

title

# Titel

Subtitle

Simon Heinke  
Lars Debor

Julius Lerm  
Petros Simidyan

Felix Griesau  
Blerta Hamzallari

Marco Klamke  
Sugandha Sachdeva

last Change: April 26, 2017

# 1 Introduction

Medicine today is highly advanced and is able to treat an immense amount of diseases and disorders. This results in high standards and a lot of pressure on people working in this branch. In order to further improve the standards this amount of errors has to be minimized, since they can have fatal consequences.

In pursuance of achieving, maintaining as well as improving these abilities, technological assistance is of utmost importance.

Since the brain is one of the main organs, caution and accuracy is essential while treating its conditions.

Techniques such as the MEG (Magnetoencephalography) or EEG (Electroencephalography) observe the brains activity by measuring and monitoring magnetic fields or electrical deviations.

These procedures help diagnosing epilepsy, brain death and other severe brain diseases.

Furthermore they are used for research in fields like psychology. A proper visualization aids the usability of the generated data.

The MNE-CPP project builds tools for the purpose of making analyzing EEG and MEG data easier. Thus the whole project is open-source and everyone can contribute. As language only C++ is used, although the project simultaneously exists in other languages, e.g. Python. MNE-Scan, -Analyze and -Browse are some of the features of the existing framework.

The new extension of the current project and focuses on real-time 3D-visualization of EEG/MEG sensor data, while being as exact and fast as possible. It is integrated into the existing code and accessible through the MNE-Client as well as compatible with operating systems Linux and Windows.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>3</b>
2.1	Mandatory criteria . . . . .	3
2.1.1	Surface Constrained Distance Calculation (Geodesic problem on meshes)	3
2.1.2	Point to plane mapping . . . . .	3
2.1.3	Interpolation algorithm . . . . .	4
2.1.4	Integration in Disp3D . . . . .	4
2.1.5	Non-Functional Requirements . . . . .	4
2.2	Optional criteria . . . . .	5
2.2.1	SCDC (2.1.1) . . . . .	5
2.2.2	Interpolation (2.1.3) . . . . .	5
2.3	Delimiting criteria . . . . .	5
<b>3</b>	<b>Product environment</b>	<b>6</b>
3.1	Scenario . . . . .	6
3.2	Software . . . . .	6
3.2.1	The MNE-CPP framework . . . . .	6
3.2.2	. . . . .	6
3.3	Hardware . . . . .	6
<b>4</b>	<b>Product Functions</b>	<b>7</b>

## 2 Requirements

The product receives EEG/MEG sensor data and constructs a real-time 3D visualization of the brains current activity. Users can choose between further options, changing the output immediately to their personal preferences.

### 2.1 Mandatory criteria

The following functions have to be implemented correctly and must fulfill given requirements.

#### 2.1.1 Surface Constrained Distance Calculation (Geodesic problem on meshes)

**Note:** From this point on **Surface Constrained Distance Calculation** will be referred as **SCDC**.

Because the brain has an uneven surface, a function for calculating the distance between two separate points is needed.

As the euclidian distance would not respect the structure of the surface, a different approach for determining the exact distances has to be implemented. The function receives input data in form of a preprocessed triangulated surface mesh and calculates the distance between the vertices.

**C111** Based on that data, the function calculates a matrix that holds values describing the distances between all vertices using double precision.

**C112** The function must be able to process up to 200,000 vertices.

**C113** The user can limit the calculation to a subset of vertices.

#### 2.1.2 Point to plane mapping

Since the sensors do not directly touch the head, therefore float slightly above it, an accurate projection is needed to exactly localize their positions regarding the brain.

Hence a function must be implemented to solve this problem. The function receives a set of sensor locations in 3D-Space and maps them onto the underlying surface mesh. Thus every sensor gets assigned to a vertex of the mesh.

**C121** The function must be able to handle data from MEG-sensors which have a known orientation.

**C122** The function must be able to handle data from EEG-sensors which are non-orientated.

### 2.1.3 Interpolation algorithm

The core feature is the ongoing interpolation, visualizing a particular set of sensor data, representing the brains activity.

Because the number of vertices is bigger than the quantity of sensor points, most vertex-values must be interpolated. Thus the algorithm receives a mesh and a subset of vertices with their respective sensor data.

**C131** Based on the said subset the algorithm must calculate the values for every vertex of the mesh.

**C132** For this, the algorithm creates a matrix storing weights for the later interpolation. The interpolation process can be summarized by the following equation:

$y_{full} = W \cdot y_{sub}$  , where  $W$  is the mentioned matrix and  $y_{sub}$  is the current dataset for the known sensors, i.e. vertices.

**C133** The calculation of the weight matrix must be based on the result of the SCDC (2.1.1).

**C134** Bad channels, a part of the given sensor data, must be considered during processing.

### 2.1.4 Integration in Disp3D

In order to ensure usability within the given framework MNE-CPP, the final visualization must be integrated into the preexisting GUI, namely Disp3D.

**C141** A new function must be added to the Disp3D tree model. Internally this function must create a new handler.

### 2.1.5 Non-Functional Requirements

**C151** The software has to run on the latest versions of 2 operating systems, namely Windows and Linux.

**C152** Features 2.1.1 and 2.1.2 are implemented in the class *GeometryInfo*, while 2.1.3 is facilitated in the class *Interpolation*.

**C153** The function to integrate the product into Disp3D (2.1.4) is named *addSensorData()*.

## 2.2 Optional criteria

As long as the mandatory requirements are fulfilled, it is desirable to further match the following criteria.

### 2.2.1 SCDC ([2.1.1](#))

**C211** The computation time should not exceed 1 second.

### 2.2.2 Interpolation ([2.1.3](#))

**C221** One interpolation cycle should take less than 17ms.

**C222** Multiple methods for calculating the weight matrix can be implemented. The user can select one.

**C223** The Computation is executed on GPU-level.

## 2.3 Delimiting criteria

The following criteria limit the functionality of the system.

**C311** The program receives preprocessed data and does not get in touch with hardware sensors.

**C312** The program does not evaluate the data medically and solely processes the data for further visualization.

## 3 Product environment

### 3.1 Scenario

As mentioned before the aim of the product is to visualize MEG and EEG data live. The signal derives from actual activity in the brain which the software should reconstruct.

### 3.2 Software

#### 3.2.1 The MNE-CPP framework

The MNE-CPP framework provides several applications dividable by front-end and back-end software. MNE Scan, MNE Browse and MNE Analyze are front-end applications which perform well in terms of usability. Beneath the front-end is the library layer which contains the core libraries provided by the framework.

The application must be integrated to the library layer.

#### 3.2.2

### 3.3 Hardware

## 4 Product Functions