

# 1 Introduction

# 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 . . . . .	3
2.1.4	Integration in Disp3D . . . . .	4
2.2	Optional criteria . . . . .	4
2.2.1	SCDC . . . . .	4
2.2.2	Interpolation . . . . .	4
2.3	Differentiating criteria . . . . .	4

## 2 Requirements

The product receives EEG/MEG sensor data and constructs a real-time 3D visualization of the brains current activity.

### 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)

The function receives input data, preprocessed by the software environment, structured as a triangulated surface mesh.

**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

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.

**C21** The function must be able to handle data from MEG-sensors which have a known orientation.

**C22** The function must be able to handle data from EEG-sensors which are non-orientated.

#### 2.1.3 Interpolation algorithm

The algorithm receives a mesh and a subset of vertices

**C31** Based on the said subset the algorithm must calculate the values for every vertex of the mesh.

**C32** 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.

**C33** The calculation of the weight matrix must be based on the result of the SCDC.

### 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.

**C41** A new function must be added to the Disp3D tree model. Internally this function must create a new handler.

## 2.2 Optional criteria

### 2.2.1 SCDC

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

### 2.2.2 Interpolation

**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.

## 2.3 Differentiating criteria

**C311** The program receives preprocessed data and does get in touch with hardware sensors.

**C312** The program does not