Point-Wise Quantification of Craniofacial Asymmetry

Stéphanie Lanche

June 2007
Ecole Suprieure de Chimie Physique Electronique de Lyon
Section for Image Analysis
Informatics and Mathematical Modelling, IMM
Technical University of Denmark, DTU

Preface

This 6-month project was carried out at the *Informatics and Mathematical Modelling (IMM)*, Technical University of Denmark (DTU), Lyngby, Denmark and the *3D Craniofacial Image Research Laboratory (3D-Laboratory)* at the School of Dentistry at the University of Copenhagen, Copenhagen, Denmark. IMM carries out teaching and research within computer science and information processing with focus on application within engineering science. The 3D-Laboratory is an independent center jointly sponsored by the School of Dentistry, University of Copenhagen; Copenhagen University Hospital and Informatics and Mathematical Modelling, Technical University of Denmark.
Associate Professor Rasmus Larsen (IMM), Research Engineer Tron A. Darvann (3D-Lab), and PhD Student Hildur Ólafsdóttir (IMM), were my supervisors during the project.

Acknowledgements

First, I would like to thank my supervisor Rasmus Larsen (IMM) for finding me this very interesting master project, as well as the Danish scholarship, which helped me with the expensive living costs in Copenhagen.

My co-supervisors Tron Darvann (3D-Lab) and Hildur Ólafsdóttir (PhD Student, IMM) are also thanked for always being *delighted* to help me and give me suggestions.

I am grateful to the whole 3D-Lab team composed of Tron Darvann, Nuno Hermann, Per Larsen, Hildur Ólafsdóttir, Jonas Hermansen, Rene Andersen, Naoko Abe, Seiki Tomita. I could not think of better people to work together with and enjoy cheese fondue during my stay in Denmark.

I also would like to acknowledge the BIOP graduate school for their financial support during this project.

Sven Kreiborg is thanked for never forgetting to invite students to the dinners organized by the 3D-Lab.

My flat mates, Jacob, Alexia and Anna are also thanked for their understanding for my lack of participation in the flat life during the writing period. It was a real pleasure to improve my movie cultur with them ($\cdots$ NOT !)

Finally a warm thanks to my family and friends who supported me despite of the distance. Great thanks to my sister, Karine, for her knowledge of the French language.

Abstract

This thesis presents a methodology of point-wise quantification of craniofacial asymmetry. The asymmetry was computed using two similar methods involving comparison of the right and left sides of the skull.

The new method of asymmetry quantification was applied to two types of craniofacial data: surface scans of infants with deformational plagiocephaly, and micro CT scans of mice with Crouzon syndrome. The asymmetry was quantified and spatially localized. In the first case, a statistical model, created by performing a principal component analysis, was used to assess treatment outcomes. In the second case, the asymmetry quantification permitted population classification by comparing the average asymmetry of the Crouzon mice to the one of a control group.

The proposed methods require establishment of full correspondence between left and right sides of the skull. This was achieved by deforming a perfectly symmetric subject (where each point on the left side had a known corresponding point on the other side) to assume "perfectly" the shape/image of a subject. The procedure combined global (affine) and local (thin-plate splines or B-splines) transformations.

Qualitative and quantitative validations of the presented methods were carried out on the presented methods. Expert measurements and an alternative "naive" method were seen to confirm the ability of our methods to localize and quantify the cranial asymmetry. Furthermore, the statistical model was checked using visual assessment.

Keywords: Asymmetry, craniofacial anomalies, treatment evaluation, population study, statistical modelling, principal component analysis, image registration, thin-plate splines, B-splines.

Contents

Chapter 1

Introduction

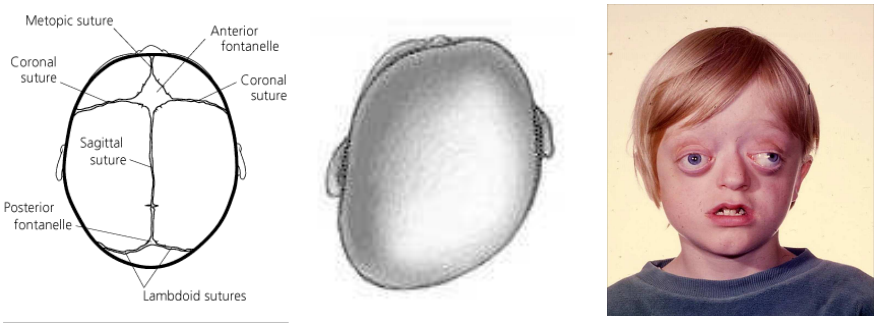
1.1 Craniofacial anomalies

In contrast to the adult skull (where all bones are fused), the skull of the new-born child is composed of a collection of small bones connected with wide growth zones (consisting mainly of connective tissue). The growth zones of the skull are often referred to as sutures (Figure 1.1.a) The sutures will stay open until the skull has finished growing. It should be noted that most of the calvarial growth takes place before one year of age.

Disturbances in the development of the skull are often referred to as craniofacial anomalies (CFA) and are either a) congenital (present at birth) or 2) due to external factors (e.g. pressure on the skull). In both conditions asymmetry is often an important measure for diagnosing and evaluating severity of the anomaly, as well as for evaluating the outcome of treatment.

The present report deals with asymmetry quantification in two different craniofacial anomalies; A) Deformational plagiocephaly (an example of a CFA that is caused by external factors) and B) Crouzon syndrome (an example of a congenital CFA).

Deformational plagiocephaly (DP)(Figure 1.1.b) is a commonly seen craniofacial anomaly. The incidence of DP has increased exponentially during the past decade due to the "back to sleep" campaign to promote supine infant positioning to reduce sudden infant death. This abnormality refers to a deformed head shape resulting from external pressure, either intrauterine or during child positiong after birth, and is normally considered to be only an esthetic problem. The deformation, combining a flattening at the location where the pressure occurs, and a bulging on the opposite side of the skull, reveals mild to severe asymmetry.



a) b) c)

Figure 1.1: Craniofacial sutures and anomalies: a) Cranial sutures[4]. b) Deformational plagiocephaly[5]. c) Crouzon syndrome [18].

Crouzon syndrome (Figure 1.1.c) is a rare craniofacial anomaly. This condition is characterized by a constellation of premature fusion of the cranial sutures (the cranial growth zones), orbital deformity, hypoplastic maxilla (underdeveloped upper-jaw), beaked nose, crowding of teeth, and high arched or cleft palate. The full or partial fusion of cranial sutures at either side of the head and at different time makes the skull grow asymmetrically. The syndrome is caused by heterozygous mutations in the gene encoding fibroblast growth factor receptor type 2 (FGFR2) [34]. Crouzon syndrome is treated by multiple craniofacial surgeries.

1.2 Asymmetry

Quantification and localization of cranial asymmetry is highly important in CFA and may improve the diagnostics, the treatment planning (depending on the severity) and the evaluation of the therapy.

In biology, symmetry refers to a balanced distribution of duplicate body parts or shapes ([6]). The body of most organisms exhibit some type and amount of symmetry. The special case of the human body reveals an organization according to a bilateral symmetry: a vertical plane passing through the middle (called mid-sagittal plane, MSP) divides the body into mirrored halves, referred as the right and left halves. In reality, the symmetry of the body's shape is approximate. For example, while considered symmetric, the mirrored left half of the head will rarely match exactly the other half. The shape of such a head presents some asymmetry. Asymmetry may be defined as the lack of symmetry between the left and right side of the MSP. It may be quantified as the amount of difference between the left and right side of the MSP.

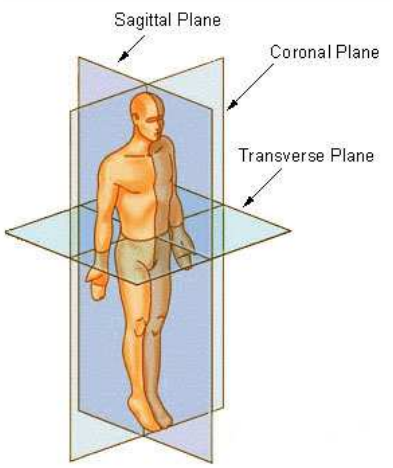


Figure 1.2: Body planes. The midsagittal plane is the sagittal plane passing through the middle of the body, dividing the shape of the body into mirror halves. Image: [3]

Asymmetry has not been studied much in the image analysis literature. The most simple forms of determining head asymmetry include using direct anthropometry of the head (e.g. [17]) or manual measurements on 3D scans (e.g. [32]). These methods produce a multitude of parameters, making the interpretation difficult in terms of asymmetry. [23, 7] defined asymmetry with respect to a sparse set of inter-landmark distances.

1.3 Objectives

The overall purpose of the project was to define a new measure to quantify and localize craniofacial asymmetry.
The developed method should be able to:

- Quantify the primary anomaly (asymmetry).
- Quantify changes due to treatment.
- Quantify differences within and between groups in population studies.

The asymmetry measure was applied to infants with DP to localize and quantify the head asymmetry, and asymmetry change after the therapy. Furthermore, the treatment was evaluated employing two statistical models created using Principal Components Analysis.
A slightly modified asymmetry measure was applied to locally quantify the asymmetry in an animal model of Crouzon syndrome (genetically modified mice) and compare it to a control group of normal mice.

1.4 Enclosed publications

Parts of this thesis work have been reported in the following papers:

S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, A.A. Kane: A Statistical Model of Head Asymmetry in Infants with Deformational Plagiocephaly. In Proceedings of B.K. Ersbøll and K.S. Pedersen. Scandinavian Conference on Image Analysis (2007) 898–907.

S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, A.A. Kane: A Method for Evaluating Treatment in Infants with Deformational Plagiocephaly. Image Analysis in Vivo Pharmalogy (2007).

H. Ólafsdóttir, S. Lanche, T.A. Darvann, N.V. Hermann, R. Larsen, B.K. Ersbøll, E. Oubel, A.F. Frangi, P. Larsen, C. A. Perlyn, G.M. Morriss-Kay, S. Kreiborg: A Point-Wise Quantification of Asymmetry Using Deformation Fields. Application to the Study of Crouzon Syndrome. Accepted for publication in: Proceedings of Medical Imaging Computing and Computer-Assisted Intervention (2007).

S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, A.A. Kane: Head Asymmetry Modelling and its Application to Treatment Evalutation in Infants with Deformational Plagiocephaly. Medical Imaging Computing and Computer-Assisted Intervention (2007). Submitted.

Chapter 2

Preliminaries

2.1 Shape

One of the most intuitive and common shape definitions is given by D.G. Kendall [11]:

> *Shape* is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.

This means that two objects have the same shape if one may be transformed to fit perfectly the other one, by performing only translation, rotation and scaling.

2.2 Landmarks

Shapes are often defined by a set of points, commonly called landmarks [11]:

> A *landmark* is a point of correspondence on each object that matches between and within populations.

Three types of landmarks exist:

- *Anatomical landmark*: A point assigned by an expert that corresponds between objects in a biologically meaningful way.
- *Mathematical landmark*: A point located on an object according to some mathematical or geometrical property, e.g. at a point of high curvature.
- *Pseudo landmark*: A point whose location is dependent on others landmarks, e.g. an equidistant point between two anatomical landmarks.

Figure 2.1 gives an example of anatomical and pseudo landmarks. The shape can be represented by the coordinates of its landmarks. For instance, in a 2D shape,

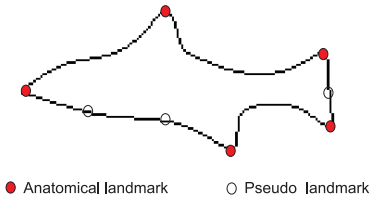


Figure 2.1: Anatomical landmarks and pseudo landmarks placed equidistantly between the anatomical, illustrated on a fish shape [26].

the n landmarks can be arranged in a column vector of size $2n$:

$$\mathbf{s} = [\ x_1\ x_2\ x_n \ldots y_1\ y_2\ y_n\ ]^{\mathsf{T}} \tag{2.1}$$

Chapter 3

Registration

Registration is a process widely used in medical imaging to match images or surfaces so that corresponding features/shapes can be compared. Registration may also be employed to estimate the deviation from a reference image or surface. In this project, registration is used in order to :

- Allow for inter-subject comparison (Chapters 5 and 6).
- Estimate "how asymmetric" the subjects are by calculating the deviation from a perfect symmetric template (Chapter 6).

Generally, the terms registration, matching and alignment are used to refer to any process that determines correspondence between data sets.

3.1 General description

The goal of registration is to determine the optimal transformation $\mathbf{T}$ that maps one image (or surface), the *source*, into the coordinate system of another image (or surface), the *target*, (Figure 3.1). This provides anatomical correspondences. Usually, with biomedical data, the differences between the *source* and the *target* images are non-rigid, i.e. rotation, translation and scale transformations alone are not sufficient to describe $\mathbf{T}$. Therefore, the transformation $\mathbf{T}$ combines global and local transformations, and in our work it satisfies:

$$\mathbf{T}(x, y, z) = \mathbf{T}_{global}(x, y, z) + \mathbf{T}_{local}(x, y, z), \tag{3.1}$$

3.2 Global transformation

The global registration provides the global transformation that captures the gross differences between the *source* and the *target*. The affine transformation chosen

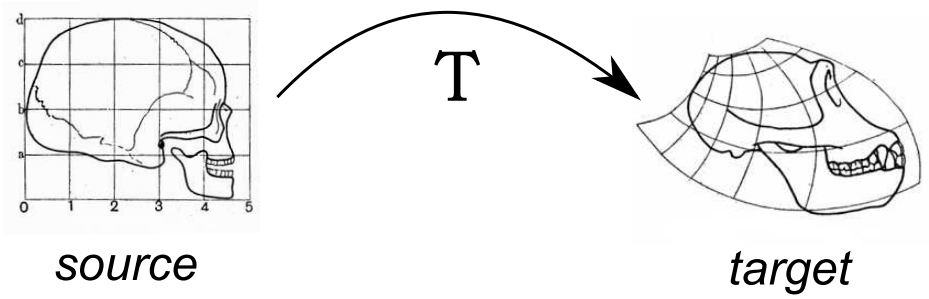


Figure 3.1: Purpose of image registration: mapping the coordinate system of the image *source* into the coordinate system of the *target*. Images: Thomson, D'Arcy W [41].

in this work is defined by nine degrees of freedom (DOF). The transformation $\mathbf{T}_{\boldsymbol{global}}$ maps a point in the source image (x,y,z) into the corresponding point in the target image (x',y',z'). It allows rotations (three DOFs), translations (three DOFs) and anisotropic scaling (three DOFs).

$$\mathbf{T}(x,y,z) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.2}$$

After this transformation, all images have the same orientation and same overall size.

3.3 Local transformation using splines

To obtain a registration focusing on local differences, non linear transformations are required. In this work, transformations based on splines are used.

The term *spline* ([13]) originally referred to the use of long flexible strips of wood or metal to model the surfaces of shapes and planes. These splines were bent by attaching different weights along their length. A similar concept can be used to model spatial transformations. For example, a 2D transformation can be represented by two separate surfaces whose height above the plane corresponds to the displacement in the horizontal or vertical direction. Mathematically, a polynomial spline of degree n is defined as a piecewise polynomial function of degree n with pieces that are patched together such as to guarantee the continuity of the function and of its derivatives up to order $n-1$.
Registration using splines is based on the assumption that a set of corresponding points or landmarks can be identified in the *source* and *target* images. These corresponding points are referred to as control points. At these control points,

spline-based transformations either interpolate or approximate the displacements which are necessary to map the location of the control points in the *target* image into its corresponding counterpart in the *source* image. There are a number of different ways to determine the control points:

1. Anatomical or geometrical landmarks identified in both images (Chapter 5)
2. Pseudo-landmarks, arranged with equidistant spacing across the images forming a regular mesh (Chapter 6)

In this work, two local transformation models based on splines are employed, both combined with a global transformation model:

- Thin-Plate Splines (TPS) combined with closest point (CP) algorithm used for surface registration (Chapter 5)
- B-splines used for volume registration (Chapter 6).

3.3.1 Thin-Plate Splines combined with closest point determination

Thin-Plate Splines (TPS) were introduced by Bookstein ([7]) for statistical shape analysis, and are widely used in computer graphics. They are a part of the family of splines that are based on n radial basis functions $U(r)$:

$$\mathbf{T}_{local}(x,y,z) = \sum_{l=1}^{n} b_j \; U(|\boldsymbol{M_i} - (x,y,z)|) \tag{3.3}$$

where b are the non-affine coefficients and $\mathbf{M}$ the locations of the control points. The radial basis functions of thin-plate splines are defined as:

$$U(r) = |r|^2 \; log(|r|) \quad in \; 2D \tag{3.4}$$

$$U(r) = |r| \quad in \; 3D \tag{3.5}$$

where $r^2 = x^2 + y^2 + z^2$.
It consists of a non-linear transform that stretches and bends an object to fit the control points (landmarks). The transform warps the template so that the source landmarks coincide perfectly with the target landmarks, and transforms the object in between the control points, using an energy-minimizing function.
After this process, the source and target shapes match perfectly at the landmarks location, and are very similar elsewhere. Subsequently, closest point determination is carried out: each point on the deformed source object is moved to the closest point location on the target object. The transformed source object aligns (almost) perfectly with the target object. In a number of cases, the global influence of control points is undesirable since it becomes difficult to model local deformations. Furthermore, for a large number of control points, the computational complexity of the radial basis functions becomes prohibitive.

3.3.2 B-Splines

An alternative to the TPS is to use Free-Form Deformations (FDDs) based on locally controlled functions, such as the B-splines [35]. The FFD model can be written as the tensor product of the one-dimensional (1D) cubic B-splines:

$$\mathbf{T}_{local}(x,y,z) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\mathbf{M}_{i+l,j+m,k+n} \tag{3.6}$$

where $i = \lfloor x/n_x \rfloor - 1, j = \lfloor y/n_y \rfloor - 1, k = \lfloor z/n_z \rfloor - 1, u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$ and $w = z/n_z - \lfloor z/n_z \rfloor$.
$\mathbf{M}$ refers to the mesh of control points of size $n_x \times n_y \times n_z$ with spacing $(\delta_x \times \delta_y \times \delta_z)$. B_0 through B_3 represent the basis functions of the B-spline:

$$\begin{aligned} B_0(u) &= (1-u)^3/6 \\ B_1(u) &= (3u^3 - 6u^2 + 4)/6 \\ B_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6 \\ B_3(u) &= u^3/6. \end{aligned}$$

The underlying image is then deformed by manipulating the mesh of control points, creating a dense deformation vector field which can be assessed at any point in the image. It controls the shape of the 3D object and produces a smooth and C^2 continuous transformation.
In contrast to radial basis function splines which allow arbitrary configurations of control points, spline based FDDs require a regular mesh of control points with uniform spacing. In particular, the basis functions of cubic B-splines have a limited support, i.e., changing control points affects the transformation only in the local neighborhood of that control point.
This method is free of manual landmarking since it uses image intensities for optimization. This automatic image registration algorithm requires the following:

- A transformation model, $\mathbf{T}$.
- A measure of image similarity.
- An optimization method to optimize the similarity measure with respect to the transformation parameters.

Similarity measure: Normalized mutual information.

The description of the similarity measure given here was inspired by [27]. In order to bring images into correspondence by automatic image registration, the degree of similarity between the two images needs to be defined. The Normalized Mutial Information (NMI) is based on entropy measures in the two images. The marginal entropy in an image relates to the information content, or more intuitively it

measures the uncertainty of guessing a voxel intensity. In image M with voxel intensities $m \in M$ the marginal entropy is defined as

$$H(M) = -\sum_{m \in M} p\{m\} log(p\{m\}) \quad , \tag{3.7}$$

where $p\{m\}$ is the marginal probability. The joint entropy is defined on the overlapping region between the two images M and N with voxel intensities $m \in M$ and $n \in N$,

$$H(M,N) = -\sum_{m \in M} \sum_{n \in N} p\{m,n\} log(p\{m,n\}) \quad , \tag{3.8}$$

where $p\{m,n\}$ is the joint probability. This corresponds to the information content of the combined scene or the probability of guessing a pair of voxel intensities. Mutual information describes the difference between the sum of the marginal entropies and the joint entropy and by dividing by the joint entropy, NMI is defined as

$$NMI(M,N) = \frac{H(M) + H(N)}{H(M,N)} \quad . \tag{3.9}$$

The strength of entropy measures, such as NMI, is their ability to cope with two different modalities [e.g. 40, 42] but they have been widely used with good results in intra-modality applications as well [e.g. 38, 31, 36].

Optimization: gradient descent

An algorithm often chosen is the gradient descent optimization. For details, the reader is referred to [33].

Chapter 4

Statistical modelling using principal component analysis

This chapter introduces the reader to the statistical modelling, which is employed to study the variation within a population. In Chapter 5, the Principal component analysis was used to study head asymmetry.

4.1 Data alignment

The statistical analysis of some data requires in some cases a preprocessing step for removal of location, scale and rotational effects from the dataset (a procedure also called alignment). In the present study, this will be achieved by carrying out image registration (Chapter 3). The data in the training set are then represented in a common coordinate system.

4.2 Principal component analysis

Principal Component Analysis (PCA) was first introduced by Harold Hotelling in 1930 based on Karl Pearson's work [14]. The purpose of this statistical method is to produce a lower dimensional description of multivariate data, obtained by rotating the data set so that the variance is maximized. Consider having N planar observations consisting of n points (variables), where each observation is represented as:

$$\mathbf{s} = [s_1^T \ \cdots \ s_N^T] = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{N1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{Nn} \\ y_{11} & y_{21} & \cdots & y_{N1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1n} & y_{2n} & \cdots & y_{Nn} \end{bmatrix} \tag{4.1}$$

where each column is the i^{th} observation defined by n points. The observation dataset is then characterized by its means $\overline{s}$ and the covariance matrix Σ_s:

$$\overline{\mathbf{s}} = \frac{1}{N}\sum_{i=1}^{N}(s_i) \tag{4.2}$$

$$\mathbf{\Sigma}_s = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{s}_i - \overline{\mathbf{s}}_i)\,(\mathbf{s}_i - \overline{\mathbf{s}}_i)^{\mathrm{T}} \tag{4.3}$$

The covariance matrix of the n observations (estimated by the maximum likelihood) characterizes the dispersion of observations within the dataset, i.e. it measures their dependency. The obtained covariance matrix $\mathbf{\Sigma}_s$ is defined symmetric. Therefore, the basis, formed by its columns (eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^{2n}$) is orthogonal.

4.3 Eigenanalysis and principal components

Eigenvalues $\boldsymbol{\lambda}_i$ and eigenvectors $\boldsymbol{\phi}_i$ are determined by an eigenanalysis of $\mathbf{\Sigma}_s$:

$$\mathbf{\Sigma}_s\,\mathbf{\Phi}_s = \mathbf{\Phi}_s\,\mathbf{\Lambda}_s \tag{4.4}$$

where

$$\mathbf{\Phi}_s = \begin{bmatrix} & & & \\ \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \dots & \boldsymbol{\phi}_{2N} \\ & & & \end{bmatrix} \quad \text{and} \quad \mathbf{\Lambda}_s = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{bmatrix} \tag{4.5}$$

where the eigenvalues are ordered so that $\lambda_1 \geq \lambda_2 \geq \dots \lambda_N$. The eigenvalues and eigenvectors indicate the amount and direction of the variations present in the dataset. The first eigenvector Φ_1 is defined so that it maximizes the variance of the original observations. Analogously, the second eigenvector Φ_2 is defined to represent the second largest variance present in the original observations. The principal components are the projections of the eigenvectors projected along the eigenvectors (directions of principal variations):

$$pc = \Phi \cdot s \tag{4.6}$$

The principal components are consequently linear combinations of the original observations and are linearly independent. They represent each observation in the direction of principal variations, defined by the eigenvectors. Notice: When there are fewer samples than dimensions in the vectors (often the case in medical image analysis) the eigenanalysis is applied to the reduced covariance matrix instead of on the covariance matrix itself. This permits a large computation time reduction.

$$\mathbf{\Sigma}_{reduc} = \frac{1}{N}\,\mathbf{s}^{\mathrm{T}}\mathbf{s} \tag{4.7}$$

The eigenvalues λ_i and the eigenvectors ϕ_i may then be computed by:

$$\mathbf{\Lambda}_s = \mathbf{\Lambda}_{reduc} \tag{4.8}$$

$$\mathbf{\Phi}_s = \mathbf{s}\,\mathbf{\Phi}_{reduc} \tag{4.9}$$

where $\mathbf{\Lambda}_{reduc}$ and $\mathbf{\Phi}_{reduc}$ contain the eigenvalues and eigenvectors of the reduced covariance matrix, respectively.

4.4 Instance

A model instance may be generated by modifying the mean observation adding a linear combination of eigenvectors:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{\Phi}_s\,\mathbf{b} \tag{4.10}$$

where $\mathbf{b} = (b_1, \ldots, b_n)$ contains the observation model parameters. The variance along the i^{th} principal component across the training set is given by the corresponding i^{th} eigenvalue λ_i. Therefore, by applying limits of $+/-\,3\sqrt{\lambda_i}$ to the parameters b_i, we ensure that the instance generated is similar to the observations in the original training set.

4.5 Model compactness

For each $i = 1, \ldots, p$, the vector ϕ_i is the i^{th}*mode* of variation and the scalar b_i is the modal magnitude of the associated variation. The i'th model parameter b_i has variance λ_i and typically instances similar to the ones modelled are assured by applying limits of $\pm 3\sqrt{\lambda_i}$, i.e. 3 standard deviations.
Usually, a particular number n_m of eigenvectors is retained so the model explains a given proportion (e.g. 95%) of the total variance exhibited in the training set:

$$\frac{\sum_{i=1}^{n_m}(\lambda_i)}{\sum_{i=1}^{n}(\lambda_i)} \geq \frac{p}{100} \tag{4.11}$$

4.6 Illustration

An illustration of the use of PCA in statistical shape modelling can be found in Appendix A, where the hand shape is analysed in a training set of 40 observations (Data: [39]).

Chapter 5

Asymmetry measure in infants with deformational plagiocephaly

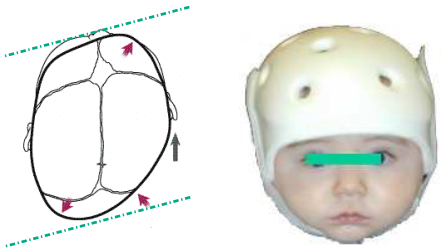
5.1 Introduction

5.1.1 Purpose

The purpose of this work was to develop a methodology for assessment and modelling of head asymmetry, and for treatment evaluation in infants with deformational plagiocephaly. A new asymmetry measure was defined and used in order to quantify and localize the asymmetry of each infant's head, and again employed to estimate the changes of asymmetry after the therapy for each infant. A statistical model of head asymmetry was then developed using a Principal Components Analysis and used to evaluate the treatment. The results were finally validated and discussed.

5.1.2 Deformational plagiocephaly

Deformational (or positional) Plagiocephaly (DP) refers to an asymmetric, deformed shape of the head, commonly seen in infants. Plagiocephaly literally means "oblique head". This deformation is thought to result from repeated pressure to the same area of the head that flattens the skull at this location. Occasionally, an infant with tight intrauterine environment may be born with this type of flattening. DP is most commonly manifested as either left-right asymmetry or brachycephaly (head shortened without asymmetry). It affects the occiput (back of the head), and to a lesser extent, the forehead contour. When viewed from above, the head shape can be inscribed within a parallelogram (Figure 5.1.a. The



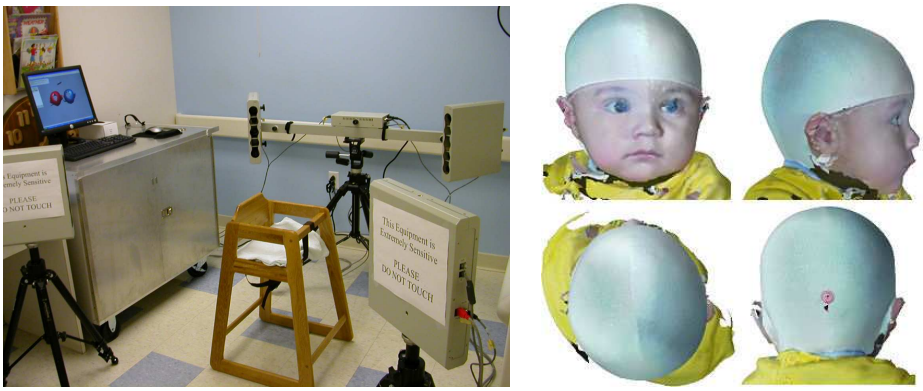
a) b)

Figure 5.1: deformational plagiocephaly. a) Skull with DP in an axial view. Areas affected by the cranial anomaly are pointed out with arrows. b) Infant wearing orthotic molding helmet for correction of DP. Image: courtesy of Dr Alex A. Kane, St. Louis Children's Hospital, St. Louis, MO, USA.

incidence of DP has been estimated to be as high as 15% in the USA [24], and has increased exponentially due to the "back to sleep" campaign to promote supine infant positioning to reduce sudden infant death. The treatment is non-surgical and is dependent on the severity. Possible treatments include parental education on how to prevent further deformations, e.g. alternating sleep positions [15] and orthotic molding helmet therapy (e.g., [1] and [22]) for moderate to severe deformities. Helmets, made of an outer hard shell with a foam lining, apply gentle and persistent pressures to capture the natural growth of an infant's head (Figure 5.1.b. While the growth in the prominent areas are inhibited, the growth in the flat regions is allowed. It is widely held that correction is best accomplished in infancy due to the sequence of skull mineralization. The average duration of treatments with a helmet is usually three to six months, depending on the age of the infant and the severity of the condition.

5.2 Material

3D full-head surfaces were captured using a 3dMD cranial system (www.3dMD.com, capture speed: 1.5 ms, accuracy:<0.5 mm RMS) at the Division of Plastic & Reconstructive Surgery, Washington University School of Medicine, St. Louis, MO, USA. This system involves projecting a random light pattern on to the subject and capturing him or her with precisely synchronized digital cameras set at various angles. Stereo-photogrammetrical reconstruction is used in order to create a 3D representation of the object [2] (Figure 5.2). 3D full-head surfaces of 42 patients with DP were captured both before and after treatment utilizing the 3dMD cranial system. All infants commenced their helmet treatments before 6 months of age, and were treated for a maximum of 6 months. Figure 5.3 presents examples of surfaces extracted from these scans.



a) b)

Figure 5.2: Data acquisition system. a) 3dMD full head surface scanner at the Division of Plastic & Reconstructive Surgery, Washington University School of Medicine, St. Louis, MO, USA. b) Four views of a surface scan example with image texture

5.3 Pre-processing: template matching

The proposed method for asymmetry measure (described in the forthcoming sections) requires establishment of a detailed point correspondence between surface points on the left and right side of the head, respectively. In order to compare the asymmetry results, an inter-subject correspondence is required. This was achieved in a previous work by deforming a symmetric ideal head surface (template) to assume the shape of the patient head surface (template matching) [10]:

1. Creation of the symmetric template head surface with full left/right side correspondence.
2. Deformation of the symmetric template to match each patient's head, using affine and TPS registration.

5.3.1 Creation of the symmetric template

A CT scan of a normal infant with near-symmetric head shape, and with a similar age as the subjects in the study group (9 months of age) was used in order to create a symmetric ideal template.
First, the 3D surface of the normal infant was extracted from the CT scan by selecting the region of the 3D scan having intensity corresponding to bone. This was achieved using intensity thresholding and surface reconstruction by use of the marching cubes algorithm [25]. The resulting surface consisted of a triangulated mesh of points (Figure 5.4).
The resulting triangulated surface was then cut manually along a plane corre-

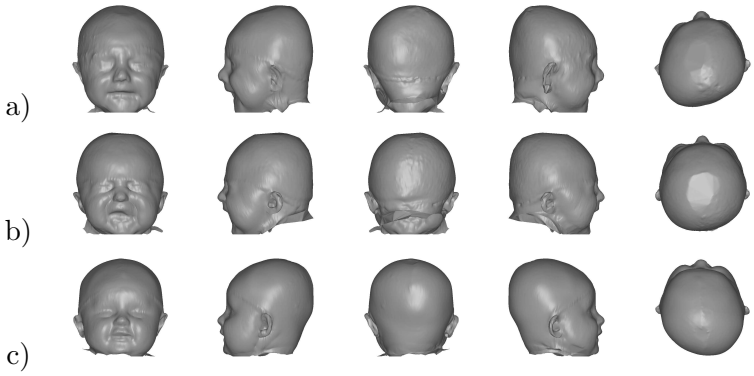


Figure 5.3: Five different views of three of the captured 3D full head surfaces. a) Right-sided flattening posteriorly and left-sided flattening anteriorly. b) Brachycephaly. c) Left-sided flattening posteriorly and right-sided flattening anteriorly.

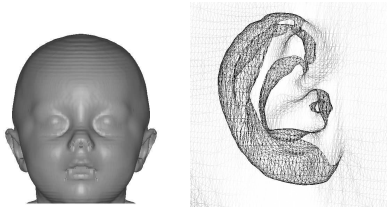


Figure 5.4: Triangulated surface created by the Marching Cubes Algorithm: a) appearance of a scan. b) Illustration with ear structure.

sponding to the midsagittal plane (MSP), which is plane dividing the head into equal left and right halves. In practice, the MSP was easily defined as the plane going through the midpoint between the left and right ear (anatomical) landmarks. The part of the skull located on the left side of the MSP was discarded (Figure 5.5.a, and replaced by a mirrored (left-right reflected) version of the part of the surface located on the right side of the MSP (Figure 5.5.b.
The resulting template is a perfect symmetric surface where each surface point on the left side has a known corresponding point on the right side (Figure 5.5.c). It is referred to as template (also called the reference frame, or atlas) for the deformation described in the next section.

5.3.2 Deformation of the symmetric template

First, the ideal template is globally deformed in order to fit each patient's head. This was achieved using affine transformations (c.f. section 3.2).
Scaling is accomplished using five manually placed landmarks: left and right ear landmarks (width scaling), nasion and a landmark at the back of the head (length scaling) and a landmark on top of the head (that together with the ear landmarks

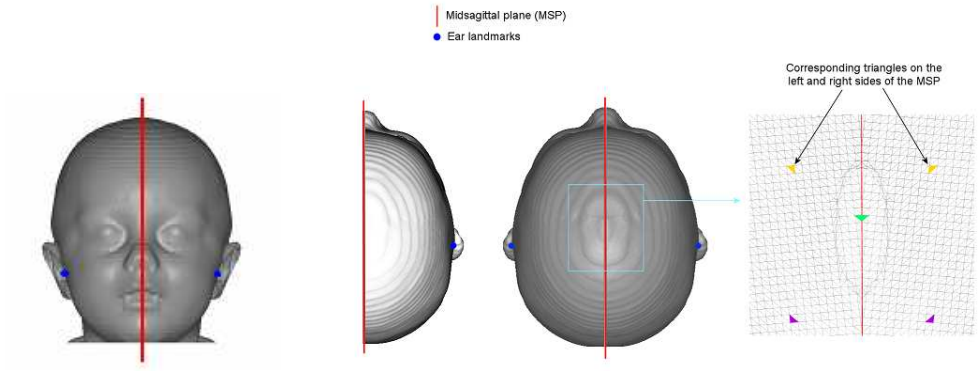


Figure 5.5: Steps for the symmetric template creation. a) Determination of the midsagittal plane: vertical plane going through the midpoint between the ear landmarks. b) Left side of the skull was discarded. c) The discarded left side is replaced by the mirror of the right side. d) Full correspondence between the left and right side of the skull: each point on the left side has a corresponding point on the other side.

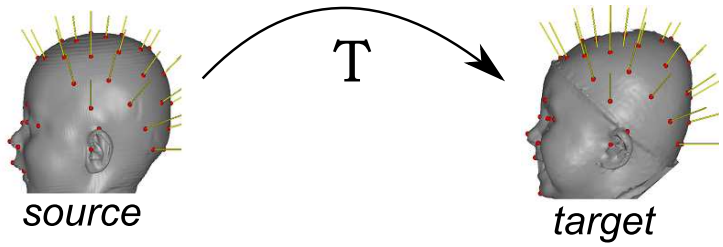


Figure 5.6: Surface registration between the symmetrical ideal template and the scan of an infant. Landmarks (red dots) control the registration. Landmarks on the top/back of the head were determined by use of spikes (yellow lines).

enable height scaling). The patient surface is oriented to the scaled template surface using a rigid transformation based on three landmarks: left and right ear landmarks and nasion. The result is translated such that the midpoint between the ears for the two scans coincide. Once the template and patient surfaces have the same orientation and size, the ideal template is locally deformed combining Thin-Plate Splines and closest point deformation (c.f. section 3.3.1). The TPS transformation is controlled by 60 landmarks (Figure 5.6): 20 anatomical landmarks (ear landmarks and facial landmarks) as well as 40 pseudo landmarks. The latter landmarks were determined by intersecting the surfaces with 40 radial lines (equidistant in terms of angle) originating from the midpoint between the ears. They were necessary in order to control the deformation at the top and back of the head where there are no visible anatomical landmarks.
Each of the resulting 42 surfaces have the "exact" patient head shape with full

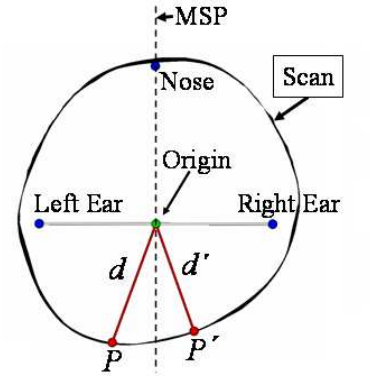


Figure 5.7: Computation of the asymmetry: illustration of the distances d and d' between the origin and the points P and P', respectively, in an axial view.

correspondence between the left and right sides of the skull. This procedure also ensures the inter-subject alignment required for the statistical modelling.

5.4 Methods

The methods employed in this Chapter are the following:
1. Asymmetry quantification (and validation)
2. Statistical modelling of asymmetry (and validation)
3. Treatment evaluation (and validation)

5.4.1 Computation of asymmetry and asymmetry change

The definition of the asymmetry A_P in a point P involves the computation of the ratio between two distances: 1) the distance d from the origin (midpoint between the ear landmarks) to the surface point P on one side of the midsagittal plane, and 2) the distance d' from the origin to the corresponding point P' on the other side of the midsagittal plane (Figure 5.7).

Since, intuitively, the amount of asymmetry at P and P' should be equal, except for a sign introduced in order to distinguish a point in a "bulged" area from a point in a "flattened" area, A_P and $A_{P'}$ are defined by:

$$\text{if } d > d' \text{ then } A_P = 1 - (\frac{d'}{d}) \text{ and } A_{P'} = -A_P \tag{5.1}$$

$$\text{if } d' > d \text{ then } A_{P'} = 1 - (\frac{d}{d'}) \text{ and } A_P = -A_{P'} \tag{5.2}$$

The change in head asymmetry is calculated as the difference between the asym-

metry absolute values before and after treatment:

$$\text{change, P} = |A_{P,\text{before}}| - |A_{P,\text{after}}| \tag{5.3}$$

Hence, a positive change may reveal improvement in the head asymmetry, as $A_{P,\text{after}}$ would be closer to 0 than $A_{P,\text{before}}$ in this case (i.e., closer to perfect symmetry).

5.4.2 Statistical asymmetry model

The PCA is performed on the asymmetry measures of each point of the skull's surface in the helmet region (Figure 5.4.d and e). The points are aligned and ordered in each scan according to the mesh points of the template scan and stored in a vector of size $M/2$ (since A_P and A'_P have the same absolute value, only the asymmetry values for the points situated on the one side of the MSP are considered):

$$\mathbf{a} = [|A_{P1}|\,,\ |A_{P2}|\,,\ \ldots\,,\ |A_{PM/2}|] \tag{5.4}$$

An asymmetry instance can be generated by modifying the mean asymmetry, adding a linear combination of eigenvectors:

$$\mathbf{a} = \overline{\mathbf{a}} + \mathbf{\Phi}_a\,\mathbf{b}_a \tag{5.5}$$

where $\mathbf{b}_a$ is a matrix containing the asymmetry model parameters.

5.4.3 Statistical model of asymmetry change

Analogously as in the previous section, a PCA is carried out on the asymmetry change (Equation 5.3):

$$\mathbf{C} = [\text{change}_{P1}\,,\ \ldots\,,\ \text{change}_{PM/2}] \tag{5.6}$$

The asymmetry change is defined for each pairs of corresponding points. An asymmetry change instance can be generated by modifying the mean, adding a linear combination of eigenvectors:

$$\mathbf{c} = \overline{\mathbf{c}} + \mathbf{\Phi}_c\,\mathbf{b}_c \tag{5.7}$$

where $\mathbf{b}_c$ is a matrix containing the asymmetry change model parameters.

5.5 Results

5.5.1 Projection of 3D surfaces into 2D flat maps

To present the 3D results in a compact way, 2D flat maps were constructed by projecting the 3D surfaces onto a sphere. This was achieved by a simple transformation from rectangular (x, y, z) to spherical coordinates (r, ϕ, θ) (Figure 5.8.a):

$$r = \sqrt{x^2 + y^2 + z^2} \tag{5.8}$$

$$\phi = \arcsin(\frac{z}{\sqrt{x^2 + y^2 + z^2}}) \tag{5.9}$$

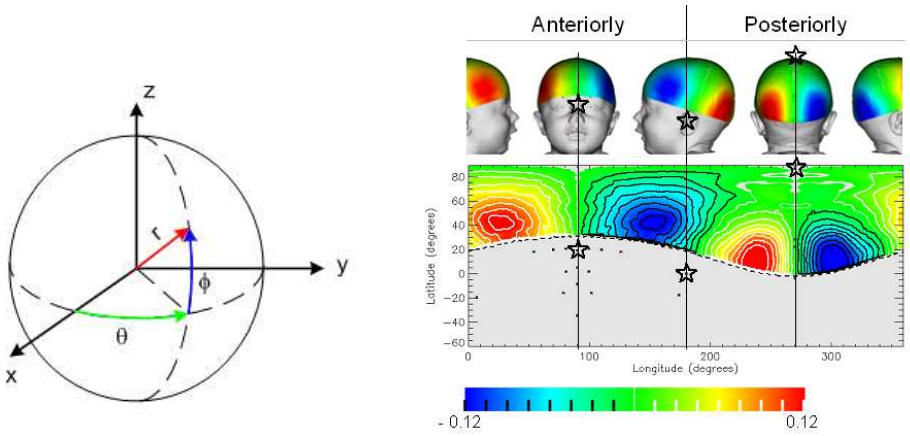
$$\theta = \arctan(\frac{y}{x}) \tag{5.10}$$

The flat map has the right ear landmark at longitude $(\theta) = 0$ degrees, the midface at 90 degrees, the left ear landmark at 180 degrees and the center of the back of the head at 270 degrees. These landmarks are displayed with a star symbol in Figure 5.8.b.
Furthermore, results are presented as surface coloring. Each surface point P is colored according to its asymmetry value using an appropriate color-lookup table (or color map) [1] . The chosen color table is symmetric and ranges from blue (negative values) to red (positive values), with gray in the middle (values equal to zero). Note that, as $A_P = -A'_P$, corresponding points on both sides of the midsagittal plane are displayed with the same intensity but with opposite colors. After projecting the color surfaces into flat maps, contours are added, enhancing asymmetry levels. There are 16 contour intervals, spanning the range of asymmetry as indicated by the color bar (Figure 5.8). The contours are equidistant in terms af asymmetry and are drawn in black for negative values and white for positive values. Hence, in the particular case of asymmetry measure, "bulged" areas become blue with black contours, "flattened" areas become red with white contours, and perfect symmetric areas are shown in gray (as illustrated in Figure 5.8). Regions below the helmet area are also shown in gray.

5.5.2 Asymmetry measure

Figure 5.9 presents the results of the asymmetry computations in three example subjects. Top views of the head before a) and after b) treatment are shown together with corresponding asymmetry flat maps for visual comparison. In addition, a map of change c) is shown.
Figure 5.9.1 shows an asymmetric DP patient with a right-sided flattening posteriorly, as well as a left-sided flattening anteriorly (a). The colors are in good

[1] A color-look up table (LUT) is used to determine the colors and intensity values with which a particular image will be displayed. In the RGB color space, it will be a mx3 matrix. Each row defines a color according to the red, green and blue components.



a) b)

Figure 5.8: Flat map construction. a) Spherical coordinate system (r, ϕ, θ). b) Example of asymmetry measures. Views of 3D colored and the corresponding flat map. Landmarks are shown as star symbols. Anterior (front) and posterior (back) parts of the head are exhibited. Lower limit of helmet region is shown as dashed curve.

agreement with the head shape (compare top views and colors in flat maps). The typical parallelogram shape in DP (c.f. section 5.1.2) is also reflected in the flat map by opposite colors anteriorly and posteriorly on the same half of the skull. Note the improvement in asymmetry after treatment in both of the affected areas (b,c). Be aware of the modification of color table in the map of change: in this case, red (positive value) reveals improvement and blue (negative value) degradation in terms of asymmetry.
Figure 5.9.2 shows a typical brachycephalic patient (a). Brachycephalic patients are generally not very asymmetric, as their deformation mainly causes a foreshortening of the skull. Note improved shape after treatment (b,c).
The third patient, Figure 5.9.3, has left-sided flattening posteriorly as well as a right-sided flattening anteriorly a). Note the improvement after treatment (b,c).

5.5.3 Statistical model of asymmetry

A statistical model was created by performing a PCA on the asymmetry results of the 84 heads (before and after treatment). The eigenvalue distribution (Figure 5.10.a indicates that 95 % of the asymmetry variation could be described using the first six model parameters. The mean asymmetry (Figure 5.10.b emphasizes posterior and anterior regions with high asymmetry, while the anterior parts exhibit smaller magnitude.
Figures 5.10.c-f show the variations $\mathbf{\Phi b}_a$ corresponding to the first four modes (cf. Equation 4.10), with $\mathbf{b}_a = -3$ standard deviations. The two major modes (c-d) occur in the posterior and anterior regions of the head, respectively. Notice that

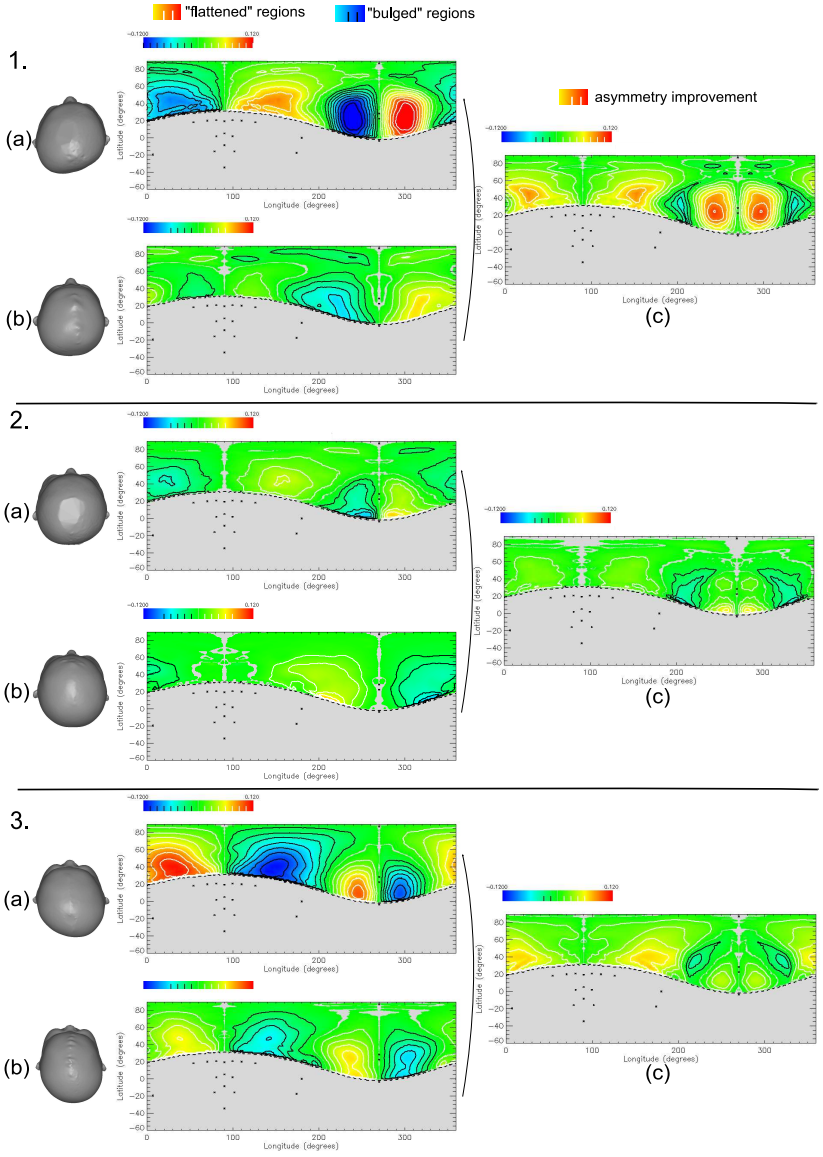


Figure 5.9: Results of the asymmetry computation and changes for: 1. Right-sided flattening posteriorly and left-sided flattening anteriorly. 2. Brachycephaly. 3. Left-sided flattening posteriorly and right-sided flattening anteriorly. a) Scans before treatment. b) Scans after treatment. c) Asymmetry changes after the therapy. In the flat maps showing asymmetry (middle column), positive and negative values denote "flattening" and "bulging" respectively. In the flat maps of change, positive values denote improvement.

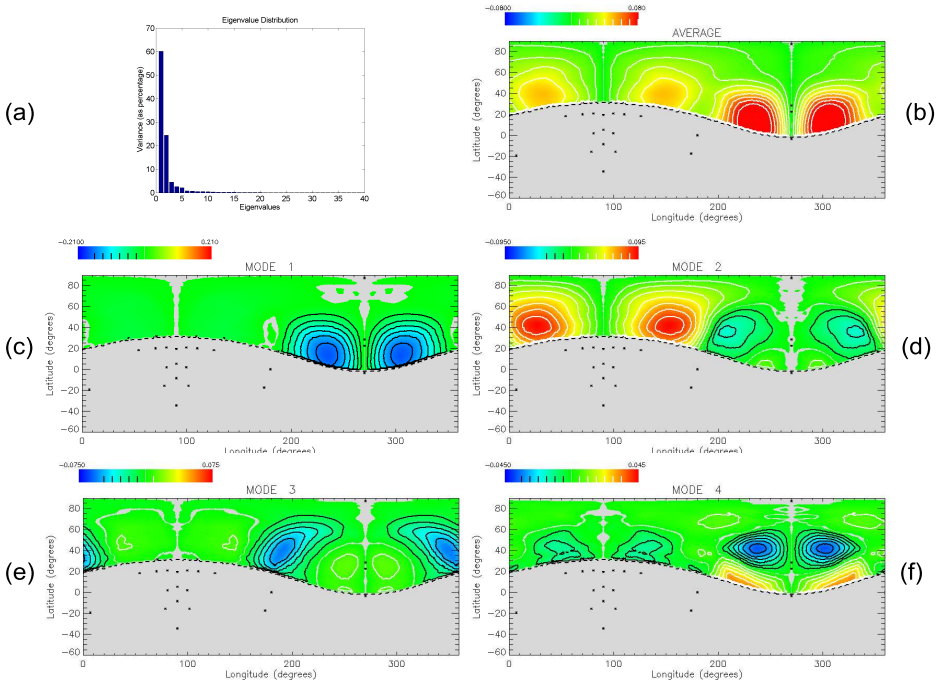
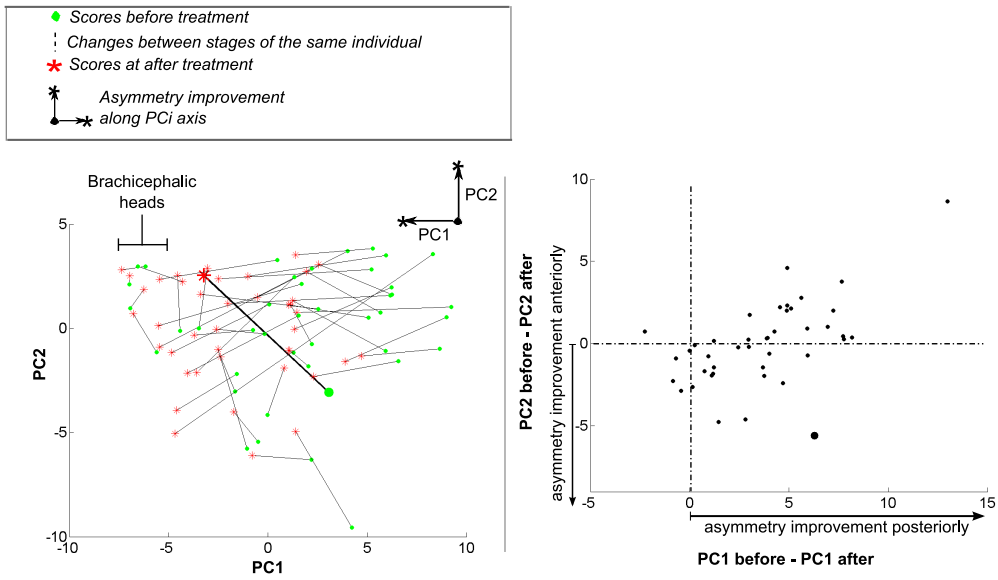


Figure 5.10: Presentation of the asymmetry model. a) Eigenvalues (as percentage of the total variation). b) Mean of absolute value of asymmetry. c)–f) Modes 1 to 4. Modes are shown as variation at -3 standard deviations from the mean. Within the same mode, regions colored by values at the opposite ends of the color range (e.g., red and blue) are displayed with opposite contour colors (i.e., black and white) vary in opposite directions.

the area emphasized by the second mode is spatially more spread out than the first mode. The variation corresponding to the third mode (e) is localized above the ears. Higher modes (as mode four (f)) revealed variability in the location of the affected area posteriorly or anteriorly (high spatial frequency). As the latter information is not important to the study, they have not been displayed.

The scores of the first two principal components, PC (Figure 5.11) demonstrate the direction and amount of asymmetry change for each individual. In Figure 5.11.a, the scores for PC2 are plotted against the scores for PC1. The amount of posterior (PC1) and anterior (PC2) asymmetry may be read off the x- and y-axes, respectively. The least amount of asymmetry is found in the upper-left corner of this figure. This is the region where good treatment outcomes are located, as well as the brachycephalic heads (small asymmetry). Scores between before and after treatment within the same patient were linked with lines in the figure, allowing the visualization of each infant's head asymmetry evolution. Individuals that improve in terms of posterior asymmetry move leftward in the diagram, whereas



a) b)

Figure 5.11: Score plots of the asymmetry model: a) PC2 vs. PC1. b) $PC2_{Change}$ vs. $PC1_{Change}$. Bold features represent the results for a patient whose head asymmetry results were shown in Figure 5.9.1

individuals that improve in terms of anterior asymmetry move upward[2].
The scores of the infant presented in Figure 5.9.1 were enhanced using bold features. They show a posterior and anterior asymmetry improvement also seen in the map of change (Figure 5.9.1.c).

5.5.4 Treatment evaluation

a) Modelling asymmetry change

A statistical model of asymmetry change was created by performing a PCA on the 37 change maps, as explained in section 5.4.3. 91% of the total variance is represented by the first six model parameters. The mean asymmetry change reveals a general improvement occurring posteriorly and anteriorly (Figure 5.12.a, where the main head asymmetries where localized. The first mode of asymmetry variation involves the back and the front of the head, where these areas are varying in opposite directions. The second asymmetry variation occurs anteriorly, with a variation in the same direction posteriorly. None of the modes describe the asymmetry changes involving separately the posterior and anterior regions.

[2]The direction of asymmetry change depend on the signs of the axes, which is arbitrarily chosen by the PCA.

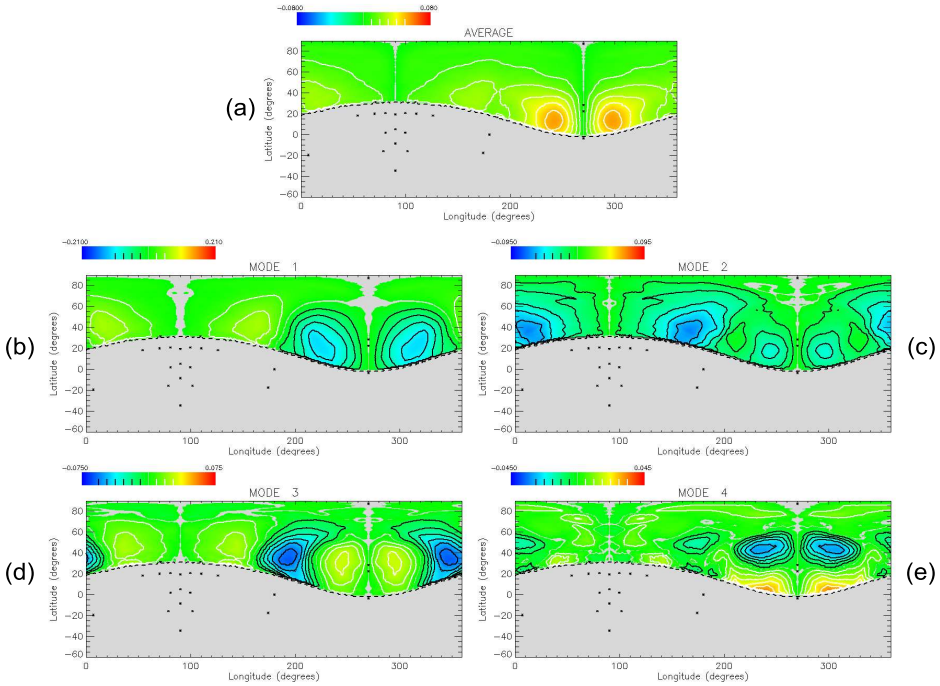


Figure 5.12: Presentation of the model of asymmetry change. a) Mean improvement. b)–e) Modes 1 to 4. Modes are shown as variation at −3 standard deviations from the mean. Within the same mode, regions colored by values at the opposite ends of the color range (e.g., red and blue) and displayed with opposite contour colors (e.g., black and white) vary in opposite directions.

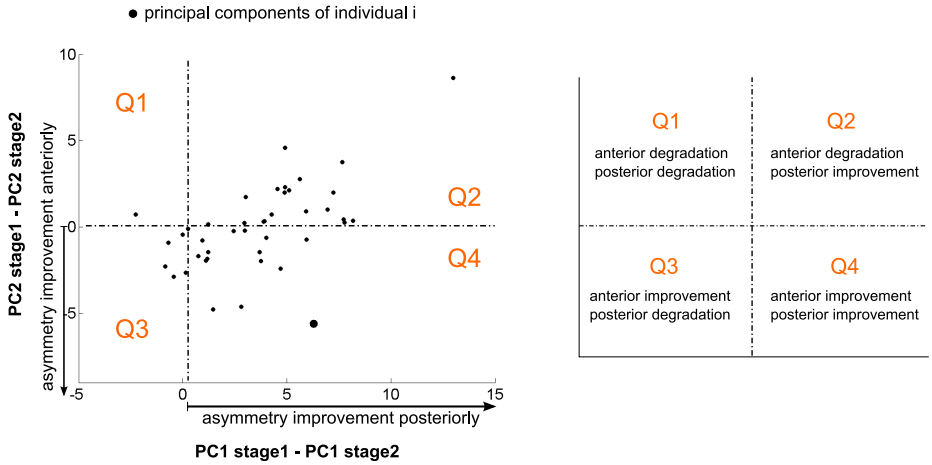


Figure 5.13: Treatment evaluation using principal components of the asymmetry model.

b) Using the principal components of the statistical asymmetry model

As seen in section 5.5.3, the principal components of the asymmetry model were in good agreement with the asymmetry change results (Figure 5.11). The principal components are then employed to quantify the posterior and anterior asymmetry changes. The latter are estimated by calculating the difference of the principal components of modes i within each patient:

$$\mathrm{pc}_{\mathrm{i,change}} = \mathrm{pc}_{\mathrm{i,before}} - \mathrm{pc}_{\mathrm{i,after}} \tag{5.11}$$

Hence, positive value reveals posterior improvement (first mode) and anterior degradation (second mode). The results are shown in Figure 5.13. The amount of posterior and anterior asymmetry changes can be read off the x- and y-axes, respectively. Individuals with global (anterior and posterior) degradation are found in the upper left quadrant (Q1). Individuals with global improvement are found in the lower-right quadrant (Q4). The helmet therapy appears to be more successful posteriorly than anteriorly:

- the amount of posterior improvement is larger than the amount of anterior improvement.
- the number of infants with posterior improvement is higher than the number of infants with anterior improvement.

5.6 Validations

The results obtained previously need to be validated, i.e. the strength of the relation between the results and the clinical parameters need to be checked. This

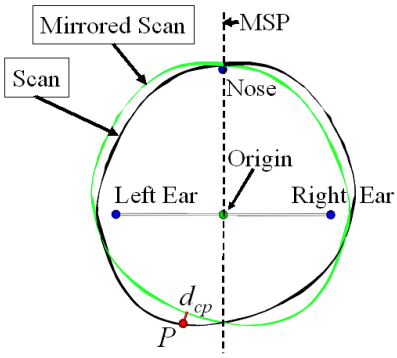


Figure 5.14: Alternative, "naive" method of asymmetry computation in an axial view.

is achieved by calculating the correlation coefficient (more details are given in Appendix B).

5.6.1 Validation of the asymmetry quantification

The asymmetry of all 42 head surfaces before treatment was re-computed using an independent method illustrated in Figure 5.14, using the original scans without template matching. This alternative, "naive" method of asymmetry computation, defines asymmetry as the closest point distance d_{cp} (Euclidean distance) between a head surface and its own mirror surface.[3] Flat maps from both methods were created and correlated against each other (R = 0.81, Table 1,V1). Furthermore, the computed asymmetry (average over the helmet region) was compared to clinical anthropometric measurements given by $|1-(\frac{RALP}{LARP})|$ where RALP is the diagonal distance from a right anterior point to a left posterior point, and LARP is the opposite diagonal, yielding R = 0.82 (Table 5.1,V2). The clinical measurement was checked against visual ranking of global asymmetry, yielding R = 0.84 (Table 5.1,V3).

5.6.2 Validation of the statistical model

The success of the statistical asymmetry model depends on its ability to capture and describe clinically relevant information in a compact way (i.e, in the first few modes). First, the model's compactness was evaluated by calculating an asymmetry distance as $\sqrt{PC1^2 + PC2^2 + PC3^2}$ and correlating it against average asymmetry in the helmet region. It yielded R = 0.92 (Table 5.1,V4). Moreover, the relation between the "magnitude of posterior asymmetry" and "magnitude of anterior asymmetry", which seemed to correspond to the two main important

[3]The Euclidean distance between a point A (x_A, y_A, z_A) and a point B (x_A, y_A, z_A) is defined as: $d_{Euclid} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$

parameters describing head asymmetry in DP, and the two first principal components was checked and yielded R = -0.97 (Table 5.1,V5) and R = -0.94 (Table 5.1,V6) for posterior and anterior asymmetry, respectively.

5.6.3 Validation of the treatment evaluation

For treatment evaluation, two of the most important parameters could be stated as "improvement in posterior asymmetry" and "improvement in anterior asymmetry". In order to validate the correspondence between these parameters and the change in the two first principal scores (Figure 5.11.b), the latter were correlated against maximum posterior and anterior asymmetry, respectively, yielding
R = 0.93 (Table 5.1, V7) and R = 0.82 (Table 5.1, V8), respectively.

Table 5.1: Validations V1-8. The linear Pearson correlation coefficient R is given with its 95% confidence interval.

	Description	*R*	*95% confidence*
V1	Asymmetry vs. Asymmetry by "Naive" Method	0.81	[0.67, 0.89]
V2	Average Asymmetry vs. Clinical Measurement	0.82	[0.73, 0.88]
V3	Clinical Measurement vs. Visual Assessment	0.84	[0.76, 0.89]
V4	$\sqrt{PC1^2 + PC2^2 + PC3^2}$ vs. Average Asymmetry	0.92	[0.88, 0.95]
V5	PC1 vs. Posterior Asymmetry	0.93	[0.87, 0.96]
V6	PC2 vs. Anterior Asymmetry	0.93	[0.87, 0.96]
V7	$PC1_{\text{Change}}$ vs. Posterior Asymmetry Change	0.93	[0.87, 0.96]
V8	$PC2_{\text{Change}}$ vs. Anterior Asymmetry Change	0.82	[0.69, 0.90]

5.7 Discussion

5.7.1 Accuracy of the asymmetry model

The computed asymmetry corresponded well to clinically measured asymmetry, to visual ranking and to values obtained by an independent method (Table 5.1). The remaining differences can be understood when taking into account that the compared methods assess somewhat different aspects of head asymmetry, and contain observer errors. The asymmetry quantifications contained errors from a) acquisition system noise, b) facial expression/head stocking smoothness, c) manual landmarking, d) template matching. In [10] these errors were investigated by comparison of results from two sets of scans acquired minutes apart, showing that a,b,d were negligible. Intra- and inter- observer reproducibility of landmarking c) has been determined showing that the asymmetry quantification had acceptable error in 96% of scans. It is anticipated that this will improve further in the future by incorporating surface texture and automatic landmark detection.

5.7.2 Limitations due to the template matching and reference points

Limitations of the method of establishing point correspondence between scans were the use of the ears (that are often affected in DP) for the registration, and the use of constructed landmarks instead of anatomical landmarks on top of the head. None of these limitations seem to have severely affected a valid asymmetry results. The choice of reference midsagittal plane (using ear landmarks) affects the results, but makes more sense in a clinical application than calculating a mathematical reference plane as in e.g., [8].

5.7.3 Asymmetry model and treatment evaluation

The asymmetry model described well both the clinical observations and the asymmetry results of the regions that where most affected by DP. The interpretability of the modes permit their application to helmet therapy evaluation. The success of the asymmetry model is related to the less complex, "global" types of asymmetry variation present in the DP dataset. The statistical model of change in asymmetry was not able to clearly show the improvement anteriorly.

Chapter 6

Asymmetry measure in mice with Crouzon syndrome

6.1 Introduction

6.1.1 Purpose

An asymmetry measure, based on the deformation vectors resulting from nonrigid registration of a perfectly symmetric atlas image to a given subject image, was developed. The asymmetry measure was applied to locally quantify the asymmetry in Crouzon mice. Furthermore, it was applied to compare the asymmetry in a group of ten Crouzon mice to the asymmetry in a group of ten normal mice.

6.1.2 Crouzon syndrome

Crouzon syndrome was first described nearly a century ago when facial anomalies (calvarial deformities) and abnormal protrusion of the eyeball were reported in a mother and her son [9]. Later, the condition was characterized as a constellation of premature fusion of the cranial sutures (craniosynostosis), orbital deformity, underdeveloped upper-jaw (hypoplastic maxilla), beaked nose, crowding of teeth, and high arched or cleft palate (Figure 6.1. Crouzon syndrome is a rare genetic disorder. The heterozygous mutations in the gene encoding *fibroblast growth factor receptor type 2* (*FGFR2*) have been identified to be responsible for Crouzon syndrome [34]. Recently a mouse model was created to study one of these mutations ($FGFR2^{Cys342Tyr}$)[12] (c.f. Figure 6.2). Incorporating advanced small animal imaging techniques such as Micro CT, allows for detailed examination of the craniofacial growth disturbances. A recent study, performing linear measurements on Micro CT scans, proved the mouse model applicable to reflect the craniofacial deviations occurring in humans with Crouzon syndrome [30]. This study was extended

Figure 6.1: Boy with Crouzon syndrome [18].

to assess the local deformations between the groups by constructing a deformable shape and intensity-based atlas of wild-type (normal) mouse skulls [27]. Statistical models of the deformation fields indicated that the skulls of Crouzon mice were more asymmetric than the wild- type skulls [28].

Asymmetry is highly relevant for the syndrome since the full or partial fusion of cranial sutures at either side of the skull and at different times makes the skull grow asymmetrically.

6.2 Materials

Production of the $Fgfr2^{C342Y/+}$ and $Fgfr2^{C342Y/C342Y}$ mutant mouse (Crouzon mouse) has been previously described [12]. All procedures were in agreement with the United Kingdom Animals (Scientific Procedures) Act, guidelines of the Home Office, and regulations of the University of Oxford. For three-dimensional (3D) CT scanning, 10 normals (wild-type) and 10 $Fgfr2^{C342Y/+}$ specimens at six weeks of age (42 days) were sacrificed using Schedule I methods and fixed in 95% ethanol. They were sealed in conical tubes and shipped to the Micro CT imaging facility at the University of Utah. Images of the skull were obtained at approximately 46μm $\times$ 46μm $\times$ 46μm resolution using a General Electric Medical Systems EVS-RS9 Micro CT scanner. Figure 6.3 shows an example of the imaging data appearance.

6.3 Pre-processing: volume matching

The correspondence between left and right voxels is established by the same process as described in 5.3.1.

6.3.1 Symmetrical atlas

In a previous study [27], the atlas has been computed as an average of a group of Micro CT scans from healthy subjects. Figure 6.4 shows the perfectly symmetrical volume obtained by performing the same steps as in section 5.3.1.

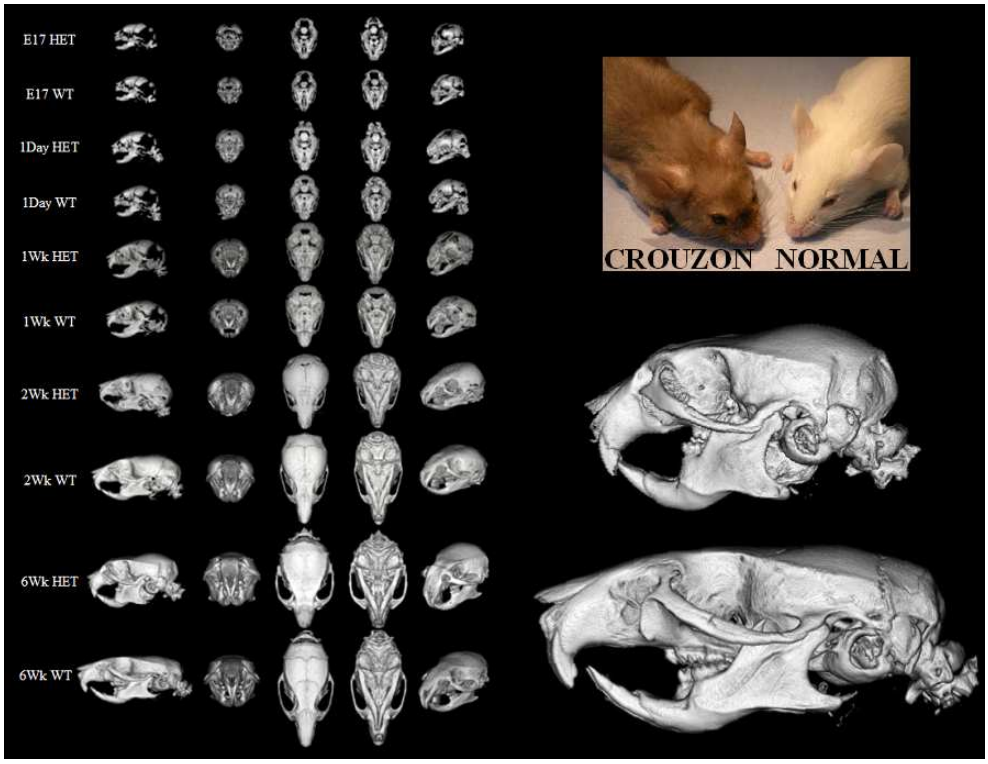
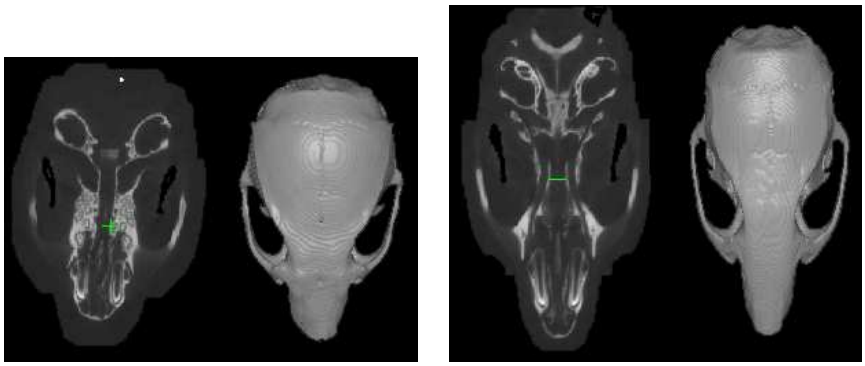


Figure 6.2: Data collection of the extracted surfaces from MicroCT scans of mice. Comparison between a normal mouse (*WT*) and a mouse with Crouzon Syndrome (*HET*) can be done at different age: embryon (*E17*), one day (*1Day*), one week(*1Wk*), two weeks(*2Wk*) and six weeks(*6Wk*). Upper right image: Courtesy of Dr. Chad Perlyn, Washington University School of Medicine, St.Louis, MO, USA.



a) b)

Figure 6.3: Data appearance of a) Crouzon mouse. b) wild-type mouse. The axial slice of the CT images (left) and the skulls extracted from the CT images are shown.

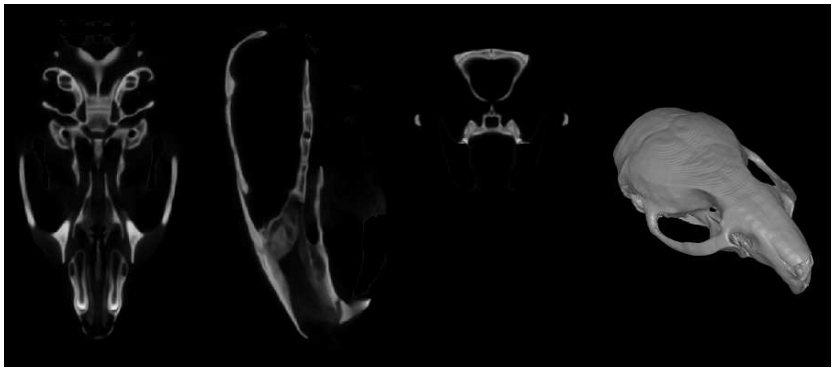


Figure 6.4: Perfectly symmetric atlas of wild-type mice: axial, sagittal and coronal slices through the data volume and a skull surface reconstruction (right).

6.3.2 Deformation using B-spline-based nonrigid registration

This symmetric atlas is deformed using B-spline-based non rigid registration combining global (affine) and local (B-splines) transformations. To establish a left/right correspondence for any mouse image, the B-spline-based nonrigid registration algorithm [35, 37] was used to create correspondence fields between the symmetric atlas and a subject image (Figure 6.5).

6.4 Methods: computation of asymmetry

For asymmetry calculations, only the local displacements at each point on the deformed symmetric atlas are considered, in order not to make pose and scaling differences affect the measure. The basic idea of the proposed asymmetry measure is to compare the displacement vector on one side to the corresponding displace-

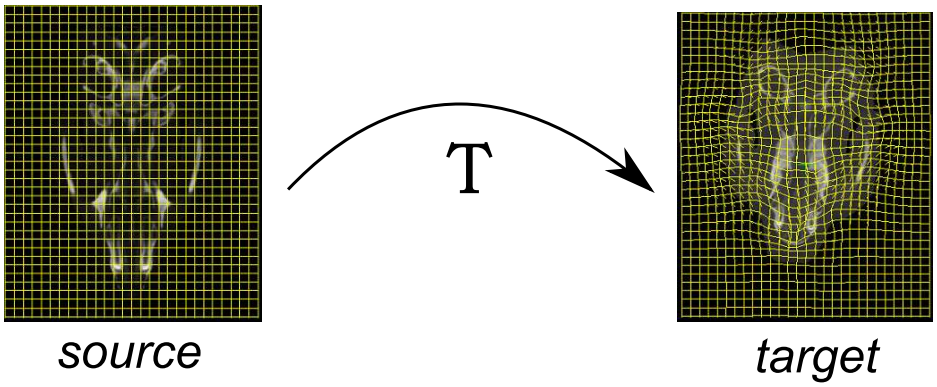


Figure 6.5: Registration of the perfectly symmetric image to a subject image.

ment vector on the other side. More formally, the asymmetry, A_P of a point P of one side involves the comparison of the local displacement vector, $\mathbf{v}_P$ in point P and the corresponding vector, $\mathbf{v}_{P'}$ in point P'. The approach taken here is to use the mirrored vector $\mathbf{v}^m_{P'}(x,y,z) = \mathbf{v}_{P'}(-x,y,z)$. The amount of asymmetry is then defined by the magnitude of the vector difference (Figure 6.6). Since, intuitively, the amount of asymmetry at P and P' should be equal, except for a sign introduced in order to distinguish between a point in an expanded region from a point in a depressed region, A_P and $A_{P\prime}$ are defined by the following:

$$\text{if } \mathbf{v}_P - \mathbf{v}^m_{P'} \text{points outwards then } A_P = ||\mathbf{v}_P - \mathbf{v}^m_{P'}|| \text{ and } A_{P\prime} = -A_P \quad (6.1)$$

$$\text{else } A_P = -||\mathbf{v}_P - \mathbf{v}^m_{P'}|| \text{ and } A_{P'} = -A_P \quad (6.2)$$

This, obviously, gives $A_P = 0$ if the original vectors are perfectly symmetric. This way of defining the direction of asymmetry is limited to surfaces extracted from the volume, since the surface normal is required. A volumetric measure could be obtained using the determinant of the Jacobian as in [16].

6.5 Results

6.5.1 Visualization of the results

In this study, the asymmetry results were only shown by coloring the 3D surfaces (without 2D projection). As explained in 5.5.1, each surface point P is colored according to its asymmetry value using a symmetric color map. Its scale ranges from blue (depressed) to red (expanded). Each value on the left side has an opposite value on the right side since $A_P = -A'_P$. Note that, in this case, the asymmetry measures are displayed in mm.

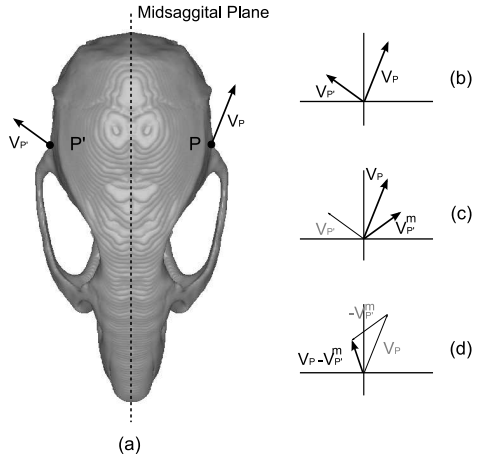


Figure 6.6: Schematic figure of vectors involved in asymmetry calculation. a) Displacement vectors shown on the symmetric atlas. b) Displacement vectors placed in the origin. c) Mirroring of $\mathbf{v}_{P\prime}$. d) Difference vector. The magnitude of the difference vector defines the absolute asymmetry, $|A_P|$.

6.5.2 Asymmetry results

Figure 6.7 presents the results of the asymmetry computations in three example subjects from each group of mice. Figure 6.8 provides a comparison of the groups in terms of absolute mean asymmetry.

6.6 Validation

Since the proposed method is based on results from image registration, the registration accuracy is essential for the method to be reliable. An extensive landmark validation using two sets of manual expert annotations as a gold standard was carried out in [27]. Landmark positions generated by the registration results were found to be non-significantly different from the gold standard and with significantly lower variance. To evaluate the asymmetry detection itself, we consider that the proposed method both localizes and quantifies asymmetry. In order to validate the two different aspects of the method, two approaches were taken. To evaluate the localization ability of the method, a clinical expert was asked to rate nine different regions of anatomical interest on the skull of the original Crouzon surfaces. Those were the nose (viewed from above and below), zygoma, anterior skull, mid skull, posterior skull, basal maxilla, anterior cranial base and posterior cranial base. The expert marked each region by 0 or 1 depending on whether the given region was symmetric or asymmetric, relatively. Similar ratings were obtained from the automatic method where regions with $|A_P| > 0.25$ mm were marked by 1 and the remaining regions by 0. Figure 6.9.a gives the number of

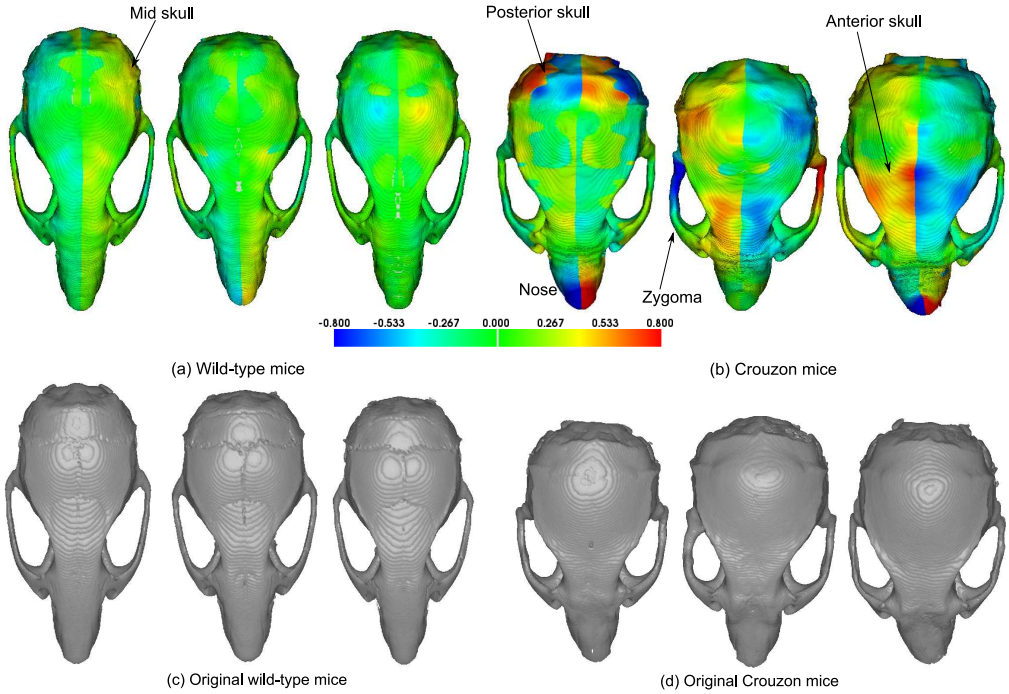
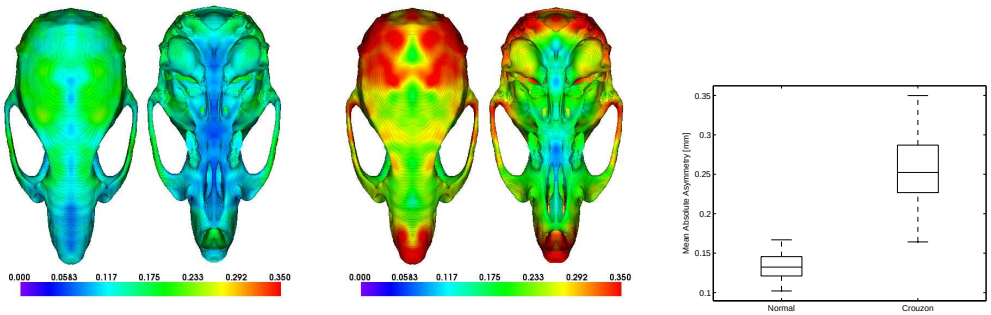


Figure 6.7: Example results for a) three wild-type mice and b) three Crouzon mice displayed on the deformed symmetric atlas. The scale ranges from blue (depressed) to red (expanded). Notice that $A_P = -A'_P$. For a visual comparison the corresponding original surfaces of c) the wild-type mice and (d) the Crouzon mice are shown.

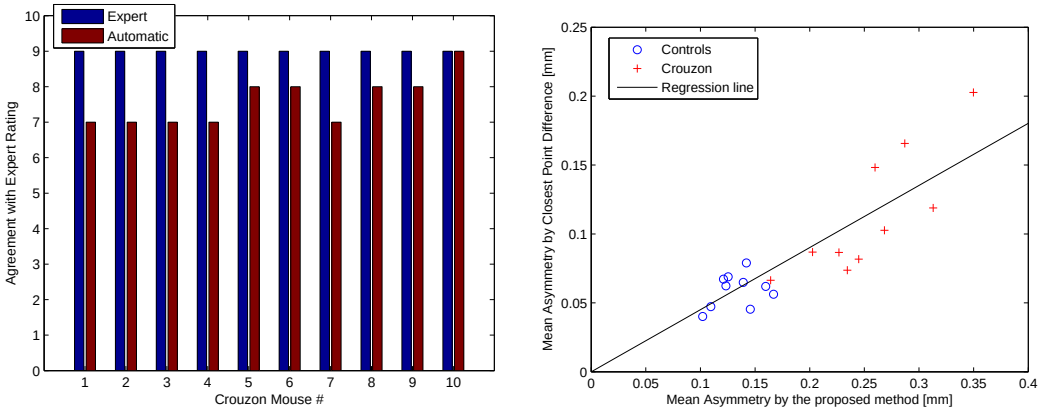
regions where the automatic approach and the expert rating agreed. Validating



a) b) c)

Figure 6.8: Difference between groups. Mean absolute asymmetry of a) wild-type and b) Crouzon mice, displayed on the symmetric atlas in top and bottom views. Note that the colorscale is different from the one in Figure 6.7. c) Global mean absolute asymmetry in Crouzon mice and control mice compared in a box plot.

the quantification ability of the asymmetry measure is more problematic, since a gold standard is not present. Here we take the approach to compare our method to a simple, crude measure of asymmetry. The original surfaces of all subjects were mirrored and closest point difference to the original surface was calculated. This method provides no point correspondences and is therefore not exact but the differences should correlate with the asymmetry values calculated by the proposed method. This is shown in Fig. 6.9(b).



a) b)

Figure 6.9: a) Validation with respect to expert rating. The point of reference (or the gold standard) is defined by the expert (blue bars), which indeed agrees with herself in rating of the nine regions for each mouse. The red bars denote the number of regions where the automatic method agrees with the expert. b) Correlation of mean asymmetry using the proposed method vs. a closest point difference approach. $R^2 = 0.75$.

6.7 Discussion

Figure 6.7.a shows that the three wild-type mice have a few asymmetric regions of up to approximately 0.5 mm in absolute asymmetry. This is not obvious by an amateur inspection of these regions on the original surfaces in Figure 6.7.c. The level of detection would be higher for a clinical expert. However, for the more asymmetric Crouzon mice, one can easily detect asymmetry on the original surfaces by the eye in the nose, zygoma and posterior skull (Figure 6.7.d. This is in good agreement with the automatic approach in Figure 6.7.b. The asymmetry of the anterior skull detected by the proposed method is closer to the symmetry plane, and is harder to confirm by the eye.

Figure 6.8 shows that the two groups differ considerably in terms of absolute

asymmetry. From Figure 6.8.a and b we note that the trend is similar, i.e. asymmetry appears in similar regions, apart from the nose seen from above. However, as expected, the Crouzon mice have much higher degree of asymmetry. This is confirmed in the box plot in Figure 6.8.c, which indicates that on average, Crouzon mice are more asymmetric than wild-type mice. This is also confirmed by a t-test on the absolute mean asymmetries (p-value of 10^{-6}). This is exactly what was expected from the analysis, i.e. the premature fusion of the cranial sutures leads to an asymmetric skull.

Figure 6.9.a shows that the localization ability of the automatic approach is in excellent agreement with the clinical expert. Most of the cases where the automatic method did not agree with the expert, were borderline, i.e. the asymmetry was just below or just above the selected threshold. The threshold of 0.25 approximately corresponds to the regions just turning into yellow in Fig. 6.7(b). Obviously, the choice of threshold is extremely important and the most correct way would probably be to use multiple parameter hypothesis testing to determine the threshold of significant asymmetry. For validation purposes, we believe that the fact that we are using the same threshold for all mice is important. For clinical practice it is perhaps even useful to be able to tune the threshold (or the range of the colorscale) with respect to the different experts' philosophical definitions of asymmetry.

Figure 6.9.b shows that the proposed method gives a relatively good correlation with a crude measure of asymmetry, with an R^2 of 0.75. The fact that the correlation is not higher is understandable due to the closest point difference approach's lack of point correspondences. Nevertheless, it definitely shows that the two methods have the same trend and we believe that this is a good indication of that the quantification is correct.

Chapter 7

Conclusion

In the present project, a new framework for quantitative assessment of head asymmetry has been developed and presented. The main advantages of the two developed methods are their capabilities of quantifying both the *amount* and *spatial location* of asymmetry. They were seen to perform well providing clinical parameters of interest and were validated. They were also seen to be suitable for diagnosis purpose and population classification.

The cranial asymmetry has been computed and analyzes in two particular applications: A) in infants with deformational plagiocephaly (Chapter 5) and, B) in mice with Crouzon syndrome (Chapter 6). Both studies have been published in scientific articles ([21], [19], [29] and [20]). Furthermore, the procedure for quantification of asymmetry in infants with deformational plagiocephaly is currently being used at the Department of Plastic and Reconstructive Surgery at St. Louis Children's Hospital, MO, USA, for research purpose.

While the developed methodology has been applied to two particular types of craniofacial anomalies, it is easily transferable to other types of craniofacial anomalies and may be used for measurement of head asymmetry. The method employed for deformational plagiocephaly may be preferred when the data consists of for application to triangulated surfaces, e.g. as acquired by a laser scanner or stereophotogrammetric system. Similarly, the method developed for Crouzon syndrome may be applied to heads where the data are present in terms of volumetric voxel data, e.g. from a CT or MR scanners.

Bibliography

[1] American academy of pediatrics task force on sudden infant death syndrome. *Pediatrics*, 116:1245–1255, 2005.

[2] URL `http://www.cs.unc.edu/~marc/tutorial/node3.html`.

[3] URL `http://www.spineuniverse.com/displaygraphic.php/320/dp_022101bridwell_fig1-BB.gif`.

[4] URL `www.aafp.org`.

[5] URL `http://www.headsupbaby.com/images/`.

[6] URL `http://en.wikipedia.org/wiki/Symmetry_%28biology%29`.

[7] F. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge, 1997.

[8] G. Christensen, G. Johnson, T.A. Darvann, N.V. Hermann, and J. Marsh. Midsagittal surface measurement of the head: an assessment of craniofacial asymmetry. In *The International Society for Optical Engineering*, volume 3661, pages 612–619. SPIE, 1999.

[9] O. Crouzon. Une nouvelle famille atteinte de dysostose cranio-faciale héréditère. *Bull Mem Soc Méd Hôp Paris*, 39:231–233, 1912.

[10] T.A. Darvann, N.V. Hermann, M.J. Tenenbaum, D. Govier, S. Naidoo, P. Larsen, S. Kreiborg, and A.A. Kane. Head shape development in positional plagiocephaly: Methods for registration of surface scans. In *"Craniofacial Image Analysis for Biology, Clinical Genetics, Diagnostics and Treatment", Workshop of the 9th MICCAI conference*, pages 59–66, Copenhagen, Denmark, 2006. Springer, T.A. Darvann, N.V. Hermann, P. Larsen, S. Kreiborg (eds.).

[11] I.L. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, Chichester, 1998. xx + 347 pp.

[12] V. P. Eswarakumar, M. C. Horowitz, R. Locklin, G. M. Morriss-Kay, and P. Lonai. A gain-of-function mutation of fgfr2c demonstrates the roles of this receptor variant in osteogenesis. *Proc Natl Acad Sci, U.S.A.*, 101:12555–12560, 2004.

[13] J.V. Hajnal, D.L.G Hill, and D. Hawkes. *Medical Image Registration*, volume 5. CRC Press, 2001.

[14] H. Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24:498–520, 1933.

[15] P. Hummel and D. Fortado. A parents' guide to improving head shape. *Adv.*

Neonatal Care, 5:341–342, 2005.

[16] S. Joshi, P. Lorenzen, G. Gerig, and E. Bullitt. Structural and radiometric asymmetry in brain images. *Medical Image Analysis*, 7(2):155–170, 2003.

[17] J.C. Kolar and E.M. Salter. *Craniofacial Anthropometry. Practical Measurement of the Head and Face for Clinical, Surgical and Research Use*. Springfield, Illinois: Charles C. Thomas, Publisher, 1996.

[18] S. Kreiborg. *Crouzon Syndrome - A Clinical and Roentgencephalometric Study*, 1981. Doctorate thesis, Institute of Orthodontics, The Royal Dental College, Copenhagen.

[19] S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, and A.A. Kane. A method for evaluating treatment in infants with deformational plagiocephaly. In *Image Analysis in Vivo Pharmacology.*, 2007.

[20] S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, and A.A. Kane. Head asymmetry modelling and its application to treatment evaluation in infants with deformational plagiocephaly. In *Medical Imaging Computing and Computer-Assisted Intervention 2007. Submitted.*, 2007.

[21] S. Lanche, T.A. Darvann, H. Ólafsdóttir, N.V. Hermann, A.E. Van Pelt, D. Govier, M.J. Tenenbaum, S. Naidoo, P. Larsen, S. Kreiborg, R. Larsen, and A.A. Kane. A statistical model of head asymmetry in infants with deformational plagiocephaly. In Bjarne Kjær Ersbøll and Kim Steenstrup Pedersen, editors, *Image Analysis*, volume 4522 of *LNCS*, pages 898–907. Springer, 2007. ISBN 978-3-540-73039-2.

[22] W.T. Lee, K. Richards, J. Redhed, and F.A. Papay. A pneumatic orthotic cranial molding helmet for correcting positional plagiocephaly. *J. Craniofac. Surg.*, 17:139–144, 2006.

[23] S.R. Lele and J.T. Richtsmeier. *An Invariant Approach to Statistical Analysis of Shapes*. Chapman & Hall/CRC, 2001.

[24] T.R. Littlefield. Cranial remodeling devices: treatment of deformational plagiocephaly and postsurgical applications. *Semin. Pediatr. Neurol.*, 11:268–277, 2004.

[25] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press. ISBN 0-89791-227-6. doi: http://doi.acm.org/10.1145/37401.37422.

[26] H. Ólafsdóttir. Registration and analysis of myocardial perfusion MRI. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, feb 2004. URL `http://www2.imm.dtu.dk/pubdb/p.php?3154`.

[27] H. Ólafsdóttir, T. A. Darvann, N. V. Hermann, E. Oubel, B. K. Ersbøll, A. F. Frangi, P. Larsen, C. A. Perlyn, G. M. Morriss-Kay, and S. Kreiborg. Computational mouse atlases and their application to automatic assessment of

craniofacial dysmorphology caused by the crouzon mutation fgfr2ĉ342y. *Journal of Anatomy (accepted)*, 2007. URL `http://www2.imm.dtu.dk/pubdb/p.php?4934`.

[28] H. Ólafsdóttir, T.A. Darvann, N.V. Hermann, E. Oubel, B.K Ersbøll, A.F. Frangi, P. Larsen, C.A. Perlyn, G.M. Morriss-Kay, and S. Kreiborg. A craniofacial statistical deformation model of wild-type mice and Crouzon mice. In *International Symposium on Medical Imaging 2007, San Diego, CA, USA*. The International Society for Optical Engineering (SPIE), feb 2007. URL `http://www2.imm.dtu.dk/pubdb/p.php?4896`.

[29] H. Ólafsdóttir, S. Lanche, T.A. Darvann, N.V. Hermann, R. Larsen, B.K. Ersbøll, E. Oubel, A.F. Frangi, P. Larsen, C.A. Perlyn, G.M. Morriss-Kay, and S. Kreiborg. A point-wise quantification of asymmetry using deformation fields. application to the study of the crouzon mouse model. In *Medical Image Computing and Computer-Assisted Intervention (to appear)*, 2007. URL `http://www2.imm.dtu.dk/pubdb/p.php?5309`.

[30] C.A. Perlyn, V.B. DeLeon, C. Babbs, D. Govier, L. Burell, T.A. Darvann, S. Kreiborg, and G. Morriss-Kay. The craniofacial phenotype of the Crouzon mouse: Analysis of a model for syndromic craniosynostosis using 3D MicroCT. *Cleft Palate Craniofacial Journal*, 43(6):740–747, 2006.

[31] D. Perperidis, R.H. Mohiaddin, and D. Rueckert. Spatio-temporal free-form registration of cardiac MR image sequences. *Medical Image Analysis*, 9(5): 441–456, 2005. ISSN 13618415.

[32] L.H. Plank, B. Giavedoni, J.R. Lombardo, M.D. Geil, and A. Reisner. Comparison of infant head shape changes in deformational plagiocephaly following treatment with a cranial remolding orthosis using a noninvasive laser shape digitizer. *J. Craniofac. Surg.*, 17:1084–1091, 2006.

[33] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1993.

[34] W. Reardon, R. M. Winter, P. Rutland, L. J. Pulleyn, B. M. Jones, and S. Malcolm. Mutations in the fibroblast growth factor receptor 2 gene cause Crouzon syndrome. *Nat Genet*, 8:98–103, 1994.

[35] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. on Medical Imaging*, 18(8):712–721, 1999. ISSN 02780062.

[36] D. Rueckert, A. F. Frangi, and J. A. Schnabel. Automatic construction of 3D statistical deformation models of the brain using nonrigid registration. *IEEE Trans. on Medical Imaging*, 22(8):1014–1025, 2003.

[37] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G. Hill, and D. J. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. *Fourth Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*

(MICCAI '01), 2208:573–581, 2001. ISSN 03029743.

[38] R. Shekhar and V. Zagrodsky. Mutual information-based rigid and nonrigid registration of ultrasound volumes. *IEEE Trans. on Medical Imaging*, 21(1): 9–22, 2002. ISSN 02780062.

[39] M.B. Stegmann and D.D Gomez. A brief introduction to statistical shape analysis, mar 2002. URL `http://www2.imm.dtu.dk/pubdb/p.php?403`. Images, annotations and data reports are placed in the enclosed zip-file.

[40] C. Studholme, R.T. Constable, and J.S. Duncan. Accurate alignment of functional EPI data to anatomical MRI using a physics-based distortion model. *IEEE Trans. on Medical Imaging*, 19(11):1115–1127, 2000. ISSN 02780062.

[41] Thompson and W. D'Arcy. *On Growth and Form*. Cambridge University Press, 1917.

[42] William M. Wells, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996. ISSN 13618415.

Appendix A

Principal component analysis illustration on a set of hand shapes

To be more familiar with the Principal Components Analysis (PCA) (described in section 4), the statistical method for shape modelling is applied to a hand dataset of 40 images (observations). [39] gives gently access to this training set. Each hand shape was defined by 56 landmarks : 43 anatomical, 9 mathematical and 4 pseudo landmarks (Figure A.1). First the images are brought into correspondence using the latter landmarks and the 56 landmarks defining the hand shape are ordered columnwise as $[xxx \cdots yyy]$. The shape variation between the set of images is then obtained by carrying out a PCA. The hand shapes are projected into a new subspace. The basis is ordered such that each principal component is ranked according to variance. The three main variations are shown in Figure A.2. The

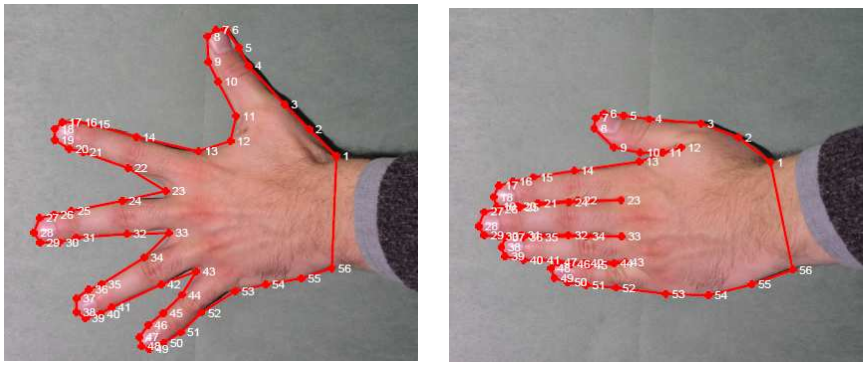


Figure A.1: Landmarks upon the hand contour on two images: 43 anatomical, 9 mathematical and 4 pseudo landmarks.

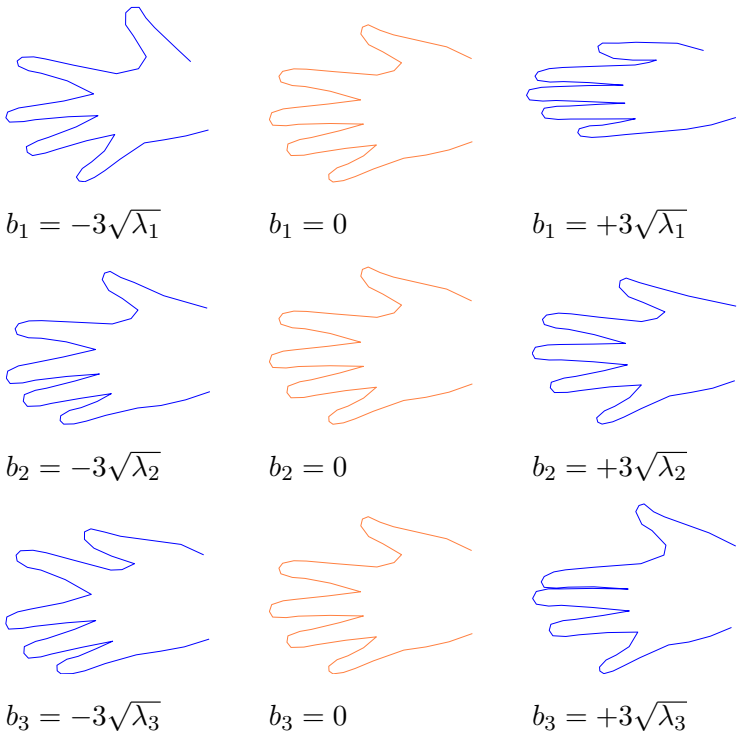
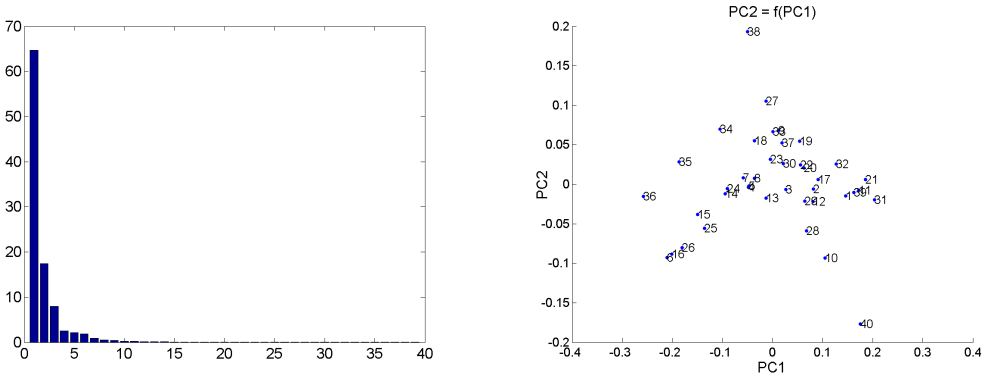


Figure A.2: Mean shape deformation using 1st, 2nd and 3rd principal mode at -/+3 standard deviation.

latter explain 92% of the shape variation, as shown in Figure A.3.a.

The first mode can be interpreted as how much the fingers are equally spread. Mode two refers to unevenly spread fingers, while mode three captures a correlation between increasing distance between the middle and index finger, and decreasing distance between the ring and little finger. In Figure A.2, the hands of -3 and +3 standard deviations of mode 1 are bigger and larger to the others, respectively. This is due to the fact that hands with fingers spread have a larger distance to their centroid and thus are scaled further down in the alignment process []. Figure A.3.b shows a projection of the two first principal components, commonly called scores. It permits to determine outliers, which, in this case, are set to be number 38 and 40. These two images do not follow the main variations detected in the other samples of the data set.



a) b)

Figure A.3: Eigenanalysis of the hand shape dispersion: a) Eigenvalues distribution. b) Projection of the two first principal components ($PC_2 vs. PC_1$).

Appendix B

Correlation coefficient

The correlation, or correlation coefficient, (r) indicates the strength and the direction of a linear relationship between two random variables X and Y:

$$r(x,y) = \frac{(\sum(X_i - \overline{X}))\ \sum(Y_i - \overline{Y})}{\sqrt{(}(\sum(X_i - \overline{X})^2)\ \sqrt{(}\ \sum(Y_i - \overline{Y})^2)} \tag{B.1}$$

where

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N}(X_i) \tag{B.2}$$

A correlation coefficient is a number between -1 and 1. It measures the degree to which the two variables X and Y are linearly related. A perfect linear relationship with positive slope gives a correlation of 1. Analogously, a perfect linear relationship with negative slope between the two variables gives a correlation coefficient of -1. The closer the coefficient is to either -1 or 1, the stronger the dependence between the variables. Two variables linearly independent have a correlation coefficient of 0 (but can have a relation).

Figure B illustrates different degree of linear dependence. Graphically, the correlation coefficient correspond to the straight line that best fit the plot between the two variables.

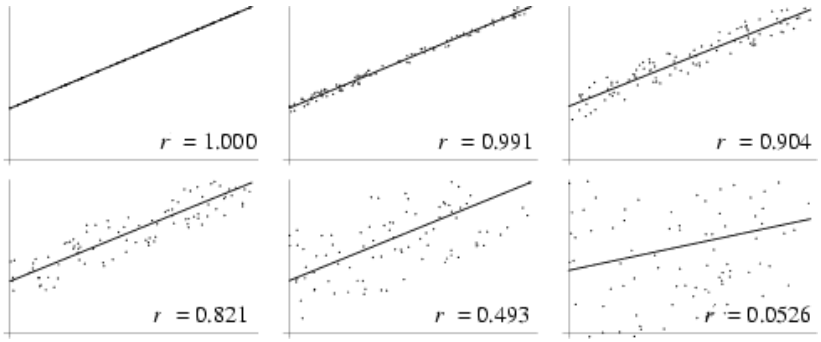


Figure B.1: Degree of linear dependence.

Appendix C

Implementation

C.1 Asymmetry computation - C using Visualization ToolKit(VTK)

```
#include "vtkSTLReader.h"
#include "vtkSTLWriter.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkPolyDataNormals.h"
#include "vtkThinPlateSplineTransform.h"
#include "vtkTransformPolyDataFilter.h"
#include "vtkPointDataToCellData.h"
#include "vtkUnstructuredGrid.h"
#include "vtkFloatArray.h"
#include "vtkIdList.h"
#include "vtkPoints.h"
#include "vtkMath.h"
#include "vtkCellLocator.h"

#include <fstream.h>
#include <iostream.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>
#include <math.h>
using namespace std;

// Program file name: asymmetry.cxx
// Purpose: compute distance between two surfaces (per cell), and output to color file.
// Author: Stephanie Lanche
// Date: October,2006
```

```
//----------------- Prototype of the functions-------------------//

void trig_center(vtkPolyData *Source, vtkPoints *Points, int _i, float *tcenter);


//----------------------Main Programm ---------------------------//
int main ( int argc, char *argv[] )
{
if (argc != 8)
{
cout << "Usage : Name_FirstScan.vtk Name_FinalScan.vtk helmet_region.clr
     Name_substraction_clr.clr Name_RatioFirstScan_clr.clr
 Name_RatioLastScan_clr.clr mask"  << endl;
cout << "Input : Name_FirstScan.vtk Name_FinalScan.vtk helmet_region.clr
         mask(y/n)" << endl;
cout << "Output : Name_substraction_clr.clr Name_RatioFirstScan_clr.clr
         Name_RatioLastScan_clr.clr" << endl;
return -1;
}

float begin_tcenter1[3], begin_tcenter2[3], begin_dright, begin_dleft;
float end_tcenter1[3], end_tcenter2[3], end_dright, end_dleft;
float min_subratio = 1000.0, max_subratio = 0.0, min_beginratio = 1000;
float max_beginratio = 0, min_endratio = 1000, max_endratio = 0;
int ncells=0,ncells_end=1;
double p0[3], p12[3], pP[3];
float *beginratio, *endratio, *subratio;
beginratio = new float[95338];
endratio = new float[95338];
subratio = new float[95338];
char mask_in;
bool mask;
float max_ratio = -2, min_ratio = 2, max_sub =-2, min_sub = 2;

//------ Mask? no Mask?
//cout << "Do you want to apply the topcol mask to the skull ? (y or n)" << endl;
//cin >> mask_in;
mask_in = *argv[7];
if (mask_in == 'n') mask = false;
else if (mask_in =='y')mask = true;
else {
cout << " Please answer by yes (y)  or no (n)" << endl;
return -1;
}


//----------- Open the output color files
ofstream out; // substraction between the ratio of the first and final scan
```

```
out.open(argv[4], ios::out);
out << "# LANDMARKER SURFACE SCALAR COLOR FILE V1" << endl;
out << "-0.11 0.11" << endl;
out << "rainbow_whitemid" << endl;
ofstream out2; // ratio of the first scan
out2.open(argv[5]);
out2 << "# LANDMARKER SURFACE SCALAR COLOR FILE V1" << endl;
out2 << "-0.25 0.25" << endl;
out2 << "rainbow_symmetric_whitemid" << endl;
ofstream out3; // ratio of the final scan
out3.open(argv[6]);
out3 << "# LANDMARKER SURFACE SCALAR COLOR FILE V1" << endl;
out3 << "-0.25 0.25" << endl;
out3 << "rainbow_symmetric_whitemid" << endl;

// Coordinates of the left_ear (p0) and right_ear (p12)
p0[0] = 154.94400;
p0[1] = 99.19290;
p0[2] = 50.92380;
p12[0] = 50.38600;
p12[1] = 99.19290;
p12[2] = 50.92380;

// Calculation of the point situated half way of both ears
pP[0] = (p0[0] + p12[0]) / 2;
pP[1] = p0[1];
pP[2] = p0[2];

// Read the first surface (VTK type)
vtkPolyDataReader *beginreader = vtkPolyDataReader :: New();
beginreader -> SetFileName(argv[1]);
beginreader -> Update();
vtkPolyData *BeginSource = beginreader -> GetOutput();

// Read the last surface (VTK type)
vtkPolyDataReader *endreader = vtkPolyDataReader :: New();
endreader -> SetFileName(argv[2]);
endreader -> Update();
vtkPolyData *EndSource = endreader -> GetOutput();

// Put BeginSource points in BeginPoints
vtkPoints *BeginPoints = BeginSource -> GetPoints();

// Put EndSource points in EndPoints
vtkPoints *EndPoints = EndSource -> GetPoints();

// Get the number of cells in the Source
ncells = BeginSource -> GetNumberOfCells();
ncells_end = EndSource -> GetNumberOfCells();
```

```
if (ncells != ncells_end)
{
cout << "Please select the correct first scan and end scan files" << endl;
cout << "Number of cells between the two files are not the same" << endl;
return -1;
}

// --------- Read the mask selecting the topcol of the skull  -------//

ifstream myfile (argv[3]); // file that will be readen
float *val;
val = new float[190676];
string line;
int cpt = 0 ;

if (mask == true) { // read the mask file only if the user has activated the mask

if (myfile.is_open())
{
while (! myfile.eof() )
           {
cpt ++;
getline(myfile,line);
if (cpt < 3) continue;
std::istringstream iss(line);
iss >> val[cpt-3];
}
myfile.close();
}

else cout << "Unable to open file";
}

//--------------Asymmetry mesasure for each cell of the skull
for (int i=0 ; i < ncells ; i++)
{
if ((val[i] != 1) && (mask == true))
{
out << "0.0" << endl;
out2 << "0.0" << endl;
out3 << "0.0" << endl;
}

if  ( (mask == false) || ( (val[i] == 1) && (mask == true) ) )
{

if (i<ncells/2)  //Left side of the skull
{
// ---------First scan
```

```
// Get the center of the cell i
trig_center(BeginSource, BeginPoints, i, begin_tcenter1);
//Get the center of the corresponding cell to i
trig_center(BeginSource, BeginPoints,(i+(ncells/2)), begin_tcenter2);

// calculating the distance between the center of these two triangles
// and the point P
begin_dright = sqrt( (pP[0]- begin_tcenter2[0])*(pP[0]- begin_tcenter2[0])
               + (pP[1]- begin_tcenter2[1])*(pP[1]- begin_tcenter2[1])
   + (pP[2]- begin_tcenter2[2])*(pP[2]- begin_tcenter2[2]));
begin_dleft = sqrt( (pP[0]- begin_tcenter1[0])*(pP[0]- begin_tcenter1[0])
              + (pP[1]- begin_tcenter1[1])*(pP[1]- begin_tcenter1[1])
  + (pP[2]- begin_tcenter1[2])*(pP[2]- begin_tcenter1[2]));

// Asymmetry calculation
if (begin_dleft > begin_dright)
beginratio[i] = -1*(1 - (begin_dright/begin_dleft));
else
beginratio[i] = 1 - (begin_dleft/begin_dright);

if (beginratio[i] > max_ratio) max_ratio = beginratio[i];
if (beginratio[i] < min_ratio) min_ratio = beginratio[i];

// ----------Last scan
//Get the points belonging to the cell i
trig_center(EndSource, EndPoints,i, end_tcenter1);
//Get the points belonging to the cell symmetric to i
trig_center(EndSource, EndPoints, (i+(ncells/2)), end_tcenter2);

// calculating the distance between the center of these two triangles
// and the point P
end_dright = sqrt( (pP[0]- end_tcenter2[0])*(pP[0]- end_tcenter2[0]) +
               (pP[1]- end_tcenter2[1])*(pP[1]- end_tcenter2[1]) +
   (pP[2]- end_tcenter2[2])*(pP[2]- end_tcenter2[2]));
end_dleft  = sqrt( (pP[0]- end_tcenter1[0])*(pP[0]- end_tcenter1[0]) +
               (pP[1]- end_tcenter1[1])*(pP[1]- end_tcenter1[1]) +
   (pP[2]- end_tcenter1[2])*(pP[2]- end_tcenter1[2]));
// Asymmetry Calculation
if (end_dleft > end_dright)
endratio[i] = -1*(1 - (end_dright/end_dleft));
else
endratio[i] = 1 - (end_dleft/end_dright);

//range of the displaied values
if (endratio[i] > max_ratio) max_ratio = endratio[i];
if (endratio[i] < min_ratio) min_ratio = endratio[i];

// Improvement in the asymmetry?
subratio[i] = fabs(beginratio[i]) - fabs(endratio[i]);
```

```
//range of the displaied values
if (subratio[i] > max_sub) max_sub = subratio[i];
if (subratio[i] < min_sub) min_sub = subratio[i];

//-----Writting the asymmetry measure in the ouput color files
out << subratio[i] << endl;
out2 << beginratio[i] << endl;
out3 << endratio[i] << endl;


} // end left side

else // Right side of the skull
{

// Writting the asymmetry measure in the ouput color files
out << subratio[i-(ncells/2)] << endl;
out2 << (-1) * beginratio [i-(ncells/2)] << endl;
out3 << (-1) * endratio [i-(ncells/2)] << endl;

}//end right side
}//end loop for each cells
}// end mask test


//close the output file
out.close();
out2.close();
out3.close();

ofstream out4;
out4.open("Range_values.txt", ios::app);
out4 << " Results for the "<< argv[1] << endl;
out4 << "min ratio :" << min_ratio << endl;
out4 << "max ratio :" << max_ratio << endl;
out4 << "min sub :" << min_sub << endl;
out4 << "min sub :" << max_sub << endl << endl;
out4.close();

return 0;

}

//------------------ Definition of the functions
void trig_center(vtkPolyData *Source, vtkPoints *Points, int _i, float *tcenter )
{
float p1[3], p2[3], p3[3];
int pId = 0;
vtkIdType cellId;
```

```
vtkIdList *cells = vtkIdList :: New();

Source -> GetCellPoints(_i,cells);
pId = cells -> GetId(0);
Points -> GetPoint(pId,p1);
pId = cells -> GetId(1);
Points -> GetPoint(pId,p2);
pId = cells -> GetId(2);
Points -> GetPoint(pId,p3);

tcenter[0] = ( p1[0] + p2[0] + p3[0] ) / 3.0;
tcenter[1] = ( p1[1] + p2[1] + p3[1] ) / 3.0;
tcenter[2] = ( p1[2] + p2[2] + p3[2] ) / 3.0;
}
```

C.2 Principal component analysis - Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        Principal Component Analysis (main program)
% This program run a statistical analysis using a PCA on a training
% set.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-- Initialization of the directories for reading and writing
directory = 'C:\Stephanie Lanche\My Documents\Data\';
outdirectory = 'Results\';

%-- Reading the data set: here, colorfiles containing the asymmetry values
%- Loading the mask for the helmet region
extension_mask ='helmet_region.clr';
g_mask = readclr_cat(directory,extension_mask);

%- Reading asymmetry results: before & after treatment
extension_begin = '*_1.clr';
g_begin = readclr_cat(directory,extension_begin);
[M,N] = size(g_begin);
g_begin = abs(g_begin .* repmat(g_mask,1,N));
extension_end = '*_2.clr';
g_end = readclr_cat(directory,extension_end);
g_end = abs(g_end .* repmat(g_mask,1,N));
g = [g_begin g_end];

%- asymmetry change
extension_diff = '*_asydiff.clr';
g_diff = readclr_cat(directory,extension_diff);
[M,N] = size(g_diff);
g_diff = g_diff .* repmat(g_mask,1,N);
```

```
%-- Computation of the Singular Value Decomposition
[phi_global, lambda_global, bg_global, gmean_global] = compute_pcasvd(g);
[phi_diff, lambda_diff, bg_diff, gmean_diff] = compute_pcasvd(g_diff);

%-- Computation of the nb_mode first modes
nb_mode = 10;
for i = 1:nb_mode
    %--- Gobal Analysis
filename_global = ['C:\Stephanie Lanche\PCA\' ...
            outdirectory 'global_' int2str(i)];
    compute_mode_deform(gmean_global, phi_global(:,i), lambda_global, i, ...
        filename_global);

    %--- Difference : Improvement of the asymmetry
filename_diff = ['C:\Stephanie Lanche\PCA\' ...
            outdirectory 'diff_' int2str(i)];
    compute_mode_deform(gmean_diff, phi_diff(:,i), lambda_diff, i, filename_diff);
end

writeclr_fromhalfskull ( 'global_mean_42.clr', gmean_global, ...
    'rainbow_symmetric_whitemid', max(max(gmean_global),abs(min(gmean_global))));
writeclr_fromhalfskull ( 'gmean_diff_42.clr', gmean_diff, ...
    'rainbow_symmetric_whitemid', max(max(gmean_diff),abs(min(gmean_diff))));

%---Display results : eigenvalues distribution, principal components & scores

%-Global analysis
figure, bar( [1:2*N], 100*lambda_global/sum(lambda_global));
xlabel = ('');
title('global eigenvalues distribution');
bplot_global = bg_global';
figure, display_pc(bg_global,1,2);
for i=1:2*N
    text(bg_global(1,i),bg_global(2,i),int2str(i),'Fontsize',15)
end
figure, diff_from_asym_model(bg_global,1,2);

%-Asymdiff scan
figure, bar( [1:N], 100*lambda_diff/sum(lambda_diff));
title('asymdiff scans eigenvalues distribution');
bplot_diff = bg_diff';
figure, display_pc2 (bg_diff,1,2);
for i=1:N
    text(bg_diff(1,i),bg_diff(2,i),int2str(i),'Fontsize',15)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              FUNCTIONS                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function X = readclr_cat(clrdir,extension)
%-------------------------------------------------------------------------%
% X = readclr_cat(clrdir,extension)
% This function reads all color files contained in the specified directory
% and concatenate it in a matrix
% Input: - clrdir: directory where are stored the color files
%        - extension:  extension of the files ('.clr')
% Output: X : matrix containing the data readen in the color files
%-------------------------------------------------------------------------%
clrdl = dir([clrdir extension]);
clrlength = length(clrdl);
X = []; %better to put the size if you know it X = zeros(txtlength,?)
for k = 1:clrlength
    c = readclr([clrdir, clrdl(k).name]);
    X = [X c(1:length(c)/2)];
end

%-------------------------------------------------------------------------%
function compute_mode_deform(xmean, phi, lambda, mode, filename)
%-------------------------------------------------------------------------%
% compute_mode_deform(xmean, phi, lambda, mode, filename)
% This function computes the deformation for the ith mode of variation. It
% writes the results in a colorfile.
% Input: - xmean: mean of the training set
%        - phi: matrix containg the eigenvectors (directions of
%        variation of the trainig set, ordered by )
%        - lambda: diagonal matrix containg the eigenvalues (amount of
%        variation of the training set, ordered by )
%        - mode: mode chosen for writing
%        - filename: name of the output file
% Output: file created by the function writeclr_fromhalfskull, displaying
% the variation corresponding to the mode chosen
%-------------------------------------------------------------------------%
[M, N] = size(xmean);
std = -3;

b_g = std * sqrt(lambda(mode));

x_deform =  phi * b_g;
maxi = max(x_deform);
mini = min(x_deform);
writeclr_fromhalfskull ( [filename '_std_1.clr'], x_deform, ...
    'rainbow_symmetric_whitemid', max(maxi,abs(mini)));

%-------------------------------------------------------------------------%
function [phi, lambda, bg, xmean] = compute_pcasvd(x)
%-------------------------------------------------------------------------%
% [phi, lambda, bg, xmean] = compute_pcasvd(x)
```

```
%
% compute_pcasvd computes the Principal Component Analysis using Singular
% Value Decomposition of the reduced covariance matrix (because of the large
% amount of data).
% The dataset has to be aligned in a preprocessing.
% # Input : - x : matrix of the dataset of size [M N] with M number of points
%           - N : number of observations
% # Output : - phi : matrix containg the eigenvalues
%            - lambda : matrix containg the eigenvalues of the dataset
%            - bg : shape model parameters
%            - xmean : mean shape of the dataset
%-------------------------------------------------------------------------%
[M,N] = size(x);

%-- Mean shape
xmean = mean(x,2);
xmean = imresize(xmean,[M N]);

%-- Maximum likelihood estimate of the covariance matrix
X = x - xmean;
if ( M < 2*N)
    %-- covariance matrix
sigma_cov = (X * X')/N;
[phi, lambda, phit] = svd(sigma_cov);
lambda = diag(lambda);
else
%-- X'X reduction of the dimension of the covariance matrix
sigma = (X' * X)/N;
[V, lambda, Vt] = svd(sigma);
%lambda = lambda * (N/M);
phi = X * V;
lambda = diag(lambda);
 end

phi = phi ./ repmat(lambda',M,1);
normphi = sqrt (sum(phi.*phi));
phi = phi./ repmat(normphi, M, 1);

%-- shape model parameters
bg = phi' * X;
xmean = xmean(:,1);

%-------------------------------------------------------------------------%
function writeclr_fromhalfskull(filename,c,clrtable,range)
%-------------------------------------------------------------------------%
% writeclr_fromhalfskull(filename,c,clrtable,range)
%
% This function writes two times the values contained in vector c to a color
% file with a specified colortable and a specified range.
```

```
% This permit to have the same values on both side of the skull.
% #Input: - filename : name of the color file with extension .clr
%         - c : vector
%         - clrtable : specific color table to display the values of c
%         - range : maximum value of the range (the minimum value is
%         defined as the opposite value
% #Output : color file (.clr)
%
% Example: writeclr_fromhalfskull('test.clr',c,'rainbow',0.60);
%--------------------------------------------------------------------------%

rangem = -1*range;
eval(sprintf('fid = fopen(''%s'', ''wt'');', filename));
fprintf(fid,'# LANDMARKER SURFACE SCALAR COLOR FILE V1\n');
fprintf(fid,'%2.6f %2.6f\n', rangem, range);
fprintf(fid,[clrtable,'\n']);

for j=1:length(c)
    fprintf(fid,'%4.6f\n',c(j));
end
% because the matrix data contains only the left part of the skull
for j=1:length(c)
    fprintf(fid,'%4.6f\n',c(j));
end
fclose(fid);

%--------------------------------------------------------------------------%
function display_pc(bg,mode1,mode2)
%--------------------------------------------------------------------------%
% display_pc(bg,mode1,mode2)
% This function plots the Principal Components contained in the matrix bg
% of mode1 vs. mode2. It was specialy written for the asymmetry values of
% the infant with DP. It will link for each patient, the pc before and
% after treatment. Patient nb 8 is diplayed in bold.
%--------------------------------------------------------------------------%
N = size(bg,1);
hold on
for i=1:N/2
    X= [bg(mode1,i) bg(mode1,N/2+i)];
    Y =[bg(mode2,i) bg(mode2,N/2+i)];
    if i ~= 8
        plot(X,Y,'k','LineWidth',0.5)
        plot(bg(mode1,i),bg(mode2,i),'g.','MarkerSize',16)
        plot(bg(mode1,N/2+i),bg(mode2,N/2+i),'r*','MarkerSize',9)
    else
        plot(X,Y,'k','LineWidth',2)
        plot(bg(mode1,i),bg(mode2,i),'g.','MarkerSize',34)
        plot(bg(mode1,N/2+i),bg(mode2,N/2+i),'r*','MarkerSize',16,'LineWidth',2)
    end
```

```
end
pcx = ['PC' int2str(mode1)];
pcy = ['PC' int2str(mode2)];
title([pcy ' = f(' pcx ')'],'FontSize',16)
xlabel(pcx,'FontSize',16)
ylabel(pcy,'FontSize',16)

hold off;

%--------------------------------------------------------------------------%
function display_pc2(bg,mode1,mode2)
%--------------------------------------------------------------------------%
% display_pc2(bg,mode1,mode2)
% This function plots the Principal Components contained in the matrix bg
% of mode1 vs. mode2.
%--------------------------------------------------------------------------%

N = size(bg,1);
hold on
for i=1:N
    plot(bg(mode1,i),bg(mode2,i),'k.','MarkerSize',16)
end
pcx = ['PC' int2str(mode1)];
pcy = ['PC' int2str(mode2)];
title([pcy ' = f(' pcx ')'],'FontSize',16)
xlabel(pcx,'FontSize',16)
ylabel(pcy,'FontSize',16)

hold off;
```

C.3 Analyze files (reading and writing) - Matlab

```
function [header, Image, byte_coding] = read_analyze(filename)
%-------------------------------------------------------------------------%
% [header, Image] = read_analyze(filename)
%
% # Input : - filename (with or without extension) of the analyze file to
%             be readen
%
% # Output : - header : structure containing the information written in the
%                       header file
%            - Image : 3D array containing the voxel values of the volume
%            - byte_coding : endian used to read the .hdr and .img files
%
% Related functions: read_hdr.m
%
% Author: Stphanie Lanche - 3D-Lab / IMM-DTU / CPE-Lyon
%-------------------------------------------------------------------------%
```

```
if (nargin~=1)
    error('Wrong number of input arguments');
end

if isempty(filename)
    error('Please give the name of the file to be read')
else
%-- Check if the filename has an extension or not
    if( strfind(filename, '.hdr') )
    filename_hdr = filename;
    filename_img = strrep(filename, 'hdr', 'img');
    elseif( strfind(filename, '.img') )
    filename_hdr = strrep(filename, 'img', 'hdr');
    filename_img = filename;
    else
    filename_hdr = [filename '.hdr'];
    filename_img = [filename '.img'];
    end
end

%--HEADER FILE: Read .hdr file
byte_coding = 'ieee-be'; % default endian- byte coding
second_time = 0;
header = read_hdr(filename_hdr,byte_coding,second_time);

if (header.header_key.sizeof_hdr < 348) | (header.header_key.sizeof_hdr > 16732)
    byte_coding = 'ieee-le'; % if the default endian choice not correct we use
                             % the little endian
    second_time = 1;
    header = read_hdr(filename_hdr,byte_coding,second_time);

    if (header.header_key.sizeof_hdr < 348) | (header.header_key.sizeof_hdr > 16732)
        error ('Problem in reading the file : not able to '...
          'read the file with little or big endian')
    end

end

%--IMAGE FILE: Read .img file
%-- Read the raw data in the .img file
%-- Get the dimension of the volume read in the header file
X = header.image_dimension.dimension(2);
Y = header.image_dimension.dimension(3);
Z = header.image_dimension.dimension(4);
datatype = get_datatype(header.image_dimension.datatype);

%-- Open, Read and Close the .img file
fid_img = fopen(filename_img,'r',byte_coding);
```

```
Image = fread(fid_img , X*Y*Z, datatype);
fclose(fid_img);

%-- Rearrange the elements read in the .img file
Image = reshape(Image, [Y X Z]); % reshape the matrix in the matlab way
Image = permute(Image, [2 1 3]); % Matlab read the files columnwise
Image = flipdim(Image,1);

%--------------------------------------------------------------------------%
function datatype = get_datatype(hdr_datatype)
%-- get the datatype in the header and give the right argument for fread
 switch hdr_datatype
   case 'DT_BINARY'
       datatype = 'ubit1=>logical';
   case 'DT_UNSIGNED_CHAR'
       datatype = 'uint8=>uint8';
   case 'DT_SIGNED_SHORT'
       datatype = 'int16=>int16';
   case 'DT_SIGNED_INT'
       datatype = 'int32=>int32';
   case 'DT_FLOAT'
       datatype = 'float32=>float32';
   case 'DT_COMPLEX'
       datatype = 'double=>double';
   case 'DT_DOUBLE'
       datatype = 'double=>double';
   case 'DT_RGB'
       datatype = 'float=>float';
end
return;
%--------------------------------------------------------------------------%
%--------------------------------------------------------------------------%

function hdr = read_hdr(filename_hdr, byte_coding, second_time)
%--------------------------------------------------------------------------%
% header = read_hdr(filename_hdr, byte_coding, second_time)
%
% # Input : - filename (with or without extension) of the analyze file to
%             be readen
%           - byte_coding : endian choosen for the opening of the file
%                           'ieee-be': big endian
%                           'ieee-le': little endian
%           - second_time : 0 (first time the function try to read the
%                               file filename.hdr)
%                           1 (second time the function try to read the
%                               file)
%
% # Output : - header : structure containing the information written in the
%                       header file
```

```
%
% Associated functions: - read_hk : read the structure header_key
%                       - read_dime : read the structure image_dimension
%                       - read_dh : read the structure data_history
%
% Author: Stphanie Lanche - 3D-Lab / IMM-DTU / CPE-Lyon
%--------------------------------------------------------------------------%

if (nargin~=3)
    error('Wrong number of input arguments');
end

fid_hdr = fopen(filename_hdr,'r',byte_coding);
if (fid_hdr == -1)
    error('File not found - may not exist')
end
hdr.header_key = read_hk(fid_hdr);
hdr.image_dimension = read_dime(fid_hdr,second_time);
hdr.data_history = read_dh(fid_hdr);
%num_bit = ftell(fid_hdr);
fclose(fid_hdr);
return;

%--------------------------------------------------------------------------%
function hk = read_hk(fid)
% read (struct) header key

hk.sizeof_hdr = fread(fid,1,'int32')';
hk.datatype     = deblank(fread(fid,10,'uchar=>char'))';
hk.db_name     = deblank(fread(fid,18,'uchar=>char'))';
hk.extents     = fread(fid,1,'int32')';
hk.session_error= fread(fid,1,'int16')';
hk.regular     = char(deblank(fread(fid,1,'uchar=>char')))';
hk.hkey_unused0 = deblank(fread(fid,1,'uchar=>char'))';
return;


%--------------------------------------------------------------------------%
function dime = read_dime(fid, second_time)
%----- read (struct) image_dimension

dime.dimension = fread(fid,8,'int16')';
fseek(fid,14,'cof');
% dime.unused8 = fread(fid,1,'int16')';
% dime.unused9 = fread(fid,1,'int16')';
% dime.unused10 = fread(fid,1,'int16')';
% dime.unused11 = fread(fid,1,'int16')';
% dime.unused12 = fread(fid,1,'int16')';
% dime.unused13 = fread(fid,1,'int16')';
```

```
% dime.unused14 = fread(fid,1,'int16')';
datatype     = fread(fid,1,'int16')';
switch datatype
   case int16(0)
       dime.datatype = 'DT_UNKNOWN';
   case int16(1)
       dime.datatype = 'DT_BINARY';
   case int16(2)
       dime.datatype = 'DT_UNSIGNED_CHAR';
   case int16(4)
       dime.datatype = 'DT_SIGNED_SHORT';
   case int16(8)
       dime.datatype = 'DT_SIGNED_INT';
   case int16(16)
       dime.datatype = 'DT_FLOAT';
   case int16(32)
       dime.datatype = 'DT_COMPLEX';
   case int16(64)
       dime.datatype = 'DT_DOUBLE';
   case int16(128)
       dime.datatype = 'DT_RGB';
   case int16(255)
       dime.datatype = 'DT_ALL';
   otherwise
       if (second_time == 1)
       error('wrong datatype');
       else
           dime.datatype = 'wrong datatype';
       end
end

dime.bitpix = fread(fid,1,'int16')';
dime.dim_un0 = fread(fid,1,'int16')';
pixdim   = fread(fid,8,'float')';
dime.pixdim     = pixdim(find(pixdim)); % to have only the non-zero values
dime.vox_offset = fread(fid,1,'float')';
fseek(fid,12,'cof');
% dime.funused1 = fread(fid,1,'float')';
% dime.funused2 = fread(fid,1,'float')';
% dime.funused3 = fread(fid,1,'float')';
dime.cal_max = fread(fid,1,'float')';
dime.cal_min = fread(fid,1,'float')';
dime.compressed = fread(fid,1,'float')';
dime.verified = fread(fid,1,'float')';
dime.glmax = fread(fid,1,'int32')';
dime.glmin = fread(fid,1,'int32')';
return;

%---------------------------------------------------------------------------%
```

```
function dh = read_dh(fid)
%---- read the structure data history :

dh.descrip = deblank(fread(fid,80,'uchar=>char'))';
dh.aux_file     = deblank(fread(fid,24,'uchar=>char'))';
orient     = fread(fid,1,'int8')';
switch orient
   case int16(0)
       dh.orient = 'transverse unflipped';
   case int16(1)
       dh.orient = 'coronal_unflipped';
   case int16(2)
       dh.orient = 'sagittal unflipped';
   case int16(3)
       dh.orient = 'transverse flipped';
   case int16(4)
       dh.orient = 'coronal flipped';
   case int16(5)
       dh.orient = 'sagittal flipped';
   otherwise
       dh.orient = 'orientation not specified';
end
dh.originator = deblank(fread(fid,10,'uchar=>char'))';
dh.generated = deblank(fread(fid,10,'uchar=>char'))';
dh.scannum     = deblank(fread(fid,10,'uchar=>char'))';
dh.patient_id = deblank(fread(fid,10,'uchar=>char'))';
dh.exp_date     = deblank(fread(fid,10,'uchar=>char'))';
dh.exp_time     = deblank(fread(fid,10,'uchar=>char'))';
dh.hist_unused0 = deblank(fread(fid,3,'uchar=>char'))';
dh.views     = fread(fid,1,'int32')';
dh.vols_added = fread(fid,1,'int32')';
dh.start_field = fread(fid,1,'int32')';
dh.field_skip = fread(fid,1,'int32')';
dh.omax         = fread(fid,1,'int32')';
dh.omin         = fread(fid,1,'int32')';
dh.smax         = fread(fid,1,'int32')';
dh.smin         = fread(fid,1,'int32')';
return;
%--------------------------------------------------------------------------%
%--------------------------------------------------------------------------%
%                       Writing                                            %
%--------------------------------------------------------------------------%
function write_analyze(filename, Image, regular, datatype, bitpix, ...
                       pixdim, byte_coding)
%--------------------------------------------------------------------%
% write_analyze(filename, Image, regular, datatype, bitpix, pixdim,
%               byte_coding)
%
```

```
% # Input : - filename (with or without extension)
%           - Image : 3D-matrix containing the values of the volume
%           - regular : (char ) to be written in the Analyze7.5 header
%                  must be 'r' to indicate that all images have the same
%                  size
%           - datatype : (short int) to be written in the Analyze7.5 header
%                        DT_BINARY        1 (Binary)
%                        DT_UNSIGNED_CHAR 2 (Unsigned character)
%                        DT_SIGNED_SHORT  4 (Signed short)
%                        DT_SIGNED_INT    8 (Signed integer)
%                        DT_FLOAT         16 (Floating point)
%                        DT_COMPLEX       32 (Complex)
%                        DT_DOUBLE        64 (double precision)
%           - bitpix :  (short int) values for the Analyze 7.5 header
%                        nb of bit per pixel 1,8,16,32 or 64
%           - pixdim[] : (short int) values for the Analyze 7.5 header
%                        specifies the voxel dimensions
%                        pixdim[1] : voxel width
%                        pixdim[2] : voxel height
%                        pixdim[3] : interslice distance
%           - byte_coding : ieee-be (big endian)
%                           ieee-le (little endian)
%
% # Output : The image (.img) and header(.hdr) files with the same name
%
% The other values to put in the header file are defined by default
% See the function write_hdr for more details
%
% Related functions: write_hdr.m
%
% Author: Stephanie Lanche - 3D-Lab / IMM-DTU / CPE-Lyon
%-------------------------------------------------------------------------%

if (nargin ~= 7)
    error('Please give all the input values')
end

%-- Check if the filename has an extension or not
if( strfind(filename, '.hdr') )
    filename_hdr = filename;
    filename_img = strrep(filename, 'hdr', 'img');
elseif( strfind(filename, '.img') )
    filename_hdr = strrep(filename, 'img', 'hdr');
    filename_img = filename;
else
    filename_hdr = [filename '.hdr'];
    filename_img = [filename '.img'];
end
```

```
%-------- Check the arguments-------%
if ~strcmp(regular,'r') % -- regular
    error('wrong value for "regular": impossible to write a volume where all'...
        'images dont have the same size');
end
%-- datatype
if (~strcmp(datatype,'DT_BINARY') & ~strcmp(datatype,'DT_UNSIGNED_CHAR') & ...
        ~strcmp(datatype,'DT_SIGNED_SHORT') & ~strcmp(datatype,'DT_SIGNED_INT') ...
        & ~strcmp(datatype,'DT_FLOAT') & ~strcmp(datatype,'DT_COMPLEX') ...
        & ~strcmp(datatype,'DT_DOUBLE') )
    error('wrong "datatype"');
else
    [hr_datatype, hr_nb_bit] = put_datatype(datatype);
end
%-- bitpix
if  (bitpix~=1) & (bitpix~=8) & (bitpix~=16) & (bitpix~=32) & (bitpix~=64)
    error('wrong value of "bitpix"');
end
%-- pixdim
if (size(pixdim,2)~=3)
    error ('wrong number of values in "pixdim"')
else
   pixdim = [0 pixdim 1 0 0 0];
end
%-- pixdim
if ~strcmp(byte_coding,'ieee-be') & ~strcmp(byte_coding,'ieee-le')
    error('wrong endian : please choose between the big (ieee-be) or' ...
        'little (ieee-le) endian')
end

%-- Write the header file
glmax = max(Image(:));
glmin = min(Image(:));
dim = size(Image);
dim = [4 dim 1 0 0 0]; %assume that the dimension of the database is 4
write_hdr(filename_hdr, byte_coding, regular,dim, hr_datatype, bitpix, ...
    pixdim, glmax, glmin);

%-- Write the .img file
Image = flipdim(Image,1);
Image = permute(Image, [2 1 3]); % Matlab read the files columnwise
Image = Image(:);
fid_img = fopen(filename_img,'w', byte_coding);

Image = fwrite(fid_img , Image , hr_nb_bit);
fclose(fid_img);

%--------------------------------------------------------------------------
```

```
function [hr_datatype, hr_bit] = put_datatype(datatype)
%-- get the datatype in the header and give the right argument for fread
switch datatype
   case 'DT_BINARY'
       hr_datatype = int16(1);
       hr_bit = 'ubit1';
   case 'DT_UNSIGNED_CHAR'
       hr_datatype = int16(2);
       hr_bit = 'uint8';
   case 'DT_SIGNED_SHORT' %
       hr_datatype = int16(4);
       hr_bit = 'int16';
   case 'DT_SIGNED_INT'
       hr_datatype = int16(8);
       hr_bit = 'int32';
   case 'DT_FLOAT'
       hr_datatype = int16(16);
       hr_bit = 'float32';
   case 'DT_COMPLEX'
       hr_datatype = int16(32);
       hr_bit = 'double';
   case 'DT_DOUBLE'
       hr_datatype = int16(64);
       hr_bit = 'double';
end

return;
%--------------------------------------------------------------------------%
%--------------------------------------------------------------------------%
function write_hdr(filename, byte_coding, regular, dim, datatype, ...
    bitpix, pixdim, glmax, glmin)
%--------------------------------------------------------------------------%
% write_hdr(filename, byte_coding, regular, dim, datatype, bitpix, pixdim,
%           glmax, glmin)
%
%   # Input : - filename : name of the header file
%             - byte_coding : endian for the opening file
%             - regular : 'r' to confirm that all images have the same size
%                         in the volume
%             - dim : dimension of the images forming the volume
%             - datatype : type of data
%             - bitpix : number of bit per pixels (1,8,16,32,64)
%             - pixdim : dimension of the pixels in mm (width, height,
%                        thickness)
%             - glmax : maximum value in the volume
%             - glmin : minimum value in the volume
%  # Output : header files : filename.hdr
%
% Other values that are fixed:
```

```
%       - sizeof_hdr = 348
%         - extends = 16384
%         - orient = 0
%
% Author: Stephanie Lanche - 3D-Lab / IMM-DTU / CPE-Lyon
%-------------------------------------------------------------------------%

if (nargin ~= 9)
    error('Wrong number of input arguments');
end

%Opening the header file
fid = fopen(filename,'w', byte_coding);

% Header key
fwrite(fid,348,'int32');                    % sizeof_hdr
fwrite(fid,blanks(10),'uchar');             % datatype[10]
fwrite(fid,blanks(18),'uchar');             % db_name[18]
fwrite(fid,16384,'int32');                  % extents
fwrite(fid,0,'int16');                      % session_error
fwrite(fid,'r','uchar');                    % regular
fwrite(fid,' ','uchar');                    % hkey_un0

% Image dimension
fwrite(fid,dim,'int16');              % dim
fwrite(fid,0,'int16');                % unused8
fwrite(fid,0,'int16');                % unused9
fwrite(fid,0,'int16');                % unused10
fwrite(fid,0,'int16');                % unused11
fwrite(fid,0,'int16');                % unused12
fwrite(fid,0,'int16');                % unused13
fwrite(fid,0,'int16');                % unused14
fwrite(fid,datatype,'int16');         % datatype
fwrite(fid,bitpix,'int16');           % bitpix
fwrite(fid,0,'int16');                % dim_un0
fwrite(fid,pixdim,'float');           % pixdim[8]
fwrite(fid,0,'float');                % funused1
fwrite(fid,0,'float');                % funused2
fwrite(fid,0,'float');                % funused3
fwrite(fid,0,'float');                % vox_offset
fwrite(fid,0,'float');                % cal_max
fwrite(fid,0,'float');                % cal_min
fwrite(fid,0,'float');                % compressed
fwrite(fid,0,'float');                % verified
fwrite(fid,glmax,'int32');            % glmax
fwrite(fid,glmin,'int32');            % glmin

% Data History
fwrite(fid,blanks(80),'uchar');         % descrip[80]
```

```
fwrite(fid,blanks(24),'uchar');          % aux_file[24]
fwrite(fid,blanks(1),'uchar');           % orient
fwrite(fid,blanks(10),'uchar');          % originator[10]
fwrite(fid,blanks(10),'uchar');          % generated[10]
fwrite(fid,blanks(10),'uchar');          % scannum[10]
fwrite(fid,blanks(10),'uchar');          % patient_id[10]
fwrite(fid,blanks(10),'uchar');          % exp_date[10]
fwrite(fid,blanks(10),'uchar');          % exp_time[10]
fwrite(fid,blanks(3),'uchar');           % hist_un0[3]
fwrite(fid,0,'int32');                     % views
fwrite(fid,0,'int32');                     % vols_added
fwrite(fid,0,'int32');                     % start_field
fwrite(fid,0,'int32');                     % field_skip
fwrite(fid,0,'int32');                     % omax
fwrite(fid,0,'int32');                     % omin
fwrite(fid,0,'int32');                     % smax
fwrite(fid,0,'int32');                     % smin

%closing the header file
fclose(fid);
```

Appendix D

Data collection of asymmetry results in infants with DP

D.1 Asymmetry results for patient 1

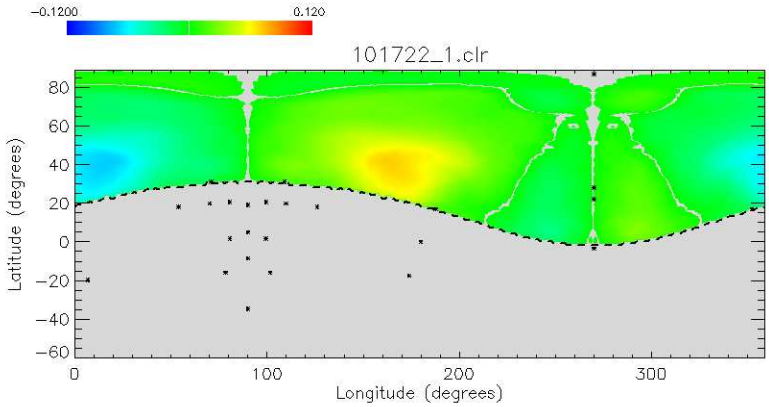


Figure D.1: Flatmap of asymmetry results before helmet therapy.

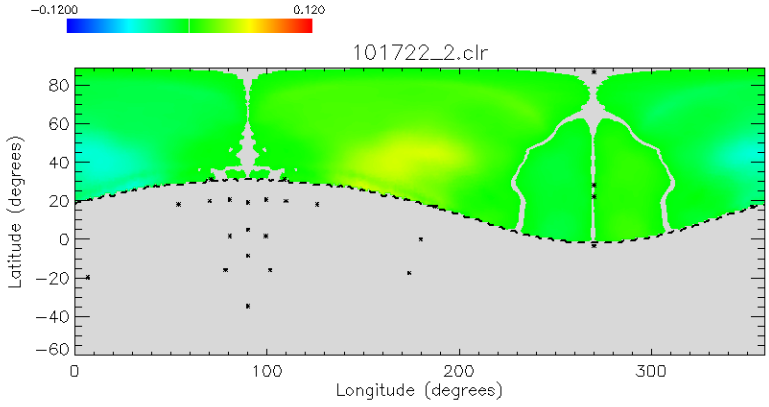


Figure D.2: Flatmap of asymmetry results after helmet therapy.

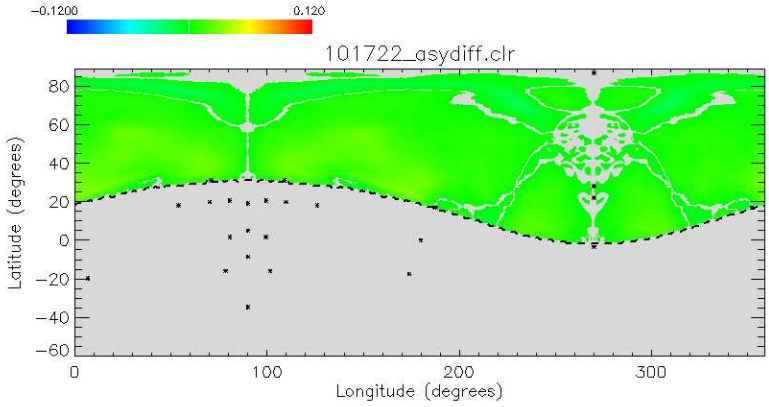


Figure D.3: Flatmap of asymmetry changes from before to after the therapy.

D.2 Asymmetry results for patient 2

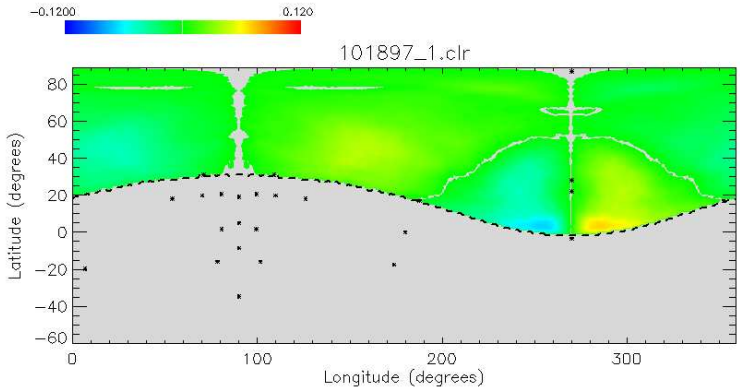


Figure D.4: Flatmap of asymmetry results before helmet therapy.

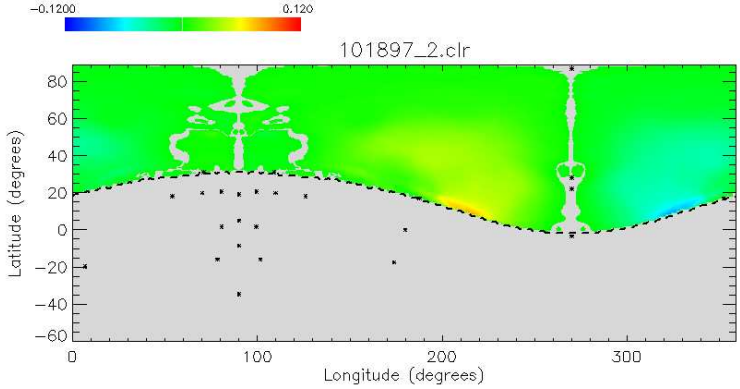


Figure D.5: Flatmap of asymmetry results after helmet therapy.

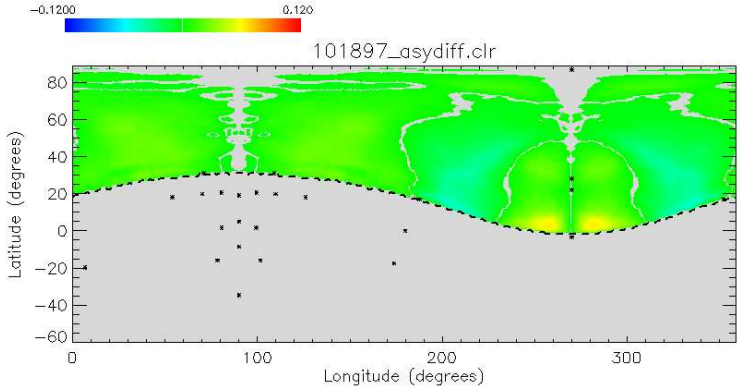


Figure D.6: Flatmap of asymmetry changes from before to after the therapy.

D.3 Asymmetry results for patient 3

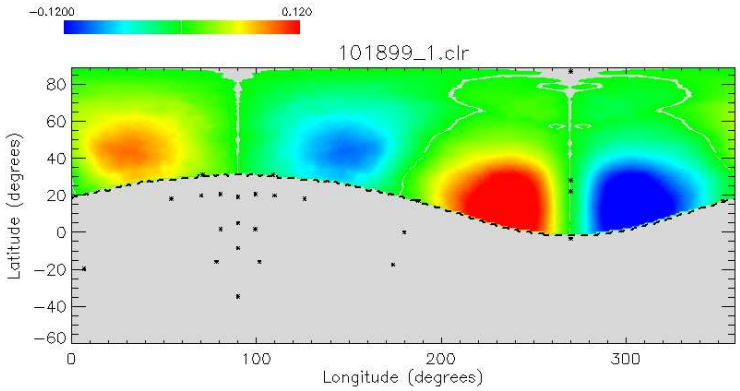


Figure D.7: Flatmap of asymmetry results before helmet therapy.

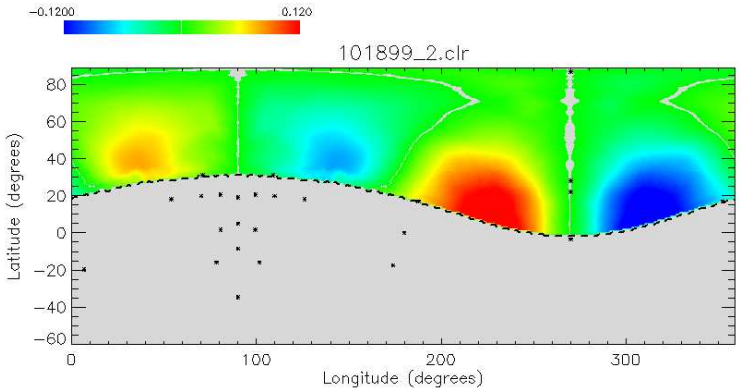


Figure D.8: Flatmap of asymmetry results after helmet therapy.

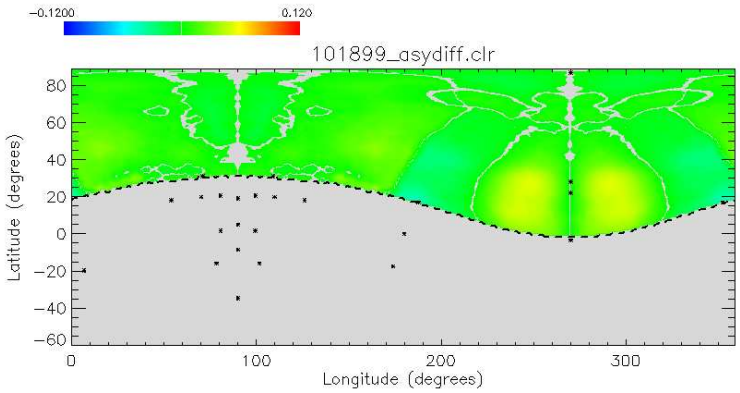


Figure D.9: Flatmap of asymmetry changes from before to after the therapy.

D.4 Asymmetry results for patient 4

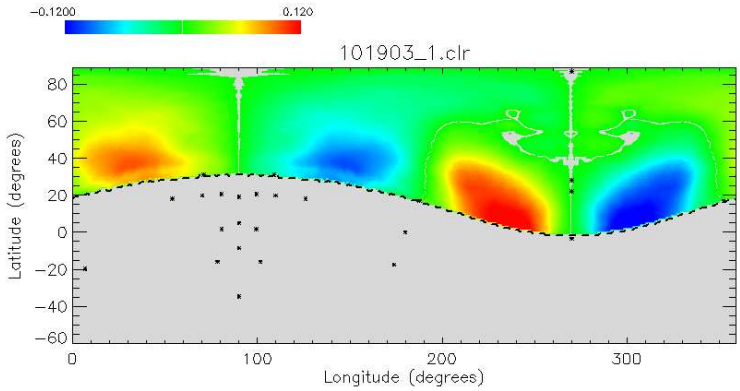


Figure D.10: Flatmap of asymmetry results before helmet therapy.

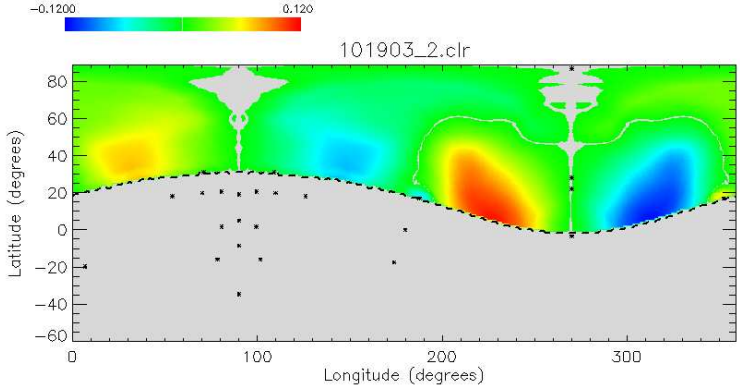


Figure D.11: Flatmap of asymmetry results after helmet therapy.

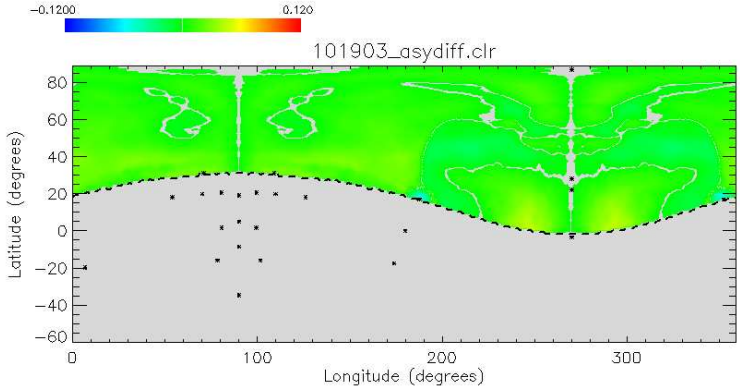


Figure D.12: Flatmap of asymmetry changes from before to after the therapy.

D.5 Asymmetry results for patient 5

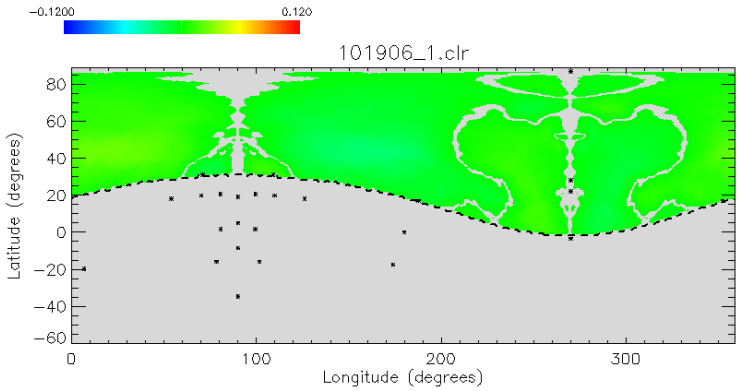


Figure D.13: Flatmap of asymmetry results before helmet therapy.

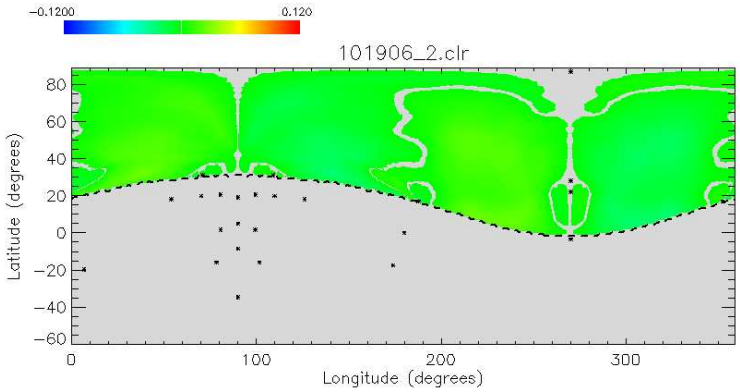


Figure D.14: Flatmap of asymmetry results after helmet therapy.

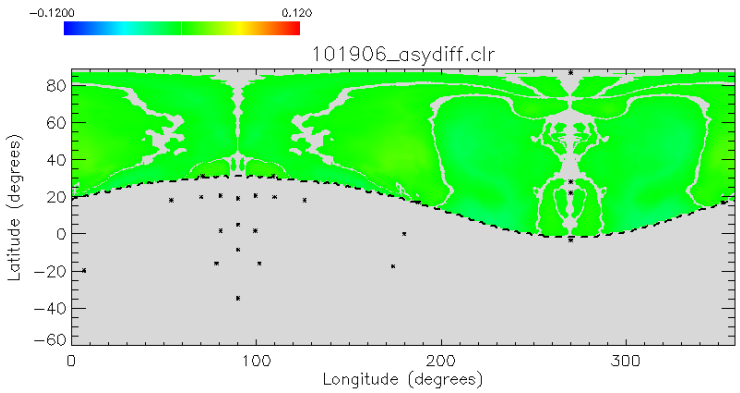


Figure D.15: Flatmap of asymmetry changes from before to after the therapy.

D.6 Asymmetry results for patient 6

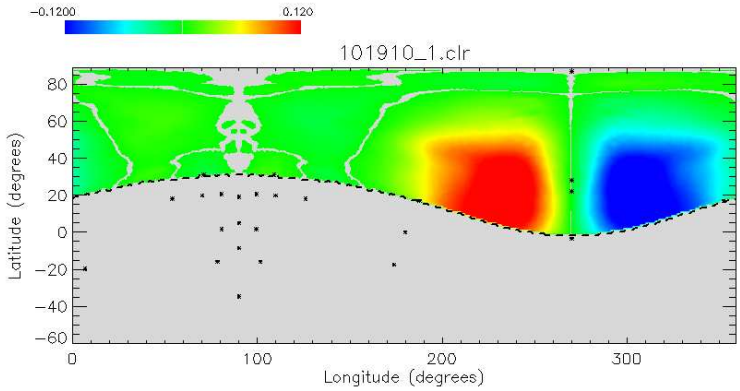


Figure D.16: Flatmap of asymmetry results before helmet therapy.

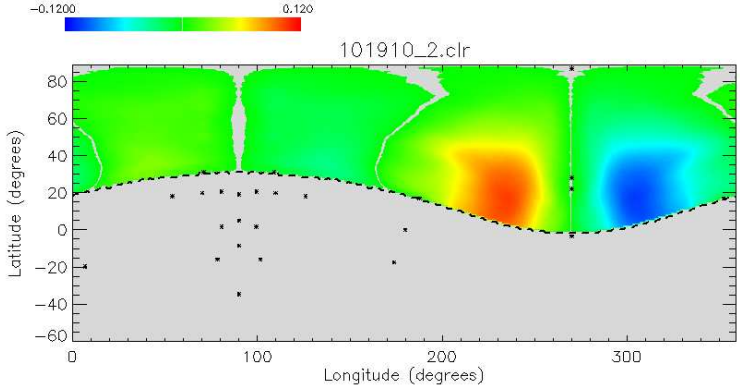


Figure D.17: Flatmap of asymmetry results after helmet therapy.

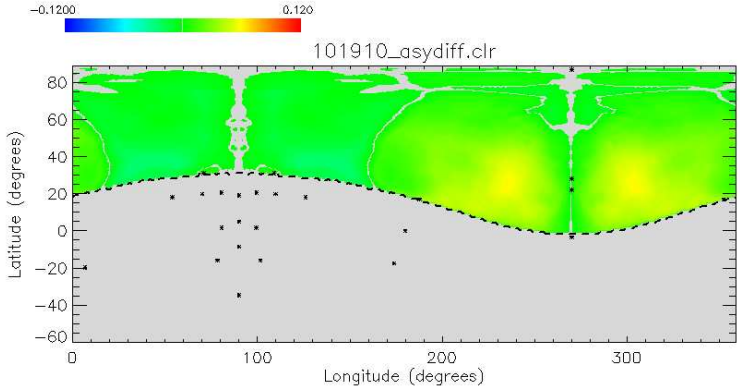


Figure D.18: Flatmap of asymmetry changes from before to after the therapy.

D.7 Asymmetry results for patient 7

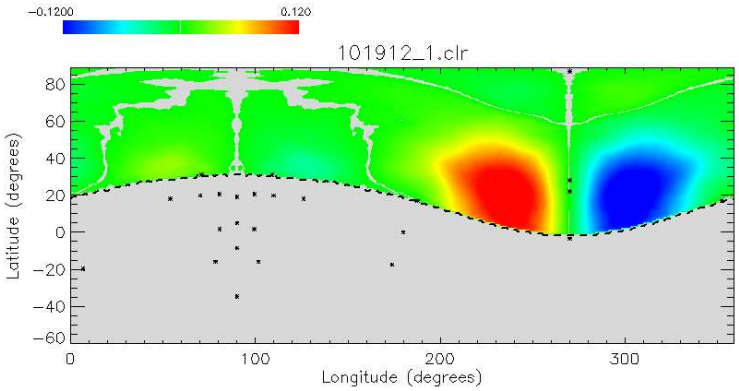


Figure D.19: Flatmap of asymmetry results before helmet therapy.

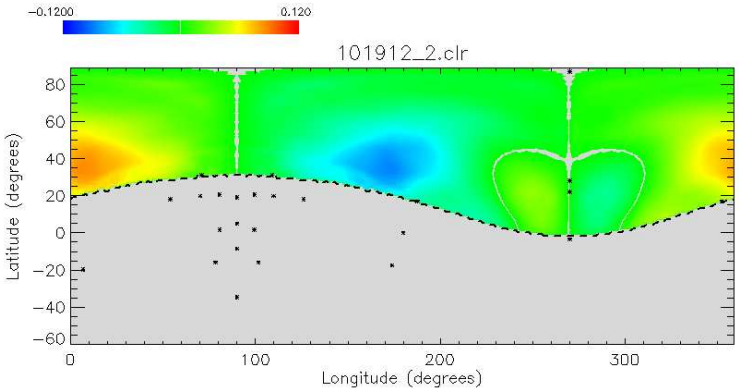


Figure D.20: Flatmap of asymmetry results after helmet therapy.

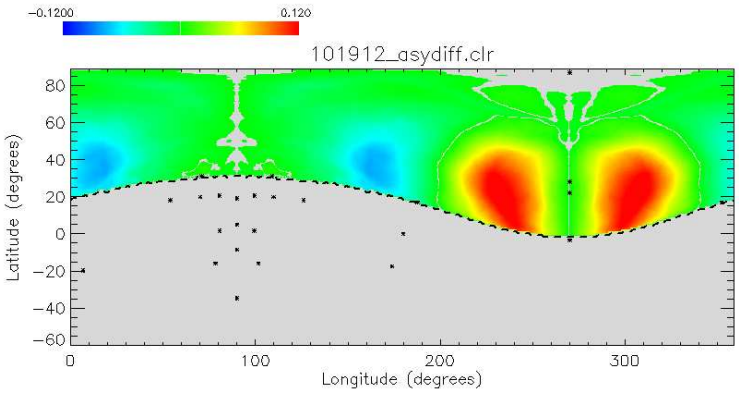


Figure D.21: Flatmap of asymmetry changes from before to after the therapy.

D.8 Asymmetry results for patient 8

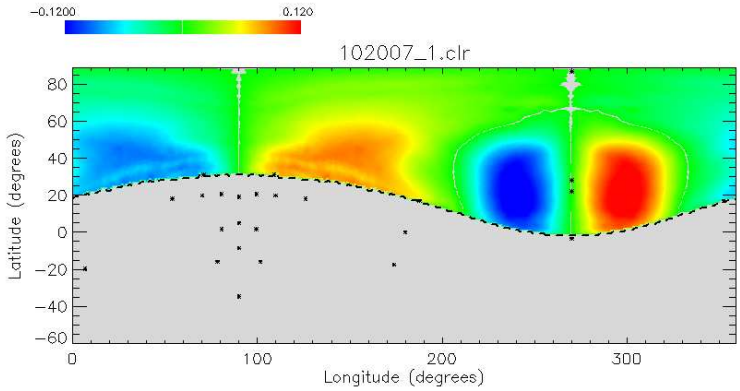


Figure D.22: Flatmap of asymmetry results before helmet therapy.

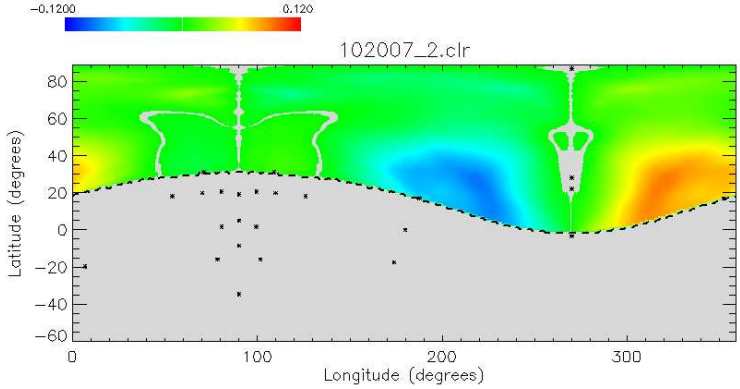


Figure D.23: Flatmap of asymmetry results after helmet therapy.

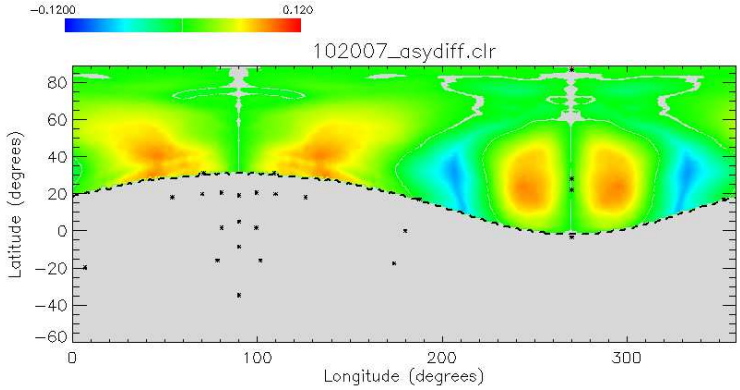


Figure D.24: Flatmap of asymmetry changes from before to after the therapy.

D.9 Asymmetry results for patient 9

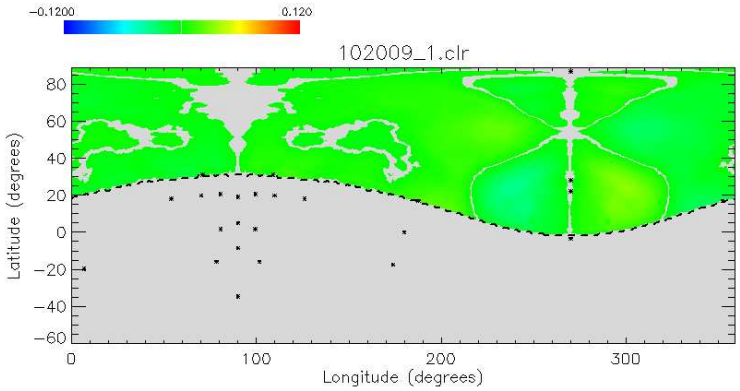


Figure D.25: Flatmap of asymmetry results before helmet therapy.

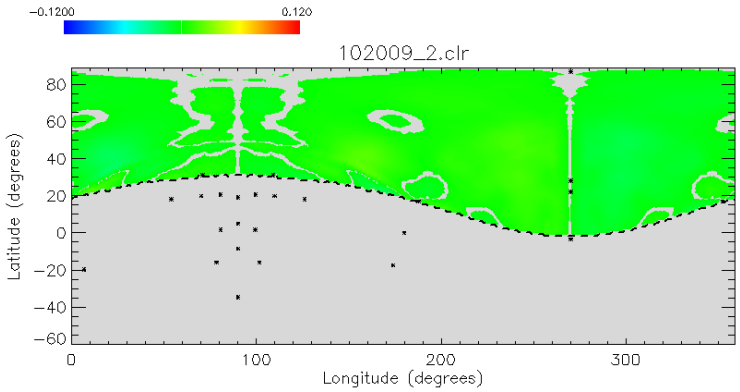


Figure D.26: Flatmap of asymmetry results after helmet therapy.

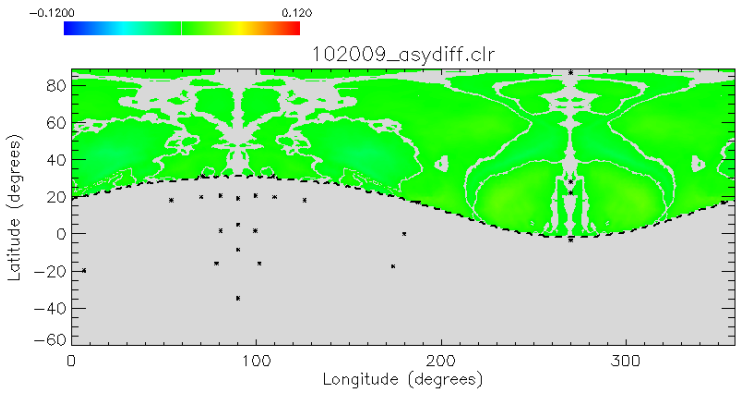


Figure D.27: Flatmap of asymmetry changes from before to after the therapy.

D.10 Asymmetry results for patient 10

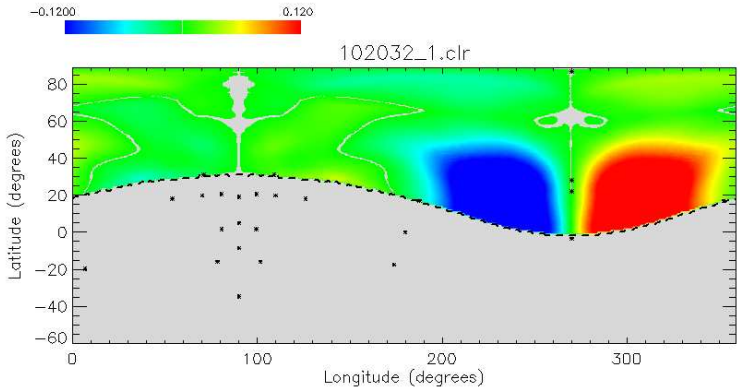


Figure D.28: Flatmap of asymmetry results before helmet therapy.

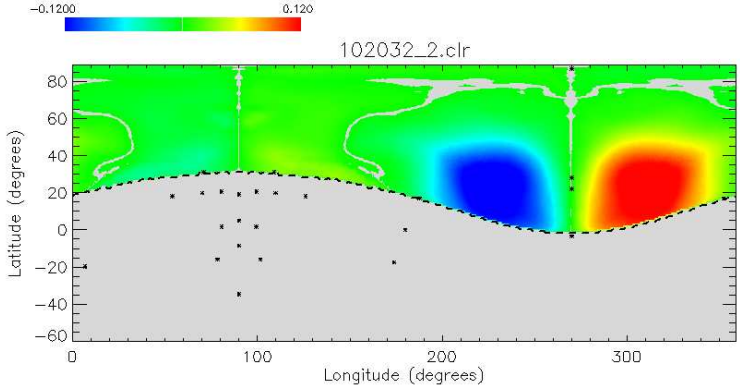


Figure D.29: Flatmap of asymmetry results after helmet therapy.

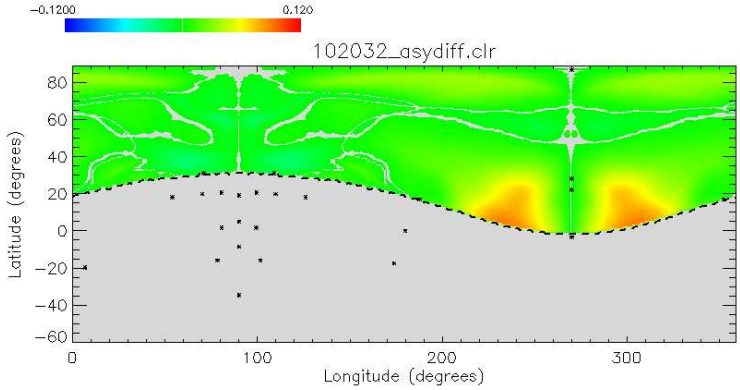


Figure D.30: Flatmap of asymmetry changes from before to after the therapy.

D.11 Asymmetry results for patient 11

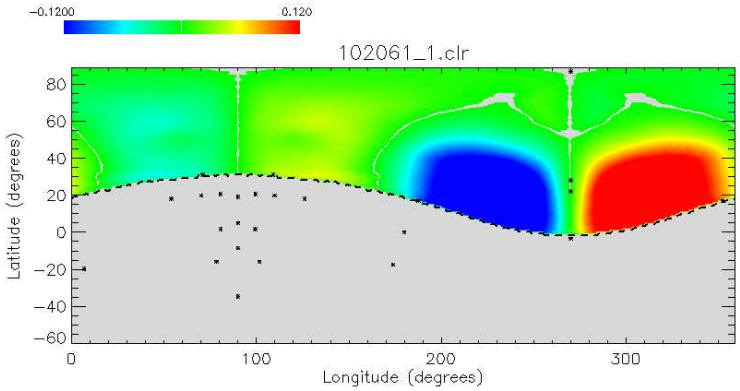


Figure D.31: Flatmap of asymmetry results before helmet therapy.

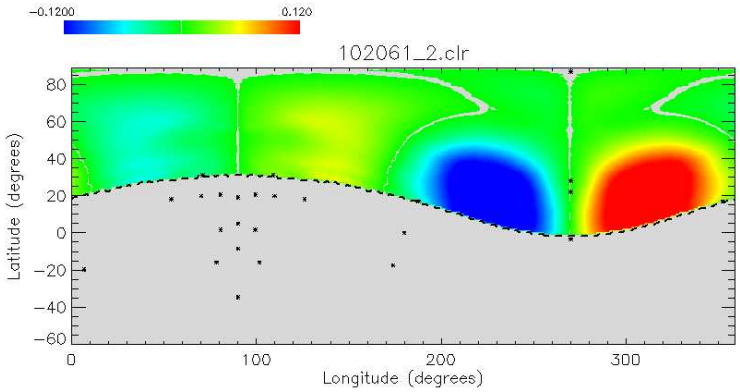


Figure D.32: Flatmap of asymmetry results after helmet therapy.

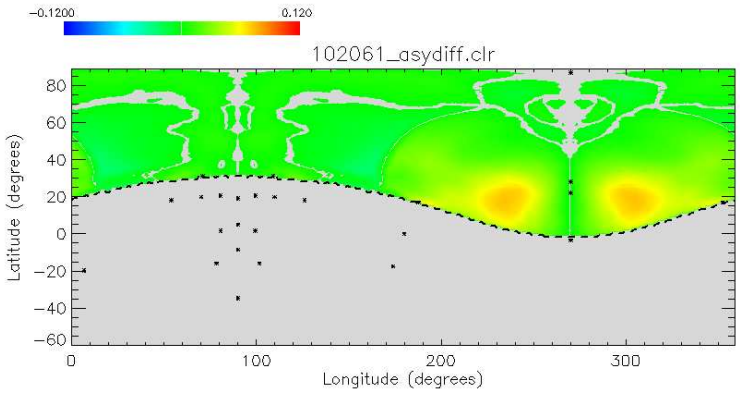


Figure D.33: Flatmap of asymmetry changes from before to after the therapy.

D.12 Asymmetry results for patient 12

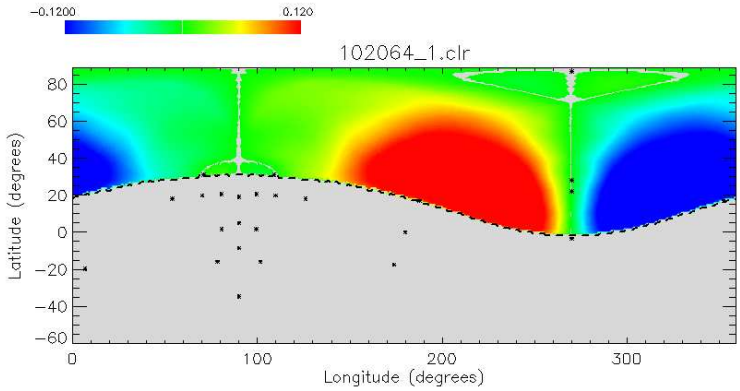


Figure D.34: Flatmap of asymmetry results before helmet therapy.

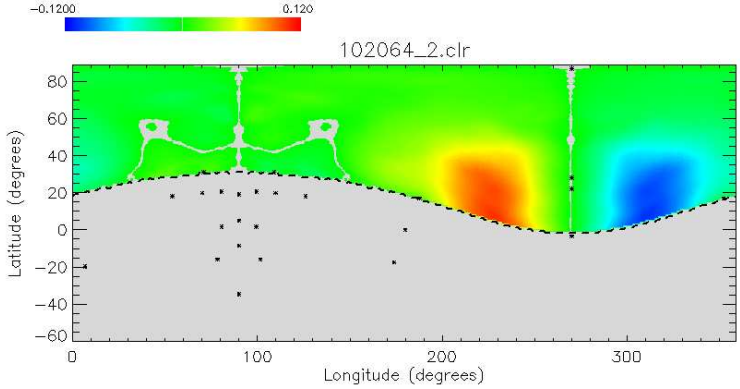


Figure D.35: Flatmap of asymmetry results after helmet therapy.

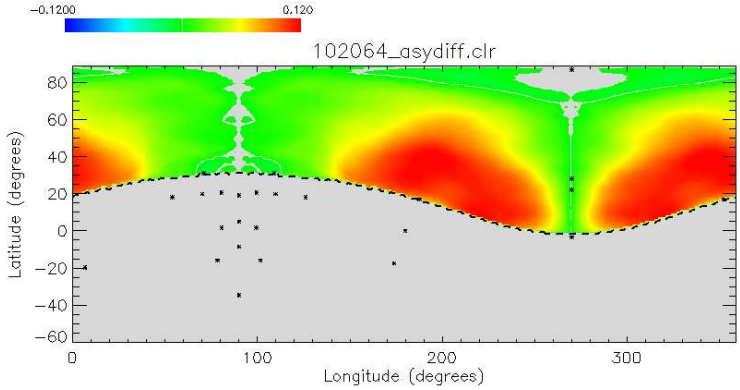


Figure D.36: Flatmap of asymmetry changes from before to after the therapy.

D.13 Asymmetry results for patient 13

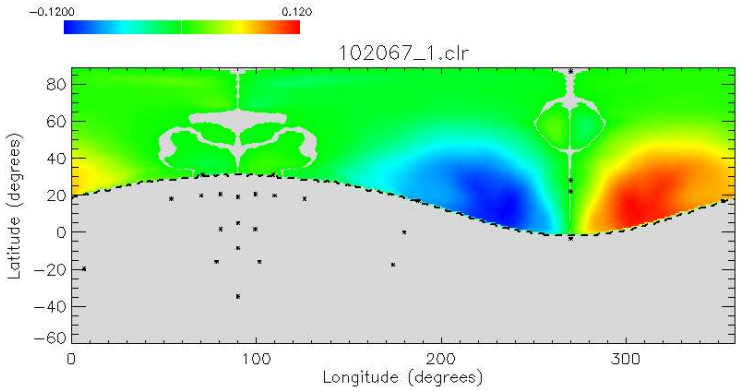


Figure D.37: Flatmap of asymmetry results before helmet therapy.

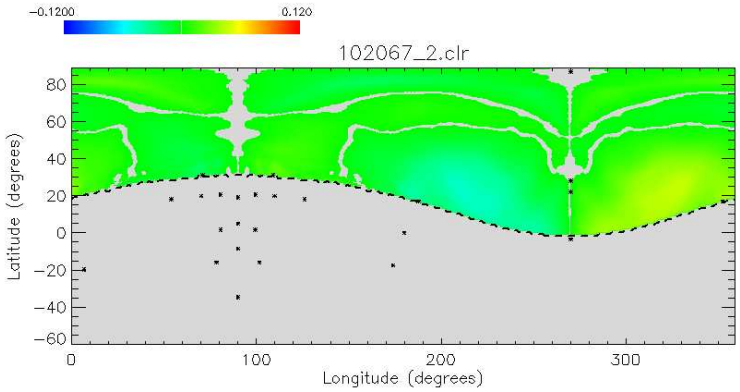


Figure D.38: Flatmap of asymmetry results after helmet therapy.

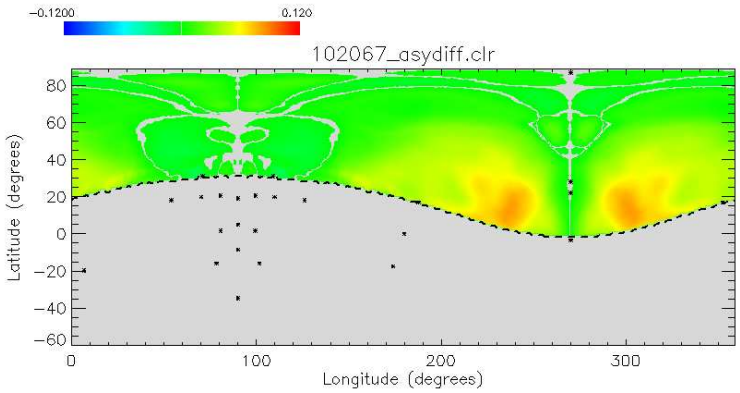


Figure D.39: Flatmap of asymmetry changes from before to after the therapy.

D.14 Asymmetry results for patient 14

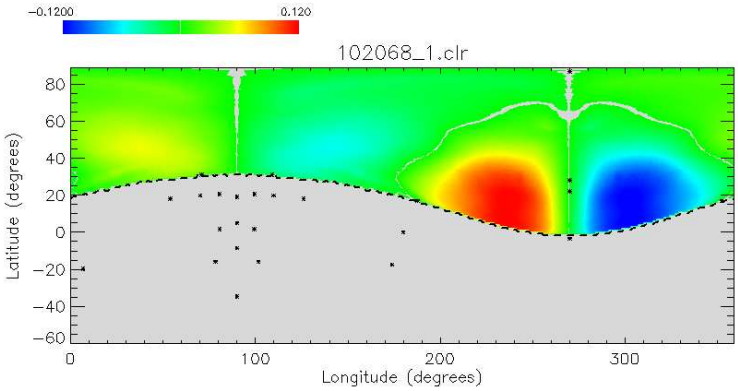


Figure D.40: Flatmap of asymmetry results before helmet therapy.

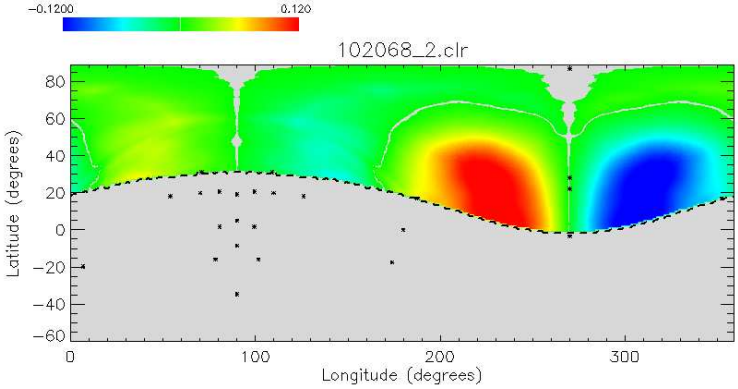


Figure D.41: Flatmap of asymmetry results after helmet therapy.

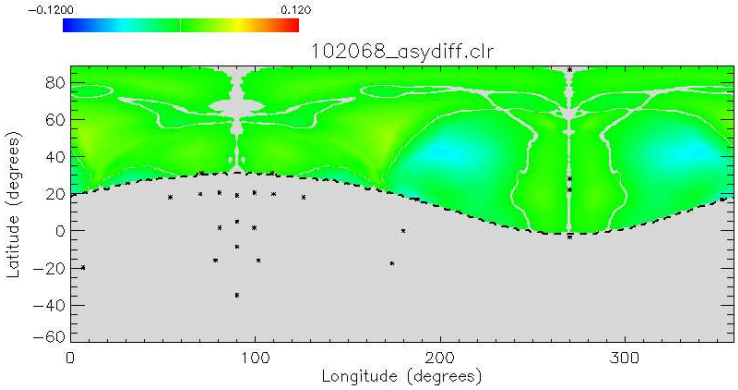


Figure D.42: Flatmap of asymmetry changes from before to after the therapy.

D.15 Asymmetry results for patient 15

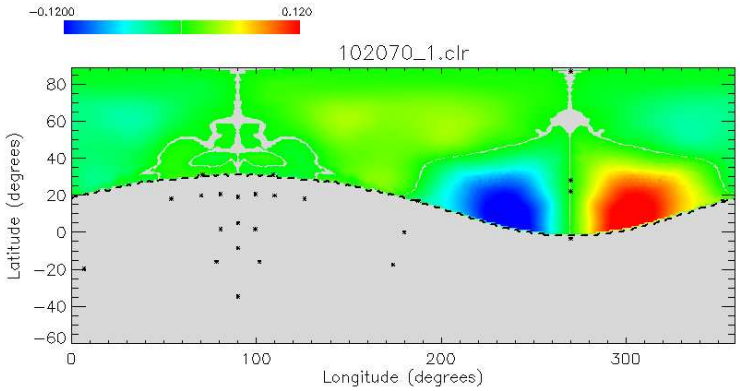


Figure D.43: Flatmap of asymmetry results before helmet therapy.

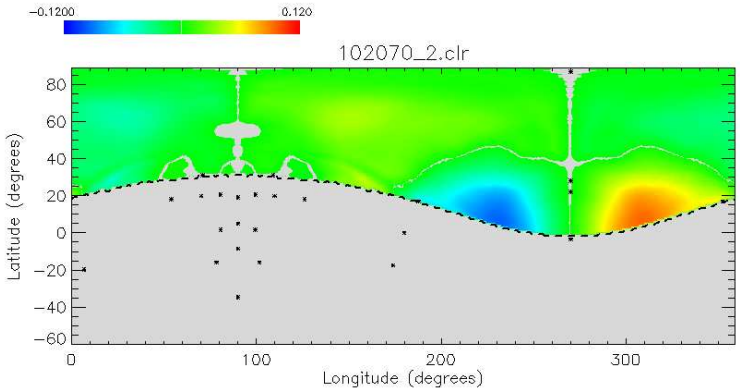


Figure D.44: Flatmap of asymmetry results after helmet therapy.

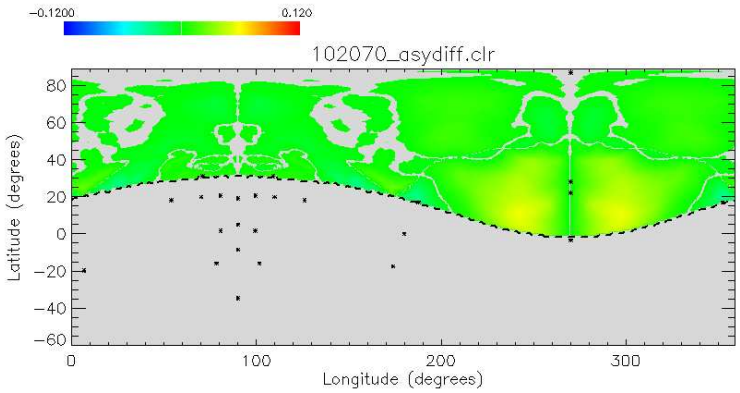


Figure D.45: Flatmap of asymmetry changes from before to after the therapy.

D.16 Asymmetry results for patient 16

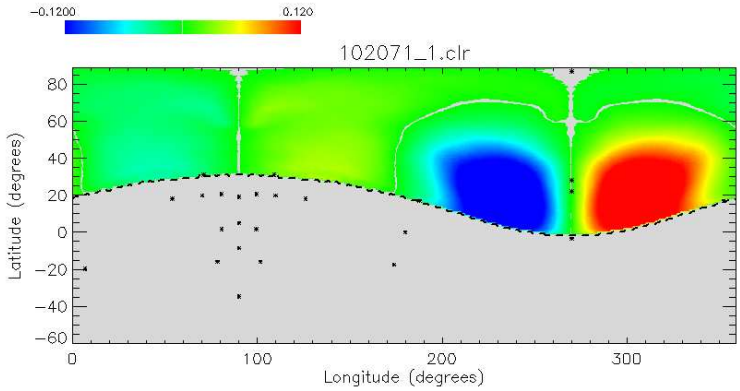


Figure D.46: Flatmap of asymmetry results before helmet therapy.

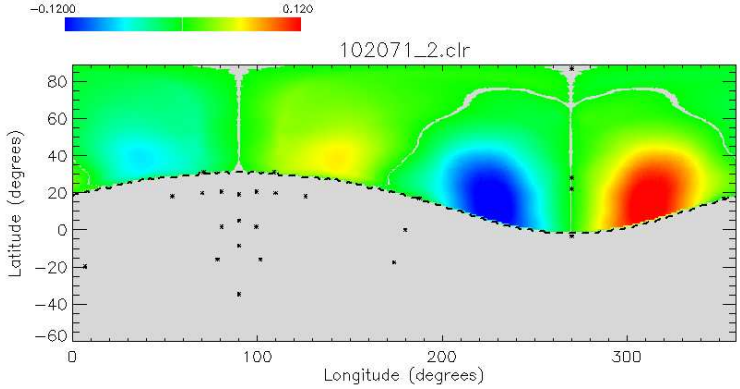


Figure D.47: Flatmap of asymmetry results after helmet therapy.

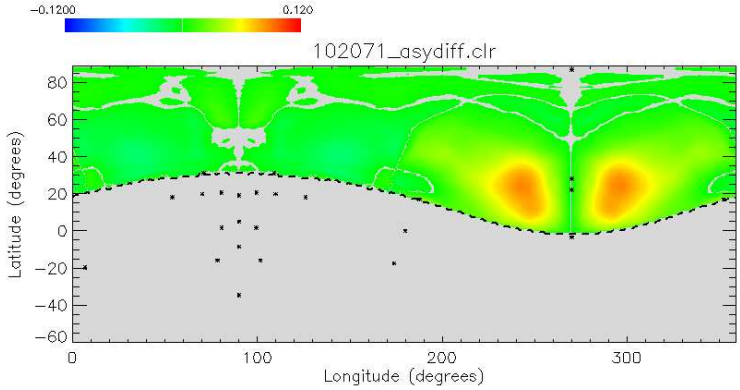


Figure D.48: Flatmap of asymmetry changes from before to after the therapy.

D.17 Asymmetry results for patient 17

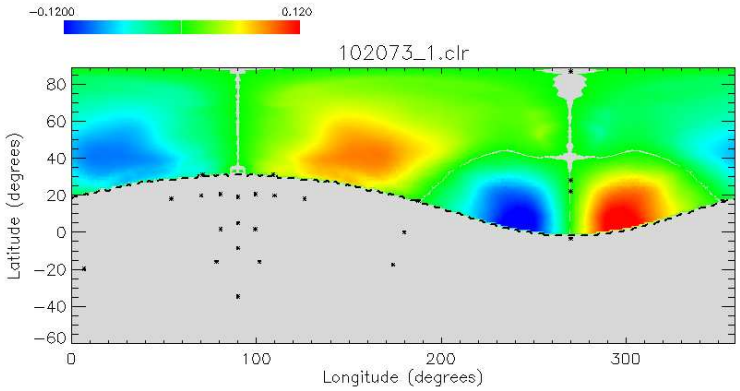


Figure D.49: Flatmap of asymmetry results before helmet therapy.

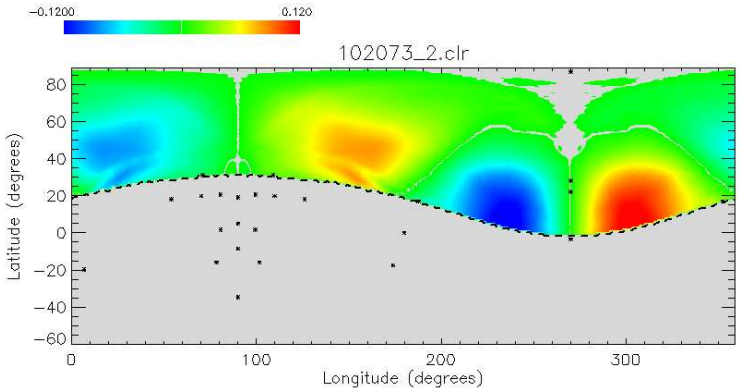


Figure D.50: Flatmap of asymmetry results after helmet therapy.

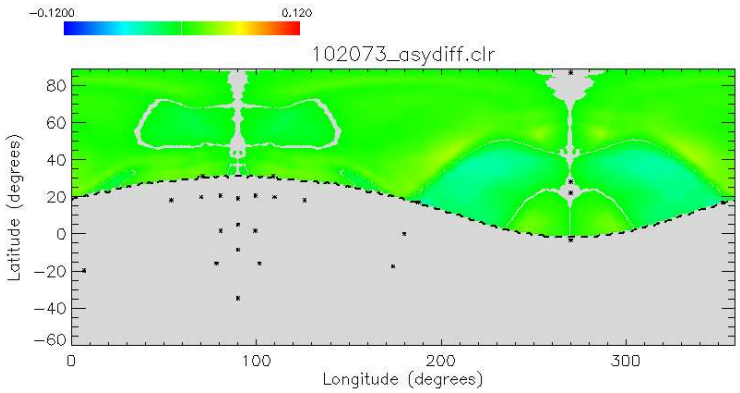


Figure D.51: Flatmap of asymmetry changes from before to after the therapy.

D.18 Asymmetry results for patient 18

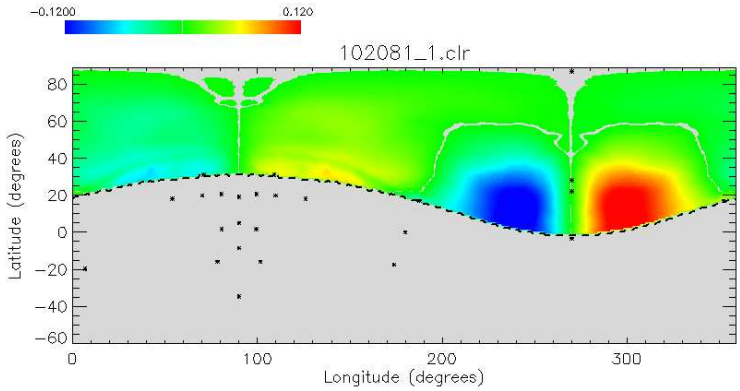


Figure D.52: Flatmap of asymmetry results before helmet therapy.

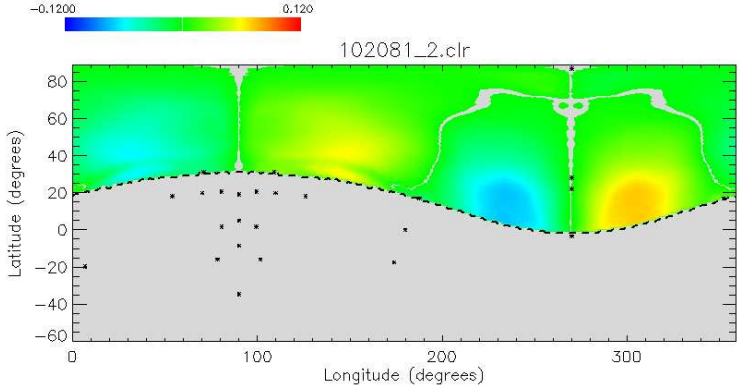


Figure D.53: Flatmap of asymmetry results after helmet therapy.

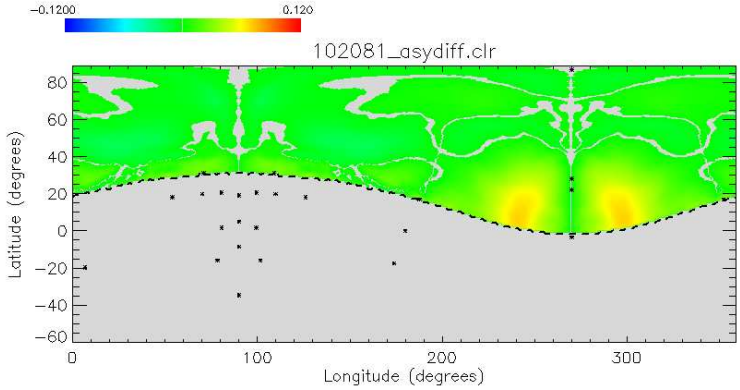


Figure D.54: Flatmap of asymmetry changes from before to after the therapy.

D.19 Asymmetry results for patient 19

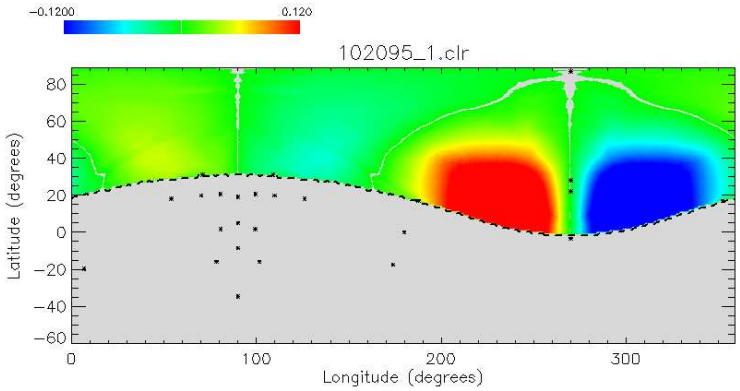


Figure D.55: Flatmap of asymmetry results before helmet therapy.

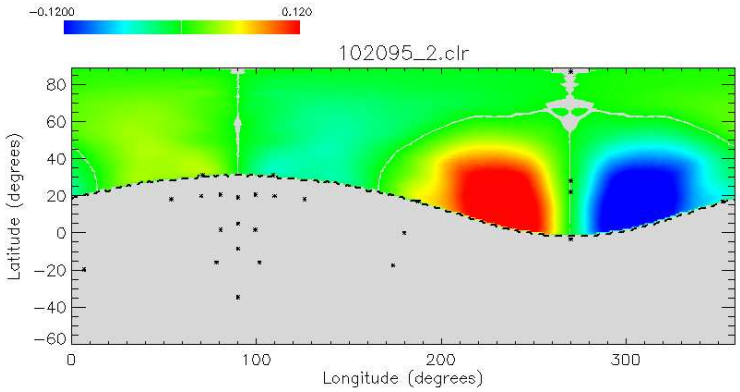


Figure D.56: Flatmap of asymmetry results after helmet therapy.

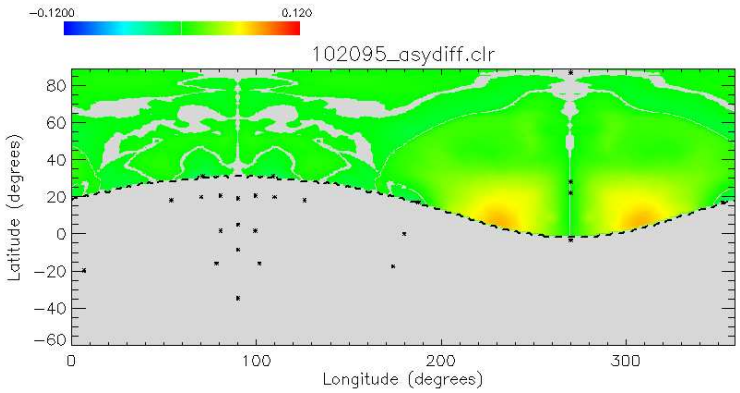


Figure D.57: Flatmap of asymmetry changes from before to after the therapy.

D.20 Asymmetry results for patient 20

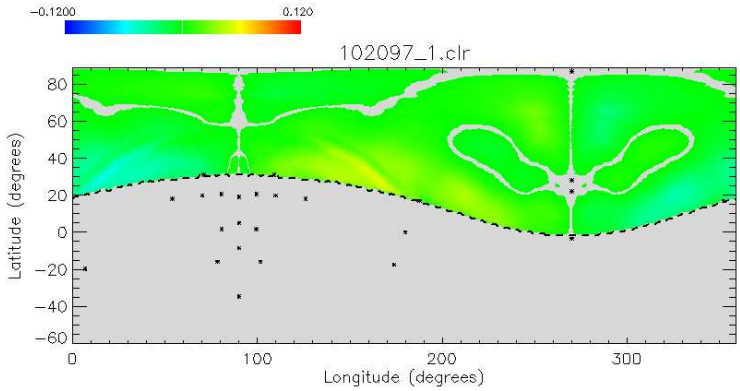


Figure D.58: Flatmap of asymmetry results before helmet therapy.

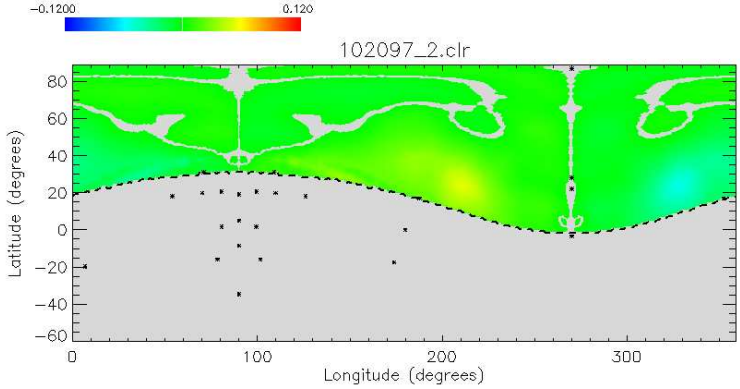


Figure D.59: Flatmap of asymmetry results after helmet therapy.

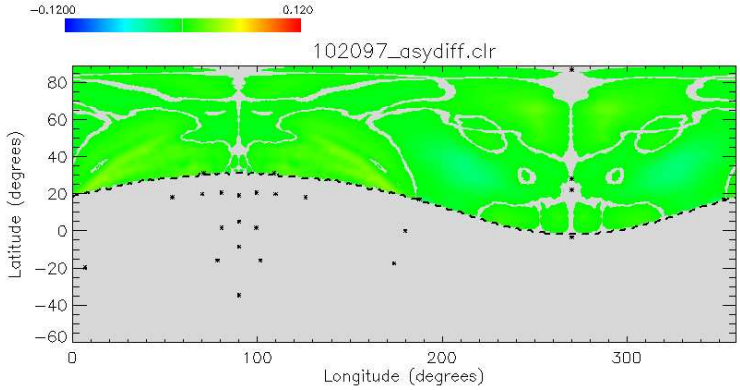


Figure D.60: Flatmap of asymmetry changes from before to after the therapy.

D.21 Asymmetry results for patient 21

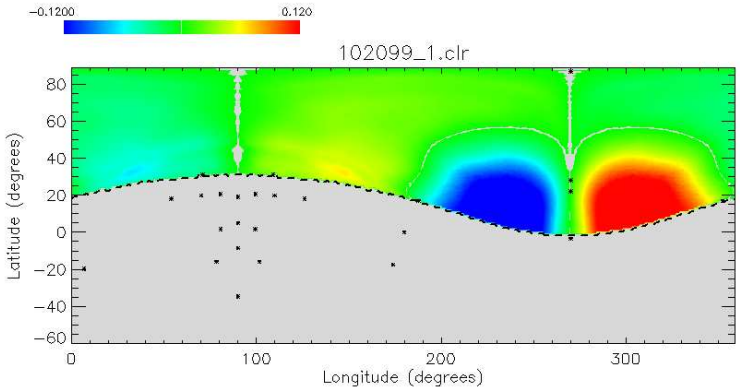


Figure D.61: Flatmap of asymmetry results before helmet therapy.

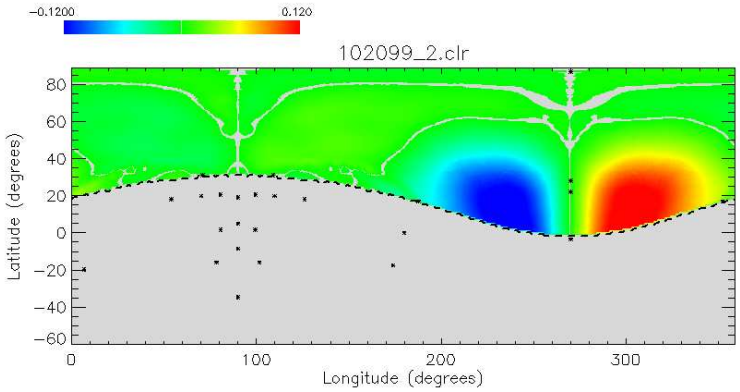


Figure D.62: Flatmap of asymmetry results after helmet therapy.

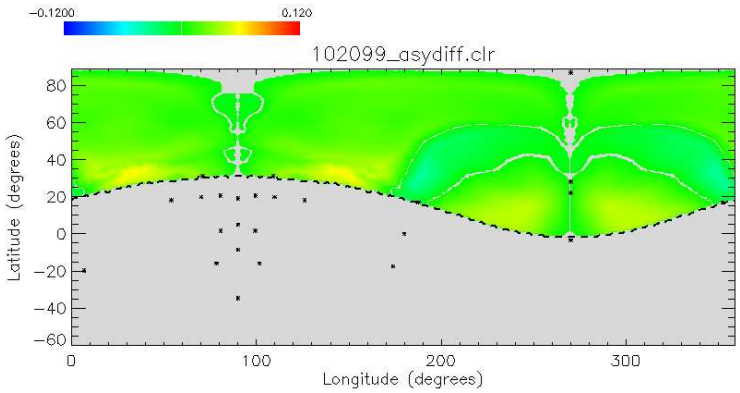


Figure D.63: Flatmap of asymmetry changes from before to after the therapy.

D.22 Asymmetry results for patient 22

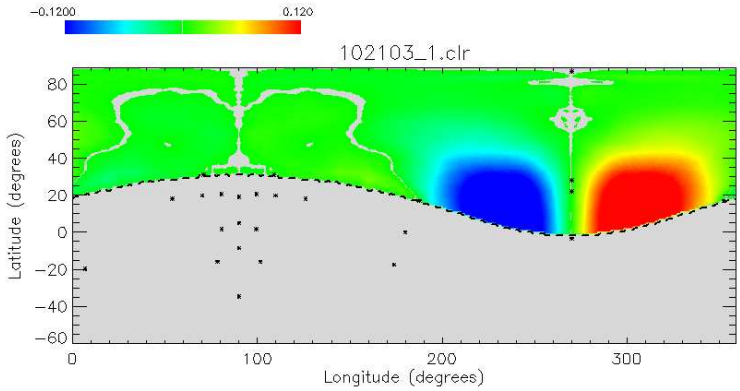


Figure D.64: Flatmap of asymmetry results before helmet therapy.

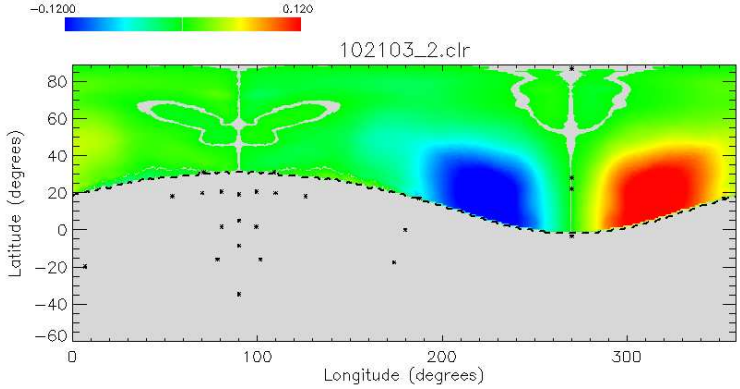


Figure D.65: Flatmap of asymmetry results after helmet therapy.

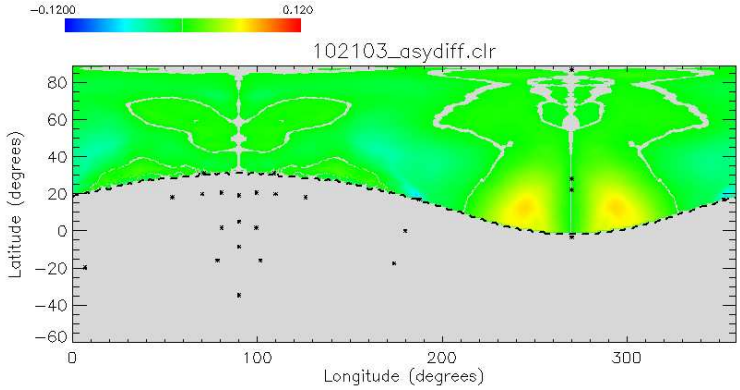


Figure D.66: Flatmap of asymmetry changes from before to after the therapy.

D.23 Asymmetry results for patient 23

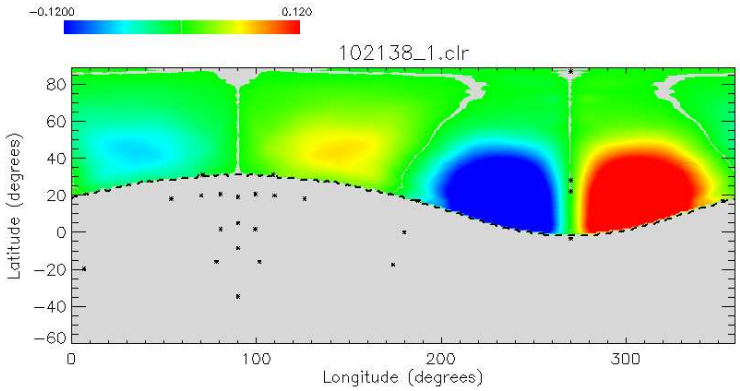


Figure D.67: Flatmap of asymmetry results before helmet therapy.

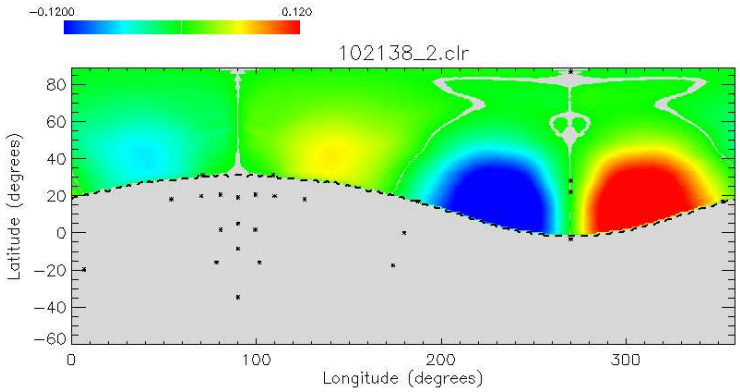


Figure D.68: Flatmap of asymmetry results after helmet therapy.

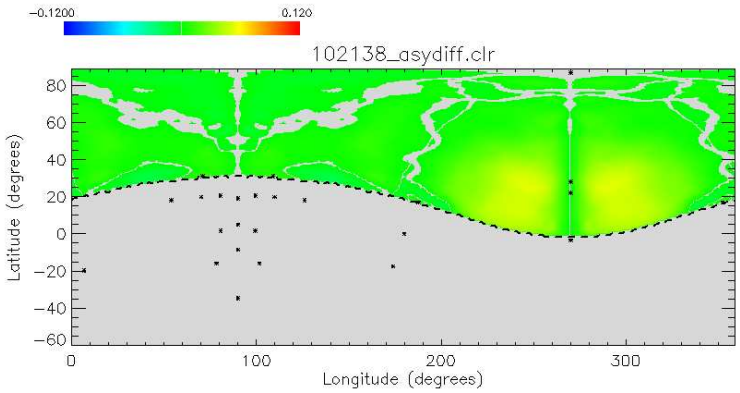


Figure D.69: Flatmap of asymmetry changes from before to after the therapy.

D.24 Asymmetry results for patient 24

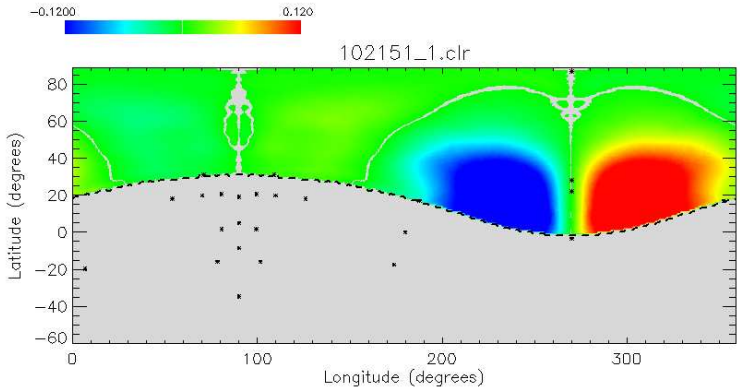


Figure D.70: Flatmap of asymmetry results before helmet therapy.

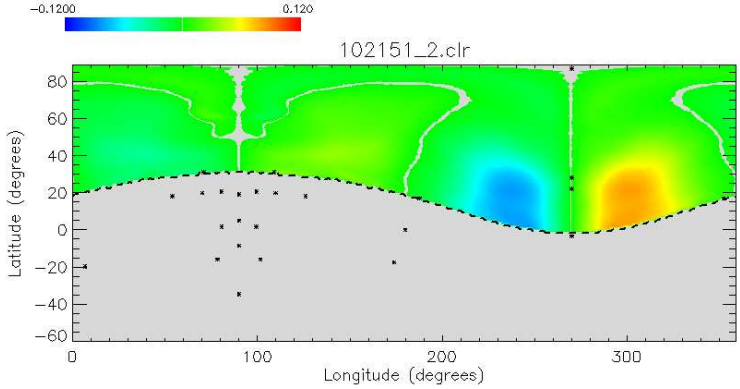


Figure D.71: Flatmap of asymmetry results after helmet therapy.

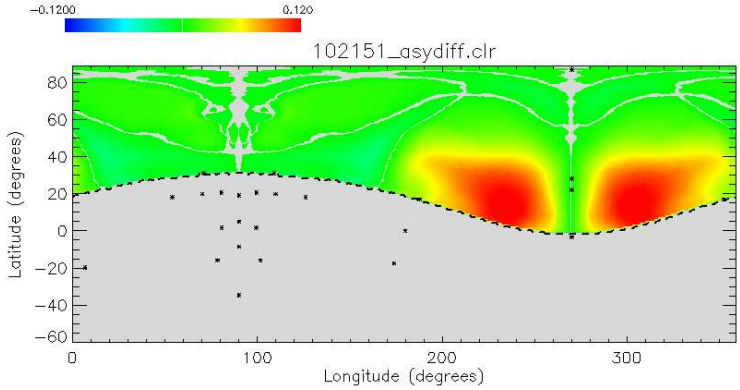


Figure D.72: Flatmap of asymmetry changes from before to after the therapy.

D.25 Asymmetry results for patient 25

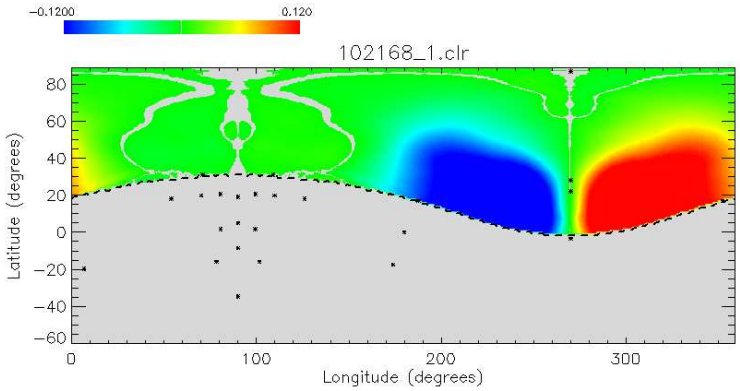


Figure D.73: Flatmap of asymmetry results before helmet therapy.

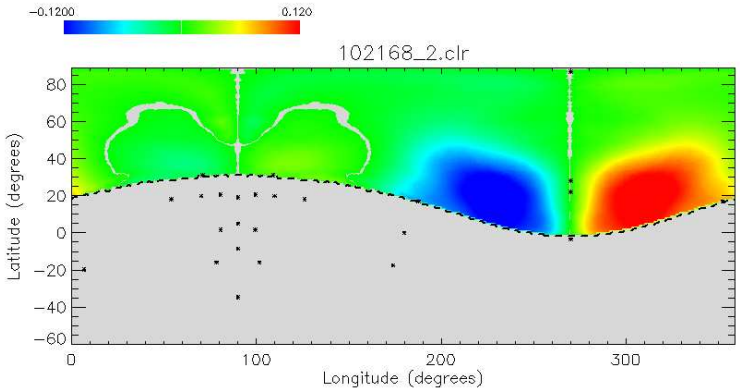


Figure D.74: Flatmap of asymmetry results after helmet therapy.

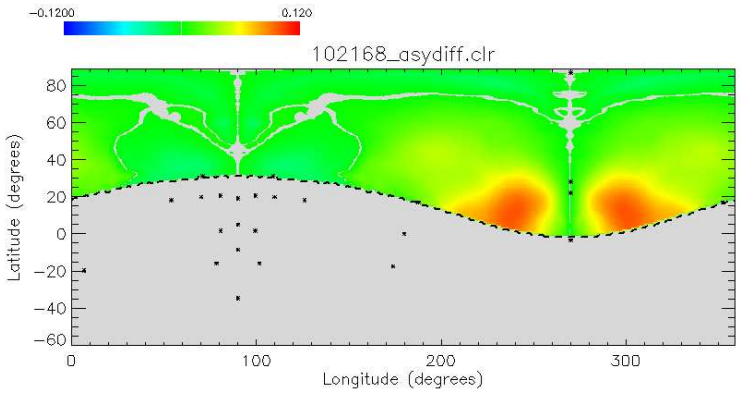


Figure D.75: Flatmap of asymmetry changes from before to after the therapy.

D.26 Asymmetry results for patient 26

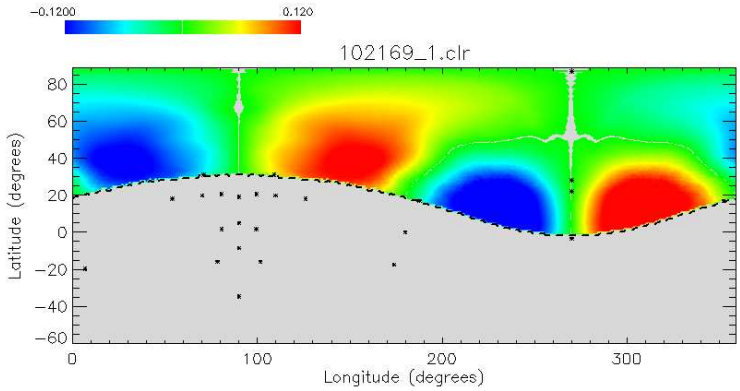


Figure D.76: Flatmap of asymmetry results before helmet therapy.

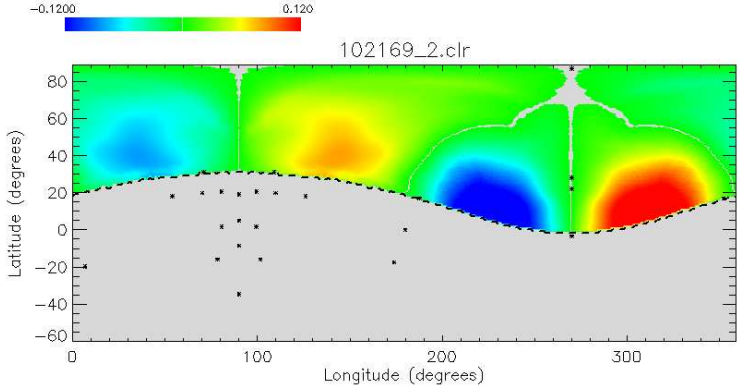


Figure D.77: Flatmap of asymmetry results after helmet therapy.

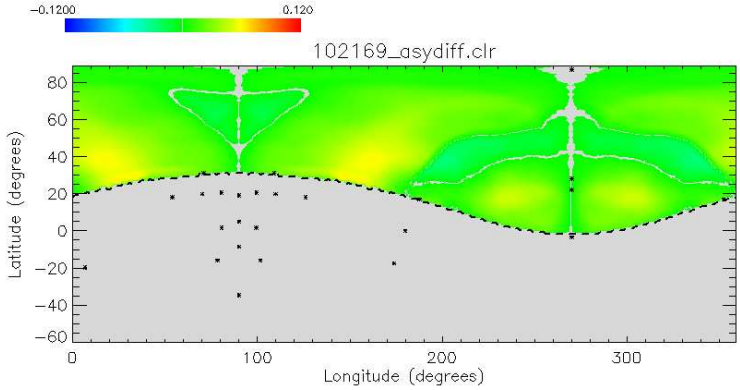


Figure D.78: Flatmap of asymmetry changes from before to after the therapy.

D.27 Asymmetry results for patient 27

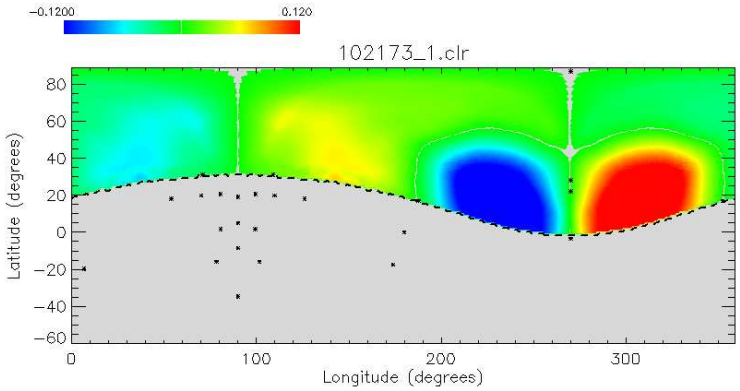


Figure D.79: Flatmap of asymmetry results before helmet therapy.

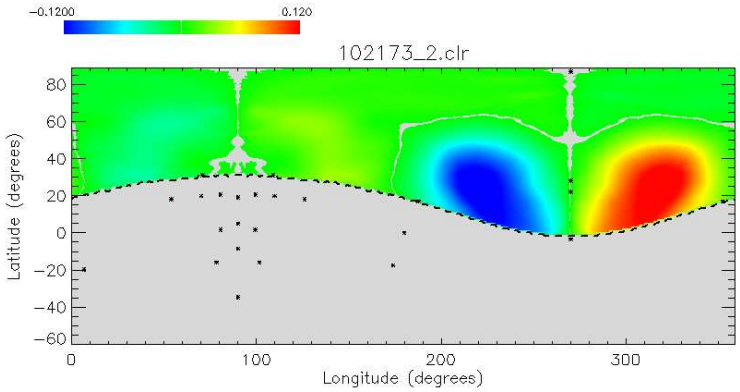


Figure D.80: Flatmap of asymmetry results after helmet therapy.

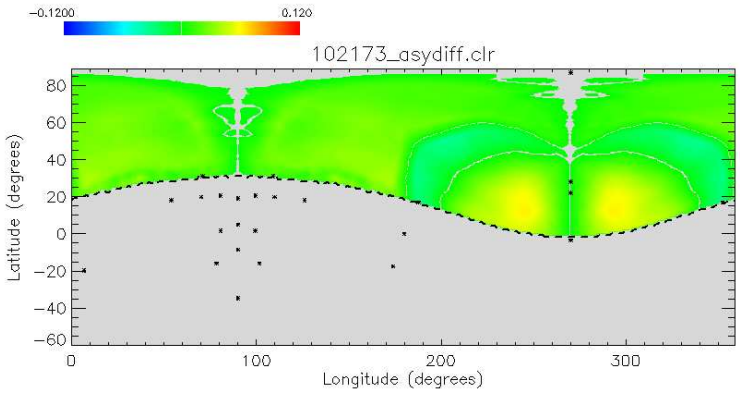


Figure D.81: Flatmap of asymmetry changes from before to after the therapy.

D.28 Asymmetry results for patient 28

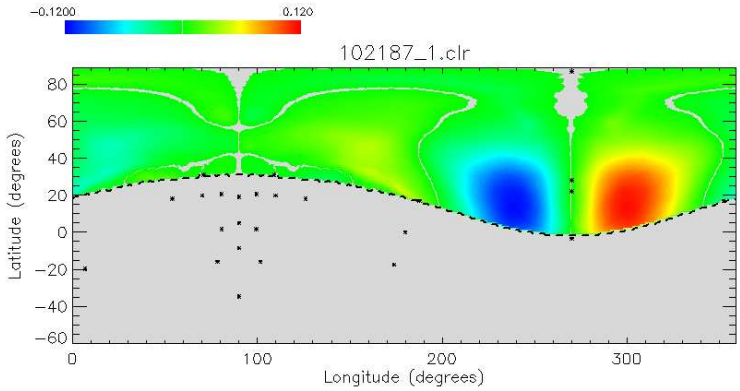


Figure D.82: Flatmap of asymmetry results before helmet therapy.

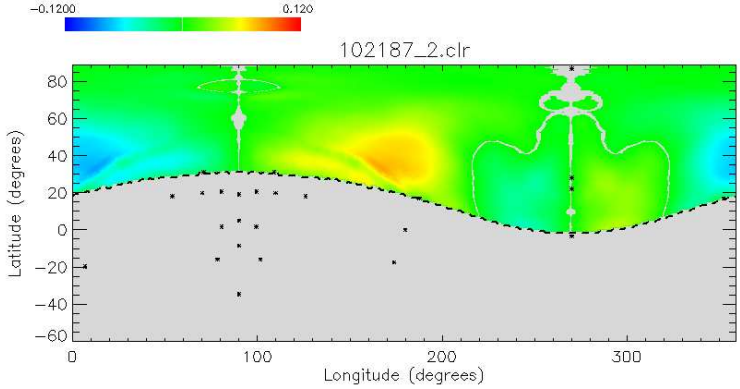


Figure D.83: Flatmap of asymmetry results after helmet therapy.

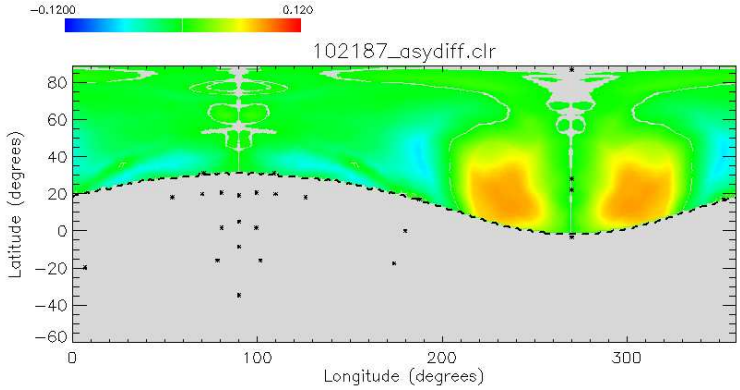


Figure D.84: Flatmap of asymmetry changes from before to after the therapy.

D.29 Asymmetry results for patient 29

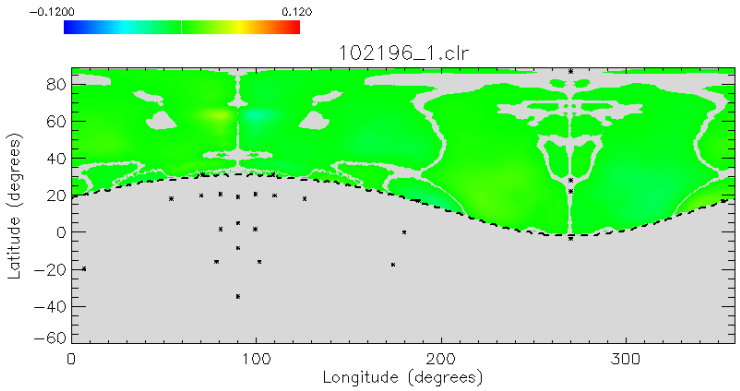


Figure D.85: Flatmap of asymmetry results before helmet therapy.

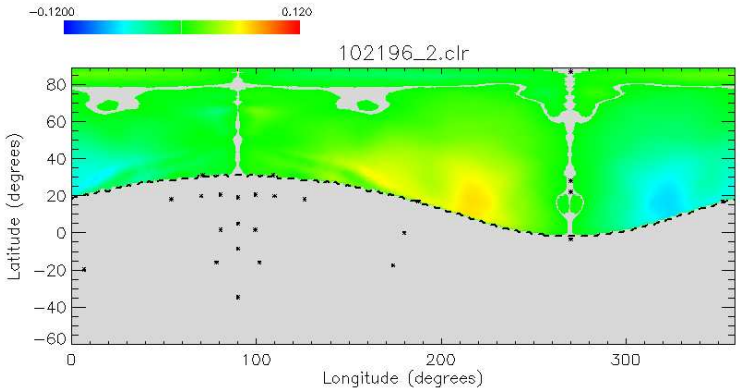


Figure D.86: Flatmap of asymmetry results after helmet therapy.

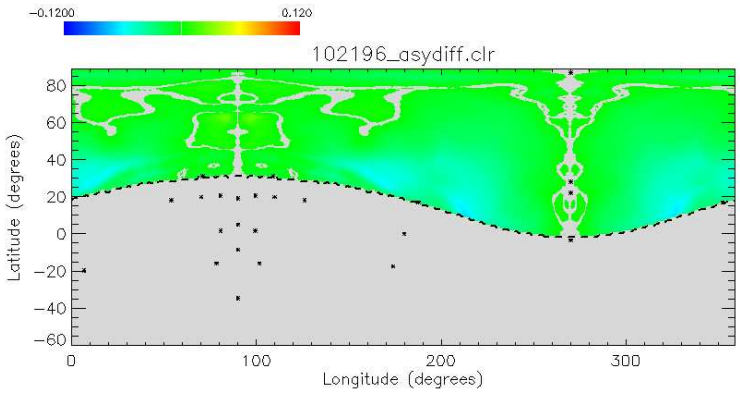


Figure D.87: Flatmap of asymmetry changes from before to after the therapy.

D.30 Asymmetry results for patient 30

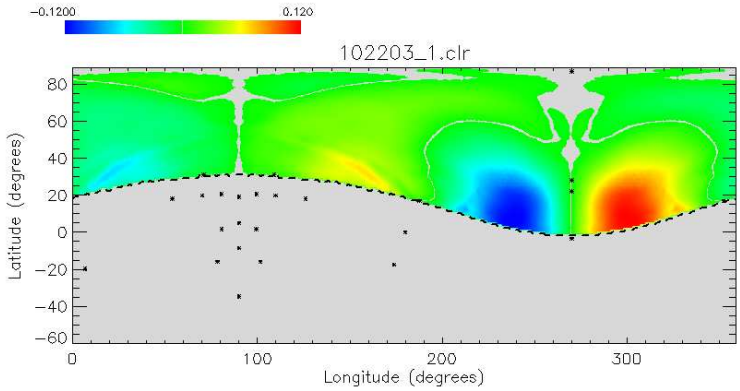


Figure D.88: Flatmap of asymmetry results before helmet therapy.

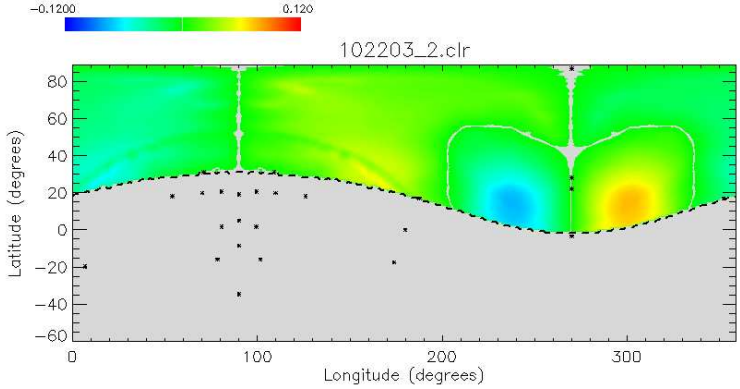


Figure D.89: Flatmap of asymmetry results after helmet therapy.

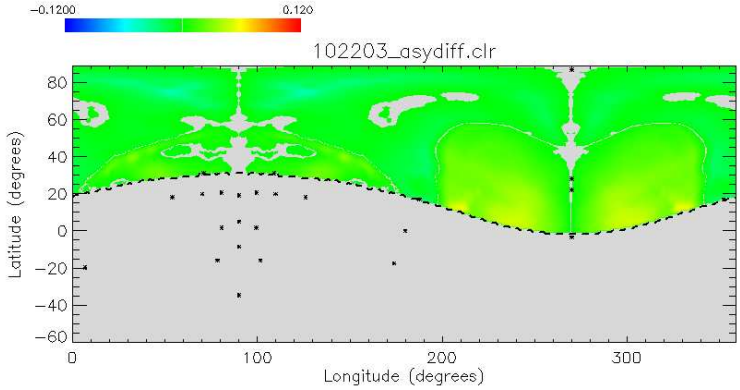


Figure D.90: Flatmap of asymmetry changes from before to after the therapy.

D.31 Asymmetry results for patient 31

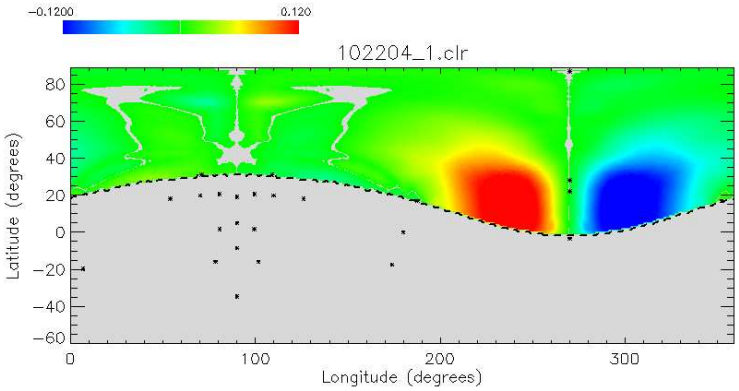


Figure D.91: Flatmap of asymmetry results before helmet therapy.

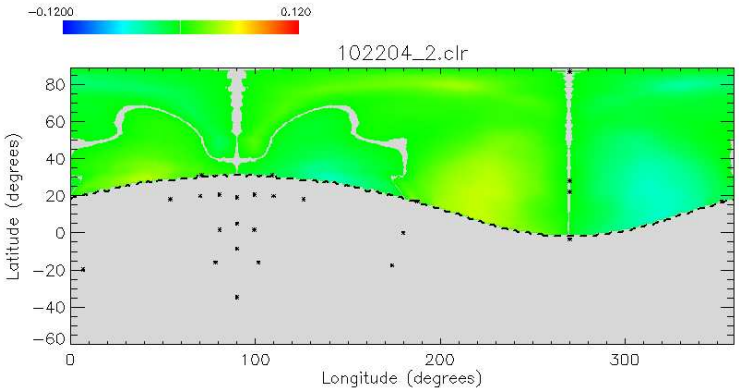


Figure D.92: Flatmap of asymmetry results after helmet therapy.

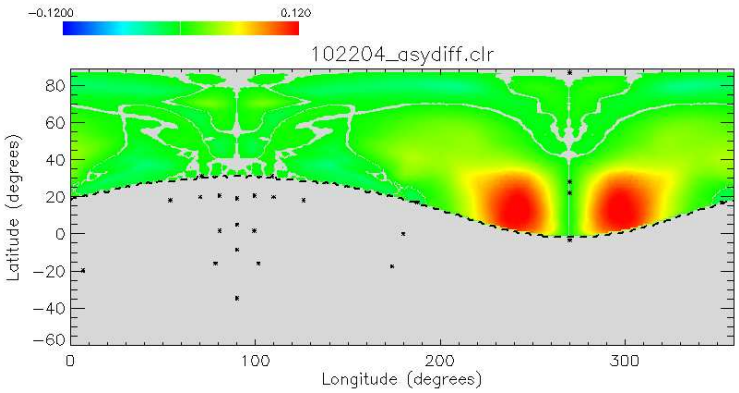


Figure D.93: Flatmap of asymmetry changes from before to after the therapy.

D.32 Asymmetry results for patient 32

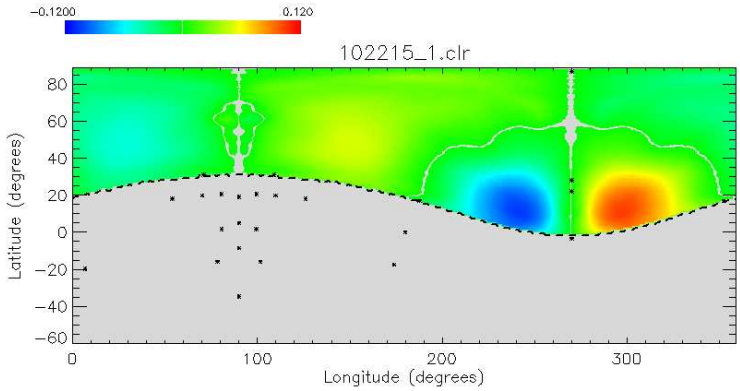


Figure D.94: Flatmap of asymmetry results before helmet therapy.

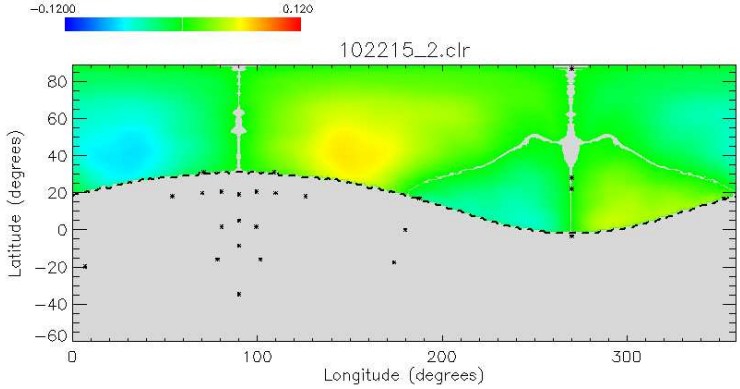


Figure D.95: Flatmap of asymmetry results after helmet therapy.

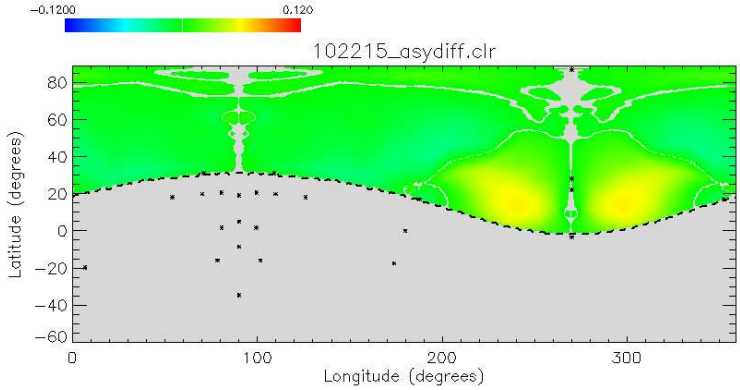


Figure D.96: Flatmap of asymmetry changes from before to after the therapy.

D.33 Asymmetry results for patient 33

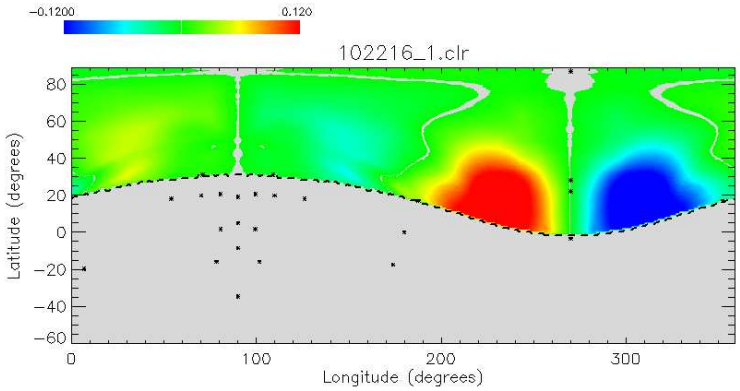


Figure D.97: Flatmap of asymmetry results before helmet therapy.

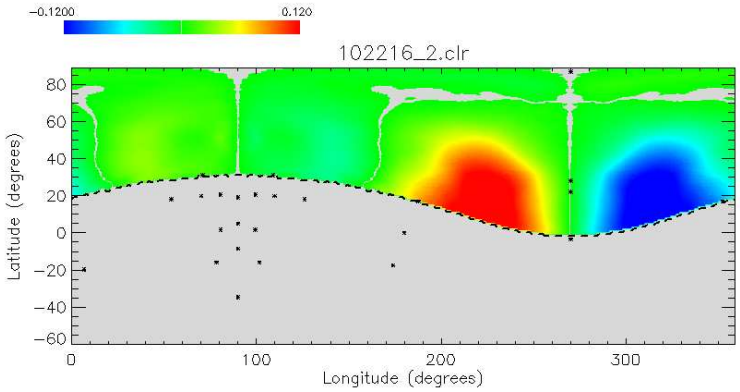


Figure D.98: Flatmap of asymmetry results after helmet therapy.

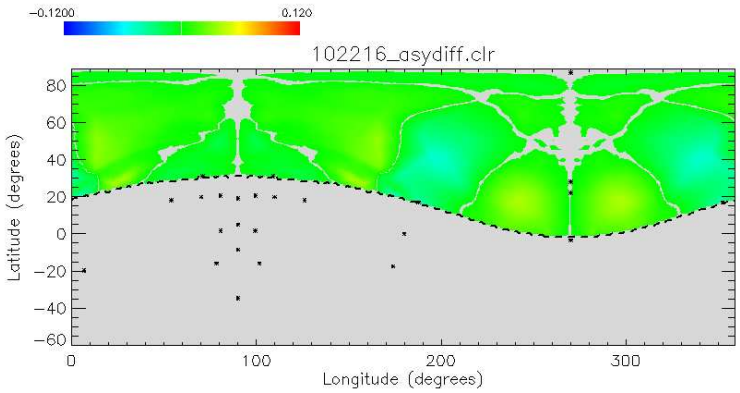


Figure D.99: Flatmap of asymmetry changes from before to after the therapy.

D.34 Asymmetry results for patient 34

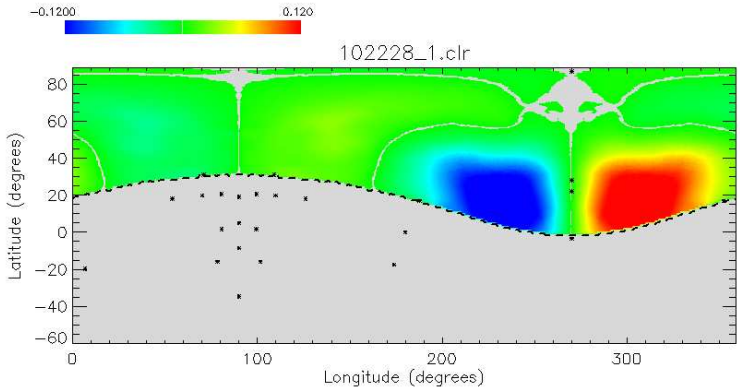


Figure D.100: Flatmap of asymmetry results before helmet therapy.

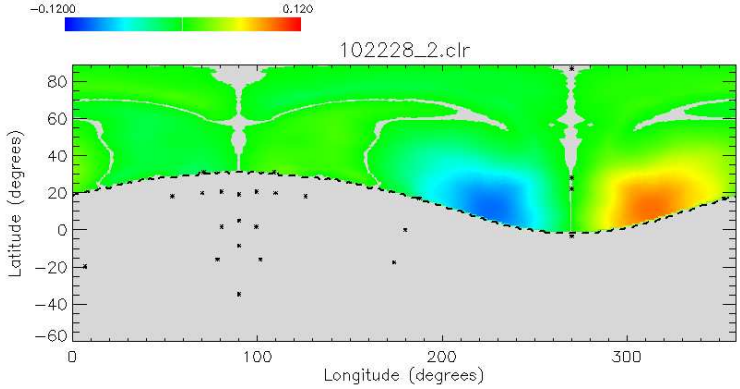


Figure D.101: Flatmap of asymmetry results after helmet therapy.

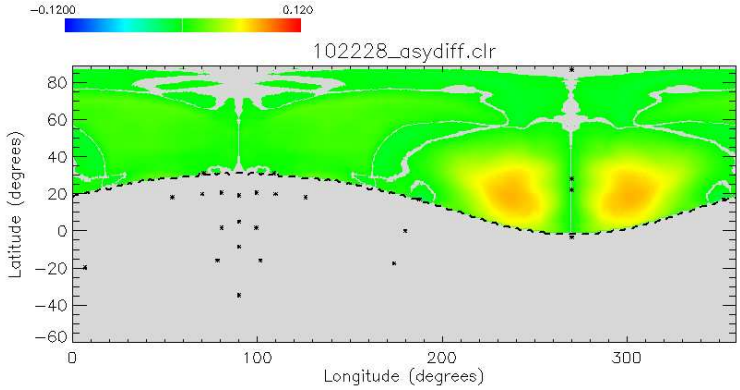


Figure D.102: Flatmap of asymmetry changes from before to after the therapy.

D.35 Asymmetry results for patient 35

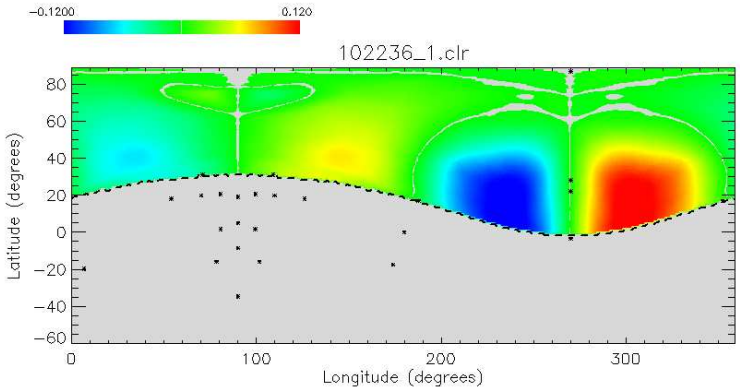


Figure D.103: Flatmap of asymmetry results before helmet therapy.

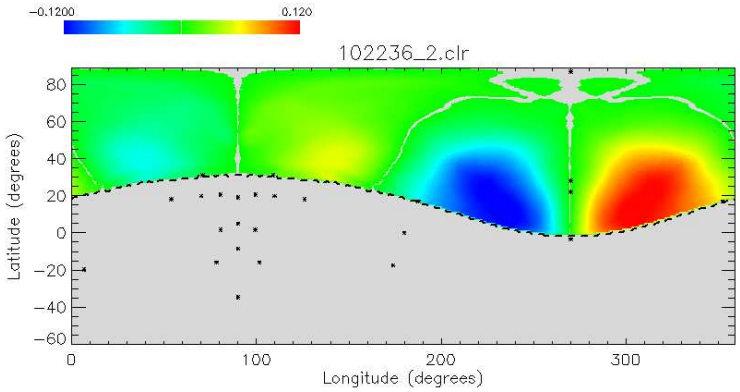


Figure D.104: Flatmap of asymmetry results after helmet therapy.

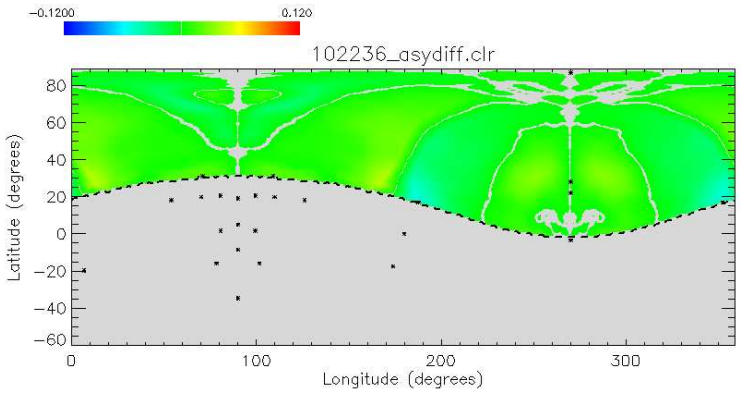


Figure D.105: Flatmap of asymmetry changes from before to after the therapy.

D.36 Asymmetry results for patient 36

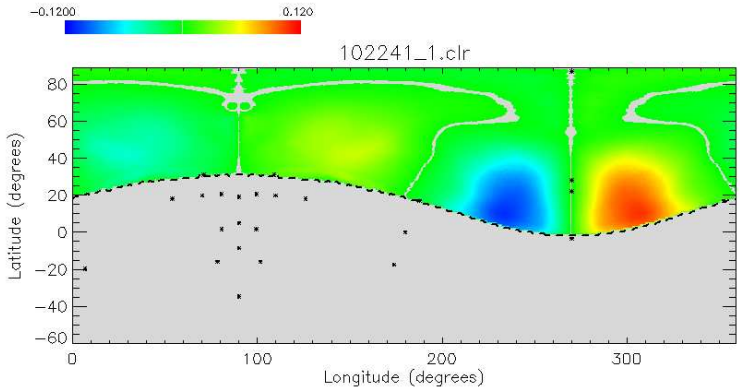


Figure D.106: Flatmap of asymmetry results before helmet therapy.

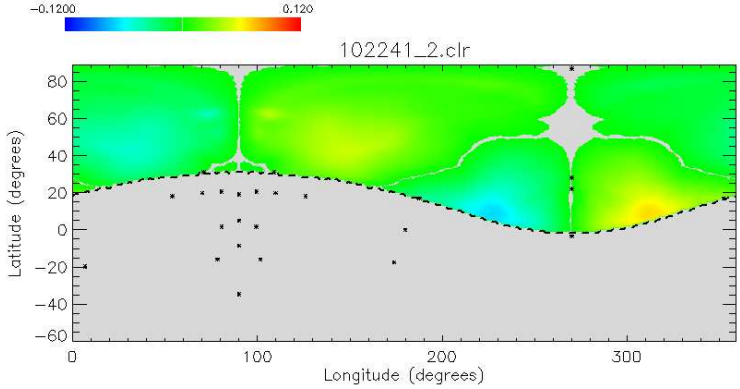


Figure D.107: Flatmap of asymmetry results after helmet therapy.

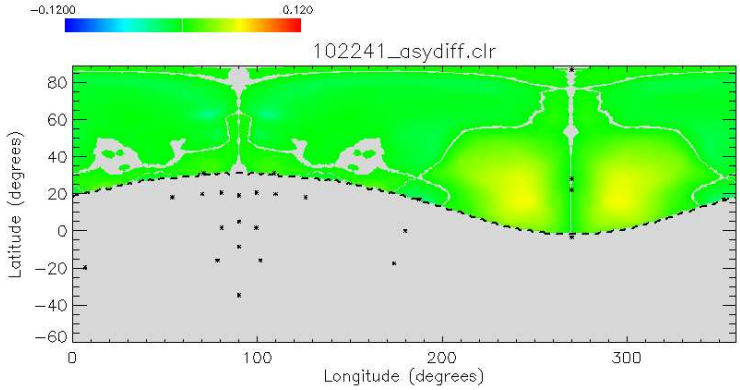


Figure D.108: Flatmap of asymmetry changes from before to after the therapy.

D.37 Asymmetry results for patient 37

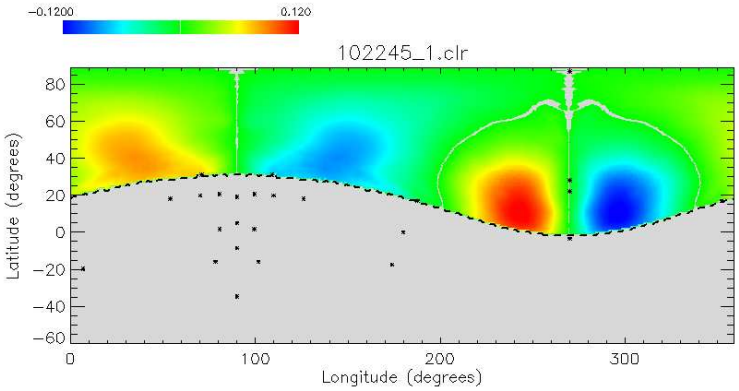


Figure D.109: Flatmap of asymmetry results before helmet therapy.

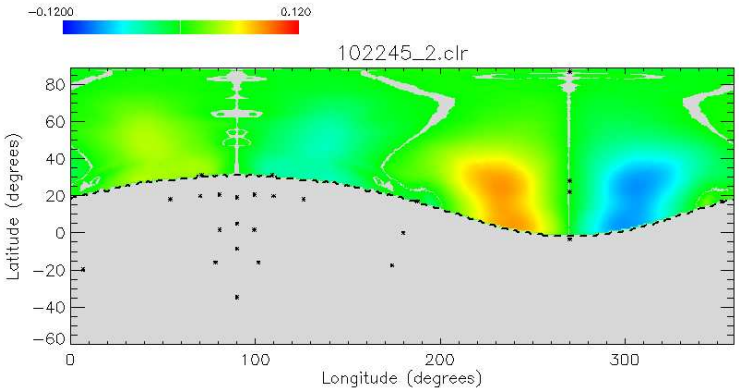


Figure D.110: Flatmap of asymmetry results after helmet therapy.

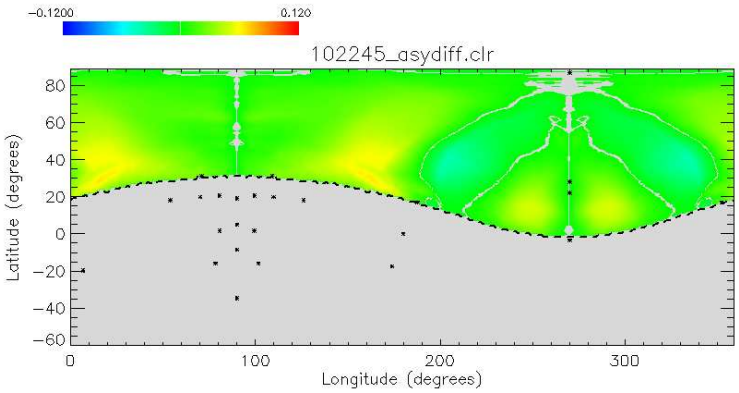


Figure D.111: Flatmap of asymmetry changes from before to after the therapy.

D.38 Asymmetry results for patient 38

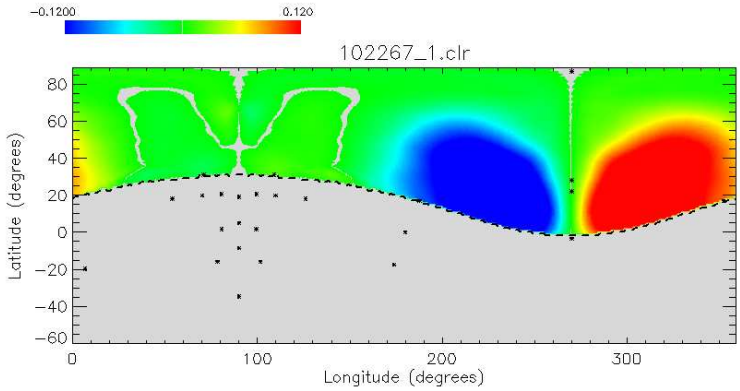


Figure D.112: Flatmap of asymmetry results before helmet therapy.

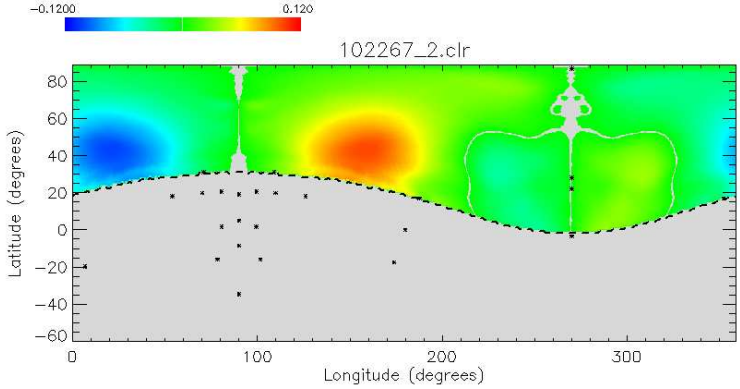


Figure D.113: Flatmap of asymmetry results after helmet therapy.

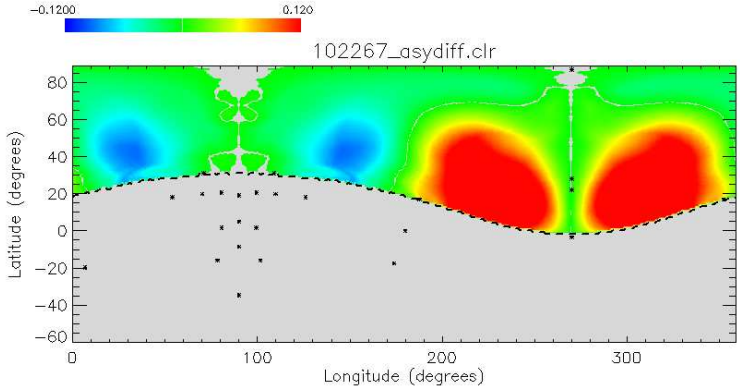


Figure D.114: Flatmap of asymmetry changes from before to after the therapy.

D.39 Asymmetry results for patient 39

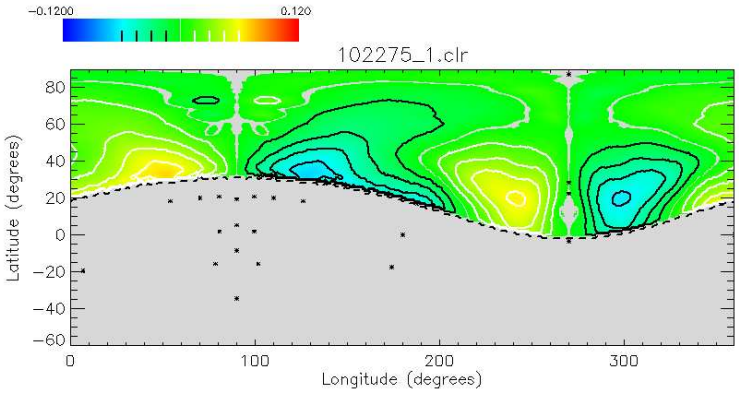


Figure D.115: Flatmap of asymmetry results before helmet therapy.

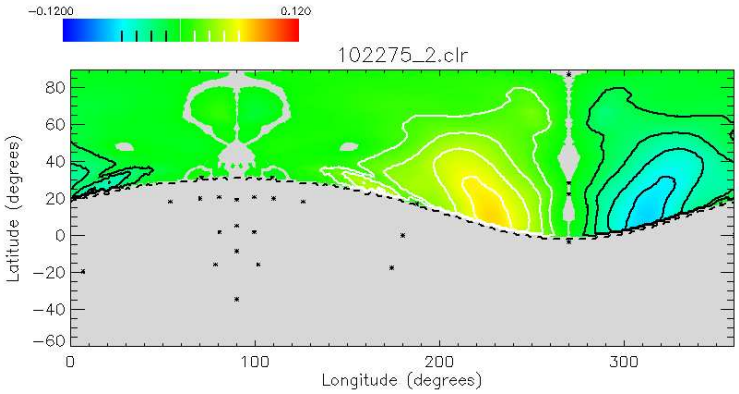


Figure D.116: Flatmap of asymmetry results after helmet therapy.

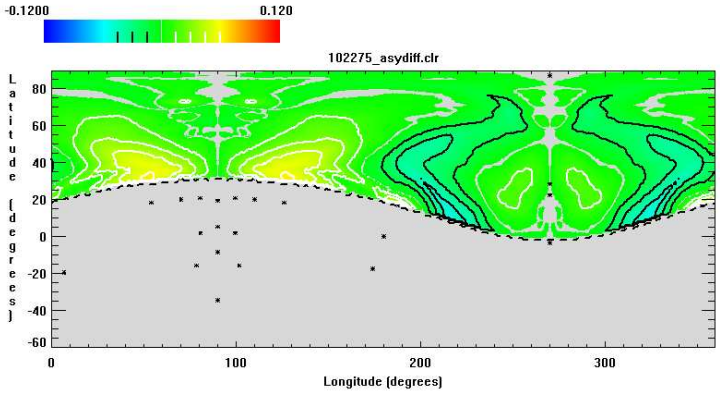


Figure D.117: Flatmap of asymmetry changes from before to after the therapy.

D.40 Asymmetry results for patient 40

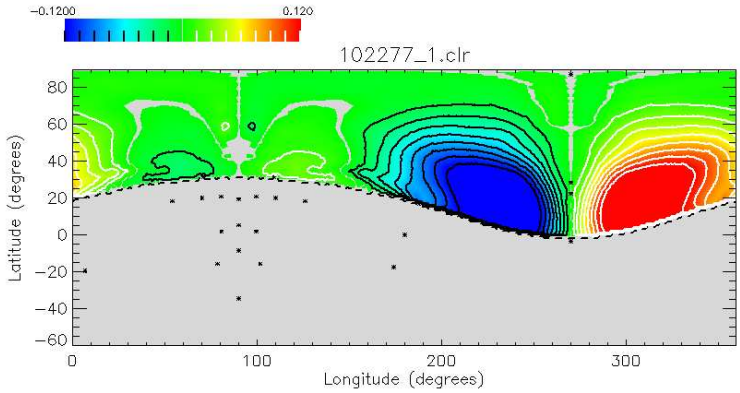


Figure D.118: Flatmap of asymmetry results before helmet therapy.

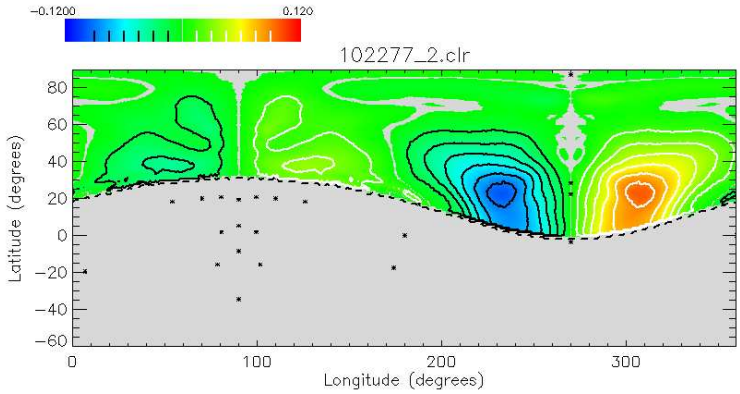


Figure D.119: Flatmap of asymmetry results after helmet therapy.

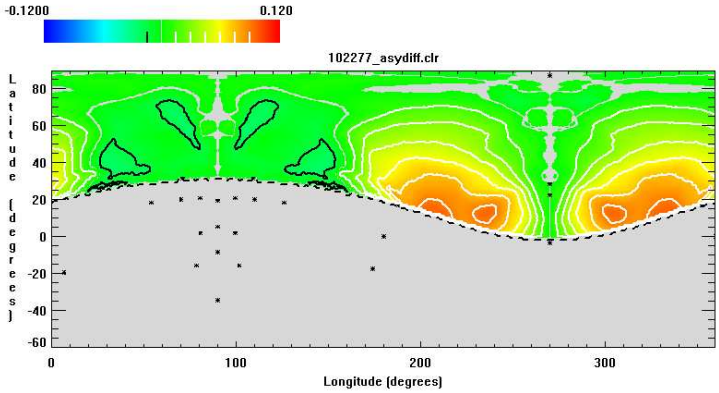


Figure D.120: Flatmap of asymmetry changes from before to after the therapy.

D.41 Asymmetry results for patient 41

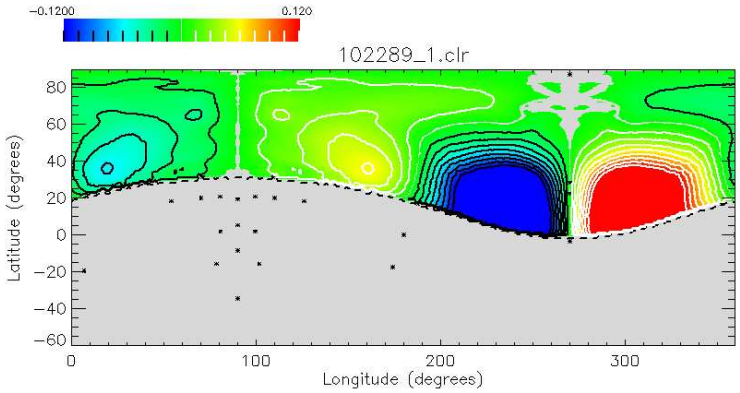


Figure D.121: Flatmap of asymmetry results before helmet therapy.

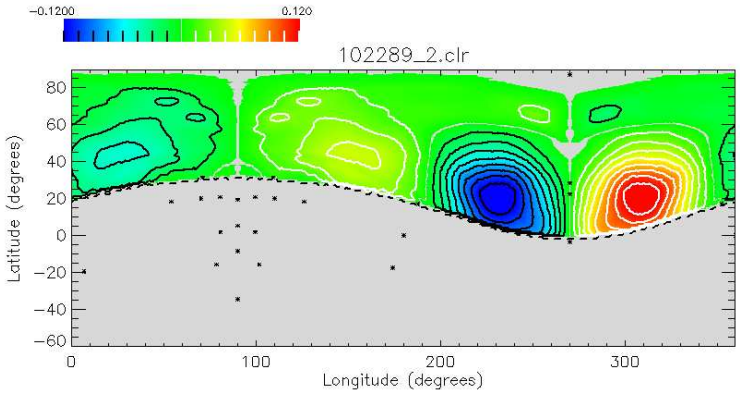


Figure D.122: Flatmap of asymmetry results after helmet therapy.

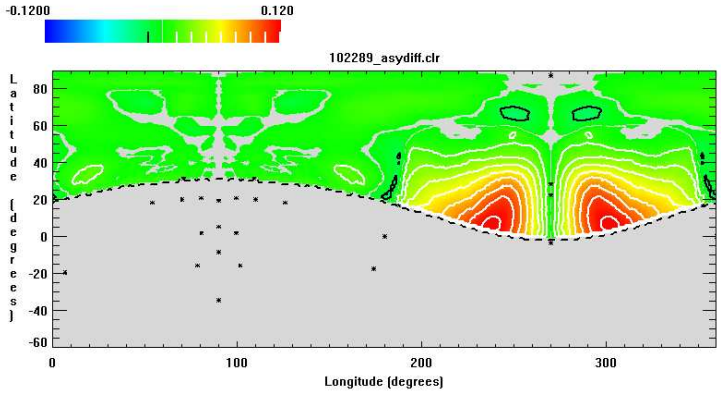


Figure D.123: Flatmap of asymmetry changes from before to after the therapy.

D.42 Asymmetry results for patient 42

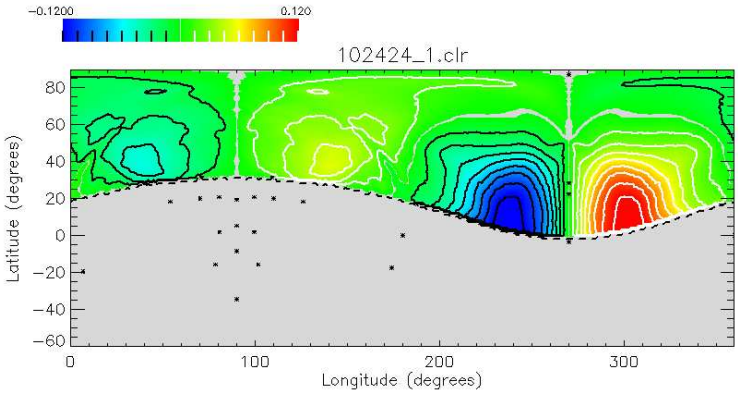


Figure D.124: Flatmap of asymmetry results before helmet therapy.

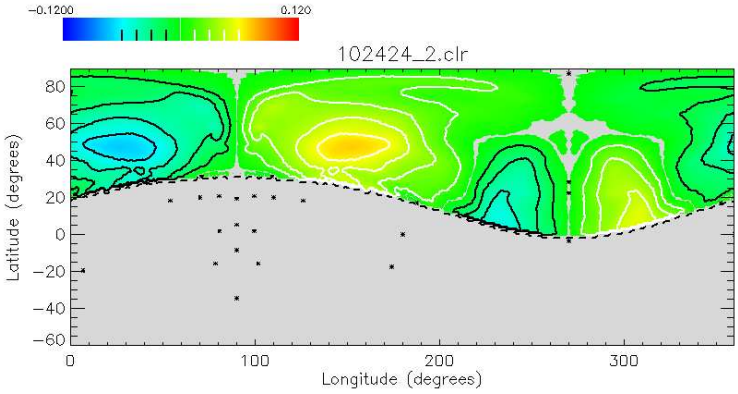


Figure D.125: Flatmap of asymmetry results after helmet therapy.

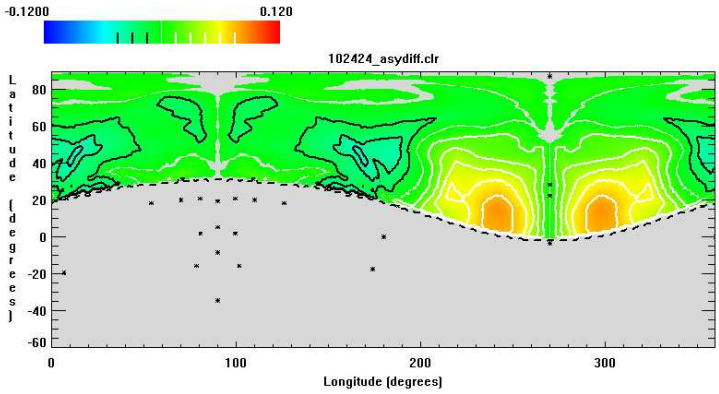


Figure D.126: Flatmap of asymmetry changes from before to after the therapy.