Alma Mater Studiorum - University of Bologna

ARCES - Advanced Research Center on Electronic Systems
for Information and Communication Technologies E.De Castro

Dottorato di Ricerca in
Ingegneria Elettronica, delle Telecomunicazioni e Tecnologie dell'Informazione

XXVIII CICLO
Settore Concorsuale di afferenza: 09/E3
Settore Scientifico disciplinare ING-INF /01

Algorithms and Methods for Imaging of Brain Activity from Non-Invasive Techniques

Presentata da: **Silvio Placati**

Coordinatore Dottorato: Relatore:

Prof. Alessandro Vanelli Coralli **Prof.ssa Eleonora Franchi Scarselli**

Esame finale anno: 2017

Abstract

The imaging of brain activity, also called “Functional Neuroimaging”, is used to understand the relationship between activity in certain brain areas and specific functions. These techniques include fMRI (functional Magnetic Resonance Imaging), PET (Positron Emittance Tomography), EIT (Electrical Impedance Tomography), EEG (ElectroEncephaloGraphy) and DOT (Diffuse Optical Tomography) and are widely used in the study of brain activity. Classical solutions such as fMRI and PET are characterized by high spatial resolution to the detriment of portability, cost, and temporal resolution, limiting their employment in a clinical environment.

In addition to clinical usage, analysis of brain activity is gaining popularity in others recent fields, i.e. Brain Computer Interfaces (BCI) and the study of cognitive processes. In these contexts, usage of classical solutions could be unfeasible, due to their low temporal resolution, high cost and limited portability. For these reasons, portable low cost techniques are objects of the proposed thesis’s research, with focus on DOT and EEG.

In particular, the research activity about DOT has been carried out in the context of the European project named “HIGH PROFILE” (HIGH-throughput PROduction of FunctIonaL 3D imagEs of the brain) while EEG research activity was developed in the scope of the European project named “CREAM” (CReativity Enhancement through Advanced brain Mapping and stimulation).

The main contribution of this thesis focuses on the implementation of a numerical solver for DOT based on the radiosity-diffusion model, integrating the anatomical information provided by a structural MRI.

DOT is an imaging technique based on evaluating how light propagates within the human head to obtain functional information about the brain.

Precision in reconstructing such an optical properties map is highly affected by the accuracy of the light propagation model implemented, which needs to take into account the presence of clear and scattering tissues.
The implemented solver is designed to run on parallel heterogeneous platforms based on multiple GPUs and CPUs and it integrates a parallel visualization toolbox based on Nvidia OptiX to obtain a 3D interactive rendering of the light distribution displayed on the human head. We demonstrate how the implemented solver provides meaningful speed-ups over traditional DOT solvers, along with a significant improvement in accuracy.
In particular, we obtained a 7x speed-up over an single run of isotropic-scattered parallel Monte Carlo engine for a domain of 2 million voxels, with an accuracy comparable to 10 runs of anisotropic scattered Monte Carlo in the same geometry. The speed-up significantly increases for larger domains, allowing one to compute the light distribution of a full human head (≈ 3 million voxels) in 116 seconds for the platform used.
The implemented DOT solver was also employed to validate experimental measurements made on a phantom mimicking the optical properties of a realistic human head, identifying an issue in the experimental setup leading the optical sensors operating in a non-linear range which represents a point for further work once a complete DOT system is available.

The secondary contribution of this thesis focuses on EEG and it concerns the implementation of software libraries for time-domain source localization in the scope of an open-source framework called *Creamino* which can be used to simplify and speed-up the design of BCI systems. It consists of firmware and software libraries that allow designers to connect new EEG platforms to software tools for BCI. Specifically, *Creamino* has been developed starting from the open-source electronic platform Arduino, and it can process multiple EEG channels on-line and operates under Windows, Linux and Mac OS X in real-time on a standard PC. Schematics, gerber files, source code and software modules of *Creamino* are available with full documentation and free of charge for research and educational purposes online at https://github.com/mchiesi/Creamino.
Since certain experiments might need to analyze EEG sources localization in the frequency domain rather than in time-domain, we propose also the

implementation of frequency EEG Source Localization algorithms.
Finally, in order to simplify the interpretation by the user of EEG source localization results, a 3D visualization toolbox able to manage the complexity of the geometries directly imported by a structural MRI and allowing the users to render the results of the DOT forward problem or EEG source localization and interact with the rendered surface.

Contents

List of Figures

List of Tables

Introduction

The Imaging of Brain Activity, also known as “Functional Neuroimaging”, is used to understand the relationship between activity in certain brain areas and specific functions and it represents a powerful tool in diagnosis and treatment of central nervous system diseases and general study of brain behaviour.

In this context, it is important to highlight that while classical solutions such as fMRI and PET are reserved to clinical environment due to their high cost and large size, alternative low cost techniques such as DOT, EEG and EIT are gaining interest.

Between the low cost techniques listed above, EEG is the well-known and widespread and it is widely used also in clinical contexts, while DOT and EIT are subject to research yet.

Combining the advanced state of development and the simple recording hardware which characterize EEG technique, it leads to an emergent interest for extra-clinical applications such as Brain Computer Interfaces (BCIs).

BCIs are communication systems interfacing human brain with external devices, e.g. computers or actuators, where user’s commands are evaluated starting from the analysis of the EEG of the subject. The study of cognitive processes is instead based on applying external stimuli to the patients, which could be of visual, auditory or somatosensory nature, and analyze the corresponding brain activity response.

In this scenario, my research activity converges into two different contributions: a main contribution about DOT, which was developed in the scope of the “HIGH PROFILE” European project, and a secondary contribution about EEG for BCI systems, which was developed in the scope of the “CREAM” European project.

DOT contribution is described in Section A of this thesis while EEG contribution is analysed in Section B.

As shown previously, DOT has the advantages of low cost, high portability and good temporal resolution; on the other hand, spatial resolution is rather limited, in particular due to the presence of various layers of tissue through which light needs to propagate before reaching the cortical brain regions [2]. Moreover, light propagation is heavily influenced by the complex shapes which characterize the interfaces between tissues; hence, the availability of anatomical information deriving from an MRI scan of the subject can lead to more accurate domain-specific solvers [3], improving the spatial resolution and accuracy of the imaging technique.
DOT image reconstruction requires an optimal combination of a forward and an inverse problem solver. The purpose of the forward problem is to compute the light distribution inside the volume under consideration given complete information as to the light sources and the optical properties within the domain, from which a set of light exitance values can be determined (i.e. the light irradiated through the surface underneath a detector). By contrast, the inverse problem solution provides an estimated reconstruction of the local optical properties of the volume under examination given the configuration of light sources and the set of experimental measurements. In general, an inverse problem solution requires multiple computations of the forward problem solution, calling for accurate and time-economical forward solvers. Several methods have been developed in order to provide an appropriate trade-off between these two requirements, ranging from accurate, highly intensive transport-based approaches [4] to diffusive approximations [5], through hybrid radiosity-diffusion methods [6].

Transport-based approaches use the Radiative Transport Equation (RTE) and provide an accurate photon density distribution within the domain. Unfortunately, solving the RTE in non-homogeneous volumes having complex shapes as required in optical tomography is an extremely computationally-intensive task if solved numerically or statistically (e.g. through Monte Carlo methods)[7][8]. Diffusion equation solvers offer lower computational times than RTE solvers, at the cost of reduced modeling accuracy.
In general, a diffusion equation can be derived as a first order spherical approximation of the radiative transfer equation and is suitable to model

light propagation through highly-scattering media, where light propagates diffusively [9]. The chief disadvantage in using a diffusion equation in a realistic human head domain is that, in a low-scattering medium such as the Cerebro-Spinal Fluid (CSF) which fills the central ventricles and the subarachnoid space, it provides inaccurate results.

Hybrid radiosity-diffusion models combine the results of a diffusion equation in highly-scattering regions with radiosity solution, which specifically models light propagation through clear regions following the principle of ray optics. This approach, if compared to the simple diffusion model, drastically improves the accuracy of the light distribution in scattering domains with clear regions and, at the same time, allows photon distribution to be calculated efficiently to an accuracy comparable with that of RTE Monte Carlo methods [10][11].

To our knowledge, radiosity-diffusion (RD) models are only available in 3D regular geometries (i.e. spheres or overlapped layers), due to the absence of radiosity algorithms able to handle the complex shape of the CSF region [12]. In RTE Monte Carlo methods, a common approach to tackling the increased level of complexity is to limit the number of simulated photons as much as possible, leading to reduction of the light penetration depth [13][14]. However, due to the highly irregular shape of the CSF region and its low-scattering and low-absorption properties, a significant amount of light that reaches its surface is irradiated towards the cortical regions with a wide light-emittance angle, spreading light into regions which may be quite far from the source position. There is therefore a need for models able to include domains which are significantly larger than the region immediately underneath the sources and detectors. Furthermore, as stated in [15], the high degree of scattering in optical imaging makes it unrealistic to assume flat photon propagation, making 3D models necessary in order to consider the third direction of light propagation.

In the Section A of this thesis, we present an innovative numerical solver for evaluating light distribution in the human head supporting both Continuous Wave (CW) and Frequency Domain (FD) DOT, which is able to handle realistic geometries by directly importing structural information about a full human head derived from MRI scans with voxel sizes of approximately 1 mm^3 (total voxel count of approximately 5 million). To our knowledge, this is the first solver able to accurately model hybrid radiosity-diffusion light propagation inside a complete human head to this degree of spatial accuracy. The very high computational load that derives from implementation of a full 3D model with a priori morphological information is tackled by exploiting a high degree of parallelization to compute the results for both the diffusive and the radiosity portion of the problem. In particular, a custom parallel ray tracer based on NVIDIA OptiX 3.8.0 is implemented to manage the whole CSF region efficiently. In order to provide the necessary computational performance while keeping the cost and size of the platform to affordable levels, the solvers are designed to run on multi-CPU/GPU systems based on latest-generation NVIDIA graphic cards [16]. The novelty of the proposed model lies in its implementation of a radiosity-algorithm which supports the complex shape and the outstanding dimension of a realistic human head exploiting a customized parallel ray-tracing engine, and integration of it in the numerical formulation of the diffusion equation through an iterative approach thus obtaining a complete model for light distribution evaluation in a full human head derived from MRI.

As shown at the beginning of this section, EEG is gaining popularity in extra-clinical applications, with particular focus on BCI systems.
The dissemination of EEG-based BCI systems is limited due to some drawbacks. The majority of EEG acquisition systems are in fact high-expensive [17], thus not affordable even for research centres and universities. While several low-cost systems in the range of $500-$1000 are available, however they are usually equipped with a small number of channels, have moderate to high noise and users cannot modify the position of the electrodes on the scalp.
Several real-time systems and open source software tools are available to help scientists to execute BCI experiments, but there are no standard libraries that allow one to connect these BCI software tools with EEG acquisition systems, making difficult to use these hardware outside of the laboratory in which were designed.
Without stable software libraries which provide a way to connect the system to a BCI software tool, it is difficult to assess how the system functions and indeed whether it functions at all. If the system does not perform as

expected it can be complicated to determine if the fault is due to a hardware or software issue and the time spent in trying to identify the source of error and correcting it can be substantial.

In this scenario, my research team developed an Arduino-based cost-effective EEG system called *Creamino*, which has a fabrication cost of about 50 euros for the first 8 channels (working system, including wet-contact active electrodes) and 30 euros per each additional 8 channels. These numbers are particularly attractive for systems designed to be used outside clinical environments, such as in home care or research-oriented applications. In addition, a set of libraries which allows the system to be used in a variety of software environments has been developed. *Creamino*, with executables, source code and documentation is available free of charge for research and educational purposes at https://github.com/mchiesi/Creamino.

In the Section B of this thesis, we present the development of Simulink libraries for EEG Source Localization able to compute the mean power of the EEG sources lying in user-defined regions of interest (ROIs).

Since certain experiments might need to analyze EEG sources localization in the frequency domain rather than in time-domain, the implementation of frequency EEG Source Localization algorithms has also been realized.

Finally, in complex geometries such as the human head, it could be difficult for the user to imagine the results of EEG source localization using multiples 2D plots of the reconstructed solution in various sections of the domain. To tackle this problem, I implemented an accelerated 3D visualization toolbox based on Nvidia OptiX ray-tracing engine, which allows displaying normalized distributions over complex 3D surfaces making possible user interactions.

This work is organized in two different sections: Section A is relative to the main contribution of my research activity developed in the scope of the "HIGH PROFILE" European project, while Section B illustrates a secondary contribution of my research activity which was developed in the scope of the "CREAM" European project.

In order to easily understand the work explained in this thesis, the theoretical basis of Brain Functional Imaging are presented in Chapter 1.

Section A is organized as follows: Chapter 2 presents the basis of light propagation theory, necessary to understand the Finite Volumes numerical formulation of the DOT forward problem solver presented in Chapter 3. In Chapter 4 the DOT inverse problem is first introduced and then solved through the implemented forward problem solver. The objective of Chapter 5 is to employ the developed DOT software using experimental measurements on phantom made by materials mimicking the optical properties of a real human head in order to validate the experimental setup. Finally, some conclusions of the research activity presented in Section A are drawn in Chapter 6.

Section B is organized as follows: Chapter 7 presents the basis of EEG Source Localization and BCI systems to easily understand the implementation of the SIMULINK libraries for EEG source localization in the time-domain presented in Chapter 8. An accelerated solver for EEG source localization in the frequency domain is proposed and validated in Chapter 9. Chapter 10 focuses on the implementation of a parallel visualization toolbox for the 3D rendering of light or electrical distributions on complex surfaces. Some conclusions of the work explained in Section B are drawn in Chapter 11. Finally, in Chapter 12, some conclusions of the whole research activity are presented.

Chapter 1

Brain Functional Imaging

Brain Imaging comprises a set of techniques to analyze the structural or functional behaviour of the brain in normal or pathological situations. While brain structural imaging provides morphological description of head tissues, brain functional imaging provides meaningful information about the relationship between brain area activation and specific cerebral or physical function.

The possible applications of brain functional imaging involves the mapping of brain activity in healthy subjects, the assessment of the effects of stroke, trauma or degenerative disease (such as Alzheimer's disease) on brain function and the monitoring of the growth and function of brain tumours. These applications make brain imaging a powerful and appealing tool in diagnosis and treatment of central nervous system diseases and general study of brain activity.

This chapter gives an overview of the common brain imaging techniques (Sec. 1.1) and their medical applications (Sec. 1.3), focusing on high-portable and economical techniques such as DOT (Sec. 1.3.1) and EEG (Sec. 1.3.2). The object of this chapter is to provide an overview about the state of the art of the Brain Imaging techniques and to introduce the research activity described in this thesis.

1.1 Brain Functional Imaging techniques

Positron Emittance Tomography (PET) detects pairs of gamma rays emitted indirectly by a positron-emitting radionuclide (tracer), which is introduced

into the body on a biologically active molecule. Three-dimensional images of tracer concentration within the body are then constructed by computer analysis.
The greatest benefit of PET scanning is that different compounds can show blood flow and oxygen and glucose metabolism in the tissues of the working brain. These measurements reflect the amount of brain activity in the various regions of the brain and allow to learn more about how the brain works. PET scans were superior to all other metabolic imaging methods in terms of resolution and speed of completion (as little as 30 seconds), when they first became available. The improved resolution permitted better study to be made as to the area of the brain activated by a particular task. The biggest drawback of PET scanning is that because the radioactivity decays rapidly, it is limited to monitoring short tasks [18].

Functional MRI (fMRI) allows to obtain functional imaging of the brain maintaining the high spatial resolution typical of the electro-magnetic techniques. This is based on the same principles of static MRI, but it is able to detect variations of oxygen concentration in the blood. When neurons are activated, the supply of blood to the active region increases. For reasons that are still unclear, the delivery of oxygenated haemoglobin to the region is greater than local oxygen consumption, resulting in a greater proportion of oxygenated to deoxygenated haemoglobin, which causes a local variation of the magnetic properties that can be detected with the MRI process. fMRI is so be used to examine the activation of the brain regions during cognitive tasks, but with a temporal resolution less than 1Hz [19].

Electroencephalography (EEG) is a brain imaging techniques based on recordings of potential differences on the scalp using a set of electrodes. These potentials originate from the activity of the neurons in the gray matter (cerebral cortex), causing an electric field to propagate to the scalp. Electric field propagation is limited by the electrical properties of skull, resulting in potential differences on the scalp of only few μV [20] which limit the spatial resolution of the method.

Electrical Impedance Tomography (EIT) is based on injection of small currents (less than 1 mA at frequency starting from 1 KHz) at the scalp surface using active electrodes. A set of passive electrodes measures the potential differences caused by the induced electric fields. Data detected are then elaborated in order to obtain an impedance distribution map on the brain, which indicates the presence of particular pathologies like epileptic foci, local ischemia and lesions [21].

Diffuse Optical Tomography (DOT) is based on the principle that visible and near-infrared (NIR) light interact with human tissues predominantly by absorption and scattering. By the injection of light inside the human head and using optical sensors on its surface, is possible to obtain a map of optical properties of the domain under study. Optical properties that characterize a medium are the absorption coefficient, the scattering coefficient and the refractive index (considered to be homogeneous in human head's tissues [8]). In particular the principle of DOT is based on the fact that variations in O_2 concentration in haemoglobin can be extracted by measuring the change in the absorption coefficient of light.

1.2 Brain Functional Techniques Comparison

Between the techniques listed above, MRI and PET represent classical solutions. Classical solutions provide high spatial resolution at the expense of high cost, complexity and size, limiting their employment to advanced research centers or hospitals. EIT, EEG and DOT are characterized by high portability, high temporal resolution, non-invasivity and limited cost but low spatial resolution. These are typically based on simple recording hardware but a computational intensive data elaboration process is often required.

Fortunately, advances in scientific computing have led to the development of architectures featuring high computational performance at accessible cost, which can be used in different research fields and applications. A typical example is the Graphic Processing Units (GPUs) [22], a programmable multicore architecture which allows high performance computing in a standard PC.

In brain imaging, this computational power allows the development of low-cost and portable techniques, where the reduced complexity and size of the acquisition system is balanced by the compute intensive data processing, also within real time constraints. A typical example of this trend is the recent development of BCI systems, which interpret human brain activity from real-time measurements picked up by portable recording hardware.

1.3 Medical Applications of DOT, EEG and EIT

EEG imaging is used in the diagnosis of particular pathologies, like detection and localization of epileptic foci [23]. In case of intractable epilepsy, surgery is required to remove the affected portion(s) of the brain. Consequently, continuous monitoring and analysis using brain imaging techniques is necessary in order to accurately identify the area to extract. This can be combined with invasive procedures, like surgical implant of intra-cranial electrodes in the subject brain. However, other brain imaging techniques (as the EIT) are able to improve localization performance of EEG and they could help to avoid this invasive treatment. For this purpose, several studies show how EIT could theoretically be used to detect epileptic foci, and others pathologies like brain lesions and local ischemia [24].

DOT offers the opportunity to image three-dimensional (3-D) spatial variations in blood parameters, particularly haemoglobin concentration and oxygen saturation, and thus metabolic factors which these concentrations reflect, along with tissue scattering characteristics. These features are extremely appealing and appreciated in breast and brain care, particularly stroke, as well as during and following brain surgery [25].

In breast imaging, DOT has unique capabilities for imaging functional parameters such as metabolism, blood flow, blood volume, and oxygen saturation. These parameters are modified by tumour angiogenesis and are also important for following the response to therapeutic intervention. Tumours generally are more highly vascularized than surrounding tissue, thus leading to differential light absorption properties, and in addition relative Hb/HbO_2 concentration may not only differentiate tumours from background tissue but also may discriminate among tumours with different activity rates (i.e., degree of malignancy).

As well as for diagnosis and treatment of pathology, alternative techniques can also be employed for study of the brain functionality. In particular, being

EEG non-invasive and painless, it is used to study cognitive processes of the brain, e.g. memory, attention and perception. These studies are based on identification of Event Related Potentials (ERPs) in neural activity [26]. ERPs are voltage variations induced in the subject's brain by external stimuli, and are used to detect brain regions involved in different cognitive tasks (attention, memory, language processing). ERPs dynamics are not detectable by standard fMRI, so EEG is preferred for its higher temporal resolution. In addition to measure concentrations of haemoglobin and blood volume with high temporal resolution, DOT has the capability to potentially measure fast scattering changes associated with neuronal activity [27][28], leading to elucidate the hemodynamic response to neuronal activity and thus lead to an understanding of the underlying mechanisms.

As introduced in Section 1.2, the reduced complexity and size of the EEG acquisition system, combined with its advanced state of employment in clinical contexts, make this technique suitable for a BCI system. As shown in Fig. 1.1, BCI is a communication system interfacing the human brain to external devices, like computers or actuators. User commands are formed by recognizing brain activity with brain imaging techniques, typically EEG or invasive electrodes surgically positioned on the cerebral cortex. BCIs are often designed to assist, augment or repair human cognitive or sensory-motor functions; particular attention is paid to the realization of prostheses controlled by BCI in order to replace damaged human functions

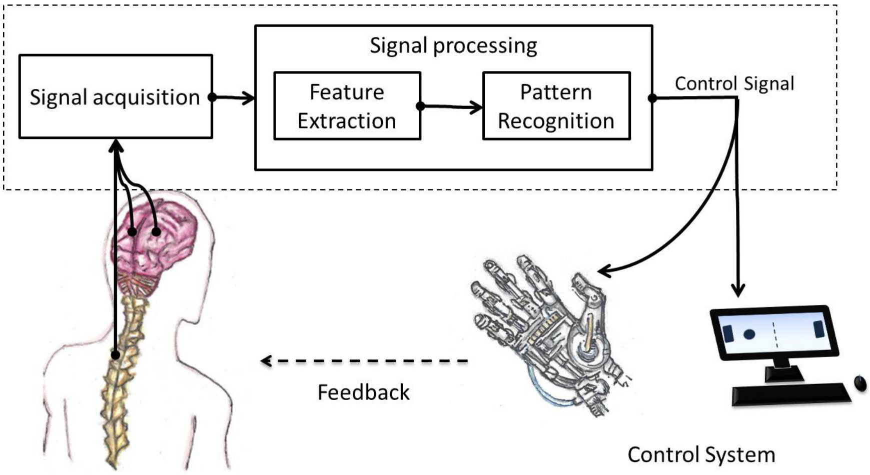


Figure 1.1: BCI neurofeedback (source: [29]).

like hearing, sight and movement. Voltages measured by electrodes are sent to a computer. Data are interpreted to compute actuators commands and the feedback is closed by the subject's perception of actuator actions or movements.

1.3.1 DOT

DOT is based on the principle that visible and Near-Infrared (NIR) light interact with biological tissue predominantly by absorption and elastic scattering. There are several physiologically interesting molecules which have characteristic absorption spectra at these wavelenghts. In particular, the spectra of haemoglobin (HbO_2) and deoxy-haemoglobin (Hb) differ markedly, as shown in Figure 1.2. Haemoglobin provides an indicator of blood volume and oxygenation and the different absorption spectra of HbO_2 and Hb are frequently exploited in physiological monitoring techniques such as pulse oximetry and near-infrared spectroscopy (NIRS). The aim of Diffuse Optical Tomography (DOT) is to produce spatially resolved images, which may display the specific absorption and scattering properties of the tissue, or physiological parameters such as blood volume and oxygenation.

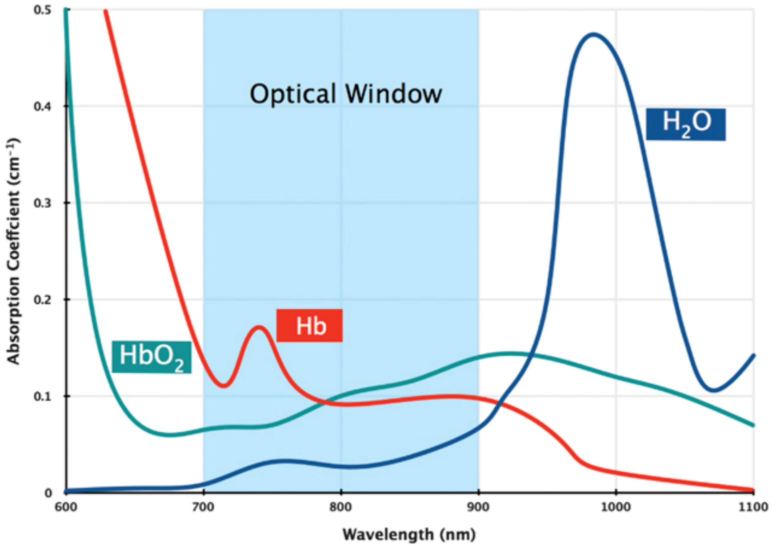


Figure 1.2: Absorption of Hb and HbO_2 in function of the wavelength (source: [30]).

DOT Forward and Inverse Problem

DOT image reconstruction requires an optimal combination of a forward and an inverse problem solver. The purpose of the forward problem is to compute the light distribution inside the volume under consideration given complete information as to the light sources and the optical properties within the domain, from which a set of light exitance values can be determined (i.e. the light irradiated through the surface underneath a detector). An example of forward problem solution on a 3D sphere with homogeneous optical properties is represented in Fig. 1.3:

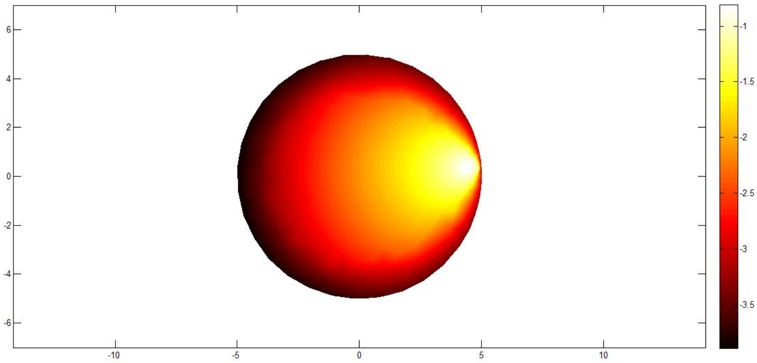


Figure 1.3: Example of Light Distribution provided by forward problem solution on a 3D Sphere with Homogeneous Optical Properties.

By contrast, the inverse problem solution provides an estimated reconstruction of the local optical properties of the volume under examination given the configuration of light sources and the set of experimental measurements [31]. In Fig. 1.4 is shown an example of DOT image reconstruction in a cylinder with homogeneous absorption coefficient which contains two anomalies. Using a light source placed at the top of the cylinder and a set of light-detectors laid on the red line, it is possible to reconstruct the spatial-resolved slice shown in the left of the figure. In general, the aim of DOT is to provide a map of absorption of the volume under study given a relative small number of light detectors on the surface of the object. Typically, the number of light detectors is much smaller than the number of voxels (in the work-flow illustrated in Chapter 3 voxels are more than 5 millions) leading to a typical under-determined system. This system can be solved using a numerical approach as the Least SQuared Residuals method (LSQR) using regularization techniques. Furthermore, an inverse problem solution requires multiple

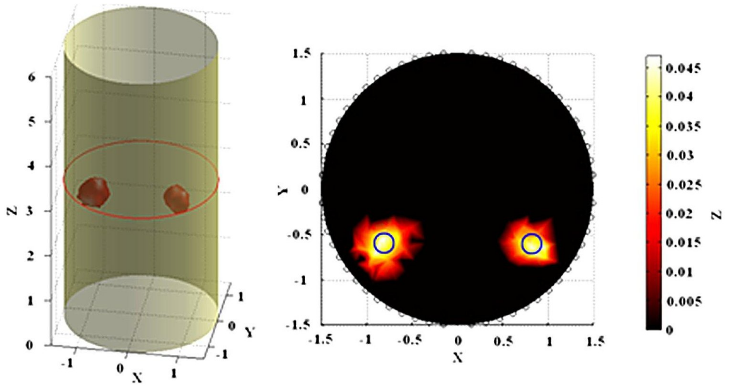


Figure 1.4: Example of DOT reconstruction using sensors (red line) and a source at the top of the cylinder (source: [32]).

computations of the forward problem solution, calling for accurate and time-economical forward solvers. Several methods have been developed in order to provide an appropriate trade-off between these two requirements, ranging from accurate, highly intensive transport-based approaches [4] to diffusive approximations [5], through hybrid radiosity-diffusion methods [6] examined in detail in Chapter 2.

1.3.2 BCI systems and EEG Source Localization

EEG is based on the measurement and mapping of electrical activity produced by the brain as it is being recorded by electrodes placed on the scalp. The recorded data are then plotted into a standard chart, the so-called *electroencephalogram* and visually examined by the physicians.
The potential measurements on the scalp are performed in a differential manner using electrodes, which are most commonly passive, but can include active circuitry [33]. These potentials originate from the activity of the neurons in the cerebral cortex, also known as gray matter. The electric field then propagates through the skull to the scalp. Field propagation is limited by the electrical properties of the skull and hence the potential difference are usually in the μV range [20].
Recently, due to the increasing computational power of cost-accessible platforms, brain imaging techniques have been presented based on the elaboration of data recorded by EEG, in order to avoid high costs of electromagnetic-based solutions, e.i. fMRI. Aim of this new field of application, known as

EEG source imaging, is to provide functional images of neuronal activity in the human cerebral cortex, in particular for the localization of active areas at each time-frame (*source localization*).

EEG source imaging is characterized by an acquisition hardware, which is relatively simple and portable (high impedance electrodes) and by an intensive task of post-processing of acquired data (*EEG inverse problem*). Unfortunately, many different sources configurations can generate the same potentials on the scalp [34], that means that the mathematical problem is ill-posed. This can be solved through algorithms which rely on different approaches, using mathematical, biophysical, statistical or anatomical constraints [35] [36].

This introduces severe computation requirements, which could limit the high temporal resolution provided by the measurement instrumentation. Difficulties arise particularly when the problem needs to be solved under real-time constraints, as required by modern BCI applications.

To satisfy real-time constraints of modern BCI systems, a massive parallel implementation of the EEG source localization workflow is necessary.

Furthermore, to facilitate the integration of EEG source localization algorithms in commercial BCI systems, the development of standard libraries is necessary and it is treated in the Section B of this thesis.

EEG Source Localization: Forward and Inverse Problem

As introduced before, EEG computational work flow involves two main tasks: forward problem and inverse problem.

A forward problem is defined as the determination of the voltage distribution in the volume under analysis, given its electrical properties and the electrical sources (active voxels) distribution (see Fig. 1.5). Due to the complexity of the domain, analytical solutions are not feasible, and one needs to rely on numerical solvers. Computational resources that are required by a realistic head model are high, thus, simplified regular geometries based on volume shape approximation are often used; however, the limits of this approach in respect to realistic head model are proven [38].

The inverse problem, on the other hand, is defined as the estimation of the electrical sources distribution inside the human brain, given the electrodes measurements and the relationship between the voltage and voxels distribu-

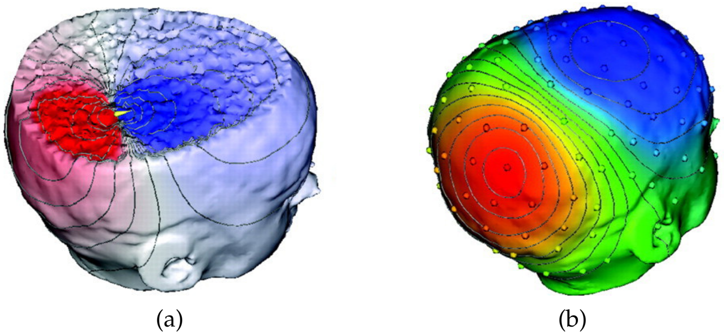

(a) (b)

Figure 1.5: The EEG Forward problem defines the relationship between the voxels's activation inside the brain volume (a) (known its head morphology and conductivity distribution) and the resulting voltage at the electrodes (b) (source: [37]).

tion defined by the forward problem (1.6). Given the relative small number of electrodes, generally from 16 to 128 in proportion to the voxels number, this problem is obviously ill-posed and with multiple solutions. A restriction in the solutions domain is necessary, in order to select the more realistic and physiologically correct solution among the others. The formulation of the inverse problem leads to an overdetermined system (more unknowns than equations), which can be solved with a numerical approach based on functional minimization coupled with regularization techniques.

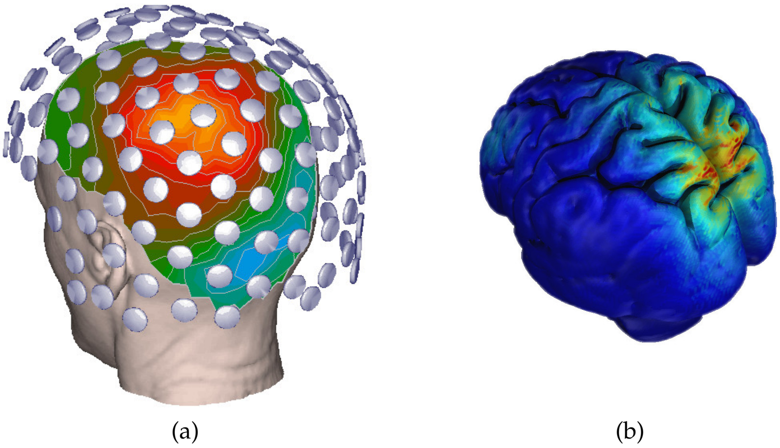

(a) (b)

Figure 1.6: EEG inverse problem. Electrodes potentials are elaborated to localize active voxels inside the brain.

Part I

Section A : DOT for Brain Functional Imaging

Chapter 2

Diffuse Optical Tomography Theory

When light propagates inside a region (in our specific case, Near-Infrared light propagating through the different tissues in the human head), the photon distribution which creates on its surface (in our specific case, the scalp) is related to internal optical properties. The aim of a photon transport model is to determine the radiance in a region Ω characterized by specific optical properties.

This chapter introduces the main equations which model light propagation within a medium, ranging from accurate, highly intensive transport-based approaches (Sec. 2.2) to diffusive approximations (Sec. 2.3), through hybrid radiosity-diffusion methods (Sec. 2.4).

The objective of this chapter is to give the theoretical basis of light propagation to understand the numerical implementation of the DOT forward solver proposed in Chapter 3.

2.1 Light Propagation: Principle and Basic Definitions

Let us consider a flow of wave energy at a point $\mathbf{r}$ in a random medium having absorption coefficient μ_a [m^{-1}], scattering coefficient μ_s [m^{-1}], single-particle absorption cross section σ_a [m^2] and single-particle scattering cross section σ_s [m^2]. Parameters (μ_a,μ_s) define the optical properties of the medium. For a given direction defined by a unit vector $\hat{\mathbf{s}}$, we define the stationary specific intensity (or radiance) $\phi(\mathbf{r}, \hat{\mathbf{s}})$ as the average power flux

density within a unit frequency band centered at frequency ν within a unit solid angle $\boldsymbol{\omega}$.
This quantity is measured in $\mathrm{W\,m^{-2}\,sr^{-2}\,Hz^{-1}}$. The amount of power dP flowing within a solid angle $d\omega$ through an elementary area da oriented in the direction given by the unit normal vector $\hat{\mathbf{s}}_\mathbf{0}$ in a frequency interval $(\nu, \nu + d\nu)$ is given by:

$$dP = \phi(\mathbf{r}, \hat{\mathbf{s}})\ \cos\theta\, da\, d\boldsymbol{\omega}\, d\nu,$$

where θ is the angle between $\hat{\mathbf{s}}$ and $\hat{\mathbf{s}}_0$. The specific intensity $\phi(\mathbf{r}, \hat{\mathbf{s}})$ and the amount of power dP are represented in Figure 2.1.

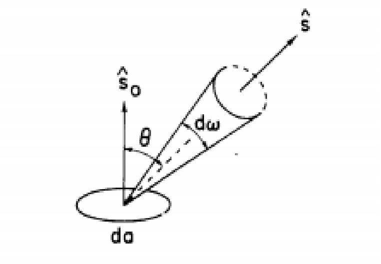


Figure 2.1: Specific intensity $\phi(\mathbf{r}, \hat{\mathbf{s}})$ and power dP (source: [39]).

2.2 Radiative Transfer Equation

A full description of light transport in tissue is provided by the radiative transport equation (RTE) [39] [40], which can be derived as an approximation of Maxwell's equation which does not include wave effects [41]. Under the assumption that the wavelength is much smaller than the dimensions of the object under study (which is the case under examination), wave effects can be neglected and RTE provides an accurate model for light propagation. The RTE is a conservation equation which states that the radiance $\phi(\mathbf{r}, \hat{\mathbf{s}}, t)$, for photons travelling from point $\mathbf{r}$ in direction $\hat{\mathbf{s}}$ at time t, is equal to the sum of all the mechanisms which increase $\phi(\mathbf{r}, \hat{\mathbf{s}}, t)$ minus those reducing it.

Let us consider a stationary specific intensity $\phi(\mathbf{r}, \hat{\mathbf{s}})$ incident upon a cylindrical elementary volume with unit cross section and length ds. This volume contains ρds particles where ρ is the number of particles in a unit volume. Each particle absorbs the power $\sigma_a(\mathbf{r})\phi(\mathbf{r}, \hat{\mathbf{s}})$ and scatters the power

$\sigma_s(\mathbf{r})\phi(\mathbf{r},\hat{\mathbf{s}})$, and therefore, the decrease of the specific intensity $d\phi(\mathbf{r},\hat{\mathbf{s}})$ for the volume ds is expressed as

$$d\phi(\mathbf{r},\hat{\mathbf{s}}) = -ds\big(\mu_a(\mathbf{r}) + \mu_s(\mathbf{r})\big)\phi(\mathbf{r},\hat{\mathbf{s}}) \tag{2.1}$$

where $\mu_a(\mathbf{r}) = \rho\sigma_a(\mathbf{r})$ and $\mu_s(\mathbf{r}) = \rho\sigma_s(\mathbf{r})$. At the same time, the specific intensity increases when the scattering from other directions $\hat{\mathbf{s}}'$ is focused into the direction $\hat{\mathbf{s}}$, as shown in Figure 2.2. In order to determine this contribution for the specific intensity $\phi(\mathbf{r},\hat{\mathbf{s}})$ we need to take into account the contribution coming from all the directions $\hat{\mathbf{s}}'$ scattered into direction $\hat{\mathbf{s}}$ thanks to the total number of particles $\rho\, ds$ in the volume. This amount is given by the following integral:

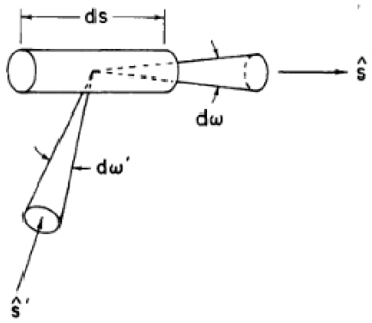


Figure 2.2: Scattering of specific intensity incident upon the volume ds from the direction $\hat{\mathbf{s}}'$ into the direction $\hat{\mathbf{s}}$ (source: [39]).

$$\int_{4\pi} \rho\, ds\, |f_s(\hat{\mathbf{s}},\hat{\mathbf{s}}')|^2\, \phi(\mathbf{r},\hat{\mathbf{s}}')\, d\boldsymbol{\omega}' \tag{2.2}$$

where the integration over all $\boldsymbol{\omega}'$ is taken to include the contribution from all directions $\hat{\mathbf{s}}'$ and $f_s(\hat{\mathbf{s}},\hat{\mathbf{s}}')$ is the scattering function. In general $f_s(\hat{\mathbf{s}},\hat{\mathbf{s}}')$ is expressed in function of the phase function $p(\hat{\mathbf{s}},\hat{\mathbf{s}}')$ which is defined as

$$p(\hat{\mathbf{s}},\hat{\mathbf{s}}') = \frac{4\pi}{\sigma_t(\mathbf{r})}|f_s(\hat{\mathbf{s}},\hat{\mathbf{s}}')|^2$$

with the following property:

$$\frac{1}{4\pi}\int_{4\pi} p(\hat{\mathbf{s}},\hat{\mathbf{s}}')d\boldsymbol{\omega} = \frac{\sigma_s(\mathbf{r})}{\sigma_t(\mathbf{r})},$$

where $\sigma_t(\mathbf{r}) = \sigma_a(\mathbf{r}) + \sigma_s(\mathbf{r})$.
Therefore (2.2) becomes:

$$\int_{4\pi} \rho\, ds\, |f_s(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')|^2\, \phi(\mathbf{r}, \hat{\mathbf{s}}\,')\, d\boldsymbol{\omega}\,' = \frac{\rho ds \sigma_t(\mathbf{r})}{4\pi} \int_{4\pi} p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')\, \phi(\mathbf{r}, \hat{\mathbf{s}}\,') d\boldsymbol{\omega}\,' \tag{2.3}$$

Finally, the specific intensity may increase due to the emission from the volume ds

$$ds\, \epsilon(\mathbf{r}, \hat{\mathbf{s}}) \tag{2.4}$$

where $\epsilon(\mathbf{r}, \hat{\mathbf{s}})$ is the source power radiation per unit volume per unit solid angle in the direction $\hat{\mathbf{s}}$. Adding the contributions (2.1), (2.3) and (2.4) we obtained the stationary Radiative Transfer Equation (RTE):

$$\frac{d\phi(\mathbf{r}, \hat{\mathbf{s}})}{ds} = -\rho\sigma_t(\mathbf{r})\, \phi(\mathbf{r}, \hat{\mathbf{s}}) + \frac{\rho\sigma_t(\mathbf{r})}{4\pi} \int_{4\pi} p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')\, \phi(\mathbf{r}, \hat{\mathbf{s}}\,') d\boldsymbol{\omega}\,' + \epsilon(\mathbf{r}, \hat{\mathbf{s}}). \tag{2.5}$$

The RTE in the time domain expressed in a compact form is:

$$\left(\frac{1}{c}\frac{\partial}{\partial t} + \hat{\mathbf{s}} \cdot \nabla + \mu_t(\mathbf{r}, t)\right)\phi(\mathbf{r}, \hat{\mathbf{s}}, t) = \mu_s(\mathbf{r}, t) \int p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')\phi(\mathbf{r}, \hat{\mathbf{s}}, t) ds' + \epsilon(\mathbf{r}, \hat{\mathbf{s}}, t). \tag{2.6}$$

where $\mu_t(\mathbf{r}, t) = \mu_a(\mathbf{r}, t) + \mu_s(\mathbf{r}, t)$ is called the extinction coefficient and c is the light speed.
In the frequency domain, the RTE can be simply re-written as:

$$\left(\frac{\imath f}{c} + \hat{\mathbf{s}} \cdot \nabla + \mu_t(\mathbf{r}, f)\right)\phi(\mathbf{r}, \hat{\mathbf{s}}, f) = \mu_s(\mathbf{r})(\mathbf{r}, f) \int p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')\phi(\mathbf{r}, \hat{\mathbf{s}}, f) ds' + \epsilon(\mathbf{r}, \hat{\mathbf{s}}, f). \tag{2.7}$$

where f is the frequency.
Unfortunately, solving the RTE in non-homogeneous volumes having complex shapes as required in optical tomography is not feasible through analytical means and is an extremely computationally-intensive task if solved numerically or statistically (e.g. through Monte Carlo methods)[7][31]. Simpler models need to be implemented, which rely on assumptions on whether the optical characteristics of the tissue under examination lead to either diffusive or ray optics models.

2.3 Diffusion Equation

In general, a diffusion equation can be derived as a first order spherical approximation of the radiative transfer equation and it is suitable to model light propagation through highly-scattering media, where light propagates diffusively [9][42].

Three variables in the RTE depend on direction: the radiance $\phi(\mathbf{r}, \hat{\mathbf{s}}, t)$, the phase function $p(\hat{\mathbf{s}}, \hat{\mathbf{s}}')$ and the source term $\epsilon(\mathbf{r}, \hat{\mathbf{s}}, t)$. If these variables are expanded into spherical harmonics under three assumptions:

- the photon flux changes slowly;
- the phase function is independent by the absolute angle;
- all sources are isotropic;

we obtain:

$$\epsilon(\mathbf{r}, \hat{\mathbf{s}}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \epsilon_{\ell,m}(\mathbf{r})\, Y_{\ell,m}(\hat{\mathbf{s}})$$

$$\phi(\mathbf{r}, \hat{\mathbf{s}}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \phi_{\ell,m}(\mathbf{r})\, Y_{\ell,m}(\hat{\mathbf{s}})$$

$$p(\hat{\mathbf{s}}, \hat{\mathbf{s}}') = p(\hat{\mathbf{s}} \cdot \hat{\mathbf{s}}') = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} p_\ell\, Y^*_{\ell,m}(\hat{\mathbf{s}}')\, Y_{\ell,m}(\hat{\mathbf{s}}).$$

Substituting the expansions listed above within RTE and using recurrence relations leads to an infinite series of equations:

$$\sigma_t\, \phi_{\ell,m} + \frac{\partial}{\partial z}\Big(\alpha_{\ell+1,m}\, \phi_{\ell+1,m}(\mathbf{r}) + \alpha_{\ell,m}\, \phi_{\ell-1,m}(\mathbf{r})\Big) +$$

$$-\frac{1}{2}\left(\frac{\partial}{\partial x} - \imath \frac{\partial}{\partial y}\right)\Big(\beta_{\ell,m}\, \phi_{\ell-1,m-1}(\mathbf{r}) - \beta_{\ell+1,-m+1}\, \phi_{\ell+1,m-1}(\mathbf{r})\Big) +$$

$$-\frac{1}{2}\left(\frac{\partial}{\partial x} + \imath \frac{\partial}{\partial y}\right)\Big(-\beta_{\ell,-m}\, \phi_{\ell-1,m+1}(\mathbf{r}) + \beta_{\ell+1,m+1}\, \phi_{\ell+1,m+1}(\mathbf{r})\Big) =$$

$$= \frac{\rho \sigma_t}{4\pi} p_\ell\, \phi_{\ell,m}(\mathbf{r}) + \epsilon_{\ell,m}(\mathbf{r}).$$

By taking the first N spherical harmonics of the RTE we obtain $(N+1)^2$ coupled partial differential equations. For a complete argumentation about RTE derivation and its spherical harmonics development please refer to [43] and [40].

The diffusion equation (DE) in the frequency domain arises from the first order expansion of RTE ($N = 1$) leading to the following formulation:

$$-\nabla \cdot \kappa(\mathbf{r}, f)\nabla\Phi(\mathbf{r}, f) + \mu_a(\mathbf{r})(\mathbf{r}, f)\Phi(\mathbf{r}, f) + \frac{\imath f}{c}\Phi(\mathbf{r}, f) = q_0(\mathbf{r}, f) \ \forall \mathbf{r} \in \Omega\,; \tag{2.8}$$

where

$$\Phi(\mathbf{r}, f) = \int_{4\pi} \phi(\mathbf{r}, \hat{\mathbf{s}}, f)\, d\boldsymbol{\omega}$$

is the spectral irradiance,

$$q_0(\mathbf{r}, f) = \int_{4\pi} \epsilon(\mathbf{r}, \hat{\mathbf{s}}, f)\, d\boldsymbol{\omega} \tag{2.9}$$

is the isotropic source distribution at frequency f and

$$\kappa(\mathbf{r}, f) = \frac{1}{3(\mu_a(\mathbf{r}, f) + \mu_s'(\mathbf{r}, f))} \tag{2.10}$$

is the diffusion coefficient, where

$$\mu_s(\mathbf{r}, f)' = (1 - g)\mu_s(\mathbf{r}, f) \tag{2.11}$$

is the reduced scattering coefficient and g is the mean cosine of the scattering angle given by:

$$g = \frac{1}{4\pi}\int_{4\pi} p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,')\cos(\theta) d\boldsymbol{\omega}\,' \tag{2.12}$$

which is also called *anisotropy factor*. The scattering phase function most typically employed is the commonly used Henyey-Greenstein scattering function [44],[45],[46]:

$$p(\hat{\mathbf{s}}, \hat{\mathbf{s}}\,') = \frac{1 - g^2}{2\big(1 + g^2 - 2g\cos(\theta)\big)^{3/2}}$$

For a detailed derivation of DE, the reader is referred to [31] and [39].

The values of g for biological tissues are typically of the order of 0.9, indicating strongly forward biased scattering.

The reduced scattering can be interpreted as the equivalent isotropic scattering coefficient and it is a fundamental parameter in the diffusion theory of light propagation.

In particular, as shown in Fig. 2.3, the purpose of $\mu_s'(\mathbf{r}, f)$ is to describe the diffusion of photons in a random walk of step size $1/\mu_s'(\mathbf{r}, f)$ where each step involves isotropic scattering.

This is equivalent to represent the photon movement by using many steps

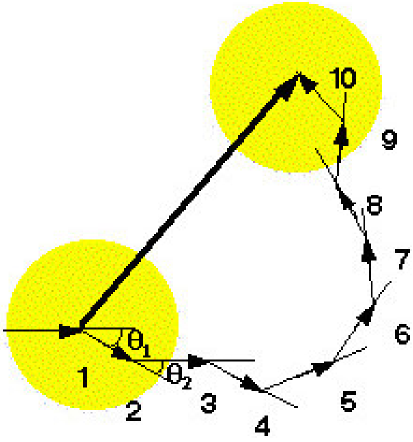


Figure 2.3: Equivalence between a step of random walk having step size $1/\mu_s'(\mathbf{r}, f)$ and 10 steps of random walk of step size $1/\mu_s(\mathbf{r}, f)$ (source: [47]).

of smaller size $1/\mu_s(\mathbf{r}, f)$ and involving a partial deflection angle $\arccos(g)$ at each step. The equivalence is valid only if there are many scattering events before an absorption event, that is verified if $\mu_a(\mathbf{r}, f) \ll \mu_s'(\mathbf{r}, f)$.

This situation of scattering-dominated light transport is called the diffusion regime, and this is the necessary state for diffusion equation to provide accurate results.

This is generally the case of the most part of tissues, but the assumption breaks down in regions of either high absorption or low scatter, for example in the Cerebro Spinal Fluid (CSF). In these situations different approaches are required.

Furthermore, diffusion approximation requires light propagation to be either isotropic or weakly anisotropic. This condition is not verified near sources and boundaries. However, comparisons of diffusion calculation with experiments [48][49][50][51] and Monte Carlo simulations [52][53][54] show that correct predictions of boundary measurement can be obtained with diffusion models.

Schweiger *et al.* in [55] state that diffusion equation represents a valid model for calculating measurement signals under the specification of appropriate

boundary conditions. In the following sections we will introduce the possible boundary conditions of diffusion equation and how light sources can be satisfyingly represented under this approximation.
The numerical model which will be introduced in Chapter 3 is implemented to support all the following boundary conditions.

2.3.1 Boundary Conditions

If Ω is the domain under analysis and $\partial\Omega$ represents its boundary, then the Dirichlet boundary condition (DBC) states that:

$$\Phi(\boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \partial\Omega\,. \tag{2.13}$$

Physically this is equivalent to a perfect absorbing medium that surrounds the domain Ω. Even if it is a simple and commonly used condition, it should be pointed out that it involves a non-realistic behaviour of the surrounding medium.
Robin boundary condition (RBC), defined as

$$\Phi(\boldsymbol{\xi}) + 2\kappa(\boldsymbol{\xi})\vec{n}\cdot\nabla\Phi(\boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \partial\Omega \tag{2.14}$$

models the more realistic condition of a non scattering medium surrounding the domain. In this equation $\vec{n}$ is the outward normal to $\partial\Omega$ in $\boldsymbol{\xi}$, and no diffuse surface reflection is considered. Equation 2.14 implies that each photon, after crossing $\partial\Omega$, exits from the domain without back-scattering phenomena, and can be modified to incorporate a mismatch between the refractive indices n_1 within the domain and n_2 in the surrounding medium. The modified Robin boundary condition becomes:

$$\Phi(\boldsymbol{\xi},\omega) + 2R\kappa(\boldsymbol{\xi},\omega)\mathbf{n}(\boldsymbol{\xi})\cdot\nabla\Phi(\boldsymbol{\xi},\omega) = 0 \quad \forall \boldsymbol{\xi} \in \partial\Omega\,; \tag{2.15}$$

with

$$R = \frac{2/(1-R_0) - 1 + |cos(\theta_c)|^3}{1 - |cos(\theta_c)|^2}$$

where $\theta_c = \arcsin(\frac{n_2}{n_1})$ is called *critical angle*, and $R_0 = \frac{(n_1-n_2)^2}{(n_1+n_2)^2}$. Setting $n_1 = n_2$ then $R = 1$, the modified RBC folds back to the standard RBC without internal reflection. On the other side, in the case where $\frac{n_2}{n_1}$ tends to zero, the critical angle tends to zero as well and total internal reflection occurs. In this situation, A tends to infinity and RBC leads to a Neumann boundary condition (NBC) which basically states that there is no flux of photons through $\partial\Omega$.

2.3.2 Modelling of collimated and diffuse sources

The diffusion equation introduces the assumption that all sources in the model are isotropic, because direction dependency can not be intrinsically represented when using this kind of approximation.

Internal and isotropic sources can be considered directly in diffusion equation by setting $q_0(\mathbf{r}, f)$ in Eq. 2.9 with the consistent light source distribution [55].

In the more realistic case of light source incident at a point on the boundary, we distinguish two cases:

1. Collimated source;
2. Diffuse source.

In the first case, we take into account a light source having a unique direction (such as a laser beam), with all photons travelling through parallel paths. The diffusion equation is not able to describe these kind of anisotropic sources correctly and, in particular, collimated beams since the equation itself derives from averaging the light behaviour over a complete solid angle, removing any information on the photon direction.

A common approach is to represent a collimated beam by an isotropic source located at a depth $1/\mu_s'(\mathbf{r})$ below the domain's boundary.

This quantity is called mean free path and represents the average distance at which all photons can be considered to have been scattered at least one time. This approach produces accurate results at distances from the source larger than the mean free path, but breaks down near the source. This is of minor importance in diffuse tomography, since tissues of interests are located at least at $1-2$ cm from the source, with a typical mean free path of $0.5-1$ mm.

A diffuse source is one which emits photons uniformly over the solid angle of interest. In the case of diffuse sources located on the domain's boundary, an inward directed diffuse photon current distributed over the illuminated surface area is represented by simply adding a right-hand side to equations 2.13 – 2.15.

2.4 Radiosity Equation

As shown in the previous section, the diffusion approximation is widely and successfully used to model light propagation in highly scattering tissues ($\mu_s'(\mathbf{r}, f) \gg \mu_a(\mathbf{r}, f)$), reducing the complexity of the RTE to a second-order partial derivative equation.
Higher-order spherical approximations of the RTE are needed to accurately model the light propagation through low-scattering regions, making this approach unsuitable within an image reconstruction scheme for complex geometries, due to the high consumption of memory and time [43].
A more convenient approach can be used in the application field we are studying, where we need to model light in a domain which is derived from an anatomical MRI. In this case, it is possible to segment the head into diffusive regions and non-diffusive region. We can then use two separate set of equations to model light transport in scattering regions (diffusion model), and in clear region (radiosity model).

In fact, in regions filled with cerebrospinal fluid (CSF) the diffusion approximation does not hold since the condition $\mu_s'(\mathbf{r}, f) \gg \mu_a(\mathbf{r}, f)$ is not verified. On the other hand, these regions can be satisfyingly modeled by radiosity methods [12].
Radiosity algorithms derive from ray-tracing optics, and are basically methods for computing the amount of light leaving a certain portion of a surface (in our case the surface between one voxel of CSF and one of a scattering medium) reaching a different portion of the surface.

Let $\mathbf{m}$ and $\mathbf{m}'$ be points on the CSF boundary while $\mathbf{n}(\mathbf{m})$ and $\mathbf{n}(\mathbf{m}')$ are their respective inward directed surface normals, as shown in the Fig. 2.4. As derived in [6], the equivalent isotropic source at point $\mathbf{m}$ due to the contribution of the outward photon current at every boundary point $\mathbf{m}'$ is given by:

$$\mathbf{\Gamma}(\mathbf{m}, f) = \int_{\partial\Omega_{csf}} h_{\mathbf{m},\mathbf{m}'}\, F(\mathbf{m}, \mathbf{m}') e^{-\mu_{a_{csf}} d_{\mathbf{m},\mathbf{m}'}} \frac{\Phi(\mathbf{m}', f)}{2R_{csf}}\, d\mathbf{m}' \tag{2.16}$$

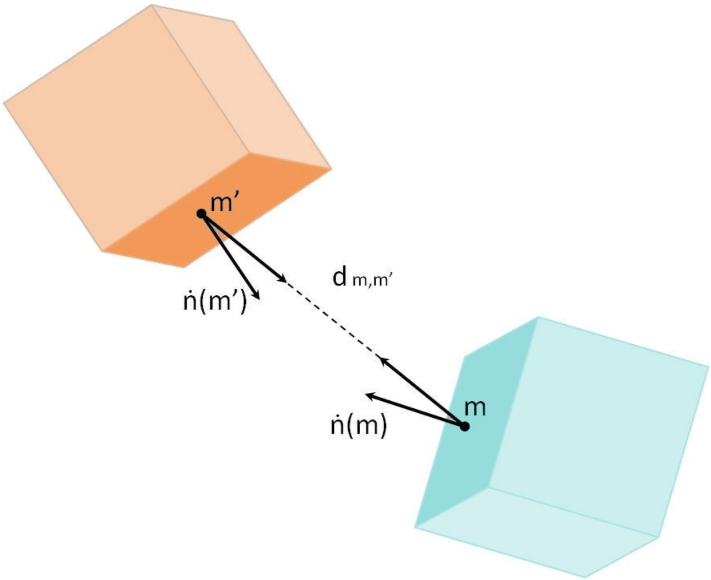


Figure 2.4: Form factor computation between CSF boundary points $\mathbf{m}$ and $\mathbf{m}'$: $\mathbf{n}(\mathbf{m})$ and $\mathbf{n}(\mathbf{m}')$ are the surface normals, $d_{\mathbf{m},\mathbf{m}'}$ is the distance between the points and $\mathbf{s}_{\mathbf{m},\mathbf{m}'}$ is the unit vector in the direction from $\mathbf{m}$ to $\mathbf{m}'$.

where Ω_{csf} is the CSF volume and $\partial\Omega_{csf}$ represents its surface, $d_{\mathbf{m},\mathbf{m}'}$ is the distance between $\mathbf{m}$ and $\mathbf{m}'$, $\mu_{a_{csf}}$ is the absorption coefficient of the CSF and $h_{\mathbf{m},\mathbf{m}'}$ is the visibility function, which is one if $\mathbf{m}$ and $\mathbf{m}'$ are mutually visible, and zero otherwise. R_{csf} incorporates the mismatches between the refractive indices within the diffusive domain and within the CSF while $F(\mathbf{m},\mathbf{m}')$ is called the *form factor* between $\mathbf{m}$ and $\mathbf{m}'$ and represents the fraction of energy which leaves $\mathbf{m}$ and reaches $\mathbf{m}'$. In particular:

$$F(\mathbf{m},\mathbf{m}') = \frac{\mathbf{s}_{\mathbf{m},\mathbf{m}'} \cdot \mathbf{n}(\mathbf{m}')\ \mathbf{s}_{\mathbf{m},\mathbf{m}'} \cdot \mathbf{n}(\mathbf{m})}{\pi d^2_{\mathbf{m},\mathbf{m}'}} \tag{2.17}$$

where $\mathbf{s}_{\mathbf{m},\mathbf{m}'}$ is the unit vector on the direction from $\mathbf{m}$ to $\mathbf{m}'$. The effect of these sources is to introduce a non-zero term on the right hand side of Eq. 2.15, leading to the following RBC for every point $\mathbf{m}$ on the boundary of the CSF region:

$$\Phi(\mathbf{m},f) + 2R_{csf}\kappa(\mathbf{m},f)\mathbf{n}(\mathbf{m}) \cdot \nabla\Phi(\mathbf{m},f) = \Gamma(\mathbf{m},f) \tag{2.18}$$

where $\Gamma(\mathbf{m},f)$ is the amount of light which irradiates point $\mathbf{m}$. Hybrid radiosity-diffusion models combine the results of a diffusion equation in highly-scattering regions with radiosity solution, which specifically models light propagation through clear regions following the principle of ray optics. This approach, if compared to the simple diffusion model, drastically improves the accuracy of the light distribution in scattering domains with clear regions and, at the same time, allows photon distribution to be calculated efficiently to an accuracy comparable with that of RTE Monte Carlo methods [10][11].

2.5 Conclusions

The objective of this chapter is to give the theoretical basis of light propagation to understand the numerical implementation of the DOT forward problem solver proposed in Chapter 3. It provides a review of the state of the art of DOT light propagation models, explaining the reasoning leading to the choice of a Radiosity-Diffusion approach.

Chapter 3

DOT Forward Problem : Numerical Implementation

As illustrated in Chapter 1, with respect to techniques such as fMRI or PET, DOT has the advantages of low cost, high portability and good temporal resolution; on the other hand, spatial resolution is rather limited, in particular due to the presence of various layers of tissue through which light needs to propagate before reaching the cortical brain regions [2]. Moreover, light propagation is heavily influenced by the complex shapes which characterize the interfaces between tissues; hence, the availability of anatomical information deriving from an MRI scan of the subject can lead to more accurate domain-specific solvers [3], improving the spatial resolution and accuracy of the imaging technique.

Furthermore, an inverse problem solution requires multiple computations of the forward problem solution, calling for accurate and time-economical forward solvers. Several methods (based on light propagation equations introduced in Chapter 2) have been developed in order to provide an appropriate trade-off between these two requirements, ranging from accurate, highly intensive transport-based approaches to diffusive approximations, through hybrid radiosity-diffusion methods.

Solving the RTE in non-homogeneous volumes having complex shapes as required in optical tomography is an extremely computationally-intensive task if solved numerically or statistically. Diffusion equation solvers offer lower computational times than RTE solvers, at the cost of reduced modeling accuracy. The chief disadvantage in using a diffusion equation in a realistic human head domain is that, in a low-scattering medium such as the CSF which fills the central ventricles and the subarachnoid space, it provides inaccurate results.

Hybrid radiosity-diffusion models combine the results of a diffusion equation in highly-scattering regions with radiosity solution, which specifically models light propagation through clear regions following the principle of ray optics. This approach, if compared to the simple diffusion model, drastically improves the accuracy of the light distribution in scattering domains with clear regions and, at the same time, allows photon distribution to be calculated efficiently with an accuracy comparable to RTE-based methods.

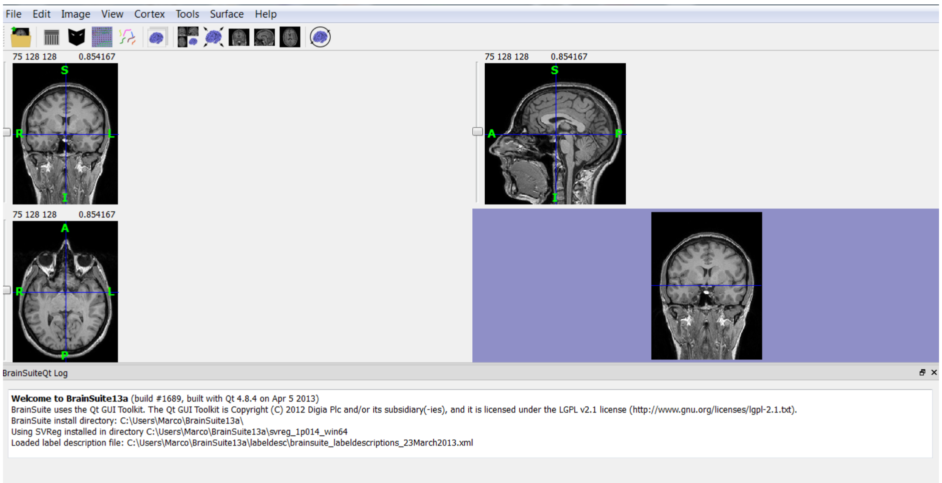
This chapter focuses on the numerical implementation of a Radiosity Diffusion solver based on the numerical integration between the diffusion equation and the radiosity equation and it is the core of this thesis. It represents an original contribution which updates the state of the art of DOT forward problem solvers providing a solver able to implement an RD model able to manage the geometrical complexity of the head volume directly imported from a full resolution structural MRI.

When a numerical implementation is employed, a regular grid of cubic voxels is generally used to discretize the physical domain, therefore it perfectly fits our final application where the domain is a human head, for which the best structural information available are those obtained from a MRI scan (Sec. 3.1), which is inherently discretized in regular voxels. Optical properties of biological tissues are investigated in Sec. 3.2 while the numerical formulation of the diffusion and radiosity equations are analysed respectively in Sec. 3.3.1 and Sec. 3.3.2, and finally combined in Sec. 3.4.3 leading to the numerical formulation of the proposed RD solver. The accuracy and performance of the proposed RD numerical solver are examined in Sec. 3.6 and Sec. 3.7, while in Sec. 3.8 a complete overview of the work-flow on a realistic human head is given.

3.1 Brain Segmentation

MRI images are segmented and classified to obtain a realistic volume composed of five different types of tissue, namely scalp, skull, CSF, gray matter and white matter. To this end, we adopted “BrainSuite” [56], an automated

open-source cortical extraction tool.
Structural MRI data in NifTi or DICOM format is imported (see Fig. 3.1) and segmentation is performed in a semi-automated way, with some parameters which can be modified to improve segmentation quality (see Fig. 3.2).



```
Welcome to BrainSuite13a (build #1689, built with Qt 4.8.4 on Apr 5 2013)
BrainSuite uses the Qt GUI Toolkit. The Qt GUI Toolkit is Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies), and it is licensed under the LGPL v2.1 license (http://www.gnu.org/licenses/lgpl-2.1.txt).
BrainSuite install directory: C:\Users\Marco\BrainSuite13a\
Using SVReg installed in directory C:\Users\Marco\BrainSuite13a\svreg_1p014_win64
Loaded label description file: C:\Users\Marco\BrainSuite13a\labeldesc\brainsuite_labeldescriptions_23March2013.xml
```

Figure 3.1: Structural MRI data in NifTi format.

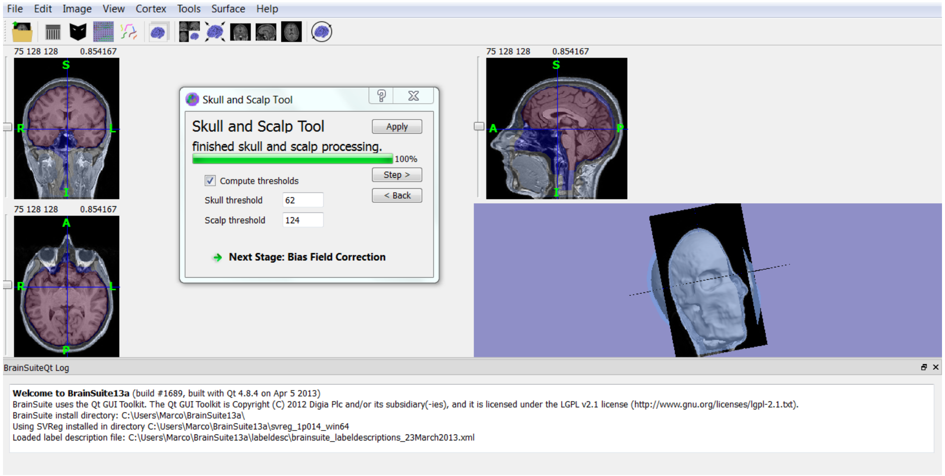
After segmentation, data is exported in ANALYZE format and imported and visualized in MATLAB through an ad-hoc routine. This operation results in classification of the MRI image voxels under the five aforementioned tissues, maintaining the full resolution provided by MRI scans. If different or more detailed segmentations were used, this would not introduce overhead in the successive forward problem definition, nor would it require any new mesh generation.

As an example, sagittal, coronal and transverse section showed in Fig. 3.3 are obtained in the central section of the segmented MRI.

3.2 Optical Properties of Tissues

The main properties which define the optical characteristics of a medium can be considered to be absorption coefficient μ_a, reduced scattering coefficient μ_s' (see Eq. 2.11) and refractive index n. The latter is usually considered to be homogeneous in human head's tissues and will therefore be generally left out from the following discussions [57].

Optical properties of brain tissues for DOT wavelengths are derived from



Skull and Scalp Tool

finished skull and scalp processing.

100%

☑ Compute thresholds

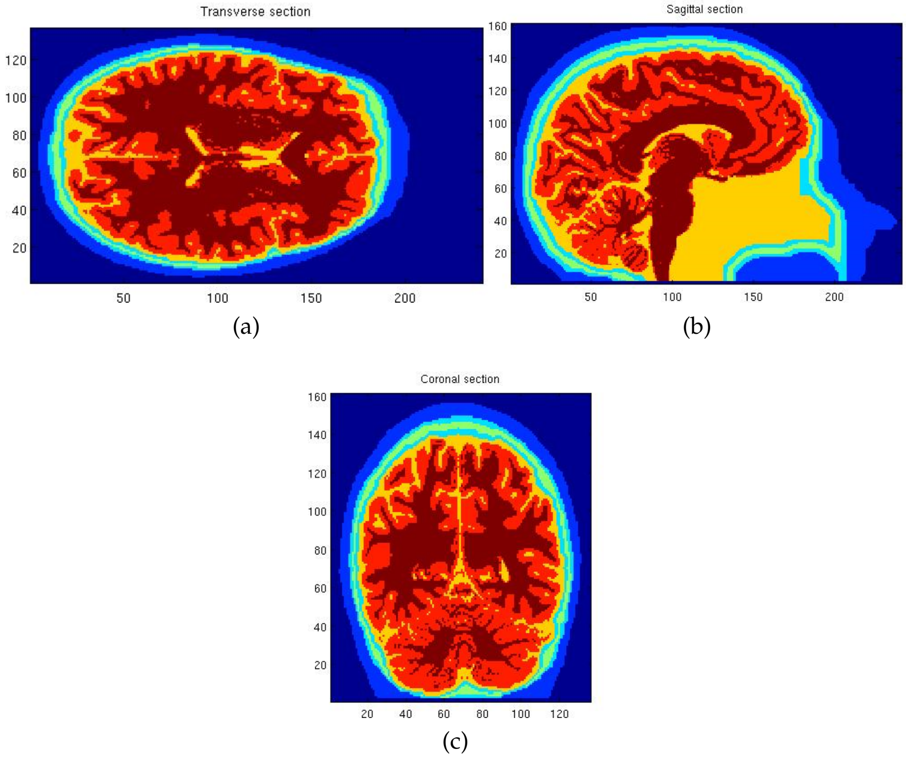
Skull threshold 62

Scalp threshold 124

Next Stage: Bias Field Correction

```
Welcome to BrainSuite13a (build #1689, built with Qt 4.8.4 on Apr 5 2013)
BrainSuite uses the Qt GUI Toolkit. The Qt GUI Toolkit is Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies), and it is licensed under the LGPL v2.1 license (http://www.gnu.org/licenses/lgpl-2.1.txt).
BrainSuite install directory: C:\Users\Marco\BrainSuite13a\
Using SVReg installed in directory C:\Users\Marco\BrainSuite13a\svreg_1p014_win64
Loaded label description file: C:\Users\Marco\BrainSuite13a\labeldesc\brainsuite_labeldescriptions_23March2013.xml
```

Figure 3.2: Segmentation of the structural MRI.



(a) (b)

(c)

Figure 3.3: Transverse, coronal and sagittal slices in the central section of the segmented MRI.

the mean of the optical properties listed in Okada *et al*. [58], Hollis [59], Boas *et al*. [60], Firbank *et al*. [61] , Branco *et al*. [62].

Table 3.1 and Table 3.2 show the absorption and the reduced scattering coefficients of the tissues considered in function of the light wavelength. These data show that absorption and reduced scattering are notably wavelength-dependent for all tissue types. As shown in Table 3.1, due to melanin in the epidermis layer, absorption coefficient of the skin is very high, especially near the ultraviolet region. Furthermore, Simpson *et al.* [63] concluded that the transmission of light through the skin is highly dependent on the pigmentation. Moreover, it should be pointed out that all these data can be influenced by pathologic situations which modify blood perfusion. In Table 3.1 and 3.2 Caucasian skin is considered.

Wavelength [nm]	600	700	800	900	1000
Skin	0.069	0.048	0.043	0.033	0.027
Cranial bone	-	0.023	0.011	0.015	0.022
CSF	-	-	-	-	-
Gray Matter	0.022	0.016	0.020	0.032	0.049
White Matter	0.080	0.075	0.087	0.101	0.116

Table 3.1: Absorption coefficient μ_a [mm^{-1}] of biological head tissues in function of light wavelength (mean of [58], [59], [60], [61] and [62] values).

Wavelength [nm]	600	700	800	900	1000
Skin	2.180	1.670	1.400	1.570	1.680
Cranial bone	-	2.210	1.948	1.803	1.710
CSF	-	-	-	-	-
Gray Matter	0.984	0.819	0.764	0.645	0.559
White Matter	6.615	5.454	4.616	3.928	3.498

Table 3.2: Reduced scattering coefficient μ_s' [mm$^-1$] of biological head tissues in function of light wavelength (mean of [58], [59], [60], [61] and [62] values).

Since the DOT portable system which has been developed inside WP3 of the HIGH-PROFILE project is based on bi-color LEDs at two different wavelengths (735 and 850 nm), we are mainly interested in absorption and reduced scattering at these particular wavelengths. These values of absorption and scattering are obtained by interpolation and are reported in Table 3.3 and 3.4. Brain tissues have a high anisotropy g factor (defined in Eq. 2.12) which means that the scatter is very forward-peaked [64], for our purposes we adopt $g = 0.9$ which is the same value used in [65].

Wavelength [nm]	735	850
Skin	0.047	0.037
Cranial bone	0.018	0.013
CSF	0.0022	0.0022
Gray Matter	0.017	0.026
White Matter	0.079	0.094

Table 3.3: Absorption coefficients of biological head tissues for 735 and 850 nm light wavelength (obtained by interpolation of Table 3.1 absorption coefficients).

Wavelength [nm]	735	850
Skin	1.590	1.490
Cranial bone	2.120	1.875
CSF	-	-
Gray Matter	0.801	0.705
White Matter	5.175	4.272

Table 3.4: Scattering coefficients of biological head tissues for 735 and 850 nm light wavelength (obtained by interpolation of Table 3.2 reduced scattering coefficients).

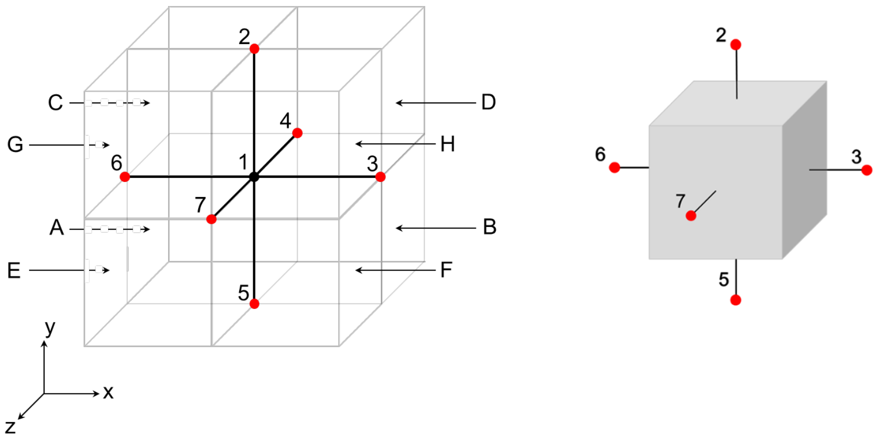
3.3 Numerical Problem Formulation

3.3.1 Numerical Problem Formulation - Diffusive model

The implemented numerical solver is based on the Finite Volume Method (FVM) formulation of Eq. 2.8. For FVM, a regular grid of cubic voxels is generally used to discretize the physical domain, so it perfectly suits our application where the domain is defined by an MRI scan which is inherently discretized into regular voxels (see Fig. 3.4 (a)). Scattering and absorption coefficients are assumed to be piecewise constant over each voxel volume, whereas photon density is computed on voxel vertices (nodes). As shown in Fig. 3.4 (b), discretization of the diffusion equation is carried out by integrating it over a control volume centered on the node of interest, where the control volume centered on node 1 is illustrated.

Applying the divergence theorem to Eq. 2.8 we obtain:

$$-\int_S \kappa(\mathbf{r},f)\nabla\Phi(\mathbf{r},f)\cdot dS = = -\int_V \left(\mu_a(\mathbf{r},f)+\frac{\imath f}{c}\right)\Phi(\mathbf{r},f)dV+\int_V q_0(\mathbf{r},f)dV \tag{3.1}$$



(a) FVM grid (b) FVM control volume

Figure 3.4: FVM discretization: nodes are represented by numbers while voxels are indicated by capital letters.

where V is the control volume that surrounds the central node and S is its surface.

The surface integral $\int_S \kappa(\mathbf{r}, f)\nabla\Phi(\mathbf{r}, f) \cdot dS$ can be physically interpreted as the photon current flowing out through the faces of the control volume. The terms $\int_V \mu_a(\mathbf{r}, f)\Phi(\mathbf{r}, f)\ dV$ represents the number of photons absorbed in the control volume while $\int_V q_0(\mathbf{r}, f)\ dV$ is the total photon density originating in the control volume by the isotropic source.

FVM formulation is based on flux conservation through the six surfaces of the control volume. To this end, by using a finite element scheme and supposing the photon density piecewise linear, it is possible to define the flux through each face of the control cube in function of the neighbouring nodal values of photon density.

Eq. 3.1 is then discretized on every control voxel, as has already been done in the literature for the Poisson's equation [66] [67]. With respect to that, the diffusion equation has an additional absorption term $\int_V \mu_a(\mathbf{r}, f)\Phi(\mathbf{r}, f) dV$. The resulting equation for spectral irradiance $\Phi_1(f)$ in node 1 of Fig. 3.4 is given by:

$$\Phi_1(f) = \sum_{i=1}^{7} A_i(f)\Phi_i(f) \tag{3.2}$$

where the coefficients A_i are:

$$\begin{aligned}
A_2(f) &= -\tfrac{h}{4}(\kappa_C + \kappa_D + \kappa_G + \kappa_H),\\
A_3(f) &= -\tfrac{h}{4}(\kappa_B + \kappa_D + \kappa_F + \kappa_H),\\
A_4(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_B + \kappa_C + \kappa_D),\\
A_5(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_B + \kappa_E + \kappa_F),\\
A_6(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_C + \kappa_E + \kappa_G),\\
A_7(f) &= -\tfrac{h}{4}(\kappa_E + \kappa_F + \kappa_G + \kappa_H),\\
A_1(f) &= -\textstyle\sum_{i=2}^{7} A_i(f) + \tfrac{1}{8}h^3(\mu_{a_A} + \ldots + \mu_{a_H} + \tfrac{\imath f}{c})
\end{aligned} \tag{3.3}$$

and h is the length of the control volume's edge, while $\kappa_A \ldots \kappa_H$ and $\mu_{a_A} \ldots \mu_{a_H}$ represent respectively the diffusion coefficients and the absorption coefficients of voxels $A \ldots H$ centered in node 1, as represented in Fig. 3.4 (a).

For the sake of readability, the frequency-dependency of optical coefficients is not explicitly indicated. It is important to notice that coefficient A_1 depends directly on the frequency f while A_2 to A_7 relies only on the frequency-dependent diffusion coefficients.

Combining these equations for every voxel, the FVM formulation results in a linear system of equations:

$$\mathbf{A}(f)\boldsymbol{\Phi}(f) = \mathbf{b}(f) \tag{3.4}$$

where $\mathbf{A}(f)$ is the system matrix, $\boldsymbol{\Phi}(f)$ is the vector containing spectral irradiances $\Phi_i(f)$ at every node, and $\mathbf{b}(f)$ is a column vector containing nodal isotropic sources.

Each equation in the above linear system represents the discretization of 3.1 on one of the N elements of the cubic grid. Since each voxel has six neighbour nodes, fluxes through the surfaces surrounding the control voxel are expressed as a function of the photon densities on the 6 neighbours. This means that every row of A has only 7 non-zero elements as we can observe in the representation of the system matrix shown in Fig. 3.5.

Furthermore, the system matrix $\mathbf{A}$ is multi-diagonal, symmetric and positive definite, making the linear system suitable for solving by highly parallel and fast and iterative techniques. The derivation is valid for nodes which are surrounded by voxels of diffusive tissues. For those bordering either

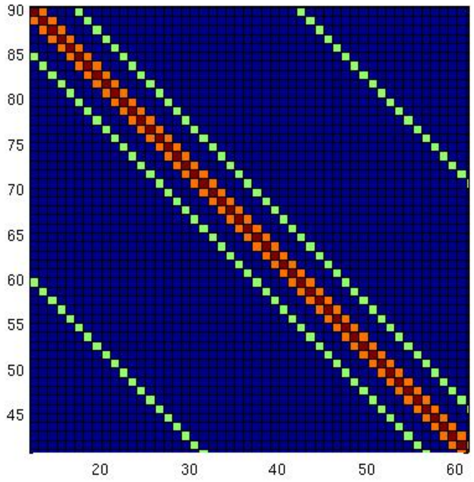


Figure 3.5: System matrix resulting from FVM discretization. Since each voxel has six neighbour nodes $A(f)$ is 7-diagonal.

on air ($\partial\Omega$) or CSF ($\partial\Omega_{csf}$), the inclusion of boundary conditions involves modifying the relevant elements of the system matrix.

In order to set Dirichlet boundary condition in the voxel i, we simply need to force a certain Φ_i to the value b_i. This simply requires to set the corresponding row of matrix A to:

$$A(i,j) = \begin{cases} 1\ , \ j = i \\ 0\ , \ i \neq j. \end{cases} \tag{3.5}$$

A different approach is required when imposing Robin boundary condition. In order to keep derivation simple, we may consider, as an example, the case where only voxel H is low-scattering tissue. Derivation can easily be extended to each possible combination. The shape of the control volume needs to be modified as shown in Fig. 3.6 where the red surfaces represent the intersection between the volumes of the control cube centered in node 1 and of the low-scattering voxel H represented in light red.

The resulting control volume is contained within the red and the blue surfaces defined by solid black lines and its surface can be defined as $S_r \cup S_b$, where S_r is the union of the red patches while S_b is the union of the blue patches. In particular, the left-hand side in Eq. 3.1 becomes the sum of two

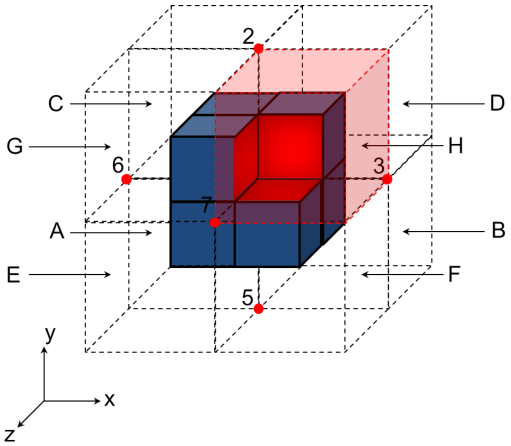


Figure 3.6: Modified control volume due to the presence of a low-scattering voxel (light red voxel H). The red area represents the surface of the control volume that lies on the boundary of the domain.

surface integrals as shown in Eq. 3.6:

$$-\int_{S_r} \kappa(\mathbf{r}, f)\nabla\Phi(\mathbf{r}, f) \cdot dS - \int_{S_b} \kappa(\mathbf{r}, f)\nabla\Phi(\mathbf{r}, f) \cdot dS. \tag{3.6}$$

The procedure for computation of all volume integrals of Eq. 3.1 and also of the surface integral over S_b is the same, taking into account the volume and surface reduction.

To compute the integral over S_b, let us assume $\Phi(\boldsymbol{\xi}, f) = \Phi_1(f)$. In this way, Eq. 2.15 becomes:

$$\kappa(\boldsymbol{\xi}, f)\nabla\Phi(\boldsymbol{\xi}, f) \cdot \mathbf{n}(\boldsymbol{\xi}) = -\frac{\Phi_1(f)}{2R} \tag{3.7}$$

and the surface integral over S_b in Eq. 3.6 becomes:

$$\int_{S_b} \kappa(\xi, f)\nabla\Phi(\xi, f) \cdot \mathbf{n}(\xi)\, dS = -\frac{N_f (h/2)^2}{2R}\Phi_1(f) \tag{3.8}$$

where N_f is the number of faces of area $(h/2)^2$ composing S_b.

For example, in order to set a Robin boundary condition in the central node $\Phi_1(f)$, we need to modify the coefficient $A_i(f)$ shown in Eq. 3.3 in the following way:

$$\begin{aligned}
A_2(f) &= -\tfrac{h}{4}(\kappa_C + \kappa_D + \kappa_G), \\
A_3(f) &= -\tfrac{h}{4}(\kappa_B + \kappa_D + \kappa_F), \\
A_4(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_B + \kappa_C + \kappa_D), \\
A_5(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_B + \kappa_E + \kappa_F), \\
A_6(f) &= -\tfrac{h}{4}(\kappa_A + \kappa_C + \kappa_E + \kappa_G), \\
A_7(f) &= -\tfrac{h}{4}(\kappa_E + \kappa_F + \kappa_G), \\
A_1(f) &= -\textstyle\sum_{i=2}^{7} A_i(f) + \tfrac{1}{8}h^3(\mu_{a_A} + \ldots + \mu_{a_G} + \tfrac{\imath f}{c}) + \tfrac{3}{8R}h^2.
\end{aligned} \tag{3.9}$$

The optical properties of voxel H disappear since it is an external voxel, and the contribution $\frac{3}{8R}h^2$ is given by Eq. 3.8 with a number of dark gray patches equal to 3 (see Fig. 3.6). Note that inclusion of RBC does not affect the matrix characteristics of multi-diagonality and symmetry.

FVM formulation of the diffusion equation leads therefore to a linear system, characterized by a multi-diagonal and symmetric matrix which is preserved by including the different boundary conditions. Furthermore, FVM is flux continuous, meaning that photon density continuity is satisfied over the entire computational domain.

3.3.2 Numerical Problem Formulation - Radiosity model

The isotropic source distribution for CSF boundary nodes leads to the boundary condition of Eq. 2.18, where $\mathbf{\Gamma}(f)$ is the vector containing the isotropic sources which represent the amount of light which irradiates every CSF boundary node. In order to evaluate $\mathbf{\Gamma}(f)$ a balance equation can be obtained as was done in [9] and [55] to include the reflection in the Robin boundary condition. The resulting equation is:

$$\mathbf{\Phi}(f) + 2R_{csf}\kappa(f)\mathbf{n}\cdot\nabla\mathbf{\Phi}(f) = \mathbf{F_a}(f)\Big(\mathbf{\Phi}(f) - 2R_{csf}\kappa(f)\mathbf{n}\cdot\nabla\mathbf{\Phi}(f)\Big) \tag{3.10}$$

for every CSF boundary node, where $\mathbf{F_a}(f) = \mathbf{F}e^{-\mu_{a_{csf}} d_{\mathbf{m},\mathbf{m}'}}$. This equation states that the total amount of light which irradiates the CSF surface (left term of Eq. 3.10) is equal to the amount of light which leaves the surface multiplied by the form factor matrix $\mathbf{F_a}$ which takes into account the absorption in the CSF (right term of Eq. 3.10). Comparison between the right terms of Eq. 2.18 and Eq. 3.10 leads to:

$$\mathbf{\Gamma}(f) = \mathbf{F_a}(f)\Big(\mathbf{\Phi}(f) - 2R_{csf}\kappa(f)\mathbf{n}\cdot\nabla\mathbf{\Phi}(f)\Big). \tag{3.11}$$

3.4 Numerical Solution

3.4.1 Numerical solution of diffusion equation

The literature shows that the Conjugate Gradient Method (BiConjugate Gradient Method (BiCG) for complex time-dependent equations) is an excellent candidate for solving the linear system presented and is adopted in several works for similar problems [67]. For our application we use the efficient and parallel BiCG and CG solvers implemented on multiple GPUs and CPUs using the CUDA drivers and software design kit for NVIDIA GPUs [68] and CPU multithreading using the free Linux library of OpenMP [69]. A dedicated CPU thread is spawned using OpenMP to handle each GPU. Synchronization between the data streams is necessary on several occasions to ensure data consistency, as discussed in [70].

3.4.2 Form factor computation

As discussed in the previous section, in order to take into account the $\mathbf{\Gamma}(f)$ contribution defined in Eq. 2.18, an equivalent isotropic source needs to be computed by means of Eq. 2.16.
This involves three basic steps: extracting the boundary voxels of the CSF region, determining the mutual visibility between them ($h_{\mathbf{m},\mathbf{m}'}$) and evaluating form factors ($F(\mathbf{m},\mathbf{m}')$).

First, boundary voxels are determined through an ad-hoc routine which imports the segmented MRI. CSF voxels are marked by a non-zero label and all the others voxels by a zero label. If we scan all the domain's voxels and compute, for each one, the differences in the six directions between voxel labels, and if a negative result occurs, it means that we are centered in a brain or skull voxel adjacent to a CSF voxel. The coordinates of these voxels, their orthocenters and the normal vectors to the boundary faces are exported in binary files to reconstruct the problem geometry for visibility computation in a ray-tracing engine environment.

Second, one proceeds to evaluate the mutual visibility between boundary voxels, which is the most computationally intensive task.
Finally, one has to evaluate and store the form factors between boundary voxels.
Due to the segmentation of the MRI into regular and cubic voxels, surfaces are obtained from composition of identical and axis-oriented voxels (as shown in Fig. 3.6). Since the geometry of the CSF surface is highly irregular and composed of a high number of patches ($\approx 4 \cdot 10^5$), the computational load is massive and a parallel engine is needed to drastically reduce the elaboration time.

Nvidia Optix 3.0.0 [71] is an interesting scalable and parallel framework for building ray-tracing applications which is optimized to run on Nvidia GPUs. The framework is mainly designed for graphical rendering; however we can exploit its high efficiency to the problem under examination. It is composed of two symbiotic parts:

1. a HOST based API that defines ray-tracing data structures,
2. a CUDA C-based programming system that can produce and manage rays and surfaces.

Optix engine avoids specification of ray tracing engine behaviours and instead provides mechanism to execute user-provided CUDA C code to implement shading, camera models and color representations. Developers provide information about rays and surfaces in the form of CUDA C-based functions, and ray tracing structures are created by APIs. The main objects in the system are:

- **Context**: instance of a running Optix engine;
- **Variable**: name used to pass data from C to Optix program;
- **Buffer**: a multidimensional array that can be bound to a variable;
- **Geometry**: one or more primitive that a ray can be intersected with;
- **Material**: a set of program executed when a ray intersects with a primitive;

- **Acceleration**: an acceleration structure object that define a hierarchy node.

After these objects are defined, it is possible to invoke the Optix parallel ray tracing engine. In general, the aim of ray-tracing is to generate a 2D image given the 3D geometry of a scene, the setup of lights and the viewpoint coordinates. The 2D image is realized by tracing the path of light rays through the 3D virtual scene and simulating the effects of its encounters with the virtual objects composing the scene [72]. The basic concept of ray tracing is represented in Fig. 3.7.

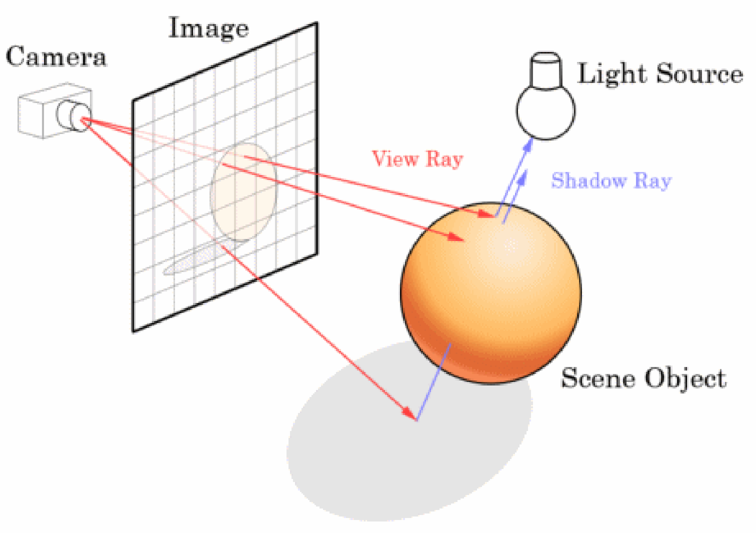


Figure 3.7: Ray tracing principle.

The standard Optix ray tracing steps for rendering are:

1. Launch a generic ray pack with eye-pixel direction by parallel processes;
2. Rays are represented by a 2D index that is also the index of the parallel process associated;
3. Intersections between camera rays and surfaces of the scene are parallel computed;
4. If an intersection occurs, a shadow ray is computed to represent occlusion and a radiance ray is computed to represent the final color of intersection point.

Since our purpose is to determine form factors between CSF boundary surfaces involving light occlusion evaluation, a custom-built ray-tracer was implemented based on the Optix engine. Its flow chart is shown in Fig. 3.8. First of all, the coordinates of boundary voxels are imported into the Optix environment to reconstruct the whole geometry of the CSF surface. The geometry reconstruction step is based on creating bounding boxes that coincide with voxels.

The bounding box is a fundamental structure of a ray-tracing engine, defining the smallest parallel-axes cube that contains a surface. This kind of structure is fundamental if one is to achieve any significant speed-up, because it creates a hierarchy that enables unnecessary computations to be avoided. Since every voxel is cubic and aligned with orthogonal axes, voxels are mapped directly onto bounding boxes and creation of the accelerated structure is instantaneous.

Furthermore, in order to speed-up the domain construction process, a single bounding box is created, centered in the origin. From this, all the other bounding boxes are created by defining a single transformation matrix through the ad-hoc Optix function; this approach was found to be significantly faster than creating every box singly.

To treat the problem as a light occlusion determination, the concept is to consider a light irradiating from the center of a single face to the orthocenters of all other CSF boundary faces, and using the optimized visibility evaluation of the Optix engine to compute the light occlusion as shown in Fig. 3.9. In order to do this, the coordinates of the orthocenters of the faces of each

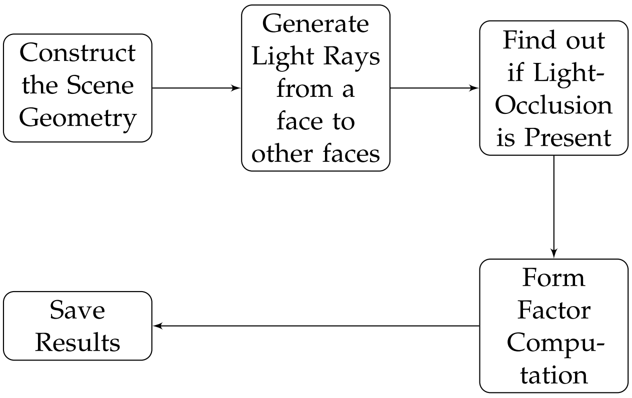
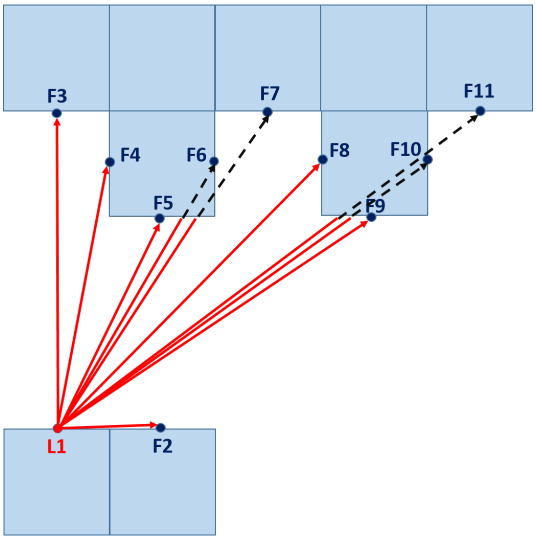
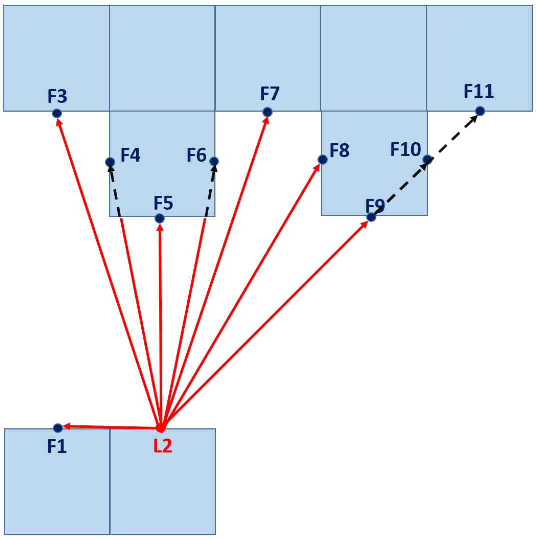


Figure 3.8: Customized Ray-Tracing Flow Chart

boundary voxel are loaded from an ad-hoc binary file. Each destination face is identified by a 1D index (F# in Fig. 3.9) which identifies the parallel process associated with each face's orthocenter for which visibility is to be computed. While different lights are serially processed, every destination point is associated with a parallel process, so that the visibility from one light to every destination face can be evaluated in a highly parallel way.



(a) First step



(b) Second step

Figure 3.9: Visibility evaluation: first and second step. In the first step the visibility from first light to other faces (F2. . .F11) is evaluated, while in the second step the visibility is computed from the second face L2 to other faces (F1 and F3. . .F11).

As can be observed from Eq. 2.17, computation of form factor $F(\mathbf{m}, \mathbf{m}')$ requires us to estimate the distance between the boundary points $d_{\mathbf{m},\mathbf{m}'}$ considered and the scalar products between vectors normal to the source and destination surfaces. Both light-point distances and scalar products are computed through the built-in Optix functions, after loading all the normal unit vectors. Finally, form factors are computed, multiplied by the term $e^{-\mu_{a_{csf}} d_{\mathbf{m},\mathbf{m}'}}$ to include CSF absorption, and saved in a compact binary file via an ad-hoc protocol, to allow parallel storage with optimal memory occupation.
In order to clarify the behavior of the protocol, let us consider the example of visibility evaluation represented in Fig. 3.10. For every serially-processed light the visibility algorithm provides a vector containing 1 in the index corresponding to the visible face, and 0 otherwise. A simple method to save the visibility data could be to construct a full matrix in the same form as the table shown in Fig. 3.10, where m is the total number of lights. The form illustrated in Fig. 3.10 is obtained from the product of the visibility result by the index of the processed light.

	face 1	face 2	face 3	…	face m
light 1	1	0	0	…	1
light 2	0	2	2	…	0
light 3	0	3	3	…	3
⋮	⋮	⋮	⋮	⋮	⋮
light m	m	0	m	…	m

Figure 3.10: Data storage using a full matrix with face and light indices.

A more convenient approach is to save only the non-zero results of each light visibility evaluation (Fig. 3.11), leading to an upward shift of every non-zero value of the full matrix of Fig. 3.10. For compact data storage one

	face 1	face 2	face 3	…	face m
	1	2	2		1
	⋮	3	3	⋮	3
		⋮	⋮		⋮
	m		m		m

Figure 3.11: Compact data storage using the parallel counter.

needs a pointer to the first empty location where it is possible to write data. This counter is realized as a parallel vector and is incremented by the result of visibility evaluation. For the example shown in Fig. 3.10, the progression is represented in Fig. 3.12. When using this kind of implementation, after evaluation of the visibility for every light composing the CSF boundary surface, the counter will contain the number of faces visible from every single light. One should note that both the counter and the visibility results are parallel vectors and these two vectors have been summed in a parallel way. To allow for parallel storage of visibility results, an empty matrix is first initialized. For every column of the matrix, the counter obtained represents a pointer to the first empty location where it is possible to write data. If the face corresponding to a specific column is visible from the processed light, the index of this light is saved in the position provided by the parallel counter and the counter is incremented; otherwise no increment occurs and in the following step the pointer will indicate the same free cell.

light 1	**visibility**	1	0	0	...	1
	counter	1	0	0	...	1
	↓					
light 2	**visibility**	0	1	1	...	0
	counter	1	1	1	...	1
	↓					
light 3	**visibility**	0	1	1	...	1
	counter	1	2	2	...	2
	↓					
⋮	⋮					
light m	**visibility**	1	0	1	...	1
	counter	k_1	k_2	k_3	...	k_m

Figure 3.12: Time evolution of the counter vector, incremented using a visibility parallel vector.

A concatenation of matrix columns is finally achieved, giving the definitive compact form. The same procedure is also used for compact storage of the form factor values. As a result, for every light we use compact binary files to save all the information needed to build a sparse form factor matrix.

Once the compact binary files containing form factors have been created, we need to rearrange them in the matrix form $\mathbf{F_a}(f)$ of Eq. 2.18 via an ad-hoc Matlab tool.

First, the form factors saved in the compact vector of the CSF boundary faces are loaded into the Matlab environment.

Second, the form factors of each boundary node are obtained by combining the form factors of its neighboring boundary faces.

The form factor computation time for a full MRI is approximately 1.5 hours using the High Performance Computing (HPC) platform of Table 3.5, taking into account the reciprocal visibility of approximately 393K points for a final compact visibility output file of about 2 GB instead of 5 TB required by the full matrix approach. It should be pointed out that this task is computed just once and is performed only once before DOT data acquisition.

CPUs	2x Intel Xeon E5-2650 v2 @ 2.60GHz
Graphic Cards	4x Geforce GTX Titan Black
RAM	64 GB

Table 3.5: Hardware specifications of the HPC Platform.

3.4.3 Numerical solution of the hybrid RD problem

Modeling light propagation in head tissues using the hybrid diffusion-radiosity method requires that we combine the results obtained from solving diffusion and radiosity equations in the appropriate regions.

The most straightforward way to do so would be to directly integrate the results of form factor computation into the structure of matrix $\mathbf{A}(f)$, basically adding the contributions from non-neighboring voxels due to light propagation in clear regions. Unfortunately this solution perturbs the matrix structure, resulting in a loss of multi-diagonality and a largely sparse matrix. Since the efficiency of the diffusion equation parallel solver relies heavily on the regular structure of the matrix, both for computation and for data storage, this solution is to be considered infeasible in the proposed framework.

One different approach is based on an iterative solver where solution of the diffusion and radiosity portions of the problem is performed in sequence as shown in Fig. 3.13 and the process is repeated till the photon distribution on the boundaries is consistent with both.

A diffusion equation is used to calculate the exitance on diffusive voxels

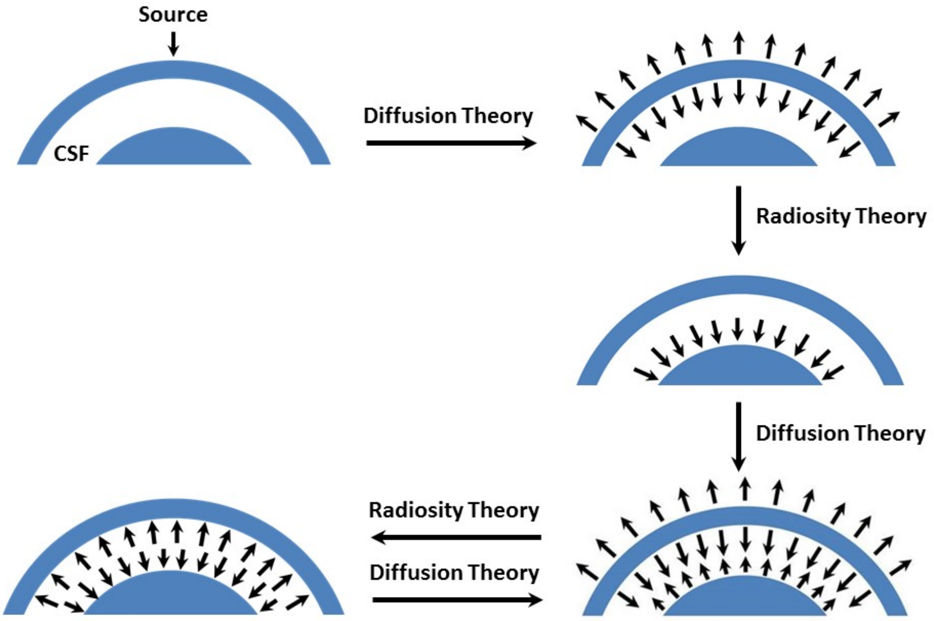


Figure 3.13: Radiosity-Diffusion Iterative solver principle.

bordering on the CSF region. The radiosity equation is then used to calculate the irradiance at each voxel on the CSF surface, due to the emission of light from other voxels. Each of these acts as a source for a further diffusion theory calculation of the light distribution in the scattering regions of the domain. This process is iterated until the change of the total photon distribution drops below a determined tolerance.
Let us analyze the integration of the results of the different governing equations 3.11 and 3.4 using an iterative approach.

Rewriting the numerical formulation of Eq. 2.18 for a generic i^{th} step as:

$$2R_{csf}\kappa(f)\mathbf{n}\cdot\nabla\mathbf{\Phi}^i(f) = \mathbf{\Gamma}^i(f) - \mathbf{\Phi}^i(f) \tag{3.12}$$

and substituting $2R_{csf}\kappa(f)\mathbf{n}\cdot\nabla\mathbf{\Phi}^i(f)$ in Eq. 3.11 leads to:

$$\mathbf{\Gamma}^i(f) = \mathbf{F_a}(f)\left(\mathbf{\Phi}^i(f) - \mathbf{\Gamma}^i(f) + \mathbf{\Phi}^i(f)\right). \tag{3.13}$$

Finally, combining Eq. 3.13 and Eq. 3.4 leads to:

$$\begin{cases} \mathbf{A}(f)\mathbf{\Phi}^i(f) = \mathbf{b}^i(f) \\ \mathbf{\Gamma}^i(f) = \mathbf{F_a}(f)\left(2\mathbf{\Phi}^i(f) - \mathbf{\Gamma}^i(f)\right), \end{cases} \tag{3.14}$$

where $\mathbf{q}(f)$ is the vector of the external isotropic source and $\mathbf{b}^i(f) = \mathbf{\Gamma}^i(f) + \mathbf{q}(f)$ with $\mathbf{\Gamma}^{(0)}(f) = 0$. Due to memory constraints, it is not possible to evaluate $\mathbf{\Gamma}^i(f)$ directly as:

$$\mathbf{\Gamma}^i(f) = (\mathbf{I} + \mathbf{F_a}(f))^{-1}\left(\mathbf{F_a}(f)\,2\mathbf{\Phi}^i(f)\right), \tag{3.15}$$

because computation of $(\mathbf{I} + \mathbf{F_a}(f))^{-1}$ would exceed the memory available. As a result, the approach we take for computing the second equation of the system 3.14 is to replace the right hand value of $\mathbf{\Gamma}^i(f)$ with its value $\mathbf{\Gamma}^{i-1}(f)$ at the previous step, leading to the following formula:

$$\mathbf{\Gamma}^i(f) = \mathbf{F_a}(f)\left(2\mathbf{\Phi}^i(f) - \mathbf{\Gamma}^{i-1}(f)\right). \tag{3.16}$$

Finally, the conclusive numerical formulation of the proposed RD solver becomes:

$$\begin{cases} \mathbf{A}(f)\mathbf{\Phi}^i(f) = \mathbf{b}^i(f) \\ \mathbf{\Gamma}^i(f) = \mathbf{F_a}(f)\left(2\mathbf{\Phi}^i(f) - \mathbf{\Gamma}^{i-1}(f)\right) \end{cases} \tag{3.17}$$

3.5 Validation

3.5.1 Diffusive model validation

For the sake of simplicity in the following we consider the CW case, obtained from the previous formulation by setting $f = 0$. A 240 mm × 240 mm × 30 mm sample, based on two overlapped diffusive layers, is used to validate the results provided by the DOT forward problem solver realized.

The configuration is shown in Fig. 3.14 and presents heterogeneous boundary conditions. The optical properties and the thickness of the overlapped layers shown in Fig. 3.14, are the same as [58].

At the top of the overlapped layers we impose a Dirichlet boundary condition ($\Phi = 1$) which corresponds to a time-constant source, on the lateral boundary surfaces a Neumann boundary condition, and on the bottom

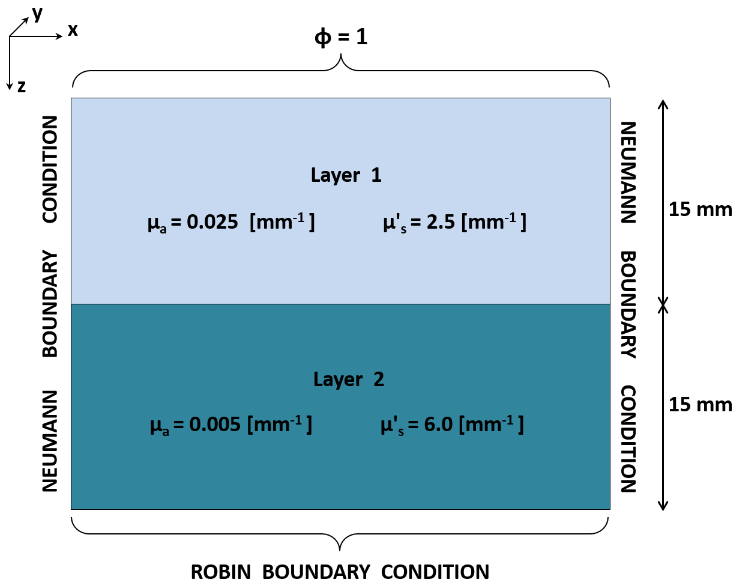


Figure 3.14: Validation sample with heterogeneous boundary conditions.

surface a Robin boundary condition with $R = 1$. Both the Dirichlet and the Neumann boundary condition can easily be applied as particular instances of the RBC. The homogeneous Dirichlet boundary condition is obtained by setting the term R in Eq. 2.15 equal to zero, while the Neumann boundary condition can be physically interpreted as an RBC where total reflection occurs. The Neumann condition prevents light leaking from the lateral surfaces of the domain, leading to a uniform solution in x and y of the Diffusion Equation which can be analytically computed as:

$$\Phi(z) = c_1\, e^{\sqrt{\frac{\mu_a}{\kappa}}z} + c_2\, e^{-\sqrt{\frac{\mu_a}{\kappa}}z}. \tag{3.18}$$

The analytical solution and the relative errors of the numerical solution obtained with a 1 mm, 0.5 mm and 0.25 mm length of the control volume's edge (h) are shown in Fig. 3.15.

Note that the trend of the relative error obtained with a half step is reduced by a factor 4 with respect to the relative error evaluated in the coarser case where it is lower than 8%.

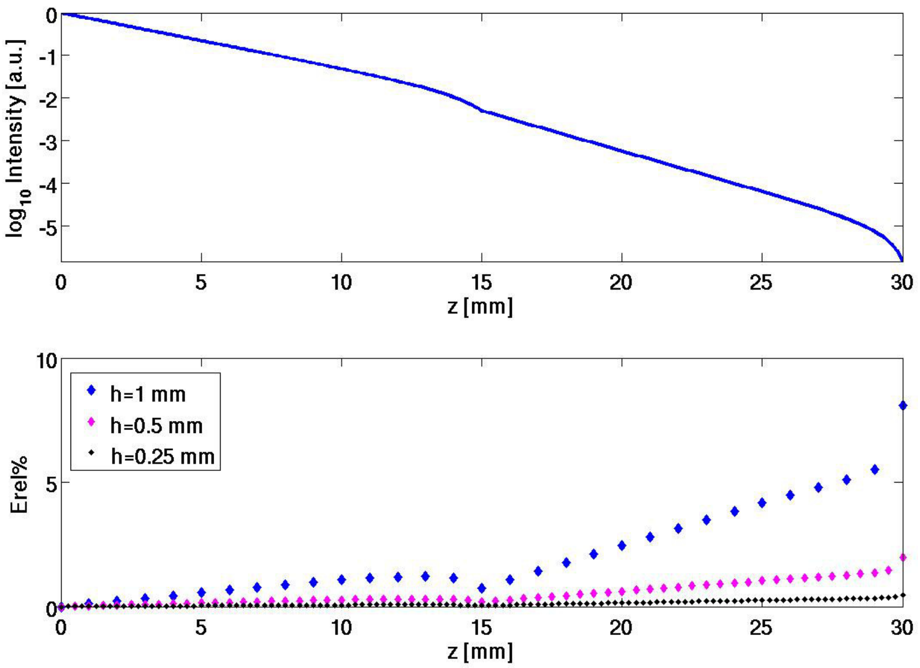


Figure 3.15: Analytical solution in the central section of the sample and comparison between relative errors obtained with $h = 1$ mm, $h = 0.5$ mm and $h = 0.25$ mm in the central section of the sample.

3.5.2 Radiosity model validation

In order to validate the radiosity portion of the algorithm as well, we use a 135 mm × 128 mm × 40 mm sample based on two overlapped layers separated by a non-diffusive film. The geometrical configuration, the optical properties and the boundary conditions of this domain are illustrated in Fig. 3.16.

The configuration of the boundary conditions is similar to the previous one (shown in Fig. 3.14) except that, on the bottom surface of the sample, we imposed a Dirichlet boundary condition ($\Phi = 0$) in order to obtain a simple analytical solution representing the ideal condition of two infinite extended layers separated by a CSF film.

The upper graph of Fig. 3.17 shows the analytical solution, while the lower one illustrates the comparison between the relative errors with h equal to 1 mm and 0.5 mm.

The result of this simulation is affected by the amount of light which escapes from the air-CSF boundary faces involving an increase in the relative error

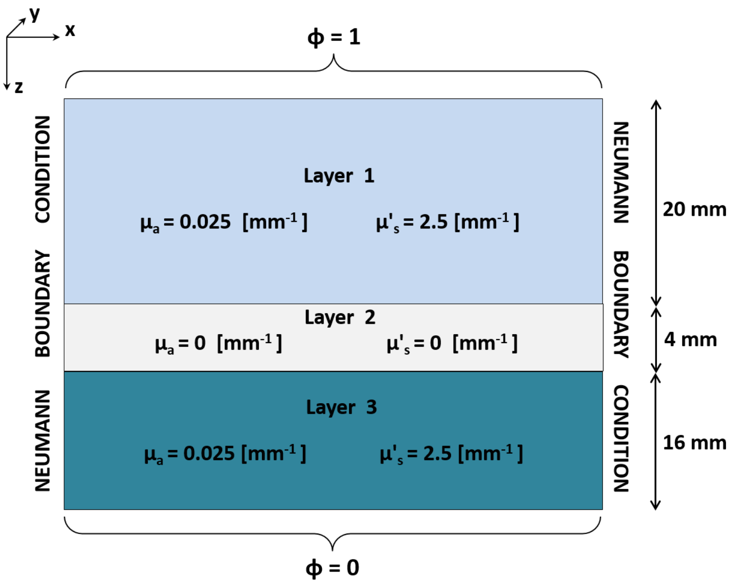


Figure 3.16: Validation sample with heterogeneous boundary conditions and CSF.

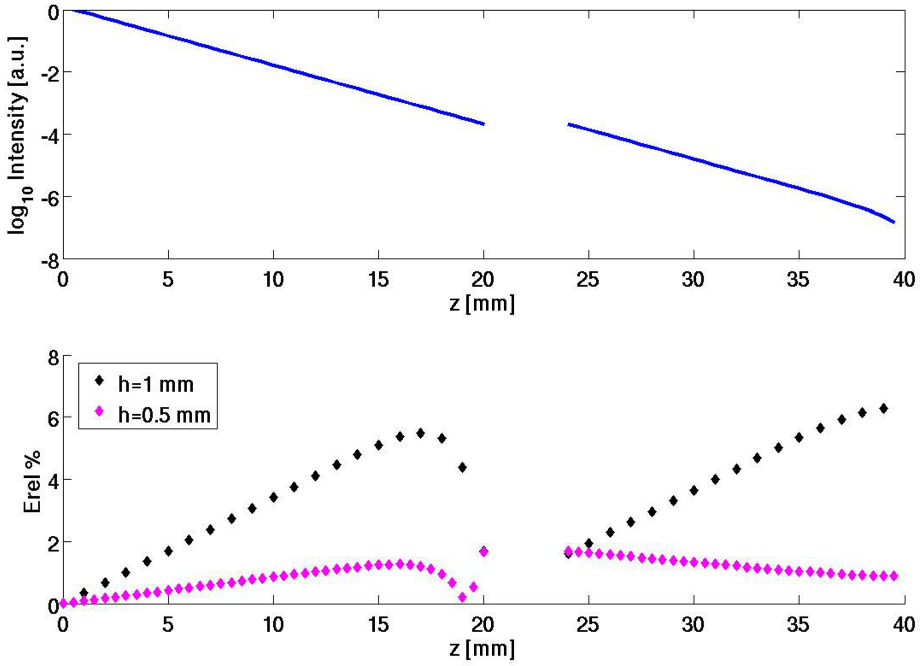


Figure 3.17: Analytical solution in the central section of the sample and comparison between relative errors obtained with $h = 1$ mm and $h = 0.5$ mm in the central section of the sample.

moving towards the sample's boundary. The error due to leaked light is about 2% in the central axis ($x = 67$ mm, $y = 64$ mm) but increases to 80% for the outermost section with $x = 135$ mm and $y = 128$ mm.

3.6 Accuracy Evaluation

3.6.1 Diffusive model accuracy

A 240 mm × 240 mm × 30 mm sample, based on two overlapped diffusive layers, is used to analyse the accuracy of the results provided by the implemented DOT forward problem solver with respect an RTE solution. Within the layers we adopt the realistic optical properties of the skin and of cranial bone as listed in Table 3.6.

Tissue	μ_a [mm^{-1}]	μ_s' [mm^{-1}]
Skin	0.018	1.9
Cranial bone	0.016	1.6
CSF	0.001	0
Gray Matter	0.036	2.2
White Matter	0.014	9.1

Table 3.6: Optical properties of biological head tissues for accuracy evaluation (source : [1]).

The configuration is shown in Fig. 3.18 and presents heterogeneous boundary conditions.

At the top we impose a Dirichlet boundary condition ($\Phi = 1$) which corresponds to a time-constant source, on the lateral boundary surfaces a Neumann boundary condition, and on the bottom surface a Robin boundary condition with $R = 1$. As shown in the previous section, the Neumann condition prevents light leaking from the lateral surfaces of the domain, leading to a uniform solution in x and y of the Diffusion Equation which can be analytically computed as:

$$\Phi(z) = c_1\, e^{\sqrt{\frac{\mu_a}{\kappa}}z} + c_2\, e^{-\sqrt{\frac{\mu_a}{\kappa}}z}. \tag{3.19}$$

To our knowledge no analytical form of the RTE solution is available in the considered 3D finite domain with heterogeneous boundary conditions, so

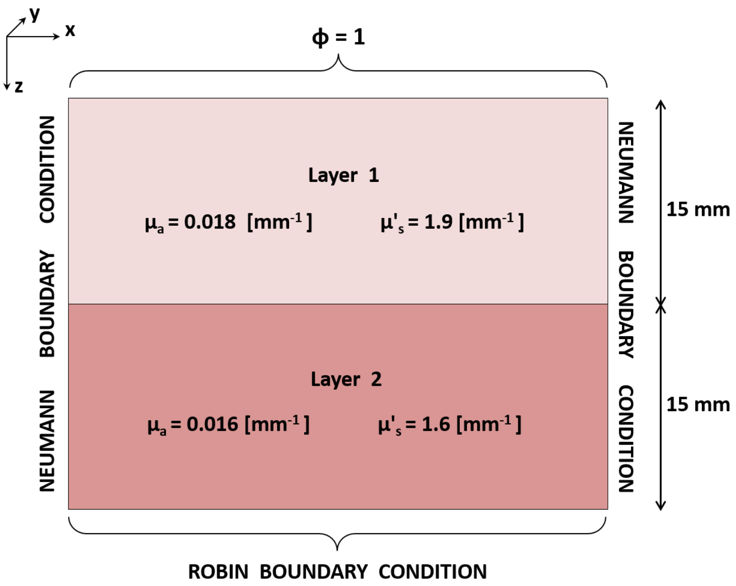


Figure 3.18: Accuracy evaluation sample with heterogeneous boundary conditions.

the benchmark solution of the RTE is achieved in two different steps.
First, the root mean square (RMS) of 100 Monte Carlo runs with $2.1 \cdot 10^9$ photons per run is done. To this end we adopt the Monte Carlo eXtreme (MCX) tool [14] , which is a parallel Monte Carlo simulation software for time-resolved photon transport in 3D domains optimized to run on a single GPU.
After that, the mean in the z-direction of the RMS is considered, exploiting the lateral Neumann condition which forces the solution to be constant in the xy plane for each fixed z. The Monte Carlo solution takes into account the anisotropic scattering of light (with anisotropy factor $g = 0.9$ for each biological tissue except CSF), providing the most realistic solution of the RTE available. This solution is shown in the top plot of Fig. 3.19 and is used to estimate the accuracy of the DE analytical solution of Eq. 3.19 with respect to an RTE in a turbid domain.

The relative error between the analytical solution of the DE and the gold-standard RTE is shown in the middle plot of Fig. 3.19. The maximum relative error between these solutions is about 7.6% and the mean relative error is 4.8%, showing a good agreement between the DE analytical solution

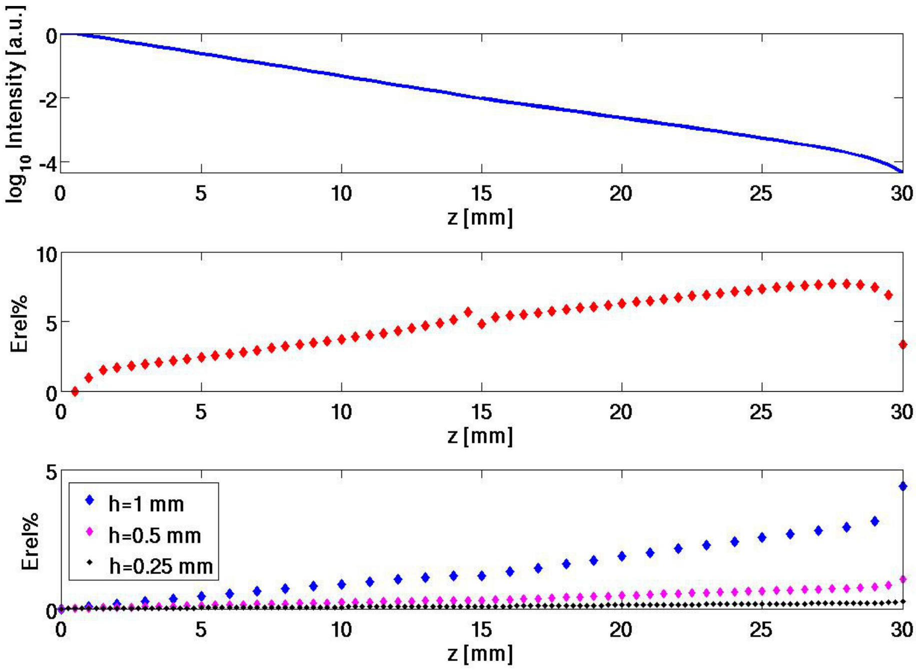


Figure 3.19: Gold-standard RTE solution computed by multiple MCX runs (top plot), relative error between gold-standard RTE solution and analytical DE solution (Eq. 3.19) (middle plot), and numerical errors between analytical DE solution (Eq. 3.19) and numerical solution (Eq. 3.4) obtained with $h = 1$ mm, $h = 0.5$ mm and $h = 0.25$ mm in the central section of the sample (bottom plot).

(Eq. 3.19) and the RTE solution computed by the MCX tool.

To maintain coherence between the RTE and the DE theories the accuracy comparison is evaluated starting from the scattering depth $1/\mu_s'$, where the anisotropic behavior of the Monte Carlo source becomes negligible.

The numerical errors between the FVM numerical implementation of the DE (Eq. 3.4) and the DE analytical solution (Eq. 3.19) obtained with different spatial discretization steps h equal to 1 mm, 0.5 mm and 0.25 mm are finally shown in the bottom graph of Fig. 3.19, validating the numerical solver with respect to the analytical solution of the equation it implements. Note that the trend of the relative error obtained with a half step is reduced by a factor 4 with respect to the relative error evaluated in the coarser case where it is lower than 5%.

3.6.2 Radiosity model accuracy

In order to validate the radiosity portion of the algorithm as well, we use a 135 mm × 128 mm × 40 mm sample based on two overlapped layers separated by a non-diffusive film. The geometrical configuration, the optical properties and the boundary conditions of this domain are the same of Fig. 3.16.

The RTE benchmark solution is obtained following the same procedure as in the previous accuracy evaluation experiment, running 100 MCX simulations with $2.1 \cdot 10^9$ photons per run. The benchmark RTE solution is shown in the top graph of Fig. 3.20.

The middle plot of Fig. 3.20 illustrates the trend of the relative error between the analytical RD solution in the form of Eq. 3.19 and the gold-standard RTE solution obtained with the MCX tool. The comparison between RTE and analytical RD solutions is evaluated from the scattering depth and shows

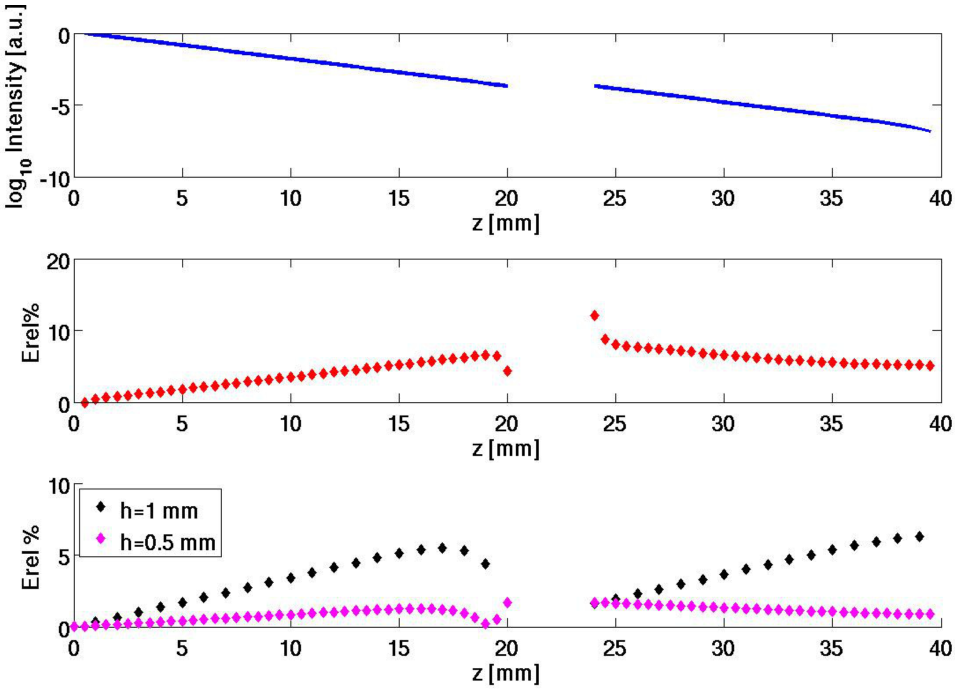


Figure 3.20: Gold-standard RTE solution computed by multiple MCX runs (top plot), relative error between gold-standard RTE solution and analytical RD solution (Eq. 3.19) (middle plot), and numerical errors between analytical RD solution (Eq. 3.19) and numerical solution (Eq. 3.17) obtained with $h = 1$ mm and $h = 0.5$ mm in the central section of the sample (bottom plot).

good agreement between the models, revealing a mean relative error of about 4.8% and a maximum relative error of about 12% at the CSF interface. Finally, the numerical errors between the proposed RD numerical solution (Eq. 3.17) and the RD analytical solution (Eq. 3.19) obtained with h equal to 1 mm and 0.5 mm are shown in the bottom graph of Fig. 3.20, revealing a maximum relative error of about 5% in the coarser case and also validating the radiosity portion of the proposed RD numerical solver.

It is important to specify that the result of this simulation is affected by the amount of light which escapes from the air-CSF boundary faces involving an increase in the relative error moving towards the sample's boundary. The error due to leaked light is about 2% in the central axis ($x = 67$ mm, $y = 64$ mm) but increases to 80% for the outermost section with $x = 135$ mm and $y = 128$ mm.

3.7 Performance

3.7.1 Time of Convergence

Four samples with different sizes (105 mm × 64 mm × 32 mm, 105 mm × 64 mm × 48 mm, 105 mm × 64 mm × 64 mm and 105 mm × 64 mm × 128 mm) based on three overlapped layers with a Robin boundary condition over each external surface are used to test the realized solver and to compare the results and the performances with the parallel MCX-Extreme solver [14] based on GPU parallel implementation of the Monte Carlo method.

The optical and the geometrical properties are summarized in Table 3.7.

In the comparison the isotropic source is placed in position ($x = 52$ mm, $y = 32$ mm, $z = 1$ mm), and we conduct two different tests (for every sample) with and without inclusion of the boundary reflection.

Layer	μ_a [mm^{-1}]	μ_s' [mm^{-1}]	Thickness [mm]
1	0.019	0.858	20
2	0.004	0.009	5
3	0.020	0.99	$7 - 23 - 39 - 103$

Table 3.7: Optical properties and thickness of the layers composing the 4 samples whose sizes are 105 mm × 64 mm × 32 mm, 105 mm × 64 mm × 48 mm, 105 mm × 64 mm × 64 mm and 105 mm × 64 mm × 128 mm.

In the case of the implemented numerical method we consider the computing time as the time required for convergence of the iterative solver, while in the case of MCX we consider the computing time as the time required to detect at least one photon by a detector of radius 1 mm at the farthest point from the source.

The HPC specifications are listed in Table 3.5.

For the 105 mm × 64 mm × 32 mm validation sample, the pre-processing time, including visibility computation and form factor matrix assembly, is 563 s. The time required for construction of matrix $\mathbf{A}$ and vector $\mathbf{b}$ is 1.25 s. The computation times are shown in the histogram of Fig. 3.21 and involve iterative solution of the linear system and updating of the $\mathbf{b}$ vector, without inclusion of the pre-processing time. The performance of Fig. 3.21 are obtained employing a single GPU run and underline a speed-up of a factor 7 with respect to MCX. We can observe how the increase in computational times is less than linearly proportional to the domain size.

No direct comparison with the MCX tool is possible for larger domains since the maximum number of photons supported by MCX is not sufficient to allow detection of photons at the required depths. Simulations show how, on increasing the depth from 32 to 48 mm, the minimum value of the light intensity in the domain decreases by a factor of 100. We can estimate that the number of photons required by a Monte-Carlo solver would scale accordingly by a similar factor, leading to unacceptable computational times.

Furthermore, it is possible to estimate the arithmetic throughput of a single Titan Black card for every performance test, knowing that 156 floating point operations are required to process one node for every iteration of the BiCG solver. The double precision arithmetic throughput for a single GTX Titan Black is 42.9 GFLOPS in the 105 mm × 64 mm × 32 mm sample, 46.2 GFLOPS in the 105 mm × 64 mm × 48 mm domain, 47.1 GFLOPS in the 105 mm × 64 mm × 64 mm volume and 51.7 GFLOPS in the 105 mm × 64 mm × 128 mm sample. By running the same tests using all four GPUs available, we obtain, respectively, 60.3, 78.8, 88.8 and 125.8 GFLOPS. Due to the presence of memory accesses, these values of arithmetic throughput are smaller than the theoretical ones, but it is important to notice that they increase for larger domain sizes.

The histogram illustrates that the numerical solver speed up significantly

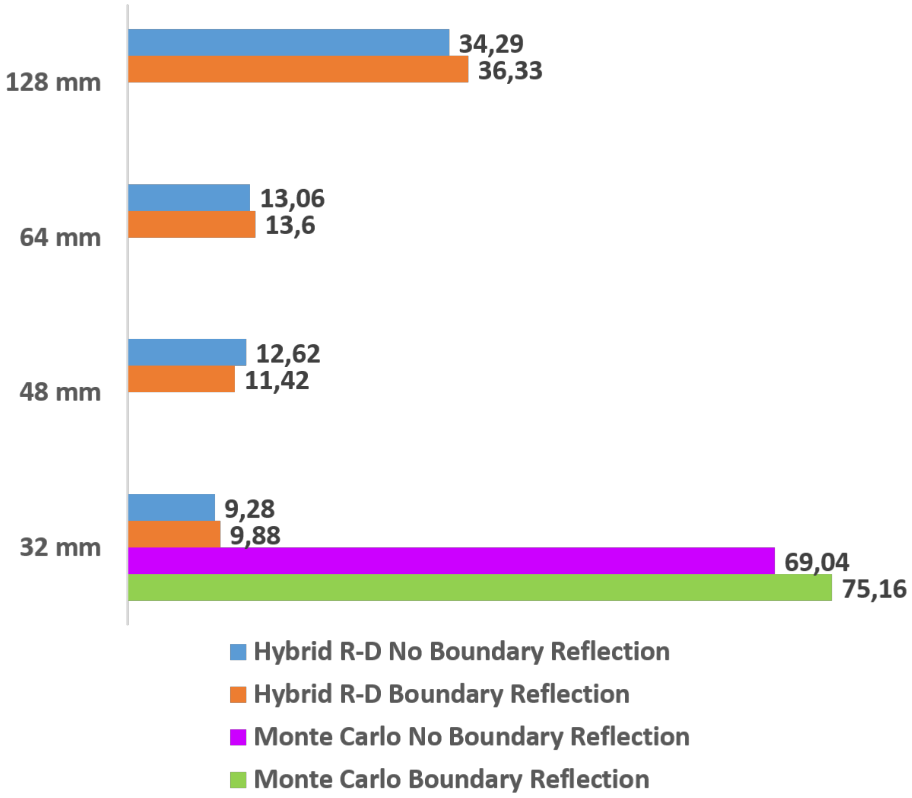


Figure 3.21: Performance [s] of the numerical solver for different depths of the sample with and without boundary reflection.

increases for larger domain's size, while the number of photons required by Monte Carlo increases in order to obtain a comparable accuracy, leading to a considerable increase of the computation time.

3.7.2 Performance in function of accuracy

In order to test the performance of the proposed algorithm, the sample of Fig. 3.16 is exploited, after extension of the third layer's thickness to 104 mm. The HPC system of Table 3.5 is employed.

The pre-processing time required by the RD solver implemented is 3815 s, including visibility evaluation, form factor matrix assembling and storage operations. The time required for construction of matrix $\mathbf{A}$ and vector $\mathbf{b}^i$ of Eq. 3.17 is 12 s. The performance of the proposed GPU-based RD numerical solver, of the isotropic-scattered MCX and of the anisotropic-scattered MCX are compared in terms both of computation times and accuracy, which is evaluated as the percentage relative error between the approximated solution and the RTE benchmark solution computed at distance z equal to 40 mm, 50 mm and 60 mm. The benchmark solution is obtained following the same reasoning as in Sec. 3.6, exploiting 300 MCX runs with $2.1 \cdot 10^9$ photons per run. The computation times of the GPU-based RD, of the isotropic-scattered MCX and of the anisotropic-scattered MCX are listed in Table 3.8. In the case of the RD solution, the elapsed time involves iterative solution of the linear system and an update of the $\mathbf{b}^i$ vector, without inclusion of the pre-processing time.

The performance of the GPU-based RD solver in Table 3.8 shows a speed-up of a factor 7 over a single run of isotropic MCX, with higher accuracy than that of 100 runs of isotropic MCX and a precision comparable to 10 runs of anisotropic MCX. As previously shown in Fig. 3.20, it can be observed that the numerical error associated with the RD model is non-monotonic with increasing depth since it has some local maxima close to the interface. However, these maxima are lower than the error at 40 mm depth.

Solver	Runs	Time [s]	Relative Error		
			40 mm	50 mm	60 mm
MCX aniso	1	4175	13.81%	39.06%	92.82%
MCX iso	1	548	19.58%	39.40%	84.76%
MCX aniso	10	42283	4.38%	12.39%	28.00%
MCX iso	10	5590	17.51%	22.28%	33.78%
MCX aniso	100	417720	1.80%	4.22%	9.56%
MCX iso	100	54827	17.50%	21.32%	25.35%
RD 1 GPU	/	81	8.9%	4.5%	14.22%

Table 3.8: Performance comparison between the MCX tool and the proposed RD numerical solver

Furthermore, it is possible to estimate the arithmetic throughput of a single Titan Black GPU card for this performance test, knowing that 156 floating point operations are required to process one node for every iteration of the BiCG solver. The double precision arithmetic throughput for a single GTX Titan Black is 51.7 GFLOPS, while when using all four GPUs available it increases to 155.3 GFLOPS. Due to the presence of memory accesses, these values of arithmetic throughput are smaller than the theoretical ones, but it

is important to notice that they increase for larger domain sizes.
In order to estimate the speed improvement that takes advantage of the massively parallel GPU implementation of the BiCG solver, an OpenMP implementation of the solver is performed on the same HPC.
By running the parallel BiCG solver on the 4 GPUs available, the speed-up factor is 20 times if compared to the single threaded CPU-based implementation and 4 times in the case of 16 CPUs running. The BiCG speed enhancement with all 4 GPUs running allows a 20% speed improvement of the RD solver (63 s instead of 81 s).

3.8 Realistic Domain

We finally evaluate the performance and accuracy of the implemented RD solver in a complete and realistic human head volume with the optical properties listed in Table 3.6. We consider a physical domain generated starting from 3T MRI images with a voxel resolution of 0.94 mm × 0.94 mm × 1.20 mm resulting in a full structural MRI with 5 million voxels with a head volume of around 2.7 million voxels. Given the anatomical structure of the domain, voxel visibility and form factors are computed only once.

The structural pre-processing phase comprises the first 4 blocks in Fig. 3.22 and takes approximately 5 hours; this phase only needs to be evaluated again if the patient-specific MRI changes. In order to achieve a reasonable pre-processing time, the parallel and efficient system for form factors computation described in Sec. 3.4.2 has been implemented, because form factors evaluation represents by far the most computationally expensive step of the whole pre-processing phase. The other pre-processing operations were implemented in non-optimized Matlab code since the total pre-processing time has already been reduced to acceptable levels.
Assembly of matrix $\mathbf{A}$ takes 39 seconds and only needs to be performed if the distribution of the optical properties change within the domain as happens during DOT inverse problem solution. If multiple light sources need to be considered, matrix $\mathbf{A}$ is preserved. Forward problem solution takes 116 seconds. For every iteration one needs to solve the linear system $\mathbf{A\Phi} = \mathbf{b}$, with reference to Eq. 3.17, and update term $\mathbf{b}$; the dimension of matrix $\mathbf{A}$ is about 5 million by 5 million ($\approx$ 19M non-zero elements).

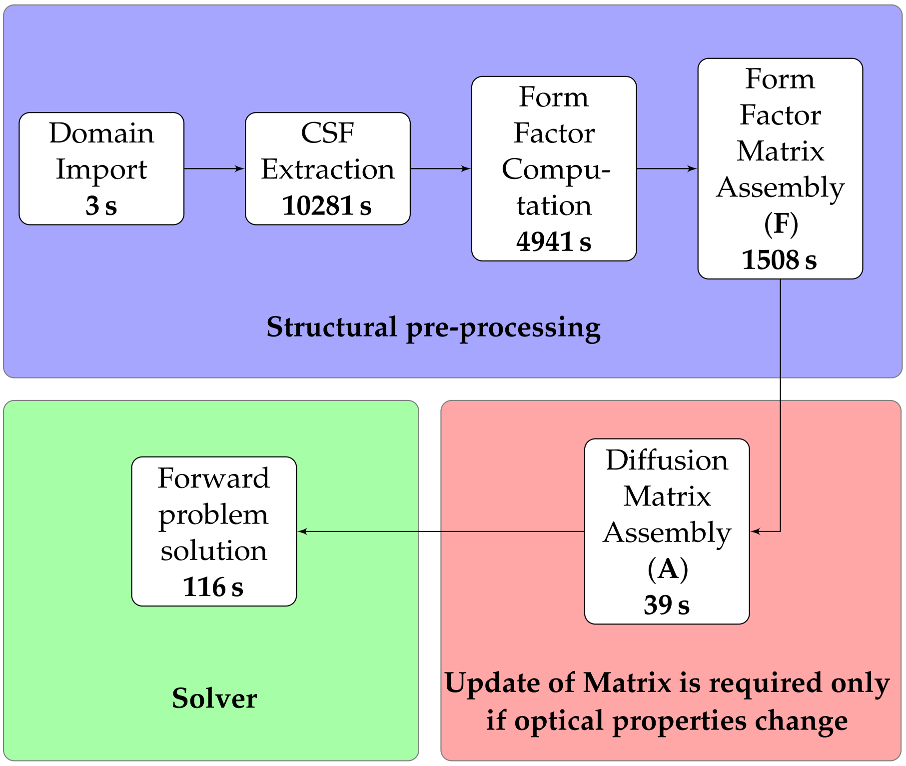


Figure 3.22: Time elapsing for forward problem solution in a complete human head domain derived from 3T structural MRI.

The time required to complete each iteration of the Radiosity-Diffusion algorithm is approximately 6 seconds, giving a total time of about 116 seconds to complete all 20 iterations required for convergence. Each iteration employs 6 seconds in all, but solution of the linear system using the iterative BiCG solver only takes 3.3 seconds to complete all 760 BiCG iterations required for a precision of 10^{-45}. Since the total number of nodes is about 5 million and for each BiCG iteration 156 floating point operations are required to process one node, the arithmetic throughput on four GTX Titan Black is about 180 GFLOPS.

A general formula for the computation of the elapsed time T required by the proposed RD method in a DOT inversion process which requires N_{it} total iterations and N_{fw} forward solver computations for each iteration can be expressed as follows:

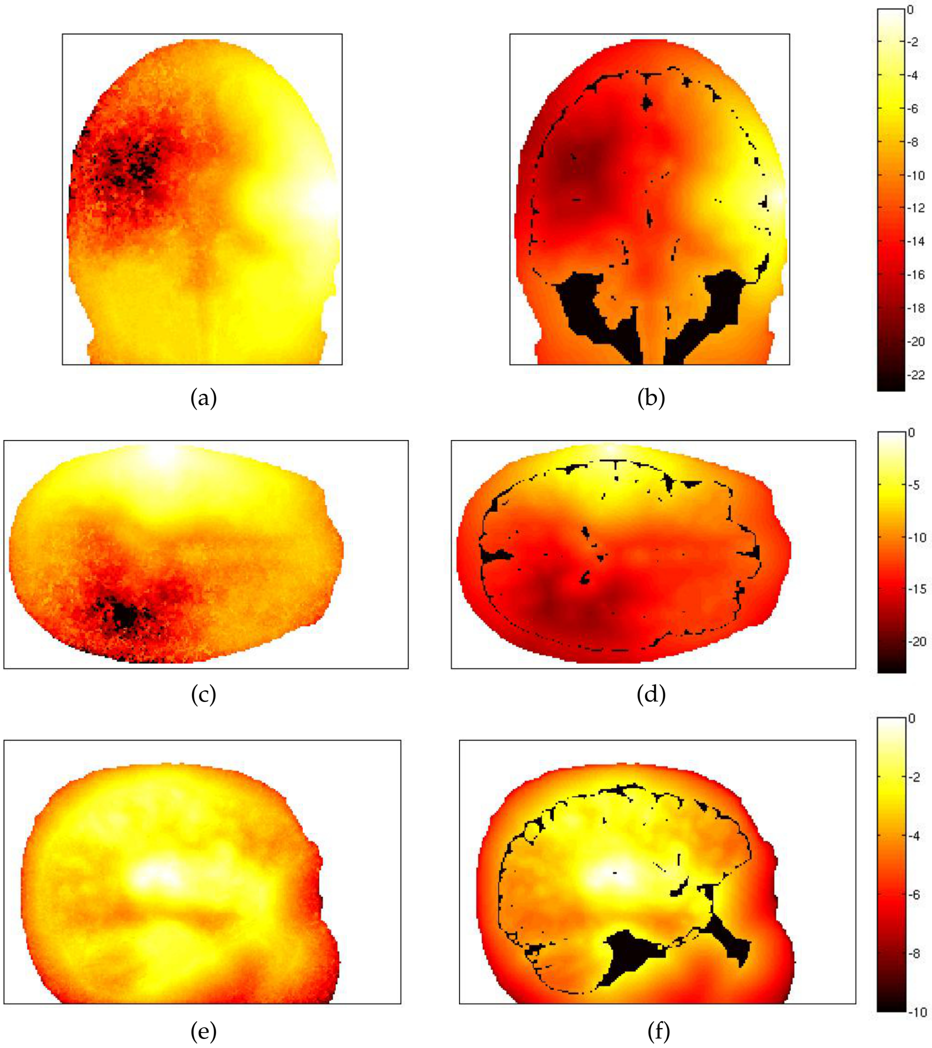
$$T = N_{it}\,(39 + 116 \cdot N_{fw}),$$

where 39 s is the elapsed time for the construction of matrix **A** and 116 s is the time necessary to obtain a single forward problem solution. The total number of iterations N_{it} is strictly related to the stopping criterion of the inverse problem solver such as N_{fw}, which is related to the total number of sources and detectors.

Results provided by the numerical RD algorithm are compared with the solutions obtained employing isotropic-scattered MCX and with the pure Diffusion Equation model. The number of photons used in the isotropic MCX simulation is $2.1 \cdot 10^9$ leading to a total elapsed time of about 2489 s. A comparison of the three solutions in the coronal, sagittal and transverse section of the head is shown in Fig. 3.23, where the main limitations of both models stand out. In fact, despite the maximum number of photons employed in the single MCX run, the resulting light distribution is markedly granular, and the computational cost of this method is remarkable, since 2489 s is the elapsed time for a single isotropic run and 15479 s for an anisotropic one. Nevertheless, MCX, being an RTE solver, has the ability to manage both the scattering anisotropy within biological tissues and the low-scattering property of CSF, contrary to what the proposed RD numerical model is able to do. The black area of Fig. 3.23 (right column) is the result of this inability of the proposed solver to determine the light distribution within the non-scattering CSF layer, where the value of the solution is fixed to zero. On the other hand, the RD solver is characterized by a lower computational cost and no granularity in deeper regions.

Finally, Fig. 3.24 illustrates the comparison between the solutions as a function of the distance from the light source in the transverse slice shown in Fig. 3.23. One can observe a good agreement between the RD numerical model and the RMS of 100 isotropic MCX runs whereas a single MCX isotropic run exhibits high granularity.

Furthermore, it should be pointed out that the Monte Carlo noise (including values listed in Table 3.8) carries a high degree of point-to-point and run-to-run stochastic variability. This can significantly increase the noise level when using differential approaches for, e.g., Jacobian computation through both perturbative and adjoint method in DOT image reconstruction [73][74][75]. The RD model also shows a substantial improvement over the pure diffusive model (DE) in a realistic head domain.



(a) (b)

(c) (d)

(e) (f)

Figure 3.23: Comparison between light distribution obtained employing MCX tool (left column) and numerical RD solver (right column) in the sagittal (top plots), transverse (middle plots) and coronal (bottom plots) section of the realistic human head.

3.9 Conclusions

This chapter focuses on the numerical implementation of a Radiosity Diffusion solver based on the numerical integration between the diffusion equation and the radiosity equation and it is the core of this thesis. It represents an original contribution which updates the state of the art of DOT forward problem solvers providing a solver able to implement a Radiosity-Diffusion model able to manage the geometrical complexity of the head volume directly imported from a full resolution structural MRI.

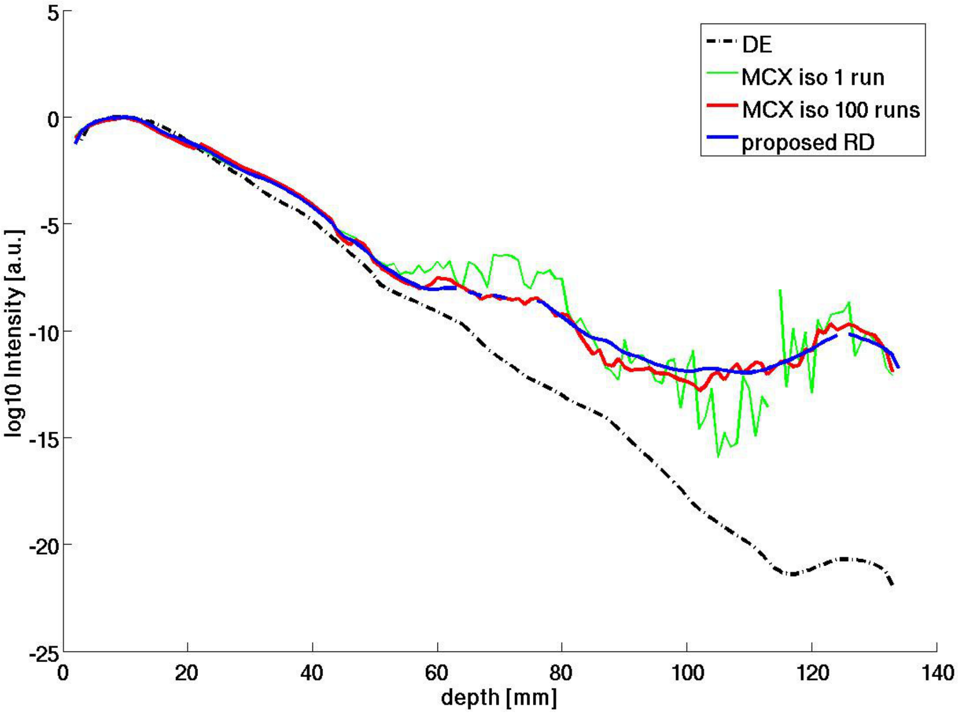


Figure 3.24: Distribution of light in the Transverse section of the realistic head domain as a function of the distance from the light source.

Validation, accuracy and performance of the proposed Radiosity-Diffusion numerical solver are examined in detail with respect to a benchmark solution obtained using multiple runs of a gold-standard Monte Carlo RTE and a complete overview of the work-flow on a realistic human head is given. Results in a full-resolution structural MRI of a human head composed by 3 million non-zero voxels underlines a 7x speed-up over a single run of isotropic-scattered parallel Monte Carlo engine with an accuracy comparable to 10 runs of anisotropic scattered Monte Carlo in the same geometry, allowing one to compute the light distribution of a full human head in 116 seconds for the platform used.

In Chapter 4 a DOT inverse problem solver based on the proposed RD numerical forward problem solver is implemented, providing a complete toolbox for a DOT imaging system.

Chapter 4

DOT Inverse Problem

The DOT forward problem described in Chapter 2 and numerically solved through the implemented RD solver of Chapter 3, involves calculation of measurements which are expected on the domain boundaries given the sources and the internal properties of the domain.
As shown in Chapter 1, to reconstruct an image it is necessary to solve the inverse problem [8] which allows to calculate the internal optical properties given a set of photon density measurement (a discrete sampling of the overall distribution on the boundary) and the intensity of light sources.
This chapter introduces the DOT inverse problem (Sec. 4.1), giving the basis of Jacobian computation through perturbation method (Sec. 4.2.1) or adjoint method (Sec. 4.2.2). Finally, Sec. 4.3 shows a typical work-flow of DOT in a whole realistic human head, which starts importing the domain and ends to the reconstruction of the brain optical properties.
The work illustrated in this chapter was developed in the scope of the HIGH PROFILE European Project and it is described in detail in [76].

4.1 Inverse Problem Theory

The inverse problem in DOT is solved by minimizing an objective function ψ over the range of optical properties in the least square sense. The objective function is defined as:

$$\psi = \|\Phi_{meas} - \Phi_p(\gamma)\|_2^2 \tag{4.1}$$

where Φ_{meas} represents the set of measurements of photon density in the domain boundary, $\Phi_p(\gamma)$ represents the numerical light distribution evaluated in the domain with optical properties $\gamma = (\mu_a, \mu_s')$, where μ_a is the absorption coefficient while μ_s' is the reduced scattering coefficient.
Two different non-linear methods are usually employed in order to minimize this function:

- Direct methods;
- Gradient based methods.

Direct methods, known as full-Newton approaches [8], are the most straightforward and relatively easy to implement. The prototypical algorithm for Newton methods is the Levenberg-Marquardt algorithm, that leads to the calculation of $(J^T \cdot J + \lambda I)^{-1}$, where λ is a stabilization coefficient and J is the Jacobian of $\Phi_p(\gamma)$.
In our problem, J is significantly ill conditioned, therefore computing $J^T \cdot J$ leads to a problem with such an ill conditioning to be practically unsolvable. This has been evaluated on small toy-problems before analysing the full domain. In fact, due to the size of the domain, $J^T \cdot J$ could not be practically computed in the full model due to memory constraints.
On the other hand, methods based on gradient optimization, which do not require an explicit inversion of $J^T\, J$are known to be computationally efficient [77][78]. Since the objective function is differentiable, using a derivative-based iterative method is most appealing.

4.2 Gradient Based Methods

Having the Jacobian matrix J of $\Phi_p(\gamma)$ at hand, the inverse problem can be processed iteratively. If we define:

$$\gamma_{i+1} = \gamma_i + \Delta\gamma_i \tag{4.2}$$

and we consider Taylor series expansion of the function $\Phi_p(\gamma_{i+1})$, it results:

$$\Phi_p(\gamma_{i+1}) = \Phi_p(\gamma_i) + J\Delta\gamma_i + \frac{1}{2}H\Delta\gamma_i^2 + \dots \tag{4.3}$$

where H is the Hessian matrix associated to the problem. Truncating the series to the second term (linearization), we have:

$$\Phi_p(\gamma_{i+1}) = \Phi_p(\gamma_i) + J\Delta\gamma_i \; . \tag{4.4}$$

Substituting the approximation into the objective function leads to:

$$\psi = \|\Phi_{meas} - \Phi_p(\gamma_i) - J\Delta\gamma_i\|_2^2; \tag{4.5}$$

where, defining $\Delta\Phi_i = \Phi_{meas} - \Phi_p(\gamma_i)$, we obtain the following objective function:

$$\psi_i = \|\Delta\Phi_i - J\Delta\gamma_i\|_2^2 \; . \tag{4.6}$$

Equation 4.6 can be solved in the least square sense in order to find the update term of optical properties $\Delta\gamma_i$.

To sum up, the linearized iterative formulation of the problem becomes:

$$\gamma_{i+1} = \gamma_i + lsqr(J, \Delta\Phi_i) \; , \tag{4.7}$$

where by *lsqr* we imply least square minimization routines such as the one directly available in MATLAB. Since this operation is relatively fast with respect to Jacobian computation, we can safely use the built-in MATLAB function without introducing significant overhead. This operation has to be repeated until ψ_i drops down a certain user-established tolerance.

This iterative method implies that the FVM stiffness matrix $A(\gamma_i)$ needs to be updated on each iteration. Since Jacobian computation is a computationally intensive task, we found that re-using the same Jacobian matrix in successive steps significantly reduces computation times, even if it can increase the number of iterations which are required to obtain convergence.

There are several methods to calculate J, in particular, for our purpose, two different methods are considered:

1. Perturbation method;

2. Adjoint method.

4.2.1 Perturbation method

The basic idea of the perturbation method is to slightly modify the optical properties of each region having unique properties which composes the domain , one-by-one, computing the difference with respect to the unperturbed solution and dividing by the magnitude of the perturbation.
If S is the number of sources, M is the number of measurements for each source and L is the number of regions with unique optical properties, for each measurement we need to solve $L \cdot S + 1$ forward problems.
Since our interest is to reconstruct the optical properties of each voxel composing the domain and given that the total number of non-zero voxels composing a structural MRI of a human head is around 3 million, this method is unsuitable to evaluate the Jacobian in our application. In practice, this method is suitable only if the domain is segmented into a small amount of homogeneous regions to be perturbed. A more efficient and appealing scheme could be derived if the so-called *reciprocity principle* is considered.

4.2.2 Adjoint method

In order to assemble the Jacobian matrix, the adjoint method is based on the *reciprocity principle* of light propagation. In short, reciprocity states that, given a light source and a detector in a medium with optical properties which are generally verified in nature (and in our case as well), the amount of light reaching the detector is the very same that would reach the detector if source and detector were switched. Reciprocity can be derived analytically from the radiative transfer equation ([79] [80]).
Arridge et al. [81] proved that reciprocity theorem allows computing one element of the Jacobian, relative to a particular voxel and a source-detector pair, by simple algebraic operations performed on the solution of two forward problems in the voxel under examination.
The first solution is computed for the real DOT problem under examination, the second one has the source placed at the original detector position. In [81], this scheme was applied for the finite element method formulation of diffusion equation solver. We extended these finding to our FVM implementation as follows.
Let Φ be a solution for the parameters (μ_a, κ) and let $\widetilde{\Phi}$ be the solution for

the same source, but a different set of parameters $(\widetilde{\mu}_a, \widetilde{\kappa})$ where:

$$\widetilde{\mu}_a = \mu_a + \alpha \qquad \widetilde{\kappa} = \kappa + \beta \; ; \tag{4.8}$$

As shown in [8], reciprocity theorem leads to:

$$\Delta\Phi(\alpha, \beta) = -\int_V \big(\beta \, \nabla\Psi \cdot \nabla\Phi + \alpha\Psi\Phi\big) \, dV \tag{4.9}$$

where V is the volume under analysis and Ψ is the solution of the adjoint problem, in which the source is placed in the detector position. By setting α to zero and calculating $\Delta\Phi(\alpha, \beta)/\beta$ we obtain the portion of the Jacobian relative to the diffusion coefficients. The portion of the Jacobian related to absorption is obtained by setting β to zero and calculating the right term of the integral. For the sake of compactness, we do not include computation of the elements related to diffusive coefficients, while we briefly report the first steps of calculation of those related to absorption.
In this case, the integral becomes:

$$\Delta\Phi(\alpha) = -\int_V \alpha\Psi\Phi \; dV \; . \tag{4.10}$$

Assuming Φ and Ψ piecewise linear we obtain:

$$\begin{aligned} \Delta\Phi(\alpha) = -\int_{V_{voxel}} &\left(\Phi_c + \frac{\Phi_x - \Phi_c}{h}\, x + \frac{\Phi_y - \Phi_c}{h}\, y + \frac{\Phi_z - \Phi_c}{h}\, z\right) \cdot \\ &\cdot \left(\Psi_c + \frac{\Psi_x - \Psi_c}{h}\, x + \frac{\Psi_y - \Psi_c}{h}\, y + \frac{\Psi_z - \Psi_c}{h}\, z\right) dV \end{aligned} \tag{4.11}$$

where Φ_c is the photon density centered in one node of the grid and Φ_x, Φ_y, Φ_z are the closest nodal values by following the axis directions. The integral is then numerically computed.
Derivation of $\Delta\Phi(\beta)$ is similar, however gradients need to be computed before integration, following the piecewise linear approximation of Φ.
Given S light sources, L voxels and M detectors, the overall computation cost is approximately that of computing $S \cdot M + 1$ forward problems. With respect to the perturbation approach, if L is in the order of some millions and M of a few tens maximum, the speed-up is more than 10.000 times, marking the boundary between a computation which can be performed on rather

low-cost workstation in reasonable times versus computation which would either require large-scale super-computers or unacceptable computation times.

4.3 Software Work-Flow in the Human Head

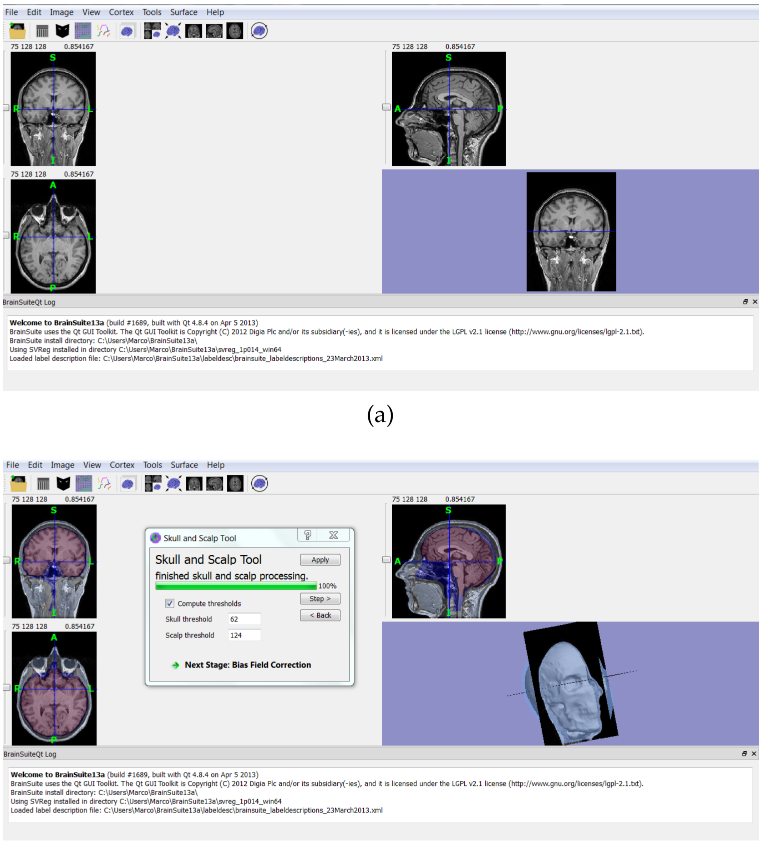
In this section we explain the implementation of the inverse problem solver using adjoint method for the Jacobian computation in a complex and large volume such as the segmented human head imported directly from a structural MRI.

In order to validate the proposed inverse problem solver, a set of ideal measures was generated by computing the light distribution in the whole human head using the implemented RD forward problem solver.

The work-flow of the inverse problem solver in the whole human head is illustrated in Fig. 4.3. Each point is then singularly analysed.

1 ▶ Segmentation of 3D Structural MRI
↓
2 ▶ Matlab Import of 3D segmented MRI
↓
3 ▶ CSF Surface Extraction
↓
4 ▶ Parallel Form Factor Evaluation Algorithm
↓
5 ▶ Assign Guess Absorption Coefficient to every voxel of the Head Domain
↓
6 ▶ Forward Solution to Predict Measurements
↓
7 ▶ Jacobian Computation (S×M+1) Forward Solvers
↓
8 ▶ Compute $\Delta\mu_a$
↓
9 ▶ Update Absorption Coefficients of the Head
↓
10 ▶ Jump to **6** and Iterate till Convergence (3/4 times)

1 ▶ Segmentation of 3D Structural MRI



```
Welcome to BrainSuite13a (build #1689, built with Qt 4.8.4 on Apr 5 2013)
BrainSuite uses the Qt GUI Toolkit. The Qt GUI Toolkit is Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies), and it is licensed under the LGPL v2.1 license (http://www.gnu.org/licenses/lgpl-2.1.txt).
BrainSuite install directory: C:\Users\Marco\BrainSuite13a\
Using SVReg installed in directory C:\Users\Marco\BrainSuite13a\svreg_1p014_win64
Loaded label description file: C:\Users\Marco\BrainSuite13a\labeldesc\brainsuite_labeldescriptions_23March2013.xml
```

(a)

Skull and Scalp Tool

finished skull and scalp processing.

100%

☑ Compute thresholds

Skull threshold 62

Scalp threshold 124

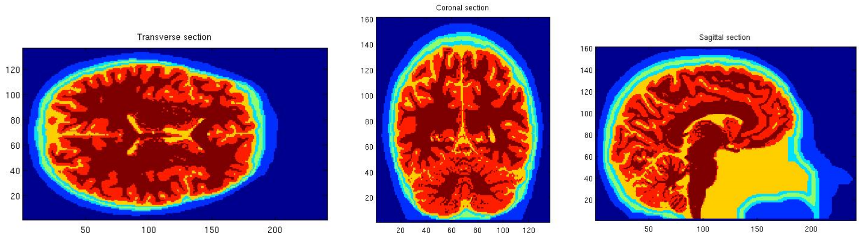
Next Stage: Bias Field Correction

```
Welcome to BrainSuite13a (build #1689, built with Qt 4.8.4 on Apr 5 2013)
BrainSuite uses the Qt GUI Toolkit. The Qt GUI Toolkit is Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies), and it is licensed under the LGPL v2.1 license (http://www.gnu.org/licenses/lgpl-2.1.txt).
BrainSuite install directory: C:\Users\Marco\BrainSuite13a\
Using SVReg installed in directory C:\Users\Marco\BrainSuite13a\svreg_1p014_win64
Loaded label description file: C:\Users\Marco\BrainSuite13a\labeldesc\brainsuite_labeldescriptions_23March2013.xml
```

(b)

Figure 4.1: Structural MRI data in NifTi format is imported in BrainSuite for domain segmentation (a)-(b). DICOM data can be converted to NifTi before processing. Segmentation is performed in a semi-automated way, with some parameters which can be modified to improve segmentation quality. Data is exported in ANALYZE format and imported in MATLAB through an ad-hoc routine. Segmentation of the structural MRI has been examined in detail in Section 3.1.

2 ▶ Matlab Import of 3D Segmented MRI

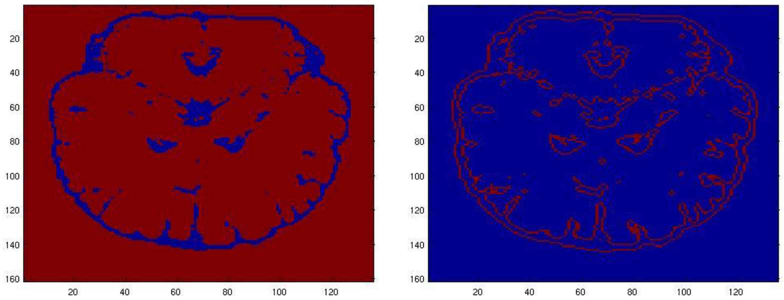


(a) (b) (c)

Figure 4.2: Segmented structural MRI information visualized through ad-hoc MATLAB routine.

Sagittal, coronal and transverse section showed above are obtained in the central section of the segmented MRI.

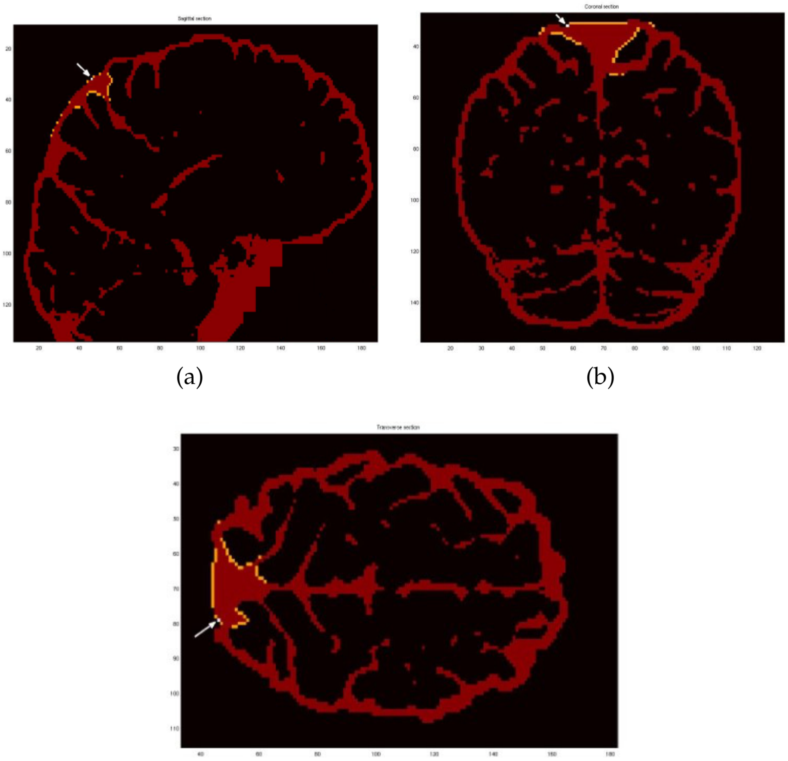
3 ▶ CSF surface extraction



(a) (b)

Figure 4.3: Section plot of CSF region (a) and extraction of CSF-boundary voxels through the ad-hoc Matlab tool (b). Coordinates, faces and normals of these voxels are saved in binary files in order to provide a suitable and efficient way to export them to the implemented Nvidia Optix Visibility Toolbox. This step is described in Sec. 3.4.2

4 ▶ Parallel Form Factor Evaluation Algorithm



(a) (b)

(c)

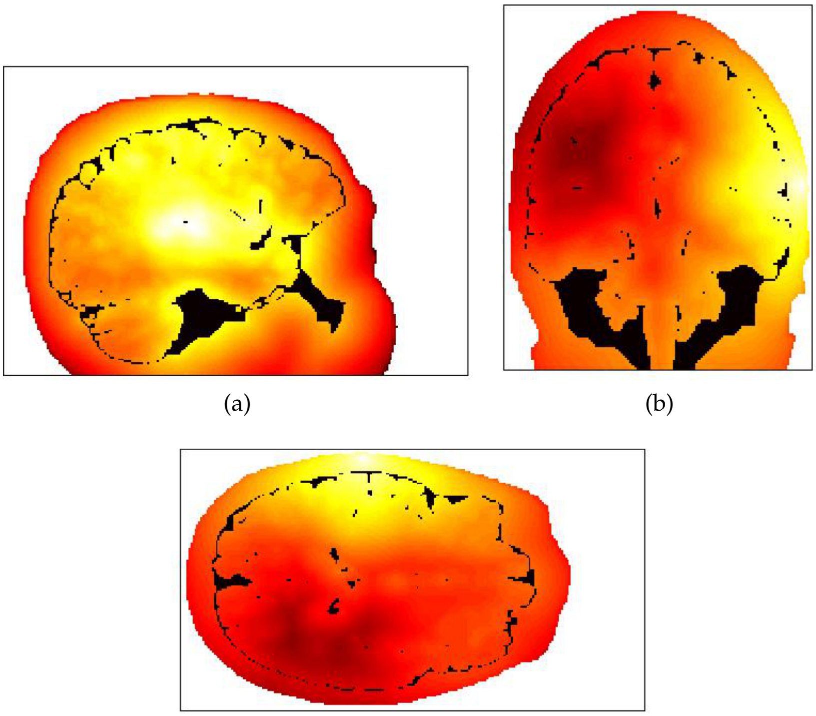
Figure 4.4: Section plot of ray tracing results on the realistic head model from the white voxel to every CSF-neighbour voxel (visible ones are in orange). This point is examined in Sec. 3.4.2.

5 ▶ Assign Guess Absorption Coefficient to every voxel of the Head Domain

Tissue	μ_a [mm^{-1}]	μ_s' [mm^{-1}]
Skin	0.047	1.590
Cranial bone	0.018	2.120
CSF	0.0022	0
Gray Matter	0.017	0.801
White Matter	0.079	5.175

Figure 4.5: Guess Optical properties of biological head tissues obtained by the mean of the value found in literature. The considered wavelength is 735 nm. In section 3.2 the optical properties of head tissues are explained on details.

6 ▶ Forward Solution to Predict Measurements



(a)

(b)

(c)

Figure 4.6: Forward problem solver results by using the RD iterative method approach. This step is analysed in Sec. 3.8 and it involves the diffusion matrix assembly and the forward problem solution computation illustrated in the work-flow of Fig. 3.23. Notice that the number of iteration required for convergence is minimum 20 for a total elapsed time of approximately 2 minutes. Due to the high absorption of light in the first layers, logarithmic plots of photon density are showed.

7 ▶ Jacobian Computation (S×M+1) Forward Solvers

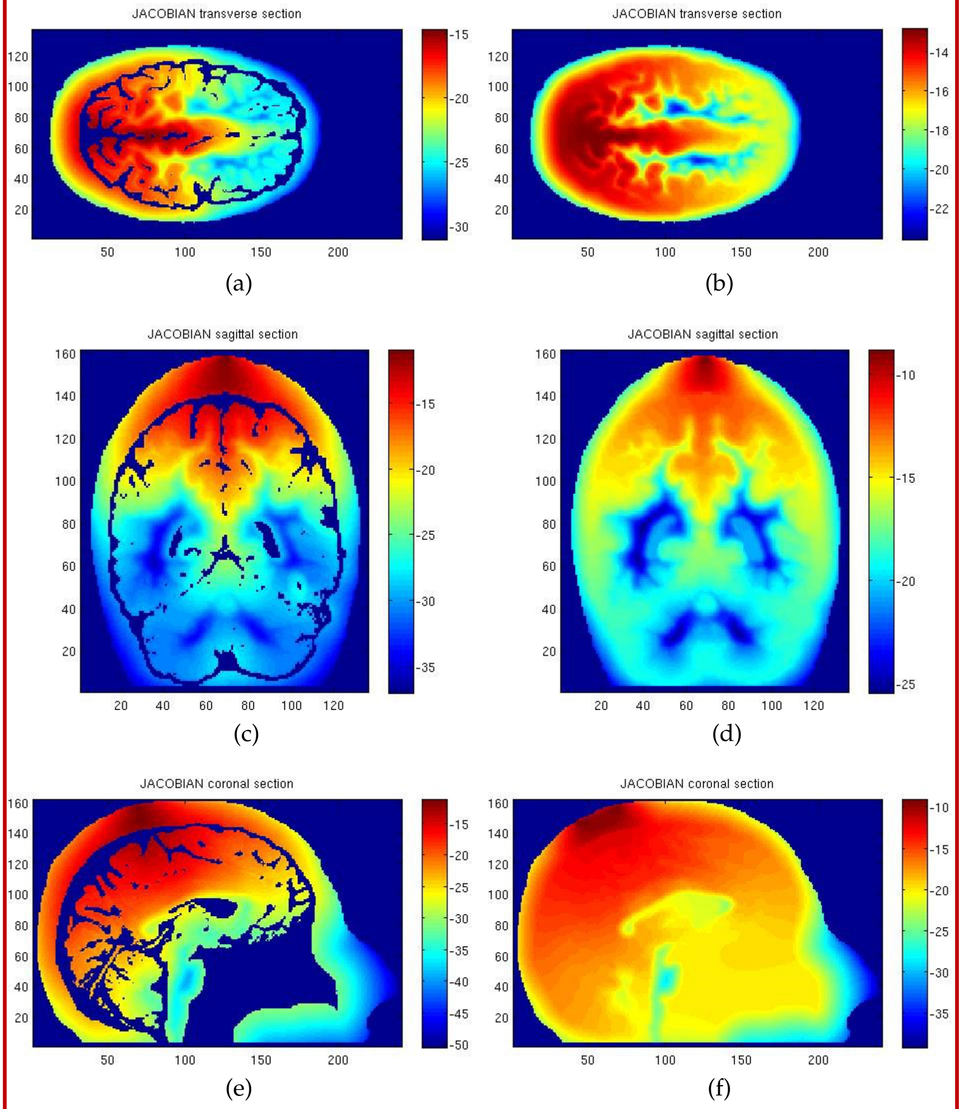


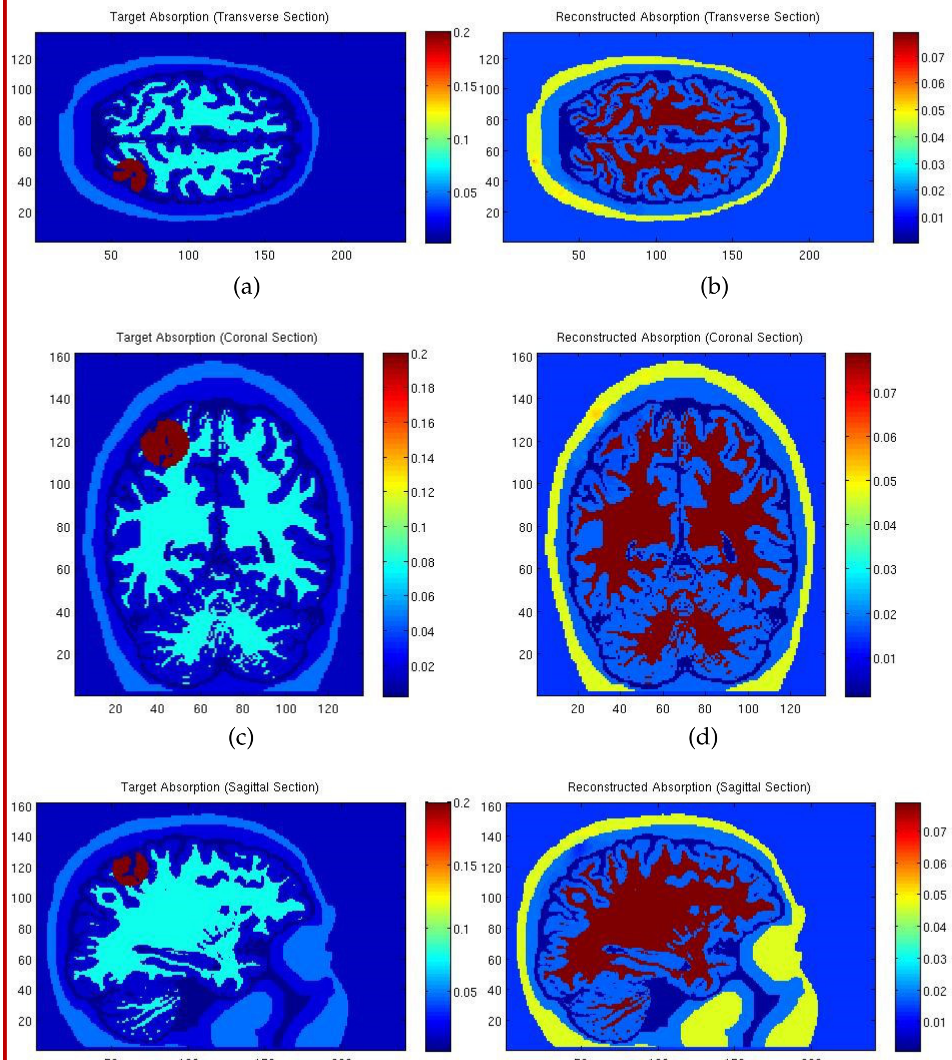
Figure 4.7: Comparisons between Jacobians calculated by using RD Iterative Model (left column) and standard Diffusion Model (right column). The second method doesn't take into account the non-scattering behaviour of CSF. This step is fully described in Sec. 4.2.

8 ▶ Compute $\Delta\mu_a$

$$\Delta\mu_a = lsqr(J, \Delta\Phi_i) \tag{4.12}$$

$\Delta\mu_a$ computation using MATLAB *lsqr* built-in function as shown in Sec. 4.2.

9 ▶ Update Absorption Coefficients of the Head



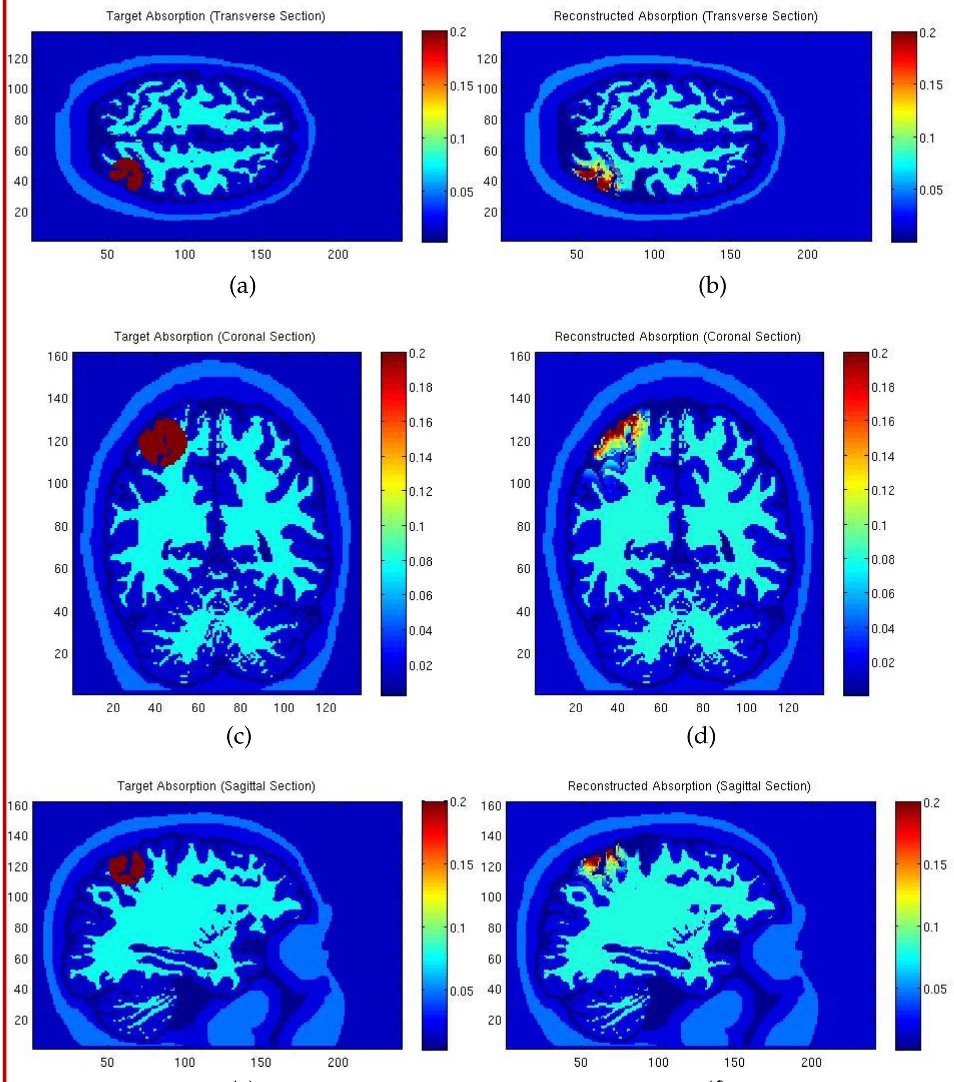
(a) (b)

(c) (d)

(e) (f)

Figure 4.8: Example of image reconstruction when an anomaly is introduced in the head domain. The anomaly affects only the white and the gray matter, it is composed by 5247 voxel with an absorption coefficient of $0.2\ mm^{-1}$. In this case we consider only absorption distribution reconstruction because it is high-correlated to the tissue and blood oxygenation. Left column shows the target absorption distribution, right column refers to the image reconstruction. Notice that, due to the adopted reconstruction technique, solution tends to be concentrated on the outermost layers and the detected variation of the absorption coefficient is rather small.

9 ▶ Update Absorption Coefficients of the Head



(a) (b)

(c) (d)

(e) (f)

Figure 4.9: Example of image reconstruction when an anomaly is introduced in the head domain. Left column shows the target absorption distribution while right column shows results obtained by forcing the solution to be confined in the brain regions, leading to a solution that fits very well the target.

4.4 Conclusions

An inverse problem solver based on the proposed forward problem solver described in Chapter 3 has been developed and described in this chapter. Due to the complexity of the full-resolution head domain, an adjoint method based on reciprocity principle was adopted to handle the large number of voxels composing the domain. Since light intensity decreases significantly before reaching cortical regions, an innovative technique has been implemented to avoid reconstruction being concentrated on the outermost layers by forcing the solution to be located in grey and white matter only, with the relevant improvement which can be appreciated in the comparison between image reconstruction of Fig. 4.3 and Fig. 4.3.

In Chapter 5, the implemented framework is used to validate DOT experimental measurements.

Chapter 5

Inverse Problem with Experimental Data

An ad-hoc software framework for simulation of light transport in both diffusive and clear regions is made available and described in Chapters 2, 3 and 4, being able to handle complex geometries with mm-range resolution. It is designed to efficiently run on multi-GPU/CPU system to significantly speed-up reconstruction of the optical properties without sacrificing resolution. Moreover, it has been used to verify the how common errors on measured data propagate to the output.

The objective of this chapter is to validate the system using a phantom made by materials of different reduced scattering and absorption coefficients (respectively μ_s' and μ_a) layered to mimic a human head.
First, in Sec. 5.1, we chose to perform a sensitivity analysis of the reconstructed parameters as a function of some deviations in the hardware behaviour with respect to an ideal one.
After that, in Sec. 5.2, we characterize the hardware composing the portable DOT system realized by STMicroelectronics, University of Palermo (UNIPA) and Catania Unit of the Institute for Microelectronics and Microsystems (CNRIMM). This platform is employed to perform the experimental measurements presented in Sec. 5.3; the proposed RD software framework is finally used to reconstruct the optical properties of the domain from measured data.

This work has been developed in collaboration with STMicroelectronics, UNIPA and CNRIMM in the scope of the "HIGH PROFILE" European project and it is fully described in [82].

5.1 Software framework for analysis of Hardware non-ideality

In order to perform a robustness analysis, we consider three main possible sources of error:

1. Deviations in the gain of the different detectors, which can arise in any point of the acquisition chain;
2. Deviations in the source optical power at the surface;
3. Instrumentation or external noise, which we model as an additive white Gaussian input referred noise.

We consider a toy problem which is related to the availability of uniform samples of the materials composing the phantoms. These are cubic and homogeneous samples of 47x47x47 mm, having the optical characteristics presented in Table 5.1.

In the following we consider the material composing sample 1, with an absorption coefficient of 0.019 mm^{-1} and reduced scattering of 0.904 mm^{-1}. In order to have a realistic comparison with measured data, we consider to have a probe which has the same geometrical properties of the hardware developed in the project and described in Sec. 5.2. We assume that one light source and four detectors are present on the top surface of the sample. Geometrical positions of source and detectors are visible in Fig. 5.1.

Sample ID	Tissue	μ_a [mm^{-1}]	μ_s' [mm^{-1}]
Sample 1	Skull	0.019	0.904
Sample 2	CSF	0.001	0.025
Sample 3	Gray Matter	0.017	0.577
Sample 4	White Matter	0.011	0.930

Table 5.1: Optical properties of the considered samples.

The triangular grid shown in the figure above is composed by equilateral triangles with side 15 mm. Caption L4 identifies the source while S5, S6, S7,

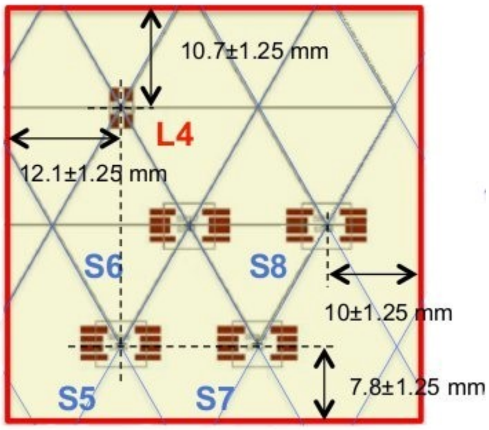
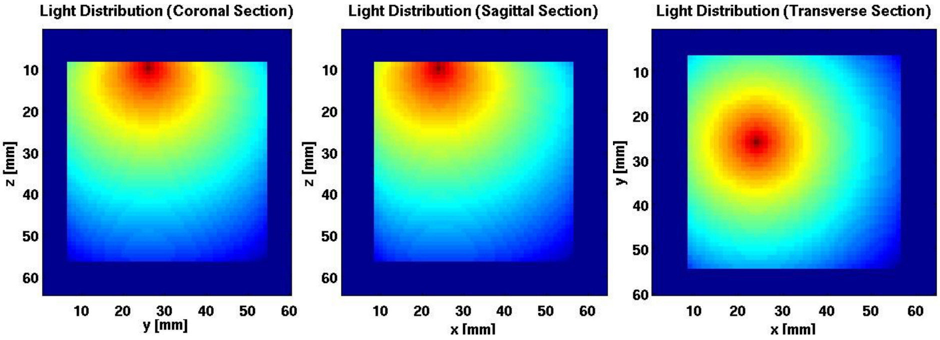


Figure 5.1: Detectors (S5, S6, S7, S8) and source (L4) placement on the cubic sample surface in the section $z = 0$.



(a) (b) (c)

Figure 5.2: Light distribution in three different sample sections corresponding to $x = 12$ mm (a), $y = 10$ mm (b) and $z = 0$ (c) with the experimental setup illustrated in Fig. 5.1

S8 indicate the detectors.

The resulting light distribution computed by the implemented RD numerical solver replicating the experimental setup of Fig. 5.1 is shown in Fig. 5.2.

We perturb the nominal solution above with the mentioned non-idealities at the detector position in order to test the sensitivity of the implemented numerical solver.

5.1.1 Deviations in the gain of the different detectors

The possible sources of gain errors could be:

- Inter-detector variations in the sensitivity;

- Intra-detector non-linearity in the sensitivity curve (non-linear relationship between output current and incident optical power);
- Mismatches in the amplification stages;
- Mismatches in analog-to-digital conversion.

A gain error is modelled by modifying the exact value of the nominal solution at the detector position as follows:

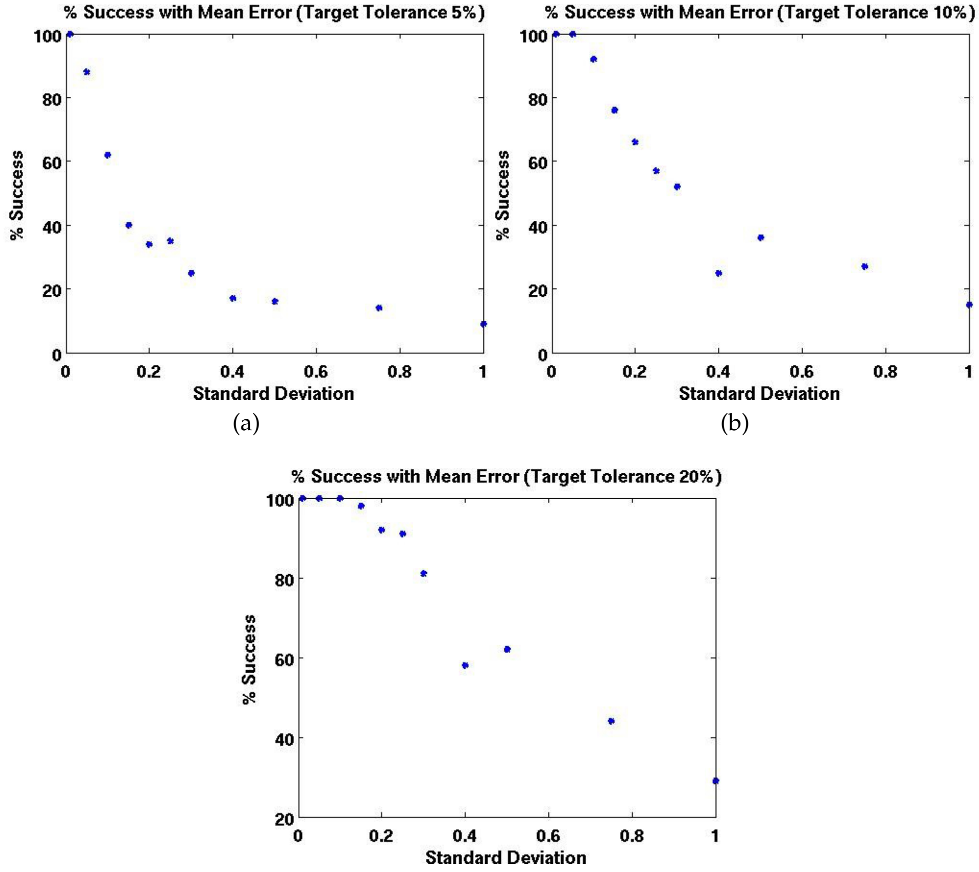
$$\Phi_{meas} = \Phi_{th}(1 + k). \tag{5.1}$$

where $k = (k_1, k_2, k_3, k_4)$ is a vector composed by samples extracted from a random Gaussian process with zero mean and standard deviation σ_k, Φ_{meas} is a 4-dimensional vector containing the 4 synthetic measures and Φ_{th} is a 4-dimensional vector containing the forward solver nominal solutions in the detector positions. The perturbations on each detector are different one from the other and come from different realizations of the random process. The reconstruction of the absorption coefficient of the sample is performed from the perturbed detected values and the solution is checked with respect to the nominal value. Different values of standard deviation σ_k and target tolerance are taken into account.

Results are shown in Table 5.2, where each value of Success Rate (SR) is obtained from statistical analysis of 100 runs. Results are summarized in Fig.5.3. It can be seen that the threshold for correct reconstruction of the

σ_k	$\mu_a \pm 5\%$ SR	$\mu_a \pm 10\%$ SR	$\mu_a \pm 20\%$ SR
0.01	100%	100%	100%
0.05	88%	100%	100%
0.10	62%	92%	100%
0.15	40%	76%	98%
0.20	34%	66%	92%
0.25	35%	57%	91%
0.30	25%	52%	81%
0.40	17%	25%	58%
0.50	16%	36%	62%
0.75	14%	27%	44%
1.00	9%	15%	29%

Table 5.2: Success rate for gain non-ideality.



(a) (b)

(c)

Figure 5.3: Success rate for gain non-ideality in function of the standard deviation of the Gaussian process.

absorption coefficient is approximately equal to the value of the uncertainty with which we accept the solution to be correct. This simple analysis allows us to infer that the inverse solver is stable with respect to this kind of perturbations and that the perturbation on the input is not amplified, rather kept almost the same at the output. Similar results have been obtained with similar setups and are not presented here for sake of compactness, allowing us to extrapolate these results to configurations which can be obtained with the developed hardware.

5.1.2 Deviations in the source optical power at the surface

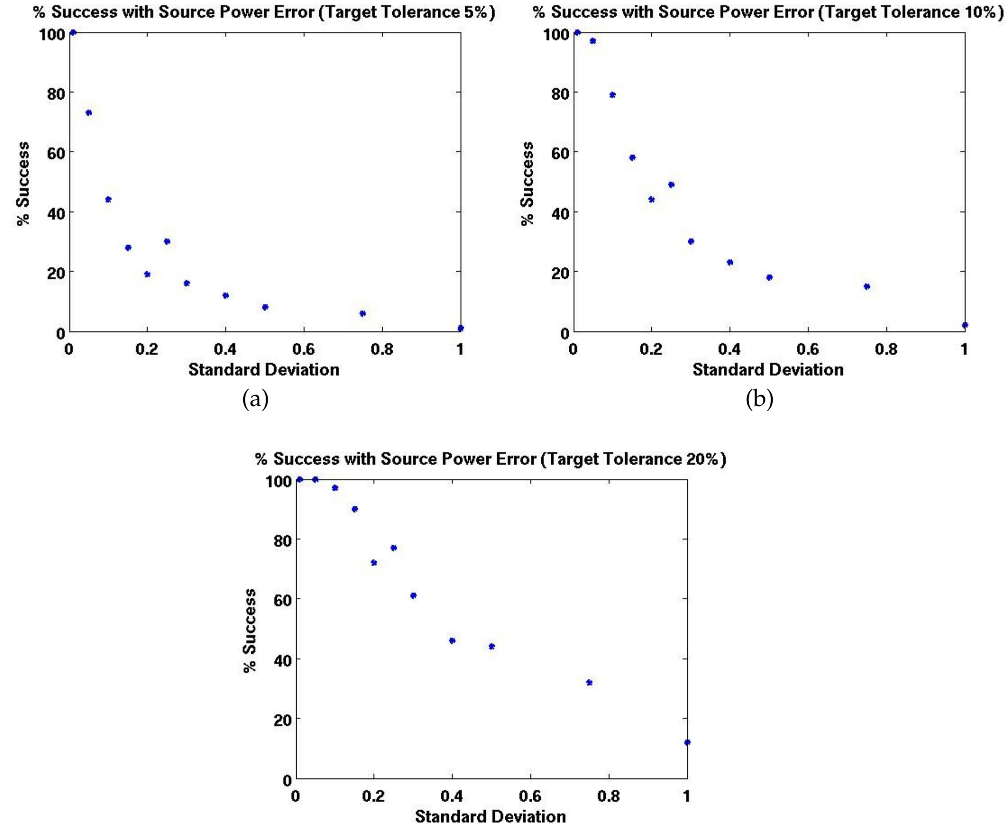
The second kind of perturbation which we consider is a deviation in the optical source power with a random Gaussian distribution with zero mean, that affects the theoretical measurement as follows:

$$\Phi_{meas} = \Phi_{th}(1 + h), \tag{5.2}$$

where $h = (h, h, h, h)$ is a sample extracted from a Gaussian distribution with zero mean and standard deviation σ_h.
The possible sources of this kind of errors could be:

- Deviations on different sources;
- Misalignments;
- Partial occlusions.

Each detector measure is perturbed by the same sample value. Results of these simulations are summarized in Table 5.3 and Fig. 5.4.
It can be noticed that the effect of a source power deviation leads to similar results with respect to the previous case, where a gain non-ideality was introduced. This kind of error has a slightly higher effect on the reconstructed properties.
However, taking into account a 10% target accuracy on the solution, a perturbation modelled by a Gaussian distribution with standard deviation equal to 0.25 (with respect to a nominal value of 1) still leads to almost 50% success rate.



(a) (b)

(c)

Figure 5.4: Success rate for source power non-ideality in function of the standard deviation of the Gaussian process.

σ_h	$\mu_a \pm 5\%$ SR	$\mu_a \pm 10\%$ SR	$\mu_a \pm 20\%$ SR
0.01	100%	100%	100%
0.05	73%	97%	100%
0.10	44%	79%	97%
0.15	28%	58%	90%
0.20	19%	44%	72%
0.25	30%	49%	77%
0.30	16%	30%	61%
0.40	12%	23%	46%
0.50	8%	18%	44%
0.75	16%	15%	32%
1.00	1%	2%	12%

Table 5.3: Success rate for source power non-ideality.

5.1.3 Instrumentation or external noise

The third kind of perturbation which we consider is additive white Gaussian noise on the detected signal and it derives from the following sources:

- Detector noise;
- Detector biasing noise;
- Acquisition chain noise (amplifiers, ADCs);
- Physiological noise (does not apply to the phantom).

The mathematical model of this error is the following:

$$\Phi_{meas} = \Phi_{th} + n; \tag{5.3}$$

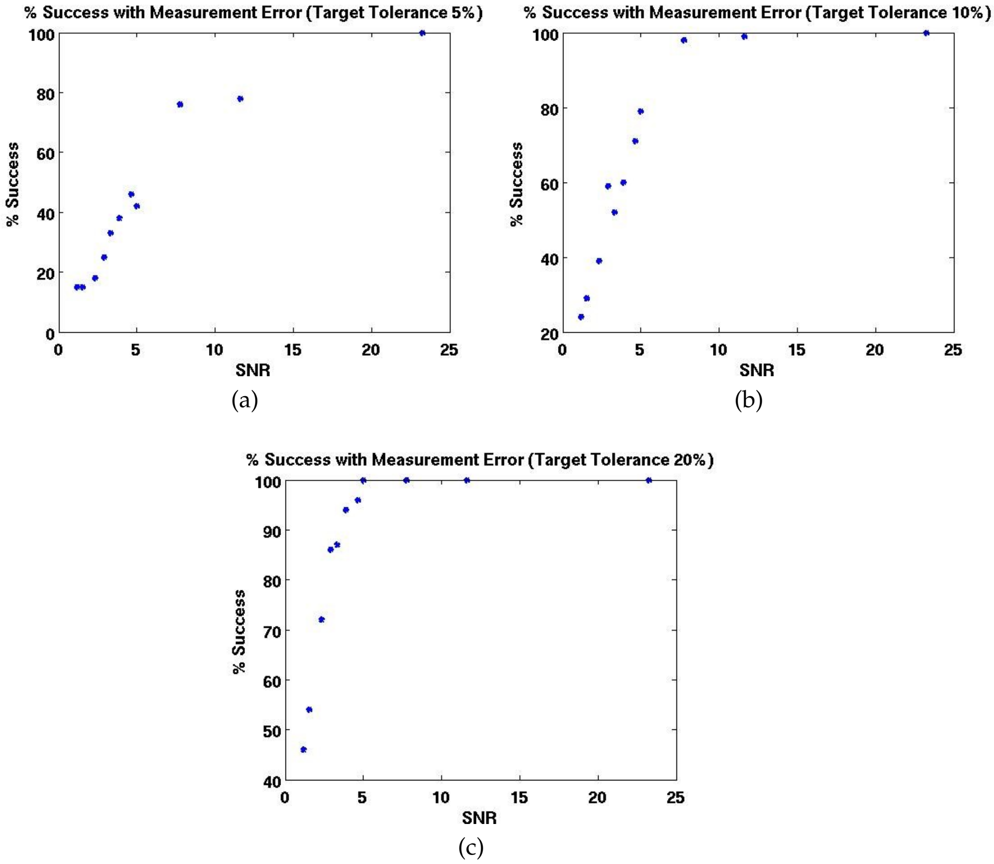
where n is a sample extracted from a random Gaussian process with zero mean and standard deviation σ_n. The perturbations on each detector are different one from the other and come from different realizations of the random process. Results are shown in Table 5.4 and in Fig. 5.5 in function of SNR that is computed as:

$$SNR = \frac{\sqrt{\frac{\sum_{i=1}^{4} \Phi_{imeas}^2}{4}}}{\sigma_n}. \tag{5.4}$$

Again, we can assume that the inverse problem solver is stable with respect to measurement noise. As an example, taking into account a 10% target

SNR	σ_n	$\mu_a \pm 5\%$ SR	$\mu_a \pm 10\%$ SR	$\mu_a \pm 20\%$ SR
23.26	0.05	100%	100%	100%
11.63	0.10	78%	99%	100%
7.75	0.15	76%	98%	100%
5.01	0.20	42%	79%	100%
4.65	0.25	46%	71%	96%
3.87	0.30	38%	60%	94%
3.32	0.35	33%	59%	87%
2.90	0.40	25%	52%	86%
2.33	0.50	18%	39%	72%
1.54	0.75	15%	29%	54%
1.16	1.00	15%	24%	46%

Table 5.4: Success rate for noise analysis.



(a) (b) (c)

Figure 5.5: Success rate in function of the SNR for noise analysis

accuracy on the solution, we can see that a SNR as low as 3 allows more than 50% correct reconstructions.

5.2 Hardware characteristics

In this Section we briefly describe the characteristics of the hardware system which has been employed to realize the experimental light measurements. The portable DOT system was realized by STMicroelectronics in the scope of the European project HIGH PROFILE.
The scope of this Section is to facilitate the interpretation of data analysis performed in Sec. 5.4.

5.2.1 STM SiPM detectors

The use of Silicon Photomultipliers (SiPMs) in a DOT system has been estimated potentially able to increase its spatial resolution. Dedicated SiPM sensors have been designed and fabricated by STMicroelectronics using an optimized process. The SiPM is a large area detector consisting of a parallel array of micro-sized Geiger Mode Avalanche Photodiodes (GMAPs) microcells with individual integrated quenching resistor. The diodes are biased above the breakdown voltage; the absorption of a single photon may trigger an avalanche current pulse; the avalanche current is quenched by an integrated quenching resistor. The current pulse amplitude does not depend on the number of photons firing the cell: it works as a digital detector. Each microcell, working as an independent photon counting device, is connected to a common output to produce a summation device whose output signal is proportional to the number of detected photons [83] [84]. Further details about the manufacturing method of STM SiPM technology are reported in [85]. The characterization results have been already reported in [86] [87][88][89][90] for 1 mm^2 area SiPMs with 324 microcells and 60 μm pitch.

5.2.2 DOT Embedded System

The design of the adopted DOT embedded prototype, hosting up to 64 IR LED sources and 128 SiPM sensors, is described in [76]. The system has been designed to realize DC and pulsed configurations and, by exploiting the small form factor of SiPM and LED sources, to cover the entire skull surface employing the above specified a high numbers optical components.

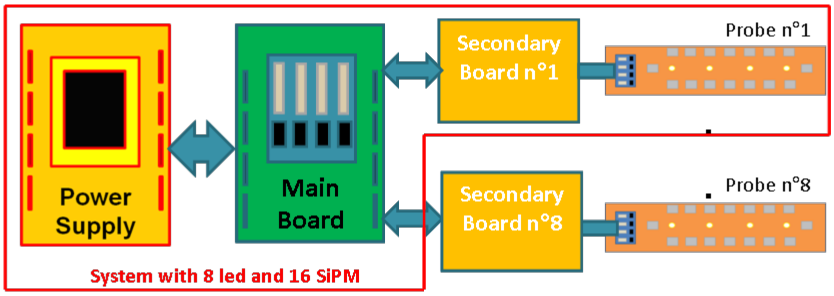


Figure 5.6: Main Reference Architecture of the portable DOT embedded system (source: [82]).

The adopted architecture is a scalable solution where every leaf consists of 8 modular and flexible probes, able to host 4 bi-color LEDs at two different wavelengths (735 and 850 nm) as light sources, 16 SiPMs as photo-detectors and a temperature sensor (see Fig. 5.6). The architecture can drive up to 8 secondary boards and probes, each hosting 16 SiPM and 8 LEDs, thus keeping a high level of design modularity. In particular the embedded system has been realized by subdividing the whole electronic design in several sub-platforms:

- a Main Board hosting a powerful ARM based microcontroller;
- a Secondary Board dedicated to the handling of a modular sub-system containing 8 IR LED and 16 SiPM sensors (Probe Board);
- a flexible Probe Board hosting the light peripheral and photo-detectors;
- a power supply delivering all the needed supply voltages (both for the analog paths and the digital chips) starting from a unique battery pack.

Each probe board is connected to the secondary board through a flexible flat cable. The secondary board is responsible for the whole multiplexing actions needed to implement the polling functionalities able to collect measurements from all LEDs and SiPMs.

In Fig. 5.7 a detail of the LEDs and SiPMs geometrical arrangement within a probe board is shown. All the devices are positioned at the vertexes of triangles, so distances of 1.5 cm or multiples are allowed. The red crosses, visible in the picture, represent the positions where sensors of an adjacent

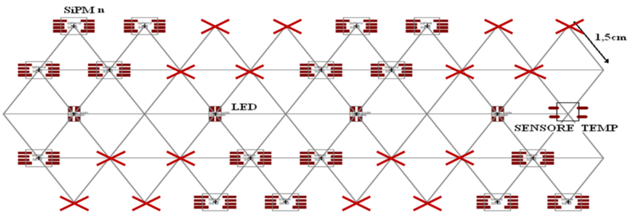


Figure 5.7: Detail of a probe board schematic.

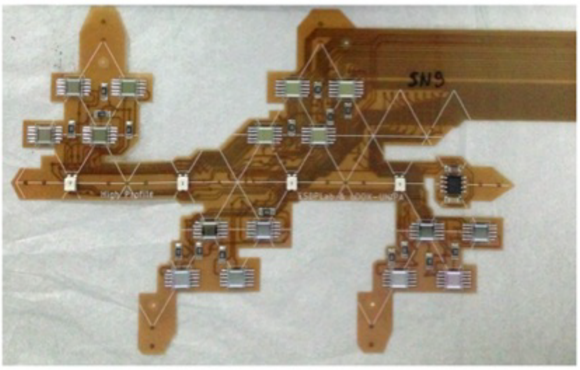


Figure 5.8: Picture of the probe board.

probe can be stitched together, thus obtaining a full uniform coverage. The probe board picture is shown in Fig. 5.8.

5.3 Measurements and data analysis

An optical phantom, mimicking a human head, has been employed to perform some experimental measurements.

The human brain tissue phantom consists in a cylinder with four different layers that mimic the optical properties of the different layers of a human brain: the skull, the cerebral spinal fluid (CSF), the grey matter and the white matter. The phantom is made of polyurethane which provides long term stability. Small particles of TiO2 were used as the scattering agents, while carbon black as absorbing dye. The thickness and the optical properties of the four layers are summarized in Table 5.1.

In addition to the optical phantom, four blocks with cubic geometry with size $4 \times 4 \times 4$ cm^3 has been realized. The phantom and the samples are shown in Fig. 5.9. The single block has the same optical property of each layers of the phantom.

STMicroelectronics in collaboration with UNIPA and CNRDIMM of Catania

Figure 5.9: Phantom and samples used for experimental measurements.

provides us a set of experimental measurements realized using two different setups:

- A portion of the probe (one source and 4 detectors) is placed on each of the four samples of the materials composing the four different layers of the phantom;
- A larger portion of the probe (three sources and 12 detectors) is placed on the phantom.

In both cases the number of measurements is higher than the number of unknown optical parameters which need to be reconstructed. The inverse problem described in Chapter 4 is determined and a unique solution can be found without the need for additional information (regularization). Measurements are performed in a dark ambient using a black and opaque screen to avoid ambient light to reach the samples and the phantom.

5.3.1 Experimental Measurements on Samples

Fig. 5.10 represents the schematic of the experimental setup for measurements on each sample with optical properties of each layer composing the phantom. Measurements were performed at three different bias voltages for the SiPM (-28.5 V, -29 V, -29.5 V) and for several different optical power of the source (16 steps of diode bias current from 1 μA to 1500 μA). Data were sampled at 166 Hz for 10 seconds so as to allow to verify stability and average data to remove high frequency noise.

Additionally, some measurements have been performed by STMicroelectronics and UNIPA to characterize the behaviour of sources (LEDs) and detectors (SiPMs).

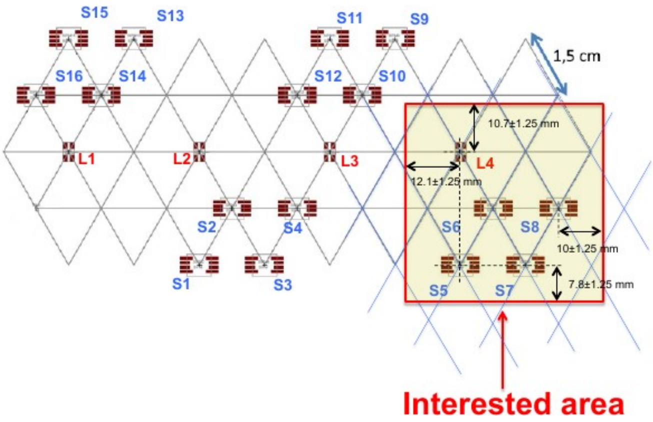


Figure 5.10: Schematic of samples experimental setup.

The breakdown voltage (V_{bd}) is the minimum bias voltage that leads to self-sustaining avalanche multiplication in Geiger-Mode avalanche photodiodes (GM-APDs). For $V_{bias} = V_{bd}$ both the detection efficiency and the gain of SiPMs are still null. Only for $V_{bias} > V_{bd}$ output current pulses are actually observed. The excess bias beyond the breakdown voltage is called overvoltage (V_{ov}). By definition:

$$V_{ov} = V_{bias} - V_{bd}. \tag{5.5}$$

In principle, the higher the overvoltage, the higher the SiPM performance. In reality, since the detection efficiency tends to saturate with V_{ov} while the noise keeps on increasing (even more than linearly) with V_{ov}, there exist an upper limit to the optimum SiPM bias voltage. Fig. 5.11 shows breakdown voltage variability among different SiPMs of the same probe. Breakdown voltages are easily extracted on an I-V plot, by notice the voltage value for which a sudden increase in current is present.

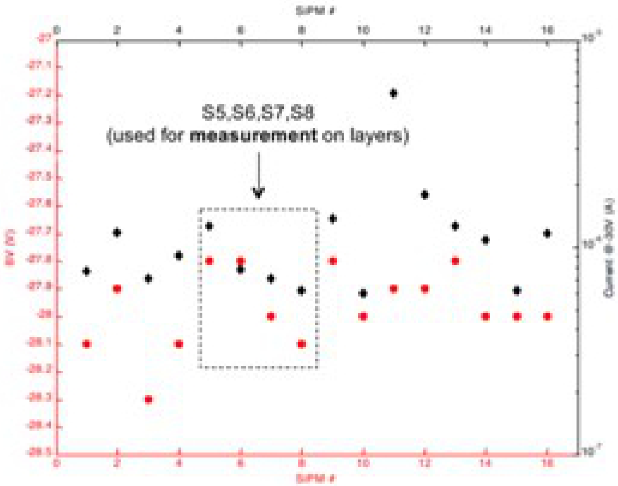


Figure 5.11: Variability of breakdown voltage among the different SiPMs.

5.3.2 Experimental Measurements on Phantom

Fig. 5.12 represents the schematic of the setup for phantom measurements, in particular the portion of the probe which is placed on the top of the phantom.

Measurements are performed in a dark ambient using a black, opaque screen to avoid any ambient light to reach the phantom.

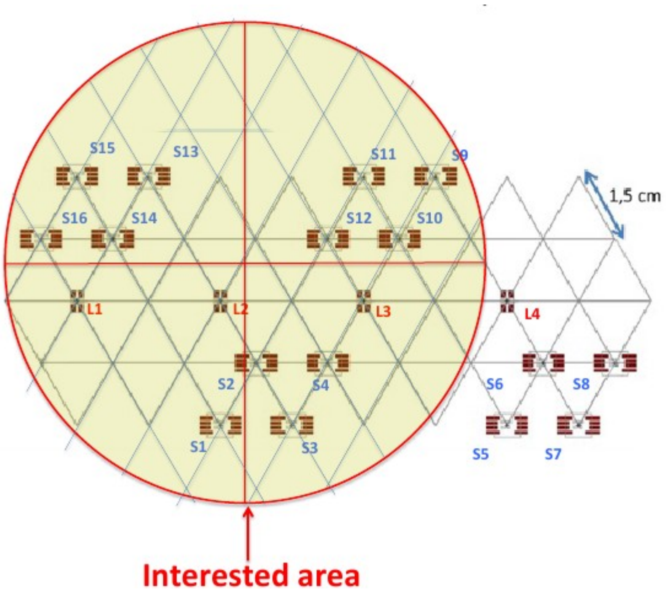


Figure 5.12: Schematic of the phantom experimental setup.

5.4 Data analysis

Data were stored in different files, each file is characterized by the same bias voltage for the SiPM and LED bias current. Data are relative to both LED wavelength and sets of 10 to 15 seconds of data are acquired for each SiPM-LED pair of the probe. Data is sampled at 166 samples per second and it is imported in MATLAB, where suitable domains are modelled to verify agreement between measured and simulated data and provide reconstructions of the optical properties.

The spectral content of the data acquired from one SiPM on a sample of Layer 1 underlines that the noise is more or less evenly distributed, without the presence of significant content at particular frequencies. In time domain, noise power is approximately stable and relatively high (up to 20% of the measured signal) and a low-pass filtering of the signals is necessary to provide reliable data for post-processing.

Fig. 5.13 shows the detected signal on four detectors on a sample of layer 1 as a function of the LED bias current. It is quite clear how the system exhibits a highly non-linear behaviour.

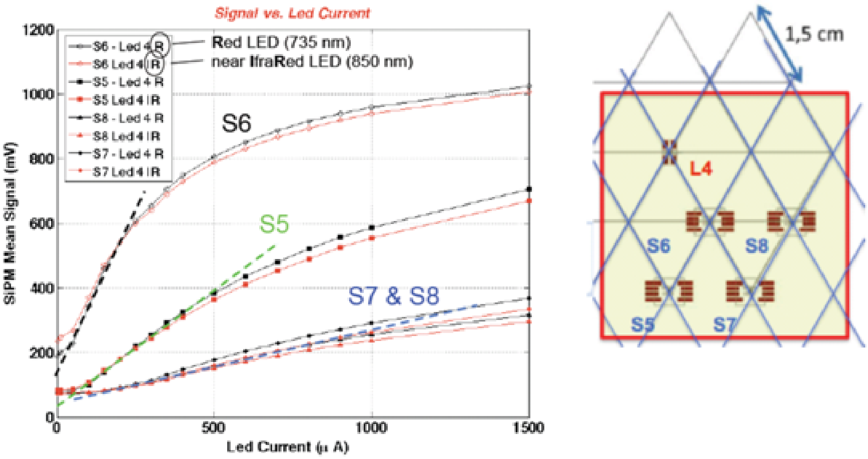


Figure 5.13: Output SiPM signals measured on a sample of layer 1, bias voltage -28.5 V.

From section 5.3, we can expect the LED emitted optical power to be roughly proportional to its bias current.

Since we can expect the domain to have a linear behavior as well, the non-linearity needs to be due to the detector behavior. This is particularly troublesome since SiPM at different positions are hit by significantly different optical power, therefore one can not rely on relative measurements among different SiPM to provide reconstructions.

Additionally, we can observe how S5 and S8 are at the same distance from the light source L4, nevertheless, the ratio between measured value on the two detectors is approximately 2. According to our simulations, this can not be justified by effects at the border of the domain and the expected ratio between the detected signals should be approximately 1.

Given the fixed value of the bias voltage and the large variability of the breakdown voltage, there is a deviation of approximately 400 mV in the overvoltage between the two sensors. This is probably responsible for the high variability in the measurements.

Fig. 5.14 shows a similar analysis for higher bias voltage. We can observe how there is still a misalignment between signals detected on S5 and S8, however the deviation is smaller.

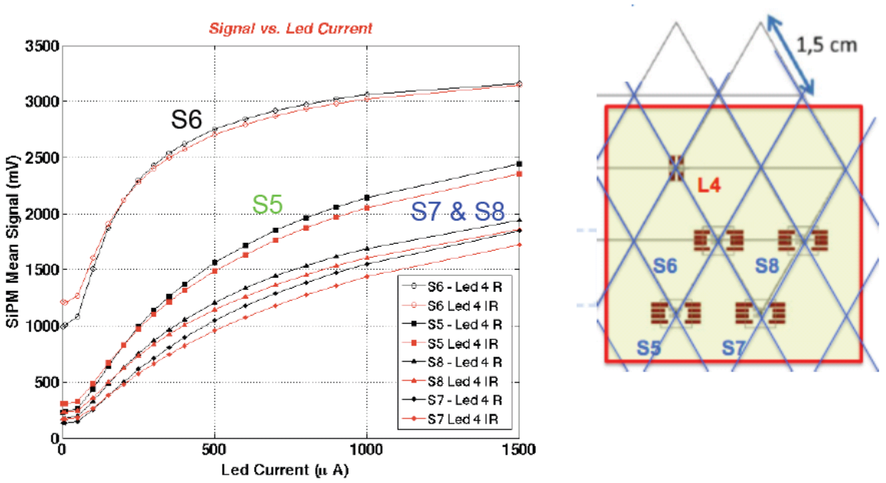


Figure 5.14: Output SiPM signals measured on a sample of layer 1, bias voltage -29.5 V.

This is consistent with the previous analysis since for higher bias voltage, the effect of the breakdown voltage variation on the overvoltage will be smaller.

Nevertheless, a deviation of more than 30% is still present. Additionally we can observe how all detectors behave highly non-linearly with respect to incident optical power. As an example S6 detected value increases by less than 10% as a consequence of a theoretical simulated 100% increase of the incident optical power.

These two effects combined lead to errors in the reconstructed optical properties which can go up to 100% or more, due to the virtual impossibility of the algorithms to converge to a solution which justify measurements. This is in accordance with Sec. 5.1 and is directly due to the fact that, even if measurement error propagates almost un-amplified from the input to the output of the reconstruction algorithms, measurements themselves have non-idealities in the range of more than 50%. Similar results are obtained

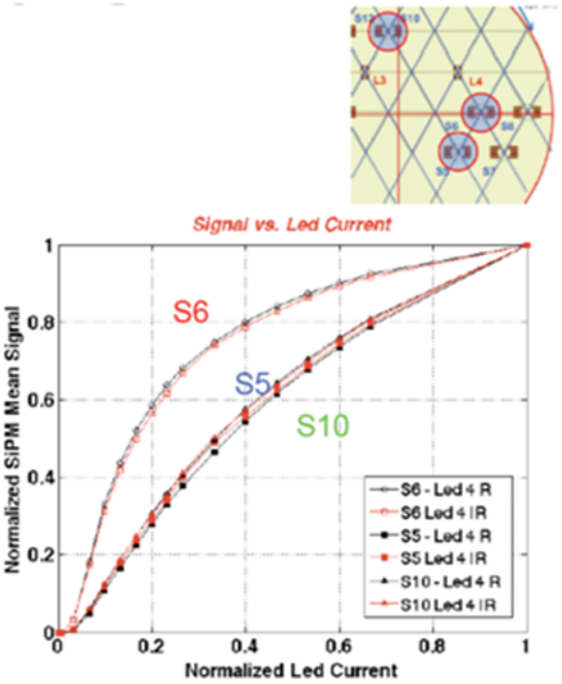


Figure 5.15: Output SiPM signals measured on the full phantom, bias voltage -29.5 V.

for measurements performed on the full phantom. In Fig. 5.15 we see that in this case S5 and S10 share correctly similar output signals having the same distance from source (L4). However we can still see that the detectors, especially S6, are operating in a highly non-linear region, leading again to a virtual impossibility to perform meaningful reconstructions.

5.5 Conclusions

In this chapter, the proposed software framework for DOT optical properties reconstruction was adapted to be used for the validation of measurements obtained with an experimental test. The proposed inverse problem solver was verified to be able to reconstruct the optical properties of a sample starting from synthetic measurements obtained by the perturbation of the numerical solution. The robustness of the algorithm versus non-idealities (noise, gain mismatches, non-linearities) was verified, resulting in an error on the reconstructed optical properties which is in the order of that on the input data.

Hardware system was characterized by STMicroelectronics,UNIPA and CNRIMM, some non-idealities where detected such as high noise level, high detector non-linearity and high spread in breakdown voltage, leading to high spread on over-voltages and on output currents.

While noise level can be reduced by filtering, the gain error in the detector signal acquisition chain can be 100% or higher. Coherently to simulations on synthetic data, this leads to error in the reconstructed properties of approximately the same value, making it impossible to attempt meaningful reconstructions, identifying some non-idealities in the experimental setup.

Possible improvements in the measurements quality could be the reduction of the light emission from LEDs to prevent the detector non-linearity due to its saturation and a more accurate selection of the sensors to solve the gain spread issue due to breakdown voltage spread.

Chapter 6

Conclusions Section A

In Section A of this thesis we have shown the main contribution of my research activity. It concerns the development of a complete software workflow for DOT and MRI data fusion based on highly parallel algorithms for improved image reconstruction. The aim is to efficiently include structural information derived from structural MRI to better evaluate the distribution of the optical properties within the head. Final objective is then to obtain information about local tissue and blood oxygenation, correlated to brain activity of specific regions.

One of the most significant improvement introduced by the implemented DOT toolbox is related to the possibility to accurately represent diffusive and clear regions with different physical and numerical models, having at hand structural information on the boundaries between regions.

The main results of Section A could be summarized as follows:

1. The anatomical information provided by MRI improves the results delivered by algorithms of optical properties image reconstruction;

2. MRI information is also relevant for the suitable modelling of light propagation, allowing to distinguish between diffusive tissues and clear regions like CSF;

3. Light propagation algorithms solution highly benefits from highly-parallel implementation. Particularly significant is the fact that the very same GPU cards are used both for the linear system solver associated to the diffusive problem and the computation of form factors associated to ray-optics part;

4. Since light intensity decreases significantly before reaching cortical regions, inverse problem solvers need to implement techniques to avoid reconstruction being concentrated on the outermost layers. An innovative approach has been implemented to avoid this issue by forcing the solution to be located in grey and white matter only, with a remarkable improvement in the accuracy of reconstruction;
5. The proposed inverse problem solver was verified to be able to reconstruct the optical properties of a sample starting from synthetic measurements obtained by the perturbation of the numerical solution. The robustness of the algorithm versus non-idealities (noise, gain mismatches, non-linearities) was verified, resulting in an error on the reconstructed optical properties which is in the order of that on the input data;
6. The proposed software framework for DOT optical properties reconstruction was adapted to be used for the validation of measurements obtained by an experimental test, identifying some non-idealities in the experimental setup leading to high noise level, high detector non-linearity and high spread in breakdown voltage, making it impossible to attempt meaningful reconstructions.

Possible improvements for a future work could be the reduction of the light emission from LEDs to prevent the detector non-linearity due to its saturation and a more accurate selection of the sensors to solve the gain spread issue due to breakdown voltage spread.

Part II

Section B : EEG for BCI systems

Chapter 7

ElectroEncephaloGraphy & BCI systems

As introduced in Chapter 1, EEG is a monitoring technique commonly applied in neuroscience research and clinical routine to explore human brain behaviour.

EEG is based on the measurement and mapping of electrical activity produced by the brain as it is being recorded by electrodes placed on the scalp. The recorded data are plotted into a standard chart (electroencephalogram) and examined by the physicians.

However, in the past years brain imaging techniques have been presented based on the elaboration of data recorded by EEG, in order to avoid high costs of electromagnetic-based solutions, e.i. fMRI. Aim of this new field of application, known as EEG source imaging, is to provide functional images of neuronal activity in the human cerebral cortex (gray matter), in particular for the localization of active areas at each time-frame (source localization).

To retrieve meaningful insights from these measurements, EEG brain imaging relies on detailed knowledge of the morphology of the subject head volume, which determines the scalp voltage distribution due to brain activity. This is obtained from numerical models of the electric field propagation in the head, whose computation is very time-consuming and computationally intensive.

The objective of this chapter is to give principles, theoretical basis and applications of EEG to understand the reserach activity explained in Chapters 8, 9 and 10.

7.1 Physiological nature of EEG

EEG brain imaging is a technique based on the estimation of cerebral activity, starting from the recording of potentials directly generated by neuronal activity. For this reason, a short overview of the physiological nature of EEG is outlined in the following paragraph.

7.1.1 Neuronal activity

Neurons are electrically excitable cells, whose activity consists essentially in the switching between two states:

- a rest state, where neurons are in a stationary electrical condition characterized by a constant potential difference across its cellular membrane given by different ion concentration;

- an active state, where neuron transmit along its axon impulses (action potentials) at a particular frequency (firing rate) exploiting a precise activation pattern for the ionic channels aimed to balance ionic disequilibrium across the membrane.

When in an active state, the instigated behaviour causes an electric field, propagating in the surrounding (conductive) medium, that is supported by an extracellular current density owing between the axon extremities. Electric field generated results in a potential distribution through the head, which can be theoretically recorded by passive electrodes placed on the scalp. However the time course of the action potential is too short (0.3 ms) to be detected from EEG electrodes, although it can be large in amplitude (70-110 mV) [91][92]. Transition between the two states is due to the occurrence of post-synaptic stimuli (PSP). PSP are stimuli sent by connected neurons at the dendrites; they flow towards the soma and along the axon, eventually determining a change of the neuron state. Differently from active potentials, PSP have a large time course (10-20 ms), although a lower amplitude (0.1-10 mV), which make them detectable by EEG electrodes [93].

7.1.2 Limits of electric field detection from the scalp: the pyramidal cells

In reality, the morphology of the head and the different order of magnitude between the electrical source (neurons) and the recording device (finite size electrodes) makes it impossible to detect activity of a single neuron switch. In fact, an electric field generated by a single neuron is too weak to be detectable by scalp electrodes, which are large and remote. Furthermore the propagation of electric fields (and of associated current densities) between the scalp and the brain is strongly limited by the skull, as it is more resistive than other head tissues provoking a considerable shunting effect on the current flow. What is more, the resulting field produced by randomly-oriented neurons can be considered practically null, because all the contributes tend to cancel each other out.

Given these limits, EEG electrodes can only detect summed activities of a large number of neurons that are coherently oriented along a common direction and synchronously electrically active [93]. These conditions are verified for a particular class of neurons located in the gray matter: the pyramidal cells. As shown in Fig. 7.1 pyramidal cells are a type of neurons located in the gray matter aligned perpendicularly at the brain surface. These cells have an activity locally synchronous, which results in a global electric field detectable by extra-cranial electrodes in the order of μV [20], so three orders

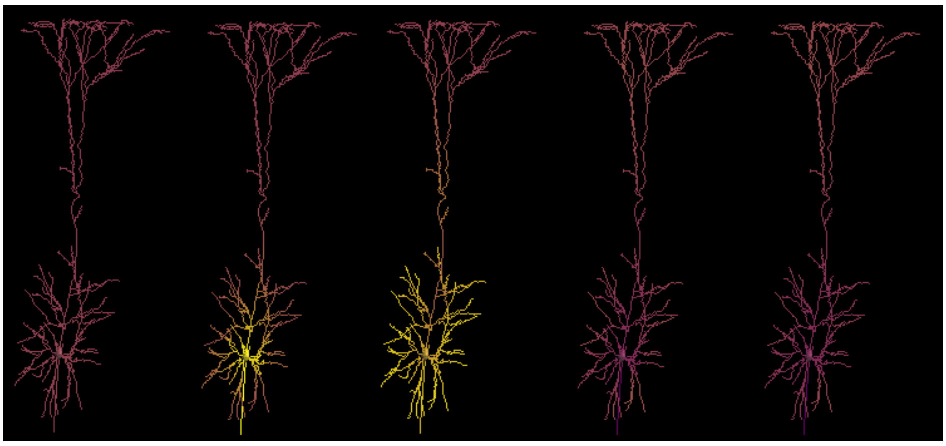


Figure 7.1: Pyramidal cells in brain cortex. Neurons are placed perpendicularly to the cortex surface; their activation is synchronous, resulting in a global electric field that is detectable from the scalp surface.

of magnitude smaller than the amplitude of the sources of a single neuron (mV).

EEG sources space results thus composed by a discretization of the gray matter in *voxels* (elementary cubic volumes) enclosing pyramidal neurons networks.

7.2 EEG Applications

To retrieve meaningful insights from these measurements, EEG brain imaging relies on detailed knowledge of the morphology of the subject head volume, which determines the scalp voltage distribution due to brain activity. This is obtained from numerical models of the electric field propagation in the head, whose computation is very time-consuming and computationally intensive (*EEG forward problem* [94]).

EEG is a technique that can provide high temporal resolution as the detectable neural activity is concentrated at low frequencies, usually below 30 Hz. Typical EEG waveforms are localized at specific frequencies, usually referred to as rhythms [95] and are indicative of the patients state (deep/light sleep, awake etc.). These rhythms are usually classified as follows:

- Delta rhythm: 0.5-4 Hz;
- Theta rhythm: 4-8 Hz;
- Alpha rhythm: 8-13 Hz;
- Beta rhythm: 13-30 Hz;
- Gamma rhythm: 30-100 Hz.

An example of brain rhythms is shown in Fig. 7.2.

Classical EEG analysis is based on the empiric examination of electroencephalograms and the spectral analysis of the basic rhythms, which finds application especially in subject monitoring, as in diagnosis of epilepsy; for example Fig. 7.2 shows as an epileptic foci which causes a characteristic periodic waveform at 3 Hz.

EEG is also used to measure event-related potentials where brain waves are triggered by an external stimulus which could be visual, auditory and somatosensory and find more field of applications in pre-surgical treatment,

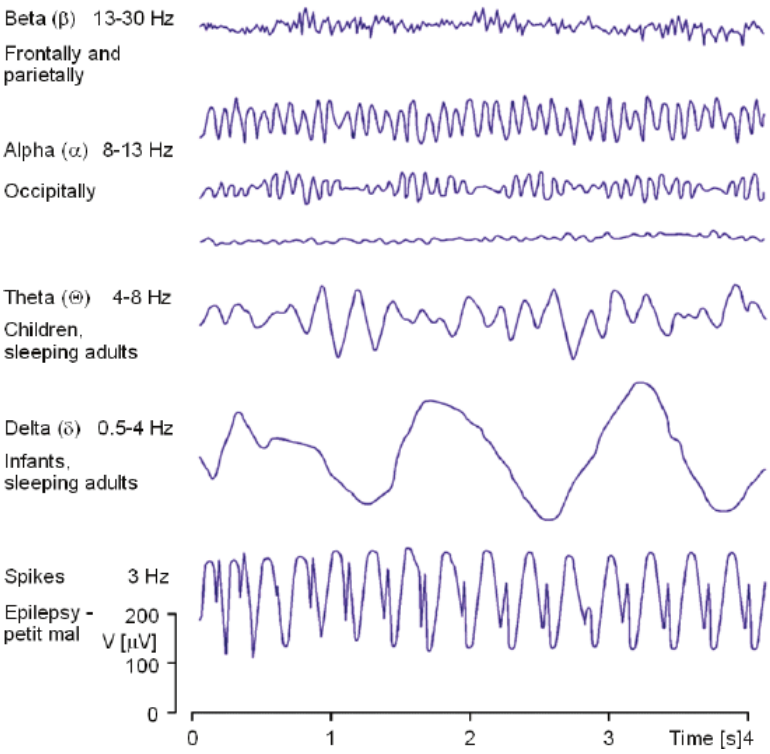


Figure 7.2: EEG rhythms. All EEG brain rhythms are at frequency less than 30 Hz. Periodic wave at the bottom is generated by an epileptic foci (source: [96]).

neurofeedback and BCI.

In the pre-surgical treatment of epilepsy it has been proven as a high resolution EEG source imaging is a valuable non-invasive functional neuroimaging technique [97].

The speed, ease, flexibility and low costs of this technique warrant its use in clinical practice. EEG is also used in neurofeedback application, where brain activation maps of the patient are computed and shown to him in real time. This creates a direct interaction between the subject and his neural activity, allowing him to try to modify his cerebral activity.

Advantages of EEG-based neurofeedback training have been proven by some studies as in the cases of severe palsies [98], in treating psychological disorders such as attention deficit/hyperactivity disorder (ADHD) [99], neurological disorders [100] as well as in the improvement and the influencing of improve cognitive performances in healthy subjects [101][102].

As shown in Chapter 1, recently EEG applications converge to Brain-Computer

Interface (BCI) systems [103] and also consumer-oriented applications ranging from home care to neurofeedback and gaming controllers.

7.2.1 BCI systems

As introduced in Sec. 1.3, BCI is a communication system interfacing the human brain to external devices, like computers or actuators (see Fig. 7.3). User commands are formed by recognizing brain activity with EEG and voltages measured by electrodes are sent to a computer. Data are interpreted to compute actuators commands and the feedback is closed by the subject's perception of actuator actions or movements. Signal processing and actuator actions has been performed by software toolboxes and depends on the objective of the experiment.

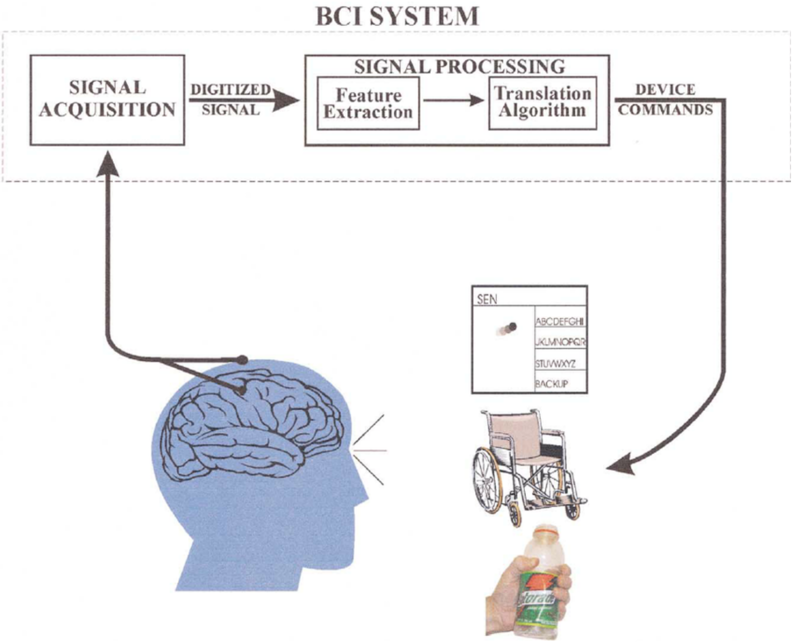


Figure 7.3: BCI neurofeedback (source: [103]).

Several real-time and open-source software platforms which allow designers and scientists to setup and execute BCI experiments in real and virtual environments have been introduced in the past few years. Examples of these platforms are BCI2000 [104], OpenVIBE [105] and BioSig [106].

BCI2000 [104] is a general purpose platform capable of potentially incorporate any brain signals coming from a set of electrodes on the scalp, process these signals to extract specific features that reflect the user's intent (e.g. amplitude of evoked potentials) and then translate them into commands that operate a device (e.g. word processing program).

OpenVIBE [105] is an open source software platform which consists of a set of software modules that can be integrated to design BCI for both real and virtual reality applications. It proposes a user-friendly graphical language that allows non programmer to design a BCI experiment without writing any code.

BioSig [106] is a toolbox which offers several modules and algorithms for signal processing and real-time BCI.

An alternative to these BCI-oriented tools is represented by general purpose tools, such as SIMULINK (MathWorks, Inc. Natick, MA) which is a graphical programming environment widely used to design, simulate and auto code software for different scientific fields. A Simulink model is a hierarchical representation of the design of a system using a set of interconnected blocks. In this context it can be used to acquire, process and extract signal features. In addition, it allows the user to easily transfer data into the MATLAB environment for a more accurate post processing using for example the open source toolbox BioSig [106].
Connected to one of the commercial EEG systems, all of these BCI software tools allow one to analyze neurophysiological signals in real time, or to develop applications capable of providing practical assistance for patient diagnosis, treatment, and rehabilitation [107, 108].
Despite of these useful applications, the dissemination of BCI systems is limited due to the drawbacks of EEG systems. The majority of these are high-expensive [17], thus not affordable even for research centres and universities. Also a few low-cost systems in the range of $500-$1000 (USD) are supported by a large part of BCI software tools. However these systems are usually equipped with a small number of channels, have moderate to high noise and users cannot modify the position of the electrodes on the scalp.
The main issue of these systems is the lack of a direct connection between BCI software tools and hardware implementation. There are no standard

libraries that allow one to connect them to the previously described BCI software tools. Hence this lack makes it difficult to use these hardware outside of the laboratory in which they were designed.
In addition, without stable software libraries which provide a way to connect the system to a BCI software tool, it is difficult to assess how the system functions and indeed whether it functions at all. If the system does not perform as expected it can be complicated to determine if the fault is due to a hardware or software issue and the time spent in trying to identify the source of error and correcting it can be substantial.

7.2.2 Creamino

In the scenario described in the previous section, the ARCES research team developed an Arduino-based cost-effective EEG system called *Creamino*, which has a fabrication cost of about 50 euros for the first 8 channels (working system, including wet-contact active electrodes) and 30 euros per each additional 8 channels. These numbers are particularly attractive for systems designed to be used outside clinical environments, such as in home care or research-oriented applications. In addition a set of libraries which allows the system to be used in a variety of software environments has been developed.
This work was developed in the scope of the European Project named CREAM. The project is focused on the multidisciplinary study of the neural substrates of creativity in different knowledge domains.
Creamino consists of a hybrid hardware/software platform capable of quickly linking the analog front-end (AFE) circuits for biopotential measurements and the PC used to acquire, visualize and process the EEG signal. *Creamino* allows one to reduce the time and the effort required to complete a new design thus leading to a rapid prototyping of an EEG-based BCI system.
Specifically the contribution of my research team can be summarized as follows:

- A sample design of a microcontroller system supporting the connection with the AFE, extract the EEG signals and transfer them to a PC;

- A custom made acquisition software with a graphical user interface (GUI) that can be used to acquire, filter, process, classify, visualize and store up to 64 EEG channels in real-time;
- A set of libraries and drivers running on a standard PC which allows to connect the system to BCI2000, OpenViBE, MATLAB and SIMULINK;
- A set of SIMULINK libraries for EEG Source Localization;
- A set of examples that shows how to easily setup a system and perform experiments.

Schematics, gerber files, source code and software modules of *Creamino* are available with full documentation and free of charge for research and educational purposes online at https://github.com/mchiesi/Creamino.
My contribution concerns the development of SIMULINK libraries for EEG source localization which are able to compute the mean power of the EEG sources lying in user-defined regions of interest (ROIs).

7.3 EEG Source Localization

As introduced in Chapter 1, the EEG computational work flow can be divided into two tasks: forward problem and inverse problem.
A forward problem is defined as the determination of the voltage distribution in the volume under analysis, given its electrical properties (local conductivity) and the sources (active voxels) distribution in the gray matter. Head volume in EEG data elaboration can be managed as concentric compartments structure, where compartments are defined by the principle head tissues (brain, skull, skin), enclosing the electrical sources volume. Due to the complexity of the domain, analytical solutions are not feasible, and one needs to rely on numerical solvers.
Computational resources that are required by a realistic head model are high, thus, simplified head geometries based on concentric spherical shells are often used; however, the limits of this approach in respect to realistic head model are proven [109] [110], also in medical usage [38].
What is more, several studies demonstrate that EEG data elaboration is influenced not only by approximations on the compartments surface, by

also by irregularities on them like varying thickness, irregular curvatures and holes in the skull [111].

The inverse problem, on the other hand, is defined as the estimation of the electrical sources distribution inside the gray matter, given the electrodes measurements and the relationship between the voltage and voxels distribution defined by the forward problem. Given the relative small number of passive electrodes, generally from 32 to 256 in proportion to the voxels number (in our workflow $\sim$ 200.000), this problem is ill-posed and with many solutions. So an estimation is necessary that takes into account a-priori informations in order to select the more realistic and physiologically correct solution among the others. The mathematically inverse problem is so defined by an overdetermined system (more unknowns than equations), which can be solved with a numerical approach based on functional minimization coupled with regularization techniques.

7.3.1 EEG Forward Problem Theory

Starting from the quasi-static Maxwell equations, localizing the neural current source generators of EEG basically breaks down to solving the equation:

$$\nabla \cdot (\sigma \nabla \varphi) = \nabla \cdot j \tag{7.1}$$

Where σ is the conductivity tensor, φ is the potential distribution and j is the current density vector field in the medium, in this case the head. Quasi-stationarity is a fair assumption considering the low frequency range of EEG generators.

There is only a limited amount of potential measurements, so we can transfer φ into a discrete vector $\mathbf{\Phi}$ of length N_e containing all instantaneous sensor measurements, with N_e being the number of electrodes. For technical feasibility, also the current sources must be discretized. Picking individual neurons for this is unfeasible, instead, a macroscopic current dipole model is commonly chosen to segment the cortical grey matter into an uniform array of discrete current sources, called “voxels”. The result is a vector $\mathbf{J} = (J_1, J_2, J_3, \dots J_{N_v})$, with J_i being the 3-dimensional current density vector at voxel i and N_v being the number of voxels.

The solution to the (discretized) forward problem is a linear map $\mathbf{K}$:

$\mathbb{R}^{N_v} \times \mathbb{R}^3 \rightarrow R^{N_e}$, mapping current densities to scalp potentials such that

$$\mathbf{J} = \mathbf{K\Phi} + \mathbf{c1}, \tag{7.2}$$

where $\mathbf{K}$ is commonly referred to as the electric lead field. The **c1** term is added due to the fact the electric potential is defined only up to an arbitrary constant; **c** is a scalar and **1** denotes a vector of ones. The linearity holds due to the superposition principle. Solution of the forward problem (i.e. obtaining the coefficients of $\mathbf{K}$) strongly depends on the volumetric conductivity model of the head and can be very challenging if the conductivity model contains some anatomical detail [93].

7.3.2 EEG Inverse Problem Theory

The functional that needs to be minimized in order to obtain a map of current densities is:

$$F = ||\Phi - KJ - c_1||^2 + \alpha||J||^2, \tag{7.3}$$

for a given K, Φ and α, α is a weighting coefficient used to modulate the reconstruction.

The explicit solution to the minimization problem is:

$$\hat{J} = T \cdot \Phi \quad \text{with} \quad T = K^T \cdot H\big[HKK^T H + \alpha H\big]^+ \tag{7.4}$$

and $H = I - \frac{(1 \cdot 1^T)}{(1^T \cdot 1)}$, $H \in \mathbb{R}^{N_e \times N_e}$.

H is the centering matrix, $I \in \mathbb{R}^{N_e \times N_e}$ is the identity matrix and $1 \in \mathbb{R}^{N_e \times 1}$ is a vector of ones. Here the superscript T denotes the transpose operator as before while the superscript $^+$ represents the Moore-Penrose pseudoinverse. Up to this point, the current densities are reconstructed according to LORETA algorithm [112]. In eLORETA [113], the matrix T of the inversion step is modified as:

$$\hat{J} = T \cdot \Phi \quad \text{with} \quad T = W^{-1}K^T\big[KW^{-1}K^T + \alpha H\big]^+ \tag{7.5}$$

where $W \in \mathbb{R}^{3N_v \times 3N_v}$ is a weighting matrix that can be computed with an iterative method. Starting from the LORETA current density power estimate of Eq. 7.4, it is then possible to obtain sLORETA reconstructions by standardizing the density power relative to each voxel according to [114]:

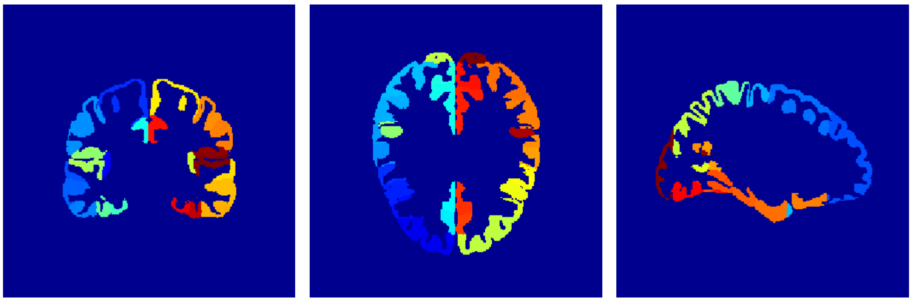
$$\hat{J}_l = J_l^T \left([S_{\hat{J}}]_{ll}\right)^{-1} \hat{J}_l \tag{7.6}$$

where $\hat{J}_l \in \mathbb{R}^{3\times 1}$ is the current density estimate at the l^{th} voxel given by Eq. 7.4 and $[S_{\hat{J}}]_{ll}$ is the l^{th} diagonal block of matrix:

$$S_{\hat{J}} = K^T \left[KK^T + \alpha H\right]^+ K. \tag{7.7}$$

Since the brain activity that is possible to detect by EEG acquisitions lays only in the cerebral cortex of the brain, the inversion process is limited to that region. We map the position of the electrodes mounted on an EEG cap according to the international 10-20 system onto the segmented head volume [115].

The current density estimated by the described methods is generally obtained for each voxel of grey matter given by the segmented MRI. Additionally, grey matter voxels can be parcellated in 96 different brain regions according to the Harvard-Oxford cortical and subcortical structural atlas (Fig. 7.4) [116].



(a) (b) (c)

Figure 7.4: Coronal (a), Transverse (b) and Sagittal (c) section of the parcellated domain of interest.

7.4 Conclusions

This chapter has given principles, theoretical basis and applications of EEG, with particular focus on BCI systems and EEG Source Localization which represents the object of the research activity illustrated in the following chapters.

Chapter 8

Simulink Libraries for Creamino BCI

EEG Source Localization is one of the most relevant topic treated in the ARCES research center in last years. A lot of work was developed and described in [117] and [94], where the implementation of many EEG inverse solvers (including LORETA's family of algorithms) is proposed, with particular focus on the method of choice of the reconstruction algorithms and detailed comparisons with respect alternative techniques.

Recently, in the scope of the CREAM project, a C/CUDA parallel implementation of LORETA's family of algorithms was developed by the ARCES research team, starting from the previous work on EEG source localization. Having at hand the parallel implementation of LORETA, eLORETA and sLORETA, my objective was to develop SIMULINK libraries which were able to compute the mean power of the EEG sources lying in user-defined regions of interest and that can be easily introduced in a real-time system such as *Creamino*.

The objective of this section is to integrate these inverse problem solvers in a user-friendly SIMULINK model, which can be easily introduced in a real-time BCI system.

The C/CUDA implementation of all three reconstruction methods is integrated with the Simulink environment by adopting a user-defined block that loads a dynamic shared library generated starting from the native code of these functions.

To facilitate the interaction with other software components, we adopted a

standard laptop computer based on Windows 10 Enterprise and equipped with a CUDA capable device.
The dynamic shared library was developed and build using Microsoft Visual Studio Professional 2013 (v12.0.40629.00 update 5) and tested within the MATLAB environment. After that, a user-defined block (S-Function) was implemented to invoke the function in the dynamic library.

8.1 MATLAB Testing of the dynamic library

The dynamic library built using VS2013 (*dll* extension for Windows OS), was loaded and tested within MATLAB by means of specific functions. First of all it was loaded in MATLAB by using the *loadlibrary* function. In order to verify that the functions of the library were loaded correctly, we used the function *libfunctionsview* that allows to display the shared library function signatures.
Finally, the function *calllib* was employed to invoke the functions in the generated dynamic library. Since the library is composed of C/C++ or CUDA functions, a specific structure has been built to pass arguments from the MATLAB environment to the dynamic shared library. This structure is realized through the *libpointer* function, which generates a pointer object that can be used within the shared library. The result of the invoked function returns in a *libpointer* object which is acquired through the *getfunction*. The last step of the MATLAB validation of the *dll* has been realized comparing the result returned by the *dll* function execution with the result given by the MATLAB implementation of the same function.

8.2 Calling the Shared Library from Simulink

Once the library has been tested in the MATLAB environment, it is possible to invoke the dynamic library functions in the Simulink environment. This integration can be realized by calling the shared library from a user-defined block. To this purpose there are several possibilities, such as:

1. Calling shared library using MATLAB S-function;

2. Calling shared library using CMEX S-function.

The first method allows to invoke the shared library functions through local functions using function handles while the second method allows the Simulink engine to directly invoke MEX S-function. As a result, a CMEX S-function simulates faster than a MATLAB S-function because the latter calls the MATLAB interpreter for every callback method.
Since LORETA algorithms are characterized by high-computational requirements which markedly increases in a real-time setting, the CMEX S-function appears to be more appealing for our purposes.
This approach allows the user to write plain C code; moreover, it is needed to load the generated library and call functions in it. Subsequently, the code is compiled into a binary file that is recognized by SIMULINK and linked against the shared library. This binary file is called S-function MEX file and can be invoked from a user-defined SIMULINK S-Function block.

8.3 Creation of the C MEX S-function

In order to manually create the C MEX S-function we use *sfuntmpl_basic.c*, one of the provided S-function templates which contain the skeleton of all the required and optional callback methods that a C MEX S-function can implement. It is important to underline that the S-function blocks implementing the parallel LORETA algorithms are defined as Level-2 S-functions, since, compared to the Level-1 functions, they provide several additional features and capabilities of SIMULINK built-in blocks, including:

- Multiple input and output ports;
- The ability to accept vector or matrix signals;
- Support for various signal attributes including data type, complexity, and signal frames;
- Ability to operate at multiple sample rates.

To provide a complete library containing the CUDA parallel implementation of the LORETA algorithms, a total of four C MEX S-function blocks are made available:

1. LORETA/eLORETA Block;

2. LORETA/eLORETA Block with Parcellated Output;
3. sLORETA Block;
4. sLORETA Block with Parcellated Output.

The LORETA/eLORETA blocks are capable to implement both the LORETA and the eLORETA algorithms, depending on the input matrix that user supplies to the block.

8.4 LORETA/eLORETA block

Let's consider N_e extracranial measurements gathered in N_s samples coming from N_e measuring electrodes placed on the scalp. In this case we have that:

- $\Phi \in \mathbb{R}^{N_e \times N_s}$ is the matrix that contains the electric potentials measured on the scalp with respect to a common reference electrode in N_s samples;
- $T \in \mathbb{R}^{N_e \times 3N_v}$ is the matrix of the LORETA inversion step described in Eq. 7.4, and N_v is the number of voxels that discretize the cerebral cortex.

The user-defined S-Function which implements the LORETA block is represented in Fig. 8.1.

The inputs of this block are the static matrix $T \in \mathbb{R}^{3 \cdot N_v \times N_e}$ (which is trans-

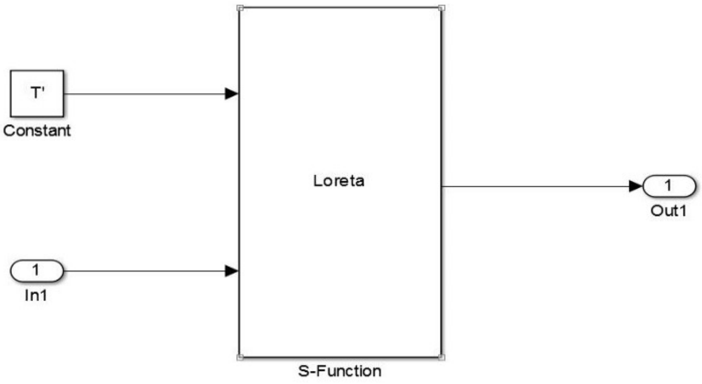
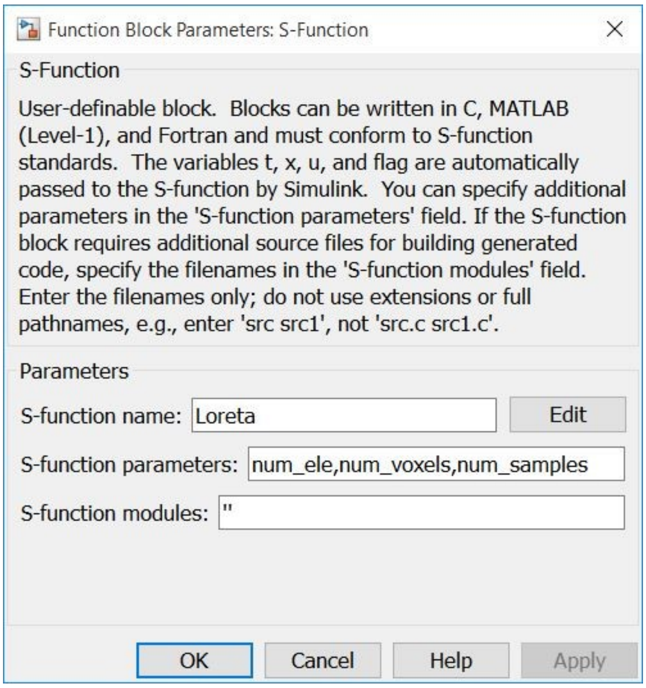


Figure 8.1: LORETA/eLORETA SIMULINK block.

posed to maintain coherence between Matlab and C languages) and the real-time gathered measurements $\Phi \in \mathbb{R}^{N_e \times N_s}$ (In1 in Fig. 8.1). N_s has to be set equal to one in order to manage instantaneous measurements.

The parameters of this block are shown in the Function Block Parameters Window represented in Fig. 8.2 where *num_ele* is the number of electrodes (N_e), *num_voxels* is the number of voxels that discretize the cerebral cortex multiplied by 3 ($3 \cdot N_v$) and *num_samples* is the number of consecutive measurements collected using the N_e electrodes (N_s).

The output is a vector composed of N_v components, which is the 2-norm of the three Cartesian components of the vector $\hat{J} \in \mathbb{R}^{3 \cdot N_v}$ shown in Eq. 7.4. For sake of simplicity, the static matrix T and the parameters are loaded directly from the MATLAB environment after the execution of a pre-processing step implemented in a Matlab m-file. In order to implement the eLORETA inversion step, it is sufficient to compute the static matrix T following Eq. 7.5.



User-definable block. Blocks can be written in C, MATLAB (Level-1), and Fortran and must conform to S-function standards. The variables t, x, u, and flag are automatically passed to the S-function by Simulink. You can specify additional parameters in the 'S-function parameters' field. If the S-function block requires additional source files for building generated code, specify the filenames in the 'S-function modules' field. Enter the filenames only; do not use extensions or full pathnames, e.g., enter 'src src1', not 'src.c src1.c'.

S-function name: Loreta

S-function parameters: num_ele,num_voxels,num_samples

S-function modules: ''

Figure 8.2: Parameters of LORETA SIMULINK block.

8.4.1 LORETA/eLORETA with Parcellation block

In this case, grey matter voxels can be parcellated in N_p different brain regions according to an atlas. This step involves the introduction of two additional static inputs for the parcellated LORETA/eLORETA SIMULINK block: *grayParcLinOut* and *select*.

The first one is the mapping of the segmented grey matter into the parcellated domain while the second one is the vector which contains the label of the ROI of interest. For example, if grey matter voxels are parcellated in 96 different brain regions according to the Harvard-Oxford cortical and subcortical structural atlas, *grayParcLinOut* identifies the label of the ROI of each voxel and select is the vector containing the labels of the ROIs the user wants to monitor.

The number of parameters is also increased, introducing two additional parameters: *lengthParc* and *lengthSelect*. The first one contains the total number of voxels mapped in the ROI labels while the second one contains the labels of the ROIs the user wants to monitor. The output of this block is the sum of the reconstructed current densities in the voxels composing each selected ROI. The block is shown in Fig. 8.3.

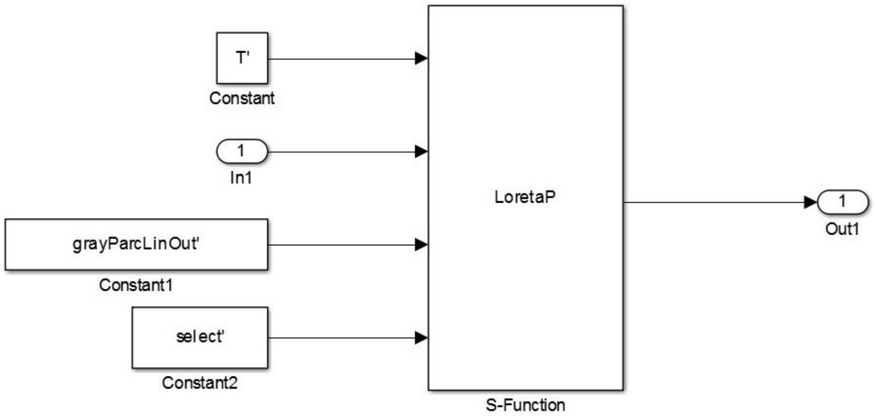


Figure 8.3: LORETA/eLORETA with Parcellation SIMULINK block.

8.5 sLORETA Block

This user-defined S-Function block executes the parallel implementation of the sLORETA inversion step, which extends the LORETA algorithm introducing the standardization step defined in Eq. 7.6.
Fig. 8.4 shows the sLORETA block where the inputs are represented by the static matrix $T \in \mathbb{R}^{3N_v \times N_e}$ (the same of the LORETA block shown previously) and by the vector $Rvetto \in \mathbb{R}^{9 \cdot N_v}$ which is the linearized vector that derives from the matrix $[S_{\hat{j}}]^{-1}$ of Eq. 7.6.
The parameters are: *num_ele* (the number of electrodes), *num_voxels* (number of voxels that discretize the cerebral cortex multiplied by 3), *num_samples* (number of consecutive measurements observed in the N_e electrodes) and *lengthR* (length of the input *Rvetto*). The output of the sLORETA block are the estimated standardized current densities for each voxel of the cerebral cortex as defined in Eq. 7.7.

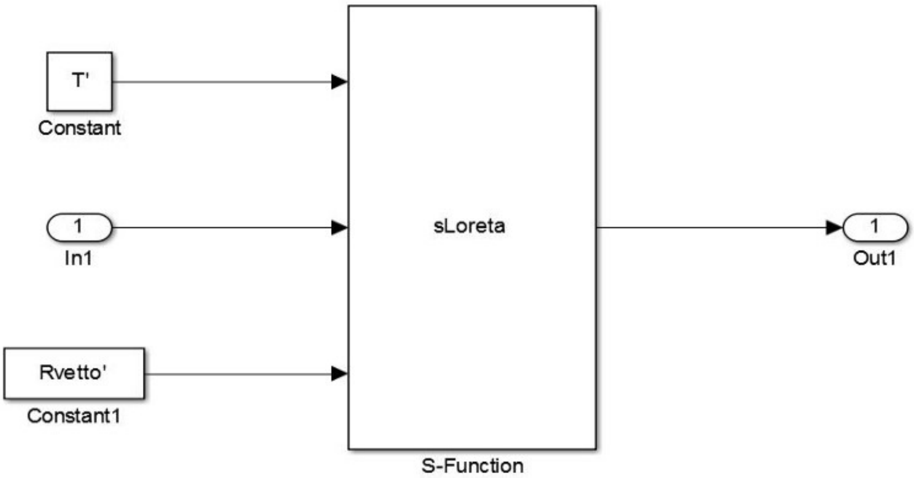


Figure 8.4: sLORETA SIMULINK block.

8.5.1 sLORETA with Parcellation block

The block which integrates the sLORETA inversion step and the parcellation is illustrated in Fig. 8.5. The additional parameters and inputs as compared to the sLORETA block are the same of the LORETA/eLORETA with Parcellation block. In fact:

- *grayParcLinOut* identifies the label of the ROI of each voxel and *select* is the vector containing the labels of the ROIs the user wants to monitor;
- the number of parameters is increased introducing *lengthParc* and *lengthSelect*.

The first one contains the total number of voxels mapped in the ROI labels while the second one contains the labels of the ROIs the user wants to monitor. The output of this block is the sum of the reconstructed standardized current densities in the voxels composing each selected ROI.

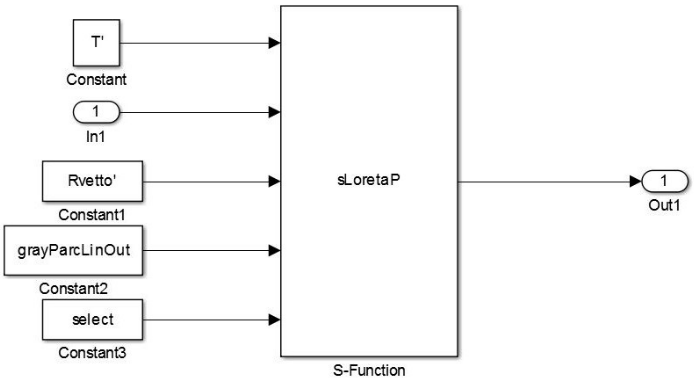


Figure 8.5: sLORETA with Parcellation SIMULINK block.

8.6 Performance

The computational performance of the implemented LORETA, eLORETA and sLORETA modules are evaluated on a standard laptop computer equipped with a recent, CUDA capable Nvidia Graphics Card. Details are listed in Table 8.1.

CPU	Intel Core i7 4710HQ Quad-Core @ 2.50GHz
Graphic Card	NVIDIA GeForce GTX 860M (4096 MB GDDR5)
RAM	16 GB DDR3

Table 8.1: Laptop LENOVO Y50 Equipment.

The real-time system used for the performance evaluation is obtained interfacing the acquisition module with the LORETA blocks, leading to the SIMULINK model shown in Fig. 8.6.
The “*Creamino*” block in Fig.8.6 controls the communication between *Creamino* BCI and the standard PC equipped with SIMULINK, allowing to manage the communication with the BCI hardware and to import the EEG signal.
The ”Buffer” block between the “Reference” and the “Data Type Conversion” blocks in Fig. 8.6, is necessary to gather the input samples to be passed to the LORETA block. If the buffer size is set equal to one, the LORETA block receive N_e instantaneous measurements collected in a vector at the user-defined sampling frequency.

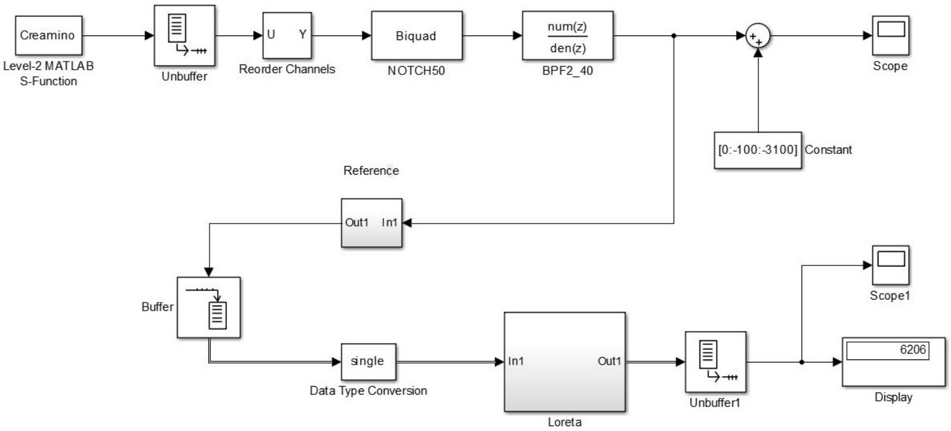


Figure 8.6: Real-time system for performance evaluation

The ”Reference” block shown in Fig. 8.6 is implemented as illustrated in Fig. 8.7 and it allows to define a channel as reference and to remove the average from the signals through the multiplication by the symmetric matrix H previously defined in Sec. 7.3.2.
The characterization of the modules has been realized in function of three fundamental parameters which modify the requirements for real-time execution of the whole system:

1. Dimension of the uspstream LORETA/eLORETA/sLORETA Buffer;
2. Sample Rate;
3. Dimension of the Matrix T.

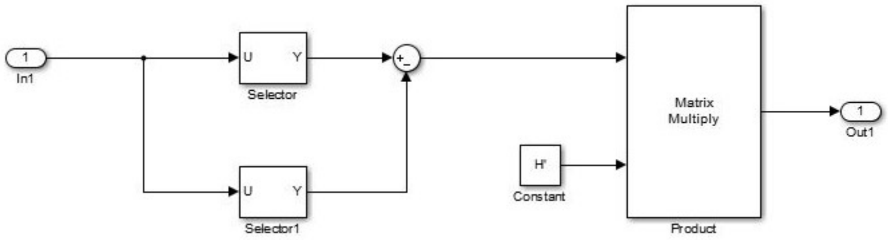


Figure 8.7: Reference block implementation.

We consider the real-time constraint satisfied when the SIMULINK running time is lower than the time obtained by dividing the number of samples with the sample rate. The bigger is the uspstream buffer dimension the lower is the temporal resolution (since we have to wait to collect the necessary samples to fill the buffer) but the real time constraint becomes easier to achieve.

Since the static matrix T coming from the full-resolution cerebral cortex segmentation is not supported by the adopted laptop computer due to the memory constraints, in the following we adopt three different configurations of the static matrix T, coming from its 2-3-5 undersampling.

The dimensions of the obtained static matrix are listed in Table 8.2.

Resolution	Dimension of Matrix T
Full Resolution	2098857×31
Undersampled 2 (U2)	261933×31
Undersampled 3 (U3)	77205×31
Undersampled 5 (U5)	16599×31

Table 8.2: Dimension of full T matrix and its 2, 3 and 5 times undersampled versions.

The first dimension of the matrix T is the number of considered cerebral cortex voxels and the second one is the number of acquisition channels. The number of measuring channel is 32 but the number of considered channel is 31 since channel 2 is used as reference.

8.6.1 Performance of LORETA/eLORETA as function of the Buffer Dimension

This test is realized adopting three different static matrices T coming from the undersampling of the full-resolution cerebral cortex and, for each matrix T, data are collected by varying the dimension of the upstream LORETA-eLORETA buffer.

In this real time system, "*Creamino*" provides $8 \times N_e$ samples at a frequency of 8/250 Hz and the following "UnBuffer" block is used to provide N_e samples at a rate of 250 Hz, leading to a sample rate of 250 Hz.

The total simulation time is 10.24 s for a total of 2560 samples for each of the 32 channels. The LORETA upstream buffer dimensions we tested are 8, 16, 32, 64, 128 and the listed results are obtained both for the LORETA/eLORETA block and for the same block with parcellation, with a number of ROIs variable from 5 to 15. The results obtained with the U2 T matrix are shown in Fig. 8.8.

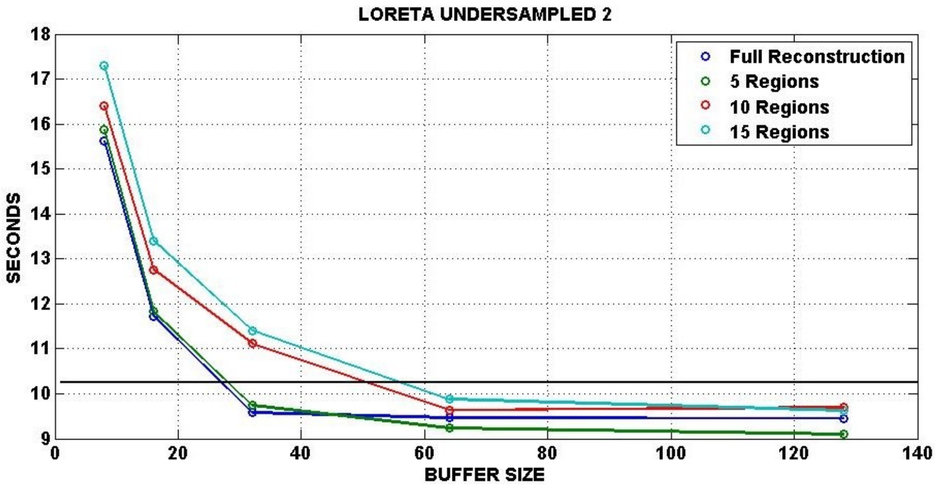


Figure 8.8: LORETA/eLORETA performance with U2 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

Fig. 8.8 indicates that real-time constraints are satisfied for a minimum buffer dimension of 64 samples.

The results obtained with the U3 and the U5 T matrices are shown in Fig. 8.9 and Fig. 8.10, highlighting a real-time compatibility for any user-defined

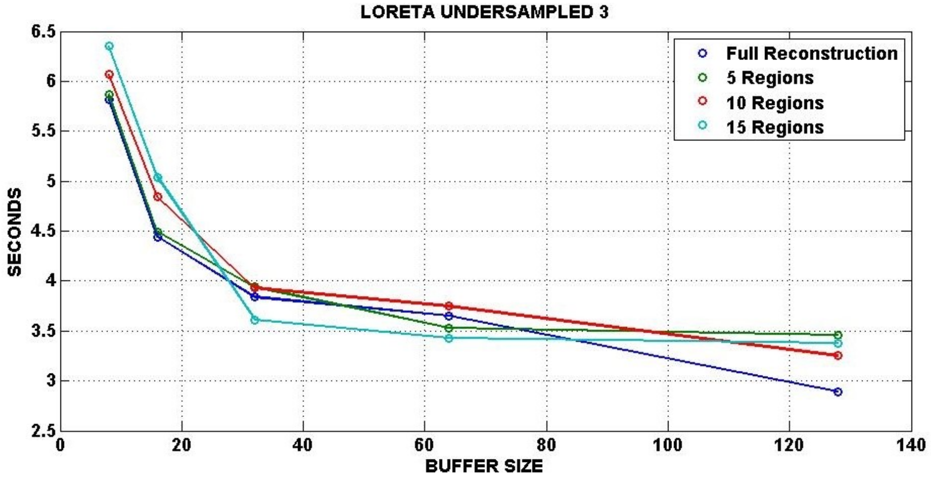


Figure 8.9: LORETA/eLORETA performance with U3 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

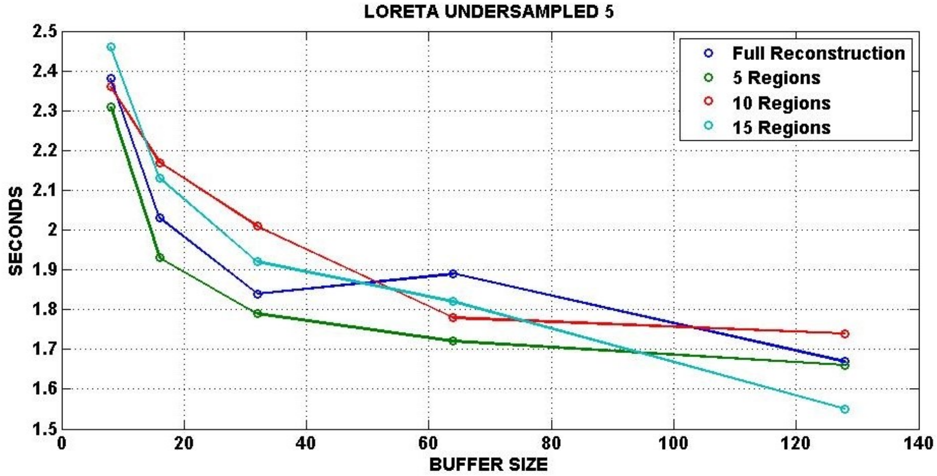


Figure 8.10: LORETA/eLORETA performance with U5 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

buffer dimension.

To conclude, the introduction of the upstream buffer in the LORETA-eLORETA blocks allows to manage a high sample rate (250 samples per second), with a negligible delay (up to 512 ms) in the output visualization. A minimum delay of 32 ms is supported for the matrix dimension corresponding to the U3 and U5 cases, but not always for the U2 case. In this scenario the LORETA block requires at least 256 ms of delay.

8.6.2 Performance of sLORETA as function of the Buffer Dimension

The setup of this test is the same of the previous performance test, except that we substituted the LORETA/eLORETA block with the sLORETA one. Even in this case results are obtained with different buffer dimensions (8,16,32,64,128) for both the sLORETA block and the same block with parcellation, with a number of ROIs variable from 5 to 15. The results obtained in the case of the U2 T matrix are collected in Fig. 8.11.

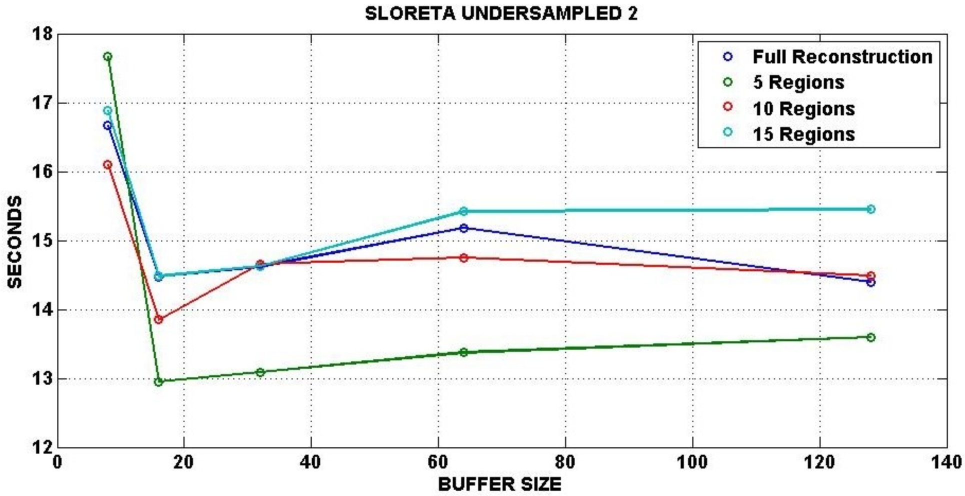


Figure 8.11: sLORETA performance with U2 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

It turns out that the proposed block does not satisfy real-time constraints for this experiment, since the running time is always greater than the simulation time. This overhead is due to the large amount of data that needs to be transferred from MATLAB to SIMULINK and from SIMULINK to the C MEX S-function. The results obtained with the U3 and the U5 T matrices are shown respectively in Fig. 8.12 and Fig. 8.13, in this case a real-time compatibility for any user-defined buffer dimension is apparent.

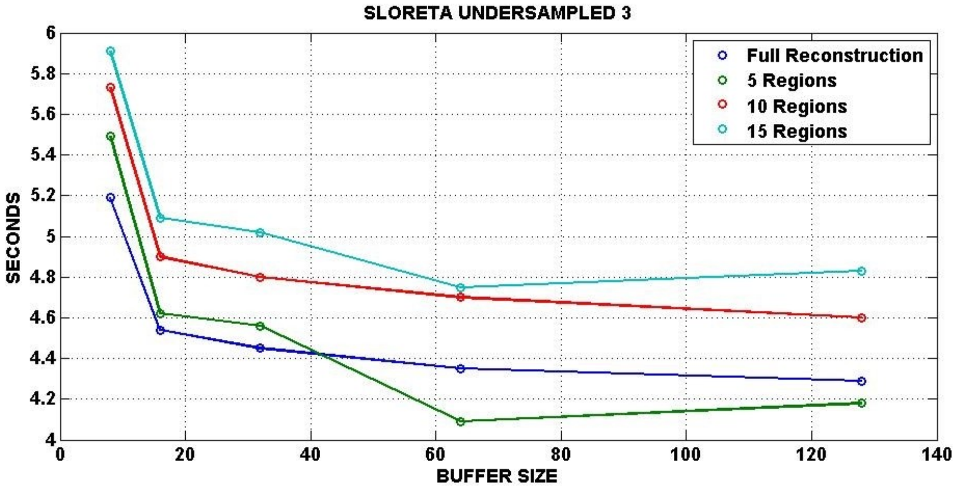


Figure 8.12: sLORETA performance with U3 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

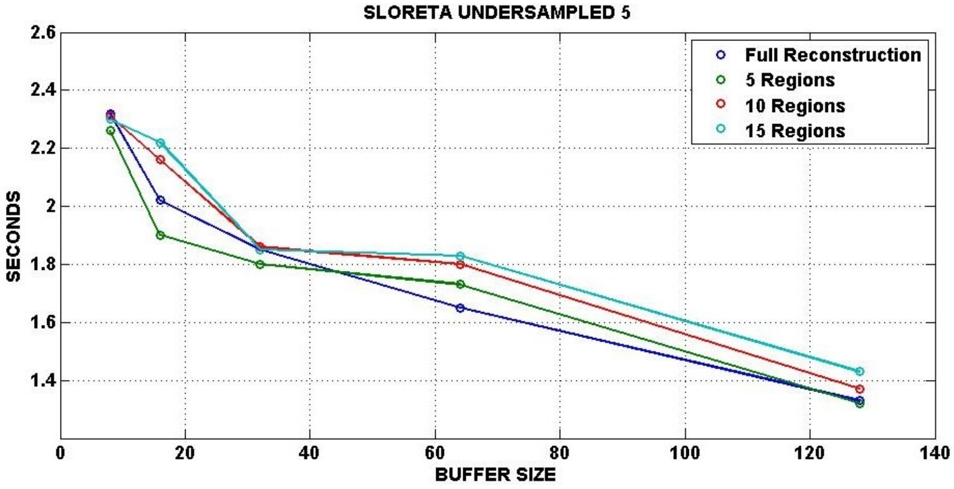


Figure 8.13: sLORETA performance with U5 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

8.6.3 Performance of LORETA/eLORETA as function of the Sample Rate

This test is realized adopting three different static matrices T coming from the undersampling (U2, U3, U5) of the full resolution cerebral cortex. For each matrix T, data are collected by tuning the sample rate of the real-time simulation, removing the upstream buffer of the LORETA/eLORETA block. The sample rate varies from 15 to 40 samples/s in the undersampled-2 experiment, from 60 to 120 samples/s in the undersampled-3 test and from 150 to 450 samples/s in the undersampled-5 case.

The simulation time is exactly 10 s, so the running time of the model must be less or equal to this benchmark in order to guarantee real-time compatibility. Results are obtained both for the LORETA/eLORETA block and for the same block with parcellation, with a number of ROIs variable from 5 to 15. Results of the U2 experiment are collected in Fig. 8.14. The graph points to a maximum sample rate of 25 samples/s for the real-time behaviour. The results of the U3 and U5 experiments are shown respectively in Fig. 8.15 and in Fig. 8.16, leading to a maximum sample rate of 80 samples/s for the U3 case and of 250 samples/s for the U5 case.

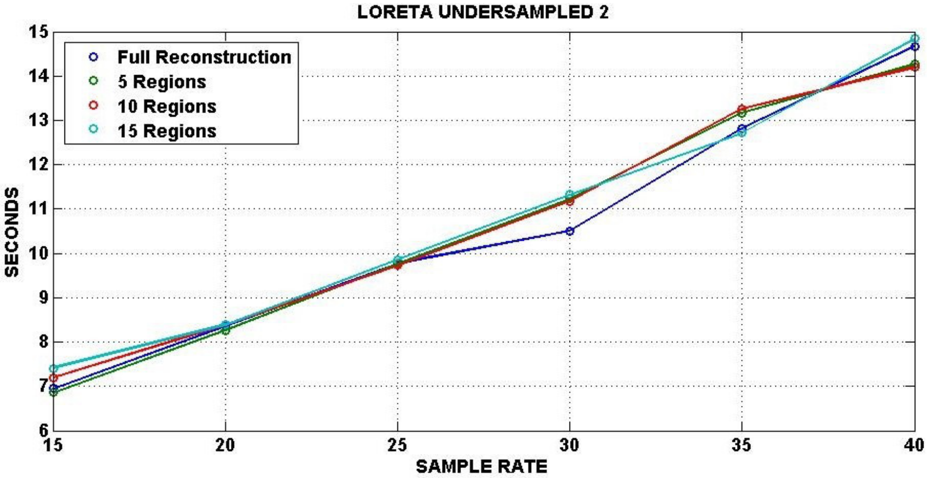


Figure 8.14: LORETA/eLORETA performance as a function of the sample rate with U2 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

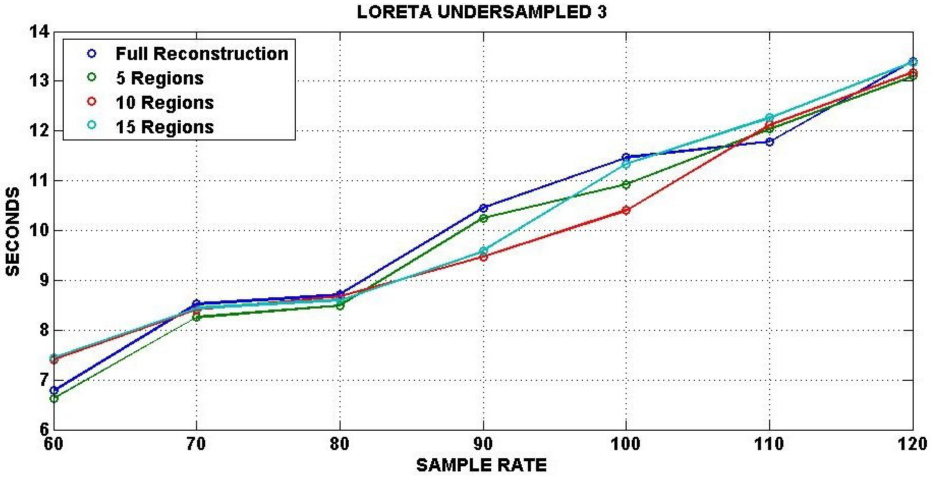


Figure 8.15: LORETA/eLORETA performance as a function of the sample rate with U3 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

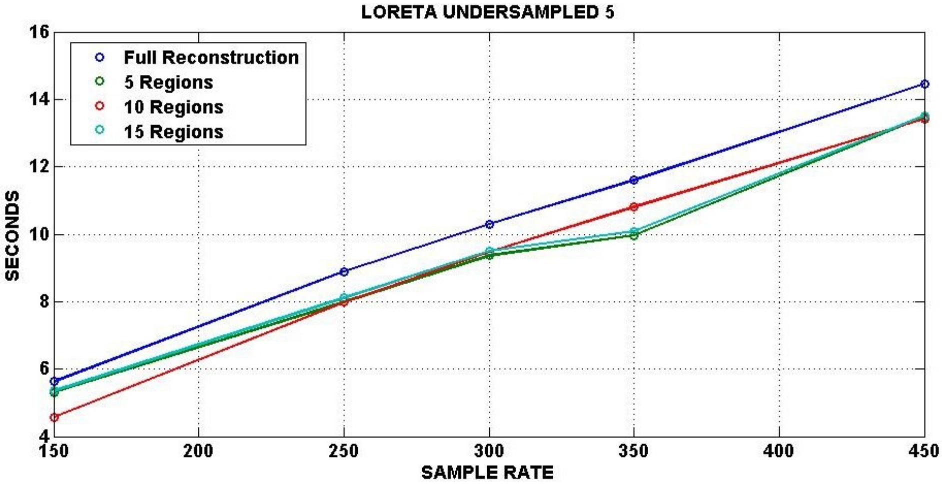


Figure 8.16: LORETA/eLORETA performance as a function of the sample rate with U5 T matrix. Results are evaluated both for the standard block and for the parcellated block with 5-10-15 ROIs selected.

8.6.4 Performance of LORETA/eLORETA as function of the dimension of the matrix T

The last point concerning the characterization of the provided LORETA-eLORETA block involves the performance evaluation with respect to the number of voxels composing the discretization of the gray matter region. To this purpose we adopted the U2, U3, U5 static matrix already introduced in the previous sections, corresponding to decreasing number of voxels.

Results are collected by using the system configuration illustrated in Fig. 8.6, adopting 8, 16, 32, 64, 128 buffer dimensions.

In this real time system, "*Creamino*" provides $8 \times N_e$ samples at a frequency of 8/250 Hz and the following "UnBuffer" block is used to provide N_e samples at a rate of 250 Hz, leading to a sample rate of 250 Hz.

The total simulation time is 10.24 s for a total of 2560 samples for each of the 32 channels.

Results obtained using the LORETA/eLORETA block with the full-resolution T matrix are collected in the graph illustrated in Fig. 8.17, while U2, U3 and U5 downsampling results are shown in Fig. 8.18, Fig. 8.19 and Fig. 8.20 respectively.

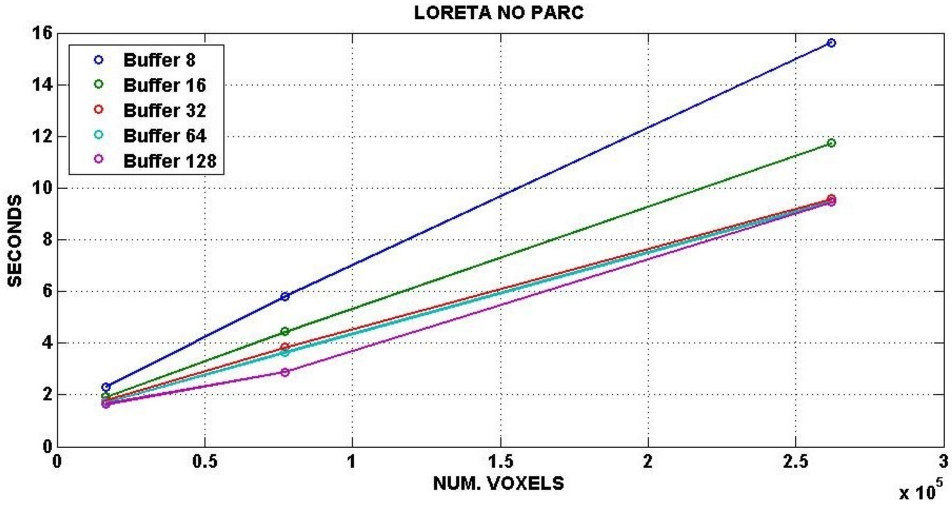


Figure 8.17: LORETA/eLORETA block performance in function of the number of voxels. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

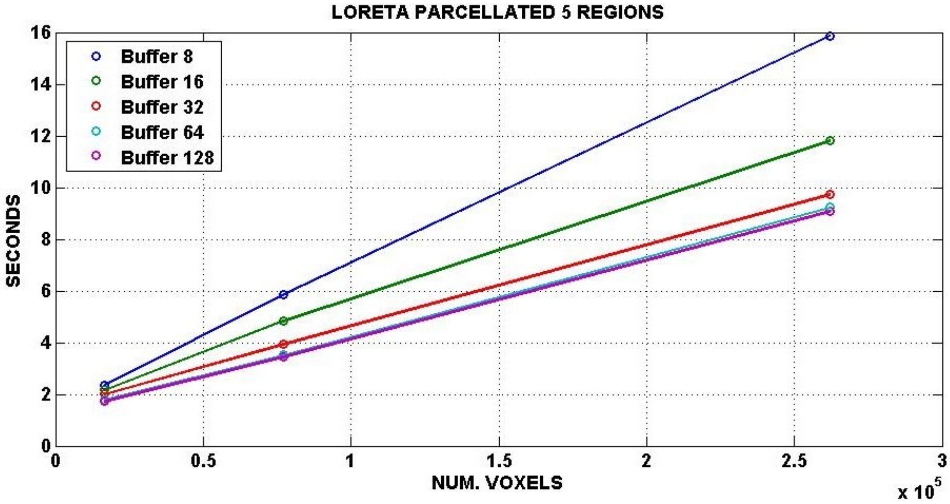


Figure 8.18: LORETA/eLORETA block performance as function of the number of voxels, with 5 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

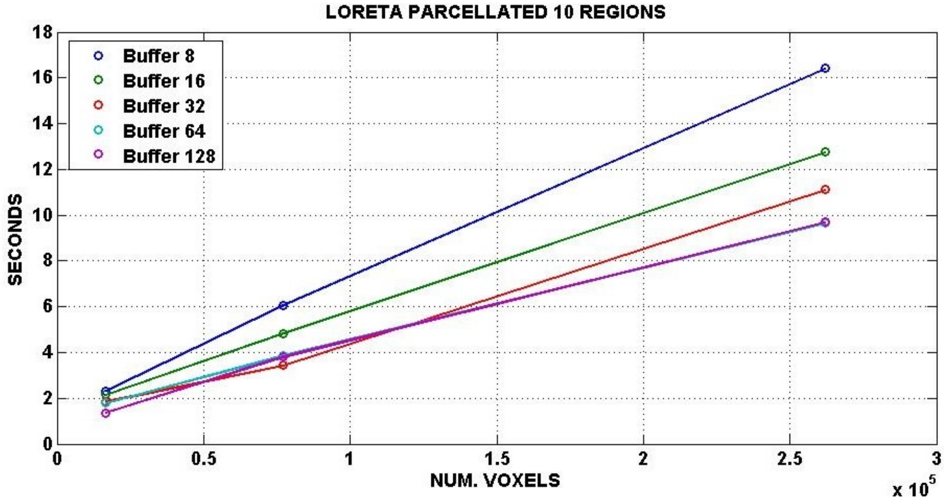


Figure 8.19: LORETA/eLORETA block performance as function of the number of voxels, with 10 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

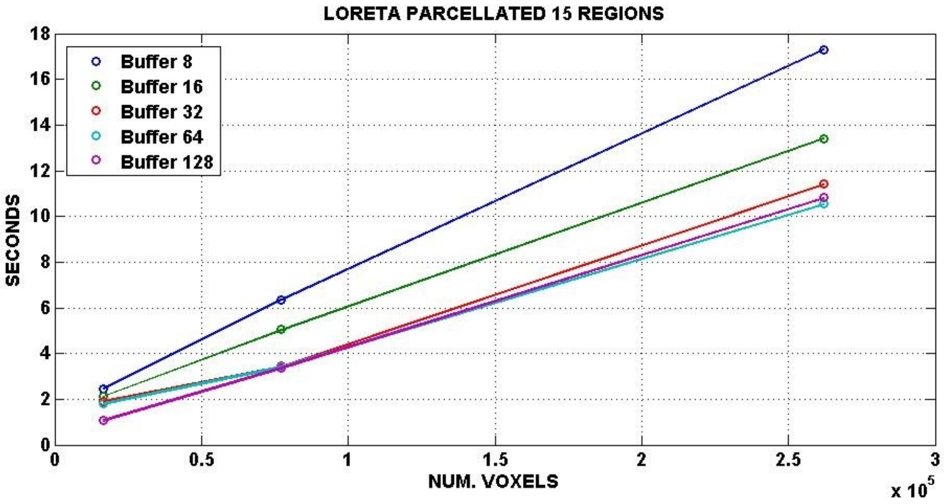


Figure 8.20: LORETA/eLORETA block performance as function of the number of voxels, with 15 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

8.6.5 Performance of sLORETA as function of the dimension of the matrix T

In this case results are obtained using the same procedure shown in the previous section except that the LORETA/eLORETA block is substituted with the sLORETA one. Figures 8.21, 8.22, 8.23, 8.24 show the same behaviour of the LORETA/eLORETA block, leading to the conclusion that in order to obtain real-time condition with 32 channels, it is necessary to adopt less than 150000 voxels of gray matter. This estimation is valid for every dimension of the upstream sLORETA buffer.

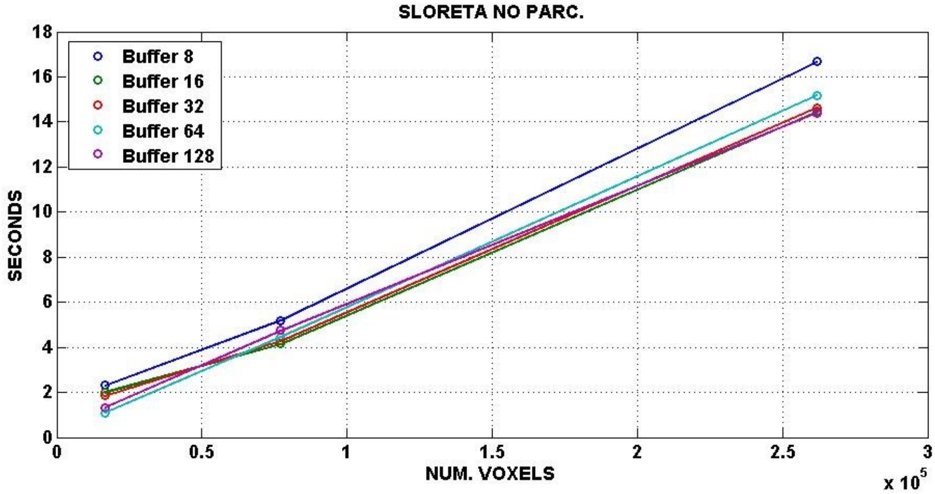


Figure 8.21: sLORETA block performance in function of the number of voxels. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

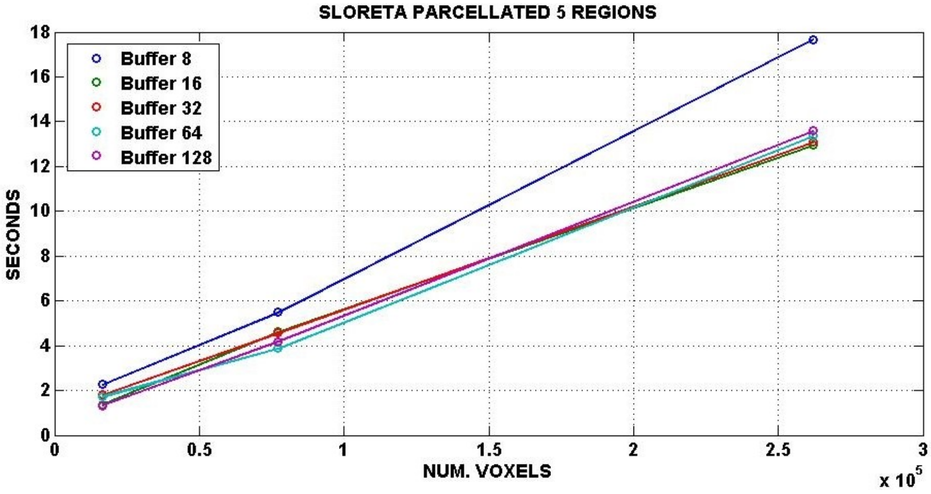


Figure 8.22: sLORETA block performance as function of the number of voxels, with 5 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

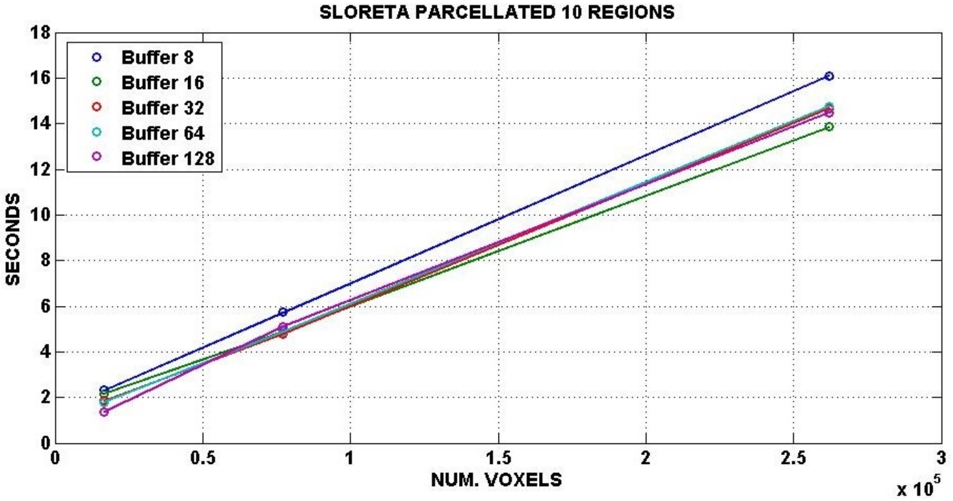


Figure 8.23: sLORETA block performance as function of the number of voxels, with 10 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

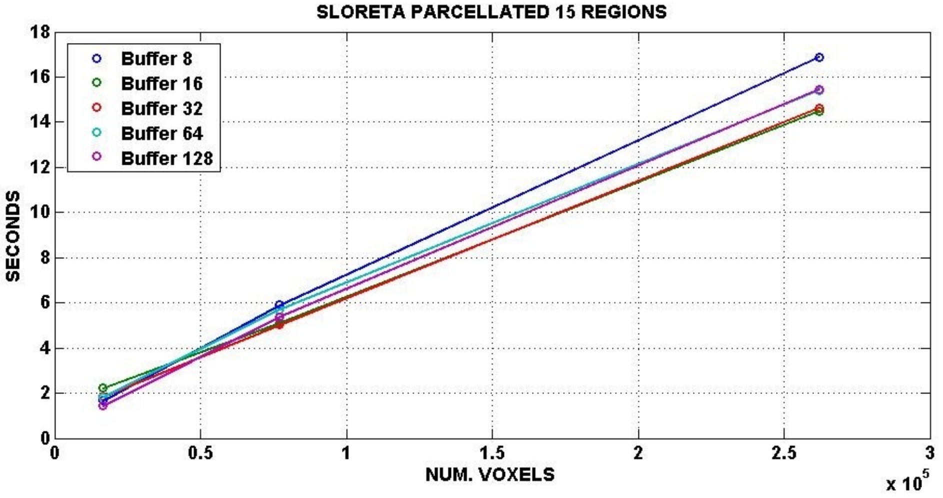


Figure 8.24: sLORETA block performance as function of the number of voxels, with 15 ROIs monitoring. Results are evaluated adopting 8, 16, 32, 64, 128 buffer dimensions.

8.7 Experimental Test

To test the whole system in a real-time configuration, we decided to monitor the alpha rhythm of a subject using the SIMULINK model depicted in Fig. 8.25. Through this model we tested all the LORETA's family blocks. The instantaneous power of the selected ROIs is visualized in a time-scope and reconstructions are performed using the LORETA/eLORETA and sLORETA blocks previously described.

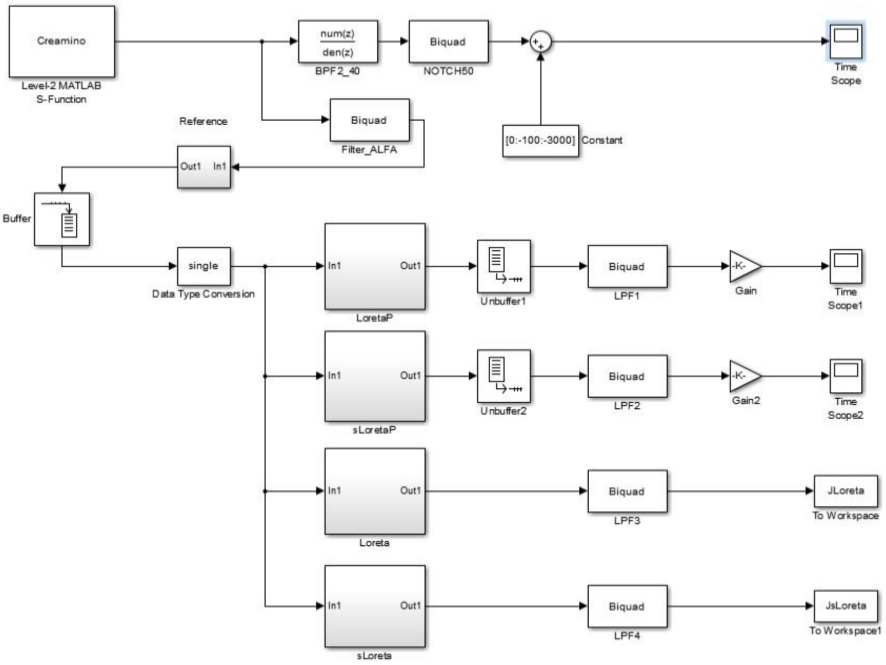


Figure 8.25: SIMULINK system for real-time alpha rhythm monitoring.

A low pass filter is placed in the paths between LORETA blocks and Time Scopes to remove the high-frequency components of the outputs. Finally, the power of the ROI given by the parcellated blocks is normalized in function of the number of voxels composing each region through a suitable gain. The dimension of the LORETA blocks upstream buffer is equal to 64 samples, and the sample rate is 250 Hz. The test has been realized adopting a global T static matrix coming from the 2-undersampling of the full-resolution cerebral cortex discretization. The total duration of the experiment is 2 minutes and every 30s the subject opens/closes the eyes.

Since the aim of the experiment is to monitor the power of the alpha signal, we adopted the parcellated version of LORETA/eLORETA and sLORETA to observe the power of the alpha signal in the occipital and the frontal ROIs: in the occipital ROI we expected to have an increase of the power when the subject close the eyes while in the frontal ROI we expected to observe no variations.

Since we are interested on the alpha frequency of the EEG signals, the Biquad Filter block named ”Filter_ALFA” in the real-time system in Fig. 8.25 allows to extract only the alpha component of the input signal.

The real-time results observed during the simulation are shown in Fig. 8.26 and Fig. 8.27.

In the reported graphs, the blue line is the trend of the normalized power of

Figure 8.26: LORETA parcellated outputs in the ROIs of interest to monitor the alpha rhythm of a subject. The yellow line is relative to the EEG signal measured in the frontal region, where the alpha signal cannot be appreciated. The blue signal was obtained measuring the response in the occipital region, where the power of the alpha signal is increased during the eyes-closed phase.

Figure 8.27: sLORETA parcellated outputs in the ROIs of interest to monitor the alpha rhythm of a subject.The yellow line is relative to the EEG signal measured in the frontal region, where the alpha signal cannot be appreciated. The blue signals was obtained measuring the response in the occipital region, where the power of the alpha signal is increased during the eyes-closed phase.

the occipital ROI while the yellow one represents the normalized power of the frontal region. As expected, during the eyes-closed period the amount of power of the occipital ROI markedly increases as compared to the normalized power obtained in the frontal region.

8.8 Conclusions

This chapter illustrates the implementation of SIMULINK Libraries for *Creamino* BCI system, starting from C/CUDA parallel implementations of LORETA, eLORETA and sLORETA in the time domain which were developed previously by the ARCES research team in the scope of the CREAM project and theoretically introduced in Chapter 7.

The implemented library are able to compute the mean power of the EEG sources lying in user-defined regions of interest and were validated through an experimental test of alpha monitoring of a subject.

We proof that performance of the proposed libraries are fast enough for BCI typical applications under different constraints, depending on the number of voxels composing the ROIs, the buffer dimension and the sample rate of the BCI system.

For example, we proof that performance of the proposed libraries are fast enough for real-time behaviour of *Creamino* BCI system applications with a maximum number of $150 \cdot 10^3$ voxels in a 32 channels system interfaced with a standard laptop computer both for LORETA/eLORETA and sLORETA reconstruction methods.

In the case of a 64 channels system the maximum number of admitted voxels decreases to $75 \cdot 10^3$.

Since certain experiments might need to analyse EEG sources localization in the frequency domain rather than in time-domain, in the following chapter we propose the implementation of frequency EEG Source Localization algorithms.

Chapter 9

EEG Frequency Source Localization

Certain experiments might need to analyze data in the frequency domain rather than in time-domain, for example when there is interest in analysing what happens at certain frequency-bands (e.g. alpha, beta) or frequencies (e.g in steady-state evoked potentials).
The object of this chapter is to illustrate the parallel implementation of frequency EEG Source Localization algorithms and its validation through multi-trial Event Related Potentials (ERP) experiments obtained from visual stimuli.
The proposed method is based on a commonly used Welch-periodogram of eLORETA current density estimation, where critical kernels are massively parallel optimized for the available HPC platform.

9.1 Implementation

One possible approach to analyze data in the frequency domain might be to filter the data among the different frequency bands of interests and then compute reconstructions in the time-domain. However, this requires to perform a reconstruction per each frequency (or band) of interest. On the other side, it is not possible to compute the frequency spectrum of the voltages on the electrodes and then perform the source reconstructions.
Therefore one needs to first reconstruct the sources and then extract the frequency spectrum of the time-series associated to one source or a particular

region.

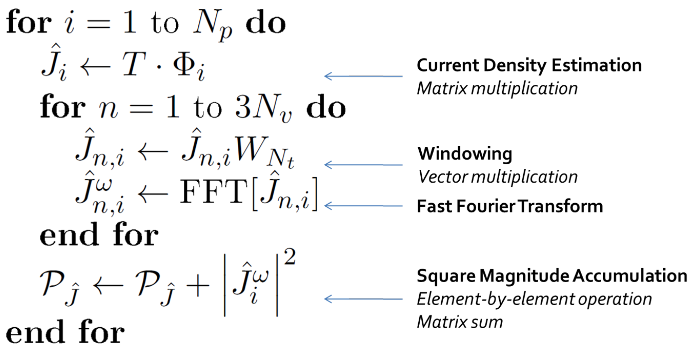
To this purpose we adopt the periodogram, which is a method to estimate the spectral density of a signal by averaging, over a number of windows, the square magnitude of the signal Discrete Fourier Transform (DFT).

We define the periodogram matrix of the current density estimate averaged over N_p epochs the following matrix:

$$\wp_{\hat{J}} = \begin{bmatrix} |\hat{J}^{\omega}_{1,i}|^2 \\ \vdots \\ |\hat{J}^{\omega}_{3\cdot N_v,i}|^2 \end{bmatrix} = \frac{1}{N_p}\sum_{i=1}^{N_p} |\hat{J}^{\omega}_i|^2 \tag{9.1}$$

whose rows are the periodogram at each voxel and where $\hat{J}^{\omega}$ is the Discrete Fourier Transform (DFT) of the current density estimate $\hat{J}$. To obtain the modified periodogram, it is sufficient to introduce a windowing and an overlap before the DFT, leading to a smoothed estimate of the spectral density.

Since the operations for the modified periodogram evaluation are highly parallelizable, we implement a CUDA routine running on GPU architectures. We implemented on GPU the pseudocode shown in Fig. 9.1.



```
for $i = 1$ to $N_p$ do
    $\hat{J}_i \leftarrow T \cdot \Phi_i$
    for $n = 1$ to $3N_v$ do
        $\hat{J}_{n,i} \leftarrow \hat{J}_{n,i} W_{N_t}$
        $\hat{J}^{\omega}_{n,i} \leftarrow \mathrm{FFT}[\hat{J}_{n,i}]$
    end for
    $\mathcal{P}_{\hat{J}} \leftarrow \mathcal{P}_{\hat{J}} + \left|\hat{J}^{\omega}_i\right|^2$
end for
```

Figure 9.1: GPU pseudocode for the modified periodogram evaluation.

In the pseudo-code represented in Fig. 9.1, matrix T is defined in Eq. 7.4 for LORETA, Eq. 7.5 for eLORETA. Due to the non-linear standardization step, sLORETA cannot be used or would lead to inconsistent results. As in the time-domain case the matrix product between T and Φ is computed on GPU.

Window functions are easily implemented by calculating the windowing value of the selected method. The implemented CUDA routine provides four different windowing functions: Hann, Hamming, Blackman and Gauss. DFT is performed through the Fast Fourier Transform (FFT) algorithm which is already available in the NVidia CUFFT library for 1D, 2D or 3D, real or complex data. This library is called by creating an FFT plan, executing the FFT batch in place and destroying the plan. In our case, FFT size is $3 \cdot N_v$ and batch number is the number of samples for window (N_t).

As for the time-domain approach, we decided to make the library available in MATLAB. Therefore, a MATLAB MEX file is realized which runs the implemented CUDA kernels for modified periodogram calculation. As a result, the eLORETA estimate of the intracerebral current density can be launched directly within the MATLAB environment, with the significant speed-ups provided by its efficient CUDA GPU implementation.

9.2 Performance

While there is no common evidence in literature whether sLORETA outperforms eLORETA or viceversa in terms of localization accuracy, for frequency domain source localization eLORETA is the only viable solution since the non-linearity of the standardization phase doesn't allow sLORETA to be used.

The developed parallel implementation of eLORETA is therefore compared to a standard MATLAB implementation.

First, we perform source localization on the grey matter region of the segmented head described in the electrical model ($\sim 700 \cdot 10^3$ voxels). In Fig. 9.2, 64 measuring electrodes are considered, and the time required to perform frequency source localization on a varying number of time samples is considered. The number of overall time-samples considered is obtained by multiplying the number of windows N_v the number of time samples composing each window. The CUDA MEX implementation is affected by significant overheads for initiating data-transfers which limit the speed-ups for small number of windows but provides very significant speed-ups as the number of samples considered increases.

In Fig. 9.3 and Fig. 9.4, the reconstruction is performed on an downsampled grey matter region with about $90 * 10^3$ voxels and the measurements are acquired with 64 electrodes. The case illustrated in Fig. 9.3 requires signif-

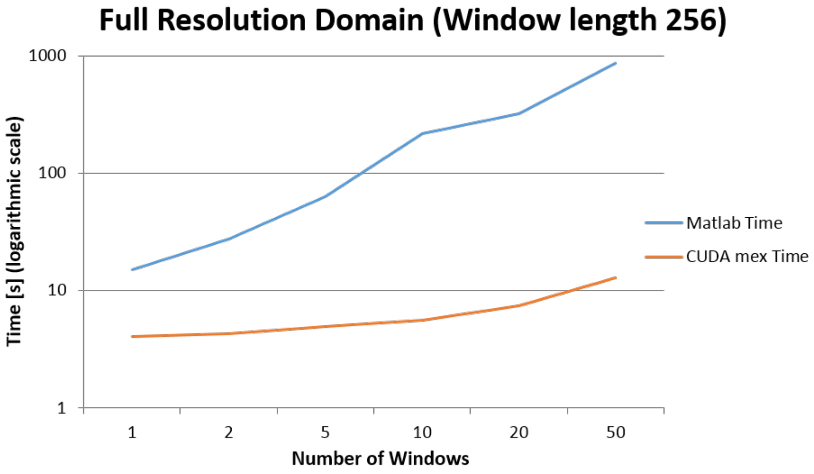


Figure 9.2: Performance comparison between MATLAB standard code and the implemented CUDA mex for frequency source localization, with a variable numbers of windows having length 256 samples.

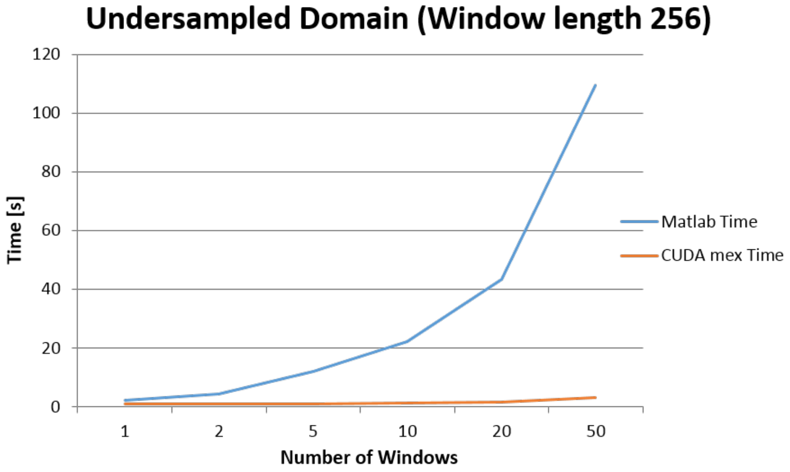


Figure 9.3: Performance comparison between Matlab standard code and the implemented CUDA mex for frequency-domain source localization in the downsampled domain, with a variable numbers of windows having length 256 samples

icantly less time than when using a full resolution grey matter region but the MATLAB implementation still requires nearly 2,5 seconds of computing time to reconstruct 256 samples while the parallel implementation can reconstruct about 50 times more samples in the same amount of time.

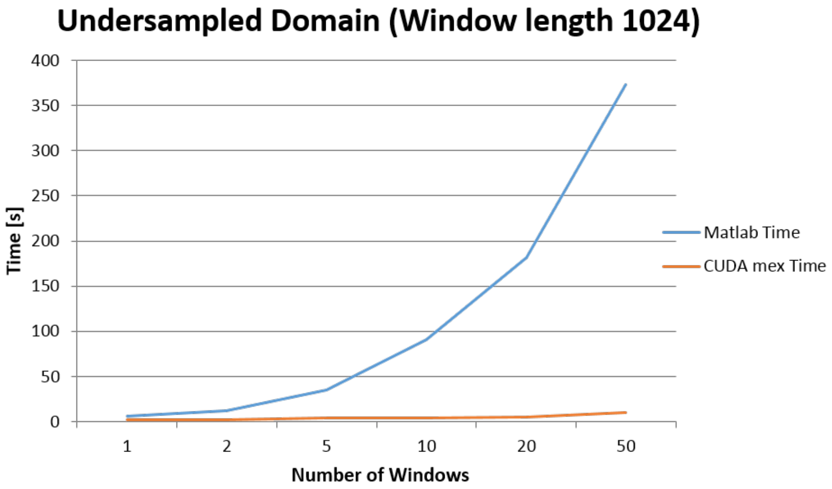


Figure 9.4: Performance comparison between Matlab standard code and the implemented CUDA mex for frequency source localization in the down-sampled domain, with a variable numbers of windows having length 1024 samples.

9.3 Experimental Validation

The objective was to determine if the algorithms for source localization in the frequency-domain work correctly for multi-trial experiments such as Event Related Potentials (ERP) obtained from visual stimuli.

In particular, we chose to apply frequency-domain source localization algorithms to steady-state visual evoked potentials (SSVEPs).

EEG signal was acquired with the cost-effective *Creamino* BCI system. For fastest setup time, the system allocates 32 dry-electrodes in the positions presented in Fig. 9.5. Ground and reference are placed on right and left earlobes respectively.

The stimuli was presented on a 24-inch LCD panel located approximately 70 cm from the subject. A square 8x8 black and white contrast checkerboard was presented, flickering at an approximate rate of 14 Hz. Three possible positions of the checkerboard on the screen were considered, one centred with respect to the visual field of the subject, one slightly displaced to the left, one to the right. The objective was to stimulate the left and right visual fields symmetrically or asymmetrically. The flickering was repeated 140 times per each sequence with a fixed interval of 68 ms. A random interval

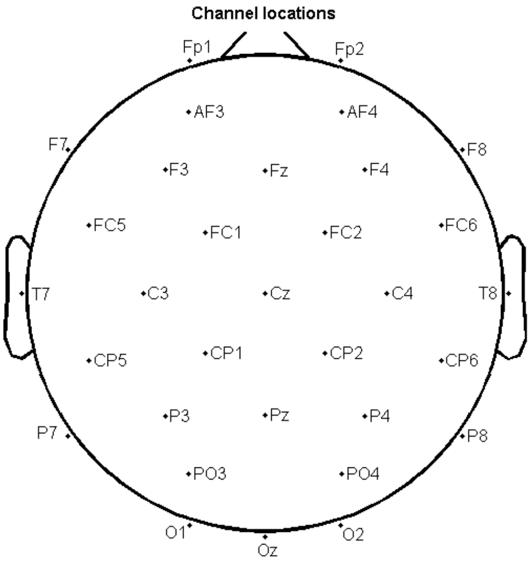


Figure 9.5: Electrode placement on the cap

of 3 to 5 seconds was inserted between each sequence in order to allow the subject to blink.

In the SSVEP experiment, the subject was comfortably seated in a quiet, dimly-lit room. In order to consider environments typical of non-clinical settings such as those envisioned for devices developed in the CREAM project, the room was not electrically or acoustically shielded. In each run, 2 repetitions per positions were presented, followed by a resting time determined by the subject by pressing a key when ready to proceed.

The objective is to locate the sources of the EEG signal at the flickering frequency, which can be supposed to be again located in the visual cortex of the contralateral hemisphere.

Sources of the ERP were reconstructed using eLORETA on the realistic head model derived from ICBM 152 Nonlinear Atlas [118][119][120]. Sources have been again localized in the contralateral visual cortex for the first harmonic (at the flickering frequency) and for the second harmonic (at twice the flickering frequency) both.

Results for the central visual field stimulus are presented in Fig. 9.6 while those relative to the right visual field stimulus are shown in Fig. 9.7. A first conclusion is that asymmetrical visual field stimulation leads to the activation of the occipital region of the contralateral hemisphere (contralateral visual cortex). The source localization reconstruction at the frequency of the second harmonic is represented in Fig. 9.8. With respect to the plot relative to the first harmonic, we can observe a significant spreading of the solution. This is probably due to the reduced signal-to-noise ratio due to the reduction of the power of the signal which is significantly lower for the second harmonic.

With respect to the time-domain source localization which allows the reconstruction of the sources of the different frequency components of the ERP, here we can only observe an average activation over the time at which stimulation is performed. However, we can confirm how the activity is located in the contralateral visual cortex leading to the experimental validation of the proposed algorithm.

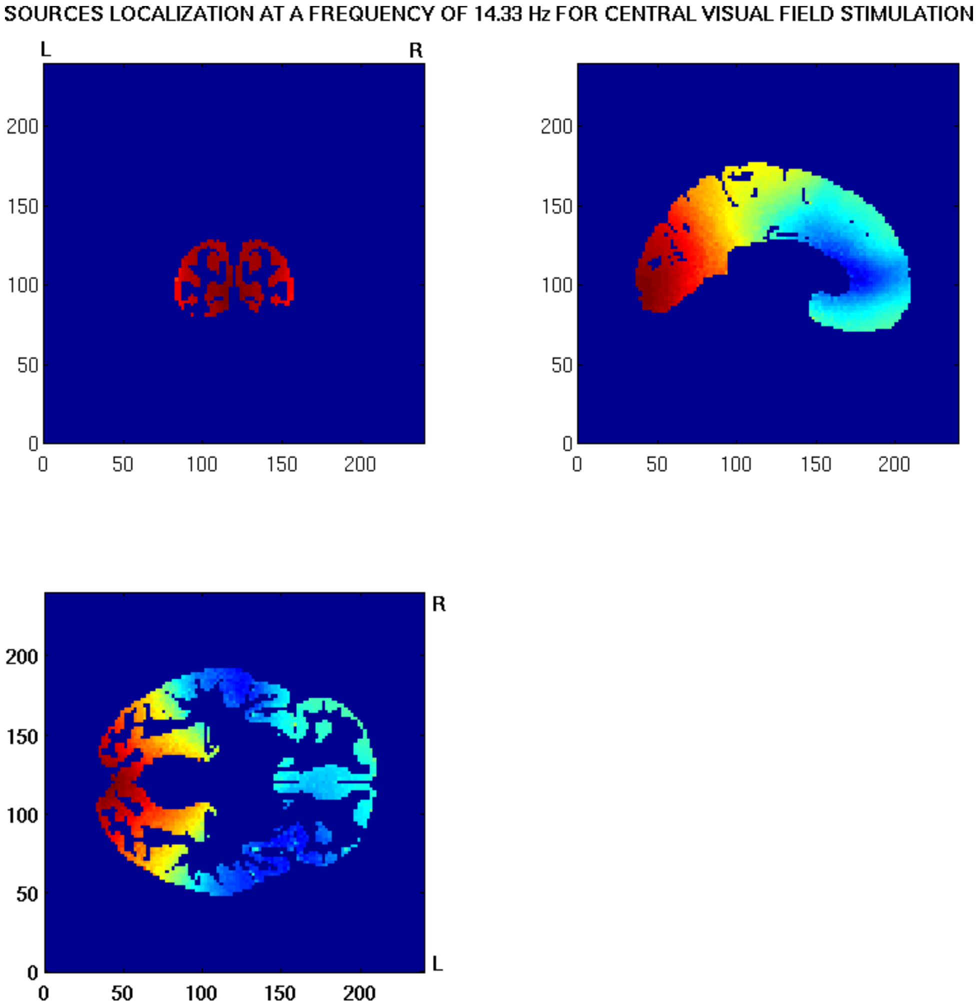

SOURCES LOCALIZATION AT A FREQUENCY OF 14.33 Hz FOR CENTRAL VISUAL FIELD STIMULATION

Figure 9.6: Reconstruction of the cortical sources at a frequency of 14 Hz for central-visual-field stimulus. Maximum is located in the occipital region.

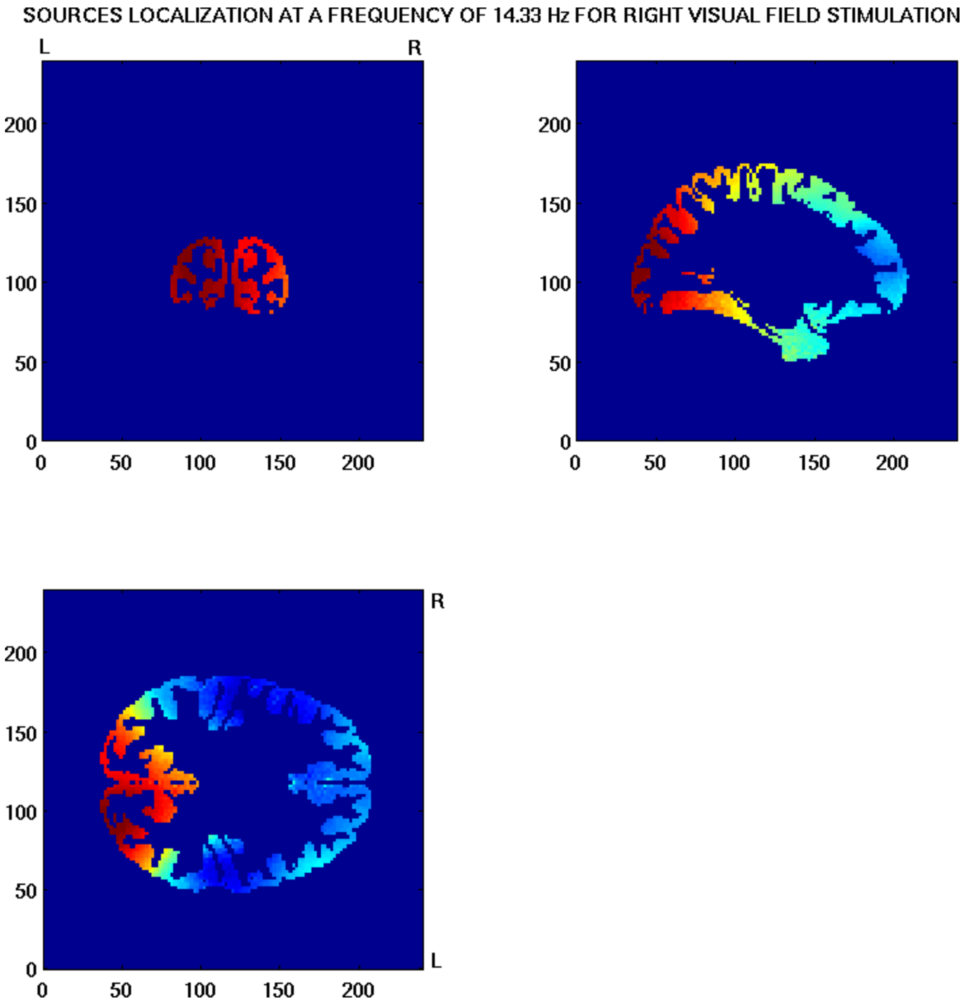


Figure 9.7: Reconstruction of the cortical sources at a frequency of 14 Hz for right-visual-field stimulus. Maximum is located in the occipital region of the contralateral hemisphere.

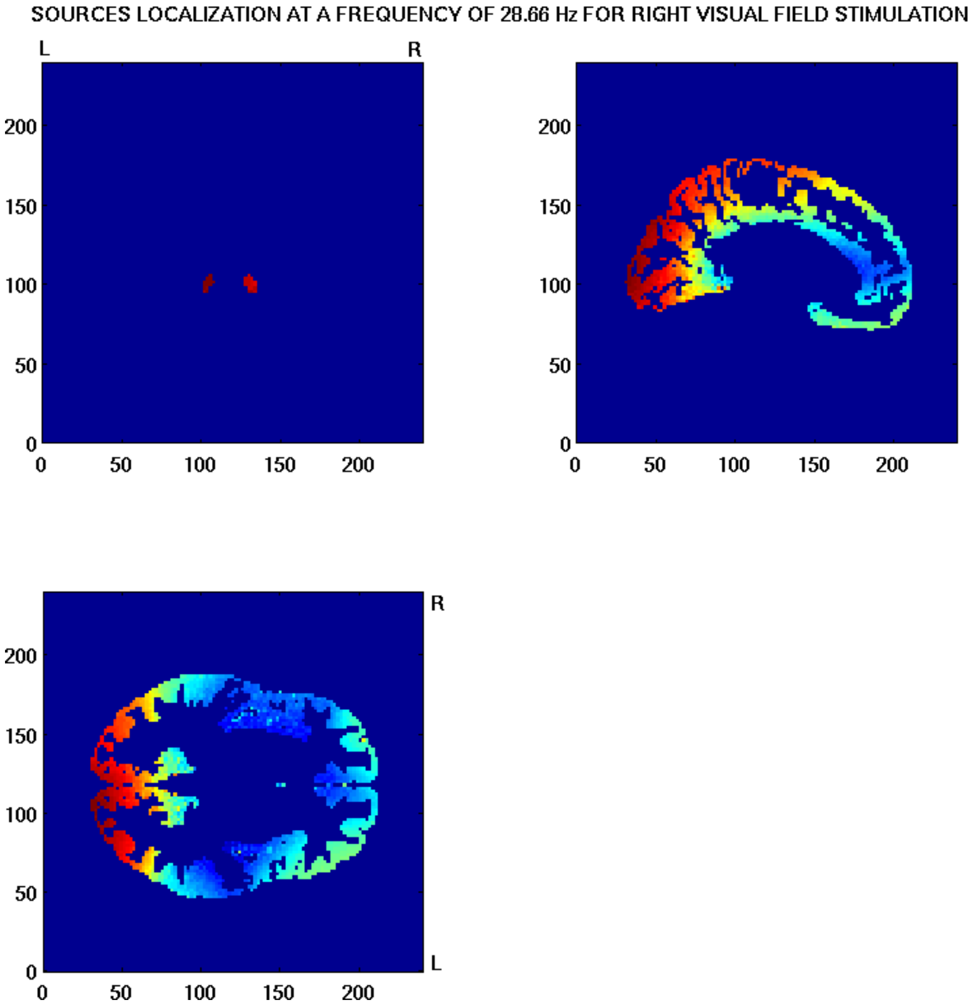


Figure 9.8: Reconstruction of the cortical sources at a frequency of 28 Hz (second harmonic of the flickering frequency) for right-visual-field stimulus.

9.4 Conclusions

In this chapter, EEG Source Localization algorithms in the frequency domain were developed and validated by multi-trial ERP experiments obtained from visual stimuli, representing a point for further work involving the implementation of SIMULINK Libraries for *Creamino* BCI system.

In complex geometries such as the whole human head, it could be difficult and onerous for the user to display EEG source localization results in the cerebral cortex through multiple 2D plots in various sections of the domain. To address this issue, in the following chapter we propose a 3D visualization toolbox able to rendering frequency/time EEG Source Localization results in the cerebral cortex enabling user interaction.

Chapter 10

Visualization Toolbox

In order to display EEG source localization results on a realistic human head, an ad-hoc visualization toolbox has been realized, which is designed to efficiently run on the same hardware as the developed imaging algorithms.
This chapter focuses on the implementation and characterization of a toolbox which allows to rendering a 3D light distribution or an EEG frequency/time source localization on a large and irregular domain, enabling the interaction between the user and the rendered surface and making easier the interpretation of results.
In general, the user needs only to provide the software with a 3D matrix whose elements represent the color intensity of the corresponding voxel (anything from a simple segmentation label to the complex result of an EEG source localization or DOT forward problem solution). The tool itself proceeds to determine the external surface of the volume, the color to be assigned and to visualize it in a 3D navigation environment where rotation, zooming and translations are available through simple movements of the mouse.

As stated previously, MRI images are segmented and classified to obtain a realistic volume composed of five tissues: scalp, skull, CSF, gray matter and white matter. This operation preserves the full resolution provided by MRI scans and leads to 3D domain composed by equally sized cubic voxels.
The complex shape and the large dimension of the segmented human head volume markedly increase the computational cost of the visualization process, making it necessary to adopt a parallel and efficient ray-tracing engine.

A pre-processing step which extracts the surfaces composing the head volume is also necessary. Furthermore, since the volume is voxel-based, an additional smoothing and regularization step is fundamental to improve the quality of the final rendering. The pre-processing step (Sec. 10.1) is realized through optimized MATLAB mex-functions, while the rendering of the surfaces (Sec. 10.2) has been realized by using NVIDIA OptiX 3.8.0.

Thanks to the complete integration with the MATLAB environment, the user enjoys a very large freedom in choosing what to plot and how to plot it. So, for example, having the segmentation and parcellation available, it is very easy to plot only certain tissues or regions by modifying the input data with very simple MATLAB operations. Similarly, sections of the volume can be easily computed and plotted.

Furthermore, the presence of a smoothing step which improves drastically the quality of the final rendering represents an original contribution in the state of the art of visualization toolboxes.

10.1 MATLAB pre-processing

Given the solution of the DOT forward problem or EEG frequency/time source localization on a generic voxel-based volume, we realized an ad-hoc MATLAB routine in order to create both the triangulation of the surface under investigation and the color of the triangulated surface.

First of all, to extract the surface of the volume under analysis, we use the MATLAB function *isosurface* which returns a structure containing the faces and the vertices of the triangles composing the triangulated surface.

After that, to define the solution within the surface given the solution on the volume underneath it, we use the MATLAB function *isocolors*. This function allows to evaluate the color of each vertex composing the triangulation, given the solution on the whole volume.

To render the solution in the surface of the volume under investigation within the MATLAB environment, the outputs of *isosurface* and *isocolors* can be passed directly to the MATLAB *patch* function.

Unfortunately, the MATLAB rendering of the surface of a segmented MRI volume, presents two main issues:

1. Irregular shape;

2. High number of triangles.

As stated before, the irregularity of the surface derives from the cubic voxels which compose the segmentation, while the high number of cubic voxels increases the computational cost and it is a consequence of the large dimension of the head domain. To regularize the sharp-cornered structure of the surfaces we adopt a free MATLAB tool: *Smooth Triangulated Mesh* by Dirk Jan Kroon [121].

This function smooths the triangulated mesh extracted using *isosurface* providing accurate curvature flow smoothing. The regularization of the surface is realized by smoothing in the direction of the normal keeping the edge ratio's the same. This toolbox also supports Laplacian smoothing with inverse vertice-distance based umbrella weights, making the edge lengths more uniform. The code is written partly in MATLAB and partly in C-code for the most computational intensive parts.

For our purposes the curvature flow smoothing with the inverse distance between vertices as weights represents the most appealing regularization approach. As a result, by introducing a smoothing step within the MATLAB pre-processing we obtain a significant reduction of the number of triangles composing the surface, and a noticeable improvement in the quality of the final rendering.

Using MATLAB is possible to visualize the result of the pre-processing step through the function *patch* but the quality of the rendering is not acceptable and it is characterized by a lack of efficiency which leads to a very low frame rate unsuitable for dynamic interaction (such as rotation or zoom) with the rendered object due to the time lag associated with these operations.

A parallel ray-tracing engine is necessary to allow a 3D dynamic and interactive visualization of the volume representing, for example, the results of the EEG source localizations or DOT forward problem solution.

Nvidia Optix 3.8.0 represents the parallel ray tracing engine that best match our requests since it is optimized to run on Nvidia GPUs on which the implemented efficient solvers have been developed. Two additional steps need to be implemented within the pre-processing phase to import the geometry of the problem within the Optix environment.

1. Creation of an Object file containing the surface to be rendered;

2. Colors specification.

To define a *Wavefront/Alias Object* from the coordinates of vertices and faces extracted through the *isosurface* function, we employ the free function *vertface2obj* available online.

For color export, given a distribution of intensities between 0 and 1 in the volume under analysis, it is sufficient to realize a binary file which contains directly the result of the *isocolor* function. To sum up, pre-processing phase involves the following steps:

1. Surface triangulation (*isosurface*);
2. Surface smoothing (*smooth triangulated surface*);
3. Colors specification in the surface (*isocolors*);
4. Object definition (*vertface2obj*);
5. Colors buffer binary file creation.

10.2 Ray-Tracing Engine

10.2.1 Object Loading

In Fig. 10.1 is shown an example of scalp loaded within the Optix environment with no colour definition (only the initialization of the light buffer has been done). In Fig. 10.1(a) it is possible to notice the irregular shape of the scalp and the high number of voxels composing the segmented MRI. The surface smoothing introduced in the pre-processing phase is fundamental to improve the quality of the final rendering (Fig. 10.1(b)). It is important to underline that this kind of smoothing is suitable for every kind of segmented tissue extracted from the segmented MRI but also for user-defined regions of the segmented head volume.

10.2.2 Surface colouring

To display a particular light/electrical distribution, the simple way is to define the colour of each vertex composing the triangulated surface. As shown in Sec. 10.1, this task is realized through the *isocolors* function, this result is stored in a binary file to import the normalized colours of the

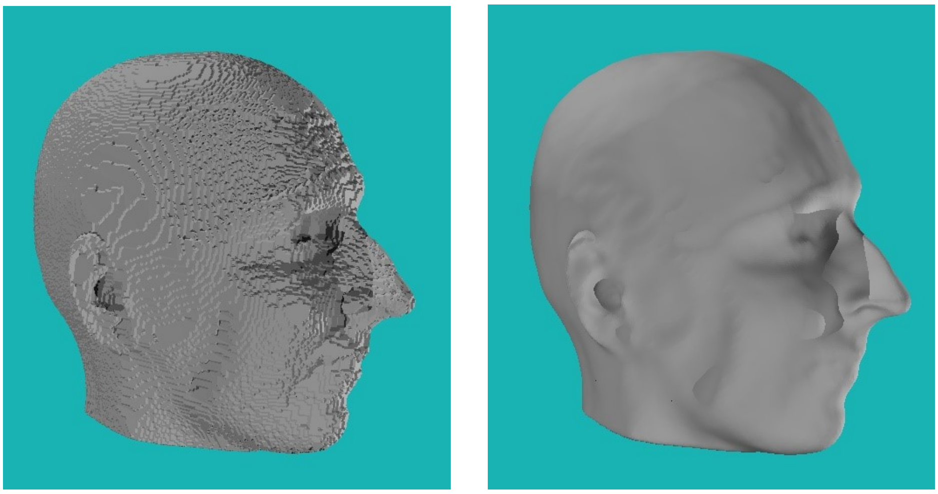

(a) (b)

Figure 10.1: Object loaded within OptiX environment:(a) scalp object with no smoothing step in the MATLAB pre-processing, (b) smoothed object through smooth triangle mesh routine.

vertex composing the triangulation into the Optix environment. To define the colour of a particular point within the surface, it is first necessary to transform the normalized solution intensities into RGB values, after the selection of a colormap. The definition of a colormap has been realized through an ad-hoc C function in the Optix routine. In the case we need to display only the surface in a uniform RGB colour, the solution can be fixed to 1 for every point composing the surface and the RGB colour can be forced to the selected RGB colour. To display intensity variations we adopt the MATLAB *jet* colormap.

Once the colormap has been defined, the vertex colours stored in the binary file have been transformed into RGB scale and saved in an Optix GPU Buffer for increased speed. To define the colour of a point in a triangulated surface, the standard Optix routine identifies the vertices of the triangle that has been hit by a parallel ray and it defines the colour of the intersection point in function of the material properties of the triangle. To assign a user-defined colour to the triangle, it is necessary to customize this function, ignoring face properties (not defined in the Object file). Once the vertices of the hit triangle have been identified, the Optix colour buffer allows to determine them RGB colours.

The barycentric interpolation of each RGB tone of the triangle vertex colours allows to define a unique RGB colour for each point within the triangle surface, with also a smoothing effect on the final rendering colour in the whole surface. As stated before, it is also possible to fix the RGB colour of the whole surface displaying its anatomical structure.

For example, fixing the RGB colour to pink (1.0,0.73,0.58) and the intensities equal to one and loading the scalp surface in the Optix environment leads to the result shown in Fig. 10.2.

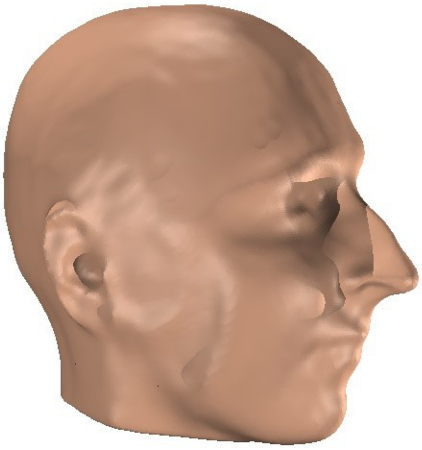
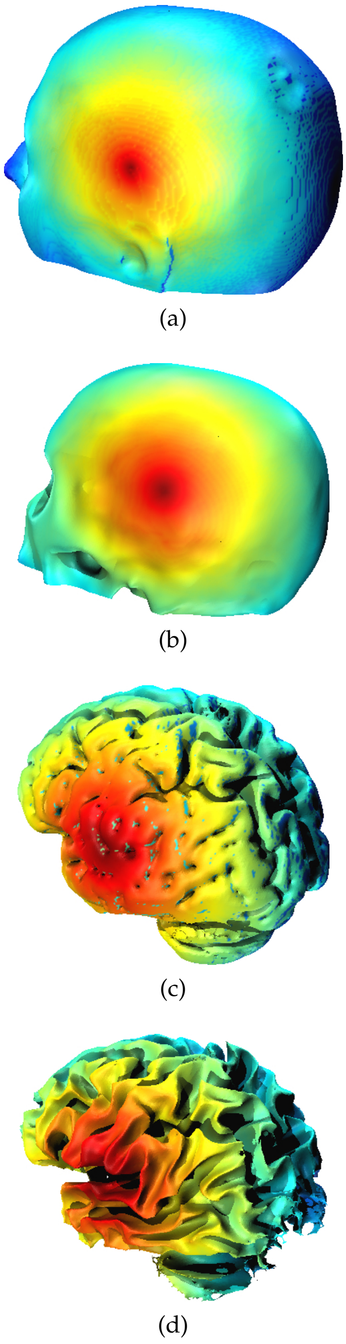


Figure 10.2: Scalp display with static and user-defined RGB colour.

The DOT forward problem solution computed in Sec. 3.8 and 2D-represented in the right column of Fig. 3.23, is processed and visualized with the *jet* colormap through the implemented toolbox, leading to the result which can be appreciated in Fig. 10.3.

Thanks to OptiX parallel ray-tracing engine implementation, a variable number of 20 to 60 frames per second have been obtained (instead of 0.11 fps of MATLAB ray-tracing), leading to a fluid and efficient 3D interactive representation of the results. We have also introduced the possibility to plot a transition between multiple solutions. It is sufficient to store multiple colors buffer vertically concatenated in the colors binary file and the OptiX tools displays a transition computed by linear interpolation of the RGB colours composing the sampled solutions, with the time evolution indication in the bottom-left corner of the output image.

Using the visualization toolbox is also possible to obtain the 3D illustration of the domain using the sLORETA output of the experimental test described



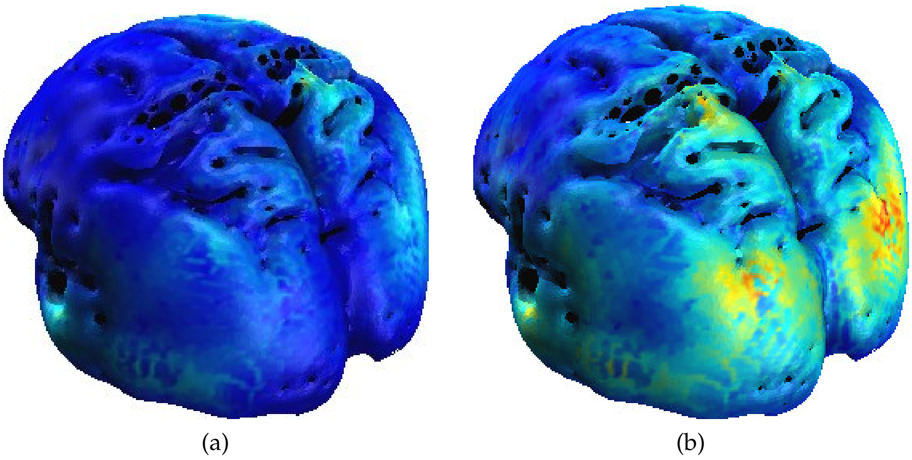
(a)

(b)

(c)

(d)

Figure 10.3: Display of a light distribution with *jet* colormap on the scalp (a), skull (b), grey matter (c) and white matter (d) of a structural MRI of a human head.



(a) (b)

Figure 10.4: 3D illustration of the domain using the sLORETA output of the experimental test described in Sec. 8.7. Mean power of alpha signal is shown in the occipital region during eyes open phase (a) and eyes closed phase (b).

in Sec. 8.7. This result is shown in Fig. 10.4. It is possible to appreciate that during eyes-open phase the mean power of the alpha signal in the occipital region illustrated in Fig. 10.4(a) is lower than the mean power of the same signal evaluated in the eyes-closed phase of Fig. 10.4(b). Finally, in Fig. 10.5, 10.6 and 10.7 we propose the 3D rendering of the frequency source localization SSVEPs experiment described in Sec. 9.3. In particular, the three slices represented in Fig. 9.6 are extracted from the volume shown in Fig. 10.5, Fig. 9.7 is related to Fig. 10.6 and Fig. 10.7 is the rendering of the slices of Fig. 9.8.

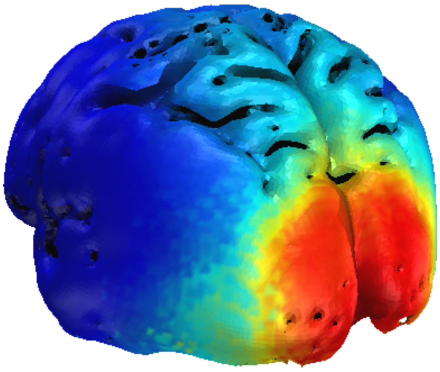


Figure 10.5: 3D Reconstruction of the cortical sources at a frequency of 14 Hz for central-visual-field stimulus.

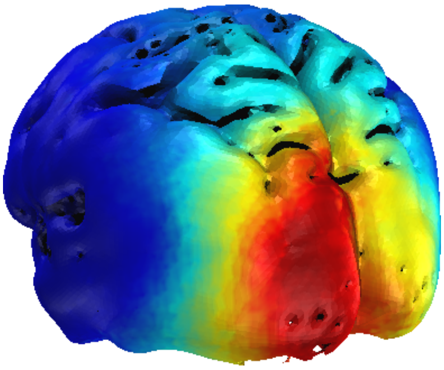


Figure 10.6: 3D Reconstruction of the cortical sources at a frequency of 14 Hz for right-visual-field stimulus.

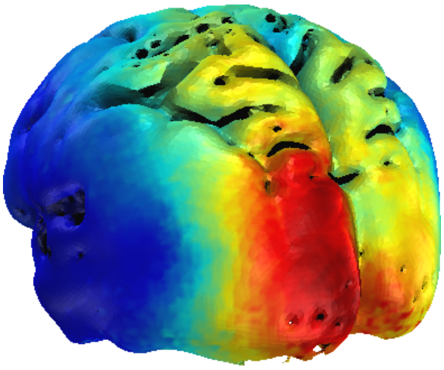


Figure 10.7: 3D Reconstruction of the cortical sources at a frequency of 28 Hz for right-visual-field stimulus.

10.3 Conclusions

This Chapter focuses on the implementation of an interactive and accelerated 3D visualization toolbox based on OptiX 3.8.0 able to display intensity distribution on complex and large 3D surfaces, leading to a variable number of 20 to 60 frames per second instead of 0.11 fps of MATLAB ray-tracing. This tool allows to display frequency/time EEG Source Localization results and light distribution obtained by the proposed DOT solver treated on the Section A of this work, but also homogeneous surfaces such as the segmented MRI tissues.

Chapter 11

Conclusions Section B

As shown before, the main contribution of my research activity is presented in the Section A of this thesis while a secondary contribution is presented in Section B.
The secondary contribution presented above concerns three different key-points:

- development of SIMULINK libraries for EEG Source Localization in the time domain;
- numerical implementation of massively parallel EEG Source localization algorithms in the frequency domain;
- realization of a rendering toolbox for ease user-interpretation of results.

The implemented SIMULINK libraries for EEG Source Localization are able to compute the mean power of the EEG sources lying in user-defined regions of interest (ROIs) and derive from the Arduino-based cost-effective EEG system called *Creamino* which I contributed to develop with the ARCES research team.
We proof that performance of the proposed libraries are fast enough for BCI typical applications under different constraints, depending on the number of voxels composing the ROIs, the buffer dimension and the sample rate of the BCI system.
For example, the proposed libraries are fast enough for real-time behaviour of *Creamino* BCI system applications with a maximum number of $150 \cdot 10^3$

voxels in a 32 channels system interfaced with a standard laptop computer both for eLORETA and sLORETA reconstruction methods.
Schematics, gerber files, source code and software modules of *Creamino* are available with full documentation and free of charge for research and educational purposes online at https://github.com/mchiesi/Creamino.

The aim of the second point is to illustrate the parallel implementation of frequency EEG Source Localization algorithms and its validation through multi-trial Event Related Potentials (ERP) experiments obtained from visual stimuli, representing a point for further work involving the implementation of SIMULINK Libraries for *Creamino* BCI system. The proposed method is based on a commonly used Welch-periodogram of eLORETA current density estimation, where critical kernels are massively parallel optimized for the available HPC platform.

In complex geometries such as the whole human head, it could be difficult and onerous for the user to display EEG source localization results in the cerebral cortex through multiple 2D plots in various sections of the domain. To address this issue the third point involves the implementation of an interactive and accelerated 3D visualization toolbox based on OptiX 3.8.0.

Future lines of research could be the integration of electrical stimulation and EIT in the *Creamino* BCI system in order to improve the quality of reconstruction and the implementation of SIMULINK libraries of the proposed frequency-domain EEG Source Localization.

Chapter 12

Conclusions

The main contribution of my research activity was presented in the Section A of this thesis.

It concerns the implementation of an innovative and complete framework for DOT forward problem solution.

The model is based on a radiosity-diffusion (RD) algorithm which takes into account the different behaviour of light propagation within clear and scattering tissues, leading to results which are more accurate than with lower complexity diffusion models, and performance that improve on Monte Carlo (MC) methods. MC methods, being RTE-based solvers, have the most generality, with the ability to manage the scattering anisotropy within biological tissues, the low-scattering property of CSF, and short source-detector separations. On the other hand, the proposed RD solver is characterized by a lower computational cost and no granularity in deeper regions such as gray or white matter. Furthermore, the ability to manage the complex shape of a non-scattering CSF layer in a human head represents a remarkable improvement over standard diffusion-based forward problem solvers.

The numerical solver for the diffusion equation is based on an FVM discretization of the human head and its critical kernels are optimized to run on high performance computing platforms based on GPUs. Modeling of light propagation in clear regions is obtained through a radiosity-algorithm based on a customized parallel ray-tracing designed to run efficiently on the same hardware. The ray-tracing engine has been designed to optimize form factors computation, the most critical phase of the pre-processing step. Integration of the two models is performed by an iterative solver which does

not compromise the efficiency of the diffusion equation solver. We obtained a meaningful speed-up if compared to an isotropic-scattered parallel MC engine for a domain of 2 million voxels, with a marked improvement in accuracy. The speed-up significantly increases for larger domains, allowing one to compute the light distribution of a full human head ($\approx$ 5 million total voxels, $\approx$ 3 million non-zero voxels) in 116 seconds for the platform used.
The implemented DOT solver was also employed to validate experimental measurements made on a phantom mimicking the optical properties of a realistic human head, identifying an experimental setup issue leading the optical sensors operating in a non-linear range which is a point for further work once a complete DOT system is available.
This work is presented in details in [O1] and it represents a meaningful update in the state of the art of DOT forward problem solvers. It is also described in HIGH PROFILE Deliverables [O2][O3]. This work was presented at the High Profile meeting of Vienna [O4].

The secondary contribution of my research activity was presented in the Section B of this thesis.
It concerns the development of SIMULINK libraries for EEG Source Localization able to compute the mean power of the EEG sources lying in user-defined regions of interest (ROIs).
This work derives from the Arduino-based cost-effective EEG system called *Creamino* which I contributed to develop with the ARCES research team [O5][O6][O7].
EEG Source Localization algorithms in the frequency domain were also developed and validated by multi-trial ERP experiments obtained from visual stimuli, representing a point for further work involving the implementation of SIMULINK Libraries for *Creamino* BCI system.
To tackle the problem of interpreting EEG source localization results usually solved by multiple 2D plots of the solution in various sections of the cerebral cortex, an interactive and accelerated 3D visualization toolbox based on OptiX 3.8.0 was developed.
The work presented in Section B of this thesis is fully described in CREAM European Project Deliverables [O8] [O9].

Appendix A

Acronyms

API Application Programming Interface

BCI Brain Computer Interface

CPU Central Processing Unit

CSF Cerebro Spinal Fluid

CUDA Compute Unified Device Architecture

DBC Dirichlet Boundary Condition

DE Diffusion Equation

DFT Discrete Fourier Transform

DOT Diffuse Optical Tomography

EEG ElectroEncephaloGraphy

ERP Event Related Potential

fMRI functional Magnetic Resonance Imaging

FLOPs Floating Point Operations per second

GPU Graphics Processing Unit

HPC High Performance Computing

LSQR Least SQuared Residuals

MRI Magnetic Resonance Imaging

NBC Neumann Boundary Condition

NIR Near-InfraRed

NIRS	Near-InfraRed Spectroscopy
PET	Positron Emittance Tomography
RAM	Random Memory Access
RBC	Robin Boundary Condition
RTE	Radiative Transport Equation
SiPM	Silicon Photomultiplier
SR	Success Rate
SSVEP	Steady-State Visual Evoked Potential

Appendix B

Availability of *Creamino* to Other Research Groups

Creamino, with schematics, gerber files, bill of materials, executables, source code and documentation is available free of charge for research and educational purposes at https://github.com/mchiesi/Creamino. Firmware and libraries will be maintained and updated. Documentation suitable for designers as well as for end users is provided.

Author's Bibliography

[O1] S Placati, M Guermandi, A Samore, Scarselli E Franchi, and R Guerrieri. Parallel solver for diffuse optical tomography on realistic head models with scattering and clear regions. *IEEE Transactions on Bio-medical Engineering*, 63(9):1874–1886, 2015.

[O2] S Placati, M Guermandi, and R Guerrieri. Software for NIRS and MRI data fusion. High Profile Project Deliverable 4.3.1, April 2014.

[O3] S Placati, M Guermandi, R Guerrieri, D Sanfilippo, P G Fallica, and R Pagano. Phantom-based validation of NIRS system. High Profile Project Deliverable 6.5.1, June 2014.

[O4] S Placati, A Samore', M Guermandi, E Franchi Scarselli, and R Guerrieri. Merging structural information from mri/dti to eeg/eit and nirs. In *HIGH Profile Workshop Poster*, 2014.

[O5] M Chiesi, M Guermandi, S Placati, A Bigucci, E Franchi Scarselli, and R Guerrieri. Creamino: An open source framework for fast development of eeg-based bci systems. *IEEE Transactions on bio-medical engineering*.

[O6] M Guermandi, A Bigucci, E Franchi Scarselli, and R Guerrieri. Eeg acquisition system based on active electrodes with common-mode interference suppression by driving right leg circuit. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3169–3172. IEEE, 2015.

[O7] A Samore', M Guermandi, S Placati, and R Guerrieri. Automatic detection and classification of compact conductivity contrasts with electrical impedance tomography. *IEEE Transactions on Instrumentation and Measurement*, 2016.

[O8] S Placati, A Samore', M Guermandi, M Chiesi, R Guerrieri, E Pereda, and J G Prieto. Software libraries for brain imaging algorithms tuned for multiple data sets and report on the validation on ERP. CREAM Project Deliverable 4.2, December 2015.

[O9] S Placati, M Guermandi, M Chiesi, and R Guerrieri. Software libraries for stimulus production. CREAM Project Deliverable 4.5, April 2016.

Bibliography

[1] S Placati, M Guermandi, A Samore, Scarselli E Franchi, and R Guerrieri. Parallel solver for diffuse optical tomography on realistic head models with scattering and clear regions. *IEEE Transactions on Bio-medical Engineering*, 63(9):1874–1886, 2015.

[2] David A Boas and Anders M Dale. Simulation study of magnetic resonance imaging–guided cortically constrained diffuse optical tomography of human brain function. *Applied optics*, 44(10):1957–1968, 2005.

[3] David A Boas, Anders M Dale, and Maria Angela Franceschini. Diffuse optical imaging of brain activation: approaches to optimizing image sensitivity, resolution, and accuracy. *Neuroimage*, 23:S275–S288, 2004.

[4] Alexander D Klose, Vasilis Ntziachristos, and Andreas H Hielscher. The inverse source problem based on the radiative transfer equation in optical molecular imaging. *Journal of Computational Physics*, 202(1):323–345, 2005.

[5] SR Arridge, M Schweiger, M Hiraoka, and DT Delpy. A finite element approach for modeling photon transport in tissue. *Medical physics*, 20(2):299–309, 1993.

[6] Simon R Arridge, Hamid Dehghani, Martin Schweiger, and Eiji Okada. The finite element model for the propagation of light in scattering media: a direct method for domains with nonscattering regions. *Medical physics*, 27(1):252–264, 2000.

[7] AP Gibson, JC Hebden, and Simon R Arridge. Recent advances in diffuse optical imaging. *Physics in medicine and biology*, 50(4):R1, 2005.

[8] Simon R Arridge. Optical tomography in medical imaging. *Inverse problems*, 15(2):R41, 1999.

[9] RAJ Groenhuis, Hedzer A Ferwerda, and JJ Ten Bosch. Scattering and absorption of turbid materials determined from reflection measurements. 1: Theory. *Applied Optics*, 22(16):2456–2462, 1983.

[10] Michael Firbank, Simon R Arridge, Martin Schweiger, and David T Delpy. An investigation of light transport through scattering bodies

with non-scattering regions. *Physics in medicine and biology*, 41(4):767, 1996.

[11] Defu Yang, Xueli Chen, Shenghan Ren, Xiaochao Qu, Jie Tian, and Jimin Liang. Influence investigation of a void region on modeling light propagation in a heterogeneous medium. *Applied optics*, 52(3):400–408, 2013.

[12] J Riley, Hamid Dehghani, Martin Schweiger, Simon R Arridge, Jorge Ripoll, and Manuel Nieto-Vesperinas. 3d optical tomography in the presence of void regions. *Optics express*, 7(13):462–467, 2000.

[13] David Boas, J Culver, J Stott, and A Dunn. Three dimensional monte carlo code for photon migration through complex heterogeneous media including the adult human head. *Optics express*, 10(3):159–170, 2002.

[14] Qianqian Fang and David A Boas. Monte carlo simulation of photon migration in 3d turbid media accelerated by graphics processing units. *Optics express*, 17(22):20178–20190, 2009.

[15] Martin Schweiger and Simon R Arridge. Comparison of two-and three-dimensional reconstruction methods in optical tomography. *Applied optics*, 37(31):7419–7428, 1998.

[16] Matteo Chiesi, Luca Vanzolini, Claudio Mucci, Eleonora Franchi Scarselli, and Roberto Guerrieri. Power-aware job scheduling on heterogeneous multicore architectures. *IEEE Transactions on Parallel and Distributed Systems*, 26(3):868–877, 2015.

[17] S Lee, Y Shin, S Woo, K Kim, HN Lee, and R Fazel-Rezai. Review of wireless brain-computer interface systems. *Brain-Computer Interface Systems-Recent Progress and Future Prospects. InTech*, 2013.

[18] Lars-Göran Nilsson and Hans J Markowitsch. *Cognitive neuroscience of memory*. Hogrefe & Huber Pub, 1999.

[19] Seong-Gi Kim, Wolfgang Richter, and KÇŐmil UÇğurbil. Limitations of temporal resolution in functional mri. *Magnetic resonance in medicine*, 37(4):631–636, 1997.

[20] Paul L Nunez and Ramesh Srinivasan. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[21] RH Bayford. Bioimpedance tomography (electrical impedance tomography). *Annu. Rev. Biomed. Eng.*, 8:63–91, 2006.

[22] Erik Lindholm, John Nickolls, Stuart Oberman, and John Montrym. Nvidia tesla: A unified graphics and computing architecture. *IEEE micro*, 28(2):39–55, 2008.

[23] Mohammad Modarreszadeh and Robert N Schmidt. Wireless, 32-channel, eeg and epilepsy monitoring system. In *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 3, pages 1157–1160. IEEE, 1997.

[24] David S Holder. *Electrical impedance tomography: methods, history and applications*. CRC Press, 2004.

[25] David A Boas, Dana H Brooks, Eric L Miller, Charles A DiMarzio, Misha Kilmer, Richard J Gaudette, and Quan Zhang. Imaging the body with diffuse optical tomography. *IEEE signal processing magazine*, 18(6):57–75, 2001.

[26] Michal Teplan. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.

[27] J Steinbrink, M Kohl, Hellmuth Obrig, G Curio, F Syre, F Thomas, H Wabnitz, H Rinneberg, and Arno Villringer. Somatosensory evoked fast optical intensity changes detected non-invasively in the adult human head. *Neuroscience Letters*, 291(2):105–108, 2000.

[28] Geert Morren, Martin Wolf, Philippe Lemmerling, Ursula Wolf, Jee H Choi, Enrico Gratton, Lieven De Lathauwer, and Sabine Van Huffel. Detection of fast neuronal signals in the motor cortex from functional near infrared spectroscopy measurements using independent component analysis. *Medical and Biological Engineering and Computing*, 42(1):92–99, 2004.

[29] Pedro Ponce, Arturo Molina, David C Balderas, and Dimitra Grammatikou. Brain computer interfaces for cerebral palsy. 2014.

[30] Tri Giang Phan and Andrew Bullen. Practical intravital two-photon microscopy for immunological research: faster, brighter, deeper. *Immunology and cell biology*, 88(4):438–444, 2010.

[31] Simon R Arridge and John C Schotland. Optical tomography: forward and inverse problems. *Inverse Problems*, 25(12):123010, 2009.

[32] Mingze Li, Fei Liu, Bin Zhang, and Jing Bai. Diffuse optical tomography guided adaptive reconstruction in fluorescence molecular tomography. *Journal of Electronic Imaging*, 21(2):023014–1, 2012.

[33] Roberto Cardu. Analog signal acquisition and conditioning for near-field capacitive communication and active combined eeg-eit monitoring. 2012.

[34] FRED PLUM. Handbook of electroencephalography and clinical neurophysiology. *Archives of Neurology*, 26(6):556–556, 1972.

[35] Christoph M Michel, Micah M Murray, Göran Lantz, Sara Gonzalez, Laurent Spinelli, and Rolando Grave de Peralta. Eeg source imaging. *Clinical neurophysiology*, 115(10):2195–2222, 2004.

[36] Sylvain Baillet, John C Mosher, and Richard M Leahy. Electromagnetic brain mapping. *IEEE Signal processing magazine*, 18(6):14–30, 2001.

[37] Robert S Macleod, JG Stinstra, S Lew, Ross T Whitaker, DJ Swenson, MJ Cole, J Krüger, DH Brooks, and Christopher R Johnson. Subject-specific, multiscale simulation of electrophysiology: a software pipeline for image-based models and application examples. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1896):2293–2310, 2009.

[38] Gang Wang, Gregory Worrell, Lin Yang, Christopher Wilke, and Bin He. Interictal spike analysis of high-density eeg in patients with partial epilepsy. *Clinical neurophysiology*, 122(6):1098–1105, 2011.

[39] Akira Ishimaru. *Wave propagation and scattering in random media*, volume 2. Academic press New York, 1978.

[40] T Khan and A Thomas. On derivation of the radiative transfer equation and its spherical harmonics approximation for scattering media with spatially varying refractive indices. *Clemson University Mathematical Sciences Technical Report*, pages 1–49, 2004.

[41] Michael I Mishchenko. Vector radiative transfer equation for arbitrarily shaped and arbitrarily oriented particles: a microphysical derivation from statistical electromagnetics. *Applied optics*, 41(33):7114–7134, 2002.

[42] RAJ Groenhuis, JJ Ten Bosch, and Hedzer A Ferwerda. Scattering and absorption of turbid materials determined from reflection measurements. 2: Measuring method and calibration. *Applied optics*, 22(16):2463–2467, 1983.

[43] P Surya Mohan, Tanja Tarvainen, Martin Schweiger, Aki Pulkkinen, and Simon R Arridge. Variable order spherical harmonic expansion scheme for the radiative transport equation using finite elements. *Journal of Computational Physics*, 230(19):7364–7383, 2011.

[44] Ashley J Welch and Martin JC Van Gemert. *Optical-thermal response of laser-irradiated tissue*, volume 2. Springer, 2011.

[45] Alexander D Klose, Uwe Netz, Jürgen Beuthan, and Andreas H Hielscher. Optical tomography using the time-independent equation of radiative transferâĂŤpart 1: forward model. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 72(5):691–713, 2002.

[46] Tanja Tarvainen, Marko Vauhkonen, Ville Kolehmainen, and Jari P Kaipio. Hybrid radiative-transfer–diffusion model for optical tomography. *Applied optics*, 44(6):876–886, 2005.

[47] SL Jacques and Scott A Prahl. Ece532 biomedical optics. *Oregon Graduate Institute*, 1998.

[48] Setsuo Takatani and Marshall D Graham. Theoretical analysis of diffuse reflectance from a two-layer tissue model. *Biomedical Engineering, IEEE Transactions on*, (12):656–664, 1979.

[49] Sten Madsen, Michael S Patterson, Brian C Wilson, Satish M Jaywant, and Andreas Othonos. Numerical modeling and experimental studies of light pulse propagation in inhomogeneous random media. In *OE/LASE'93: Optics, Electro-Optics, & Laser Applications in Science& Engineering*, pages 90–102. International Society for Optics and Photonics, 1993.

[50] Joshua B Fishkin and Enrico Gratton. Propagation of photon-density waves in strongly scattering media containing an absorbing semi-infinite plane bounded by a straight edge. *JOSA A*, 10(1):127–140, 1993.

[51] RF Bonner, R Nossal, S Havlin, and GH Weiss. Model for photon migration in turbid biological media. *JOSA A*, 4(3):423–432, 1987.

[52] Lihong V Wang and Steven L Jacques. Analysis of diffusion theory and similarity relations for light reflectance by turbid media. In *OE/LASE'93: Optics, Electro-Optics, & Laser Applications in Science& Engineering*, pages 107–116. International Society for Optics and Photonics, 1993.

[53] Lee F Suddeath, Vivek Sahai, Andrew Wisler, Christina L Hutchinson, Britton Chance, and Eva M Sevick-Muraca. Finite element solution of the forward imaging problem associated with time-and frequency-domain measurements of photon migration. In *OE/LASE'93: Optics, Electro-Optics, & Laser Applications in Science& Engineering*, pages 117–128. International Society for Optics and Photonics, 1993.

[54] Thomas J Farrell, Michael S Patterson, and Brian Wilson. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties invivo. *Medical physics*, 19(4):879–888, 1992.

[55] M Schweiger, SR Arridge, M Hiraoka, and DT Delpy. The finite element method for the propagation of light in scattering media: boundary and source conditions. *Medical physics*, 22(11):1779–1792, 1995.

[56] David W Shattuck and Richard M Leahy. Brainsuite: an automated cortical surface identification tool. *Medical image analysis*, 6(2):129–142, 2002.

[57] David T Delpy, Mark Cope, Pieter van der Zee, SR Arridge, Susan Wray, and JS Wyatt. Estimation of optical pathlength through tissue from direct time of flight measurement. *Physics in medicine and biology*, 33(12):1433, 1988.

[58] Eiji Okada, Michael Firbank, Martin Schweiger, Simon R Arridge, Mark Cope, and David T Delpy. Theoretical and experimental investigation of near-infrared light propagation in a model of the adult head. *Applied optics*, 36(1):21–31, 1997.

[59] Veronica S Hollis, Tiziano Binzoni, and David T Delpy. Noninvasive monitoring of brain tissue temperature by near-infrared spectroscopy. In *BiOS 2001 The International Symposium on Biomedical Optics*, pages 470–481. International Society for Optics and Photonics, 2001.

[60] David A Boas, Constantinos Pitris, and Nimmi Ramanujam. *Handbook of biomedical optics*. CRC press, 2011.

[61] M Firbank, M Hiraoka, M Essenpreis, and DT Delpy. Measurement of the optical properties of the skull in the wavelength range 650-950 nm. *Physics in medicine and biology*, 38(4):503, 1993.

[62] Gilberto Branco. *The development and evaluation of head probes for optical imaging of the infant head*. 2007.

[63] C Rebecca Simpson, Matthias Kohl, Matthias Essenpreis, and Mark Cope. Near-infrared optical properties of ex vivo human skin and subcutaneous tissues measured using the monte carlo inversion technique. *Physics in medicine and biology*, 43(9):2465, 1998.

[64] Stephen T Flock, Brian C Wilson, and Michael S Patterson. Total attenuation coefficients and scattering phase functions of tissues and phantom materials at 633 nm. *Medical Physics*, 14(5):835–841, 1987.

[65] Jae Hoon Lee, Seunghwan Kim, and Youn Tae Kim. Diffuse–diffuse photon coupling via nonscattering void in the presence of refractive index mismatch on the void boundary. *Medical physics*, 31(8):2237–2248, 2004.

[66] Michael JD Cook and Zoltan J Koles. A high-resolution anisotropic finite-volume head model for eeg source analysis. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 4536–4539. IEEE, 2006.

[67] Tommaso De Marco, Florian Ries, Marco Guermandi, and Roberto Guerrieri. Eit forward problem parallel simulation environment with anisotropic tissue and realistic electrode models. *IEEE Transactions on Biomedical Engineering*, 59(5):1229–1239, 2012.

[68] CUDA Nvidia. Compute unified device architecture programming guide. 2007.

[69] Leonardo Dagum and Ramesh Menon. Openmp: an industry standard api for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55, 1998.

[70] Florian Ries, Tommaso De Marco, and Roberto Guerrieri. Tuning solution of large non-hermitian linear systems on multiple graphics processing unit accelerated workstations. *International Journal of High Performance Computing Applications*, 26(3):296–309, 2012.

[71] Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. Optix: a general purpose ray tracing engine. In *ACM Transactions on Graphics (TOG)*, volume 29, page 66. ACM, 2010.

[72] Tomas Nikodym. Ray tracing algorithm for interactive applications. *Czech Technical University in Prague*, 2010.

[73] Xiaofeng Zhang. Construction of the jacobian matrix for fluorescence diffuse optical tomography using a perturbation monte carlo method. In *SPIE BiOS*, pages 82160O–82160O. International Society for Optics and Photonics, 2012.

[74] Jin Chen and Xavier Intes. Time-resolved perturbation monte carlo for 3d optical imaging in small animals. *transformation*, 10(3):2, 2008.

[75] Juliette Selb, Tyler M Ogden, Jay Dubb, Qianqian Fang, and David A Boas. Comparison of a layered slab and an atlas head model for monte carlo fitting of time-domain near-infrared spectroscopy data of the adult head. *Journal of biomedical optics*, 19(1):016010–016010, 2014.

[76] P G Fallica, D Sanfilippo, A Busacca, and C Giaconia. System prototype of this nirs platform. High Profile Project Deliverable 3.4.2, January 2013.

[77] Andreas H Hielscher, Alexander D Klose, and Kenneth M Hanson. Gradient-based iterative image reconstruction scheme for time-resolved optical tomography. *IEEE Transactions on medical imaging*, 18(3):262–271, 1999.

[78] Simon R Arridge and Martin Schweiger. A gradient-based optimisation scheme for optical tomography. *Optics Express*, 2(6):213–226, 1998.

[79] RL Barbour, H Graber, R Aronson, and J Lubowsky. Model for 3-d optical imaging of tissue. In *Geoscience and Remote Sensing Symposium, 1990. IGARSS'90.'Remote Sensing Science for the Nineties'., 10th Annual International*, pages 1395–1399. IEEE, 1990.

[80] Howard R Gordon. Equivalence of the point and beam spread functions of scattering media: a formal demonstration. *Applied optics*, 33(6):1120–1122, 1994.

[81] Simon R Arridge and M Schweiger. Photon-measurement density functions. part 2: Finite-element-method calculations. *Applied Optics*, 34(34):8026–8037, 1995.

[82] Silvio Placati, Marco Guermandi, Roberto Guerrieri, Delfo Sanfilippo, Piero Giorgio Fallica, and Roberto Pagano. Phantom-based validation of NIRS system. High Profile Project Deliverable 6.5.1, June 2014.

[83] Emilio Sciacca, Andrea C Giudice, Delfo Sanfilippo, Franco Zappa, Salvatore Lombardo, Rosario Consentino, Cinzia Di Franco, Massimo Ghioni, Giorgio Fallica, Giovanni Bonanno, et al. Silicon planar technology for single-photon optical detectors. *IEEE Transactions on Electron Devices*, 50(4):918–925, 2003.

[84] P Finocchiaro, A Campisi, L Cosentino, A Pappalardo, F Musumeci, S Privitera, A Scordino, S Tudisco, G Fallica, D Sanfilippo, et al. A new generation of low-voltage single-photon micro-sensors with timing capability. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 567(1):83–88, 2006.

[85] Massimo Mazzillo, Giovanni Condorelli, Delfo Sanfilippo, Giuseppina Valvo, Beatrice Carbone, Giorgio Fallica, Sergio Billotta, Massimiliano Belluso, Giovanni Bonanno, Luigi Cosentino, et al. Silicon photomultiplier technology at stmicroelectronics. *IEEE Transactions on Nuclear Science*, 56(4):2434–2442, 2009.

[86] Gabriele Adamo, Diego Agrò, Salvatore Stivala, Antonino Parisi, Costantino Giaconia, Alessandro Busacca, Massimo C Mazzillo, Delfo Sanfilippo, and Pier Giorgio Fallica. Responsivity measurements of n-on-p and p-on-n silicon photomultipliers in the continuous wave regime. In *SPIE OPTO*, pages 86291A–86291A. International Society for Optics and Photonics, 2013.

[87] Gabriele Adamo, Diego Agrò, Salvatore Stivala, Antonino Parisi, Giuseppe Costantino Giaconia, Alessandro Busacca, Massimo Mazzillo, Delfo Sanfilippo, and Giorgio Fallica. Measurements of silicon photomultipliers responsivity in continuous wave regime. *IEEE transactions on electron devices*, 60(11):3718–3725, 2013.

[88] G Adamo, D Agrò, S Stivala, A Parisi, GC Giaconia, A Busacca, M Mazzillo, D Sanfilippo, and G Fallica. P-on-n and n-on-p silicon photomultipliers: responsivity comparison in the continuous wave regime. In *Photonics North 2013*, 2013.

[89] R Pagano, G Valvo, D Sanfilippo, S Libertino, D Corso, PG Fallica, and S Lombardo. Silicon photomultiplier device architecture with dark current improved to the ultimate physical limit. *Applied Physics Letters*, 102(18):183502, 2013.

[90] Massimo Mazzillo, Anatoly Ronzhin, Sergey Los, Salvatore Abbisso, Delfo Sanfilippo, Giusy Valvo, Beatrice Carbone, Angelo Piana, Giorgio

Fallica, Michael Albrow, et al. Electro-optical performances of p-on-n and n-on-p silicon photomultipliers. *IEEE Transactions on Electron Devices*, 59(12):3419–3425, 2012.

[91] Ramesh M Gulrajani. *Bioelectricity and biomagnetism*. J. Wiley, 1998.

[92] Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[93] Hans Hallez, Bart Vanrumste, Roberta Grech, Joseph Muscat, Wim De Clercq, Anneleen Vergult, Yves D'Asseler, Kenneth P Camilleri, Simon G Fabri, Sabine Van Huffel, et al. Review on solving the forward problem in eeg source analysis. *Journal of neuroengineering and rehabilitation*, 4(1):46, 2007.

[94] Tommaso De Marco. *Parallel modeling of the electric field distribution in the brain*. PhD thesis, alma, Maggio 2011.

[95] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006.

[96] Jaakko Malmivuo and Robert Plonsey. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA, 1995.

[97] Christoph M Michel, Göran Lantz, Laurent Spinelli, Rolando Grave de Peralta, Theodor Landis, and Margitta Seeck. 128-channel eeg source imaging in epilepsy: clinical yield and localization precision. *Journal of Clinical Neurophysiology*, 21(2):71–83, 2004.

[98] C Neuper, GR Müller, A Kübler, N Birbaumer, and G Pfurtscheller. Clinical application of an eeg-based brain–computer interface: a case study in a patient with severe motor impairment. *Clinical neurophysiology*, 114(3):399–409, 2003.

[99] Frank Masterpasqua and Kathryn N Healey. Neurofeedback in psychological practice. *Professional Psychology: Research and Practice*, 34(6):652, 2003.

[100] Jonathan E Walker and Gerald P Kozlowski. Neurofeedback treatment of epilepsy. *Child and adolescent psychiatric clinics of North America*, 14(1):163–176, 2005.

[101] David Vernon, Tobias Egner, Nick Cooper, Theresa Compton, Claire Neilands, Amna Sheri, and John Gruzelier. The effect of training distinct neurofeedback protocols on aspects of cognitive performance. *International journal of psychophysiology*, 47(1):75–85, 2003.

[102] John Gruzelier, Tobias Egner, and David Vernon. Validating the efficacy of neurofeedback for optimising performance. *Progress in brain research*, 159:421–431, 2006.

[103] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[104] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.

[105] Yann Renard, Fabien Lotte, Guillaume Gibert, Marco Congedo, Emmanuel Maby, Vincent Delannoy, Olivier Bertrand, and Anatole Lécuyer. Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence*, 19(1):35–53, 2010.

[106] Alois Schlogl and Clemens Brunner. Biosig: a free and open source software library for bci research. *Computer*, 41(10):44–50, 2008.

[107] Erwei Yin, Zongtan Zhou, Jun Jiang, Yang Yu, and Dewen Hu. A dynamically optimized ssvep brain–computer interface (bci) speller. *IEEE Trans. Biomed. Eng.*, 62(6):1447–1456, 2015.

[108] Carmen Vidaurre, A Schlogl, Rafael Cabeza, Reinhold Scherer, and Gert Pfurtscheller. A fully on-line adaptive bci. *IEEE Trans. Biomed. Eng.*, 53(6):1214–1219, 2006.

[109] Bart Vanrumste, Gert Van Hoey, Rik Van de Walle, RP D Michel, Ignace A Lemahieu, and Paul AJM Boon. Comparison of performance of spherical and realistic head models in dipole localization from noisy eeg. *Medical engineering & physics*, 24(6):403–418, 2002.

[110] Geertjan Huiskamp, Maurice Vroeijenstijn, René van Dijk, George Wieneke, and Alexander C van Huffelen. The need for correct realistic geometry in the inverse eeg problem. *IEEE Transactions on Biomedical Engineering*, 46(11):1281–1287, 1999.

[111] Nicolas Chauveau, Xavier Franceries, Bernard Doyon, Bernard Rigaud, Jean Pierre Morucci, and Pierre Celsis. Effects of skull thickness, anisotropy, and inhomogeneity on forward eeg/erp computations using a spherical three-dimensional resistor mesh model. *Human Brain Mapping*, 21(2):86–97, 2004.

[112] Roberto D Pascual-Marqui, Michaela Esslen, Kieko Kochi, Dietrich Lehmann, et al. Functional imaging with low-resolution brain electromagnetic tomography (loreta): a review. *Methods and findings in experimental and clinical pharmacology*, 24(Suppl C):91–95, 2002.

[113] Roberto D Pascual-Marqui. Discrete, 3d distributed, linear imaging methods of electric neuronal activity. part 1: exact, zero error localization. *arXiv preprint arXiv:0710.3341*, 2007.

[114] Roberto Domingo Pascual-Marqui et al. Standardized low-resolution brain electromagnetic tomography (sloreta): technical details. *Methods Find Exp Clin Pharmacol*, 24(Suppl D):5–12, 2002.

[115] Richard W Homan, John Herman, and Phillip Purdy. Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology*, 66(4):376–382, 1987.

[116] Krzysztof J Gorgolewski, Gael Varoquaux, Gabriel Rivera, Yannick Schwarz, Satrajit S Ghosh, Camille Maumet, Vanessa V Sochat, Thomas E Nichols, Russell A Poldrack, Jean-Baptiste Poline, et al. Neurovault. org: A web-based repository for collecting and sharing unthresholded statistical maps of the human brain. *Frontiers in neuroinformatics*, 9:8, 2015.

[117] Florian Ries. *Heterogeneous multicore systems for signal processing*. PhD thesis, alma, Aprile 2011.

[118] Vladimir Fonov, Alan C Evans, Kelly Botteron, C Robert Almli, Robert C McKinstry, D Louis Collins, Brain Development Cooperative Group, et al. Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1):313–327, 2011.

[119] Vladimir S Fonov, Alan C Evans, Robert C McKinstry, CR Almli, and DL Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47:S102, 2009.

[120] D Louis Collins, Alex P Zijdenbos, Wim FC Baaré, and Alan C Evans. Animal+ insect: improved cortical structure segmentation. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 210–223. Springer, 1999.

[121] Dirk-Jan Kroon. Smooth triangulated mesh, 2010.

[122] Kirill Stytsenko, Evaldas Jablonskis, and Cosima Prahm. *Evaluation of consumer EEG device Emotiv EPOC*. 2011.