IJCARS manuscript No.
(will be inserted by the editor)

Automatic Computation of Electrodes Trajectories for Deep Brain Stimulation

A Hybrid Symbolic and Numerical Approach

Caroline Essert · Claire Haegelen · Florent Lalys · Alexandre Abadie · Pierre Jannin

Received: date / Accepted: date

Abstract
Purpose The optimal electrode trajectory is needed to assist surgeons in planning Deep Brain Stimulation (DBS). A method for image-based trajectory planning was developed and tested.
Methods Rules governing the DBS surgical procedure were defined with geometric constraints. A formal geometric solver using multimodal brain images and a template built from 15 brain MRI scans were used to identify a space of possible solutions and select the optimal one. For validation, a retrospective study of 30 DBS electrode implantations from 18 patients was performed. A trajectory was computed in each case and compared with the trajectories of the electrodes that were actually implanted.
Results Computed trajectories had an average difference of 6.45 degrees compared with reference trajectories and achieved a better overall score based on satisfaction of geometric constraints. Trajectories were computed in 2 min for each case.
Conclusion A rule-based solver using pre-operative MR brain images can automatically compute relevant and accurate patient-specific DBS electrode trajectories.

Keywords Geometric constraint solving · Deep Brain Stimulation · Path planning · Decision Making · 3D models · Optimization

C. Essert
LSIIT / University of Strasbourg
Pôle API, Boulevard S. Brant - 67412 Illkirch, France
E-mail: essert@unistra.fr

C. Haegelen, F. Lalys, A. Abadie and P. Jannin
INRIA Centre Rennes - Bretagne Atlantique / INSERM
Campus de Beaulieu - 35000 Rennes, France

C. Haegelen
Department of Neurosurgery
Pontchaillou University Hospital - 35000 Rennes, France

1 Introduction

An increasing number of patients suffering from movement disorders such as Parkinson's disease or essential tremors are treated by Deep Brain Stimulation (DBS) [8,7,25]. This intervention, discovered in the early 1990's, consists in implanting an electrode in deep brain structures in order to stimulate a functional target with high-frequency electrical impulses, causing reliefs of the disease symptoms. This treatment has become very popular, as it is an adjustable and reversible alternative to the ablation of functional targets.

The procedure of DBS involves several phases [6]. Phase 1, setup: A stereotactic frame is set on the patient's head. Phase 2, pre-operative planning: The neurosurgeon chooses on pre-operative MR and CT images the best location for target stimulation, as well as the optimal entry point and direction (which we will often call *trajectory* or *position* in this paper), which will be used to reach the target with the electrode. In case of a bilateral implantation, both trajectories are planned during this phase. Phase 3, intervention and preparation of access: A hole is drilled in the skull at the previously defined entry point, in order to have an access toward the target. Phase 4, electrophysiological exploration: A test electrode is inserted along the planned trajectory, varying the depths of insertion with a micrometric precision. Sometimes, other test electrodes are inserted 2 mm beside the central electrode simultaneously or separately, to test nearby locations. The clinical response to stimulation is evaluated, in order to find the placement having the best benefits/side effects ratio. Phase 5, permanent electrode placement: When the best location is found, a permanent electrode replaces the test electrode in the brain. The permanent electrode is fixed on the skull. In case of a bilateral

implantation, Phases 3-5 are performed on the other side of the brain. Phase 6: The permanent electrode is connected to a pulse generator surgically implanted below the clavicle. This generator then needs to be programmed and adjusted.

This treatment is very efficient and brings a valuable quality of life to the patients, but it is also difficult to plan. The planning time varies depending on the experience of the physicians and the software tool that is used. This is a crucial step where the decision-making process looks for all available data (*e.g.*, medical images and clinical scores), information (*e.g.*, clinical data) and knowledge (*e.g.*, digital atlases, books). In some centers, this process involves several clinicians (e.g., neurosurgeons, neurologists, neuropsychologists) and can take up to 1h 30 minutes to find a good planning, half of the time to locate the target and the other half to choose the direction of the electrode. The tedious and long planning phase mainly relies on the study of the pre-operative patient images (such as MRI and CT) and requires many of manual operations. Even if the navigation software provides assistance for the different operations, the registration of the images acquired with different modalities still requires many of manual adjustments, the target needs to be manually defined, and the standard placement proposed by the software is rarely appropriate, forcing the neurosurgeon to adjust it using a trial and error approach.

Some studies underlined the importance of the trajectory and electrode location on possible clinical and neuropsychological side effects following surgery [40], and on the complications such as hemorrhages which is the most common [24], underlining the interest of planning safe trajectories. The objective of the work presented in this paper is to provide the neurosurgeon with a planning tool able to assist him/her in the decision-making process, by finding a safe and optimal linear trajectory for the implantation of a DBS electrode from given patient's medical images.

2 Related and previous works

In DBS surgical planning, various approaches with different purposes have been already proposed: 1) to help in finding automatically the target [15,18], 2) to simulate the electric field of the electrode stimulation and its propagation in the surrounding tissues [12], and 3) more recently to automatically compute safer trajectories of the surgical electrodes. In this study we focus on this last topic, for which we will detail a few interesting works below. Some authors proposed interesting attempts of automatic targeting methods, for various kinds of surgeries. However, they also have some drawbacks that we would like to overcome.

The first studies on automatic computations of optimal linear trajectories for surgery planning were described for thermal ablations of hepatic tumors [2,1,36, 4,3,32,31]. Actually, from the computer science point of view, planning a placement of thermal probes in the liver has many of similarities with planning a placement of electrodes in the brain. The action of computing an optimal trajectory is the same, only the conditions of the "optimality" and the modeling of the tools and effects change. Altrogge et al. [2] studied hepatic RadioFrequency Ablation (RFA) needle placement: authors tried to fit a volume of effect to a target by using numerical optimization. They modeled the distribution of the temperature by taking into account some surrounding vessels, however they did not take into account the presence of all other surrounding organs and the possible intersection of the tool with vital structures.

Robotically assisted interventions also need an accurate planning to be safe and successful. In 2003, Adhami et al. [1] proposed an interesting method using a translation of the qualitative and quantitative description of an intervention in a mathematical structure that could be formally defined, manipulated, and optimized. However, the authors confessed a long computation time. The same year, Essert-Villard et al. also set the foundations of an approach based on the formalization of surgical constraints that could be solved to propose optimal trajectories for RFA [36]. The present study is inspired from further works of this team [4,3] showing how to combine several constraints. This approach has the advantages of taking into account the whole anatomy of the patient, and to be fast enough for being used by the surgeons. The computation times have recently been improved even more with the works in collaboration with Seitel et al. [32]. More recently, Schumann et al. [31] presented a similar approach based on segmentation masks rather than polygonal representations of the anatomical structures. They also used a weighted combination of constraints. The authors reported a total computation time of 6 seconds for determining suitable insertion trajectories on a standard computer. However, an evaluation showing the clinical suitability of such a trajectory planning system was not performed.

In image guided neurosurgery, a number of authors proposed planning methods. But most of the early planning tools required many manual interventions [10,17, 23,30,35]. More recently, Shamir et al. [33] proposed an approach for planning trajectories in image-guided keyhole neurosurgery. Like Schumann et al. [31], they

worked directly on voxels of the brain images, and assigned risk values to them. In this approach, only the risk criterion was taken into account. Brunenberg et al. presented in [11] one of the works most related to ours, as they focused on DBS electrodes placement. However, authors restricted the search to a limited set of possible entry points, avoiding possibly good trajectories to be discovered. We prefer to let the software decide by itself possible entry points, in order to study all possible solutions as long as they satisfy the rules of the intervention. We would also like to keep in mind an idea of genericity, with a constraint-based solver usable on any kind of linear trajectory planning, and easy to adapt to the description of new surgical rules with new constraints.

This paper presents a method for automatic computation of DBS electrode trajectories, based on the resolution of geometric constraints inspired from [4,3], and extending the preliminary results presented in [16]. Our approach is based on two types of data: pre-operative patient-specific images, and the rules specific to the intervention. We first detail in Section 3.1 the approach we used to determine the surgical rules followed by surgeons for DBS planning, and detail how we treated them, either formalizing them into geometric constraints or simply adjusting some parameters. Then in Section 3.2, we explain how we obtained the patient-specific data from images. In Section 3.3, we expose our approach and the formal solver we developed to solve geometric constraints with image data. Section 3.4 exposes the formalized constraints we defined for DBS planning. Then we present the experiments including 3 different targets that are usually stimulated: the Sub-Thalamic Nucleus (STN), the Globus Pallidus internus (GPi) [34] and the caudal part of the Ventro-Lateral nuclei of the thalamus (VLc) (sometimes also called the Ventral Intermediate nucleus of the thalamus, or VIM) [20], and the retrospective study on 30 implantations from 18 patients. We compare the results of our automatic trajectory computation with the reference trajectory segmented from post operative CT images.

3 Materials and methods

3.1 Analysis of rules governing DBS planning

When designing a method for automated computer-assisted surgical planning, it is crucial to follow knowledge elicitation approaches for understanding the surgical decision-making process. For the computation of the DBS electrode trajectory, we aimed at identifying implicit and explicit rules used by neurosurgeons when selecting a best possible trajectory. These rules were intended to be translated in a second step into numerical constraints used by the solver to calculate the trajectory.

From an initial study of the literature about DBS planning, we designed a questionnaire that allowed us to gather the surgeon's knowledge on which the reasoning will be based. During 3 interviews, we submitted the questionnaire individually to two neurosurgeons who participated in our study: one of them with an experience of approximately 100 DBS implantations, and the other with about 200 implantations.

The questionnaire was intended to converge towards a set of identified rules constituting the main concerns used for planning in clinical routine. It contained a list of proposed rules and asked if they were relevant or not. The neurosurgeons also had the possibility to add extra rules that were not mentioned. Additionally, we asked them to give weights to the rules, according to the importance they see for each of them. As expected, some of the proposed rules were set aside by the surgeons while others were added, and they underlined the importance of some of the rules while lowering the importance of others. Some of the rules were mentioned as being mandatory to satisfy, whereas others could be seen as preferences. For this last type of rules, we jointly defined a weighting with both surgeons, which represented at best the impression they had of their importance. Then, an average of the proposed weightings has been computed for each rule (which will be referred to as k_i in the rest of the paper), and used for our experiments.

After the questionnaire was filled, we debriefed with these neurosurgeons to sum up the rules. The ones we chose and their formalization or parameterization are presented below.

(1) **Place the electrode into the target**. The first obvious rule that was identified was that the electrode tip must be in the target. This rule strongly restricts the space of possible positions. Our solver expects the definition of a target, and has been natively designed to consider only the 3D positions of the tool with the tip inside the target (see Fig.1). This rule must be satisfied.

(2) **Position of the insertion point**. The insertion point on the head has to be located in the upper surface of the skin of the head, as the surgeon will never implant the electrode through the lower parts of the head. This is not only due to accessibility reasons, but also to aesthetics reasons. We provide our solver with an initial insertion zone corresponding to the scalp of the head (see Fig.2): this mesh will often be referred to as *skin mesh* or *initial inser-*

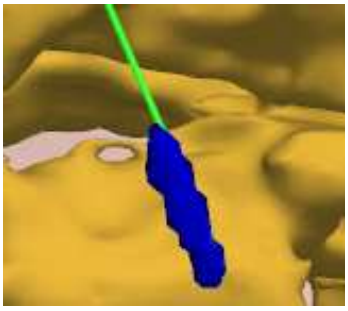


Fig. 1 The tip of the electrode has to be in the target. Here the target is the blue shape and the computed trajectory represented as a green line. As explained in section 3.2, the target we used is not the actual anatomic structure, but a segmentation of the contacts of the reference target. The yellow shape in the back is a part of the ventricles mesh.

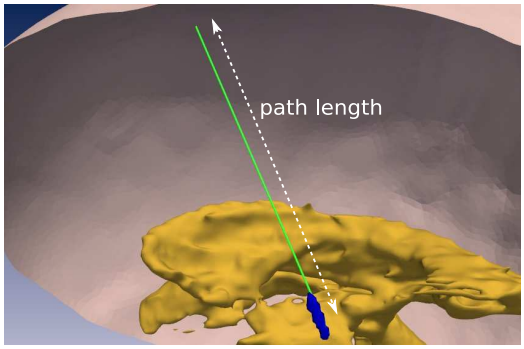


Fig. 3 The path length is the distance between the candidate entry point and the tip of the electrode, here represented by its trajectory (green line). The tip is located inside the mesh of the target.

Fig. 2 Initial insertion zone: upper surface of the skull's skin mesh. This mesh is used as a basis to determine the possible solutions. As explained in section 3.2, this mesh is built from a mono-subject anatomical template made from 15 3T MR acquisitions, and adapted to each specific patient's preoperative MRI (background) with an affine registration.

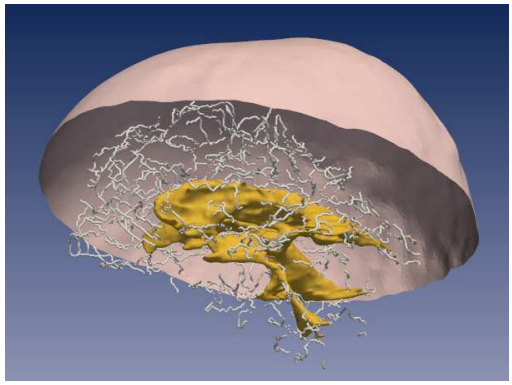


Fig. 4 The trajectory has to avoid ventricles (here in yellow) and the cortical sulci. Here the cortical sulci are represented by a set of meshes located at their external traces (white intricate mesh).

tion zone in the rest of the paper. This rule must be satisfied.

(3) **Path length restriction**. This rule concerns the maximal length of the path (see Fig.3), which restricts the field of research. According to neurosurgeons instructions, we formalized this rule by assuming that the path has to be shorter than 90mm. In practice, this rule mainly allows us to eliminate implicitly the half of the head which is opposite to the side of the target as possible locations of insertion. This rule must be satisfied.

(4) **Avoid risky structures**. We need to find an electrode placement that avoids crossing vital or risky structures. For DBS, the identified "obstacle" structures include the ventricles and the vessels. Vessels are numerous in the brain, and generally located in cortical sulci. Unfortunately they are often invisible when images are acquired without contrast agent or angiography. So the neurosurgeons usually rely on the anatomical location of the sulci and avoid trajectories passing through them. We chose to add the avoidance of cortical sulci as a proper rule (see ventricles and sulci on Fig.4). We formalized this by assuming that the insertion point in the skull of the patient has to be visible from the target, without any occlusion by one of the cerebral structures considered as obstacles. This rule must be satisfied.

(5) **Minimize the path length**. Even if we are sure that the path is shorter than the maximum length defined by rule #3, minimizing the length of the path as much as possible reduces the risks of an imprecision when performing the implantation. We formalized this rule by assuming that the proportion between the length of the path and the shortest distance between target and the skin mesh (defined in rule #2) has to be minimal. This rule is a preference, so it can't be merged with rule #3 because their natures are different. Let us note however that this rule is not always mentioned by surgeons, so it has been given a low weighting.

(6) **Maximize the distance between electrode and risky structures**. Even if we are sure that the electrode will not cross any risky structure thanks to rule #4, it is less risky if the trajectory passes as far as possible from those structures. We formalized this rule by assuming that the distance between the electrode and the structures designated as risky has to be maximized. The distance between the electrode and a structure is the distance between the candidate trajectory and the closest point of the mesh representing the structure. This rule is a preference.

(7) **Optimize the orientation of the electrode depending on target shape**. Another rule expressed by surgeons was to have the trajectory axis as close as possible to the main axis of the target, when the target is oblong (see Fig.1). This way, they can try several possible depths by simply pulling the electrode in or out to cover almost all of the target without needing another insertion at a different location. We formalized this rule by assuming that the angle between the axis of the electrode and the main axis of the target has to be minimized. This rule is a preference.

(8) **Placing the tip as close as possible to the center of the target**. Once again, even if we are sure that the tip of the electrode will be placed inside the target thanks to rule #1, we would like to have the tip as close as possible to the center of the target. We formalized this rule by assuming that the distance between the tip and the center should be minimized. This rule is a preference.

3.2 Input data

Our algorithm uses a set of spatially defined objects, such as: an initial insertion zone, the target, and all structures that come into play in the constraints as obstacles or risky structures. They need to share the same reference coordinate system so we chose to use the patient's pre-operative MRI coordinate system. In this section, we describe how we obtained the data.

The images were acquired with the same image acquisition protocols (see Fig.5): one pre-operative 3T T1-weighted MR (1 mm x 1 mm x 1 mm, Philips Medical Systems), one pre- and one post-operative CT scans (0.44 mm x 0.44 mm x 0.6 mm for post-operative acquisitions and 0.5 mm x 0.5 mm x 0.6 mm for pre-operative acquisitions, GE Healthcare VCT 64). The MRI and the pre-operative CT were acquired just before the intervention, and the post-operative CT was acquired a few days after, however we didn't use the pre-operative CT in our study. MRI and post-operative

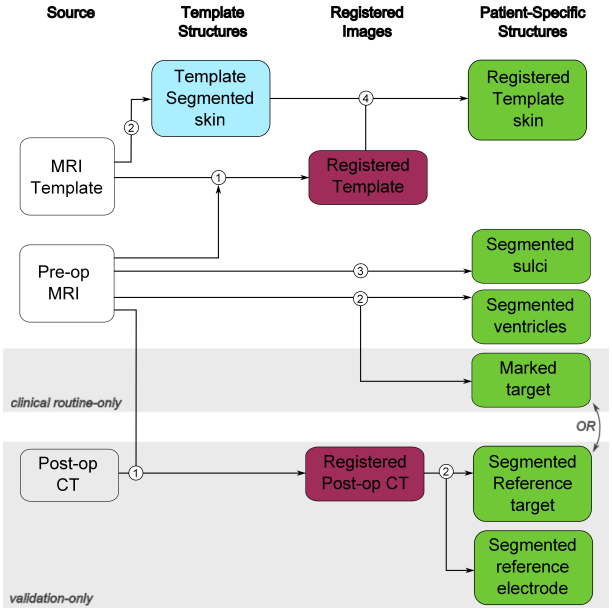


Fig. 5 Graph of input data. Three images sources (left, in white) and four different treatments (①, ②, ③, ④) were used to obtain the reconstructed anatomical structures (in medium green). Intermediate steps produce registered images (in dark pink) and template structures (in light blue). In the lower part of the graph, the 2 parts with the gray backgrounds correspond to the treatments and structures required respectively if the system is used in clinical routine, or for the validation process. For the experiments described in this paper, we used the *"validation-only"* part.

CT images were first subjected to a series of fully automatic processes (numbered ① on Fig.5). They were denoised with a Non-local means algorithm [14]. A bias correction algorithm based on intensity values [28] was applied on MR images. Then, CT images were rigidly registered (cost function : Mutual Information) to pre-operative MR images using Newuoa optimization [37].

From these co-registered images, we extracted anatomical structures either with semi-automatic segmentation using MITK (Medical Imaging Interaction ToolKit [27] developed by the DKFZ (German Cancer Research Center in Heidelberg)): operations numbered ② on Fig.5, or thanks to an automatic process: numbered ③ on the figure. The brain was segmented from pre-operative MRI thanks to an extraction approach [29] from BrainVISA image processing software platform [13]. From this initial segmentation, the cortical sulci were automatically segmented using an algorithm based on curvature information [22].

In this study, we focused on the computation of the trajectory to reach a target, and not on the delineation of the target. Our goal was to compare our proposition with reference trajectories. To this end, we needed to use the exact same target position as the one actually performed during the intervention so that the compar-

isons would not be disturbed by possible errors of locations of the target. For this purpose we segmented the contacts of the electrode (that appear as bright in the post-operative CT) as a small oblong 3D mesh and used this mesh as the target (as seen on Fig.1, blue shape), instead of segmenting the actual target. This is represented by the lower gray *"validation-only"* part of Fig.5, with the *"segmented reference target"*. In clinical routine, the neurosurgeon simply marks the location of the target in the planning software, sometimes assisted by an atlas – this is illustrated on Fig.5 by the *"clinical routine-only"* gray part, with the so-called *"marked target"*. Additionally, we segmented the reference electrode from post-operative CT for validation purposes.

We then computed 3D meshes from the segmented volumes by using the Marching Cubes approach [26].

We applied a particular process to one of the anatomical structures: we manually segmented the initial insertion zone (skin) once for all on a mono-subject anatomical template made from 15 3T MR acquisitions at high-resolution (0.25 mm x 0.25 mm x 0.25 mm) [21]. This zone does not include the entire surface of the skin surrounding the skull but it covers the usual location of the hair to take into account aesthetics. An additional process was applied to the 3D mesh of this surface: we homogenized the sizes and angles of the triangles of the mesh thanks to an isotropic remeshing algorithm described in [9]. As we will explain later, this mesh is used as a base for the 3D mesh of the solution space and subdivided on the borders, so an homogeneous mesh is advisable. Then thanks to an affine registration [19], this surface mesh was automatically adapted to each studied patient's MRI (operation numbered ④, see also Fig.2), sparing us a systematic manual segmentation.

As all 3D structures were either obtained from pre-operative MRI, or obtained from a CT that was registered to pre-operative MRI, or obtained from an MRI template and then registered to pre-operative MRI, anatomical structures and objects were defined in the same common space.

3.3 Global strategy

Our strategy consists in formalizing the rules described in Section 3.1 into geometric constraints in order to solve them with a constraints solver. Among those constraints, some are boolean (#1, #2, #3, #4): they are rules that must be satisfied. Others are numerical (#5, #6, #7, #8), and were mentioned earlier as preferences. The first ones are called *strict constraints*, and define the space of all possible solutions. The second ones are *soft constraints* that need to be optimized at best, according to weighting factors k_i defined by the surgeon that represent the importance of each particular soft constraint in relation to the others. Among the space of possible solutions, the optimal path will be the one that satisfies the soft constraints at best. Let us note that constraints #1 and #2 are not solved but included as input image data in our solver (as target and skin meshes). The two other strict constraints and four soft constraints are translated into geometric constraints as explained in Section 3.4.

When programming our solver, we gave a great importance to the genericity in our approach, our goal being to dispose of a generic solver able to be used for any surgical intervention involving a path planning for a rectilinear tool. Therefore, our solver takes as input data not only the images and segmented cerebral structures, but also the rules of the intervention written in a specifically defined meta-language, and written in a separate XML (Extensible Markup Language) file which is loaded when the software is launched. If we want to add an extra constraint, we only have to write it in this file.

A solution is constituted by a position in the 3D space for the electrode. It can be represented indifferently either by a point (*i.e.* the tip of the electrode) and a direction, or by two points. We chose to use the second alternative that was more intuitive, by using the tip and the insertion point on the skin. For the insertion point, we start with an *initial solution space* constituted by the mesh of the *initial insertion zone* representing the upper part of the scalp. For the tip point, we start with the whole target.

The solving process is performed in two steps. The first phase consists in reducing the initial insertion zone by eliminating the triangles of the polygonal mesh that do not satisfy the totality of the strict constraints. Triangles of the mesh that satisfy only partially the strict constraints are subdivided into 4 subtriangles thanks to a simple quadrisection operator, in an iterative process, such as in [5]. The second phase consists in a numerical optimization of the soft constraints. Each soft constraint corresponds to a cost function to minimize. In order to take into account all constraints with a weighting factor defined by the expert, we chose to combine them into an aggregative cost function. After an initialization of the process, consisting in a rough evaluation of the values at some insertion points homogeneously spread over the zone of possible insertion points, we compute some connected components around the best candidate points, and we start an optimization using Nelder-Mead optimization method from the best candidate. This way we avoided to fall into local minima, as described in a previous paper [3].

3.4 Implementation of geometric constraints

Using the meta-language, the rules (or their corresponding cost functions) are translated as geometric constraints represented as terms, combining operators and known data, according to a geometric universe. The geometric universe we defined for our constants and unknowns includes the usual types (*e.g.* integers, real numbers, booleans) and composed types such as *point*, *tool*, *shape*, or *solution*. We also defined operators: usual operators such as plus, minus, multiply, and, or, as well as complex operators as for instance *distMin*, *distToolOrgan*, *angle*, *visible*. In order to add an extra constraint in the XML file, the necessary operators must have already been defined. The terms can be seen as trees, which are solved using a depth-first approach.

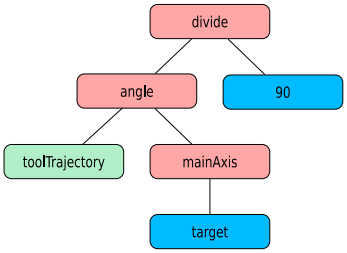
3.4.1 Strict constraints

The rules corresponding to strict constraints are either natively satisfied if they are implicit rules (Rules #1 and #2 which are integrated under the form of input image data *target* and *skin* meshes), or transcribed as boolean terms, *i.e.* using boolean operators (Rules #3 and #4).

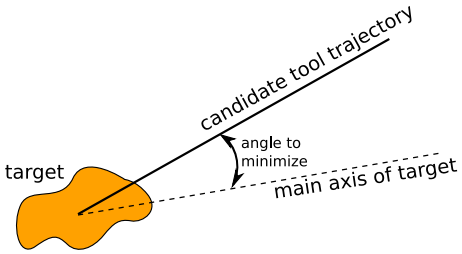
3.4.2 Soft constraints

The rules corresponding to soft constraints are translated into cost functions to minimize, then written under the form of terms using the available operators.

As an example of soft constraint, let us analyze Rule #7. This rule aims at optimizing the orientation of the electrode according to the shape of the target (as shown on Fig.6(b)). It is translated into a soft geometric constraint expressing that the angle between the trajectory of the electrode and the main axis of the target has to be minimal. It is computed by the minimization of a numerical cost function $f_{ori} : \mathbb{R}^5 \rightarrow [0, 1]$. This function applies the geometric operators *angle* (computing the angle between two directions) and *mainAxis* (computing the main axis of a 3D shape thanks to a PCA (Principal Component Analysis) of its vertices) to the unknown X and the given mesh *target*. The cost function is chosen in a way that the resulting values are between 0 and 1, in order to obtain an order of magnitude comparable to the cost functions of the other rules before combining them. Without this normalization, a rough combination of these functions would be meaningless. So we transform the formalization by saying that the ratio between the angle expressed in degrees and 90 has to be minimal. This way, f_{ori} tends to 0 if the angle is close to 0, and to 1 if the angle is close to 90 degrees. Function f_{ori} is then defined by cost function (1). In this equation, X represents the degrees of freedom in $\mathbb{R}^5$ of the trajectory of the tool.



(a) Tree representing Rule #7



(b) Scheme representing Rule #7

Fig. 6 Different representations of Rule #7. (a) The tree shows how the term formalizing the rule is built, using operators (in pink), constants (in blue), and the variable (in green). (b) The scheme illustrates in a simple way the aim of rule #7: minimization of the angle between the candidate trajectory and the main axis of a target which has an oblong shape. Figure reprinted from [16]

$$f_{ori}(X) = \frac{angle(X, mainAxis(target))}{90} \quad (1)$$

To express this function as a constraint understandable by our solver, we use our meta-language and write it as a term. This term uses existing operators defined in the solver (*divide*, *angle*, *mainAxis*), constant data (*target* shape coming from the images, integer *90*), and the variable *toolTrajectory* which will be filled with a candidate value. The final constraint in XML syntax is shown in Table 1 line (3). The corresponding tree is illustrated on Fig.6(a): operators are in red, given constant data are in blue, and the variable is in green. In the solver, we use this tree structure to represent the constraints. If a data or variable node is used in more than one constraint, it exists only once and doesn't have to be re-evaluated several times.

3.4.3 Aggregative soft constraint

The constraints we detailed in Section 3.1 were all written in XML syntax with the same approach, using our operators and data, and are shown on Table 1. The last soft constraint ("*sc_final*", line (4)) is the aggregative

Table 1 Strict and soft constraints in XML formalization

Strict constraints

```
<strict_constraint name="path length restriction" impact="insertionZone">
lower( distMin ( center(target), insertionPoint ), maxPathLength )
</strict_constraint>

<strict_constraint name="avoiding risky structures restriction" impact="insertionZone">
visible( insertionZone, targetVolume )
</strict_constraint>
```

Soft constraints

```
<soft_constraint name="minimize depth" label="sc_depth">
divide( minus( distMin(toolTip, insertionPoint), distTargetSkull ), minus(maxPathLength, distTargetSkull) )
</soft_constraint>

<soft_constraint name="minimize risk" label="sc_risk">
max( divide( minus( 10.0 , distMin (sulci, toolTrajectory) ), 10.0 ), 0 )
</soft_constraint>

<soft_constraint name="optimized orientation" label="sc_ori">
divide( angle ( toolTrajectory, mainAxis (target) ), 90.0 )
</soft_constraint>
```

(3)

```
<soft_constraint name="centering" label="sc_center">
divide( distmin ( toolTip, center (target) ), 10.0 )
</soft_constraint>

<soft_constraint name="final constraint" label="sc_final">
plus( mult(weight_sc_depth, sc_depth), mult(weight_sc_risk, sc_risk),
      mult(weight_sc_ori, sc_ori), mult(weight_sc_center, sc_center) )
</soft_constraint>
```

(4)

constraint, that combines the four previously defined soft constraints with the chosen weighting factors k_i. It corresponds to the XML transcription of the aggregative cost function f, written line (2), which combines the cost functions f_{depth}, f_{center}, f_{risk}, f_{ori}, associated to the four soft constraints #5 (depth), #6 (risk), #7 (orientation), and #8 (centering) respectively.

$$f(X) = k_d.f_{depth}(X) + k_r.f_{risk}(X) + k_o.f_{ori}(X) + k_c.f_{center}(X) \quad (2)$$

3.5 The software

To apply our method, we developed our own geometric constraints solver. It is developed in C++ and based on MITK software system [39], and using ITK and VTK libraries. It includes a parser to read the XML constraint file and build the constraints tree. It also includes a set of operators on 3D meshes such as subdivision of triangles by quadrisection, and computations of the volume, or the distance to a line or a point.

To compute the non-intersection between the trajectory and the 3D meshes of the organs to preserve, we introduced a specific operator called "*visible*". Even with spatial partition algorithm accelerating the computations, a naive ray/mesh intersection computation would still have been too time-consuming. We chose to take advantage of the GPU (Graphics Processing Unit) by performing renderings from the target's point of view in the six possible directions covering the whole 3D space (right / left / front / back / up / down). This is illustrated with a 2D projection on Fig.7 showing only four of the views (right / left / front / back). On this example, only seven beams reach the initial insertion zone without hitting any obstacle (such as ventricles or cortical sulci). View 4 is totally occluded by the ventricles. We compute occlusions of the initial insertion zone thanks to the OpenGL extension GL_ARB_occlusion_query. Each triangle of the initial insertion zone which is not rendered because it is occluded by an organ is eliminated from the solution space. Triangles partly visible can be subdivided up to a fixed threshold. This algorithm has recently been improved in speed thanks to works with DKFZ Heidelberg (MITK) [32], making the occlusion computation quite instantaneous (below 1 s.).

Our software finally includes a graphical interface, which allows to visualize the results and modify interactively some of the parameters. The results are shown under the form of a global color map representing the values of cost function f at each point of the insertion zone. The color map pictures the insertion points having the best results for f in green, those giving the worst

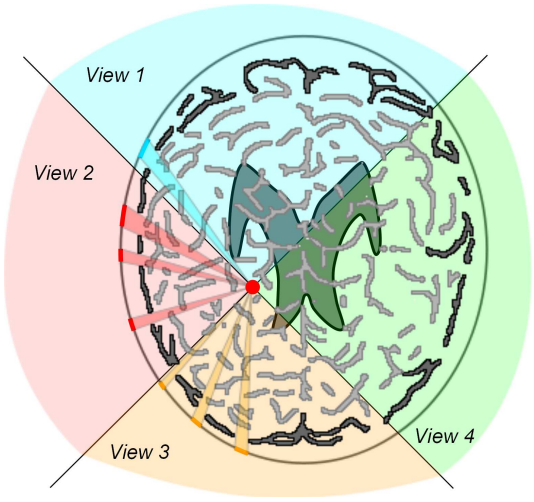


Fig. 7 Scheme in 2D illustrating the six 3D renderings used to compute parts of the initial insertion zone not occluded by any obstacle mesh (ventricles, sulci). Here only 4 views from the target (red point) are represented: front/back/right/left. Up/down views are not shown.

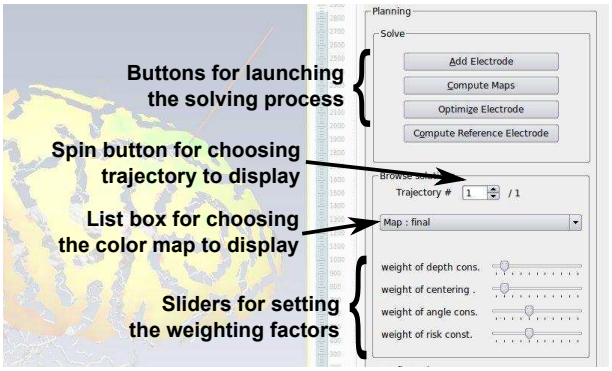


Fig. 8 Snapshot of our software, showing the interface allowing to modify the settings, to change the color map to display, and to browse the proposed optimal trajectories if several are found

results in red, and a range of progressive intermediate colors for the other points. Among the zones of best results, the optimal trajectory (*i.e.* passing through the point of minimal f) is shown as a red line. Let us note that in some cases there can be several trajectories considered as optimal, each one in a different "valley". This can be configured by adjusting a parameter, choosing either an ϵ defining a range of acceptable values around the best value of f, or a number of first best trajectories to display. In that case, the user can browse the proposed alternative trajectories by simply clicking on a spin button.

If the user wants to see the results provided by another choice of weighting factors, it is possible to modify them by using the corresponding sliders (as illustrated on Fig.8). The software recalculates the global color map and the optimal trajectory and updates the view instantaneously. Indeed, the 4 color maps of the individual soft constraints (representing respectively the values of $f_{depth}, f_{center}, f_{risk}$, and f_{ori}) are computed once for all and don't change if the weighting factors are modified. When the weighting factors change, only the simple aggregative cost function needs to be recomputed and this is instantaneous. The user can also consult the four color maps corresponding to the individual soft constraints, by selecting which map to visualize in a list.

3.6 Validation method

We performed a retrospective validation study of our method. For each case, we compared our solution with the actual position of the electrode that has been implanted in the patient, used as the reference trajectory. We worked on the assumption that the actual electrode was placed at best by the surgeons to fit simultaneously all the criteria, so our objective was to check if the solution we found automatically was close to this reference, or even a better solution regarding the criteria. That is why we measured the proximity of our computed trajectory with the reference trajectory, as well as the scores of both trajectories.

For each case, we computed the angle between our proposed optimal electrode trajectory and the reference trajectory. Let us recall that the reference electrode was segmented from post-operative CT, as explained in section 3.2 and illustrated on Fig.5 in the "validation-only" part. From this segmented electrode, we extracted its main direction by performing a PCA, to obtain a line we called "reference trajectory". For both trajectories (computed and reference) we also computed the minimal distance to the sulci, and compared the results. Finally, we reported the scores of both trajectories for all of the individual soft constraints and the aggregative soft constraint, *i.e.* the result of their respective cost functions, in order to quantify their quality regarding the rules defined by the neurosurgeons.

The experimentations were performed on images from the Department of Neurosurgery at Pontchaillou University Hospital in Rennes, France. We decided to retrospectively validate our software with DBS interventions targeting different anatomical structures in the brain: the globus pallidus interna (GPi), the subthalamic nucleus (STN), and the caudal part of the ventro-lateral nuclei of the thalamus (VLc). The two first ones are the most common sites for the placement of the lead in the case of dystonia and Parkinson's disease, and the 3rd one is more often used for non-Parkinsonian essential tremor. These targets are between 5 and 10 mm wide.

Table 2 Comparison between the trajectory produced by our planning (with weightings of $k_d = 1/k_c = 1/k_r = 4/k_o = 4$), and the reference trajectory

Type of target	Case #	Reference traj. T_{ref}		Computed traj. T_{plan}		Gain		Angle α between traj. (deg.)	Execution times		
		Score $[0-1]$	Dist. to sulci (mm.)	Score $[0-1]$	Dist. to sulci (mm.)	Score (%)	Dist. to sulci (mm.)		Solution space (s.)	Color maps (s.)	Optimization (s.)
GPi	1	0.398	2.052	0.233	6.573	41.46%	4.520	19.22	6.53	124.57	15.14
	2	0.200	6.222	0.114	8.206	43%	1.985	5.82	6.69	122.92	18.51
	3	0.230	6.042	0.221	6.299	3.91%	0.257	0.46	6.02	109.62	19.21
	4	1.000	0.054	0.166	9.424	83.4%	9.370	24.17	6.40	149.89	14.22
	5	0.298	4.396	0.172	7.491	42.28%	3.094	7.15	6.61	139.83	16.9
	6	0.430	1.263	0.230	8.103	46.51%	6.841	23.36	6.11	126.81	54.99
	7	0.400	5.714	0.262	11.677	34.5%	5.964	14.97	6.26	101.35	63.24
	8	1.000	0.555	0.177	8.918	82.3%	8.362	16.72	6.16	121.09	17.89
	9	1.000	0.648	0.174	7.606	82.6%	6.958	10.15	6.49	142.36	19.02
	10	0.343	3.157	0.121	9.178	64.72%	6.021	6.38	6.20	149.91	13.93
	11	0.360	1.480	0.200	5.961	44.44%	4.481	5.78	6.54	120.16	11.83
	12	0.345	3.228	0.257	6.795	25.51%	3.568	23.98	6.69	104.52	14.17
	13	0.228	6.140	0.178	6.809	21.93%	0.668	4.40	6.56	108.88	13.22
	14	0.309	3.892	0.199	6.446	35.6%	2.554	3.20	6.60	122.47	10.78
STN	15	0.264	5.746	0.121	8.133	54.17%	2.387	10.02	6.89	87.19	14.63
	16	0.350	2.850	0.239	5.379	31.71%	2.529	7.54	9.36	112.65	89.47
	17	1.000	0.750	0.165	8.301	83.5%	7.551	20.60	5.29	100.56	22.47
	18	0.361	2.790	0.209	7.107	42.11%	4.316	13.53	6.95	92.63	14.32
	19	0.329	3.490	0.262	5.753	20.36%	2.263	8.66	9.47	49.99	16.66
	20	0.401	1.740	0.219	4.900	45.39%	3.160	18.17	10.22	48.26	13.97
	21	0.397	2.425	0.305	4.768	21.19%	2.343	3.93	7.90	62.66	115.21
	22	0.338	2.748	0.244	5.365	27.81%	2.617	3.34	7.50	79.99	17.98
	23	0.362	1.946	0.270	4.858	25.41%	2.912	9.89	3.44	68.03	27.81
	24	0.408	1.184	0.195	7.275	52.21%	6.091	12.53	6.05	86.09	14.32
	25	0.354	2.239	0.135	7.958	61.86%	5.718	4.99	6.50	100.71	13.87
VLc	26	1.000	0.150	0.189	6.426	81.1%	6.096	11.76	5.80	87.98	16.23
	27	0.400	1.715	0.299	5.186	25.25%	3.471	9.18	7.22	80.89	125.90
	28	1.000	1.409	0.262	7.356	73.8%	5.947	20.50	6.98	92.51	19.28
	29	0.155	7.429	0.033	10.001	78.71%	2.572	5.15	7.10	88.52	14.81
	30	0.393	1.366	0.181	6.996	53.94%	5.630	7.78	1.68	112.06	17.58
AVERAGES		0.468	2.827	0.201	7.169	47.69%	4.342	11.11	6.61	103.17	28.58

Different types of electrodes were used, depending on the target: Medtronic 3389 when targeting the STN, and Medtronic 3387 for GPi and VLc. The contacts are more spaced on electrode Medtronic 3387, however this has no incidence on the planning. Our method can be used with any of these electrodes or others, as the important issue is the path towards the target that we model as a line.

Our retrospective study was performed on 30 cases, constituted by 18 patients, some of which had a bilateral electrode implantation. We treated each side as a separate case for our study. Among those patients, there were 8 GPi targetings (14 electrodes), 6 STN (11 electrodes), and 4 VLc (5 electrodes). For each case, we performed the registrations, segmentations, and reconstructions described in Section 3.2 on preoperative and postoperative images, and we obtained the necessary 3D meshes of the structures and the targets. Then we launched the automatic planning application, using the constraints defined in Section 3.1, and for a start we chose weighting factors of 1 for k_d (soft constraint #5), 1 for k_c (soft constraint #8), 4 for k_r (#6), and 4 for k_o (#7) to compute f. The weighting factors were defined in collaboration with the neurosurgeons.

4 Results

The first step of our method computes a solution space, i.e. the surface where the insertion point can be located according to the soft constraints. An example of solution space can be seen on Fig.9 representing patient case #3. Then, the second step computes colors maps on the solution space, to represent the quality of the different possible locations in relation to the soft constraints. Snapshots of color maps for patient case #3 are shown on Fig.11, where red parts represent the worst insertion locations, green parts represent the best insertion locations, with a range of intermediate colors. Figs.11(a), 11(b) and 11(c) show respectively the individual color maps for constraints #5 (length of path)

Table 3 Comparison between trajectories computed with different weighting factors: $k_d = 1/k_c = 1/k_r = 4/k_o = 4$ for $T_{plan}1$, and $k_d = 1/k_c = 1/k_r = 4/k_o = 10$ for $T_{plan}2$

Type of target	Case #	Computed trajectory $T_{plan}1$ 1/1/4/4			Computed trajectory $T_{plan}2$ (forced orient.) 1/1/10/4			Differences $T_{plan}1$ / $T_{plan}2$ (gain/loss)	
		Score $[0-1]$	Dist. to sulci (mm.)	Angle α_1 (degrees)	Score $[0-1]$	Dist. to sulci (mm.)	Angle α_2 (degrees)	Angle (%)	Dist. to sulci (mm.)
GPi	1	0.233	6.573	19.22	0.192	3.802	7.58	−60.53%	−2.771
	2	0.114	8.206	5.82	0.082	8.204	5.61	−3.67%	−0.003
	3	0.221	6.299	0.46	0.163	6.297	0.43	−7.05%	−0.002
	4	0.166	9.424	24.17	0.167	8.922	21.24	−12.12%	−0.501
	5	0.172	7.491	7.15	0.123	7.496	6.99	−2.33%	+0.006
	6	0.230	8.103	23.36	0.205	4.321	2.93	−87.47%	−3.782
	7	0.262	11.677	14.97	0.205	4.335	3.71	−75.24%	−7.342
	8	0.177	8.918	16.72	0.172	5.820	13.90	−16.82%	−3.098
	9	0.174	7.606	10.15	0.149	7.515	7.70	−24.09%	−0.091
	10	0.121	9.178	6.38	0.122	9.144	6.10	−4.38%	−0.034
	11	0.200	5.961	5.78	0.148	5.959	5.72	−1.01%	−0.001
	12	0.257	6.795	23.98	0.179	4.467	3.76	−84.31%	−2.328
	13	0.178	6.809	4.40	0.137	6.808	4.40	+0.01%	−0.001
	14	0.199	6.446	3.20	0.158	6.446	3.34	+4.22%	0.000
STN	15	0.121	8.133	10.02	0.078	8.147	10.25	+2.25%	+0.015
	16	0.239	5.379	7.54	0.154	5.631	7.32	−2.82%	+0.252
	17	0.165	8.301	20.60	0.167	5.423	3.78	−81.64%	−2.878
	18	0.209	7.107	13.53	0.177	4.661	8.51	−37.13%	−2.446
	19	0.262	5.753	8.66	0.193	4.559	1.42	−83.55%	−1.194
	20	0.219	4.900	18.17	0.210	4.900	6.20	−65.85%	0.000
	21	0.305	4.768	3.93	0.221	4.702	4.22	+7.34%	−0.066
	22	0.244	5.365	3.34	0.167	4.744	2.75	−17.76%	−0.622
	23	0.270	4.858	9.89	0.196	4.483	7.19	−27.25%	−0.374
	24	0.195	7.275	12.53	0.161	5.234	8.58	−31.48%	−2.040
	25	0.135	7.958	4.99	0.108	7.958	5.09	+1.94%	0.000
VLc	26	0.189	6.426	11.76	0.144	6.209	10.37	−11.79%	−0.036
	27	0.299	5.186	9.18	0.211	4.893	6.28	−31.54%	−0.293
	28	0.262	7.356	20.50	0.186	3.385	6.43	−68.66%	−3.971
	29	0.033	10.001	5.15	0.027	10.087	5.16	+0.17%	+0.086
	30	0.181	6.996	7.78	0.145	6.983	6.46	−16.89%	−0.013
AVERAGES		0.201	7.169	11.11	0.158	6.051	6.45	−27.98%	−1.118

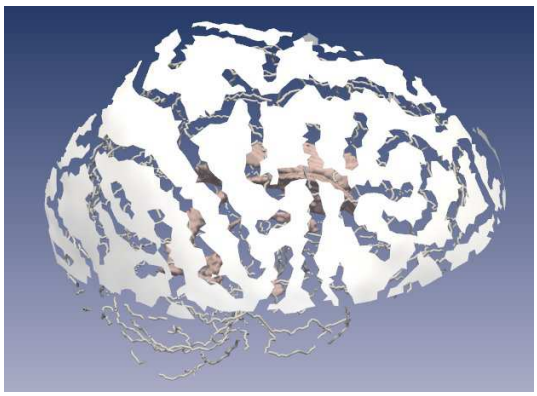


Fig. 9 3D mesh of the solution space resulting from the solving of strict constraints during phase 1 of the process for case #3. The initial *insertion zone* (skin mesh) has been reduced to the surface of points that satisfy the strict constraints

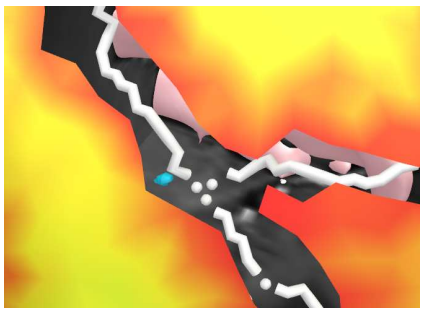


Fig. 10 Detail of the risk map (in red/orange/yellow). The target is in blue, the cortical sulci in white, the pink shape in the back is a part of the ventricles, and the background is the grayscale MRI. The direction of the sight is so close to a cortical sulcus that it doesn't cross the surface of possible entry points and we can see the target. On the sides of the cortical sulci, the areas of the surface of possible entry points are red on the borders, showing a risky proximity with the nearby cortical sulci. Figure reprinted from [16]

representing values of f_{depth}, #6 (distance to sulci) representing values of f_{risk}, #7 (orientation) representing values of f_{ori}. A detail of the risk map is shown on

Fig.10: the direction of the sight is so close to a cortical sulcus that it doesn't cross the surface of possible entry

points and we can see the target. On the sides of the cortical sulci, the areas of the surface of possible entry points is red on the borders, showing a risky proximity with the nearby cortical sulci. The color map for constraint #8 ("'tip close to center of the target"') is not shown as it offers little interest: it is mainly green and is only useful for adjustment. Fig.11(d) shows the color map resulting of the aggregative soft constraint combining the others. The red line represents the optimal trajectory.

Let us call T_{plan} this optimal trajectory produced by our path planning algorithm, T_{ref} the reference trajectory of the electrode segmented from the post-operative CT, and α the angle between them. Angle α assumes a same origin for both trajectories which are coplanar: α is measured along this common plane. Results are shown on Table 2, along with the global score (*i.e.* the value of f) of each trajectory T_{plan} and T_{ref}, and the minimal distance to sulci. As explained in Section 3.4, the values of f are real numbers between 0 and 1 (0 being the best score and 1 the worst). They are the result of a minimization, and their proximity with zero represents in some way the degree of satisfaction of the weighted soft constraints. We can notice that in all cases T_{plan} has a better (smaller) global score than T_{ref}. The scores are improved by an average of 47.69%. Let us notice that in some cases such as case #4, the score of reference trajectory T_{ref} is 1.000: its minimal distance to the sulci is so small that it received a disqualifying score.

On this table we can also see that in all cases our trajectory improves the distance to the sulci, which is increased by an average of 4.342 mm, signifying a less risky path. Angle α between T_{ref} and T_{plan} has an average of 11.11 degrees, which is a good result but not entirely satisfactory if our aim is to be as close as possible to the reference trajectory.

Execution times are also shown on this table. The computation of the solution space (surface of possible insertion) takes about 6.6 seconds. Then, coloring the maps takes an average of 103 seconds. The final optimization of the optimal trajectory takes less than 30 seconds. At the end, the entire process takes a little more than 2 minutes (average of 138.36 seconds). The experiments were performed on a 15" laptop, with a Dual Core CPU at 2.26 GHz and 4GB RAM, equipped with a NVIDIA GeForce Go 9300M GS GPU which is used to speed up occlusion computation.

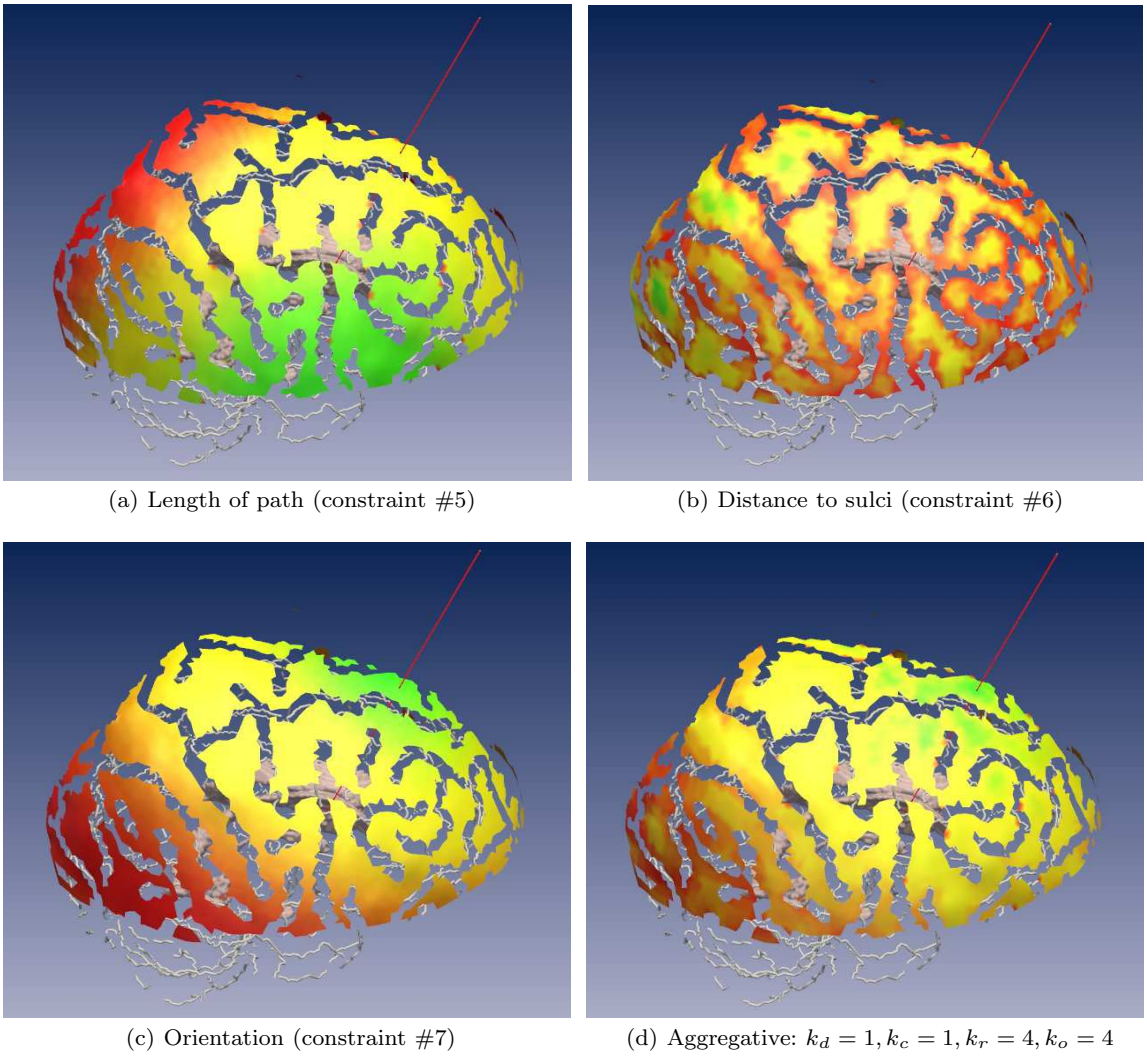
In order to test the importance of the weighting factors, we performed again the experimentation with alternative factors. We chose to give more importance to the orientation of the trajectory by giving a factor of 10 to k_o (leaving the others unchanged). The results of this second experimentation is shown on Table 3. On this table, we compare the new trajectory $T_{plan}2$ with the previous trajectory T_{plan} (which we renamed for the occasion $T_{plan}1$). The scores are improved, except for patient cases #4, #10, and #17, where the scores are nearly equal. The important observation here is that the angle between the reference trajectory and the computed trajectory is significantly improved: trajectory $T_{plan}2$ is nearly 30% closer from T_{ref} than $T_{plan}1$. The improvement reaches up to 84%. Fig.12 illustrates an example of the proximity between our computed trajectory (T_{plan}, in red) and the reference trajectory (T_{ref}, in green). In the same time, the distance to sulci is smaller, which can be more critical. However, it is still greater than for the reference electrode, and the loss is only of about 1 mm which is quite small. So such weightings could also be considered. The setting of the weighting factors will be discussed in the next section.

Fig.13 shows a snapshot of the color map of the global aggregative soft constraint with the second set of weighting factors. Again, on this figure, the trajectory computed as optimal T_{plan} using the new weighting factors is shown as a red line. We can observe that the repartition of colors is different from Fig.11(d): the new weighting factors favoured orientation over distance to sulci, and we see that there is less yellow around green zones in the upper right part, and inside red zones in the lower left part. The green zone is more homogeneous and focused on the upper right part.

Fig.10 shows a detail of the color map of the risk constraint. The map stops where a trajectory towards the target (in blue) would meet an obstacle: here an approximation of a sulcal vessel represented by the white shape. This is resulting from the solving of strict constraint during phase 1. Along the borders, the map tends to be red, signifying the risk of passing too close. This is computed during the solving of soft constraints at phase 2.

5 Discussion

We proposed a method for assisting the surgeon's decision making process during DBS surgical planning. It first provides information for the neurosurgeon in the form of colored maps that allow the individual scores of the possible insertion areas for each of the defined rules to be seen very quickly and easily, facilitating the decision making. This method also computes an optimal path according to those rules, *i.e.* the path that has the best global score, result of the aggregative cost function, according to the chosen weightings. The computed angle α demonstrated that the trajectories performed by the expert neurosurgeon were not so far from our



(a) Length of path (constraint #5)

(b) Distance to sulci (constraint #6)

(c) Orientation (constraint #7)

(d) Aggregative: $k_d = 1, k_c = 1, k_r = 4, k_o = 4$

Fig. 11 Color maps of the soft constraints obtained after phase 2 of the solving process: best zones are in green and worst are in red (patient case #3). In all snapshots the red line is the final trajectory T_{plan}, optimizing the aggregative soft constraint

proposed path (average of 6.45 degrees with the second set of weightings), but not exactly the same. However in this kind of retrospective study it is difficult to compare the results with ground truth, as the trajectories that were actually used might not be the optimal ones. That is why we used constraints and weighting factors defined by the neurosurgeon and computed the scores from them, even if we were aware that using such validation metrics was biased by the fact that the method was built for minimizing these values.

We can analyze the results in different ways. The weighting factors in the aggregative cost function might have not been chosen at best, and might need to be refined to obtain a better fitting with the reference trajectory. Indeed, in most of cases (24 over 30) the angle between the computed trajectory and the reference trajectory was improved when using weighting factors favorable to the orientation aspect. This suggests that in clinical routine, the optimization of the orientation of the electrode relatively to the axis of the target may be of greater importance for the neurosurgeons than they thought when setting the weighting factors. However we can also say on the contrary that, as the rules and the weighting factors we used were defined by the neurosurgeons, the trajectories we plan better fit their theoretical criteria (the global scores of T_{plan} are better than the scores of T_{ref}), and maybe the planning tool they used in clinical routine did not provide them sufficient information and visibility for a correct selection. Anyway, we consider to study more precisely and refine the setting of the weighting factors in future works.

The method by Seitel et al. [32] we cited in Section 2 uses the concept of Pareto-optimality to provide a set of optimal insertion points, *i.e.* all the points that satisfy

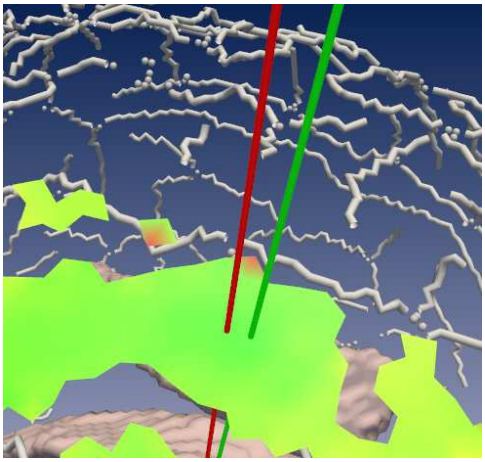


Fig. 12 Detail of the result of Fig.13. The red line is the computed trajectory T_{plan}, and the green line is the reference trajectory T_{ref}

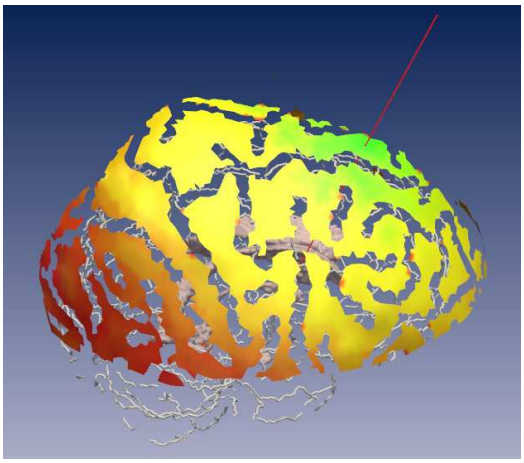


Fig. 13 Color map of the aggregative soft constraint for patient case #3, for alternative values of the weighting factors: $k_d = 1, k_c = 1, k_r = 4, k_o = 10$. T_{plan} is at another location.

all the criteria simultaneously at best for any weighting. Let us note that the optimal insertion point that our software proposes is on the Pareto frontier, *i.e.* it satisfies all the criteria simultaneously at best for a particular weighting, it is one of the Pareto-optimal points. So when a proposition has been done and the surgeon modifies the weighting factors interactively, the new optimal solution that is shown is the Pareto-optimal point which corresponds to the new weightings. This way, the surgeon does not need to browse all the Pareto-optimal points to know which one satisfies which criteria the most. He/she already knows when giving the preference to some criteria over the others that the proposed trajectory is the one that satisfies this setting at best.

We might also discover in the future that a constraint is implicitly used by the neurosurgeon but has not been expressed so far. In that case, a great advantage of our approach is the modularity. The new constraint only has to be translated into a cost function, formalized, and simply added into the XML file, and it will be automatically taken into account in the next planning. For instance, if we want to eliminate the insertion areas located on the posterior part of the head, behind the central sulcus, we could easily add for instance a strict constraint restricting the angle of the trajectory with respect to the Talairach coordinate system to a certain range in the anterior-posterior rotation.

Other 3D shapes of structures could also be considered to participate in the computation. For instance, if the 3D mesh of vessels was available, we could load it with the other 3D structures in the software, and mark them as obstacles in order to take them into account in a simple way. In the same way, if we dispose of information about the functional zones and their 3D shape and location, for example thanks to an atlas or functional imaging, we can also integrate them as obstacles in the process. Our system is ready to receive any extra 3D mesh to consider, and any new constraint which can be expressed with the available operators. In terms of performance, each new constraint added to the list will slow down the process, in a proportion that usually depends on the size and level of refinement of the involved polygonal meshes. We did not compute this exactly, but we can estimate that each constraint represents a few seconds more, which keeps the process within a time reasonable enough to be used in practice.

We chose to use the 3D polygonal representations of all the anatomical structures, even if segmentation and reconstruction steps are required to obtain the meshes. Our choice was leaded by two factors. Firstly, the human eye is used to see surfaces of objects and, to our opinion, the volume rendering with transparent layers is still not ergonomic enough to be the best choice for visualizing the trajectory and its surroundings with a sufficient precision in perception. Secondly, one long term goal of this work is to use it in conjunction with neuronavigation, and to display the trajectory and the surrounding anatomical structures with augmented reality techniques. For this purpose, the polygonal meshes will be required and reconstructed anyway. This implies that we reconstruct the target, even in the case of targets that cannot be easily defined with an enclosed structure, such as for example the ventral interior internal capsule. This could be done with the help of anatomical atlases.

A point which could be improved is the shape of the trajectory itself. In the first phase of the intervention, when inserting the micro-electrodes for the electrophysiological tests, the neurosurgeon can use multiple tracks. So it would be beneficial if our solver would

be able to take into account the possibility to have multiple tracks and find a cylinder-shaped trajectory clear from obstacles instead of just the linear trajectory. This long micro-electrodes mapping is often used to compensate the phenomenon of *brain shift* [38] which has been recognized as a possible source of inaccuracy.

Let us add that in order to measure the error, we used as the reference trajectory a line resulting from a PCA on the segmented reference electrode. This could be subject to discussion, as we could have chosen another ground truth: either the line joining the hole in the skull to the center of the target, or the planned trajectory recorded by the surgeons. Both the first one and the validation method we chose are approximations, not taking into account the bending of the electrode due to pneumoencephalus and brain shift. Taking the bending into account would suggest to be able to model and predict those deformations. We will work on this point in a close future. Using the trajectory that was planned by the surgeon before the intervention would have been an interesting alternative for our study, as this reference trajectory is also straight and does not take the bending into account either, however as this planned trajectory can sometimes be readjusted during the intervention if a vessel appears to be on the way, we preferred to use the real post-operative data.

Let us also notice that the different errors that might be present in the different steps of the process leading to our input data have not been measured. With current imaging technologies, intra-subject rigid registration appears to be very effective, especially when images are subject to reliable pre-processing tools. However, there could be slight uncertainties in the patient-to-template registration, the segmentation, or the reconstruction processes, which can possibly be cumulative, and temper our results. When dealing with deep brain structures in the brain, or vessels, the different image processing steps of the workflow should be as reliable as possible to minimize errors and reach sub-millimetric precision. Futures works will integrate these parameters in order to optimize trajectories accordingly.

Finally, this approach does not account for targets where the electrode tip should be outside of the target, for example multipolar stimulation of the dorso-ventral region of the subthalamic nucleus. For now we consider the trajectory as a line. In future works we will also study volumetric tools. This might be a lead if we consider the multiple stimulators as one volumetric tool.

6 Conclusion

We described an approach using a geometric constraint solver fed with two types of input data: the formalization of the rules governing DBS planning and patient-specific and generic images, to automatically compute an optimal placement for an electrode in the framework of assistance to DBS planning. The results we obtained show that the solutions proposed by our solver slightly differed from the solutions that were actually performed in clinical routine, but they were however close to them and had better scores regarding the rules defined by the surgeon themselves.

In the future we plan to feed the solver with other types of information which may be expressed by new rules and associated constraints, especially when new targets will be considered. The choice of the weighting factors could also be refined in a further study. The modularity of our system will facilitate such extensions.

Acknowledgements The authors would like to thank everyone from DKFZ Heidelberg (MITK), and more particularly Markus Engel, for their useful advices and help.

References

1. Adhami, L., Coste-Manière, E.: Optimal planning for minimally invasive surgical robots. IEEE Transactions on Robotics and Automation **19**(5), 854–863 (2003)
2. Altrogge, I., Kröger, T., Preusser, T., Büskens, C., Pereira, P., Schmidt, D., Weihusen, A., Peitgen, H.: Towards optimization of probe placement for radiofrequency ablation. In: proceedings of MICCAI'06, *Springer LNCS*, vol. 4190, pp. 486–493 (2006)
3. Baegert, C., Essert-Villard, C., Schreck, P., Soler, L., Gangi, A.: Trajectory optimization for the planning of percutaneous radiofrequency ablation of hepatic tumors. Computer Aided Surgery **12**(2), 82–90 (2007)
4. Baegert, C., Villard, C., Schreck, P., Soler, L.: Multi-criteria trajectory planning for hepatic radiofrequency ablation. In: proceedings of MICCAI'07, *Springer LNCS*, vol. 4791, pp. 584–592 (2007)
5. Baegert, C., Villard, C., Schreck, P., Soler, L.: Precise determination of regions of interest for hepatic rfa planning. In: proceedings of Medicine Meets Virtual Reality (MMVR 15), vol. 125, pp. 31–36. IOS Press (2007)
6. Benabid, A., Chabardes, S., Mitrofanis, J., Pollak, P.: Deep brain stimulation of the subthalamic nucleus for the treatment of parkinson's disease. The Lancet Neurology **8**(1), 67–81 (2009)
7. Benabid, A., Pollak, P., Gross, C., Hoffmann, D., Benazzouz, A., Gao, D., Laurent, A., Gentil, M., Perret, J.: Acute and long-term effects of subthalamic nucleus stimulation in parkinson's disease. Stereotact Funct Neurosurg **62**, 76–84 (1994)
8. Bergman, H., Wichmann, T., DeLong, M.: Reversal of experimental parkinsonism by lesions of the subthalamic nucleus. Science **249**(4975), 1436–1438 (1990)
9. Botsch, M., Kobbelt, L.: A remeshing approach to multiresolution modeling. In: proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 185–192 (2004)
10. Bourbakis, N., Awad, M.: A 3-D visualization method for image-guided brain surgery. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **33**(5), 766–781 (2003)

11. Brunenberg, E., Vilanova, A., Visser-Vandewalle, V., Temel, Y., Ackermans, L., Platel, B., ter Haar Romeny, B.: Automatic trajectory planning for deep brain stimulation: A feasibility study. In: proceedings of MICCAI'07, *Springer LNCS*, vol. 4791, pp. 584–592 (2007)
12. Chaturvedi, A., Butson, C.R., Lempka, S.F., Cooper, S.E., McIntyre, C.C.: Patient-specific models of deep brain stimulation: Influence of field model complexity on neural activation predictions. Brain Stimulation **3**(2), 65–77 (2010)
13. Cointepas, Y., Mangin, J., Garnero, L., Poline, J., Benali, H.: BrainVISA: Software platform for visualization and analysis of multi-modality brain data. NeuroImage **13**(6), 98–98 (2001). URL http://brainvisa.info/
14. Coupé, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., Barillot, C.: An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. IEEE Transactions on Medical Imaging **27**(4), 425–441 (2008)
15. D'Haese, P.F., Cetinkaya, E., Konrad, P.E., Kao, C., Dawant, B.M.: Computer-aided placement of deep brain stimulators: from planning to intraoperative guidance. IEEE Transactions on Medical Imaging **24**(11), 1469–1478 (2005)
16. Essert, C., Haegelen, C., Jannin, P.: Automatic computation of electrodes trajectory for deep brain stimulation. proceedings of Medical Imaging and Augmented Reality pp. 149–158 (2010)
17. Fujii, T., Emoto, H., Sugou, N., Mito, T., Shibata, I.: Neuropath planner-automatic path searching for neurosurgery. In: proceedings of CARS'03, vol. 1256, pp. 587–596. Elsevier (2003)
18. Guo, T., Parrent, A., Peters, T.: Automatic target and trajectory identification for deep brain stimulation (DBS) procedures. In: proceedings of MICCAI'07, *Springer LNCS*, vol. 4791, pp. 483–490 (2007)
19. Hellier, P., Barillot, C.: Coupling dense and landmark-based approaches for nonrigid registration. IEEE Transactions on Medical Imaging **22**(2), 217–227 (2003)
20. Koller, W., Pahwa, R., Lyons, K., Wilkinson, S.: Deep brain stimulation of the Vim nucleus of the thalamus for the treatment of tremor. Neurology **55**(12), S29–S33 (2000)
21. Lalys, F., Haegelen, C., Ferre, J., El-Ganaoui, O., Jannin, P.: Construction and assessment of a 3-T MRI brain template. NeuroImage **49**(1), 345–354 (2010)
22. Le Goualher, G., Barillot, C., Bizais, Y.: Modeling cortical sulci with active ribbons. Int. J. Pattern Recogn. Artif. Intell. **11**(8), 1295–1315 (1997)
23. Lee, J., Huang, C., Lee, S.: Improving stereotactic surgery using 3-D reconstruction. IEEE Engineering in Medicine and Biology Magazine **21**(6), 109–116 (2002)
24. Lezcano, E., Gmez-Esteban, J.C., Zarranz, J.J., Lambarri, I., Madoz, P., Bilbao, G., Pomposo, I., Garibi, J.: Improvement in quality of life in patients with advanced parkinson's disease following bilateral deep-brain stimulation in subthalamic nucleus. European Journal of Neurology **11**(7), 451–454 (2004)
25. Limousin, P., Krack, P., Pollak, P., Benazzouz, A., Ardouin, C., Hoffmann, D., Benabid, A.L.: Electrical stimulation of the subthalamic nucleus in advanced parkinson's disease. New England Journal of Medicine **339**(16), 1105–1111 (1998)
26. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. **21**(4), 163–169 (1987)
27. Maleike, D., Nolden, M., Meinzer, H., Wolf, I.: Interactive segmentation framework of the medical imaging interaction toolkit. Computer methods and programs in biomedicine **96**(1), 72–83 (2009). URL http://www.mitk.org
28. Mangin, J.: Entropy minimization for automatic correction of intensity nonuniformity. In: proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 162–169 (2000)
29. Mangin, J., Coulon, O., Frouin, V.: Robust brain segmentation using histogram scale-space analysis and mathematical morphology. proceedings of MICCAI98 **1496**, 1230–1241 (1998)
30. Nowinski, W., Yang, G., Yeo, T.: Computer-aided stereotactic functional neurosurgery enhanced by the use of the multiple brain atlas database. IEEE Transactions on Medical Imaging **19**(1), 62–69 (2002)
31. Schumann, C., Bieberstein, J., Trumm, C., Schmidt, D., Bruners, P., Niethammer, M., Hoffmann, R., Mahnken, A., Pereira, P., Peitgen, H.: Fast automatic path proposal computation for hepatic needle placement. In: proceedings of SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Modeling, vol. 7625, p. 76251J (2010)
32. Seitel, A., Engel, M., Sommer, C., Redeleff, B., Essert, C., Baegert, C., Fangerau, M., Fritzsche, K., Yung, K., Meinzer, H.P., Maier-Hein, L.: Computer-assisted trajectory planning for percutaneous needle insertions. Medical Physics **38**(6), 3246–3260 (2011)
33. Shamir, R., Tamir, I., Dabool, E., Joskowicz, L., Shoshan, Y.: A method for planning safe trajectories in image-guided keyhole neurosurgery. In: proceedings of MICCAI'10, vol. 6363, pp. 457–464. Springer LNCS (2010)
34. Starr, P., Turner, R., Rau, G., Lindsey, N., Heath, S., Volz, M., Ostrem, J., Marks Jr, W.: Microelectrode-guided implantation of deep brain stimulators into the globus pallidus internus for dystonia: techniques, electrode locations, and outcomes. Journal of neurosurgery **104**(4), 488–501 (2006)
35. Vaillant, M., Davatzikos, C., Taylor, R., Bryan, R.: A path-planning algorithm for image-guided neurosurgery. In: proceedings of CVRMed-MRCAS'97, *Springer LNCS*, vol. 1205, pp. 467–476 (1997)
36. Villard, C., Soler, L., Papier, N., Agnus, V., Gangi, A., Mutter, D., Marescaux, J.: RF-Sim: a treatment planning tool for radiofrequency ablation of hepatic tumors. In: proceedings of Information Visualization, pp. 561–566. IEEE Computer Society Press (2003)
37. Wiest-Daesslé, N., Yger, P., Prima, S., Barillot, C.: Evaluation of a new optimisation algorithm for rigid registration of MRI data. In: Proceedings of SPIE Medical Imaging, vol. 6512, p. 651206 (2007)
38. Winkler, D., Tittgemeyer, M., Schwarz, J., Preul, C., Strecker, K., Meixensberger, J.: The first evaluation of brain shift during functional neurosurgery by deformation field analysis. Journal of Neurology, Neurosurgery & Psychiatry **76**(8), 1161–1163 (2005)
39. Wolf, I., Vetter, M., Wegner, I., Bttger, T., Nolden, M., Schbinger, M., Hastenteufel, M., Kunert, T., Meinzer, H.P.: The Medical Imaging Interaction Toolkit. Medical Image Analysis **9**(6), 594–604 (2005). URL http://www.mitk.org
40. York, M.K., Wilde, E.A., Simpson, R., Jankovic, J.: Relationship between neuropsychological outcome and DBS surgical trajectory and electrode location. Journal of the Neurological Sciences **287**(1-2), 159 – 171 (2009)