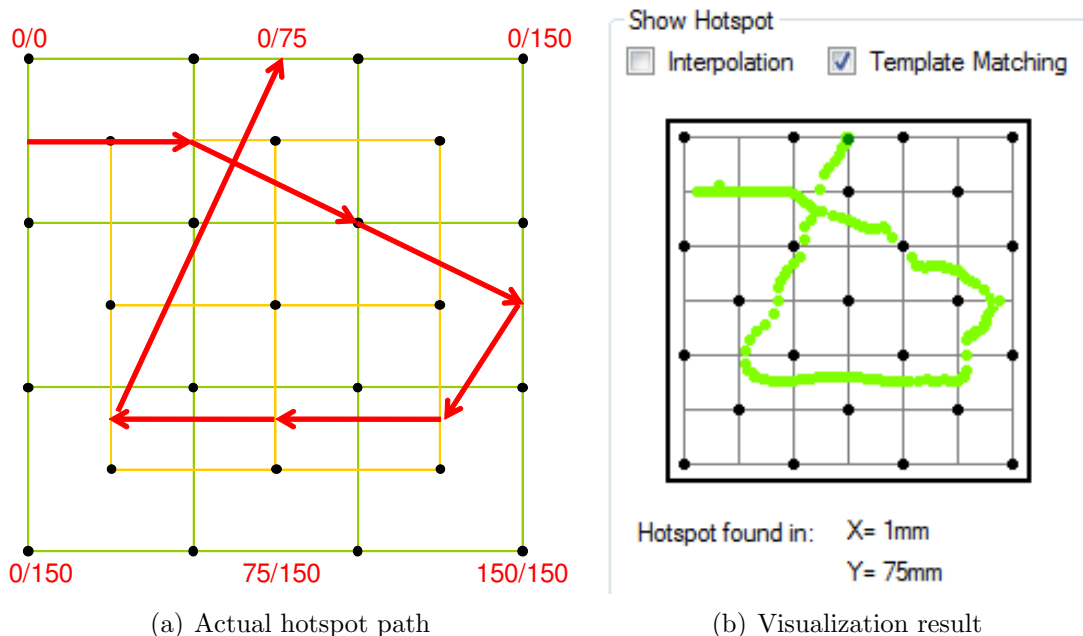


Figure 3.7: Sequence 2. (a) The red arrows show the path the hotspot takes. Along this path the size and the rotational angle change (b) Result of the localization algorithm, where bright green dots indicate the path of the hotspot and the dark green dot is the current hotspot position.

From this results one can tell, that although the size and the rotational angle of the hotspot change, the algorithm still determines the correct hotspot position. Again a small deviation from the actual path can be seen. As expected this deviation reaches its maximum, when the hotspot position is far away from any electrode.

3.3 Patient Tests - Real Data

Contrary to the previous chapter, this chapter elaborates on the actual purpose of the software, to apply this algorithm to real data in order to find hotspots on patients. Therefore the first subchapter provides information about the test set up and all preparations, which have to be made prior the test. Subsequently the next chapter deals with the patient test itself (i.e. the software settings or the desired movements of the patient). The last subchapter shows the results of a particular patient and explains them in detail.

3.3.1 Test Set-up and Preparations

Before starting the test, some agreements with the patient and requirements for the set-up have to be made, in order to create equal and hence comparable conditions for every test. Since the measurements strongly depend on a good contact between the electrodes and the patient's skin, all electrodes have to be tested, whether they provide sEMG signals. Furthermore one should wait at least 5 minutes between putting on the electrode matrix and starting the test, to reduce the influence of sweat or the heat up of the electrode's contact area to the signal, which will decrease over time. Supplementary the patient is told to adopt a natural standing upright posture. To get a visual feedback for the therapist, whether the patient performs the desired movement correctly, he is instructed to perform the movement bilateral. In case of a left-side amputee, he would perform the movement of his phantom limb in his mind and in addition to that physically move his right arm. Within the software some basic setting have to be made. Therefore patient related information, like the patient ID or the level of amputation, is entered. In addition to that information about the test itself, e.g. strength of contraction or electrode arrangement, has to be entered as well.

3.3.2 Test Procedure

The test itself uses movement sequences of the patient in order to assign hotspot locations to every single movement. These movement sequences were always constructed in the same way. Every sequence starts with a resting state, followed by the first movement, another resting state and at the end the second movement. Since patients need different time to perform the full range of a movement, this

time can be set in the software. Additionally this sequence is repeated several times in order to make it more representative, which can be also set in the software. Figure 3.8 graphically shows a sequence.

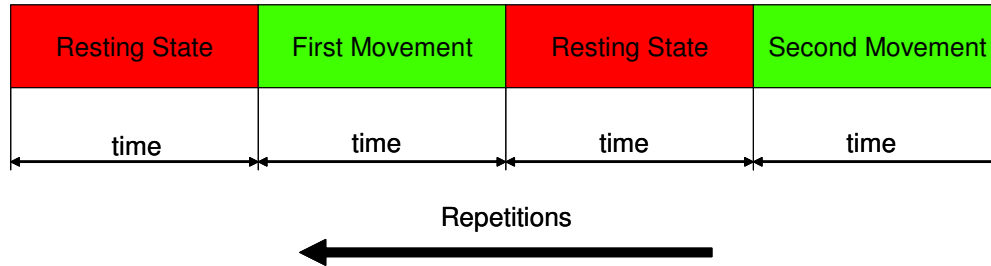


Figure 3.8: Movement sequence, starting with a resting state, followed by the first movement, another resting state and at the end the second movement.

First movement denotes the movement of a joint and second movement its counter movement. In case of the elbow, first movement would be flexion and second movement extension. Since TMR patients also perform wrist and hand movements and the testing additionally uses combinations of these movements as well, the following seven sequences can be achieved:

- Elbow (flexion and extension)
- Hand (close and open)
- Wrist (pronation and supination)
- Elbow and hand
- Elbow and wrist
- Hand and wrist
- Elbow, hand and wrist

Every test procedure contains all seven sequences and between every sequence a pause of about 1 minute is taken, in order to give the patient enough time to focus on the next movements. For later analysis the recorded data of every sequence is saved to a comma-separated values (CSV) file. To make the recordings more representative the test procedure (seven sequences) is repeated for several strengths of contraction (e.g. 100%, 75%, 50%, and 25% of MVC).

3.3.3 Evaluation of the Results

This chapter exemplarily shows a snippet of the results stemming from a patient test carried out on March the 25th, 2013. The patient is a 42 years old man, who suffers from a left-sided shoulder disarticulation. Since this patient already uses a prosthesis the hotspot positions are already known and therefore the computed positions can be compared to the actual hotspot positions. Figure 3.9 (a) shows the patient’s chest and additionally three tags, featuring the three electrode positions (hotspots) the patient already uses in order to control the prosthesis. These three hotspots are responsible for elbow flexion (FL), open hand (OH), and wrist pronation (PR) respectively. The hotspots for elbow extension, close hand and wrist supination, which in this case are not of interest, are located at the back and below the armpit.

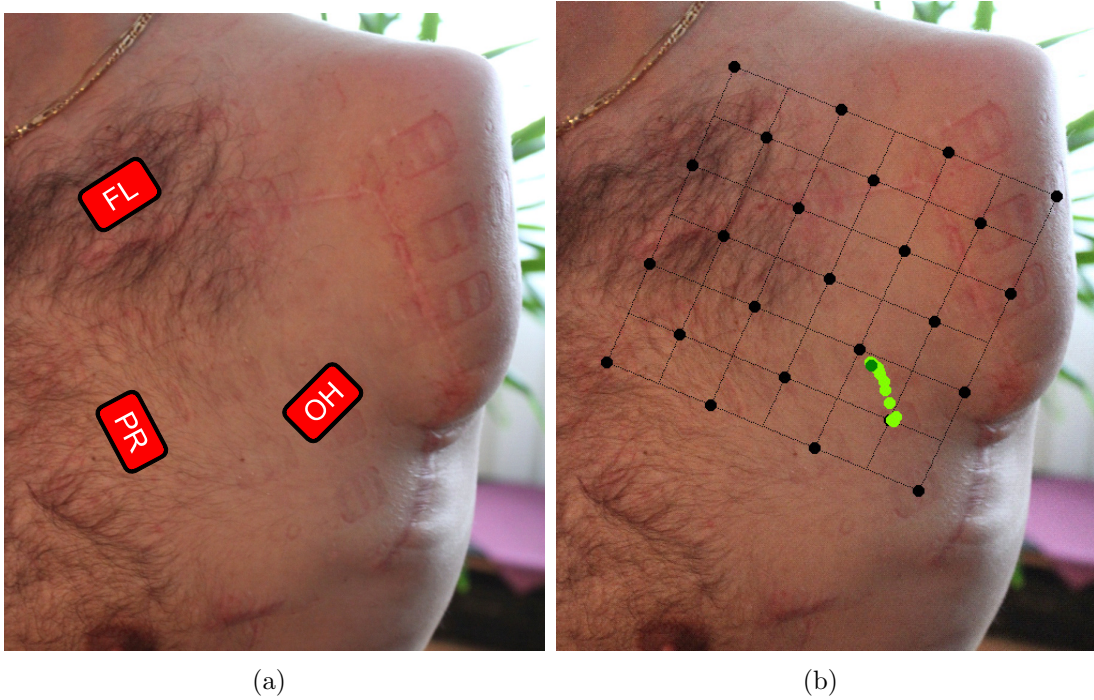


Figure 3.9: Evaluation of real data. (a) The positions of three electrodes on the patient’s chest, which denote the actual hotspot positions. (b) The result of the hotspot localization shown on an projection to the image of the patient. The bright green dots are the hotspot’s path and the dark green dot is the current hotspot

Figure 3.9 (b) shows the result of the hotspot localization, projected to an image of the patient. This outcome was achieved by performing a “hand close”-movement using a contraction strength of 100% of MVC. The bright green dots indicate the hotspot’s path while contraction and the dark green dot marks the current hotspot position at the time the screenshot was taken. As one can see the computed hotspot position coincides with the electrode position and therefore the computation is correct. Conclusively, in order to get a single dot as a recommen-

dition for the electrode position, the software analyzes the path and decides, due to a frequency distribution, which position to take as a prime selection.

CHAPTER 4

Conclusion

This section elucidates the basic findings of this thesis and it elaborates conclusions regarding goals of the thesis, which had been made in advance. Additionally the strengths and limitations of the current stage of development are mentioned, which leads to possible future research directions in order to further improve the procedure of finding hotspots. Furthermore the final outcome of this work and its contribution to an easier hotspot localization is mentioned as well.

4.1 Basic Findings

As already presented in the introduction, the main goal of this thesis was to find a solution for the software part for the localization of hotspots, in order to simplify the current procedure. Therefore different algorithms were developed and checked out. On the one hand bicubic interpolation and on the other hand template matching, which itself was achieved by applying normalized cross correlation or sum of differences respectively, were tested. For that, different datasets were used, which contained simulated data as well as real data, derived from patients. In addition to that the algorithm's behavior to faulty electrodes was tested, in order to check their performance to electrode lift-offs, which unfortunately occurred even with the latest electrode array build-up. In conclusion template matching using NCC showed the best results. On the one hand the average difference between the actual hotspot position and the one computed is way lower than the bicubic interpolation's average difference and outliers are less frequent. On the other hand within template matching NCC is a more rugged method than SAD, which faces tremendous outliers.

Another important part of the software was the visualization of the results. Therefore two different approaches were developed, which handle the distinct results derived from template matching (position of the hotspot) and bicubic interpolation (whole interpolated field) respectively. Since template matching was the prime selection, the visualization is restricted to displaying the hotspot and its path. To make it more vivid these two outcomes are projected onto an image of the patient.

In spite of actually not being part of this work a lot of different electrode array build-ups were build and tested. On the one hand these tests showed that the target of building an array suitable on a multitude of patients is not feasible

(or at least really hard to achieve) and on the other hand an increase in electrodes improves the localization result.

4.2 Strengths and Limitations

From the findings mentioned above the electrode array build-up constitutes the biggest limitation. The reason for that is the occurrence of electrode lift-offs, which distort the hotspot finding algorithm. Although the algorithm can deal with a small amount of faulty electrodes, the current build-up is too unstable and therefore produces too much lift-offs. In addition to that, one has to know which electrodes lifted off, in order to skip or estimate those faulty signals. Therefore future research should focus on the development of convenient hardware, which provides a proper input to the software, which accomplishes all requirements made in advance. If this is not possible a detection of faulty signals within the software would improve the issue as well.

The strengths of the software is the algorithm's robustness. Independently of the hotspot's size or shape (within a certain range, typical for sEMG signals) and rotational angle the hotspot can be located with an average accuracy of approximately 1mm. Therefore it can be applied to other parts of the body as well. For instance the algorithm was also applied to the upper arm of a patient having a transhumeral amputation. Needless to say, in this case an adaption of the electrode array has to be made. Furthermore, the visualization can be seen as a strengths, because it is easy to handle and intuitive. In addition to that, in case of an alteration of the electrode array, it can easily be adjusted to the new conditions.

4.3 Final Outcome

After all, a novel approach was developed, which makes it possible to localize hotspots on a patient's chest by means of software-assisted interpretation of sEMG signals. Even though the mentioned limitation concerning the electrode array build-up inhibits a final product, the tests of the algorithm showed the effectiveness and potential of this novel approach. Compared to the previous procedure, which used repositioning of electrodes by hand in order to find hotspots, it is more precise. This enables the locating of more intense signals, which contribute to a better control of the prosthesis. In addition to that, the new approach is much faster and hence more convenient for both, the patient and the therapist.

APPENDIX A

Software - Basic Structure And Important Methods

The whole software is divided into smaller parts, in order to get a clearer structure. In the beginning variables are defined, which are used in the whole program. Afterwards all events (e.g. button clicks, form load, timer ticks, ...) are covered and subsequently subregions are introduced, which contain variables and methods of smaller functional parts. In total ten regions were used:

- Graphical User Interface (GUI)
- Data Acquisition
- Signal Conditioning
- Bicubic Interpolation
- Template Matching
- Gaussian Distribution
- Real Time Visualization on Graphs
- Real Time Visualization of Hotspot
- Real Time Visualization of Hotspot on Image
- Other Methods

Additionally another region, called „Development“, contains all source code used in the course of this project in order to develop, test, and improve algorithms. On the graphical user interface (GUI) the development part is separated from the actual software by the usage of tab pages.

Important Variables

As previously mentioned, the most important variables are defined in the beginning of the source code. The array „scanData“ contains the data derived from the 32 channels. Its second dimension is set to 600000, which corresponds to 600000

samples. Since the sampling rate is 1kHz a maximum of 10 minutes can be acquired. If necessary the dimension of the array can be increased, in order to get more sampling time. The array „replayData“ takes recorded data from a CSV file and has the same purpose as „scanData“. Variable „path to folder“ sets the path to the folder containing all program data, like measurements or simulated data files. If recorded data is written to a CSV file, the variable „header“ contains all necessary information in order to generate the header for the file. On the other hand, if a CSV file is read, all header information from the file is written to „header“, which is consequently used in the source code. The current result derived from the hotspot localization methods are written to the variables „hotspot position BI“ (results of bicubic interpolation) and „hotspot position TM“(results of template matching). In addition to that „interpolated potential field“ represents the currently computed potential field and „hotspot path“ contains the hotspot path of the last 500 hotspots derived from template matching. The variable „faulty electrode mask“ denotes which electrodes are not working properly.

Events

This chapter describes all events, which can occur while execution of the software. They are in chronological order as they appear in the source code.

Form Load and Form Closing

These two events are used in order to initialize and clear the DAQ devices respectively. Since it is not possible to clear the device when it is still scanning, the current scan status is requested. If it is running a stop message is sent and a short time delay is performed as well, in order to prevent the devices of being cleared too early. After that the devices are cleared.

Button Clicks: Scan Start and Scan Stop

„Scan Start“ calls the method responsible for starting the scan and also sets a flag and enables/disables other GUI elements, which can not be pressed while scanning. In addition to that „Timer Write Scan To Graph“ is started in order to start the visualization. The „Stop Scan“ button event calls the method „DAQ Stop Scan“, stops the timer, also enables/disables GUI elements and additionally calls the method „generate header“, which generates the header needed to write the recorded data to a CSV file.

Button Clicks: Sequence Start and Sequence Stop

These button click events start and stop the activity sequences. These sequences are responsible to show the patient the movements he is supposed to perform. They always start with a „REST“ state, followed by the „FIRST ACTIVITY“ state, again a „REST“ state, and last the „SECOND ACTIVITY“ state. Depending on the settings made on the GUI this sequence is repeated (2 to 5 times) and the time of each state can be set to values from 2 to 5 seconds. „FIRST ACTIVITY“ state and „SECOND ACTIVITY“ state also depend on the user’s input. For example „FIRST ACTIVITY“ could be flexion and „SECOND ACTIVITY“ could be extension. Basically the two button events are responsible for setting and resetting some variables and flags and to enable/disable other GUI elements.

Button Click: Write CSV File

This button click writes the information derived from the electrodes while scanning and the header information to a CSV file. To keep the size of the file as small as possible the method „DAQ Get Raw ScanData“ is used in order to get the essential data (actually acquired samples) from the variable „scanData“.

Timer Tick: Write Scan To Graph

This timer plays an important part within the software. Its interval is set to 90ms, which is slightly smaller than the DAQ device’s callback rate, which is set to 100ms. So the first step within the timer tick is to check whether new scan information is available. If there is new information the visualization of it starts. If not, nothing happens and the timer tick will be called 90ms later. This step was necessary, because if the interval would have been set to 100ms, due to its inaccuracy the timer would have been too slow in order to follow the callback rate. Within the timer the output for the activity sequence labels is determined and the visualization of the hotspot (bicubic interpolation and template matching) and visualization of each channel is made.

Button Click: Read CSV File

This event reads the data from a CSV file (recorded channels as well as header information). The channel data is written to „replayData“ and the header information to „header“. In addition to that the information derived from the header is written to the corresponding labels on the GUI, other elements are enabled/disabled and some initializations are done.

Button Click: Load Image

Within this event the image of the patient, on which the hotspot position will be projected, is loaded. Since it is used within the XNA-Framework it was loaded to a two dimensional texture, which requires a file stream from which the image can be taken. Additionally all necessary information about the image (width and height) are used to determine coordinates for the positioning of the picture on the drawing panel. Afterwards method „RT Visualization of HS on Image Get CornerPoints“ is called, which asks the user to declare the corner points on the image.

Mouse Click: HotspotOnImage DrawingPanel

This event cooperates with the just mentioned method „RT Visualization of HS on Image Get CornerPoints“ by providing the coordinates of the mouse click at the panel. It works like a state machine, where the states are specified by the just mentioned method. Therefore depending on the „get corner points state“ the coordinates of the mouse click are written to the corresponding position of the array „corner points“. If the last state is reached the projected electrode grid is shown on the image.

Button Clicks: Simulation Start and Scan Stop

These two events basically have the same function as those described before (Button Clicks: Scan Start and Scan Stop), but they start another timer („Timer Write Simulation To Graph“ instead of „Timer Write Scan To Graph“).

Timer Tick: Write Simulation To Graph

Basically this event behaves in the same way the timer tick for scan data does (Timer Tick: Write Scan To Graph). In addition to that it has to be checked, whether the end of the data is reached. If this is the case several variables are reset and other GUI elements are enabled/disabled. The part within the main routine of the timer tick, which is responsible for drawing the hotspot position to the drawing panel, is extended by the projection of the the results to the image. This is not included in the real time scan visualization, because the image with the exact electrode positions can not be taken prior measurements.

Value Changed: Trackbar AdjustTime

The trackbar is used to fast-forward and rewind the replayed data in time. Therefore it works in the same way as the timer mentioned above does. Instead of counting a time variable it takes the value from the trackbar. The only difference

is the behavior if the end of data is reached and the inversion of the variable „previous mode“ in order to force the drawing buffer to be update as a whole, which is necessary if you move back in time.

Functional Parts (Regions)

Region: Graphical User Interface

This region is responsible for updating the GUI. Therefore all necessary properties of the elements (e.g. color, visibility, checked, ...) are represented using variables. Among others the most important methods within this region are „GUI Update GUI Input()“ and „GUI Update GUI Output“. The former writes all information currently present on the GUI to the corresponding variables, from which further executions are made, and in contrast the latter uses the variable's information in order to update all elements. Therefore they are always used together - before calculations are made „GUI Update GUI Input“ is called and afterwards the GUI is updated using „GUI Update GUI Output“.

Region: Data Acquisition

All methods within this region are responsible for the acquiring of the data from the two analog digital converters. Important variables are „devices ready“, which provides information whether the devices are ready and can be used, and the two „block count“ variables (one for each device), which indicate the actual amount of blocks written to „scanData“. The first important method is called „DAQ Initialize Devices“. It initializes both ADC, by declaring „device 1“ and „device 2“ by its serial numbers. This is important in order to assign the 32 channels. „Device 1“ represents channel 1 to 16 and „device 2“ represents channel 17 to 32. Afterwards the calibration on both devices starts, which takes a few seconds. While calibrating a dialog shows up which uses a progressbar and a timer to illustrate the progress. If calibration is done, both devices are set to single ended mode. Method „DAQ Start Scan“ enables all callback methods, configures the trigger signal and sets all scan properties on the ADCs. After that the message to start the scan is sent. In contrast „DAQ Stop Scan“ sends a stop message and disables all callbacks. In common there are six callback methods:

- DAQ ReadScanData Dev1
- DAQ ReadScanData Dev2
- DAQ ScanComplete Dev1
- DAQ ScanComplete Dev2

- DAQ ScanError Dev1
- DAQ ScanError Dev2

The former two are the most important, because they take the acquired data from the devices and store it to „scanData“. Since the devices are operated in block mode (block size is set to 100 samples) these callbacks are called when ever 100 new samples are available. Since the sampling rate is 1kHz the callbacks are called every 100ms. Every time they are called a count variable is increased, in order to know how many blocks already have been acquired. This important for the timer, which is responsible for the visualization.

Region: Signal Conditioning

This region contains all variables representing means ,e.g. overall means (means of each channel of whole data set) and current means (means of each channel of the current block) and a 7×7 array containing the current conditioned data at a certain point of time. The most important methods are „SigCon Scale Value“, „SigCon Offset Comp“, and „SigCon Offset Comp And RMS“, which perform exactly what one can expect from their name.

Region: Bicubic Interpolation

This region contains all methods necessary to perform a bicubic interpolation. The most important are „BicubicInt Interpolation“, „BicubicInt RealTime“, and „BicubicInt Get Hotspot“. The former performs a bicubic interpolation to an array (arbitrary in size) with respect to the desired resolution. The second one is used for the interpolation of the 25 electrode grid in real time. Therefore the resolution and the size of the input array is determined a priori (size is 7×7 and the resolution is 25). In addition to that 24 nodes of the 7×7 array have to be estimated (by bicubic interpolation and nearest neighbor method respectively) in order to get a regular grid. The last method determines the position of the hotspot, by searching the interpolated field for its maximum.

Region: Template Matching

This region consists of only one method called „TemplateMatching Get Hotspot“. It is a downgraded version (because the original version’s computational cost is too high) of the cross correlation method used in the development region. Therefore it uses the rough and precise scan procedure, which initially performs a rough scan to the field and after that refines the scan in the region of the hotspot.

Region: Gaussian Distribution

The only method within this region is called „GaussDistribution at specific electrode“. It is needed in template matching to generate the template value at a certain position. This value depends on a lot of parameters of the Gaussian distribution, which are also input to this method:

- Position of the center of the peak (in X and Y)
- σ_x and σ_y
- Rotational Angle

Region: Realtime Visualization on Graphs

In the beginning of this region a lot of variables are defined. The most important are „rtgraph drawing buffer“, „rtgraph raw data buffer“, and „rtgraph cond data buffer“. The former is the actual buffer which is used to drawn on the drawing panel. Its first dimension is 32 in size and it represents the channels. The second dimension is 500 in size which corresponds to the amount of pixels the graph occupies in y direction. The third dimension is of size 2, because when using raw data every pixel in y needs two values. The other two buffers contain the actual values for each channel and each pixel, which have to be plotted. Depending on the visualization settings (show raw or conditioned data) derived from the GUI, whether „rtgraph raw data buffer“ or „rtgraph cond data buffer“ is converted to „rtgraph drawing buffer“. Additionally „lineListIndices Channel raw“, „lineListIndices Channel cond“, and „primitiveList Channel“ play an important part. They are used within the XNA framework for drawing lines. Furthermore offset, frame, axis, and feeder variables are defined in subregions. The most important methods are:

- „RTGraph Draw Frame“
- „RTGraph Draw Axis“
- „RTGraph Draw Feeder“
- „RTGraph Draw Channels“
- „RTGraph Update Buffer“

The former three are quite trivial. „RTGraph Draw Frame“ and „RTGraph Draw Axis“ use predefined variables in order to draw the desired lines for the frame and the axis, whereby the visualization of the axis depends on the amount of channels selected on the GUI. „RTGraph Draw Feeder“ draws a green line at the current feeder position. „RTGraph Draw Channels“ is responsible for the actual

visualization of all selected channels. Therefore the whole „rtgraph drawing buffer“ is plotted. If some electrodes are denoted as faulty they will be plotted in red, if they are working properly they will be plotted in blue. The most important method is „RTGraph Update Buffer“. Since the data is stored in different variables, first of all a distinction between scan and replay mode has to be made. After that, the current feeder position is determined. This is important in order to know the position to which new values have to be written within the buffer. Depending on the signal representation mode (left to right or continuous) selected, the buffer has to be shifted by a certain amount of values (depending on the time axis), or not. If the user changes the time axis setting, the whole buffer has to be updated. The last step within this method is the transformation of the data buffers („rtgraph raw data buffer“ or „rtgraph cond data buffer“) to the actual drawing buffer. Therefore the data is converted to values between 0 and „pixels per channel“, where „pixels per channel“ contains information about the pixels a channel can occupy in x direction (which depends on the amount of channels selected).

Region: Realtime Visualization of Hotspot

Within this region the XNA framework is used in order to draw the hotspot position derived from template matching and the interpolated potential field derived from bicubic interpolation respectively. Therefore in the beginning some XNA classes are defined and the method „RT Visualization of HS Initialize XNA“ performs all necessary initialization steps. The method „RT Visualization of HS TemplateMatching“ draws the hotspot position within the electrode grid. Therefore a frame, each electrode (black circles) and the grid is drawn. In addition to that, faulty electrodes are denoted (small red circles). Furthermore, the current hotspot and the path of former hotspots is drawn (circles) using following colors for the different movements:

- No Move ... Blue
- Flexion ... Yellow
- Extension ... Green
- Hand Close ... Gray
- Hand Open ... Magenta
- Pronation ... Purple
- Supination ... Red

Method „RT Visualization of HS BicubicInterpolation“ works in the same way the previously explained method works, but instead of drawing the hotspot position as a circle the whole interpolated potential field is drawn on a heat map. Therefore the data values on method’s input are converted to RGB values.

Region: Realtime Visualization of Hotspot on Image

This region basically has the same purpose as the region mentioned above. But in addition to that the coordinates of the electrodes, the grid, the hotspot position, and its path have to be converted to the new projected coordinates on the image. Therefore vector algebra is used. „RT Visualization of HS on Image Get CornerPoints“ represents the starting point of the state machine responsible for the determination of the four corner points (e.g. it sets „get corner points state“ to 1). Further manipulation of this variable is performed within the responsible mouse click event (Mouse Click: HotspotOnImage DrawingPanel). The previously mentioned vector algebra is done within method „RT Visualization of HS on Image Get New Coordinates“, which takes the old coordinates from its input and converts them to the new ones. The drawing itself works in the same way it is described in the previous chapter.

Region: Other Methods

This region contains methods which can not be associated with any other region. Method „update mask for simulated faulty electrodes“ and „enhance electrode values“ can be linked with the handling of faulty electrodes. The former is called if the current settings (made from the user on the GUI) have to be updated. The last estimates the missing (faulty) electrode values by taking its neighboring electrodes into account. This is necessary because tests showed, using estimated values for faulty electrodes eventuates in a better result than ignoring faulty electrodes. Another method within this region is called „generate header“. It uses all necessary data from the GUI and from the recorded data itself in order to generate the header of the CSV file. The header is composed of 13 ushort values and it is structured like this:

- Position 1: Patient ID: ranges from 00001-65535
 - range 00001 - 20000 for simulated data
 - range 20001 - 65535 for patients
- Position 2: TMR Region: ranges from 00001-00002
 - 00001 ... chest
 - 00002 ... upper arm
- Position 3: Contraction: ranges from 00001 - 00100 [in % of maximum contraction]
- Position 4: Electrode Arrangement: ranges from 10000 - 10165 [in ° counter-clockwise], where 0° is horizontal

- Position 5+6: Date:
 - Position 5 ranges from 00101-03112 and represents the day and month
 - Position 6 ranges from 00000-65535 and represents the year
- Position 7+8: Samples:
 - Position 7 ranges from 00000-65535 and represents digit 5-9
 - Position 8 ranges from 00000-09999 and represents digit 1-4
- Position 9: Activity-Sequence: ranges from 00001-00007
 - 00001 ... elbow
 - 00002 ... hand
 - 00003 ... wrist
 - 00004 ... elbow + hand
 - 00005 ... elbow + wrist
 - 00006 ... hand + wrist
 - 00007 ... elbow + hand + wrist
- Position 10+11: Start at Sample: indicates the start of the activity sequence
 - Position 10 ranges from 00000-65535 and represents digit 5-9
 - Position 11 ranges from 00000-09999 and represents digit 1-4
- Position 12: Time per Activity: ranges from 00002-00005 [in seconds]
- Position 13: Repetitions: ranges from 00002-00005 [in amount of repetitions]

The last three methods within this region deal with the path of the hotspot. The most important is „update hotspot path“, which adds the computed hotspot, while a movement is made, to the array containing the path. On the one hand this array contains the coordinates of each hotspot and in addition to that information about the movement, which is important for the visualization to assign colors to the different movements.

Region: Development

This region contains all buttons and methods used while developing and analyzing the different algorithms (interpolation, sum of differences, and correlation coefficients). To separate the user part from the development part two different tab pages are used. The development part contains four buttons - „Interpolate“, „OpenMatrixCSV“, „SumOfDifferences“ and „Correlation“. The click event on „OpenMatrixCSV“ opens a CSV file, picks the 25 electrode values from this data, searches the data for its maximum (to find out the actual hotspot position) and

plots the data to the panel. „Button Interpolate Click“ performs a bicubic interpolation to the desired data (values from GUI or electrode values from file). For the visualization of the interpolated data a zoom factor and the resolution is taken into account, before it is plotted. „Button SumOfDifferences Click“ and „Button Correlation Click“ perform template matching to the data by using the sum of differences and correlation coefficients respectively. By checking radiobuttons one can choose between a slow and a fast approach. For visualization again a XNA framework is used.

LIST OF FIGURES

1.1	Path of a motor command	8
1.2	TMR - operative procedure	10
1.3	Muscle structure	11
1.4	Signal transmission - nerve to muscle	12
1.5	Parts of a sarcomere	13
1.6	Sliding filament theory	14
1.7	Simulated three dimensional potential field distribution	15
2.1	Burst mode	18
2.2	MyoBock raw signal electrodes	19
2.3	Electrode arrangements	19
2.4	First approach of the electrode array build-up	20
2.5	Second approach of the electrode array build-up	21
2.6	Third approach of the electrode array build-up	22
2.7	Fourth approach of the electrode array build-up	23
2.8	Scan procedure	25
2.9	Signal conditioning	27
2.10	Generative model for sEMG	28
2.11	Three dimensional generic EMG model using FEA	29
2.12	Basics of bicubic interpolation	30
2.13	Usage of unit squares for interpolation	34
2.14	Bicubic interpolation - implementation to the source code	35
2.15	Generated templates	36
2.16	Template matching - scan procedure	37
2.17	Positions of the electrode values (E01 - E25) within matrix E	38
2.18	Template matching - implementation to the source code	40
2.19	Compression of the raw signal	42
2.20	Color coding	43
2.21	Visualization - implementation to the source code	44
2.22	Projection onto an image	45
2.23	Visualization results	46
2.24	Software overview	47

3.1	Electrode arrangements used for testing bicubic interpolation . . .	49
3.2	Datasets used for testing bicubic interpolation	50
3.3	Localization problem	53
3.4	Datasets used for testing template matching	54
3.5	Electrode labeling and hotspot positions	56
3.6	Sequence 1	58
3.7	Sequence 2	59
3.8	Movement sequence	61
3.9	Evaluation of real data	62

LIST OF TABLES

2.1	Parameters of the three example templates	37
3.1	Datasets used for testing bicubic interpolation	51
3.2	Results of testing bicubic interpolation	52
3.3	Results of testing template matching, using constant hotspot sizes	54
3.4	Results of testing template matching, using varying hotspot sizes .	55
3.5	Results of the comparison of bicubic interpolation and template matching	56
3.6	Results of testing bicubic interpolation and template matching at the presence of faulty electrodes	57

REFERENCES

- [1] NLLIC. Amputation statistics by cause - limb loss in the united states, 2008.
- [2] Statistic-Brain. Amputee statistics. <http://www.statisticbrain.com/amputee-statistics/>, Accessed 04.03.2012.
- [3] Roberto Merletti and Philip A. Parker. *Electromyography - Physiology, Engineering, and Noninvasive Applications*. IEEE Press, 2004.
- [4] Larry R. Squire, Darwin Berg, and Floyd E. Bloom. *Fundamental Neuroscience*. Academic Press (Elsevier), 3 edition, 2008.
- [5] Rudolf Nieuwenhuys, Jan Voogd, and Christiaan van Huijzen. *The Human Central Nervous System*. Springer, 4 edition, 2007.
- [6] Laura A. Miller, Kathy A. Stubblefield, and Robert D. Lipschutz. Improved myoelectric prosthesis control using targeted reinnervation surgery: A case series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2008.
- [7] Todd A. Kuiken, G.A. Dumanian, and Robert D. Lipschutz. The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticulation amputee. *Prosthetics and Orthotics International*, 2004.
- [8] Todd A. Kuiken, Guanglin Li, and Blair A. Locks. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *JAMA - The Journal of the American Medical Association*, 2009.
- [9] D.S. Childress, R.F. Weir, and D.J. Smith. Control of limb prostheses. *Atlas of Amputations and Limb Deficiencies*, 2004.
- [10] T.W. Williams. Control of powered upper extremity prostheses. *Functional Restoration of Adults and Children with Upper Extremity Amputation*, 2004.
- [11] J.A. Hoffer and G.E. Loeb. Implantable electrical and mechanical interfaces with nerve and muscle. *Ann Biomed Eng*, 1980.
- [12] Todd A. Kuiken, D.S. Childress, and W.Z. Rymer. The hyper-reinnervation of rat skeletal muscle. *Brain Res*, 1995.
- [13] Todd A. Kuiken, Laura A. Miller, and Robert D. Lipschutz. Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study. *The Lancet*, 2007.
- [14] Paul Blakey. *Muscle Book*. Himalayan Institute Press, 2008.

- [15] Brian R. MacIntosh, Philip F. Gardiner, and Alan J. McComas. *Skeletal Muscle - Form and Function*. Human Kinetics, 2 edition, 2005.
- [16] Joachim Herz Stiftung. Muscle structure. http://www.leifiphysik.de/web_ph07_g8/umwelt_technik/09muskelkraft/muskelkraft.htm, Accessed 07.03.2012.
- [17] Mary K. Floeter. Structure and function of muscle fibers and motor units. *Cambridge University Press*, 2010.
- [18] F. Buchtal and Poul Rosenfalck. On the structure of motor units. *New developments in electromyography and clinical neurophysiology*, 1973.
- [19] R.F. Schmidt and G. Thews. *Physiologie des Menschen*. Springer, 1990.
- [20] H.E. Huxley. Past, present and future experiments on muscle. *Philosophical Transactions of the Royal Society B*, 2000.
- [21] K. Maruyama. Birth of the sliding filament concept in muscle contraction. *J Biochem*, 1995.
- [22] Hank van Helvete. Molecular mechanism of muscle contraction. <http://commons.wikimedia.org/wiki/File:Muskel-molekular.png>, Accessed 07.03.2012.
- [23] John V. Basmajian and Carlo J. DeLuca. *Muscle Alive - Their functions revealed by electromyography*. William and Wilkins, 1985.
- [24] Poul Rosenfalck. Intra- and extracellular potential fields of active nerve and muscle fibers. *Akademisk Forlag*, 1969.
- [25] Roberto Merletti and Carlo J. DeLuca. New techniques in surface electromyography. *Computer-aided electromyography and expert systems*, 1989.
- [26] H.J. Hermens and B. Freriks. The state of the art on sensors and sensor placement procedures for electromyography: a proposal for sensor placement procedures. *Roessingh Research and Development*, 1997.
- [27] Measurement Computing. *USB-1608G Series - User's Guide*, 2 edition, April 2012.
- [28] Otto Bock Healthcare GmbH. 13e200 myobock electrode. http://www.ottobock.com/cps/rde/xchg/ob_us_en/hs.xsl/16573.html?id=16619#t16619, Accessed 23.04.2013.
- [29] Jacques Duchêne and Francis Goubel. Surface electromyogram during voluntary contraction: Processing tools and relation to physiological events. *Critical Reviews in Biomedical Engineering*, 21, 1993.

- [30] Uwe Windhorst. *Modern Techniques in Neuroscience Research*. Springer, 1 edition, 1999.
- [31] Peter Konrad. *Emg-fibel*, 2005.
- [32] Ning Jiang, Kevin B. Englehart, and Philip A. Parker. Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal. *IEEE Transactions on Biomedical Engineering*, 56(4):1070–1080, April 2009.
- [33] Jakob Honeder. A quasi-stationary approach to the approximate solution of a fea 3d subject-specific emg model. Master’s thesis, University of Applied Sciences Technikum Wien, 2013.
- [34] William H. Press, Brian P. Flannery, and Saul A. Teukolsky. *NUMERICAL RECIPES in Fortran 77 - The Art of Scientific Computing*. Cambridge University Press, 2 edition, 1992.
- [35] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 1 edition, 2009.
- [36] Wikipedia. Gaussian function. http://en.wikipedia.org/wiki/Gaussian_function, Accessed 23.03.2013.
- [37] J.P. Lewis. Fast normalized cross-correlation. *Vision Interface*, 1995.
- [38] D.U. Silverthorn. *Human Physiology*. Pearson Education, 5 edition, 2009.