Bonn-Aachen International Center for Information Technology
University of Bonn
Master Programme in Life Science Informatics
Master Thesis

Brain Tractography Registration with Nonrigid ICP

Mohammed Abdelgadir Awadelkareem Hassan

Supervisor: Prof. Thomas Schultz
Bonn-Aachen International Center for Information Technology
University of Bonn

Supervisor: Dr. Martin Vogt
Bonn-Aachen International Center for Information Technology
University of Bonn

Date: February 1, 2019

ACKNOWLEDGMENT

I would like to start by expressing my sincere gratitude to Prof. Dr. Thomas Schultz for providing me an excellent opportunity to work on this project under his supervision.

Also, I would like to thank Mohammad Khatami for his valuable support throughout the course of the project.

I would like to extend my sincere thanks to Dr. Martin Vogt for his expert guidance and valuable input.

Last but not least, I express my very profound gratitude to all my friends and family especially my parents for always being supportive and encouraging throughout the course of my studies.

I dedicate all my hard work to my wife for always believing in me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Registration of two or more threedimensional data sets is the process of transforming the data and bringing it together in order for the semantically corresponding components in the data sets to be aligned in the same coordinate system[1].

We show how to extend the ICP framework to nonrigid registration, while retaining the convergence properties of the original algorithm. The resulting optimal step nonrigid ICP framework allows the use of different regularisations, as long as they have an adjustable stiffness parameter. The registration loops over a series of decreasing stiffness weights, and incrementally deforms the template towards the target, recovering the whole range of global and local deformations. To find the optimal deformation for a given stiffness, optimal iterative closest point steps are used. Preliminary correspondences are estimated by a nearestpoint search. Then the optimal deformation of the template for these fixed correspondences and the active stiffness is calculated. Afterwards the process continues with new correspondences found by searching from the displaced template vertices. We present an algorithm using a locally affine regularisation which assigns an affine transformation to each vertex and minimises the difference in the transformation of neighbouring vertices. It is shown that for this regularisation the optimal deformation for fixed correspondences and fixed stiffness can be determined exactly and efficiently. The method succeeds for a wide range of initial conditions, and handles missing data robustly. It is compared qualitatively and quantitatively to other algorithms using synthetic examples and real world data[2].

1

INTRODUCTION

1 brain fiber pathways

THE HUMAN BRAIN is the central organ of the human nervous system, it is made up of two main components, gray matter and white matter. Scientists have learned a lot about gray and white matter and the two halves of the brain through autopsies and imaging techniques and by studying diseases or conditions associated with brain damage.

WHITE MATTER refers to areas of the central nervous system that are mainly made up of myelinated axons, also called tracts or fiber pathways [3]. Long thought to be passive tissue, white matter affects learning and brain functions, modulating the distribution of action potentials, acting as a relay and coordinating communication between different brain regions [4].
White matter is composed of bundles, which connect various gray matter areas of the brain to each other, and carry nerve impulses between neurons. Myelin acts as an insulator, which allows electrical signals to jump, rather than coursing through the axon, increasing the speed of transmission of all nerve signals [5].

ANTERIOR THALAMIC RADIATION (ATR) refers to fiber pathways that connect the anterior nuclear group of the thalamus and the midline nuclear group of the thalamus with the frontal lobe through the anterior thalamic peduncle, the anterior limb of the internal capsule and other parts of the cerebral white matter [6][7].
ATR abnormalities have a possible link with cognitive abnormalities and negative symptoms in schizophrenia[8].

CORPUS CALLOSUM (CC) is a wide, flat bundle of nerve fibers, located at the longitudinal fissure beneath the cortex, which acts a link between the two hemispheres of the brain and facilitates communication between them. The term corpus callosum means ’tough body’ in Latin. With approximately 200 - 250 million contralateral axonal projections to its credit, it is the largest among the various white matter structures in the central nervous system. The anterior portion of this structure is called ’**genu**’, while the posterior structure is called

′**splenium**′. In between its anterior and posterior portions, lies the ′truncus′ or its ′**body**′. Studies have revealed that the anterior of corpus callosum in left-handed people is 11 percent larger than that of right-handed people [9].

GENU OF THE CORPUS CALLOSUM (GENU) refers to the rostral most portion of the corpus callosum. It is bounded caudally by the body of the corpus callosum and ventrocaudally by the rostrum of the corpus callosum [6].

SPLENIUM OF THE CORPUS CALLOSUM (SPLENIUM) refers to the caudal most portion of the corpus callosum. It is bounded rostrally by the body of the corpus callosum [6].

BODY OF CORPUS CALLOSUM (TRUNCUS) refers to the portion of the corpus callosum located between the genu of the corpus callosum and the splenium of the corpus callosum. In a common parcellation, corpus callosum, it is divided into four parts: the rostral body of the corpus callosum, the anterior midbody of the corpus callosum, the posterior midbody of the corpus callosum and the isthmus of the corpus callosum [6].

CINGULUM (CING) refers to a fiber pathway that runs longitudinally in the cingulate white matter; it connects portions of the cingulate gyrus, the parietal lobe and the prefrontal cortex with the parahippocampal gyrus and adjacent structures of the temporal lobe. ″All connectios entering and exiting the cingulate gyrus pass through the cingulum bundle″. It is composed of the Cingulum ammonale and the Cingulum limitans [6].
Cingulum receives afferent fibers from the parts of the thalamus that are associated with the spinothalamic tract. This, in addition to the fact that the cingulum is a central structure in learning to correct mistakes, indicates that the cingulum is involved in appraisal of pain and reinforcement of behavior that reduces it [10].

CORTICOSPINAL TRACT (CST) refers to a fiber pathway from the cerebral cortex to the spinal cord. Its fibers originate from pyramidal neurons of the precentral gyrus, and on their way to the spinal cord, they pass through parts of the cerebral white matter (including the posterior limb of the internal capsule), the crus cerebri, the longitudinal pontine fibers, the pyramid of the medulla (where they are known as the pyramidal tract) and the pyramidal decussation. In the ducussation, some fibers cross to the other side of the brainstem to form the lateral corticospinal tract. Those fibers that do not cross split to form the anterolateral corticospinal tract and the anterior corticospinal tract [6].

fornix (fornix) The fornix (Latin, ”vault” or ”arch”) is a C-shaped bundle of fibers (axons) in the brain, and carries signals from the hippocampus to the hypothalamus. The fibres begin in the hippocampus on each side of the brain (where they are also known as the fimbria); the separate left and right sides are each called the crus of the fornix. The bundles of fibres come together in the midline of the brain, forming the body of the fornix. The inferior edge of the septum pellucidum (a membrane that separates the two lateral ventricles) is attached to the upper face of the fornix body. While its exact function and importance in the physiology of the brain are still not entirely clear, it has been demonstrated that surgical transection – the cutting of the fornix along its body – can cause memory loss [11].

inferior fronto-occipital fasciculus (ifo) The occipitofrontal fasciculus passes backward from the frontal lobe, along the lateral border of the caudate nucleus, and on the medial aspect of the corona radiata; its fibers radiate in a fan-like manner and pass into the occipital and temporal lobes lateral to the posterior and inferior cornua [9].

inferior longitudinal fasciculus (ilf) The inferior longitudinal fasciculus connects the temporal lobe and occipital lobe, running along the lateral walls of the inferior and posterior cornua of the lateral ventricle. The existence of this fasciculus independent from the occipitotemporal fasciculus has been questioned for the human being, such that it has been proposed that the term inferior longitudinal fasciculus be replaced by the term ”occipitotemporal projection” [9].

superior longitudinal fasciculus (slf) is a pair of long bi-directional bundles of neurons connecting the front and the back of the cerebrum. Each association fiber bundle is lateral to the centrum ovale of a cerebral hemisphere and connects the frontal, occipital, parietal, and temporal lobes. The neurons pass from the frontal lobe through the operculum to the posterior end of the lateral sulcus where numerous neurons radiate into the occipital lobe and other neurons turn downward and forward around the putamen and radiate to anterior portions of the temporal lobe [9].

ventral tegmental area (vta) is in the midbrain, situated adjacent to the substantia nigra. Although it contains several different types of neurons, it is primarily characterized by its dopaminergic neurons, which project from the VTA throughout the brain. The VTA is considered an integral part of a network of structures, together known as the reward system, that are involved in reinforcing behavior.

1.1 *From Thesis*

The surface registration of two or more three-dimensional data sets by definition is the process of transforming the data and bringing it together in order for the semantically corresponding components in the data sets to be aligned. The scanning of three-dimensional object often gives occasion for the application of surface registration, since perfect conditions are rarely the case as the data sets are produced by scanning objects from different view points and assigning them in different coordinate systems. Those scans produce noisy and incomplete surfaces that should later be combined and the missing data should be reconstructed, for which surface registration is used, thus spreading its importance in a variety of fields as computer graphics and computer vision. Surface registration can be divided in two types: rigid and nonrigid, depending on the types of transformations that need to be done on the data in order for a proper alignment to be achieved. In contrast to the rigid surface registration, the nonrigid allows not only translations, rotations and reflections to be used, but also deformations, which leads to much more possible registration results that can be considered. This master thesis will examine that case of nonrigid surface registration to both mesh and point cloud data. Those two types of problems can be reduced to one when translating both types to a connected undirected graph, using the triangulation of the mesh for the edges and a KNN algorithm for constructing a graph from the point cloud. The thesis is focused on an already existing algorithm, developed and described in [1] by Brian Amberg, Sami Romdhani and Thomas Vetter. It works to find a registration of a template surface over a target one (Figure 1.1), using an iterative closest point method and imposing constraints on the deformation that is occurring on each step, defining a cost function that needs to be minimized by solving an equation. The iterations of the algorithm are guided by a changing stiffness parameter that controls the nature of the transformations that are applied, starting from more global ones and reducing to local ones by the end of the run. The goal of the master thesis is to propose an optimization of the given algorithm, alternating both its steps and cost function, adapting it in a way so that it works much faster for meshes and point clouds with a large number of points. Described as patch-based, the optimization works by clustering the template graph and performing deformations on the patches instead on the vertices, which means that clusters are moving together, decreasing the computation time of solving the equation due to fewer number of elements being considered. The vertices get transformed depending on the way the cluster they are in is moving, as well as its neighboring clusters, which is achieved using soft membership weights. The algorithm starts with a big number of patches that get divided into smaller ones on each step, this way the new algorithm mimics the old one by applying global deformations at first and then moving the vertices locally to the target. The proposed algorithm is a nonrigid optimal step ICP algorithm as the one in [1], with a different convergence criterion. The idea of the proposed method went through multiple iterations itself before reaching the final definition that is depicted in this text. Dierent method were tested for the clustering until the best one was found and applied. The role of the clustering also grew bigger when the initial idea, to apply the original algorithm on a one-time clustered template, proved unsatisfactory in practice. The thesis consists of six Chapters: Introduction, Related Works, Method, Implementation, Results and the Conclusion. In the next Chapter Related Works the state of the art is reviewed. The main theoretical Chapter Method starts with briefly defining the setup of the algorithm, describing the preliminary steps and the essence of the optimization - the clustering. After this the original cost function from [1] is reviewed and its alternations are explained, finishing with the whole algorithm idea step by step. In the Chapter Implementation we talk briefly about how the implemented tool works, what libraries are used for it and how the output looks. The Chapter Results is where the experiments are described and both algorithms, the original one from [1] and the optimized one from this thesis, are compared in terms of time and registration. The thesis ends with a Conclusion Chapter, where everything is summarized.

2 REGISTRATION

The 3D-3D registration problem (also known as 3D-3D alignment, 3D absolute orientation, 3D pose) is one of the outstanding and very basic problems in computer vision. In this problem, two sets of 3D points are given and the task is to optimally align these two sets of points by estimating a best transformation between them. Due to its fundamental importance, it arises as a subtask in many different applications (e.g., object recognition, tracking, range data fusion, graphics, medical image alignment, robotics and structural bioinformatics etc [14].

2.1 *Iterative closest point (ICP)*

ICP is an algorithm employed to minimize the distance between two clouds of points. In ICP (in our case) one point cloud (vertex cloud), the reference, or target, is kept fixed, while the other one, the source, is transformed to best match the reference. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize a distance from the source to the reference point cloud. ICP is one of the widely used algorithms in aligning three dimensional models given an initial guess of the rigid body transformation required [15].

2.2 *PCA Transformation*

2.3 *Clustering*

2.4 *Distance Functions*

KD TREE

POINT CLOUD BASED

2

INTRODUCTION

1 BRAIN FIBER PATHWAYS

THE HUMAN BRAIN is the central organ of the human nervous system, it is made up of two main components, gray matter and white matter. Scientists have learned a lot about gray and white matter and the two halves of the brain through autopsies and imaging techniques like MRIs, and by studying diseases or conditions associated with brain damage.

WHITE MATTER refers to areas of the central nervous system that are mainly made up of myelinated axons, also called tracts or fiber pathways [3]. Long thought to be passive tissue, white matter affects learning and brain functions, modulating the distribution of action potentials, acting as a relay and coordinating communication between different brain regions [4].
White matter is composed of bundles, which connect various gray matter areas of the brain to each other, and carry nerve impulses between neurons. Myelin acts as an insulator, which allows electrical signals to jump, rather than coursing through the axon, increasing the speed of transmission of all nerve signals [5].

ANTERIOR THALAMIC RADIATION (ATR) refers to fiber pathways that connect the anterior nuclear group of the thalamus and the midline nuclear group of the thalamus with the frontal lobe through the anterior thalamic peduncle, the anterior limb of the internal capsule and other parts of the cerebral white matter [6][7].
ATR abnormalities have a possible link with cognitive abnormalities and negative symptoms in schizophrenia[8].

CORPUS CALLOSUM (CC) is a wide, flat bundle of nerve fibers, located at the longitudinal fissure beneath the cortex, which acts a link between the two hemispheres of the brain and facilitates communication between them. The term corpus callosum means 'tough body' in Latin. With approximately 200 - 250 million contralateral axonal projections to its credit, it is the largest among the various white matter structures in the central nervous system. The anterior portion of this structure is called '**genu**', while the posterior structure is called

′**splenium**′. In between its anterior and posterior portions, lies the ′truncus′ or its ′**body**′. Studies have revealed that the anterior of corpus callosum in left-handed people is 11 percent larger than that of right-handed people [9].
Genu of the Corpus Callosum (genu) refers to the rostral most portion of the corpus callosum. It is bounded caudally by the body of the corpus callosum and ventrocaudally by the rostrum of the corpus callosum [6].
Splenium of the Corpus Callosum (splenium) refers to the caudal most portion of the corpus callosum. It is bounded rostrally by the body of the corpus callosum [6].
Body of Corpus Callosum (truncus) refers to the portion of the corpus callosum located between the genu of the corpus callosum and the splenium of the corpus callosum. In a common parcellation, corpus callosum, it is divided into four parts: the rostral body of the corpus callosum, the anterior midbody of the corpus callosum, the posterior midbody of the corpus callosum and the isthmus of the corpus callosum [6].

CINGULUM (CING) refers to a fiber pathway that runs longitudinally in the cingulate white matter; it connects portions of the cingulate gyrus, the parietal lobe and the prefrontal cortex with the parahippocampal gyrus and adjacent structures of the temporal lobe. ″All connectios entering and exiting the cingulate gyrus pass through the cingulum bundle″. It is composed of the Cingulum ammonale and the Cingulum limitans [6].
Cingulum receives afferent fibers from the parts of the thalamus that are associated with the spinothalamic tract. This, in addition to the fact that the cingulum is a central structure in learning to correct mistakes, indicates that the cingulum is involved in appraisal of pain and reinforcement of behavior that reduces it [10].

CORTICOSPINAL TRACT (CST) refers to a fiber pathway from the cerebral cortex to the spinal cord. Its fibers originate from pyramidal neurons of the precentral gyrus, and on their way to the spinal cord, they pass through parts of the cerebral white matter (including the posterior limb of the internal capsule), the crus cerebri, the longitudinal pontine fibers, the pyramid of the medulla (where they are known as the pyramidal tract) and the pyramidal decussation. In the ducussation, some fibers cross to the other side of the brainstem to form the lateral corticospinal tract. Those fibers that do not cross split to form the anterolateral corticospinal tract and the anterior corticospinal tract [6].

FORNIX (FORNIX) The fornix (Latin, ″vault″ or ″arch″) is a C-shaped bundle of fibers (axons) in the brain, and carries signals from the hippocampus to the hypothalamus. The fibres begin in the hippocampus on each side of the brain (where they are also known as the fimbria); the separate left and right sides are each called the crus of the fornix. The bundles of fibres come together in the midline of the brain, forming the body of the fornix. The inferior edge of the septum pellucidum (a membrane that separates the two lateral ventricles) is attached to the upper face of the fornix body. While its exact function and importance in the physiology of the brain are still not entirely clear, it has been demonstrated that surgical transection – the cutting of the fornix along its body – can cause memory loss [11].

INFERIOR FRONTO-OCCIPITAL FASCICULUS (IFO) The occipitofrontal fasciculus passes backward from the frontal lobe, along the lateral border of the caudate nucleus, and on the medial aspect of the corona radiata; its fibers radiate in a fan-like manner and pass into the occipital and temporal lobes lateral to the posterior and inferior cornua [9].

INFERIOR LONGITUDINAL FASCICULUS (ILF) The inferior longitudinal fasciculus connects the temporal lobe and occipital lobe, running along the lateral walls of the inferior and posterior cornua of the lateral ventricle. The existence of this fasciculus independent from the occipitotemporal fasciculus has been questioned for the human being, such that it has been proposed that the term inferior longitudinal fasciculus be replaced by the term ”occipitotemporal projection” [9].

SUPERIOR LONGITUDINAL FASCICULUS (SLF) is a pair of long bi-directional bundles of neurons connecting the front and the back of the cerebrum. Each association fiber bundle is lateral to the centrum ovale of a cerebral hemisphere and connects the frontal, occipital, parietal, and temporal lobes. The neurons pass from the frontal lobe through the operculum to the posterior end of the lateral sulcus where numerous neurons radiate into the occipital lobe and other neurons turn downward and forward around the putamen and radiate to anterior portions of the temporal lobe [9].

VENTRAL TEGMENTAL AREA (VTA) is in the midbrain, situated adjacent to the substantia nigra. Although it contains several different types of neurons, it is primarily characterized by its dopaminergic neurons, which project from the VTA throughout the brain. The VTA is considered an integral part of a network of structures, together known as the reward system, that are involved in reinforcing behavior.

1.1 *From Thesis*

The surface registration of two or more three-dimensional data sets by definition is the process of transforming the data and bringing it together in order for the semantically corresponding components in the data sets to be aligned. The scanning of three-dimensional object often gives occasion for the application of surface registration, since perfect conditions are rarely the case as the data sets are produced by scanning objects from different view points and assigning them in different coordinate systems. Those scans produce noisy and incomplete surfaces that should later be combined and the missing data should be reconstructed, for which surface registration is used, thus spreading its importance in a variety of fields as computer graphics and computer vision. Surface registration can be divided in two types: rigid and nonrigid, depending on the types of transformations that need to be done on the data in order for a proper alignment to be achieved. In contrast to the rigid surface registration, the nonrigid allows not only translations, rotations and reflections to be used, but also deformations, which leads to much more possible registration results that can be considered. This master thesis will examine that case of nonrigid surface registration to both mesh and point cloud data. Those two types of problems can be reduced to one when translating both types to a connected undirected graph, using the triangulation of the mesh for the edges and a KNN algorithm for constructing a graph from the point cloud. The thesis is focused on an already existing algorithm, developed and described in [1] by Brian Amberg, Sami Romdhani and Thomas Vetter. It works to find a registration of a template surface over a target one (Figure 1.1), using an iterative closest point method and imposing constraints on the deformation that is occurring on each step, defining a cost function that needs to be minimized by solving an equation. The iterations of the algorithm are guided by a changing stiffness parameter that controls the nature of the transformations that are applied, starting from more global ones and reducing to local ones by the end of the run. The goal of the master thesis is to propose an optimization of the given algorithm, alternating both its steps and cost function, adapting it in a way so that it works much faster for meshes and point clouds with a large number of points. Described as patch-based, the optimization works by clustering the template graph and performing deformations on the patches instead on the vertices, which means that clusters are moving together, decreasing the computation time of solving the equation due to fewer number of elements being considered. The vertices get transformed depending on the way the cluster they are in is moving, as well as its neighboring clusters, which is achieved using soft membership weights. The algorithm starts with a big number of patches that get divided into smaller ones on each step, this way the new algorithm mimics the old one by applying global deformations at first and then moving the vertices locally to the target. The proposed algorithm is a nonrigid optimal step ICP algorithm as the one in [1], with a different convergence criterion. The idea of the proposed method went through multiple iterations itself before reaching the final definition that is depicted in this text. Different method were tested for the clustering until the best one was found and applied. The role of the clustering also grew bigger when the initial idea, to apply the original algorithm on a one-time clustered template, proved unsatisfactory in practice. The thesis consists of six Chapters: Introduction, Related Works, Method, Implementation, Results and the Conclusion. In the next Chapter Related Works the state of the art is reviewed. The main theoretical Chapter Method starts with briefly defining the setup of the algorithm, describing the preliminary steps and the essence of the optimization - the clustering. After this the original cost function from [1] is reviewed and its alternations are explained, finishing with the whole algorithm idea step by step. In the Chapter Implementation we talk briefly about how the implemented tool works, what libraries are used for it and how the output looks. The Chapter Results is where the experiments are described and both algorithms, the original one from [1] and the optimized one from this thesis, are compared in terms of time and registration. The thesis ends with a Conclusion Chapter, where everything is summarized.

2 DIFFUSION MRI AND TRACTOGRAPHY

Key technological developments in MRI during the late 1980s and early 1990s engendered some powerful methods for measuring structure and function non-invasively. A direct consequence of these developments is diffusion MRI tractography; to date, it is the only available tool to estimate the trajectories of brain white matter in vivo. Diffusion MRI is sensitive to the random thermal motion of water molecules, which is hindered by tissue microstructure. When this microstructure is organized, such as in white matter, water diffusion is anisotropic, in that diffusion is less hindered parallel than perpendicular to axons. Thus, by measuring the orientational dependence of water diffusion, we can estimate axonal orientations in vivo. Tractography algorithms use local information on orientation to infer long-range connections and allow us to perform in vivo ’virtual dissection’ of white matter bundles. By repeating tractography for many brain locations, we can construct a connectome, a comprehensive map of macroscopic connections as estimated by diffusion MRI [12]. The directional information obtained in each voxel is used to generate virtual, three-dimensional white matter maps[13].

3 REGISTRATION

The 3D-3D registration problem (also known as 3D-3D alignment, 3D absolute orientation, 3D pose) is one of the outstanding and very basic problems in computer vision. In this problem, two sets of 3D points are given and the task is to optimally align these two sets of points by estimating a best transformation between them. Due to its fundamental importance, it arises as a subtask in many different applications (e.g., object recognition, tracking, range data fusion, graphics, medical image alignment, robotics and structural bioinformatics etc [14].

3.1 *Iterative closest point (ICP)*

ICP is an algorithm employed to minimize the distance between two clouds of points. In ICP (in our case) one point cloud (vertex cloud), the reference, or target, is kept fixed, while the other one, the source, is transformed to best match the reference. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize a distance from the source to the reference point cloud. ICP is one of the widely used algorithms in aligning three dimensional models given an initial guess of the rigid body transformation required [15].

3.2 *PCA Transformation*

3.3 *Clustering*

3.4 *Distance Functions*

KD TREE

POINT CLOUD BASED

3

METHOD

The main subject of this master thesis is a computationally effective surface registration algorithm that produces satisfactory results when faced with the problem of the registration between meshes or cloud point data with a large number of points. In this Chapter the idea of this method is described theoretically. It is based heavily on the method presented in [1] as it is built as a direct extension of an already existing algorithm. The main differences will be addressed here in more detail. We define a surface registration the process of aligning two 3D surfaces, a target and a template one, so that they meet when placed at the same position and orientation. The result should be a solution of the corresponding problem, namely a mapping that lists the positions of the semantically corresponding points between the two surfaces. Because this is a dense registration method, correspondences for all points are looked for, not only for selected feature points, as in sparse methods. This is done by deforming the template surface, locally moving it closer and closer to the target, in order to wrap it onto it. As described in [1] the template surface is denoted by S = (V,E), where V is a set of n vertices and E is a set of m edges and the target surface is T. In the sense of this algorithm the term registration means finding parameters X that describe a set of displaced vertices V (X). As shown on Figure 3.1, the template surface S is deformed onto the target surface T by applying the locally affine transformations Xi. These transformations are guided by the previously found correspondences, in our case the closest points in the target for the template points. Those points are denoted by ui for each displaced vertex Xivi from the template. In our case, because the transformations Xi are affine and we are using homogeneous coordinates in the 3D Euclidean space, the parameters Xi are a 3 4 matrices for each vertex in the template. The unknown Xi transformations are organized in the following 4n 3 matrix X: X = [X1, . . . ,Xn]T (3.1) Having this defined, this transformations need to be regularized, so that the vertices that are originally close to each other will stay that way after each step of the deformation has occurred. This is done by defining constraints and using them in finding the transformations. A cost function is build and minimized by setting its derivative to zero and solving the resulting equation. The result is the needed matrix X. In the subsection Setup we describe how the input data is presented and the rest of the needed definitions are cleared out. It also covers the topic of different kinds of input data, as the tool can be applied on both triangle meshes and point cloud data. The the second subsection two preliminary steps are introduced as well as the nature of their necessity. The subsection Clustering is about the main part of the optimization, its idea, the stages it went through and how is it applied. The role that the clustering played in this thesis and the developing of the tool went from optimization of the solving part to the main instrument of the regularization. The subsection Assigning Soft Membership Weights briefly defines the essential topic of the influence the clusters have on the vertices in their area. The second to last subsection and its subsections describe how the equation changed due to the optimization and what is its final look. They also address questions like how the correspondences are found, how the method works for unconnected graphs, and how are the found transformations applied on the template graph using soft membership weights. The last subsection Nonrigid optimal step ICP algorithm shows how exactly the base algorithm works after the alternation.

1 SETUP

The method as well as the developed tool will recognize two kinds of input: triangle mesh data and point cloud data. In both cases the template mesh needs to be adjusted to the previously defined representation of the input data, namely a graph S = (V,E), where V and E are sets of vertices and edges respectively. The only important thing about the representation of the target mesh T is to allow quick finding of the closest point ui in the target mesh for any given point of the template mesh. Back to the template surface representation, the case of the triangle mesh is clear, the edges set will be constructed based on the triangulation, each triangle will produce 3 edges in the set with no duplications, meaning that if an edge is already in the set and a currently observed triangle contains it, it will not be added a second time. In the case of the point cloud data a dierent approach is needed, because there is no available information in the input for the connectivity of the graph. An efficient algorithm for constructing a K-nearest neighbors graph is used. The only required input for it is the number k of the nearest neighbors that are to be found. In the cases when k is too small and the resulted graph is unconnected the method will still perform without a problem if some conditions are fulfilled, as it will be explained in further details later in this Chapter, the subsection Clustering. The same applies for the triangle mesh input, it can contain unconnected groups of vertices without causing a problem for the method. From here on we will address the input data as target and template graphs, independent of their origin. Another important thing to be noted about the input graph is that each edge has a weight and in our case this weight is the Euclidean distance between the two vertices that are connected with that edge. Usually this information doesn't come with the triangle meshes or the point cloud, so it is computed after the scaling step. These weights are also updated after each transformation step, because the vertices change their position and the Euclidean distance between them also varies. This will be addressed again in the subsection Nonrigid optimal step ICP algorithm, where each step will be noted again in a sequence.

2 PREPARATION STEPS

Two preparations steps are done on both the template and the target graphs before any clustering or building of equations has started: principal components analysis and scaling to the [-1, 1]3 cube. The first is necessary so that the two graphs are aligned as much as possible with one another, which will contribute to finding adequate corresponding points to the template vertices in the target graph, as shown in Figure 3.2. The tutorial cited in [36] was used for the purpose of applying the Principal component analysis (PCA) to the target graph and after this is done to apply it the template as well, but according to the orientation of the already aligned target. The steps are the following: 1. Collect the vertices of the graph and their coordinates in a matrix Data, where each vertex is on a row and its coordinates are on the columns. Because we are considering 3D data this matrix will have the size of —V —3. 2. Subtract the mean (X, Y and Z accordingly) from each data dimension to get a new matrix DataAdjust. 3. Calculate the covariance matrix C = 2 4 cov(x, x) cov(x, y) cov(x, z) cov(y, x) cov(y, y) cov(y, z) cov(z, x) cov(z, y) cov(z, z) 3 5 where cov(X, Y) = Pn i=1 (Xi - X)(Yi - Y) (n - 1) 4. Calculate the eigenvectors and eigenvalues of the covariance matrix. 5. Build a feature vector (a matrix of eigenvectors with size of 3 3) according to a chosen orientation FeatureV ector = (eig1 eig2 eig3) where the eigenvectors are ordered by their eigenvalues from highest to lowest and depending on the signs of the coordinates of the eigenvectors there are 8 orientations available: 1 (x1, y1, z1) (x2, y2, z2) (x3, y3, z3) 2 (-x1, y1, z1) (-x2, y2, z2) (-x3, y3, z3) 3 (x1, -y1, z1) (x2, -y2, z2) (x3, -y3, z3) 4 (x1, y1, -z1) (x2, y2, -z2) (x3, y3, -z3) 5 (-x1, -y1, z1) (-x2, -y2, z2) (-x3, -y3, z3) 6 (-x1, y1, -z1) (-x2, y2, -z2) (-x3, y3, -z3) 7 (x1, -y1, -z1) (x2, -y2, -z2) (x3, -y3, -z3) 8 (-x1, -y1, -z1) (-x2, -y2, -z2) (-x3, -y3, -z3) 6. Compute the final data to work with FinalData = FeatureV ectorT DataAdjustT This gives us our old data in a matrix 3—V —, but oriented regarding its eigenvectors instead of the usual axes When those steps are completed for the target graph, they are also run on the template graph 8 times for each possible orientation and it is checked which one gives the closest alignment of the template graph to the target one. Once the right orientation is known, the PCA is also run on the template and the scaling step can begin. It is as simple as finding the maximum absolute value of a coordinate of a vertex in both the target and the template graph and then dividing all the coordinates by this maximum. It is explained in [1] that the scaling of the data to the [-1, 1]3 cube is done in order to simplify the stiness part of the equation. While this will be discussed also in the subsection about the The Original Cost Function, it is enough to say here that if the parameter ! is used to weight the differences between the rotational and skew parts of the deformation against the translational part, and it is determined based on the graph's units, then there will be no need of taking it into consideration when the graph is scaled in advance and ! will be set to 1. After the graphs are aligned and scaled the registration can begin. One of the biggest parts in it is the clustering of the data. Why it is done and how is explained in the next subsection of this Chapter.

3 FINDING THE AFFINE TRANSFORMATION

Now that we have the cluster graph and we want to perform a registration on it, it is time to go back to [1] and look more into the details of constructing the cost function, the solving of the equation that works as regularization and how to use the result for the registration, all in the current case of the clustering. This subsection is divided in three, the first part explains how the things are done in [1], the second part shows how the cost function differs from the original when using clusters and the third part covers the topic of the transformation of the template graph once the parametrization of the mapping (3.1) is found.

3.1 *Cost Function*

In this subsection the template graph is the original one and no talks of clustering will be taking place. All of the optimization topics will be covered in the next subsection, while the goal here is to show the original cost function and the way it is defined in [1]. Also in this subsection we say that n = —V — and e = —E— for simplicity. This cost function consists of three terms: distance term, stiffness term and landmark term. Those terms express different aspects of what we should achieve when minimizing the cost function. The distance term describes that the distance between the transformed template graph and the target graph needs to be small. The stiffness term is used for the regularization, so that neighbor vertices move together and there won't be any huge deformations that contradict with the overall connectivity. The landmark term is used for initialization, assigning starting correspondences between the two graphs. Lets look all those terms in detail. In [1] the cost function is defined in the following way: and this is the way it is defined in this thesis as well with few differences that will be pointed out. The first term is the distance term and it is defined in the following way: Ed(X) = X vi2V widist2(T,Xivi) (3.7) In this representation the template vertices are noted with vi and are also given in homogeneous coordinates vi = [x, y, z, 1]T . This is also the reason why Xi are 3 4 matrices in (3.1). The distance between the point in the template v and it closest point in the target is denoted as dist(T, v) and wi is a weight that is one, if a correspondence for this vertex is found in the target, and zero if not. The correspondences are used so that each vertex in the template moves in the direction of the vertex in the target that corresponds to it. Some vertices have no correspondences, but even then they move along their neighbors because of the stiffness term. In [1] is said that in order to detect which vertices have no correspondence three tests are used, defined in the following way: A correspondence (Xivi, ui) is dropped if 1. ui lies on a border of the target mesh, 2. the angle between the normals of the meshes at Xivi and ui is larger than a fixed threshold or 3. the line segment Xivi to ui intersects the deformed template. In our implementation we simplify things and say that a correspondence is not found if the distance between the point and its closest target correspondence is greater than a value that is set in the beginning as a maximum correspondence distance. As noted before, it is important for the representation of the target graph to be convenient enough for an efficient search for correspondences. It will be explained in the Chapter Implementation how this can be approached. The second term, the stiness term that has the job to regularize the deformation, is defined in the following way in [1]: Es(X) = X i,j2E k(Xi - Xj)Gk2 F (3.8) where G is the diagonal matrix diag(1, 1, 1,!). As mentioned before, ! is set to one, because the data was scaled into the [-1, 1]3 cube, which is described in the Preparation Steps subsection. The kkF is the Frobenius norm defined for a matrix A with size m n like this: kAkF = vuut Xm i=1 Xn j=1 —aij —2 With this term the dierence between the transformations of two vertices with an edge between them is minimized, keeping neighbor vertices together as discussed. The stiness weight that is also a part of (3.6) changes on each step of the ICP algorithm and controls the transformation. It starts with a high value that keeps big chunks of the mesh together so that more global transformations can occur, and each step it gets lower value, so that the small details can fall into place as well. The starting value, the step with which it change each step and the minimum of the is user defined and it depends on those parameters how many solver steps the algorithm will produce and how accurate the end result will be. As explained in [1] and also as will be noted again in the Applying the Transformations section, twelve parameters are used to describe the displacement of the vertex, which are the matrices Xi with size 3 4 in (3.8). This is done in order to construct the cost function as a quadratic function that can be minimized directly. The last term is the landmark term: El(X) = X (vi,l)2L k(Xivi - l)k2 (3.9) where L is a set of given landmarks (vi1, l1), . . . , (vil, ll). A landmark (vi, l) is a mapping from the template graph to the target one and L has a role to (3.6) is used to control this landmark term, having a strong influence in the first steps of the ICP algorithm and fading gradually, so that any noise from the initial landmarks is avoided. In this implementation the landmark term is omitted, because the registration can very well give good results without it. Also after the PCA is done as a preparation step and both graphs are aligned the need for landmarks is even less notable. In order to transform the cost function (3.6) in a way that can be minimized, we need to assume that the correspondences (vi, ui) are fixed. This way the part of the distance term (3.7) dist2(T,Xivi) (3.10) can become kXivi - uik2 (3.11) and the distance term can be rewritten. The entire cost function can be presented using matrices and it can be solved directly. With the changed distance term and the the other two terms also rewritten with matrices we have ¯E that can be minimized enough times so that a local minimum of (3.6) is found. Now let's look at the changed distance term in more detail. When applying the change from (3.10) to (3.11) on (3.7) we have: ¯E d(X) = X vi2V wikXivi - uik2 = = ******* (W XOR I3) 2 64 X1 . . . Xn 3 75 2 64 v1 ... vn 3 75 - 2 64 u1 ... un 3 75 ! ******* 2 (3.13) where W is diag(w1, . . . ,wn), the matrix containing the correspondences weights on its diagonal, I3 is the 3 3 identity matrix and the Kronecker product is denoted by XOR. By definition the Kronecker product of twomatrices is the following matrix: A XOR B = 2 64 a11B .. .a1nB ... . am1B .. .amnB 3 75 This is done so that the W XOR I3 part will have size of 3n 3n, while the rest of the matrices have sizes of 3n4n, 4n1 and 3n1 accordingly, the weight wi will still influence the Xivi - ui part and the weights can be put inside of the norm. The equation (3.13) still needs to be rearranged into its canonical form in order to be clearly visible what the unknowns are and for them to be arranged in a compact matrix X as defined in the beginning in (3.1). This is done by swapping the positions of the transformation matrices Xi and the vertices vi and by defining the following matrix D: D = 2 6664 vT 1 vT 2 . . . vT n 3 7775 (3.14) In the end we get the final simplified form of the distance term: ¯E d(X) = kW(DX - U)k2 F (3.15) where the matrix W has size nn, thematrix D had size n4n, because the vertices are presented with homogeneous coordinates, as it was noted earlier, and the matrix X is as before with size 4n3. The matrix U still holds the correspondences ui and is with size n 3. Following similar simplifications as the distance term and the new notations, we get the stiness term (3.8) in its new matrix form: Es(X) = k(M XOR G)Xk2 F (3.16) where M is the node-arc incidence matrix for the template graph. It is defined so that if edge r connects the vertices (i, j) the nonzero entries of M in row r are Mri = -1 and Mrj = 1 as shown in Figure 3.6. It is important to have in mind here that the edges and vertices of the mesh are numbered and its edges are directed from the lower numbered vertex to the higher numbered. The matrix M has size e n and so M XOR G has size of 4e 4n. At last the landmark (3.9) term also gets its makeover: El(X) = kDLDX - ULk2 F (3.17) where DL is the matrix D from (3.14), but with only those vertices that are landmarks, and the matrix UL consists of the corresponding vertices of those landmarks in the target, so UL = [l1, . . . , ll]T . The final cost function (3.12) will look like this: ¯E (X) = ****** 2 4 M XOR G WD 3 5X - 2 4 0 WU UL 3 5 ****** 2 = kAX - Bk2 F (3.18) Using that ¯E(X) takes on its minimum at X = (ATA)-1ATB we solve the resulting system in each step of the inner loop of the algorithm. Because the goal of this registration is to be used on meshes with large number of points, the matrix A tends to get large, and even though it is sparse and most of it is full of zeros, it causes the solving step to be the most time consuming part of the whole algorithm. This is why we discuss its optimization in the next subsection.

3.2 *Optimizing the Cost Function*

As it was mentioned before, the clustering step described in the subsection Clustering creates a cluster graph that has the medoids of the clustered template graph as vertices, and edges connecting them as the template graph is. The goal of the optimization is to use the cluster graph instead of the original template one and still get good results. This also changes the way the algorithm steps are defined. While in [1] goes from a user defined value to another, for example from 100 to 1 with a step 1, in the case of clustering the algorithm stops when the template cannot be divided into smaller clusters. This will be further explained in the Nonrigid optimal step ICP algorithm subsection. So lets look again at (3.18) and discuss in this subsection how it diers in the case of the cluster optimization. We can even consider the situation when the landmark term is omitted. In this case the cost function looks like this: ¯E (X) = ****

M XOR G WD 0 X -

0 WU 0**** 2 (3.19) and so the matrices A and B that will be addressed couple of times in this subsection are accordingly: A =

M XOR G WD 0 (3.20) B =

0 WU 0 (3.21) The stiness weight is one of the things that change in the clustering optimization. It is in fact removed entirely, because of the way the cluster graph was constructed. All its edges have their own weight as discussed and its substance is to say how strong the two medoids connected by an edge are holding together. This is done by counting how many edges there are between the vertices of the two clusters and this number is the weight of the edge between the medoids that act as vertices in the cluster graph. Those weights start big enough, because in the beginning the clusters are fewer and therefore bigger, that brings a lot of edges adding their numbers to the sum. This is the alternative to the original algorithm starting with a large . Along the run of the algorithm the clusters are divided into smaller chunks until they reach a minimum size that is defined by the user, and so the weights grow smaller. Again this bring the same results as the decreasing of the value. The clusters act like the stiness weight also because all the vertices that belong to a cluster are transformed in a way that depends on the medoid of this cluster and its neighbor medoids as was mentioned in the Assigning Soft Membership Weights section and will be explained again in Applying the Transformations subsection. At the same time omitting entirely the stiness term gave more unsatisfying results in the registration part. And yet even with the part of the matrix A, that accounts for the connectivity of the graph, still present, it is small enough through the large part of the algorithm run, because the cluster graph is much smaller than the original template graph even when the clusters get divided in time. Next optimization aects the part of matrix A that is the distance term, the matrix multiplication WD, and also WU that is a part of matrix B also belonging to the distance term. In this subsection again it is defined that the n = —V — and e = —E— for the original template graph, and also ¯n = — ¯ V — and ¯e = —¯E — for the cluster graph. The new matrix ¯D unlike in (3.14) is defined the following way: Each row corresponds to a vertex in the original graph and every four columns correspond to a cluster. If a vertex vi belongs to a cluster Cj then the coordinates of the vertex are set in the row i and on the 4 columns for the cluster 4 ¡- j, 4 ¡- j +1, 4 ¡- j +2 and

4 ¡- j +3. More formal an element — ¯ dij — in matrix ¯D is defines in the following way: — ¯ dij — = (vi if vi 2 Cj 0 otherwise (3.22) Defined like that the new matrix ¯D has size n 4¯n. The algorithm works for unconnected graphs, as it was mentioned in the subsection Clustering. The reason for this is the modification of the matrix D that was just described. Because of this, even if there is a unconnected cluster, in the case of the cluster graph - unconnected vertex, or many, the matrix will not have a column entirely constructed of zeros. This would otherwise prevent the solving of the equation, making it underdetermined. But with a matrix ¯D the matrix A has a determinant different than zero and the solving step can occur. Unfortunately there are exceptions of this, for example when the unconnected component is constructed of too small number of vertices and no correspondences for them are found. Then the system is underdetermined and cannot be solved. Even a better optimization in the sense of better results can be made if the matrix ¯D is modified little more. For this the membership weights from Assigning Soft Membership Weights section are used. We say that the transformation of a vertex depends on the medoid of the cluster it belongs to, but also on the clusters that are neighbors of this cluster and soft membership weights for each vertex and its clusters are assigned using (3.5). Having this, another version of matrix D is denoted with ˆD and constructed in the following way: Again each row corresponds to a vertex in the original graph and every four columns correspond to a cluster. The coordinates of the vertex vi are set in the row i and on the 4 columns of the clusters Cj that the vertex "belongs" to, multiplied by the membership weight for the vertex and the according cluster. A vertex "belongs" to a cluster if it is in a cluster or if the cluster is a neighbor of the cluster the vertex is in. Again more formal this means the following: — ˆ dij — = (mvi j vi if Cj 2 Cvi 0 otherwise (3.23) where the membership weight is defined as in (3.5) as well as the cluster set Cvi. The matrix ˆD still has the same size as ¯D: n 4¯n and even though fewer elements in it are zeros and the sparse optimization is less notable, the results in the registration are better. Because of those modifications the matrix W contains the corresponding weights for the original template vertices, not the cluster graph's ones. The same goes for the matrix U in the matrix B, where the correspondences ui are n and not ¯n. If we include the landmark term, only the DL matrix needs to be modified in a similar fashion as the matrix D, keeping the vertices that have landmarks in their own rows, but distributed in their corresponding clusters' columns, After those optimizations the matrix A as written in (3.20) has size of (4¯e+n)4¯n and the matrix B as written in (3.21) has size of (4¯e+n)3. But as the main equation was defined in (3.2), for the solving we need the matrices ATA and ATB and those have sizes accordingly 4¯n4¯n and 4¯n3 which is really small in the beginning, because the clusters are a small number, and even later in the algorithm run, it doesn't get too big if the minimum number of vertices in a cluster is set to something reasonable. Thus we get better computation time. In the next subsection the applying of the resulting transformation matrix X is discussed and

4

RESULT

5

CONCLUSION

BIBLIOGRAPHY

[1] Margarita Ganeva. Patch-based nonrigid registration of triangle meshes and point clouds. Master's thesis, Institute for Computer Science, University of Bonn, 2015.

[2] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[3] Hal Blumenfeld. *Neuroanatomy through Clinical Cases*. Oxford University Press, 2010.

[4] R. Douglas Fields. White matter matters. *Scientific American*, 298(3):54–61, mar 2008.

[5] Stephen B. Klein. *Biological Psychology*. Worth Publishers, 2008.

[6] University of Washington. Braininfo. http://braininfo.rprc.washington.edu, 1994.

[7] Sabine E. Grimm, Nigel Armstrong, Bram L. T. Ramaekers, Xavier Pouwels, Shona Lang, Svenja Petersohn, Rob Riemsma, Gillian Worthy, Lisa Stirk, Janine Ross, Jos Kleijnen, and Manuela A. Joore. Nivolumab for treating metastatic or unresectable urothelial cancer: An evidence review group perspective of a NICE single technology appraisal. *PharmacoEconomics*, oct 2018.

[8] Daniel Mamah, Thomas E. Conturo, Michael P. Harms, Erbil Akbudak, Lei Wang, Amanda R. McMichael, Mokhtar H. Gado, Deanna M. Barch, and John G. Csernansky. Anterior thalamic radiation integrity in schizophrenia: A diffusion-tensor imaging study. *Psychiatry Research: Neuroimaging*, 183(2):144 – 150, 2010.

[9] Pr Denis Ducreux. Connectopedia knowledge database. http://www.fmritools.com/kdb/index.html, 2015.

[10] Per Brodal. *The Central Nervous System*. OXFORD UNIV PR, 2016.

[11] Henry Gray. *Anatomy of the Human Body*. Philadelphia : Lea & Febiger, 1918.

[12] Saad Jbabdi, Stamatios N Sotiropoulos, Suzanne N Haber, David C Van Essen, and Timothy E Behrens. Measuring macroscopic brain connections in vivo. *Nature neuroscience*, 18(11):1546, 2015.

[13] M. Yogarajah, J.S. Duncan, and M.J. Koepp. Imaging — structure and function: Imaging connectivity using tractography. In Philip A.

Schwartzkroin, editor, *Encyclopedia of Basic Epilepsy Research*, pages 1586 – 1592. Academic Press, Oxford, 2009.

[14] Hongdong Li and Richard Hartley. The 3d-3d registration problem revisited. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007.

[15] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE Comput. Soc, 2001.