

PyDBS: an automated image processing workflow for deep brain stimulation surgery.

Tiziano d'Albis, Claire Haegelen, Caroline Essert, Sara Fernández-Vidal, Florent Lalys, Pierre Jannin

▶ **To cite this version:**

Tiziano d'Albis, Claire Haegelen, Caroline Essert, Sara Fernández-Vidal, Florent Lalys, et al.. PyDBS: an automated image processing workflow for deep brain stimulation surgery.. International Journal of Computer Assisted Radiology and Surgery, 2015, 10 (2), pp.117-28. 10.1007/s11548-014-1007-y . inserm-01116063

HAL Id: inserm-01116063

https://inserm.hal.science/inserm-01116063

Submitted on 12 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PyDBS: An automated image-processing workflow for deep-brain stimulation surgery

Tiziano D'Albis · Claire Haegelen · Caroline Essert · Sara Fernández-Vidal · Florent Lalys · Pierre Jannin

Abstract

Purpose Deep-brain stimulation (DBS) is a surgical procedure for treating motor-related neurological disorders. DBS clinical efficacy hinges on precise surgical planning and accurate electrode placement, which in turn call upon several image-processing and visualization tasks, such as image registration, image segmentation, image fusion, and 3D visualization. These tasks are often performed by a heterogeneous set of software tools, which adopt differing formats and geometrical conventions, and require patient-specific parameterization or interactive tuning. To overcome these issues, we developed PyDBS, a fully-integrated and automated image-processing workflow for DBS surgery.

Methods PyDBS consists of three image-processing pipelines and three visualization modules assisting clinicians through the entire DBS surgical workflow, from the preoperative planning of electrode trajectories to the the postoperative assessment of electrode placement. The system's robustness, speed and accuracy were assessed by means of a retrospective validation, based on 92 clinical cases.

Results The complete PyDBS workflow achieved satisfactory results in 92% of tested cases, with a median processing time of 28 minutes per patient.

Conclusion The results obtained are compatible with the adoption of PyDBS in clinical practice.

Tiziano D'Albis
INSERM, U1099, Rennes Cedex, France
LTSI, Campus de Villejean, Université de Rennes 1
2 av Léon Bernard, Cedex F-35043, Rennes

Tel: +33 (0)2 23 23 49 20
Fax: +33 (0)2 99 84 71 71

Email: tiziano.dalbis@univ-rennes1.fr

Claire Haegelen

INSERM, U1099, Rennes Cedex, France
LTSI, Université Rennes I, av du Prof. Léon Bernard, F-35042 Rennes
CHU Rennes, Département de Neurochirurgie, 2 rue Henri Le Guilloux, F-35000 Rennes

Caroline Essert

LSIIT, University of Strasbourg, Pôle API
Boulevard S. Brant, F-67412 Illkirch

Sara Fernández-Vidal

CRICM, UPMC, INSERM UMR S975, CNRS UMR 7225
47 Bd de l'Hôpital, F-75013 Paris

Florent Lalys

INSERM, U1099, Rennes Cedex, France
LTSI, Université Rennes I, 2 av du Prof. Léon Bernard, F-35042 Rennes

Pierre Jannin

INSERM, U1099, Rennes Cedex, France
LTSI, Université Rennes I, 2 av du Prof. Léon Bernard, F-35042 Rennes

Keywords: Surgical planning · Deep Brain Stimulation · 3D visualization

Introduction

High-frequency deep brain stimulation (DBS) is a minimally-invasive surgical therapy for treating motor-related disorders (*e.g.*, Parkinson's disease, essential tremor, and dystonia), which has recently been extended to the treatment of severe psychiatric disorders (*e.g.*, obsessive-compulsive disorder and severe depression). DBS consists in implanting electrodes in the basal ganglia, located deep in the patient's brain, and delivering high-frequency electric pulses to the surrounding neural tissues. The reversibility of this therapy, along with its spectacular outcomes, have largely contributed to its success. It is important to note that both the quality of clinical improvement and the occurrence of adverse effects have been found to correlate with electrode placement accuracy [1-3]. Nevertheless, accurate positioning of DBS electrodes is still a challenging and laborious procedure, requiring extensive surgical, anatomical, and functional knowledge of the brain, and typically relying on several manual adjustments and tuning [4].

Electrode trajectories are planned prior to surgery based on preoperative magnetic resonance (MR) imaging. This is a complex task, given that the brain structures targeted for stimulation are often not directly visible on conventional MR sequences due to limitations in image contrast and spatial resolution. Surgeons therefore often rely on multiple co-registered MR sequences or digital anatomical atlases for targeting. Once a trajectory has been planned, a frame-based stereotactic procedure is conducted. To express the planned trajectory in stereotactic coordinates, the frame is detected on a MR or computer tomography (CT) scan of the patient's skull (after frame fixation), and this scan is then registered to the MR image used for targeting. During surgery, the planned trajectory is refined by means of neurophysiological microelectrode recordings (MERs) and direct observation of the stimulation's clinical outcomes. Following surgery, the actual position of the electrode contacts is assessed by segmenting the corresponding artifacts on a postoperative CT scan. It is essential to know the exact position of the electrode contacts after surgery for the purposes of both lead programming and clinical research studies.

The DBS surgical workflow described above requires several image-processing and visualization tasks, such as intra-modal and mono-modal image registration, intensity-based and atlas-based image segmentation, and image fusion and 3D visualization. The accuracy, robustness, and usability of the software tools employed for these tasks are all critical elements for the success of the entire surgical workflow. At present, most clinical centers rely on commercial software systems for DBS surgery. These systems, however, often restrict their functionalities to the planning phase of the workflow, demonstrate a lack of state-of-the-art registration and segmentation algorithms, and typically require several manual adjustments before the final result can be obtained. In medical-imaging research, great effort is put into the design, implementation, and validation of novel image-processing algorithms, which address very specific computational problems and are recognized for their ever-increasing performance levels. Nevertheless, only in a few instances have research studies taken a

broad perspective at the surgical-workflow level. This kind of wide view is necessary to bridge the gap between research and application, with the aim of developing a software system that is effective in clinical practice. In this respect, effort should be spent in the integration and automation of the different algorithms and software tools available, in the interest of developing a coherent and usable software environment. This coherent software environment would then also be beneficial for the research itself by providing a shared platform where different algorithms can be easily compared, or even combined for solving problems at higher levels of abstract complexity. The study presented in this paper contributes to this goal.

Related work

Several solutions have been proposed to assist physicians through the different stages of DBS surgical workflow. The study by Guo *et al.* [5] has presented a visualization and navigation system for stereotactic DBS procedures. This system aids physicians in the planning phase by registering and fusing anatomical and functional atlases to preoperative patient images and automatically detecting the stereotactic frame on MR or CT scans. Intraoperatively, the system simulates the microelectrode descent along the stereotactic trajectory and collects MER data for the construction of functional atlases. Similar functionalities have also been implemented into the Cicerone system [6]. Despite this system not providing an automatic frame-detection algorithm, it does offer the additional functionality of visualizing a predicted volume of tissue activated by the stimulation. Neither of these systems, however, addresses the postoperative phase of the workflow, where the implanted electrodes and stimulation contacts are to be detected.

Another software environment developed in order to provide assistance for DBS surgery is CranialVault [7]. This software combines a centralized data repository with a series of software tools for automatic data processing and visualization (CRAVE tools). The planning module includes functionalities for rigid registration of patient preoperative MR and CT scans and for non-linear deformation of multiple anatomical atlases in patient space. These atlases are used for both atlas-based segmentation of brain structures and localization of two anatomical landmarks used in stereotactic neurosurgery: the anterior commissure (AC) and the posterior commissure (PC). Furthermore, a proposed stimulation target is estimated by projecting target centroids from previously implanted patients onto patient space, with electrophysiological and somatotopic maps also provided. By means of a standard-angle approach, the system provides pre-computed implantation plans that can be modified or validated by the surgeon. CranialVault and the CRAVE tools also include an intraoperative module with similar features to those implemented in the first two studies mentioned [5,6], along with a postoperative module for electrode contact detection.

Even though it includes a large set of functionalities, CranialVault does not provide segmentations of the brain vessels or cortical sulci. Given that brain vessel damage, particularly that which affects the large arteries lying within the cortical sulci, is a major risk factor in DBS surgery, it is highly desirable to segment these structures and visualize them during preoperative surgical planning. Furthermore, as CranialVault relies on a patient-

specific stereotactic frame, it does not provide functionalities for the segmentation and detection of standard stereotactic frames, such as the Leksell G-frame (Elekta AB, Stockolm, Sweden) or CRW frame (Radionics, Burlington, MA, USA). It should also be noted that, to date, only the planning module has been preliminary validated [7]. This consisted in measuring the accuracy of a) co-registration of patient's preoperative images, b) detection of points AC and PC, and c) prediction of the stimulation targets, by ensuring planning consistency across surgeons [8]. Nevertheless, a validation of the segmented brain structures has not been reported.

Contribution

To the best of our knowledge, a complete, fully-integrated and automatic image-processing workflow for planning and postoperative assessment of DBS interventions has not yet been reported, despite all the previous related works. Furthermore, software's robustness, speed and accuracy have never been assessed in a systematic validation study. In this paper we present PyDBS, an automated image-processing workflow aiding surgeons and clinicians in the planning and assessment of DBS procedures. Leveraging on state-of-the-art algorithms and software tools, PyDBS provides a fully-integrated and automatic software environment for DBS surgery. The system's robustness, speed and accuracy were evaluated through a retrospective study involving 92 clinical cases. The results obtained are compatible with the adoption of PyDBS in clinical practice.

Materials and methods

PyDBS comprises three image-processing pipelines and three visualization/interaction modules. Each pipeline is executed at a different phase of the DBS surgical workflow, with an associated 3D scene collating all results generated at completion. The 3D scenes, specified in the Medical Reality Markup Language (MRML), can be readily visualized by means of a 3D Slicer [9, 10]. Using the MRML format, the full set of parameters required for 3D visualization, such as screen layout, colors, transparencies, camera viewpoints, and geometrical transformations, can, in fact, be pre-computed and stored within the scene file itself. Furthermore, the user can easily interact with a generated 3D scene by means of the three PyDBS visualization/interaction modules implemented as 3D Slicer plugins.

PyDBS in the surgical workflow

Fig. 1 displays typical usage of PyDBS within the DBS surgical workflow. Prior to surgery, T1-weighted and T2-weighted MR images of the patient's brain are acquired. The two images, along with a digital anatomical atlas, are the input to the PyDBS inclusion pipeline. The inclusion pipeline, following interactive localization of the anatomical landmarks AC and PC, then performs an intensity-based segmentation of the scalp, brain, and cortical sulci, as well as an atlas-based segmentation of the brain ventricles and basal ganglia. The scalp surface

serves as definition of the space of possible entry points for the implantation, with the cortical sulci and brain ventricles representing structures to be avoided by the electrode trajectory [11], and the basal ganglia the stimulation targets. The T2-weighted image is then also registered onto the T1-weighted image for an improved visualization of the deep basal ganglia nuclei (*e.g.*, the red nucleus). The pipeline output consists of an MRML scene containing the registered MR images, in addition to the masks and meshes of the segmented anatomical structures. Based on this scene, and by using the dedicated PyDBS plugin, the surgeon validates the results and interactively defines the electrode trajectories for the implantation (Fig. 2).

On the day of surgery, a surgical stereotactic frame is fixed on the patient's skull and head CT scan is performed. This scan, along with a geometrical 3D model of the surgical frame, provides the input to the PyDBS preoperative pipeline. At this point, the head CT scan is registered onto the T1-weighted MR volume, the skull segmented, and the surgical frame detected. With the benefit of frame detection, the planned trajectories originally defined in the patient's AC-PC space can be expressed in the coordinates of the stereotactic frame. The pipeline output consists of an MRML scene containing the results of both the inclusion and preoperative pipelines. Based on this scene, the surgeon validates the MR-CT registration and frame detection, then retrieves the stereotactic coordinates of the planned trajectories. The surgeon is also able to visualize the estimated positions of the five micro-electrodes used for intraoperative trajectory refinement. Furthermore, if a trajectory angle has been modified during surgery, the new trajectory can be readily updated and visualized in the space of the preoperative MR image.

Following surgery, a postoperative head CT scan is performed. This image, along with a geometrical 3D model of the implanted electrodes, represents the input to the PyDBS postoperative pipeline. At this point, the postoperative CT is registered onto the preoperative CT and projected onto the patient's AC-PC space. Furthermore, the implanted electrodes are segmented based on the postoperative CT image, and the stimulation contacts are localized. Lastly, an MRML scene containing the results of the three pipelines is generated. Based on this scene, the surgeon assesses the electrode contact positions and compares the real trajectories with the planned ones.

System perspective

From a computing system perspective, PyDBS operates from input data consisting of patient-specific information, such as patient images and clinical data, and generic models, such as an anatomical atlas, stereotactic frame model, and electrode model, then outputting a patient-specific model for the planning and postoperative assessment of DBS surgery (Fig. 3). The generated patient-specific model consists of images, segmented anatomical structures (*i.e.*, binary masks and meshes), and geometrical transformations (*i.e.*, linear registration matrices and non-linear deformation fields). All images, masks, and meshes are mapped to a shared

reference space (the patient's AC-PC space).

Data import

All patient-specific data is managed by an internal relational database. An automatic import procedure populates the database with relevant image metadata extracted from DICOM media (*e.g.*, modality, volume size, and sequence type), converting the DICOM series into NIfTI format [12]. NIfTI images are stored in predefined locations on the file-system and then referenced by the internal database. Clinical data, such as the brain structure targeted for stimulation, the lateralization, and electrode type, can be either automatically imported from a clinical database or interactively provided at the beginning of a pipeline execution. Additional patient clinical data (*e.g.*, clinical scores) can also be imported into the database. All data in the database is easily accessible from a web-based application or a graphical user interface, both of which are provided by PyDBS.

Pipeline execution

PyDBS pipelines are executed from the command-line or by means of a simple graphical user interface. At the beginning of each pipeline, the user is asked to select the patient image(s) to be processed from a list of matching entries in the internal database. Following this initial selection, a pipeline is automatically executed with no need for further user interaction (with the exception of manually localizing AC and PC points in the inclusion pipeline). On completion, the results, namely the images, masks, meshes, and geometrical transformations, are stored in a predefined folder on the local file-system, with an associated MRML scene description file generated for each pipeline.

Inclusion pipeline

The inclusion pipeline is composed of six processing steps (Fig. 4, left). Firstly, the two MR images of the patient brain (T1-weighted and T2-weighted) are imported from the internal database. Secondly, the user localizes the anatomical landmarks AC and PC on the T1-weighted image. This is the only step out of all three pipeline procedures that requires user interaction. We opted for the manual detection of AC and PC due to the high inter-surgeon variability that characterizes this localization [15]. Furthermore, there are several localization strategies currently in use in clinical practice, and we decided not to constrain the users in their choice. Once the AC-PC reference frame has been defined, a batch execution of the BrainVISA Morphologist pipeline [13, 14] is triggered. This pipeline performs an intensity-based segmentation of the scalp, brain (gray and white matter), and cortical sulci. We selected the BrainVISA Morphologist pipeline due to its proven robustness for segmenting the cortical sulci [13], a particularly important segmentation for the purposes of our work. The T2-weighted and T1-weighted MR images are then subject to rigid co-registration. All linear registrations are computed by means of the FMRIB's Linear Image Registration Tool (FLIRT), which has been proven to outperform other commonly-

used linear-registration algorithms in terms of accuracy and robustness [16, 17]. For both mono- and multi-modal linear-registrations, we have adopted a cost function based on mutual information.

Atlas-based segmentation

The T1-weighted image of the patient's brain is registered onto an anatomical atlas for performing an atlas-based segmentation of the brain ventricles and basal ganglia. The atlas we have adopted was constructed by averaging 57 T1-weighted MR images of Parkinson's disease patients, defining 24 anatomical structures, all manually segmented by an anatomist [18]. Atlas registration is performed over five processing steps (Fig. 5), including both linear and non-linear registrations and taking a similar global-to-local approach to the one previously described by Lalys *et al.* [19]. Firstly, all non-brain voxels are masked out of both images, followed by rigid co-registration of both (six degrees of freedom [DOF]). Secondly, the two images are affinely co-registered (12 DOF). Thirdly, a box volume corresponding to a region-of-interest (ROI) around the basal ganglia and defined in the anatomical atlas space is cropped from both images. Fourthly, the two cropped ROIs are linearly co-registered (12 DOF). Finally, the ROIs are non-linearly co-registered using the symmetric diffeomorphic registration algorithm (SyN) with cross-correlation as cost function [20, 21]. We chose this deformation algorithm based on a recent validation study that reported SyN to be one of the best performing algorithms for iconic non-linear brain image registration [22]. At the end of the registration process, brain-ventricle and basal-ganglia segmentations are projected from the atlas space onto the patient space (Fig. 6). In the final step, an MRML scene is generated.

Preoperative pipeline

The preoperative pipeline is composed of six processing steps (Fig. 4, middle). Firstly, the preoperative CT scan of the patient's head (fixed to the stereotactic frame) is imported from the internal database. Secondly, this CT image is rigidly registered onto the preoperative T1-weighted MR image of the patient's brain. Thirdly, an intensity-based segmentation of the patient's skull and the four frame screws is performed by means of a chain of thresholding and connected-component-labeling operations (Fig. 7a, top). In the fourth step, the frame artifact is segmented using an analogous strategy (Fig. 7a, bottom). In the fifth step, the frame model is rigidly registered onto the patient's CT scan. This is done by means of a two-step procedure, the first step consisting of ten landmark points automatically detected on the frame artifact, namely the two most superior points for each frame side and the three points at half of the frame height (Fig. 7b). These landmarks are used to estimate an initial rigid registration matrix between frame model and the patient's CT scan (point-based optimization algorithm [23]). Then, the two images are rigidly-co registered using the estimated matrix as initialization. At the final step, a second MRML scene is generated with these results.

Postoperative pipeline

The postoperative pipeline is composed of six processing steps (Fig. 4, right). Firstly, the postoperative CT is imported in from the internal database. Secondly, this image is thresholded in order to obtain a coarse skull segmentation. In the third module, the postoperative CT is rigidly registered onto the preoperative CT, a process which takes two steps, namely a global-search registration between the two segmented skulls, followed by a local-search registration between the two original volumes. In the fourth step, the electrode artifacts are segmented on the postoperative CT by applying the brain masks computed on the T1-weighted MR image (one per hemisphere) and by thresholding the resulting volume. In the fifth step, the electrode contacts are detected by registering the geometrical model of the implanted electrodes to the segmented artifacts. The electrode tip position is calculated using the coordinates of the most inferior voxel in the electrode artifact, and the direction of the electrode axis is estimated by applying principal-component analysis to the artifact voxels within a 15mm radius from the electrode tip (Fig. 8).

Validation

The three pipelines were retrospectively validated against a dataset comprising 92 patients available at our site. We included all patients having been implanted from September 2006 to September 2012 at the Neurosurgical Department of the University Hospital of Rennes (France). The inclusion criteria consisted of availability of: 1) 3T T1-weighted MR image with gadolinium injection (1mm x 1mm x 1mm, Philips Medical system), 2) 3T T2-weighted MR image (1mm x 1mm x 1.1mm, Philips Medical system), 3) preoperative CT scan with stereotactic frame (0.55mm x 0.55mm x 0.6mm, GE Healthcare VCT 6), and 4) postoperative CT scan with implanted electrodes (0.44mm x 0.44mm x 0.6mm, GE Healthcare VCT 64). Informed consent was obtained from all patients. Despite there being only T1-weighted MR images acquired after frame fixation available for a subset of patients (n=37), in all cases, the stereotactic frame was detected from a preoperative CT scan. The dataset comprised patients having undergone unilateral (n=14) or bilateral (n=78) DBS surgery. The targeted anatomical structures were the sub-thalamic nucleus (n=37), globus pallidus internum (n=32), or the caudal part of the ventro-lateral thalamic nucleus (n=23). Patient pathologies included Parkinson's disease (n=63), essential tremor (n=12), dystonia (n=9), Tourette syndrome (n=4), and obsessive-compulsive disorder (n=4). For all patients, the Leksell G stereotactic frame (Elekta Instruments AB, Stockholm, Sweden) has been used, and the implanted electrodes were Medtronic 3389 or 3387 electrodes (Medtronic Sofamor Danek).

The pipelines were executed on a 64-bit laptop with an Intel® i7-2860QM CPU (8 cores, 2.50GHz clock frequency) equipped with 16GB of RAM. The results were qualitatively validated by an expert, with the processing times measured for each pipeline. This qualitative validation consisted of seven steps (three for the inclusion pipeline, two for the preoperative pipeline, and two for postoperative pipeline), each supplemented by a 3D scene visualized by the expert (Fig. 9). The corresponding results were evaluated as either satisfactory or

unsatisfactory, with respect to their potential use for preoperative planning of electrode trajectories and postoperative assessment of electrode placement. If any element within the 3D scene was considered misleading for the purposes of these two tasks, the corresponding validation step was marked as unsatisfactory. Pipeline results were considered satisfactory only if all their validation steps were satisfactory.

The inclusion pipeline was validated in three steps: 1) validation of the intensity-based brain segmentation, 2) validation of the T2-weighted image registration, and 3) validation of the atlas-based segmentation of the brain ventricles and basal ganglia (Fig. 9 a-c). The preoperative pipeline was validated in two steps: 1) validation of preoperative CT registration, and 2) validation of frame detection (Fig. 9 d-e). Lastly, the postoperative pipeline was validated in two steps: 1) validation of postoperative CT registration, and 2) validation of electrode segmentation and electrode contact detection (Fig. 9 f-g).

Results

Validation results are reported in Fig. 10a. The inclusion pipeline achieved satisfactory results for all patients except two (#75 and #127), corresponding to a 98% success rate. For patient #75, the atlas-based segmentation produced unsatisfactory results. This was due to the patient exhibiting very large ventricles compared to the anatomical atlas, resulting in a mismatch between the region-of-interest cropped from the patient image and the corresponding region on the atlas. For patient #127, the intensity-based brain segmentation terminated with errors. This can probably be attributed to the presence of movement artifacts in the original T1-weighted MR volume. Given this pipeline's unsuccessful termination, the preoperative and postoperative pipelines could not be executed for this patient.

The preoperative pipeline produced satisfactory results for 86 patients out of the 91 tested, corresponding to a 94% success rate. The pipeline failed on one occasion due to an unsatisfactory registration of the preoperative CT with the preoperative MR (patient #79), and on four occasions due to an unsatisfactory frame detection (patients #39, #54, #89, and #116). Where frame detection failed, the problem was owing to an inaccurate segmentation of the frame artifact caused by non-frame voxels that were not masked out from the image. In all cases, the preoperative pipeline terminated with no reported errors.

Satisfactory results were obtained with the postoperative pipeline for 90 patients out of the 91 tested, corresponding to a 99% success rate. For one patient (#89), the pipeline failed due to an unsatisfactory registration of the postoperative CT with the preoperative CT. For this patient, despite the electrode contacts being correctly detected, their coordinates could not be correctly expressed in the AC-PC reference frame. In all cases, the postoperative pipeline terminated with no reported errors.

Combining the results of the three pipelines, satisfactory results were obtained for 85 patients out of the 92 tested, corresponding to a 92% success rate. Pipeline processing times are reported in Fig. 10b. Median processing times were approximately 15 minutes for the inclusion pipeline, 7 minutes for the preoperative

pipeline, and 4 minutes for the postoperative pipeline, with a total processing time of 28 minutes per patient. The processing times for the most time-consuming modules of each pipeline have also been reported. For the inclusion pipeline, the most time-consuming module was the intensity-based brain segmentation (Morphologist pipeline), which took approximately 7 minutes to complete, followed by the atlas registration (approximately 6 minutes), and the interactive detection of the AC and PC points (approximately 2 minutes). For both the preoperative and postoperative pipelines, the most time-consuming module was the CT volume registration, which took a median time of 3.5 minutes to complete.

Discussion and future work

In this article, we have presented PyDBS, an integrated image-processing workflow for DBS surgery. PyDBS includes three image-processing pipelines and three visualization/interaction modules. The three pipelines are fully automatic (exception made for the manual localization of the points AC and PC) and completely general, meaning that no patient-specific parameter tuning is needed. Each pipeline addresses a specific phase of the surgical workflow, generating as output images, meshes, and geometrical transformations, all collected in a 3D scene readily available for visualization. Furthermore, for each 3D scene, a PyDBS visualization/interaction module provides an easy means to access and manipulate the displayed data.

We retrospectively tested all three pipelines on a dataset of 92 patients available at our site, with the results subjected to qualitative validation by an expert. Satisfactory results were obtained across the entire workflow for 85 patients out of the 92 tested, corresponding to a 92% success rate. The cases where pipelines failed were analyzed in order to further improve our software. Particular mention should be made that the frame-detection module was found to be the least robust, as it resulted in the highest error rate (4%). In this respect, we believe that a new semi-automatic procedure for frame detection should be implemented for the cases where the automatic algorithm fails. In more general terms, we plan to supplement the three PyDBS visualization/interaction modules with additional error-correction functionalities, ensuring that the user is able to interactively adjust the automatically-generated results, whenever this may be required.

PyDBS is implemented in the Python programming language, hence its name, language which was chosen based on the clarity of its syntax and its rapidity of development, making it well-suited for research prototyping. The processing speed was not, however, compromised, as all low-level image-processing tasks, such as registration and segmentation, were delegated to pre-compiled routines. We were, in fact, able to obtain a median processing time of about 28 minutes per patient, and the preoperative pipeline, the most critical in terms of time performance, completed in about 7 minutes. This is a time performance that is compatible with clinical adoption.

Next we will discuss some of the limitations facing our work. Firstly, the system performances were evaluated by means of a qualitative validation study. Qualitative validation is commonly used in medical-imaging research when quantitative performance measures are difficult to define, or when a ground truth is not directly available

to serve as a comparison. This was the case for our study, in which we aimed to validate a complete system at the surgical-workflow level. The algorithms and software tools that are used by the three PyDBS pipelines had, in fact, been individually validated for the most part in separate studies, though their level of integration and automation, as well as their effectiveness in the specific context of DBS surgery, had yet to be assessed. This could have been conducted prospectively by evaluating the potential effects of adopting the proposed system on the DBS surgical workflow, in terms of its total duration or the associated clinical outcomes, for example. A prospective validation, however, also implies several clinical risks, and it would be preferably postponed to the late stages of a system evaluation when its performances have been already proved otherwise. We therefore opted for a retrospective validation, and measured our system's performances in terms of robustness, speed, and accuracy. In terms of accuracy, however, only a qualitative evaluation was possible owing to the lack of ground-truth for comparison. An alternative could have been to quantitatively assess system accuracy in comparison with a gold-standard. This would have been very difficult in our study, however, due to the heterogeneity of surgical procedures adopted at different clinical centers, and the substantial engineering effort that would have been required to reproduce and adapt alternative solutions to our data.

For qualitative validation, one would ideally involve a pool of experts and evaluate the results at the root of the intra-observer and inter-observer variability in the assessment. This was also difficult in our study, however, owing to the cost of involving several experts in such a demanding validation study, and the fact that only one expert was available at our site. We thus believe that a multi-observer, and possibly even multi-site validation study, is a necessary milestone we shall aim for in the future, even though the validation proposed in this paper proved very informative for future developments of the project.

Overall, the results we obtained proved PyDBS to be a valuable tool for assisting clinicians through the DBS surgical workflow. Furthermore, we plan to employ the PyDBS inclusion pipeline output as input data for an automatic-trajectory-planning software, such as the one proposed by Essert *et al.* [24]. With the advantage of its modularity and flexibility, PyDBS is also well-suited for supporting clinical research studies. For example, the PyDBS postoperative pipeline, supplemented by additional functionalities not described here, has already been employed in a clinical research study on electrode deformations [25], as well as in a study focusing on the correlation between stimulation targets and clinical scores [3].

Future projects also include the integration of anatomo-clinical atlases [3] into the PyDBS inclusion pipeline. Once registration with preoperative patient’s images has been conducted, anatomo-clinical atlases could, in fact, be used in order to identify the best stimulation targets according to the clinical outcomes of previously implanted patients. In this setting, PyDBS could be used both to visualize the atlases during planning and to build and update them whenever a new patient is treated. Finally, it is our objective to validate PyDBS prospectively by using it in parallel with the commercial DBS-planning system currently in use at our institute.

Acknowledgements

The authors thank the French National Research Agency (ANR) who founded this work through the ACouStiC project grant (ANR 2010 BLAN 0209 01).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Tisch, S., Zrinzo, L., Limousin, P., Bhatia, K.P., Quinn, N., Ashkan, K., Hariz, M.: Effect of electrode contact location on clinical efficacy of pallidal deep brain stimulation in primary generalised dystonia. J Neurol Neurosurg Psychiatry 78(12), 1314–1319 (2007)
2. York, M.K., Wilde, E.A., Simpson, R., Jankovic, J.: Relationship between neuropsychological outcome and dbs surgical trajectory and electrode location. J Neurol Sci 287(1), 159–171 (2009)
3. Lalys, F., Haegelen, C., Mehri, M., Drapier, S., Verin, M., Jannin, P.: Anatomo-clinical atlases correlate clinical data and electrode contact coordinates: Application to subthalamic deep brain stimulation. J Neurosci Methods (2012)
4. Dormont, D., Seidenwurm, D., Galanaud, D., Cornu, P., Yelnik, J., Bardinet, E.: Neuroimaging and deep brain stimulation. AJNR Am J Neuroradiol 31(1), 15–23 (2010)
5. Guo, T., Finnis, K.W., Parrent, A.G., Peters, T.M.: Visualization and navigation system development and application for stereotactic deep-brain neurosurgeries. Comput Aided Surg 11(5), 231–239 (2006)
6. Miocinovic, S., Noecker, A., Maks, C., Butson, C., McIntyre, C.C.: Cicerone: stereotactic neurophysiological recording and deep brain stimulation electrode placement software system. In: Operative Neuromodulation, pp. 561–567. Springer (2007)
7. DHaese, P.F., Pallavaram, S., Li, R., Remple, M.S., Kao, C., Neimat, J.S., Konrad, P.E., Dawant, B.M.: Cranialvault and its crave tools: A clinical computer assistance system for deep brain stimulation (DBS) therapy. Med Image Anal 16(3), 744–753 (2012)
8. Pallavaram, S., Phibbs, F.T., Tolleson, C., Davis, T.L., Fang, J., Hedera, P., Li, R., Koyama, T., Dawant, B.M., D'Haese, P.F.: Neurologist consistency in interpreting information provided by an interactive visualization software for deep brain stimulation postoperative programming assistance. Neuromodulation: Technology at the Neural Interface, 17: 11–15 (2013)
9. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., et al.: 3D Slicer as an image computing platform for the quantitative imaging network. Magn Reson Imaging (2012)

10. 3D Slicer. http://www.slicer.org. Accessed: 2013-04-05
11. Zrinzo, L., van Hulzen, A.L., Gorgulho, A.A., Limousin, P., Staal, M.J., De Salles, A.A., Hariz, M.I.: Avoiding the ventricle: a simple step to improve accuracy of anatomical targeting during deep brain stimulation: Clinical article. J Neurosurg 110(6), 1283–1290 (2009)
12. Neuroimaging Informatics Technology Initiative. http://nifti.nimh.nih.gov. Accessed: 2013-04-05
13. Rivière, D., Régis, J., Cointepas, Y., Papadopoulos-Orfanos, D., Cachia, A., Mangin, J.: A freely available anatomist/brainvisa package for structural morphometry of the cortical sulci. Neuroimage 19 (2 part 2), 19–22 (2003)
14. BrainVISA Morphologist. http://brainvisa.info. Accessed: 2013-04-05
15. Pallavaram, S., Yu, H., Spooner, J., DHaese, P.F., Bodenheimer, B., Konrad, P.E., Dawant, B.M.: Intersurgeon variability in the selection of anterior and posterior commissures and its potential effects on target localization. Stereotact Funct Neurosurg 86(2), 113–119 (2008)
16. Jenkinson, M., Smith, S., et al.: A global optimisation method for robust affine registration of brain images. Med Imag Ana 5(2), 143–156 (2001)
17. Jenkinson, M., Bannister, P., Brady, M., Smith, S., et al.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. Neuroimage 17(2), 825–841 (2002)
18. Haegelen, C., Coupé, P., Fonov, V., Guizard, N., Jannin, P., Morandi, X., Collins, D.L.: Automated segmentation of basal ganglia and deep brain structures in MRI of Parkinson's disease. Int J Comput Assist Radiol Surg 8(1), 99–110 (2013)
19. Lalys, F., Haegelen, C., Abadie, A., Jannin, P., et al.: Post-operative assessment in deep brain stimulation based on multimodal images: registration workflow and validation. In: Proc. of SPIE vol. 7261, pp. 72,612M–1 (2009)
20. Avants, B.B., Epstein, C.L., Grossman, M., Gee, J.C.: Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Med Imag Ana 12(1), 26–41 (2008)
21. Advanced Normalization ToolS. http://www.picsl.upenn.edu/ANTS. Accessed: 2013-04-05
22. Klein, A., Andersson, J., Ardekani, B.A., Ashburner, J., Avants, B., Chiang, M.C., Christensen, G.E., Collins, D.L., Gee, J., Hellier, P., et al.: Evaluation of nonlinear deformation algorithms applied to human brain MRI registration. Neuroimage 46(3), 786 (2009)
23. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Trans Pattern Anal Mach Intell 9(5), 698–700 (1987)
24. Essert, C., Haegelen, C., Lalys, F., Abadie, A., Jannin, P.: Automatic computation of electrode trajectories for deep brain stimulation: a hybrid symbolic and numerical approach. Int J Comput Assist Radiol Surg 7(4), 517–532 (2012)

25. Lalys, F., Haegelen, C., Dalbis, T., Jannin, P.: Analysis of electrode deformations in deep brain stimulation surgery. Int J Comput Assist Radiol Surg 9(1), 107-117 (2014)

.

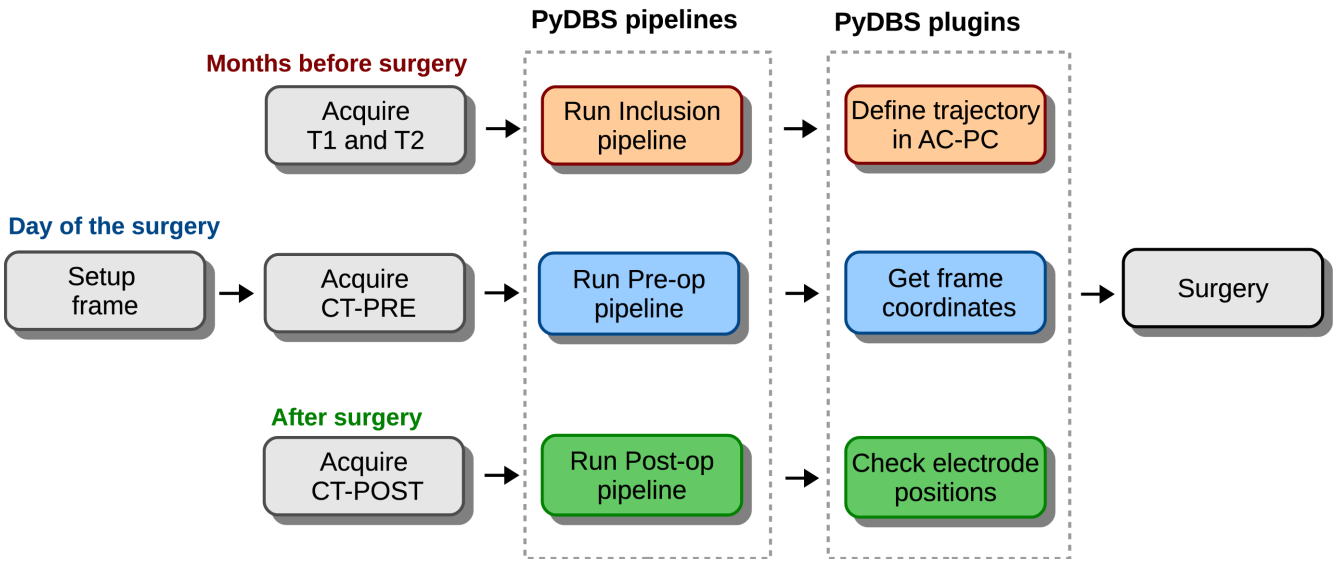


Fig. 1 Integration of PyDBS within the DBS surgical workflow.

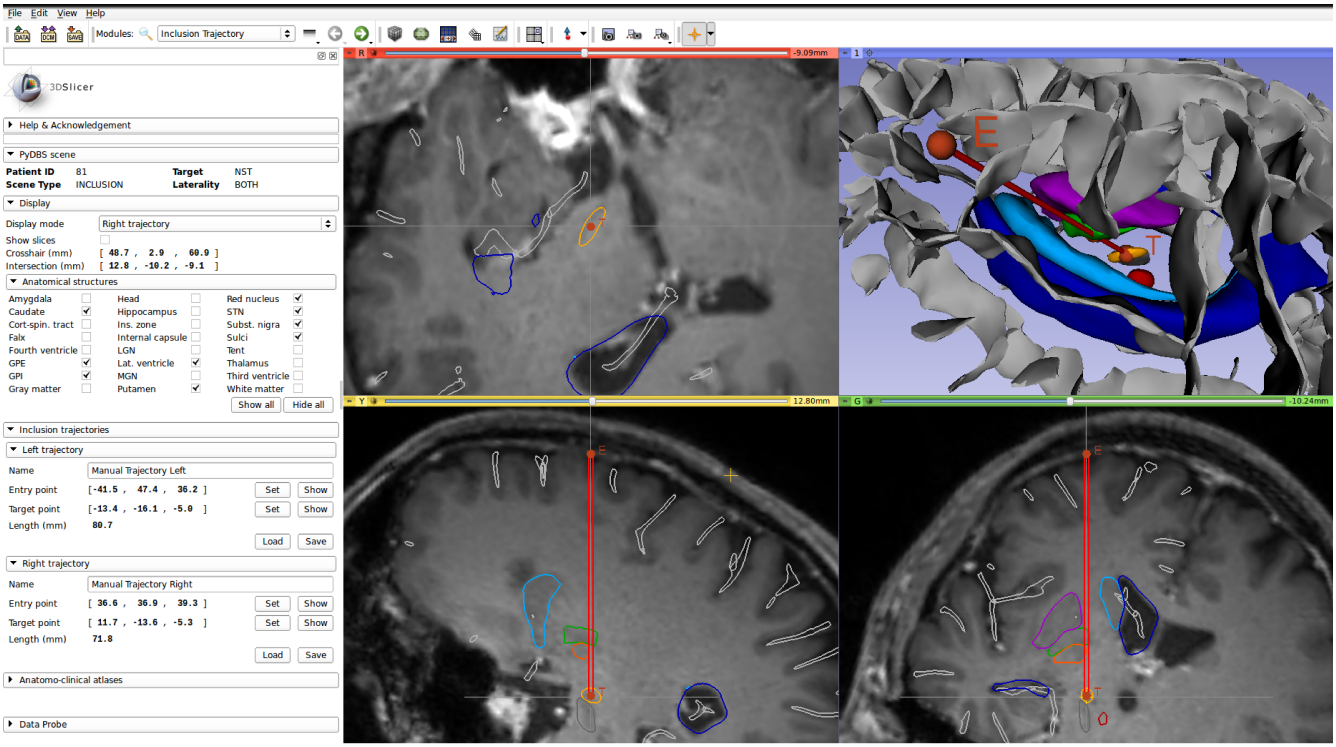


Fig. 2 MRML scene generated by the PyDBS inclusion pipeline. The three slice views are presented in "probe-eye" mode, that is, projected perpendicularly to the electrode trajectory (red cylinder). The left panel shows the PyDBS plugin for interactive trajectory planning. The user defines a trajectory by selecting an entry and target point on the MR image (red spheres, E: entry point, T: target point). A trajectory can be refined by dragging the two points on the 3D view, such as for maximizing the distance from the cortical sulci (gray meshes). Multiple trajectory plans can be defined, and plans can be saved to an XML file for later inspection.

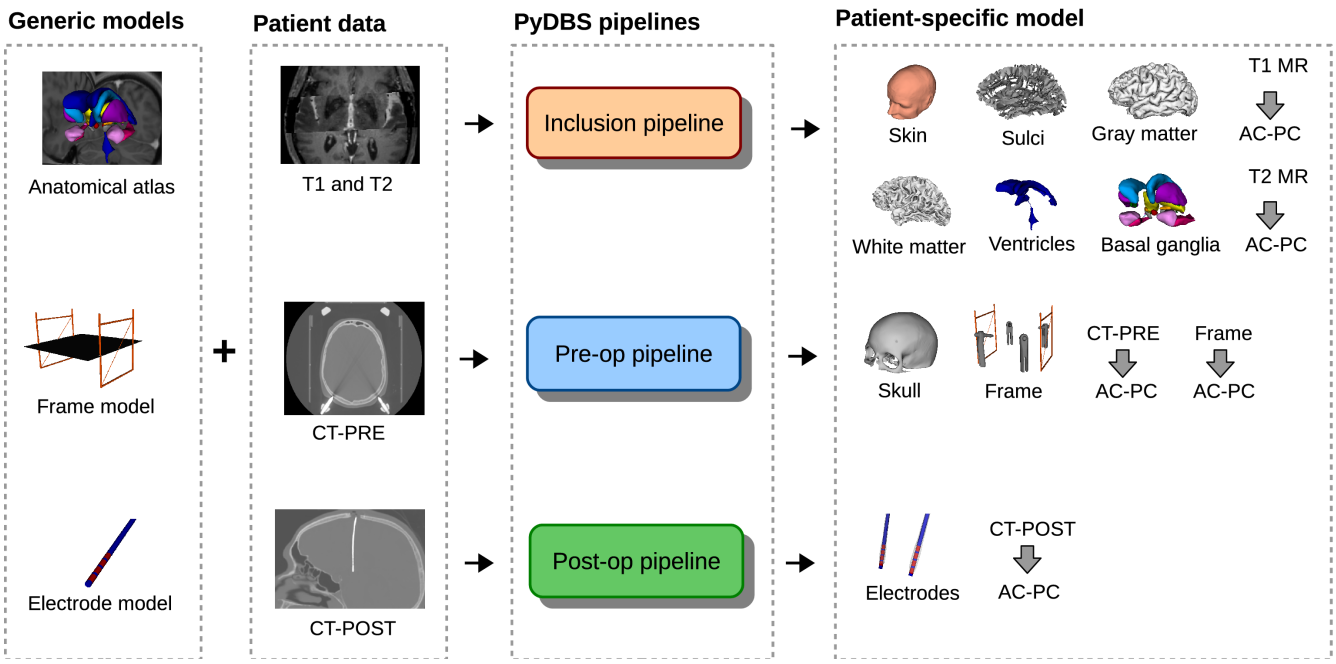


Fig. 3 PyDBS from a system perspective. PyDBS inputs are patient-specific data and generic models. The PyDBS output is a patient-specific model for DBS surgery.

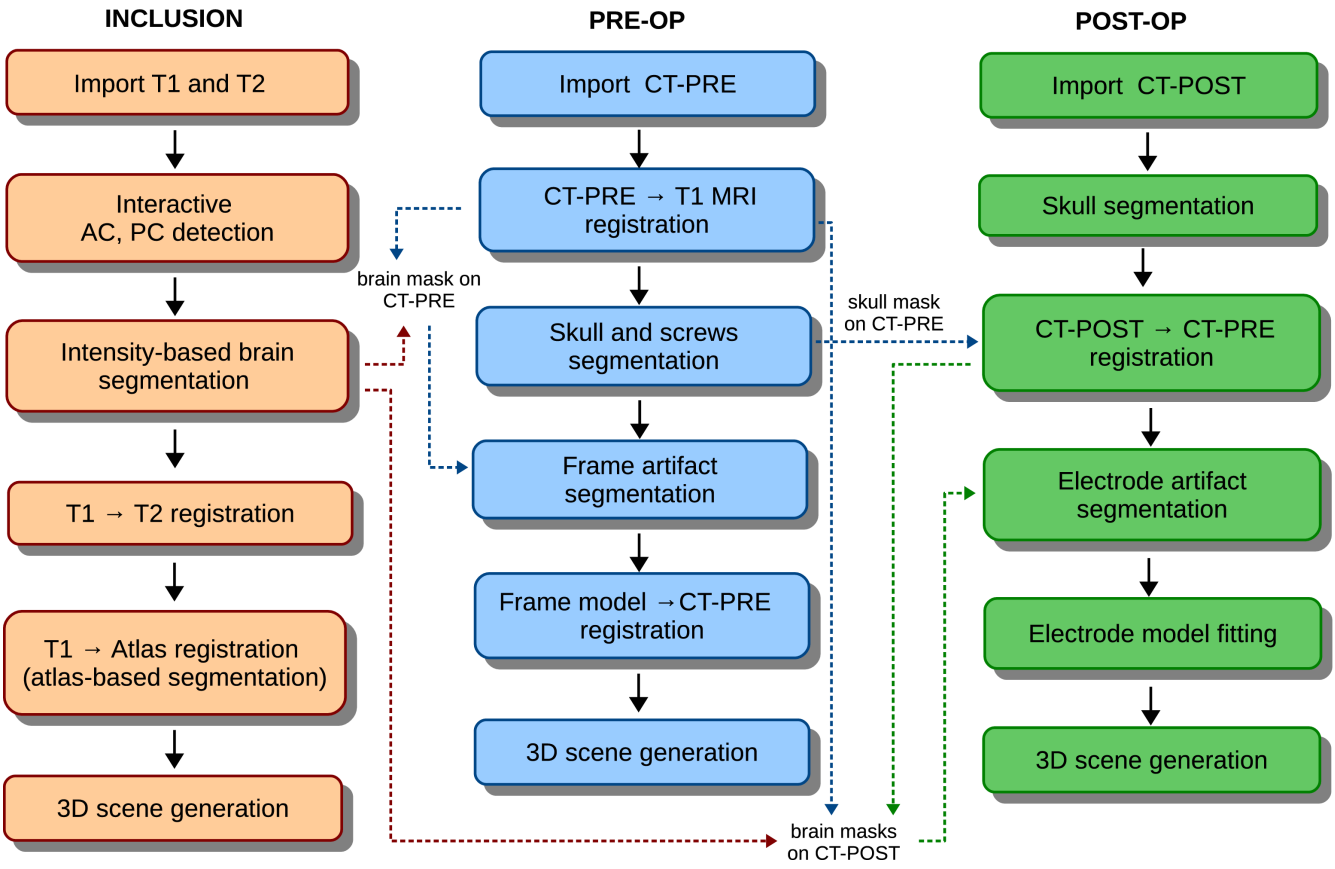


Fig. 4 Detailed diagram of the PyDBS pipelines: processing modules and dependencies. Dashed arrows depict inter-pipeline dependencies and indicate the specific result on which the target processing module depends. Within-pipeline dependencies are not shown.

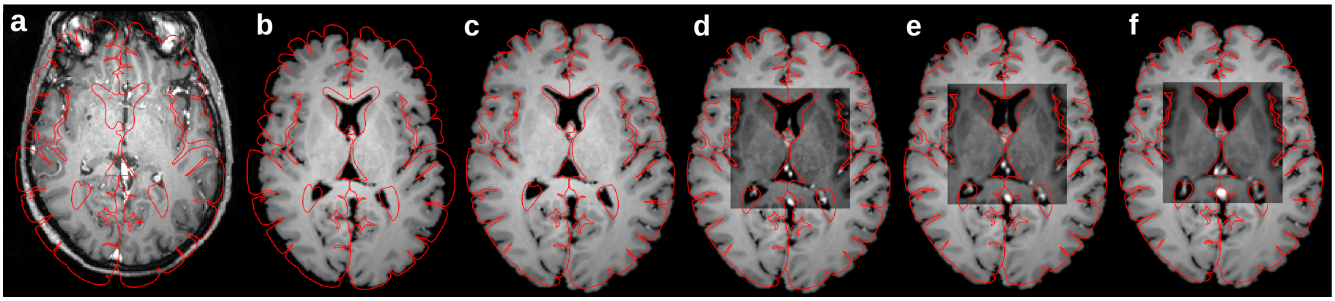


Fig. 5 Registration between the T1-weighted patient image and anatomical atlas. The contour of the atlas brain (red line) is superimposed onto an axial slice of the patient image. The darker square in d-f indicates the region of interest (ROI). Initially, the two volumes are not registered (a). The registration process is composed of five phases (b-f): rigid registration (b), affine registration (c), crop of the ROI (d), affine registration of the two ROIs (e), and non-linear registration of the two ROIs (f).

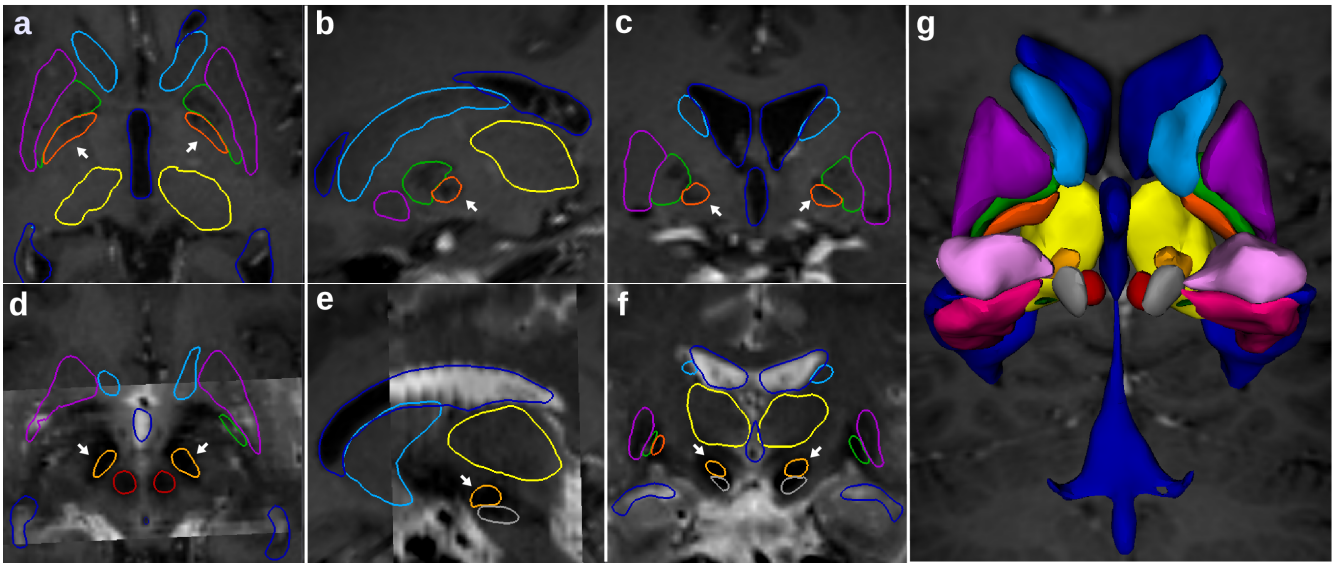


Fig. 6 Atlas-based segmentation of brain ventricles (blue) and basal ganglia: caudate (cyan), putamen (magenta), globus pallidus externum (green), globus pallidus internum (dark orange), thalamus (yellow), sub-thalamic nucleus (light orange), red nucleus (red), and substantia nigra (gray). Two common targets for DBS are the globus pallidus internum (arrows in a-c) and sub-thalamic nucleus (arrows in d-f). The first three columns show axial, sagittal, and coronal slices, respectively. In a-c only the T1-weighted MR volume is shown. In d-f the T2-weighted volume is overlaid onto the T1-weighted volume for improved visualization of deep structures. g) 3D view of brain ventricles and basal ganglia on the T1-weighted volume.

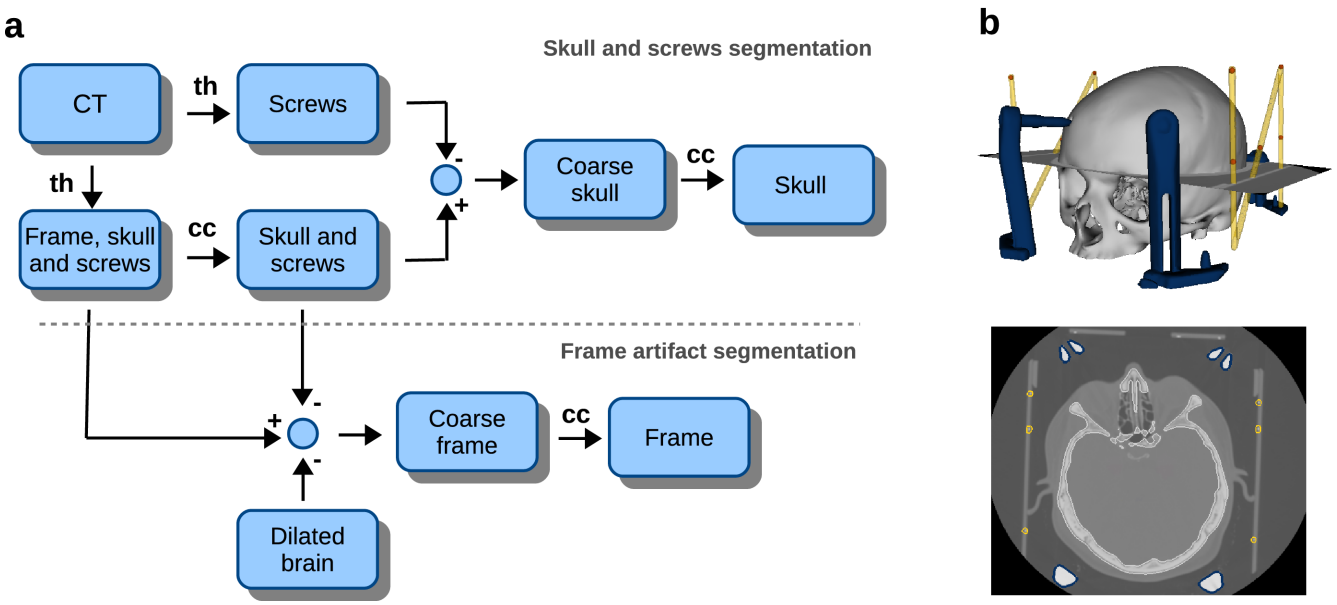


Fig. 7 Segmentation of skull, frame screws, and frame artifact. a) Flow diagram of the segmentation algorithm. All images are binary, except for the input CT volume. Thresholding and connected-component-labeling operations are abbreviated as “th” and “cc”, respectively. For segmenting skull and screws we retained only the largest connected component, while for the frame artifact we retained the six largest ones. This is on account of the frame artifact being composed of three separate parts (*i.e.*, one anterior part and two lateral parts), each of which may be split into another two parts if the frame is not entirely contained in the CT volume. When segmenting the frame artifact, the dilated brain mask is used to clean up small skull components. b) Segmentation results for one patient. The patient’s skull (gray), frame screws (blue), and frame artifact (yellow) are all displayed. The anterior part of the frame artifact is not shown, since it was not included in the frame model. The red spheres (five per side) indicate the automatically-detected landmark points used for the point-based registration between the frame model and the CT scan.

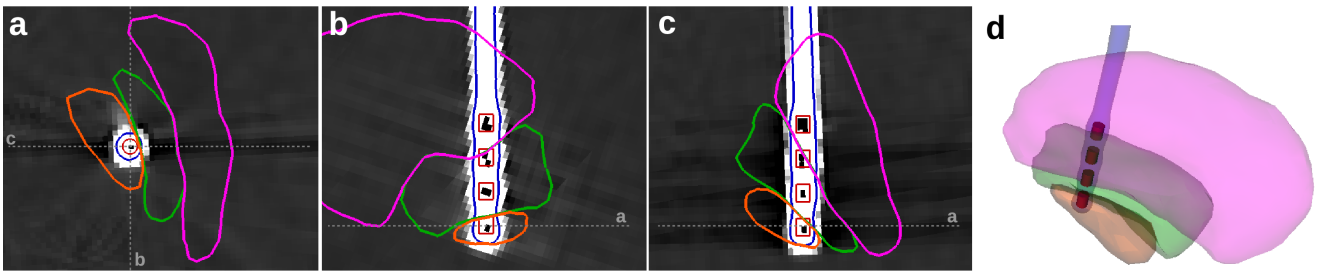


Fig. 8 Electrode segmentation and contact detection. Panels a-c depict three slices of one patient's postoperative CT image magnified onto the electrode artifact and overlaid with the contours of detected objects, namely the electrode artifact (blue), electrode contacts (red), and three surrounding anatomical structures: putamen (magenta), globus pallidus externum (green), and globus pallidus internum (orange). The slice in a) is perpendicular to the electrode axis and centered at the lowest contact. The slices in b) and c) are aligned to the electrode axis and perpendicular to each other. In d) a 3D view is displayed.

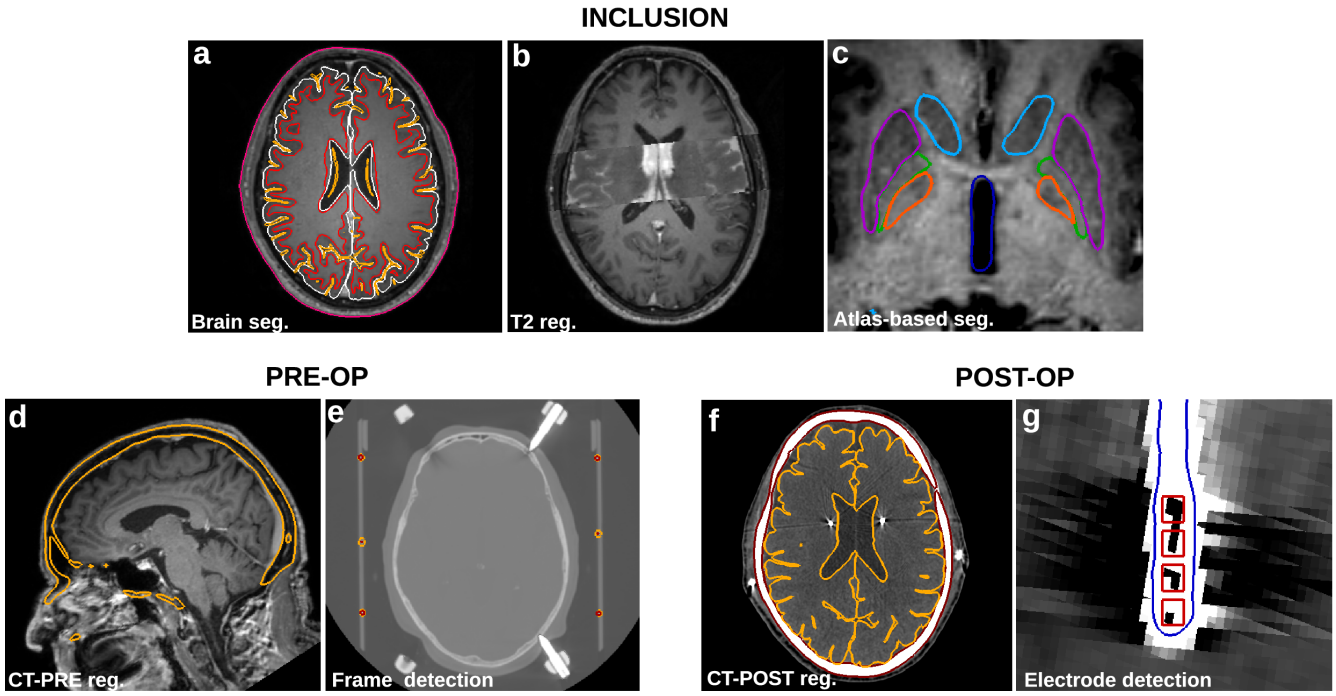


Fig. 9 Validation steps for the three pipelines. Each step is illustrated with a screen-shot of the 3D scene visualized by the observer. a) Skin (magenta), gray matter (white), white matter (red), and cortical sulci (yellow) are superimposed onto the T1-weighted volume. b) The T2-weighted volume is superimposed onto the T1-weighted volume. c) Brain ventricles (blue) and basal ganglia are superimposed onto the T1-weighted volume (basal-ganglia colors have been presented in Fig. 5). d) The skull segmented on the preoperative CT (yellow) is superimposed onto the T1-weighted volume. e) The frame artifact (yellow) and the registered frame model (red) are superimposed onto the preoperative CT. f) The skull segmented on the preoperative CT (red) along with the gray matter segmented on the T1-weighted volume (yellow) are superimposed onto the postoperative CT. g) The electrode artifact (blue) and electrode contacts (red) are superimposed onto the postoperative CT. Registration and segmentation are abbreviated as “reg.” and “seg.”, respectively.

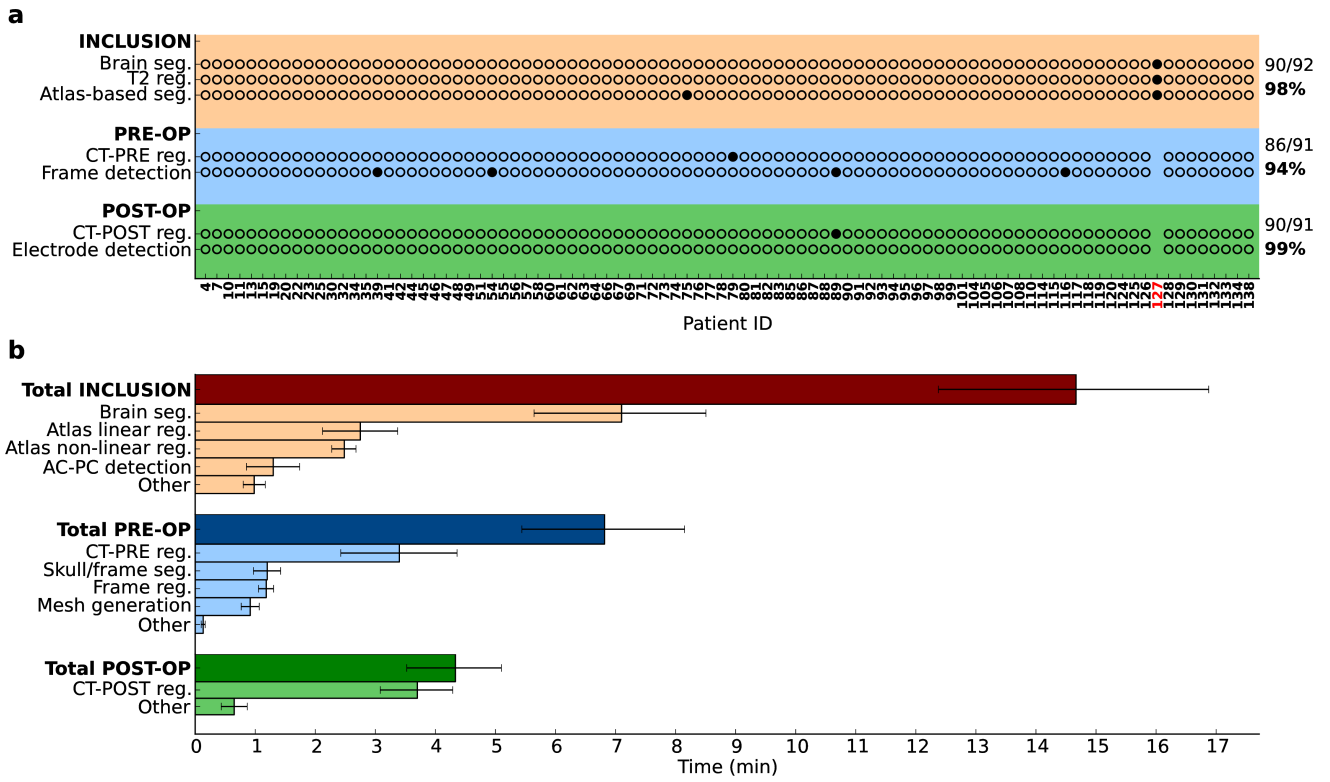


Fig. 10 Validation results. a) For each validation step and for each patient an empty (filled) circle indicates a satisfactory (unsatisfactory) result. For the three pipelines we reported the fraction and percentage of patients with entirely satisfactory validation steps. When one of the pipelines terminated with a runtime error, the corresponding patient code was marked in red. b) Pipeline processing times. Colored bars represent median values; error bars span from the lower to the upper quartile of the distribution. Patient #127, for whom runtime errors occurred, was excluded from time performance estimation.