
Theses and Dissertations

Fall 2010

A method for automated landmark constellation detection using evolutionary principal components and statistical shape models

Wei Lu
University of Iowa

Follow this and additional works at: <https://ir.uiowa.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Copyright © 2010 Wei Lu

This thesis is available at Iowa Research Online: <https://ir.uiowa.edu/etd/851>

Recommended Citation

Lu, Wei. "A method for automated landmark constellation detection using evolutionary principal components and statistical shape models." MS (Master of Science) thesis, University of Iowa, 2010.
<https://doi.org/10.17077/etd.zlutla9y>

Follow this and additional works at: <https://ir.uiowa.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

A METHOD FOR AUTOMATED LANDMARK CONSTELLATION
DETECTION USING EVOLUTIONARY PRINCIPAL COMPONENTS AND
STATISTICAL SHAPE MODELS

by

Wei Lu

A thesis submitted in partial fulfillment
of the requirements for the
Master of Science degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisor: Assistant Professor Hans J Johnson

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Wei Lu

has been approved by the Examining Committee
for the thesis requirement for the Master of
Science degree in Electrical and Computer Engineering
at the December 2010 graduation.

Thesis Committee: Hans J Johnson, Thesis Supervisor

Gary E Christensen

Vincent A Magnotta

ACKNOWLEDGMENTS

This research was supported by funding from grants NS050568 and NS40068 from the National Institute of Neurological Disorders and Stroke, grants MH31593, MH40856 from the National Institute of Mental Health, and grant U54EB005149 from the National Institute of Biomedical Imaging and Bioengineering.

I would like to give my most sincere gratitude to my supervisor Hans J Johnson, who has made his support in a number of ways in my thesis writing, research work, academic study, and job searching. I would like to thank all members of the *BRAINS* software development team who provided me with insightful ideas and generous help in cultivating the method, preparing test datasets, and developing the software. My thanks also go to my family who are always my strongest supporters. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this work.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
CHAPTER	
1 INTRODUCTION	1
1.1 Motivations	1
1.2 Challenges	2
1.3 Thesis overview	2
2 BACKGROUND	3
2.1 Landmark constellation detection	3
2.2 Spatial localization using principal components	4
2.3 Image interpretation using statistical shape models	6
2.4 Image registration	6
2.4.1 Image registration challenges	7
2.4.2 General image registration framework	8
2.4.3 The definition of a registered brain image	9
2.5 Atlas construction using generalized splines	9
2.6 Brain imaging using magnetic resonance	11
2.7 Sphere detection using radial Hough transform	12
3 METHOD	17
3.1 Method overview	17
3.1.1 Constellation modeler	19
3.1.2 Constellation detector	21
3.1.3 Anatomical relationships	22
3.2 Detection methods for base landmarks	22
3.2.1 Centroid of head mass estimation	23
3.2.2 Mid-sagittal plane transform estimation	24
3.2.3 Eye detection using radial Hough transform	27
3.2.4 Morphometric constraining	30
3.3 Linear model estimation using evolutionary PCA	32
3.4 Local search using statistical shape models	37
3.5 Resampling in-place	42
3.6 Landmark estimation validation criteria	45
4 RESULT	47
4.1 Landmark constellation detection	47
4.2 Atlas construction	51
5 CONCLUSION	56

5.1	Contributions	56
5.2	Future work	58
APPENDIX		
A	ABBREVIATIONS AND NOTATIONS	59
B	NOTE ON PROBABILISTIC RADIAL HOUGH TRANSFORM	63
C	NOTATIONAL REFERENCE FOR APPENDIX B	66
D	NOTATIONAL REFERENCE FOR LME-EPCA ALGORITHM (SECTION 3.3)	67
E	NOTE ON PRINCIPAL COMPONENT ANALYSIS	69
F	CONSTELLATION PROFILES	72
G	BRAINS CONSTELLATION MODELER MANUAL	79
H	BRAINS CONSTELLATION DETECTOR MANUAL	84
I	BRAINS CONSTELLATION GUI CORRECTOR MANUAL	90
J	SOFTWARE USAGE FOR ITK RESAMPLE IN-PLACE IMAGE FILTER	94
	REFERENCES	95

LIST OF TABLES

Table

3.1	This table shows the <i>ac</i> locations in input/ output and index/physical space. The output image has the same voxel contents as that of the input image but with different physical representation due to the impact of an input rigid transform.	44
4.1	Detection errors for several base, midbrain, and off-midbrain landmarks in 10 T_1 and T_2 human brain datasets. RMS = root mean squared (error), Std = standard deviation.	48
4.2	Midbrain and off-midbrain landmark detection errors improvement by local search. RMS = root mean squared (error), Std = standard deviation.	49
4.3	Mean errors in LPS space (in <i>mm</i>) for a few base landmarks. RMS = root mean squared (error)	50

LIST OF FIGURES

Figure

2.1	A 2D illustration showing the rationale behind radial Hough transform	13
2.2	A 2D illustration showing the rationale behind probabilistic radial Hough transform	14
2.3	A 2D illustration comparing the deterministic and probabilistic radial Hough transform on circle radius robustness	16
3.1	Overview of the landmark detection process in this work in top-level data flow diagram and collaboration diagram	18
3.2	Level 1 data flow diagrams for the landmark modeler and detector in this work	20
3.3	Center of head mass is more robust than the center of tissue mass for images with different amount of neck and shoulder.	24
3.4	A demonstration of the estimated mid-sagittal plane	26
3.5	Distribution of human adult interpupillary distance for the entire ANSUR database	28
3.6	Eye center detection using radial Hough transform	29
3.7	The information of <i>CEC</i> improves the detection for <i>ac</i> and $4VN$	30
3.8	Dispersion comparison between constellation distribution of the training dataset and error residuals between linear estimation result and manual labeling result (3D)	40
3.9	Dispersion comparison between constellation distribution of the training dataset and error residuals between linear estimation result and manual labeling result in sagittal view	41
3.10	Resample in-place demonstration	44
4.1	Consistency comparison between automatically detected <i>pc</i> and manual labeled <i>pcs</i> for <i>pc</i>	50
4.2	One of the failing cases in the automated test due to bad scans without eye response as the magnetic field is interfered by metals such as metallic false teeth	51
4.3	An example result of applying thin-plate spline transform to brain images	52

4.4	Comparison of brain atlases using different number of automatically detected landmarks in central slices	53
4.5	Comparison of brain atlases using different number of automatically detected landmarks in selected slices	54
B.1	Geometry of probabilistic radial Hough transform	64
F.1	Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the centroid of anterior commissure (<i>ac</i>) and the superior aspect of posterior commissure (<i>pc</i>)	73
F.2	Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the midbrain pontine junction (<i>MPJ</i>) and the fourth ventricle notch (<i>4VN</i>)	74
F.3	Typical acpc-aligned, T_1 -weighted central slices showing the junction of cerebral aqueduct and the fourth ventricle (<i>aq-4V</i>) and basal pons (<i>BPons</i>)	75
F.4	Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for genu in anterior corpus callosum (<i>genu</i>) and the pointed tip of rostrum of the corpus (<i>rostrum</i>)	76
F.5	Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for centroid of optic chiasm (<i>optic_chiasm</i>) and the centroid of the left/right eye (<i>LE/RE</i>)	77
F.6	Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the most anterior CSF in left/right ventricular nucleus (<i>left/right_ventricular_head</i>)	78
I.1	A screenshot for <i>BRAINS</i> constellation GUI corrector	92

CHAPTER 1 INTRODUCTION

1.1 Motivations

Medical imaging technologies such as MRI, CT, PET, *etc.* enable the use of higher resolution 3D digital image data for research and clinical treatment. The new technologies provide improved spatial resolution at the cost of increased data processing time. Manual identification of anatomical landmarks is still a common practice in many neuroimaging and other medical imaging applications. It often takes a neuroradiologist or neurosurgeon up to a few minutes to manually label a landmark. Manually labeling of characteristic brain structures is commonly treated as a golden standard for landmark identification, but it is labor-intensive, subjective, and suffers from intra-/inter-rater inconsistency.

A natural and possible resolution to the difficulties of manual labeling would be to develop a computer-aided method to conquer the challenge. However, this also indicates that the program has to be “intelligent” enough to “understand” the scene. Although the computer algorithm cannot know all the related domain knowledges of neuroanatomy, pathology, physiology, psychiatry, and radiology, *etc.* that a human expert would probably apply to achieve the complex human placement of landmark points, it has to “learn” patterns to detect different landmarks from some very limited source of information such as image intensity distribution.

An objective of this work is to provide an automated, consistent, and efficient method of detecting important landmarks in human brain by extracting the information of morphometric relationships among the landmarks and the landmark intensity information in high resolution medical imaging data. The detected landmarks could define the Talairach space that is commonly used in stereotactic neurosurgery, and

provide vital information in assisting atlas construction, tissue classification, and surface analysis, *etc.* Note the landmark constellation detection method discussed in the thesis was originally developed and optimized for brain images, but is suitable to a larger class of medical imaging problems.

1.2 Challenges

The first challenge stems from the difference of input images. The proposed method must be applicable to different kinds of images with various modalities, orientation, spacing, and origin; and in a longitudinal and multisubject scenario. The method must also be robust to noisy data and have a reasonable exception handling mechanism. Another major challenge springs from the definition of landmarks. For example, in the task of brain landmark labeling, the number of salient and accurately locatable landmarks of the morphology of the visible anatomy is rather few. On the other hand, the vast majority of the landmarks have high localization uncertainties in all 3 spatial directions. To find consistent locations for such landmarks are often very tough even for a human expert.

1.3 Thesis overview

In this work, an automated landmark constellation detection method is presented. The motivation and major challenges are given in **Chapter 1**. **Chapter 2** provides the related background knowledges for this study. **Chapter 3** discusses the method used in the research work. **Chapter 4** validates the proposed method in numerous aspects. **Chapter 5** concludes with a summary of contributions and future work. Brain landmark constellation profiles, method proofs, and manuals of corresponding software can be found in corresponding appendices.

CHAPTER 2 BACKGROUND

2.1 Landmark constellation detection

In morphometrics, landmarks are good for the study of covariance of biological form. Biologists like to refer to landmarks as “homologous” which means that landmarks are the similarity between characteristics of organisms that is due to their shared ancestry [1]. Point landmarks can be

Anatomical/normal: salient and accurately locatable points of the morphology of the visible anatomy, usually identified interactively by the user [2]. A normal point landmark has a distinctive intensity template and low localization uncertainties in all three spatial directions. In human brain, anterior commissure (*ac*) and the midbrain pontine junction (*MPJ*) are such landmarks.

Geometrical/quasi: points at the locus of the optimum of some geometric property, *e.g.* local curvature extrema, corners *etc.*, generally localized in an automatic fashion [2]. Such points are not uniquely definable in all directions, and they are used, for example, in the reference system of Talairach to define the 3D bounding box of the human brain. The incorporation of such landmarks is important since normal point landmarks are hard to define, for example, at the outer parts of the human head [3].

A list of landmarks investigated in this work are discussed in **Appendix F**.

Landmark-based registration methods are versatile in the sense that they can be utilized in a wide class of imaging applications. Landmark-based methods are mostly used to define rigid, affine, elastic body spline, thin plate spline transformations, *etc.* If the sets of points are large enough, they can be used for more complex transformations such as spline kernels transform. Anatomical landmarks are often

used to initialize high-dimensional registration methods that rely on optimization of parameter spaces that are not convex and are prone to getting stuck in local optima, possibly resulting in a large registration mismatch. By constraining the search space according to anatomical landmarks, such mismatches are less likely to occur. Moreover, the search procedure can be sped up considerably [2].

Algorithms for landmark detection are important for clinical applications since manual selection of landmarks is time-consuming and often lacks accuracy. Bookstein employed a linear regression model to detect landmarks [4]. Rohr *et al.* used a semi-automatic landmark detection procedure that is based on 3D differential operators [3]. These approximation schemes have a common advantage that the related landmark-based registration methods do not require to know the exact landmark positions. Some other methods such as in the work proposed by Babak and Alvin [5] search the landmark using trained landmark templates. In application of automated MRI brain landmark detection, people are more interested in the detection of certain landmarks, especially the anterior commissure (*ac*) and the posterior commissure (*pc*). Some previously published algorithms rely on edge enhancement filters and successful localization of the corpus callosum [6], or a reasonable estimation of midbrain-pons junction (*MPJ*) [5].

2.2 Spatial localization using principal components

Principal Component Analysis (PCA) is a popular multivariate statistical technique that is used by in many scientific disciplines [7]. Its main applications are among feature extraction, dimension reduction, and classification, *etc.* PCA is applied to reducing the dimensionality of the landmark feature space while preserving the essential basis in the proposed method. Details on how to use it for high quality spatial localization will be given in **Section 3.3**.

Information theory asserts that random variables with higher variance contain

more information. PCA projects a point set \mathbf{X} in a M -dimensional space to a R -dimensional space ($R \leq M$) with minimal loss of information. Mathematically, PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate component (called the first principal component), the second greatest variance on the second coordinate component, and so on [8]. PCA is theoretically the optimum transform for given data in least square terms.

Given a high dimensional (M -dimensional) landmark feature space, the objective is to reduce the space dimension to R so that the representation is simplified and the processing time is reduced while vital information is still well preserved by PCA. Given N observations in the space, the landmark vectors \mathbf{Z} in the new space can be represented by the landmark vectors \mathbf{X} in the original space with coordinate mapping \mathbf{W} as:

$$\mathbf{Z} = \mathbf{W}^T \mathbf{X} \quad (2.1)$$

Note it is required by PCA that \mathbf{X} is a zero mean matrix in different datasets in advance. Let $\|\mathbf{W}\| = 1$ for uniqueness, PCA states that the optimum transform should be:

$$\mathbf{W}_{opt} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_r] \quad (2.2)$$

Where

$$\mathbf{\Sigma} := E(\mathbf{X}\mathbf{X}^T) \quad (2.3)$$

$$\mathbf{w}_i := eig(\mathbf{\Sigma}, i), \forall i \in \{1, 2, \dots, R\} \quad (2.4)$$

$$R \leq rank(\mathbf{X}) \quad (2.5)$$

\mathbf{X} is a M -by- N matrix, \mathbf{W} is a M -by- R matrix, and \mathbf{Z} is a R -by- N matrix; $\mathbf{\Sigma}$ is the self-covariance of a zero mean matrix \mathbf{X} ; $eig(\mathbf{X}, i)$ takes the eigenvector of \mathbf{X} associated with its i -th largest eigenvalue; \mathbf{w}_i is called the i -th principal component.

A rigorous proof is provided in **Appendix E**.

2.3 Image interpretation using statistical shape models

The proposed method includes a process to model registration process that needs to build a 3D statistical shape model (SSM) from training datasets. The SSM has proven to be an effective basis for image interpretation and shape analysis [9]. To build a SSM is similar to “encoding” the image or the shape with its unique features. A straightforward way to achieve this is to sample and store the pixel vectors (*i.e.*, the spatial and magnitude information) around important landmarks of an image using *a priori* knowledge. Therefore, the similarity between two image regions can be computed for example, by a template-matching technique. In addition, by computing Mahalanobis distance [10] from a test image to a classified statistical shape model representing for *e.g.*, normal or disease, it is possible to determine the probability of a patient having a certain disease. Some preprocessing such as registration, normalization and denoising are often required to obtain a reasonable result for this approach. Data compression techniques such as run-length encoding, entropy encoding, predictive coding, and adaptive dictionary algorithms, *etc.* are applicable to this modeling task as well. They act as postprocessing techniques that make representations more compact.

2.4 Image registration

A registered image will greatly simplify the landmark detection process. The proposed method benefits from image registration process in that most of the landmarks are linearly estimated by a rigid transform using the information of key base landmarks so that the *ac*, *pc*, and *MPJ* are aligned. In this section, image registration and its application in brain image analysis is discussed.

2.4.1 Image registration challenges

Image registration is a prerequisite to an assortment of imaging applications in various fields, and it is widely used in remote sensing, medical imaging, computer vision *etc.* In medical imaging, it is often required for 3D reconstruction, multimodality image mappings, atlas construction and arithmetic operations such as image averaging, subtraction and correlation [11]. The registration tasks introduced here are classified into three categories according to the manner of the image acquisition:

2.4.1.1 Longitudinal analysis

Images of the same subject are acquired at different times, often on regular basis, and possibly under different conditions. The aim is to find and evaluate changes in the subject that appeared between the consecutive image acquisitions. Examples of applications: Monitoring of the healing therapy, monitoring of the tumor evolution [12].

2.4.1.2 Multimodal analysis

Images of the same subject are acquired from different sensors or scanning protocols. The aim is to integrate the information obtained from different source streams to gain more complex and detailed subject representation. Examples of applications: Combination of sensors recording the anatomical body structure like magnetic resonance image (MRI), ultrasound or computed tomography (CT) with sensors monitoring functional and metabolic body activities like positron emission tomography (PET), single photon emission computed tomography (SPECT) or magnetic resonance spectroscopy (MRS). Results can be applied, for instance, in radiotherapy and nuclear medicine [12].

2.4.1.3 Multisubject analysis

Images are acquired from different subjects. The aim is to build a statistical shape model [9] from a set of training data that share some common traits. One important application of brain image registration is the construction of brain atlases that are statistical shape models, and describe one or more aspects of brain structure and/or function and their relationships after applying appropriate registration and warping strategies [11].

2.4.2 General image registration framework

Due to the diversity of images to be registered and due to various types of degradations, it is difficult to design a universal method applicable to all registration tasks. Every method should take into account not only the assumed type of geometric deformation between the images but also noise corruption, required registration accuracy and application-dependent data characteristics. Nevertheless, many registration methods consist of the following three steps: [2]

Feature acquisition: Salient and distinctive objects such as landmarks, contours, surface boundaries, *etc.* are manually or automatically detected. For further processing, these features can be represented by their point representatives, *a.k.a.*, control points. The extracted features are the locations and intensity templates of human brain landmarks, and some other morphometric traits among them. (See detailed discussion in **Subsection 3.2.4.**)

Feature matching: In this step, the correspondence between the features detected in the moving image and those detected in the fixed or reference image is established.

Transform model estimation: The type and parameters of the warping functions, aligning the sensed image with the reference image, are estimated recursively

by means of minimizing a predefined cost function or minimizing the correspondence between the transformed sensed image and the reference image. In the proposed method, the reflective correlation metric was employed as the cost function. (See detailed discussion in **Subsection 3.2.2.1.**)

2.4.3 The definition of a registered brain image

The Talairach coordinate system is defined by making the superior surface of the anterior commissure (ac) to the center of the posterior commissure (pc) lie on a straight horizontal line. Since these two points lie on the mid-sagittal plane (MSP), the coordinate system is completely defined by requiring this plane to be vertical. The ac point is commonly defined as the origin of a Talairach space. Thus, in a left-posterior-superior (LPS) coordinate system for example, the pc point is located at $(0, -D_{acpc}, 0)$ where D_{acpc} is the distance between ac and pc . However, the Talairach system actually only defines the position of two landmarks (ac and pc) in the physical space. In the proposed method, the MSP is further required to pass the midbrain pontine junction (MPJ) to uniquely define the plane. Therefore, an accurate detection of the ac , pc , and MPJ are vital to the methods defined in this work. This definition for a register brain is not unique, one may expect to get a even better registration result by aligning all the $acpc$ -aligned moving images to one $acpc$ -aligned fixed image with additional information from more landmarks by using for example spline transforms to be introduced in the next section.

2.5 Atlas construction using generalized splines

An atlas can be defined as an image that minimizes the total warping errors between it and warped training datasets using the same transformation. However in practice, people often apply a much simpler approximation, *i.e.*, randomly choosing one image (*i.e.* the fixed image) from the training datasets, warping the other images

(*i.e.* the moving images) to the selected image, and then averaging the selected image and the warped images to obtain the atlas image. In the application of medical imaging, atlases are often obtained by landmark-based kernel transforms using generalized splines, such as the thin plate spline (TPS) [13], the volume spline (VS) [14], and the elastic body spline (EBS) [15], *etc.*

Mathematically, given N locations of matched control points \mathbf{p}_i from input space and \mathbf{q}_i from output space, $i = 1, 2, \dots, N$, the deformed point location $d(\mathbf{x})$ in the output space can be represented as a transform of a point location \mathbf{x} by

$$d(\mathbf{x}) = \mathbf{x} + \sum_{i=1}^N c_i \mathbf{S}(\mathbf{x} - \mathbf{p}_i) + \mathbf{A}\mathbf{x} + \mathbf{b} \quad (2.6)$$

Where \mathbf{S} is the spline kernel. The sum term represents a linear combination of the splines. The affine portion, *i.e.*, $\{\mathbf{A}, \mathbf{b}\}$ of the transform can be determined by first letting $\mathbf{x} = \mathbf{p}_i$, $i = 1, 2, \dots, N$ as long as there are enough well-defined matched points. Then by definition, by replacing $d(\mathbf{p}_i)$ with \mathbf{q}_i in **Equation 2.6**, the problem of finding $\{\mathbf{A}, \mathbf{b}\}$ is converted to a least squares optimization problem for the following N equations:

$$\mathbf{q}_i = \mathbf{p}_i + \mathbf{A}\mathbf{p}_i + \mathbf{b}, i = 1, 2, \dots, N \quad (2.7)$$

The coefficients c_i of kernel splines can be determined by the method proposed in [15]. Some popular kernel splines \mathbf{S} are listed as follows

Thin plate spline: There are two commonly used kernels for the thin plate splines, they are

$$\mathbf{S}(\mathbf{x}) = \|\mathbf{x}\|\mathbf{I} \quad (2.8)$$

and

$$\mathbf{S}(\mathbf{x}) = \|\mathbf{x}\|^2 \log\|\mathbf{x}\|\mathbf{I} \quad (2.9)$$

Where \mathbf{I} is the identity matrix. The first kernel is the optimal solution in 3D case such that the resulted continuously differentiable transform space contains all the landmarks, and minimizes the bending energy. The second kernel is such an optimal kernel in 2D [1].

Volume spline:

$$\mathbf{S}(\mathbf{x}) = \|\mathbf{x}\|^3 \mathbf{I} \quad (2.10)$$

Elastic body spline: There are two commonly used kernels for the elastic body splines, they are

$$\mathbf{S}(\mathbf{x}) = \alpha_1 \|\mathbf{x}\|^3 \mathbf{I} - 3 \|\mathbf{x}\| \mathbf{x} \mathbf{x}^T \quad (2.11)$$

and

$$\mathbf{S}(\mathbf{x}) = \alpha_2 \mathbf{I} - \mathbf{x} \mathbf{x}^T / \|\mathbf{x}\| \quad (2.12)$$

Where α_1 and α_2 are scalar constants.

2.6 Brain imaging using magnetic resonance

In the experiment to be discussed in **Chapter 4**, magnetic resonance (MR) image is chosen for processing because it is widely used in radiology to visualize internal human body structures; and it is a common imaging technique to generate brain images thanks to its high resolution and high contrast in different soft tissues. When protons are placed in a magnetic field, they become capable of receiving and then transmitting electromagnetic energy. The strength of the transmitted energy is proportional to the number of protons in the tissue. Signal strength is modified by properties of each proton's microenvironment, such as its mobility and the local homogeneity of the magnetic field. MR signal can be weighted to accentuate some properties and not others [16].

Among those "weighted" images, brain T_1 -weighted scans can be acquired quickly while still providing high-resolution 3D datasets and especially with a good gray matter/white matter contrast. T_1 -weighted scans are also bright on fat; subacute hemorrhage; melanin; protein-rich fluid; slowly flowing blood; paramagnetic substances: gadolinium, manganese, copper; and laminar necrosis of cerebral infarction as well. On the other hand, T_2 -weighted images are bright on increased water, as in edema, tumor, infarction, inflammation, infection, subdural collection, and methemoglobin

(extracellular) in subacute hemorrhage [16]. Proton density (PD) images are obtained either as T_1 -weighted images but with longer repetition time (T_R); or it can be obtained as T_2 -weighted images but with shorter echo time (T_E).

2.7 Sphere detection using radial Hough transform

The radial Hough transform (HT) is used to detect circles ($2D$) or spheres ($3D$) in an image. In this section, two different kinds of radial HT will be introduced: One is the deterministic one [17], the other is the probabilistic one [18]. For the deterministic one, given any edge voxel or feature voxel with large gradient magnitude $\mathbf{p}(x, y, z)$ in an image containing spheres and the radius r_t of the target sphere to detect, the voxel will vote with the same weights for the possible locations of the sphere center on all locations in the image that are r_t distant away from \mathbf{p} . That is, from the information of $\mathbf{p}(x, y, z)$ and r_t , the possible locations of the target sphere center are on a sphere surface centered at $\mathbf{p}(x, y, z)$ and with radius r_t . As a result, the center of the target sphere will be voted by all N_r voxels on the sphere surface in an ideal situation. As each location can only be voted by the voxels on a sphere surface that is centered at the location and with a radius of r_t , N_r will be the upper bound of the number of votes a location can have. Thus, a sphere center candidate is detected if the number of votes is high or very close to N_r . A demonstration showing the rationale behind the radial Hough transform is given in **Fig. 2.1**.

The sphere detection task can also be solved in a probabilistic way [18]. An illustration to show how probabilistic radial HT works is provided in **Fig. 2.2**. Given a prospective point $\mathbf{p}_0(x_0, y_0, z_0)$ (Cartesian space) of the $3D$ image and with the help of the gradient information $\vec{g}(\mathbf{p}_0)$, their contribution to determining the center of the sphere can be represented as a Gaussian probability distribution function (PDF)

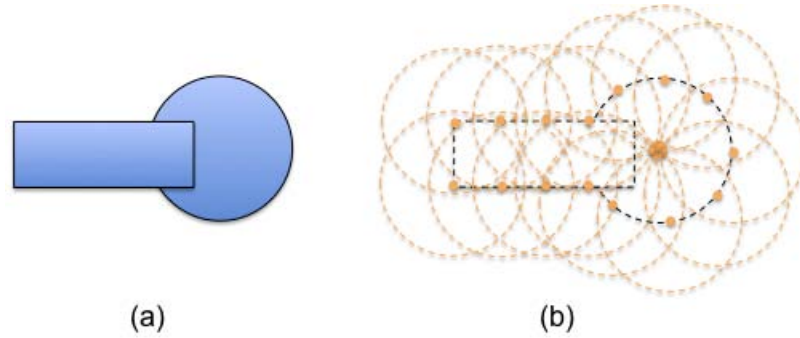


Figure 2.1: A 2D illustration showing the rationale behind radial Hough transform: (a) a dark circle partially covered by a dark square on a bright background; (b) for each sample pixel (orange dot) on the rim of the object, it votes for the possible locations of the target circle center on a “voting” circle (dashed orange circle) that is centered at itself and with the same radius of the target circle. Although the circle is partially covered by a rectangular, the center (the biggest orange dot) of the target circle still has the highest number of votes.

$f_{HT}()$ centered at $\mathbf{p}_0(x_0, y_0, z_0) + \vec{g}(\mathbf{p}_0)$ and having a $3\sigma = r_t$ of the voting radius:

$$f_{HT}(\mathbf{p}, \sigma, \mathbf{p}_0 + \vec{g}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{[\mathbf{p} - \mathbf{p}_0 - \vec{g}(\mathbf{p}_0)]'[\mathbf{p} - \mathbf{p}_0 - \vec{g}(\mathbf{p}_0)]}{2\sigma^2}\right) \quad (2.13)$$

The probability of a location to be the center of the target sphere is computed as the cumulative distribution function (CDF) that superimposes all the PDFs of prospective voxels in the image.

Both the deterministic and the probabilistic radial HT exhibit excellent robustness to image noise and partial occlusion. If the voxels on a sphere surface are affected by the noise or partially covered by another object, some voxels will not contribute voting spheres in the case of deterministic one; some gradient vectors will not perfectly point to the center of the sphere in the case of probabilistic one. However, as long as there are enough good voxels (*i.e.* edge voxels not corrupted by the noise or covered by other objects) on the sphere surface, the deterministic radial HT is able to generate enough voting spheres so that the center location of the target sphere still has high number of votes. On the other hand, even if the gradient vectors do not perfectly point from the voxel to the center of the sphere for a probabilistic radial

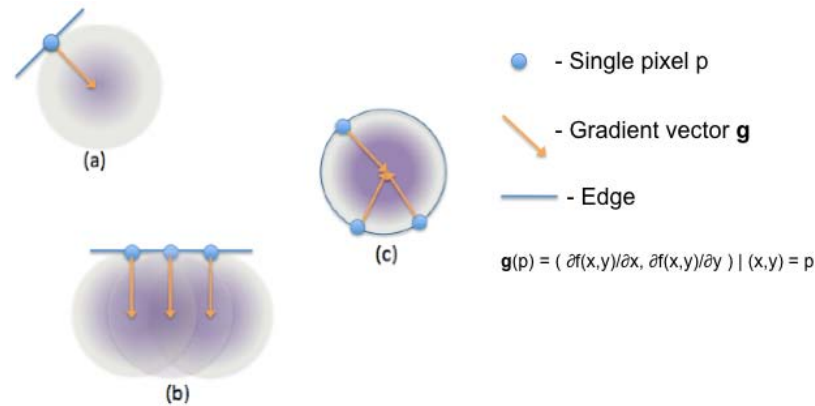


Figure 2.2: A 2D illustration showing the rationale behind probabilistic radial Hough transform: (a) A probability distribution function (PDF) in magenta spherical cloud is generated by a point (in blue) and its gradient vector (in orange); (b) the cumulative distribution function (CDF) obtained by superimposing the PDFs from three points on a line; (c) CDF obtained by superimposing three identical PDFs from three points on a dark circle on a bright background. The center of the circle corresponds to the place with highest value of CDF, *i.e.* highest probability of being the center of target circle.

HT, the sum of many close Gaussian distributions will approach to a Gaussian distribution with an averaged mean and a deviation of the sum of deviations of nearby distributions. As a result, the locations close to the center of the sphere will still have highest probabilities to be good sphere center candidates.

The circle/sphere size robustness for the deterministic radial HT depends on the difference of the target radius r_t of the input object and the radius r_v of the voting circles/spheres. The robustness for the probabilistic radial HT depends on the ratio of r_v and r_t . That is if $|r_v - r_t| > 0.5 \text{ pixel/voxel}$, deterministic radial HT will fail to detect the center of the circle/sphere; if $\frac{|r_v - r_t|}{r} > \alpha$ (α small scalar, and is shown in **Appendix B** that $\alpha \approx 0.1$ sufficient for circle detection), the probabilistic radial HT will fail the task. In applications of detecting spheres with variable size but within a small variation like different human eyes that usually vary in the range of 11 to 13 *mm* approximately [19], $|r_v - r_t|$ might be as large as 1 *mm* and if the spacing of the image is smaller than 2 *mm* in either spatial direction, the deterministic radial HT might

totally reject a target circle/sphere center due to radius mismatch as illustrated in **Fig. 2.3 (d)**. In reality, digitized human brain images usually have spacing less than 2 mm in non-scanning directions to achieve a higher resolution, which indicates the necessity of performing deterministic radial HT multiple times with different voting radius in the eye detection application. On the other hand, as the ratio $|r_v - r_t|/r_t$ is probably less than $\frac{1}{11}$ ($= \frac{|12-11|}{11}$), the Gaussian probability distribution functions are close enough to give a high cumulative distribution function value to the locations close to the target circle/sphere center as illustrated in **Fig. 2.3 (e)**.

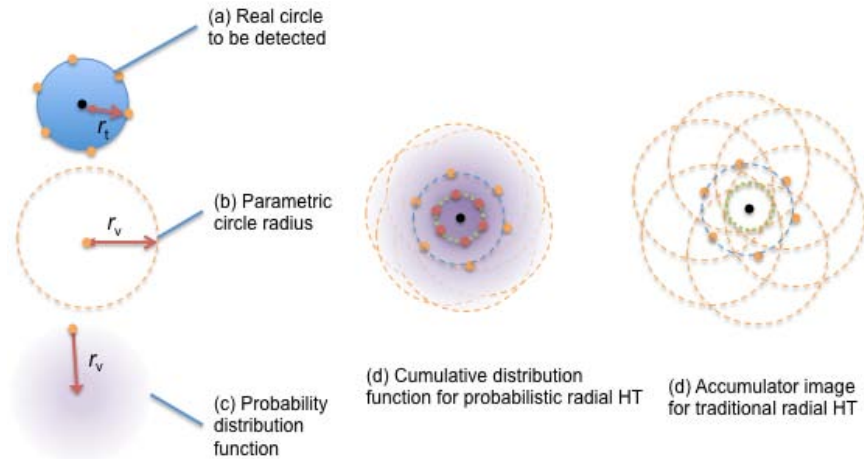
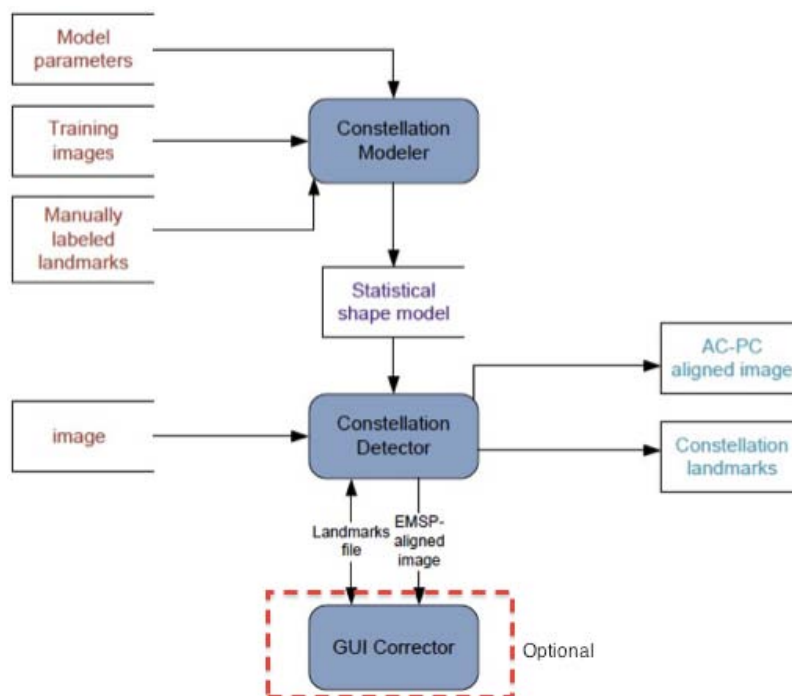


Figure 2.3: A 2D illustration comparing the deterministic and probabilistic radial Hough transform on circle radius robustness. (a) shows dark circle with radius r_t on a bright background to be detected. The black dot is the circle center and the orange points are sample pixels on the circle rim. (b) is the voting circle with radius $r_v > r_t$ for each sample pixel. (c) is the Gaussian probability distribution function (PDF) with $3\sigma = r_v$ contributed by each sample pixel. (d) shows the cumulative distribution function (CDF) by superimposing six PDFs. Each PDF (whose center is represented as a red point) is centered on the innermost green circle that is r_v distant from its related sample pixel on the blue dashed circle such that the sample pixel, center of the target circle, and the center of the related PDF are on a line. As shown in **Appendix B**, as long as the ratio of $r_v : r_t$ is close to one, the locations close to the center of the target circle will have highest CDF values. (e) shows the parameter space of using deterministic radial HT. The six sample pixel on the circle of the target circle is transformed to six voting circles centered at these sample pixels. The far end of each voting circle is on the innermost green dashed circle. As can be seen, if the diameter of the green circle is larger than one pixel due to a large difference between r_v and r_t , the surface pixels will never add votes to the center location of the target circle.

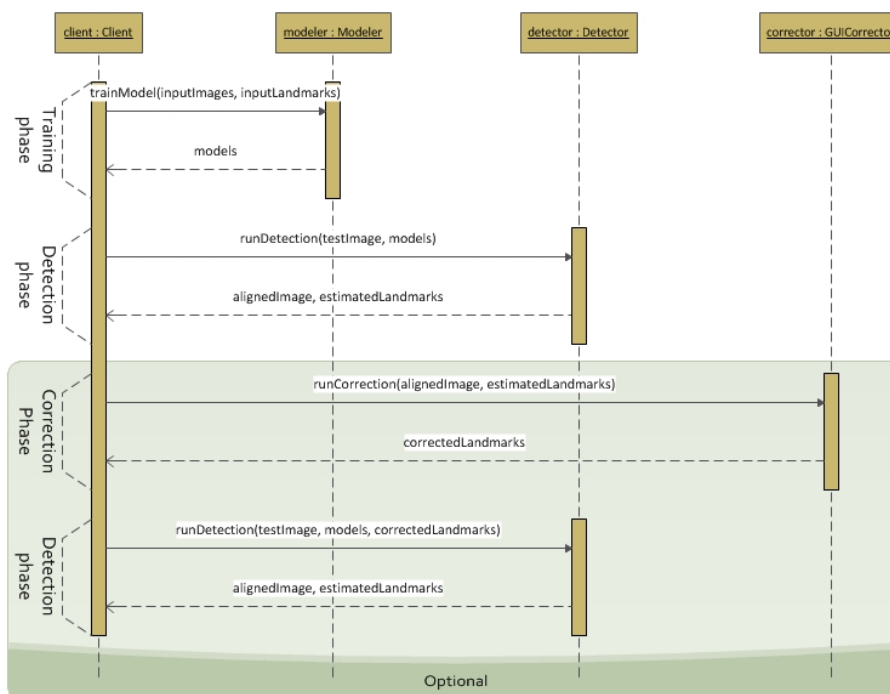
CHAPTER 3 METHOD

3.1 Method overview

Several methods are composed to achieve a more reliable estimation of landmark locations. To detect landmarks and align an input test image, the system needs to be trained on general properties of landmarks. More specifically, some related questions would be: What do landmarks look like? Where are their possible/probable locations? What's the spatial relationship among those landmarks, in terms of morphometrics? The system is taught all these characteristics in the training phase from a group of representative training datasets including images and manually labeled landmarks. The overview of the modeler can be found in **Subsection 3.1.1**. The information is then written to a statistical shape model for the detector to retrieve and apply later. On the other hand, the “constellation detector” reads in the target input image and the model file, generates an *acpc*-aligned output image and a constellation of detected landmarks with the statistical shape model. The overview of the detector can be found in **Subsection 3.1.2**. A “GUI corrector” module acts as an observer is developed to provide the user a convenient way to interact with the detector. The detector has the ability to adjust its detection process based on user feedback. Detailed information about the software can be found at [20]. The top-level data flow diagram (DFD) of the software is shown in **Fig. 3.1a**. The collaboration diagram of the software components are shown in **Fig. 3.1b**. Related software manuals can be found in **Appendix G, H, I**. The details of the proposed detection method in this work can be found through **Section 3.2 to 3.4**. The idea of minimizing interpolation errors by resampling in-place and the landmark detection validation criteria for this work is discussed in **Section 3.5** and **Section 3.6**, respectively. The rest of this section will give more details on the modeler, detector, and some anatomical relationships being



(a) Top-level data flow diagram



(b) The collaboration diagram

Figure 3.1: Overview of the landmark detection process in this work in top-level data flow diagram and collaboration diagram

used throughout the entire process.

3.1.1 Constellation modeler

The statistical shape model is built for all the landmarks in the constellation-modeling phase. The collected information includes:

- Spatial relationship between landmarks that is represented by mean vectors between landmarks in the estimated mid-sagittal plane (EMSP) aligned space. The information will be used mainly for the detection of some base landmarks in our work in **Subsection 3.2**.
- A linear model that finds principal components of each landmark and minimizes the estimation error in terms of least squares. The detailed discussion of this model can be found in **Training phase-2** of the linear model estimation method using evolutionary principal components as described in **Section 3.3**.
- Landmark template represented by the average local intensity distribution of a corresponding landmark over all training datasets. The template size should be picked in a way such that it is large enough to have sufficient intensity distinctions from others; and it should not be too large to impede the computation efficiency nor to raise too much false detection rate. The template will be used for the local search process as described in **Subsection 3.4** in the detection phase to estimate landmarks. In the proposed method, the shape of the template is set to cylinder with radius R and height H . The size of a landmark template is set to $R = 5 \text{ mm}$, $H = 10 \text{ mm}$ if the landmark has clear anatomical definition such as anterior commissure; or the size is set to $R = 8 \text{ mm}$, $H = 16 \text{ mm}$ for quasi-landmarks having large intensity localization uncertainties.

The data flow diagram of the modeler is shown in **Fig. 3.2a**. Some of the typical landmark templates are given in **Appendix F**.

3.1.2 Constellation detector

The general strategy for the detector is to estimate landmarks one by one, from simple to hard. The summary of brain landmark estimation order is given as follows:

- Center of head mass (CM) by Otsu thresholding [21] and a top-down maximum sphere radius estimation as described in **Subsection 3.2.1**
- Estimated mid-sagittal plane (EMSP) by optimizing a rigid transform with Powell's optimizer [22] implemented by Brent's method [23] that maximizes the reflective correlation of the brain image as described in **Subsection 3.2.2**
- Left/right eye centers (LE and RE) by radial Hough transform [17] as described in **Subsection 3.2.3**
- Midbrain pontine junction (MPJ) by template matching-based local search method centered at CM as described in **Subsection 3.4**
- Search center for the fourth ventricle notch ($4VN$) by morphometric constraining (MC) among center of eye centers (CEC), MPJ , and $4VN$ as described in **Subsection 3.2.4**
- Search center for anterior commissure (ac) by MC among CEC , MPJ , and ac as described in **Subsection 3.2.4**
- Search center for posterior commissure (pc) and the rest of landmarks by linear model estimation using evolutionary principal components (LME-EPCA) as described in **Subsection 3.3**
- Each search center is used in a corresponding landmark template matching local search process to estimate the landmark location as described in **Subsection 3.4**.

The data flow diagram of the detector excluding the LME-EPCA process is shown in **Fig. 3.2b**.

3.1.3 Anatomical relationships

Practical anatomical relationships are constructed from the observation of landmarks in training datasets. In this subsection and in **Subsection 3.2.4**, the symbol \mathbf{p}_x means a landmark point named x . \mathbf{q}_x refers to the landmark point in the training datasets. $\angle \mathbf{p}_x \mathbf{p}_y \mathbf{p}_z$ means the angle from landmark x to landmark y to landmark z . The relationships include:

1. Human brains share a similar angle of $\angle \mathbf{p}_{CEC} \mathbf{p}_{MPJ} \mathbf{p}_{4VN}$
2. Human brains share a similar angle from $\angle \mathbf{p}_{CEC} \mathbf{p}_{MPJ} \mathbf{p}_{ac}$
3. Human brains share a similar landmark distance $\|\mathbf{p}_{4VN} - \mathbf{p}_{MPJ}\|$
4. Human brains share a similar landmark distance $\|\mathbf{p}_{ac} - \mathbf{p}_{MPJ}\|$
5. $4VN$ is always below or inferior to the $\overrightarrow{\mathbf{p}_{MPJ} - \mathbf{p}_{CEC}}$ line on the EMSP plane
6. ac is always above or superior to the $\overrightarrow{\mathbf{p}_{MPJ} - \mathbf{p}_{CEC}}$ line on the EMSP plane
7. The CEC , MPJ , $4VN$, ac and pc are all very close to the EMSP plane
8. Landmarks share a linear relationship in location

3.2 Detection methods for base landmarks

In this section, a few methods for base landmarks detection are introduced. The base landmarks are used for arbitrary landmarks detection by a linear estimation method as described in **Section 3.3**. The base landmark detection method for centroid of head mass (CM) is described in **Subsection 3.2.1**; the mid-sagittal plane

estimation is described in **Subsection 3.2.2**; eye detection method is given in **Subsection 3.2.3**; detection method utilizing the morphometric constraints among the landmarks is described in **Subsection 3.2.4**.

3.2.1 Centroid of head mass estimation

As discussed in the overview of **Chapter 3**, the proposed method will detect landmarks in a simple to hard, and special to general way. The centroid of head mass (CM) is estimated first because it can be found independently. It only needs to be a rough estimation of anatomical orientation but it acts as a very important reference point for the rest of processes. Two directly affected processes are the estimation of mid-sagittal plane transform and the estimation of eye centers.

To estimate the CM , the tissue region needs to be segmented from noisy background. Fortunately, there exist established clustering methods to deal with the problem. Among them, K -means variation [24] and Otsu thresholding [21] are two relatively simple and commonly used methods. In the proposed method, Otsu thresholding is used to find the optimal threshold value to best separate the foreground image from background image such that the within-class variance or the sum of variances of background image and foreground image is minimized.

In real clinical MRI datasets, head scans sometimes contain a large variable portion of neck and shoulder tissue. Rather than pick the center of mass as a stable reference point, the proposed method calculates the centroid of head mass (CM) that is estimated through a top-down maximum sphere radius estimation process. It starts by scanning the foreground object along superior to inferior direction if it is in an LPS space; it stops at the slice such that the foreground volume above the slice is greater than the head sphere volume estimated by the max-so-far cross section area in the foreground. Finally, the center of head mass is calculated by finding the center of mass of the foreground volume above the most inferior slice. The improvement can

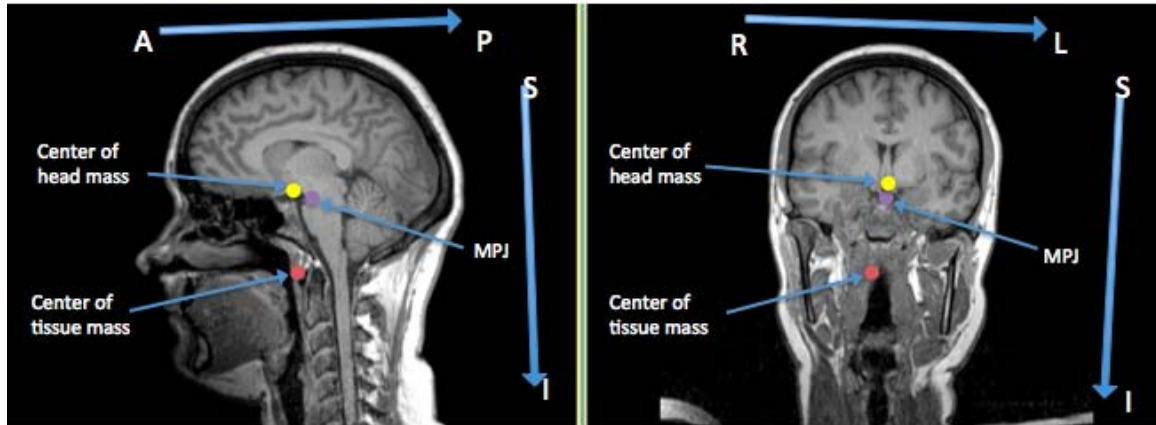


Figure 3.3: The center of head mass in yellow is very close to the *MPJ* in purple as is expected, and its location is more robust than that of the center of tissue mass of the entire foreground object in red to the amount of neck and shoulder portion in the input image. The image is shown in a left-right, posterior-anterior, superior-inferior (LPS) coordinate system.

be seen in **Fig. 3.3**.

3.2.2 Mid-sagittal plane transform estimation

As mentioned in **Section 2.4** that adequate registration can improve the quality of landmark detection. The MSP transform is used to find a rough aligned image in a phase that the system has very little knowledge about most of the landmarks so that base landmarks can be estimated more easily. The transform is estimated by repeatedly

- Applying a 3D rigid transform $T(\chi)$ to the coordinate of the input image
- Calculating the reflective correlation (RC) of the image in the transformed space
- Iterating and maintaining the rigid transform parameter set χ with maximum RC

3.2.2.1 Reflective correlation metric

In statistics, the correlation of two random variables \mathbf{X} and \mathbf{Y} can be measured by correlation coefficient $\rho_{\mathbf{XY}}$ as

$$\rho_{\mathbf{XY}} = \frac{cov(\mathbf{X}, \mathbf{Y})}{\sqrt{\mathbf{D}(\mathbf{X})\mathbf{D}(\mathbf{Y})}} \quad (3.1)$$

Where $cov(\mathbf{X}, \mathbf{Y})$ is the covariance of variable \mathbf{X} and \mathbf{Y} . $\mathbf{D}(\mathbf{X})$ is the variance of variable \mathbf{X} . The more linearly dependent of \mathbf{X} and \mathbf{Y} , the closer the value of $\rho_{\mathbf{XY}}$ is to 1. For the conveniency of later explanation on reflective correlation, $cov(\mathbf{X}, \mathbf{Y})$ is expressed in a sum of product of elements in \mathbf{X} and \mathbf{Y} as follows,

$$cov(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^N (x_i - EX)(y_i - EY) \quad (3.2)$$

Where N is the number of elements in \mathbf{X} and \mathbf{Y} ; EX is the expected value for \mathbf{X} .

In the case of reflective correlation, if \mathbf{X} refers to all the pixels in the left half of a bounding box centered at CM of a brain image in LPS coordinate system, then \mathbf{Y} refers to the pixels in the right half of the bounding box as shown in **Fig. 3.4**. Moreover, $\forall \mathbf{x}_i \in \mathbf{X}, \mathbf{y}_i \in \mathbf{Y}$ in **Equ 3.2**, pixel pair \mathbf{x}_i and \mathbf{y}_i satisfy

$$\begin{aligned} \mathbf{x}_i \cdot \mathfrak{L} + \mathbf{y}_i \cdot \mathfrak{L} &= 2\mathbf{p}_{CM} \cdot \mathfrak{L} \\ \mathbf{x}_i \cdot \mathfrak{P} &= \mathbf{y}_i \cdot \mathfrak{P} \\ \mathbf{x}_i \cdot \mathfrak{S} &= \mathbf{y}_i \cdot \mathfrak{S} \end{aligned} \quad (3.3)$$

Where subscriptions $\mathfrak{L}, \mathfrak{P}, \mathfrak{S}$ represents the left-right, posterior-anterior, and superior-inferior component of the location of a pixel in LPS coordinate system, respectively. \mathbf{p}_{CM} is the location for landmark CM (*i.e.* center of head mass).

3.2.2.2 Powell's optimizer

Powell's conjugate gradient descent method, or simply Powell's method [22] is a real-valued and digitized (*i.e.*, designed for real-valued and finite input/output) algorithm for finding the optimal parameters to minimize a multidimensional function without requiring explicitly defined derivatives of the function. In this work, the parameter set is $\chi = \{\theta, \phi, t\}$ that includes a translation t along left-right direction,

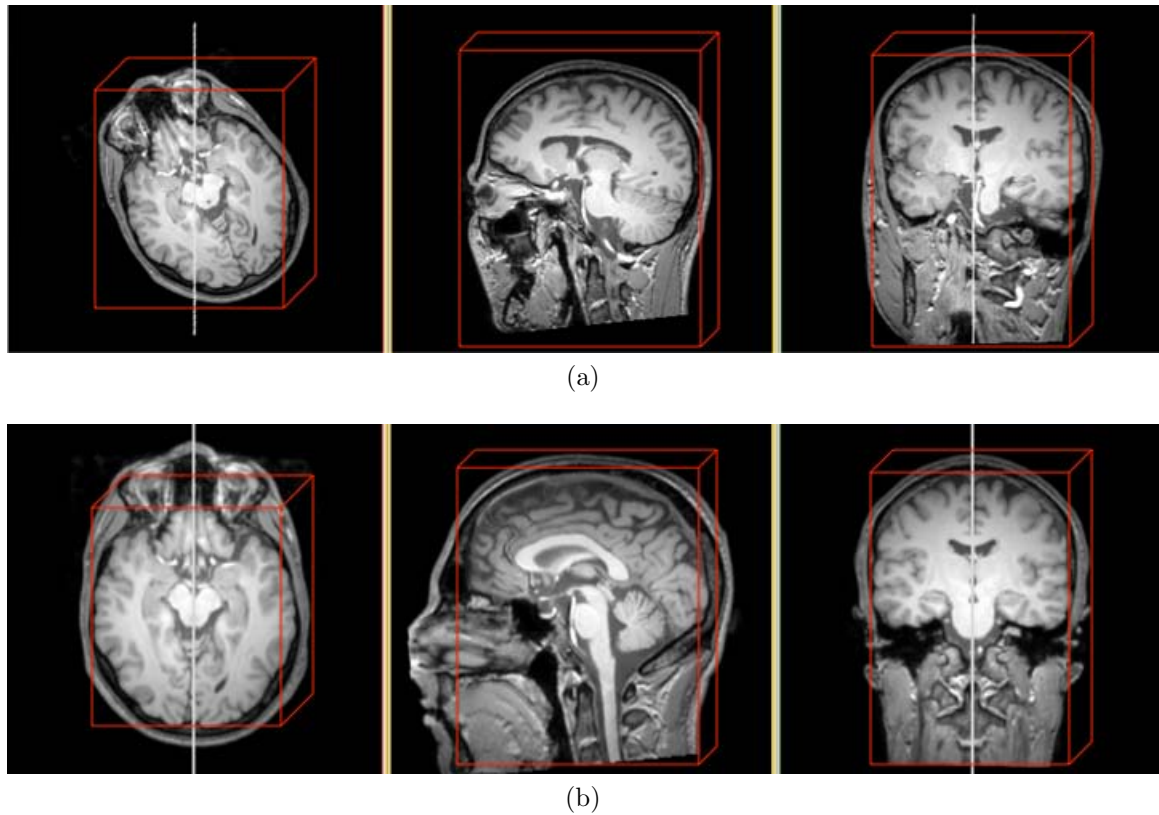


Figure 3.4: A demonstration of the estimated mid-sagittal plane in sagittal view (middle), and as a white line in axial view (left) and coronal view (right) along with center slices of (a) original image and (b) transformed image by maximizing the reflective correlation using Powell's optimizer. The reflective correlation is computed in the red bounding box of size $190 \times 260 \times 320 \text{ mm}^3$.

a rotation angle θ about the posterior-anterior direction, and a rotation angle ϕ about the superior-inferior direction. As the estimated mid-sagittal plane (EMSP) is initialized at the center of head mass (CM) and fixed at plane $\mathfrak{L} = 0$ where \mathfrak{L} is the left-right component in a LPS coordinate system, the goal of the optimizer is to iteratively find an optimal rigid transform so that the reflective correlation (discussed in **Subsection 3.2.2.1**) of the transformed brain image inside a centered bounding box is maximized in left-right direction. The search direction, iteration step size, *etc.* are all controlled by Powell's optimizer. A typical EMSP is demonstrated in **Fig. 3.4**.

3.2.3 Eye detection using radial Hough transform

Radial Hough transform tells how to detect a general sphere. To detect adult human eyes, a set of parameters should be set properly. The following parameters are considered in the proposed method in this work.

Diameter of eye: is set to 24 *mm*. According to *Encyclopedia Britannica Macropedia*, "the dimensions of the eye are reasonably constant, varying among individuals by only a millimetre or two; the sagittal (vertical) diameter is about 24 millimetres (about one inch) and is usually less than the transverse diameter." [19]

Normal adult interpupillary distance (IPD) range: is set to 40 to 85 *mm*. According to Dodgson's survey [25] on the ANSUR database containing about four thousand subjects aged 17 to 51, the mean IPD is about 63 *mm*, and is most likely between 50 *mm* to 75 *mm*. A histogram from the survey is shown in **Fig. 3.5**. The eye detector will report an exception if the resulted human adult interpupillary distance (IPD) is abnormal.

The region of interest (ROI): is set to a truncated spherical sector (*aka* spherical

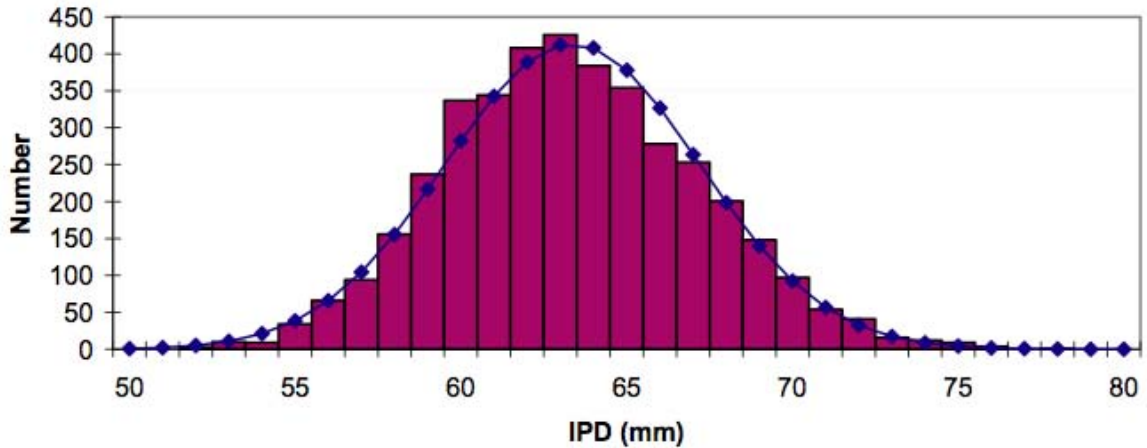


Figure 3.5: Distribution of human adult interpupillary distance (IPD) for the entire ANSUR database (3976 subjects). The bars show the number of subjects with each IPD (measured to the nearest millimetre). The dotted line shows the normal distribution with the same mean and standard deviation, for comparison. Figure courtesy of Neil A. Dodgson, Proc. SPIE Vol. 5291 [25]

cone) centered at CM and facing the anterior direction in a LPS coordinate system. Mathematically,

$$ROI = S_1 - S_2 \quad (3.4)$$

Where S_1 and S_2 are two spherical sectors with the same center O , the same spread angle θ , and different radius $R_1 > R_2$, the operator $A - B$ returns the region of A excluding B . Based on some common sense (*e.g.* eyes are on the front of head, and they should not be very close to the center of head mass) and verified by practical experiment, in the proposed method in this work, R_1 is set to 120 mm, R_2 is set to 30 mm, and the spread angle θ is set to 2.4 rad. They are tuned in an experiment on 20 training datasets so that all of the eyes fall into the proposed ROI. A typical ROI for eyes and superposed on a clinical brain image can be found in **Fig.** 3.6a.

The accumulator image for eye centers are shown in **Fig.** 3.6b. The higher the intensity is the more likely the location is for eye center. After the first eye



(a) A typical ROI image for eye centers in (a) axial view, (b) sagittal view, and (c) coronal view



(b) A typical accumulator image for eye centers in (a) axial view, (b) sagittal view, and (c) coronal view

Figure 3.6: Eye center detection using radial Hough transform

center is found, a spherical region with radius of eye at the location of first eye is excluded from next search process to avoid the disturbance from local extrema. The eye centers estimated by the Hough eye detector are considered to be the automated detection result for eye centers (*i.e.* no local search process is involved in estimating eye centers). Finally, the center of eye centers (\mathbf{p}_{CEC}) is computed as $(\mathbf{p}_{LE} + \mathbf{p}_{RE})/2$.

- The search center of MPJ is set to CM .
- The search centers of ac , and $4VN$ are estimated by the morphometric constraining approach.
- The search center of pc is estimated by a direct linear model estimation model.
- The search center of the rest of landmarks is estimated by linear model estimation using LME-EPCA (in **Section 3.3**).

3.2.4 Morphometric constraining

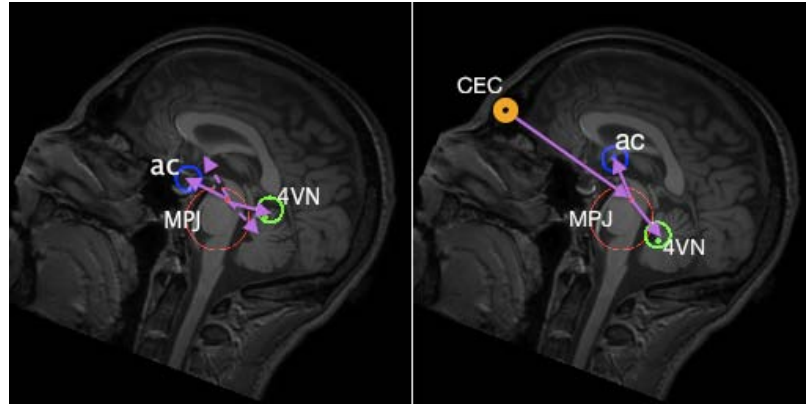


Figure 3.7: The information of CEC in orange improves the detection for ac in blue and $4VN$ in green especially when there is a large rotation difference along the left-right direction between the test image space the training space. (There is a 30 degree rotation difference in the presented image space.) MPJ is shown in red. A circle represents the search boundary of the landmark on the slice. A solid dot inside the circle and with the same color represents the local searched result for the landmark. For example in the left inset, if there is no such rotation difference, ac and $4VN$ can be predicted closely to their anatomical locations as indicated by dashed pointers; however, the average MPJ to $ac/4VN$ vector is vulnerable to the rotation and get bad result in this case as shown in solid pointer. In the right inset, an average CEC to MPJ to $ac/4VN$ angle is computed in the training phase, and can cancel off the rotation difference between the test image space and the training space so that the ac and $4VN$ can be estimated accurately.

In the proposed method, landmarks like ac and $4VN$ are estimated from their morphometric relationship with MPJ and CEC . The detection improvement of ac by adding the CEC information is demonstrated in **Fig. 3.7**. To show how it works, recall the notations used in **Subsection 3.1.3** that the symbol \mathbf{p}_x means a landmark point named x . \mathbf{q}_x refers to the landmark point in the training datasets. $\angle \mathbf{p}_x \mathbf{p}_y \mathbf{p}_z$ means the angle from landmark x to landmark y to landmark z . Any landmark points mentioned in this subsection are in the left-posterior-superior coordinate system. $\mathbf{p}_x \cdot \mathcal{L}$ is the left-right component of landmark point x , $\mathbf{p}_x \cdot \mathcal{P}$ is the posterior-anterior component of x , $\mathbf{p}_x \cdot \mathcal{S}$ is the superior-inferior component of x .

By **Relationship 2** and **4** in **Subsection 3.1.3**, the two constraints can be approximated as the following equations:

$$\begin{aligned} \cos \theta &:= \cos \angle \mathbf{p}_{CEC} \mathbf{p}_{MPJ} \mathbf{p}_{ac} = \cos \angle \mathbf{q}_{CEC} \mathbf{q}_{MPJ} \mathbf{q}_{ac} \\ &= \frac{(\mathbf{q}_{CEC} - \mathbf{q}_{MPJ}) \cdot (\mathbf{q}_{ac} - \mathbf{q}_{MPJ})}{\|\mathbf{q}_{CEC} - \mathbf{q}_{MPJ}\| \cdot \|\mathbf{q}_{ac} - \mathbf{q}_{MPJ}\|} \end{aligned} \quad (3.5)$$

$$\|\mathbf{p}_{ac} - \mathbf{p}_{MPJ}\| = \|\mathbf{q}_{ac} - \mathbf{q}_{MPJ}\| \quad (3.6)$$

By **Relationship 7** in **Subsection 3.1.3**, constraints can be approximated as the following equations:

$$\mathbf{p}_{ac} \cdot \mathcal{L} = \mathbf{p}_{CEC} \cdot \mathcal{L} = \mathbf{p}_{MPJ} \cdot \mathcal{L} \quad (3.7)$$

By **Equ 3.5** and **3.7**, vector **Equ 3.6** can be extended to the following 2nd-order equations:

$$\begin{aligned} &(\mathbf{p}_{CEC} \cdot \mathfrak{P} - \mathbf{p}_{MPJ} \cdot \mathfrak{P}) \cdot (\mathbf{p}_{ac} \cdot \mathfrak{P} - \mathbf{p}_{MPJ} \cdot \mathfrak{P}) + \\ &(\mathbf{p}_{CEC} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S}) \cdot (\mathbf{p}_{ac} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S}) \\ &= \|\mathbf{p}_{CEC} - \mathbf{p}_{MPJ}\| \cdot \|\mathbf{q}_{ac} - \mathbf{q}_{MPJ}\| \cos \theta \end{aligned} \quad (3.8)$$

$$\begin{aligned} &(\mathbf{p}_{ac} \cdot \mathfrak{P} - \mathbf{p}_{MPJ} \cdot \mathfrak{P})^2 + (\mathbf{p}_{ac} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S})^2 \\ &= (\mathbf{q}_{ac} \cdot \mathfrak{P} - \mathbf{q}_{MPJ} \cdot \mathfrak{P})^2 + (\mathbf{q}_{ac} \cdot \mathfrak{S} - \mathbf{q}_{MPJ} \cdot \mathfrak{S})^2 \end{aligned} \quad (3.9)$$

Solve **Equ 3.8** and **3.9** with **Relationship 6** in **Subsection 3.1.3**, the posterior-anterior component and the superior-inferior component of ac can be estimated by the following expressions:

$$\mathbf{p}_{ac} \cdot \mathfrak{S} = \mathbf{p}_{MPJ} \cdot \mathfrak{S} + \frac{-b + \sqrt{\Delta}}{2a} \quad (3.10)$$

$$\begin{aligned} \mathbf{p}_{ac} \cdot \mathfrak{P} &= \mathbf{p}_{MPJ} \cdot \mathfrak{P} + \frac{1}{\mathbf{p}_{CEC} \cdot \mathfrak{P} - \mathbf{p}_{MPJ} \cdot \mathfrak{P}} \\ &\cdot [\|\mathbf{p}_{CEC} - \mathbf{p}_{MPJ}\| \cdot \|\mathbf{q}_{ac} - \mathbf{q}_{MPJ}\| \cos \theta \\ &- (\mathbf{p}_{CEC} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S}) \cdot (\mathbf{p}_{ac} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S})] \end{aligned} \quad (3.11)$$

Where

$$a = \|\mathbf{p}_{CEC} - \mathbf{p}_{MPJ}\|^2$$

$$b = -2\|\mathbf{p}_{CEC} - \mathbf{p}_{MPJ}\| \cdot \|\mathbf{p}_{ac} - \mathbf{p}_{MPJ}\| \cdot (\mathbf{p}_{CEC} \cdot \mathfrak{S} - \mathbf{p}_{MPJ} \cdot \mathfrak{S}) \cdot \cos \theta$$

$$c = \|\mathbf{q}_{ac} - \mathbf{q}_{MPJ}\|^2 \cdot \|\mathbf{p}_{CEC} - \mathbf{p}_{MPJ}\|^2 \cdot \cos^2 \theta$$

$$\begin{aligned}
& - (\mathbf{p}_{CEC} \cdot \mathfrak{P} - \mathbf{p}_{MPJ} \cdot \mathfrak{P})^2 \\
\Delta & = b^2 - 4ac
\end{aligned} \tag{3.12}$$

As MPJ is already obtained by a direct local search centered at CM , it will be straightforward to get the search center of \mathbf{p}_{ac} from **Equ** 3.7, 3.10 to 3.12. The search center for $4VN$ can be found in a similar way to that of ac . Simply replace all ac -related variables to $4VN$ -related variables in **Equ** 3.10 to 3.12. Also change the sign before $\sqrt{\Delta}$ in **Equ** 3.10 due to **Relationship** 5 in **Subsection** 3.1.3.

3.3 Linear model estimation using evolutionary PCA

Before this phase, several base landmarks have already been estimated by methods as described in **Section** 3.2. The rest of the landmarks can be estimated in a more general way such as employing a linear model estimation method. However, some potential problems exist when applying linear model estimation directly to the arbitrary landmark detection task:

- If it estimates landmarks from a fixed number of landmarks, it does not know which landmarks contain better information for the detection task.
- Linear estimation does not scale well, *i.e.* it is not capable of extracting the most efficient, with no redundant basis to represent a new landmark vector especially when the number of new landmarks is extremely large.

To overcome these obstacles of using linear model estimation alone, the proposed method in this work introduces an algorithm, named linear model estimation using evolutionary PCA (LME-EPCA). The goal is to predict landmarks from the knowledge of some already estimated landmarks, and evolutionarily take the advantage of the additional landmarks for a better prediction. The basic idea of PCA and its application in dimension reduction can be found in **Section** 2.2. For clarity, the landmarks estimated in this phase are called **EPCA landmarks**. LME-EPCA

algorithm asserts that the following statements are true:

- The landmark vectors are linear combinations of a certain basis set.
- For each EPCA landmark, the landmark vector observations are distributed normally in the feature space.
- Large variances have important dynamics or information.
- Rotation and scaling are good enough to find an initial coordinate basis as PCA involves only rotation and scaling.

The rest of the section is devoted to a detailed description of the LME-EPCA algorithm. A notational reference for this section can be found in **Appendix D**. First of all, the algorithm is composed of three phases, **training phase-1** for training principal components, **training phase-2** for training optimal linear model coefficients, and **detection phase** for using the training information to find arbitrary EPCA landmarks. For simplicity, if it is not stated otherwise in this section, i represents the iteration index or EPCA landmark index; j represents the training dataset index; k represents the landmark index. The scalar superscription j of a landmark \mathbf{x}^j indicates that the variable is related to the j th training dataset ($j = 1, 2, \dots, N$).

Training phase-1 Collect the following information in the training datasets: (a) landmark dimensions K ; (b) number of training datasets N ; (c) a constellation of B base landmarks $\{\mathbf{x}_k | k = 1, 2, \dots, B\}$ (K -by-1 vectors) to be estimated by the methods mentioned in **Section 3.2**; (d) T EPCA landmarks $\{\mathbf{x}_k | k = B + 1, B + 2, \dots, B + T\}$ to be estimated by the LME-EPCA algorithm, where \mathbf{x}_{B+i} is the i th EPCA landmark to be estimated in processing order.

The algorithm will first construct a landmark vector space matrix \mathbf{X}_i for each

iteration as

$$\mathbf{X}_i := \begin{pmatrix} \mathbf{x}_2^1 - \mathbf{x}_1^1 & \mathbf{x}_2^2 - \mathbf{x}_1^2 & \dots & \mathbf{x}_2^N - \mathbf{x}_1^N \\ \mathbf{x}_3^1 - \mathbf{x}_1^1 & \mathbf{x}_3^2 - \mathbf{x}_1^2 & \dots & \mathbf{x}_3^N - \mathbf{x}_1^N \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_{B+i-1}^1 - \mathbf{x}_1^1 & \mathbf{x}_{B+i-1}^2 - \mathbf{x}_1^2 & \dots & \mathbf{x}_{B+i-1}^N - \mathbf{x}_1^N \end{pmatrix} - \mathbf{I}_{\mathbf{s}_i} \quad (3.13)$$

$$\mathbf{s}_i := \frac{1}{(B+i-2)N} \sum_{k=2}^{B+i-1} \sum_{j=1}^N (\mathbf{x}_k^j - \mathbf{x}_1^j) \quad (3.14)$$

Where the $-\mathbf{x}_1^j$ term in each element of \mathbf{X}_i makes the landmark vector space independent of the choice of the reference frame for a physical space; the choice of \mathbf{x}_1 is relatively flexible as long as it has small training root mean estimation error (*e.g.* $< 1.6 \text{ mm}$) such midbrain pontine junction (as is chose in this work); \mathbf{s}_i is a kernel vector to regularize \mathbf{X}_i so that the landmark vector space is hopefully independent of the choice of training datasets and thus making the landmark vector space applicable to different test datasets; $\mathbf{I}_{\mathbf{s}_i}$ is a $(B+i-2)$ -by- N block matrix having all elements be \mathbf{s}_i so that \mathbf{X}_i has zero mean. The \mathbf{s}_i 's are stored into the statistical shape model file for the detector to retrieve and use later.

The algorithm then computes the T principal component matrices $\{\mathbf{W}_i | i = 1, 2, \dots, T\}$ of the landmark vector space from T $(B+i-2)K$ -by- N matrices \mathbf{X}_i 's by

$$\mathbf{W}_i := \text{eig}(\mathbf{X}_i \mathbf{X}_i') \quad (3.15)$$

$$\mathbf{W}_i^{r_i} := [\mathbf{w}_i^1, \mathbf{w}_i^2, \dots, \mathbf{w}_i^{r_i}] \quad (3.16)$$

The operator $\text{eig}(\mathbf{X})$ returns the eigenvectors of a matrix \mathbf{X} that can be found by a singular value decomposition (SVD) process; the eigenvalues of $\mathbf{X}_i \mathbf{X}_i'$ is used to determine the number of the desired principal components r_i in i th iteration so that the sum of the eigenvalues for the employed principal components is comparable to for example, 99% of the sum of all the eigenvalues of $\mathbf{X}_i \mathbf{X}_i'$. If

the landmark vector space of zero mean is not too large to compute efficiently as it is the case in the proposed work (a 40 landmarks-by-20 datasets space), r_i should be set to the rank of the landmark space to minimize the transform error introduced by PCA. That is

$$r_i = \text{rank}(\mathbf{X}_i) \quad (3.17)$$

$\mathbf{W}_i^{r_i}$, the first r_i ($r_i \leq \text{rank}(\mathbf{X}_i)$) principal components of \mathbf{W}_i , will be returned as one of the input parameters for **Training phase-2**. Note $\mathbf{W}_i^{r_i}$ is a $(B + i - 2)K$ -by- r_i matrix.

Training phase-2 First construct a matrix \mathbf{Y}_i (a N -by- K matrix) that contains from all training datasets only the i th EPCA landmark (that is landmark to be estimated for a test image in the proposed method in this section) as

$$\mathbf{Y}_i := \begin{pmatrix} (\mathbf{x}_{B+i}^1 - \mathbf{x}_1^1)' \\ (\mathbf{x}_{B+i}^2 - \mathbf{x}_1^2)' \\ \vdots \\ (\mathbf{x}_{B+i}^N - \mathbf{x}_1^N)' \end{pmatrix} \quad (3.18)$$

The algorithm then trains a linear relationship between each EPCA landmark \mathbf{x}_{B+i} , $i = 1, 2, \dots, T$ and the PCA mapped landmarks of $\{\mathbf{x}_k | k = 1, 2, \dots, B+i-1\}$ by the first r_i principal components $\mathbf{W}_i^{r_i}$ such that

$$\mathbf{Y}_i \approx \mathbf{X}_i' \mathbf{W}_i^{r_i} \mathbf{C}_i \quad (3.19)$$

$$\mathbf{M}_i := \mathbf{W}_i^{r_i} \hat{\mathbf{C}}_i \quad (3.20)$$

Where $(\mathbf{W}_i^{r_i})' \mathbf{X}_i$ can be interpreted as the PCA mapped landmark vector space with reduced dimensionality but preserved dynamics by the r_i principal components; \mathbf{M}_i is a $(B + i - 2)K$ -by- K matrix; \mathbf{C}_i is a r_i -by- K matrix that can be optimized in terms of least squares, *i.e.*

$$\begin{aligned} \hat{\mathbf{C}}_i &= \underset{\mathbf{C}_i}{\text{argmin}} (\mathbf{Y}_i' - \mathbf{C}_i' (\mathbf{W}_i^{r_i})' \mathbf{X}_i) (\mathbf{Y}_i' - \mathbf{C}_i' (\mathbf{W}_i^{r_i})' \mathbf{X}_i)' \\ &= [(\mathbf{W}_i^{r_i})' \mathbf{X}_i \mathbf{X}_i' \mathbf{W}_i^{r_i}]^{-1} (\mathbf{W}_i^{r_i})' \mathbf{X}_i \mathbf{Y}_i \end{aligned} \quad (3.21)$$

A usual and better way to solve the overdetermined problem is to apply Gaussian elimination without forming the inverse. The solution would be valid even if \mathbf{X}_i does not have full rank. The derivation of **Equ 3.21** can be achieved by using a lemma on the property of matrix derivative:

$$\frac{\partial \mathbf{X}'\mathbf{C}\mathbf{X}}{\partial \mathbf{X}} = \mathbf{X}'(\mathbf{C}' + \mathbf{C}) \quad (3.22)$$

Vectors $\{\mathbf{M}_i | i = 1, 2, \dots, T\}$ are stored in the statistical shape model file for the detector to retrieve and use later. The total number of parameters to be stored in the two training phases is

$$\sum_{j=1}^T (B + i - 2)K^2 + KT \quad (3.23)$$

This indicates a space complexity of $O(K^2T^2)$ scalars if $T \geq B$. (T is the number of EPCA landmarks to be estimated by LME-EPCA)

Detection phase for each new test dataset, B base landmarks are estimated first by applying the methods as described in **Section 3.2**. The T EPCA landmarks are estimated in the same order as they are trained in the two training phases. EPCA landmarks $\{\mathbf{x}_{B+i}^t | i = 1, 2, \dots, T\}$ in the test dataset are estimated iteratively using the following equation

$$\mathbf{x}_{B+i}^t = \mathbf{x}_1^t + \mathbf{M}_i' \mathbf{X}_i^t \quad (3.24)$$

Where the superscription t indicates the variable is related to the test dataset to be predicted; \mathbf{M}_i 's are obtained in **Training phase-2**; \mathbf{X}_i^t 's are $(B + i - 2)K$ -by-1 matrices that can be constructed as

$$\mathbf{X}_i^t = \begin{pmatrix} \mathbf{x}_2^t - \mathbf{x}_1^t \\ \mathbf{x}_3^t - \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_{B+i-1}^t - \mathbf{x}_1^t \end{pmatrix} - \mathbf{I}_{\mathbf{s}_i}^t \quad (3.25)$$

Where \mathbf{s}_i 's are regularizing vectors obtained in **Training phase-1** and $\mathbf{I}_{\mathbf{s}_i}^t$ is a $(B + i - 2)$ -by-1 block matrix having all elements be \mathbf{s}_i . A local search process centered at \mathbf{x}_{B+i}^t is followed to improve the estimation result of each EPCA

landmark. Depending on the trait of the landmark, *i.e.* if the landmark is salient that has a distinctive intensity distribution, then the local search result of the landmark will be used in search for the landmarks in the iterations; else if the landmark is quasi that has a high localization uncertainty in all three spatial directions, then the linear predicted result is carried in search for the remaining landmarks in the subsequent iterations (so that the bad local search result of the landmark wouldn't contaminate the entire process). Finally, the detection process iterates T times to find all the T EPCA landmarks. By observing **Equ** 3.24, the computation complexity of the proposed detection phase for EPCA landmarks can be estimated by T matrix multiplications of a 1-by- $(B+i-2)K$ matrix and a $(B+i-2)K$ -by- K matrix, a T K -by-1 vector additions, and an additional time expenses on T local search processes for the EPCA landmarks. As a result, the computation complexity is thus equivalent to $\sum_{i=1}^T K^2(B+i-2)$, *i.e.* $O(K^2T^2)$ scalar multiplications and $\sum_{i=1}^T [K^2(B+i-2) + K]$, *i.e.* $O(K^2T^2)$ scalar additions if $T \geq B$, plus an additional time consuming on T local search processes for the EPCA landmarks.

In summary, the LME-EPCA algorithm estimates EPCA landmarks in **Algorithm 1**.

3.4 Local search using statistical shape models

The location of a landmark is finalized by a local search process within a bounding box centered at the corresponding search center of the landmark. A cylindrical template of the landmark is moved through the space of that bounding box in order to compute the dot product of the template and local pixel values. The intensities of the template and the compared image region are rescaled to unity before the dot product operation to eliminate the intensity heterogeneity across MRI images. Also

Algorithm 1 Arbitrary landmarks detection using LME-EPCA

< TRAINING PHASE-1 >

Given:

- Base landmarks in training datasets $\{\mathbf{x}_i^j | i = 1, 2, \dots, B; j = 1, 2, \dots, N\}$ such as *ac, pc, MPJ, eye centers, etc.* in this work
- Landmark dimensions K ($2D$ or $3D$ usually)
- Number of base landmarks B
- Number of EPCA landmarks T
- Number of training datasets N

for $i = 1, 2, \dots, T$ **do**

Compute and save the sum \mathbf{s}_i of landmarks in all training datasets $\{\mathbf{x}_k^j | k = 1, 2, \dots, B+i-1; j = 1, 2, \dots, N\}$ using **Equ 3.14**

Construct a training landmark vector space of zero mean \mathbf{X}_i using **Equ 3.13**

Compute the number of desired principal components r_i using **Equ 3.17**

Compute and save the first r_i principal components $\mathbf{W}_i^{r_i}$ of the landmark vector space using **Equ 3.15, 3.16**

end for**Write** $\{\mathbf{s}_i | i = 1, 2, \dots, T\}$ to file**Return** $\{\mathbf{W}_i^{r_i} | i = 1, 2, \dots, T\}$

< TRAINING PHASE-2 >

Given:

- Base landmarks in training datasets $\{\mathbf{x}_i^j | i = 1, 2, \dots, B; j = 1, 2, \dots, N\}$
- EPCA landmarks in training datasets $\{\mathbf{x}_{B+i}^j | i = 1, 2, \dots, T; j = 1, 2, \dots, N\}$
- $\{\mathbf{W}_i^{r_i} | i = 1, 2, \dots, T\}$ returned from **Training phase-1**

for $i = 1, 2, \dots, T$ **do**

Construct the EPCA landmark vector space \mathbf{Y}_i using **Equ 3.18**

Compute the optimal linear model coefficients $\hat{\mathbf{C}}_i$ using **Equ 3.21**

Compute and save parameter \mathbf{M}_i using **Equ 3.20**

end for**Write** $\{\mathbf{M}_i | i = 1, 2, \dots, T\}$ to file

< DETECTION PHASE >

Given:

- Base landmarks in the test dataset $\{\mathbf{x}_i^t | i = 1, 2, \dots, B\}$
- EPCA landmark intensity templates $\{\mathbf{H}_{B+i} | i = 1, 2, \dots, T\}$
- Training parameters $\{\mathbf{s}_i | i = 1, 2, \dots, T\}$ and $\{\mathbf{M}_i | i = 1, 2, \dots, T\}$
- Test image \mathbf{G}
- Landmark dimensions K
- Number of base landmarks B
- Number of EPCA landmarks T

for $i = 1, 2, \dots, T$ **do**

Construct a test landmark vector space \mathbf{X}_i^t using **Equ 3.25**

Compute i th EPCA landmark \mathbf{x}_{B+i}^t for the test image using **Equ 3.24**

Compute and save EPCA landmarks \mathbf{x}_{B+i}^L that are improved by local search process using *LocalSearch*($\mathbf{G}, \mathbf{H}_i, \mathbf{x}_{B+i}^t$) as described in **Section 3.4**.

end for**Return** $\{\mathbf{x}_{B+i}^L | i = 1, 2, \dots, T\}$.

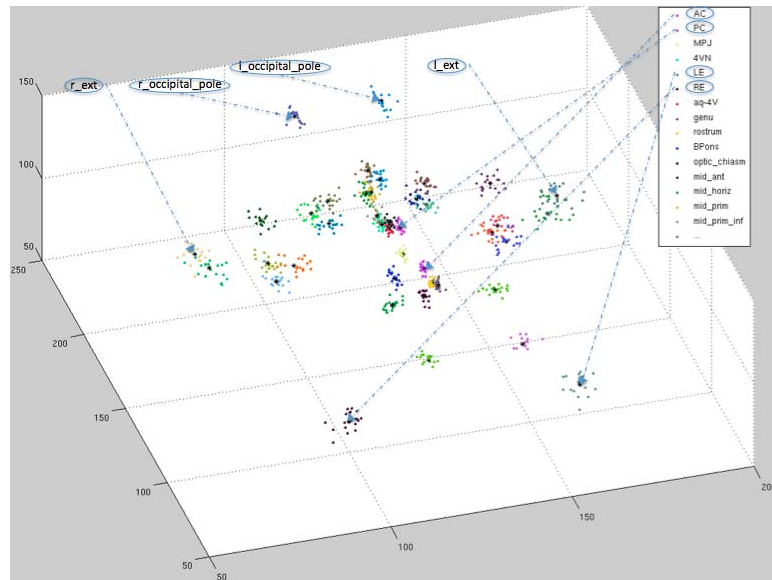
in order to reduce the anisotropic error, ten uniformly-rotated copies of different direction for each landmark template are stored in the model file. The best estimation of a landmark is given by the pixel with largest dot product between its surrounding pixel values and the template.

It is a fact in the proposed method that almost all the landmark locations are improved by the local search process, the author would like to discuss the determination of search radius for each landmark in this section. An optimal radius should be large enough to achieve high anatomical variation coverage, yet not too large to incur high false detection rate especially when there are multiple close landmarks with very similar visual appearance or intensity distribution (e.g., cerebellum landmarks). The formula to calculate the search radius is given as follows,

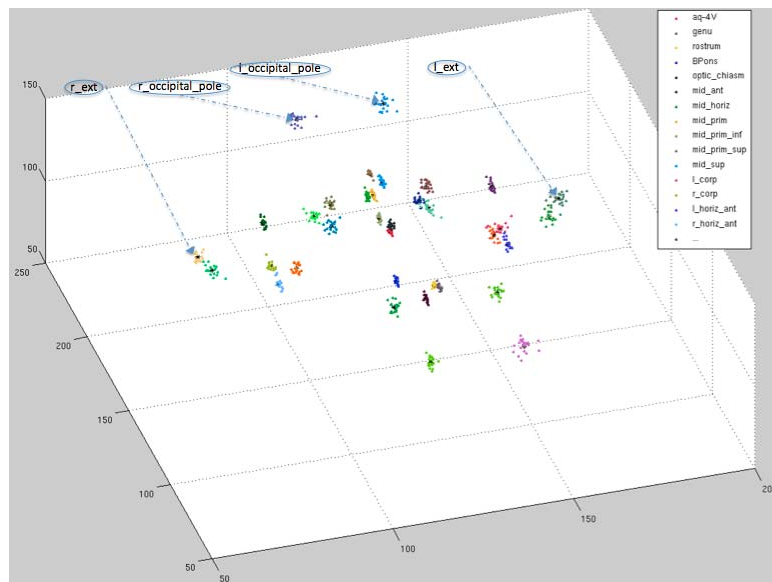
$$\begin{aligned} search_radius = \min(\max(mean_err + 3 \times standard_deviation, \\ (1 + \alpha) \times max_err), upper_bound) \end{aligned} \quad (3.26)$$

Where *mean_err*, *standard_deviation*, and *max_err* are statistics of training datasets; α is a small positive number; and *upper_bound* is a practical value reflecting the minimal distance between landmarks.

At this stage, it is obvious to see the significance of obtaining an accurate linear estimation before the local search process. More accurate linear estimation results in smaller search radius that effectively reduces the false detection rate. The constellation distribution and linear estimation error residuals for the training datasets are plotted in **Fig. 3.8a** and **Fig. 3.8b**. For each landmark, if the manually labeled one is put at the center of its mean location, its linear estimation of each subject is represented as a dot with the same color near the center in the error residual plot. A narrower dispersion of error residuals is expect so that the estimated location is closer than mean location of the landmark in training datasets using a direct search to ground-truth location of the landmark. A sagittal view of the landmark dispersion is shown in **Fig. 3.9a** and **Fig. 3.9b**. In the two plots, the sphere of each landmark

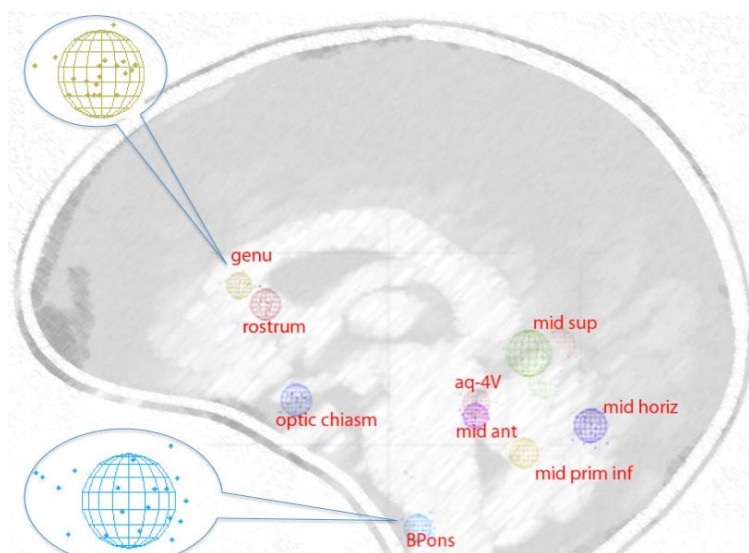


(a) Constellation distribution

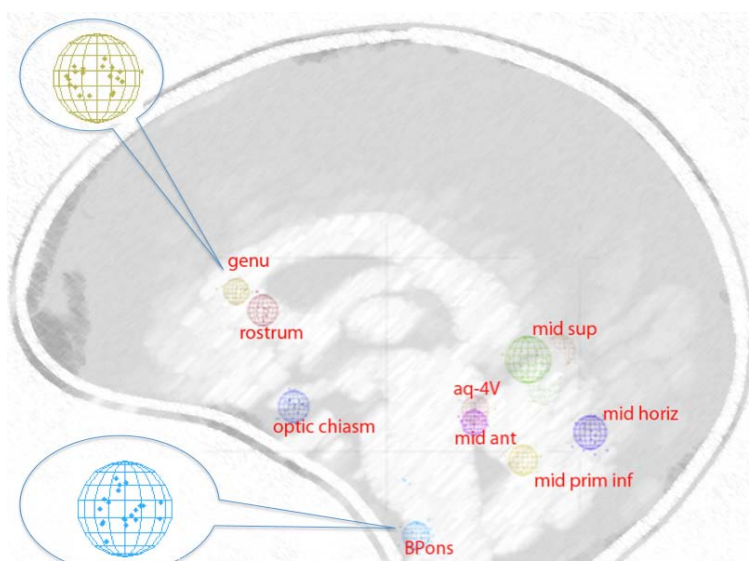


(b) Error residuals using LME-EPCA

Figure 3.8: 3D plots showing (a) constellation distribution of manually labeled landmarks in training datasets, and (b) error residuals plot by linear estimation using evolutionary PCA. In (a), samples (a dot in the figure) with the same color represent a certain landmark. In (b) for each landmark, if the manually labeled one is put at the center of its mean location, its linear estimation of each subject is represented as a dot with the same color near the center in this plot.



(a) Constellation distribution



(b) Error residuals using LME-EPCA

Figure 3.9: Sagittal views showing the dispersion of midbrain landmarks in (a) manually labeled space, and (b) linear estimation error residual space of midbrain landmarks by linear estimation using evolutionary PCA, blended with a typical brain sketch layer. The sphere of each landmark represents the search radius of the landmark that is determined by linear estimation error residuals of training datasets. Note, not all of the samples are encompassed in a related search sphere. It is expected as the aim is for 99.7% anatomical variations coverage. Also note that if two search spheres overlap with each other, it might be a challenge for the local search discriminator to find the right landmark if the two have very similar intensity templates.

represents the search radius of the landmark that is determined by error residuals in training datasets. Note in the plot showing the sagittal view of the dispersion for linear estimation error residuals of midbrain landmarks using LME-EPCA, not all of the samples are encompassed in a related search sphere. This is expected as the aim is for 99.7% anatomical variations coverage. Also note that if two search spheres overlap with each other, it might be a challenge for the local search discriminator to find the right landmark if the two have very similar intensity templates.

3.5 Resampling in-place

During the entire process, several intermediate/output resampled images are generated. The implementation of traditional resample filter in popular image processing library such as *ITK* [26] transforms, interpolates, and resamples an input image \mathbf{I} into a new voxel lattice and physical space specified by the programmer. The newly created output image updates the voxel contents with interpolated values from the input image \mathbf{I} . A side effect of this process is that it introduces interpolation error, *i.e.*, even given the inverse of the nontrivial input transform and the resampled image, it is generally impossible to obtain the original image. The problem can be even worse if several resample filters are used to generating intermediate output images in the entire process. Therefore, a new resample in-place filter is presented. It is named as *itkResampleInPlaceImageFilter* [27] that takes advantage of the fact that the voxel lattice to physical space transform is represented as a rigid transform in *ITK*. If all the transforms to be applied are rigid, then by composing the transforms before hand, the desired physical space representation can be achieved without accumulating interpolation errors. The usage of the filter can be found in **Appendix J**.

An *ITK* image class has a rigid transform representation that maps a physical

space coordinate system onto the discrete voxel lattice. The physical space representation includes the orientation matrix of the image \mathbf{R}_0 , the origin vector of the image \mathbf{T}_0 , and the spacing vector of the image \mathbf{S}_0 . Thus each *ITK* image \mathbf{I} can be represented as a composition of an image \mathbf{I}_0 with isotropic spacing, zero origin, and identity orientation, and an affine transform \mathbf{A} by

$$\mathbf{I} = \mathbf{A}\mathbf{I}_0 \quad (3.27)$$

Where

$$\begin{aligned} \mathbf{A} &= \mathbf{T}_0\mathbf{R}_0\mathbf{S}_0 \\ &= \begin{bmatrix} \mathbf{R}_0\mathbf{S}_0 & \mathbf{T}_0 \\ & 1 \end{bmatrix} \end{aligned} \quad (3.28)$$

Recall that a rigid transform matrix \mathbf{M} is composed of a translation matrix \mathbf{T} and a rotation matrix \mathbf{R} , *i.e.*,

$$\begin{aligned} \mathbf{M} &= \mathbf{TR} \\ &= \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ & 1 \end{bmatrix} \end{aligned} \quad (3.29)$$

The output image \mathbf{I}_1 can be obtained by

$$\mathbf{I}_1 = \mathbf{MI} = (\mathbf{MA})\mathbf{I}_0 \quad (3.30)$$

Therefore, the composed image info matrix can be expressed as

$$\begin{aligned} \mathbf{MA} &= \mathbf{TRT}_0\mathbf{R}_0\mathbf{S}_0 \\ &= \begin{bmatrix} (\mathbf{RR}_0)\mathbf{S}_0 & \mathbf{RT}_0 + \mathbf{T} \\ & 1 \end{bmatrix} \end{aligned} \quad (3.31)$$

By contrasting **Equ** 3.28 and 3.31, the output image origin vector \mathbf{T}_1 , the orientation matrix \mathbf{R}_1 , and the spacing vector \mathbf{S}_1 can be computed as:

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{RT}_0 + \mathbf{T} \\ \mathbf{R}_1 &= \mathbf{RR}_0 \\ \mathbf{S}_1 &= \mathbf{S}_0 \end{aligned} \quad (3.32)$$

Given an input image I_1 that is centered at the origin of an LPS coordinate system, and a rigid transform \mathbf{T} that will first translate the image 300 *mm* along

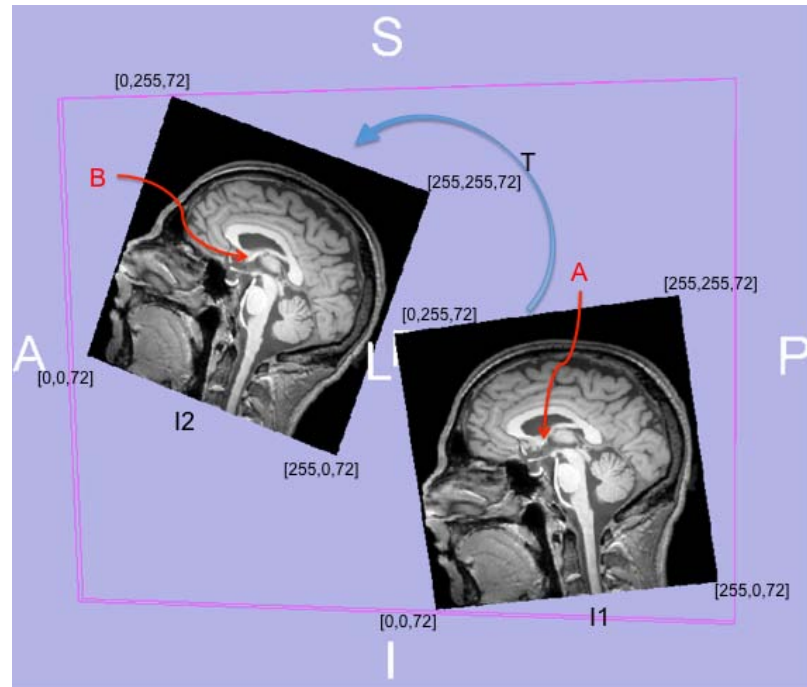


Figure 3.10: Input image I_1 and the transformed output image I_2 are shown in the same physical space but lie in their own index space. The index extent are labeled in black for each image.

Table 3.1: This table shows the ac locations in input/output and index/physical space. The output image has the same voxel contents as that of the input image but with different physical representation due to the impact of an input rigid transform.

	Index space	Physical space
Input (A)	[119, 141, 72]	(0, 29.0, 19.4)
output (B)	[119, 141, 72]	(0, 280.3, 174.8)

$-p$ -axis, and then a 0.5 rad rotation along $-l$ -axis about the origin. The mid-sagittal plane of both the input image and the output image I_2 from the *itkResampleInPlaceImageFilter* are shown in the same physical space but lie in their own index space in **Fig. 3.10**. Also, a distinctive landmark (*ac*) has been labeled for each image (*i.e.*, the point *A* in I_1 and the point *B* in I_2). Their locations in index space and physical space are listed in **Table 3.1**. As can be seen that the “Resample In-Place Filter Output Volume” has identical index location with the input image, and its physical representation correctly reflects the impact of the transform.

3.6 Landmark estimation validation criteria

The following two validation factors will be considered when evaluating the proposed method.

Accuracy is defined in this work as the comparison result of a statistical measure of automated detection errors in an instance to the average manual detection error. The average manual detection error is set to 1.6 mm root mean squared (RMS) error for brain MR images as mentioned in the work proposed by Grachev *et al.* [28] that “intrarater performance on repeated measurement showed good reliability, with an RMS error of 1.6 *mm* for both the left and right hemisphere”. If the estimation error is comparable to the manual labeling error, or even much smaller, the proposed method would be considered as accurate.

In order to validate the accuracy of the proposed detection method, 20 datasets are collected. Each dataset has one T_1 and one corresponding T_2 NIfTI [29] brain image, and one landmark list file containing dozens of manual-labeled landmarks in physical space. Landmarks can be further classified into base landmarks that are estimated by specific methods, midbrain landmarks excluding base landmarks and off-midbrain landmarks that are predicted by linear estimation and improved by local search. Base and other midbrain landmarks

are almost all salient landmarks that have distinctive intensity distributions. Off-midbrain landmarks are almost all quasi landmarks that have high localization uncertainties in all three spatial directions. A list of landmarks of interest is given in **Appendix F**.

In addition, the *acpc* line of each image has already close to the horizontal line in posterior-anterior coordinate component. The 20 datasets are denoted as “Test Suite A”. Further pick 10 datasets from “Test Suite A” for training and the rest 10 of them for testing/validation. Each test image is sent to the constellation detector (discussed in **Subsection 3.1.2**), detected landmarks in the input space are collected and ready for accuracy validation with their corresponding manual-labeled reference counterparts.

Reliability is the requirement that the algorithm should behave consistently, given a reasonable range of possible clinical input. The author will validate the reliability of the proposed method in this work over hundreds of clinical datasets, with different image modality, orientation, spacing, and origin.

To validate the reliability of the proposed method in this work, all 20 datasets in “Test Suite A” are chosen for training, and another two larger clinical datasets (named *peg_MR* and *AV_MR*) with very different image information and in a more general status (*e.g.* the *acpc* line might have a large angle with the horizontal line) as test/validation datasets compared to “Test Suite A”. The two datasets contain scans from different subjects, scans acquired from the same subject but in a different time, and scans of the same subject, same time, but in a different modality. As landmarks are generally unavailable in clinical practice, the reliability of the proposed method is only validated on several base landmarks.

CHAPTER 4 RESULT

4.1 Landmark constellation detection

Applying the accuracy validation method as described in **Section 3.6**, the corresponding test result is shown in **Table 4.1** when linear model estimation and local search are both performed. The improvement by performing a local search after a linear model estimation is also computed in **Table 4.2a** and **Table 4.2b**. Compared to 1.6 mm RMS errors for manual placement of landmarks [28] as discussed in **Section 3.6**, the test result in the tables shows that

- The estimation of base landmarks ac , pc , MPJ , $4VN$ is generally accurate.
- The estimation of other midbrain landmarks is a little bit worse but still reasonable.
- The estimation of off-midbrain quasi-landmarks has much error.
- Generally, the estimation result for T_1 images is a little bit better than that of T_2 images probably because their spacing is different. That is, large spacing usually results in a worse detection accuracy.
- The local search improves the linear prediction result for landmarks with consistent intensity distribution such as the midbrain landmarks.

Applying the reliability validation method as described in **Section 3.6**, and compared to the 1.6 mm root mean squared average error for manual labeling accuracy on brain images as mentioned in the work proposed by Grachev *et al.* [28], **Table 4.3a** and **Table 4.3b** tell that ac detection is generally very accurate and reliable; the estimation for pc and $4VN$ is reasonable but a little bit worse. In order to investigate the cause of larger errors in the two landmarks, the author visually inspect lots of the

Table 4.1: Detection errors for several base, midbrain, and off-midbrain landmarks in 10 T_1 and T_2 human brain datasets. RMS = root mean squared (error), Std = standard deviation.

		T_1 Images		T_2 Images	
Landmark Name		RMS	Std	RMS	Std
Base	<i>ac</i>	0.97	0.14	1.11	0.54
	<i>pc</i>	1.05	0.76	1.51	0.80
	<i>MPJ</i>	0.78	0.47	0.85	0.33
	<i>4VN</i>	0.96	0.33	1.34	0.48
Midbrain	<i>aq-4V</i>	1.20	0.43	1.16	0.36
	<i>genu</i>	2.64	2.36	3.18	2.20
	<i>rostrum</i>	2.47	1.57	2.25	1.49
	<i>BPons</i>	1.89	1.20	2.02	0.89
	<i>optic_chiasm</i>	2.72	2.15	2.79	1.61
Off-midbrain	<i>l_ventricular_head</i>	4.10	3.99	4.38	3.04
	<i>r_ventricular_head</i>	2.80	1.68	3.69	2.03
	<i>l_corp</i>	5.73	5.39	6.65	4.09
	<i>r_corp</i>	3.68	3.55	4.96	3.56
	<i>l_horiz_ant</i>	6.83	5.60	8.80	4.82
	<i>r_horiz_ant</i>	7.12	4.95	7.83	4.82
	<i>l_sup</i>	8.06	3.77	8.14	4.66
	<i>r_sup</i>	6.71	4.40	7.32	4.59

Table 4.2: Midbrain and off-midbrain landmark detection errors improvement by local search. RMS = root mean squared (error), Std = standard deviation.

(a) Improvement made by local search (in %) for several midbrain landmarks in T_1 images

Name	RMS	Std
<i>aq-4V</i>	38	57
<i>genu</i>	30	4
<i>rostrum</i>	36	23
<i>BPons</i>	42	17
<i>optic_chiasm</i>	9	-48

(b) Improvement made by local search (in %) for several off-midbrain landmarks in T_1 images

Name	RMS	Std
<i>l_ventricular_head</i>	27	-16
<i>r_ventricular_head</i>	33	8
<i>l_corp</i>	19	-74
<i>r_corp</i>	35	-16
<i>l_horiz_ant</i>	21	-51
<i>r_horiz_ant</i>	6	-30
<i>l_sup</i>	5	-7
<i>r_sup</i>	1	-2

test datasets. The major reason turns out to be the inconsistency in manual-labeled landmarks. In contrast, the automated detection usually behaves very consistent. It is one of the major goals in writing this thesis to overcome the disadvantage of the inconsistency in manual labeling process. A close snapshot of a test dataset with large estimation errors in *pc* is given in **Fig. 4.1**.

Note the automated test failed 2 of the datasets due to bad scans without eye response as shown in **Fig. 4.2**. As the proposed method relies on a reasonable estimation of eye centers, a GUI corrector (discussed in **Subsection 3.1**) is developed to make it possible for the user to manually correct the eye center locations for the detector in case the eye detector fails.

Table 4.3: Mean errors in LPS space (in mm) for a few base landmarks. RMS = root mean squared (error)

(a) T_1 peg_MR datasets (215 subjects)

Name	L	P	S	RMS
<i>ac</i>	0.48	0.53	0.45	0.83
<i>pc</i>	0.42	1.47	0.62	1.65
$4VN$	0.61	0.73	0.93	1.33

(b) T_1 AV_MR datasets (98 subjects)

Name	L	P	S	RMS
<i>ac</i>	0.46	0.73	0.50	1.00
<i>pc</i>	0.34	1.02	1.14	1.57
$4VN$	0.50	0.89	0.95	1.39

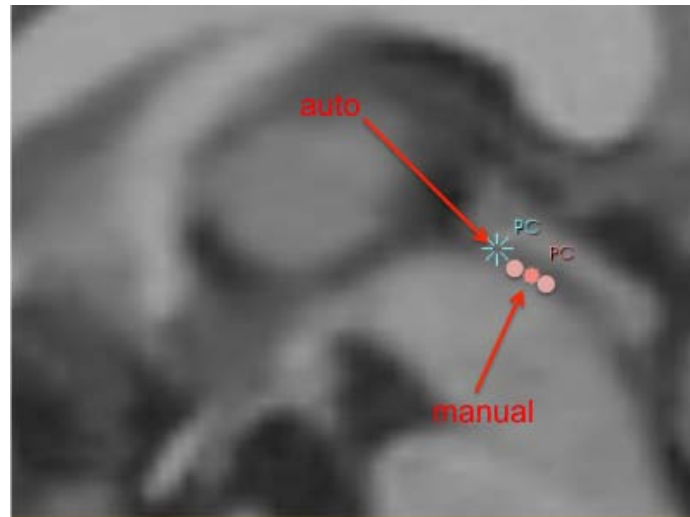


Figure 4.1: Different *pc* definitions in sagittal view: (a) some of the possible locations of manual labeling landmarks (in red) and (b) typical location of automated detection (in blue). The automated estimation appears consistently at the superior and anterior entrance of the cerebral aqueduct as is defined in the model file. However, the manual-labeled landmark can appear at any upper-half location along the cerebral aqueduct. Almost all the estimated *pcs* with distance error $> 2 mm$ (composing 35% of the test result) are due to this factor.

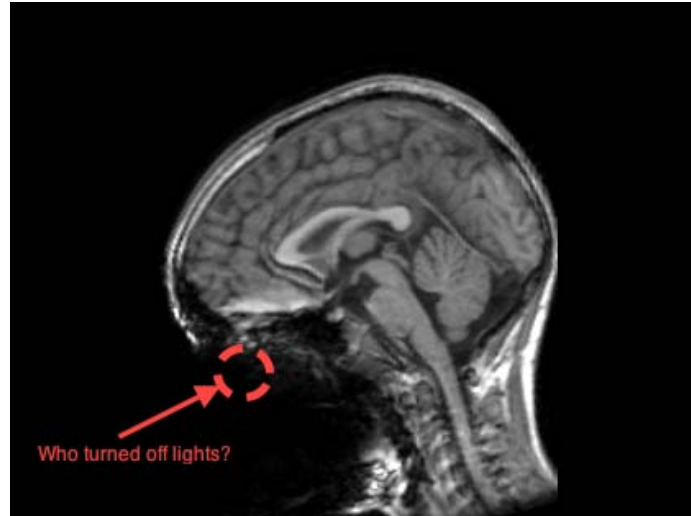


Figure 4.2: One of the failing cases in the automated test due to bad scans without eye response as the magnetic field is interfered by metals such as metallic false teeth

4.2 Atlas construction

As mentioned in **Section 2.5**, atlas is often built by first warping a set of moving images to a fixed image using some kind of transform like the thin-plate spline transform (TPS). The atlas is then obtained by averaging the warped images (and the fixed image) together. A demonstration of applying thin-plate spline transform to the brain image is shown in **Fig. 4.3**. All landmarks presented are from the automated detection result of the fixed image in the middle inset of **Fig. 4.3a**. All images presented in this figure are in the same physical space. The moving image in the left inset of **Fig. 4.3a** is transformed to a space such that the landmarks are aligned in both spaces in fixed image and moving image, and by TPS, the transform minimizes the bending energy of the transformed space, and the space is continuously differentiable in any place. The deformed image is shown in the right inset of the same figure. The corresponding deformation field is shown in **Fig. 4.3b**.

With a similar idea, one brain atlas was built from the averaging of 206 TPS-warped clinical images using the information of 41 automatically detected landmarks by the proposed constellation detector. Another reference atlas was built from a

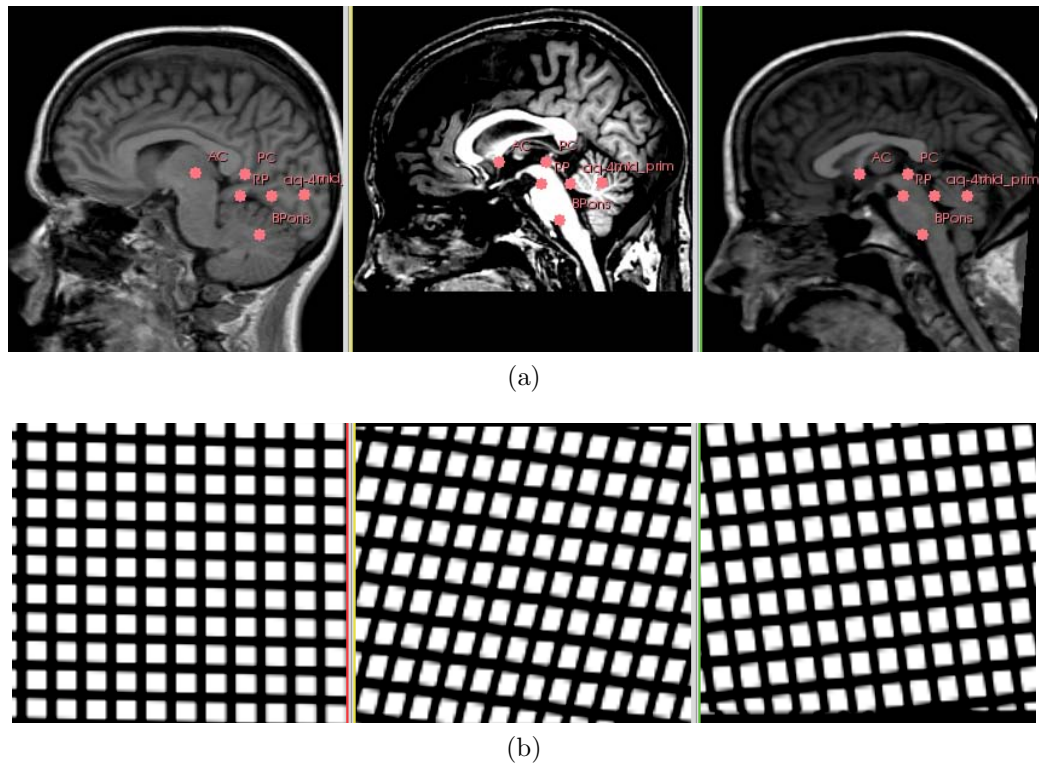


Figure 4.3: An example result of applying thin-plate spline transform to brain images: (a) A TPS demonstration: moving image in the left, fixed image in the middle, and the deformed image in the right inset; (b) the deformation field of the transform in axial view in the left, sagittal view in the middle, and coronal view in the right inset. The deformation field grid images are generated by DTI-TK [30].

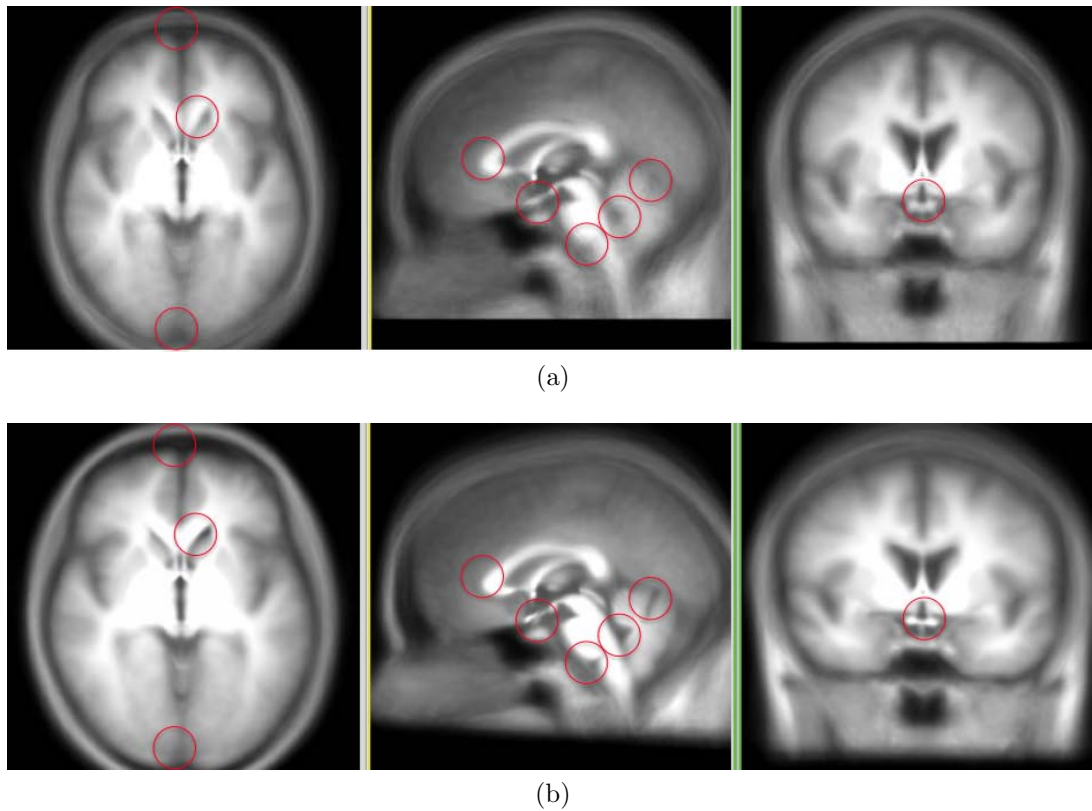


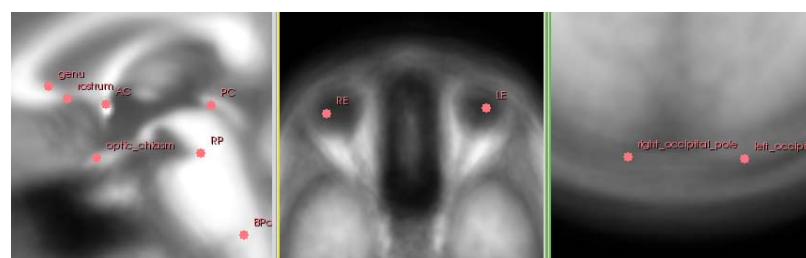
Figure 4.4: Central slices of brain atlas built from (a) direct averaging of 206 rigidly *acpc*-aligned images, and (b) averaging of 206 TPS-warped images using 41 automatically detected landmarks. Some regions with better quality of (b) over (a) are highlighted in red circles. The detailed inspection can be found in **Fig. 4.5**.



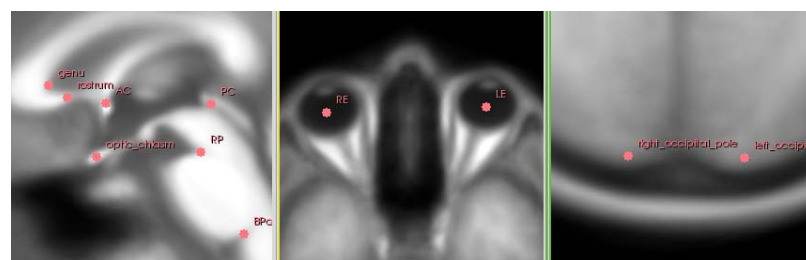
(a)



(b)



(c)



(d)

Figure 4.5: Selected slices demonstrating the improvement of using automated landmarks: (a) *acpc* version of and (b) TPS version of sagittal views of cerebellum in the left, coronal views of cerebellum in the middle, and sagittal views of ventricular head in the right inset; (c) *acpc* version of (d) TPS version of sagittal views of the area near *ac* and *pc* in the left, axial views of eyes in the middle, and axial views of occipital poles in the right inset.

direct averaging of the same 206 raw images with a rigid-transform that aligning 3 automatically detected landmarks (*i.e.* the *ac*, *pc*, and *MPJ*) so that the images are *acpc*-aligned. The center slices of the constructed brain atlases are shown in **Fig. 4.4**. The TPS-warped atlas by using 41 landmarks has a great advantage over the atlas using a rigid transform that aligning 3 landmarks in that it generally has much richer information near a defined landmark than the reference atlas while maintains at least about the same quality in other regions. More detailed inspection on the two atlases are shown in **Fig. 4.5**. For example, in the left inset of **Fig. 4.5b**, the atlas has richer information at the 4th ventricle notch, the junction of cerebral aqueduct and the fourth ventricle, and the primary fissure of the cerebellum; a clearer 4th ventricle notch and a coronal cross-sectional view of the cerebellum is given in the middle inset of **Fig. 4.5b**; the ventricular head is more clear in the right inset of the same figure than its counterpart, which reflects the benefit of consistent selection of the ventricular head landmark as the most anterior cerebrospinal fluid (CSF) in left/right ventricular nucleus at the modeling phase. Similarly in the left inset of **Fig. 4.5d**, image regions near the optic chiasm, basal pons, genu, and rostrum, *etc.* are much finer than their counterparts in **Fig. 4.5c**. Note the image regions near *ac*, *pc*, and *MPJ* in both the TPS-warped atlas and the *acpc*-aligned one have about the same quality, which is expected, as all of the three landmarks are used in both construction process. Image regions near eyes are more clearer in the TPS-warped atlas as the information of eye centers are using in the the construction process of the atlas, which can be seen from the middle insets of **Fig. 4.5c** and **Fig. 4.5d**. The cornea of eyes are crisp in the TPS warped atlas that uses more landmarks information. Finally, in the right inset of **Fig. 4.5d**, the boundary of skull and brain is clearly visible thanks to the information provided by the occipital poles. In contrast, it is very blurred in the *acpc*-aligned atlas.

CHAPTER 5 CONCLUSION

5.1 Contributions

This work explored one way of estimating a landmark constellation automatically, consistently, and efficiently. The proposed method demonstrated a successful application on how to effectively utilize image processing in tackling clinical challenges. It is shown that the cooperation of spatial localization using linear model prediction with evolutionary principal components and local search estimation using statistical shape models is capable of effectively extracting important landmark detection information from both morphometric relationships of landmarks and consistent intensity distribution of images. This reinforces the idea proposed by Johnson and Christensen in [31] and sheds light on the possibility of allying landmark information and image intensity information to improve landmark detection result: a reasonable linear prediction derived from landmark information provides a start point of the local search process and keeps it on a right track; on the other hand, a consistent local search estimation can generate new high quality landmarks that in return sends positive feedback to the linear estimator. Without using landmark information, the intensity-based local search would be blind and vulnerable to false detection; without using intensity information, the landmark-based linear prediction would never achieve a higher quality. The proposed method is also efficient in that it is usually ten times faster than human expert. It is accurate (compared to 1.6 *mm* root mean squared errors of manual labeling of brain landmarks), consistent, reliable in predicting many salient midbrain point landmarks such as *ac*, *pc*, *MPJ*, *etc.* in a longitudinal, multisubject environment, and throughout large datasets with different modalities and image information such as orientation, spacing, and origin. The framework of linear model estimation method using evolutionary principal components and the idea of

local search using statistical shape models are generalized to the detection task for arbitrary number of landmarks in other organs, creatures, or even any other physical objects in the world as long as the landmarks present intensity consistency and satisfy regularity in spatial organization. It is not sufficient to give a high quality estimation for quasi-landmarks that have high localization uncertainties in all three spatial directions and do not have consistent intensity distributions. Landmarks having neighbors with very similar intensity distributions are generally very hard to discern by global intensity matching-based search process.

The proposed method in this work takes the advantages of work from both principal components [8] and statistical shape models [9], and integrates them in an evolutionary way in the application of landmark constellation detection so that they can benefit from each other and team up for a better detection result. Principal components make the method scalable so that it is possible to deal with arbitrary number of landmarks by representing them in a most efficient way. On the other hand, the statistical shape models encode the intensity information and morphometric relationships in a way so that the computer is able to make informed decision on where and how to find landmarks. The proposed method in this work is very similar to the work described by Babak *et al.* in [5] in the search means for the *MPJ*. However, this method works in nature physical space, and is much robust to rotation by exploiting the extra information from eye centers. The proposed method is capable of detecting various other landmarks by morphometric constraining, linear model prediction, and local search estimation. The proposed method also adapted a Hough transform radial voting image filter proposed by Mosaliganti *et al.* in [18] to the task of eye detection. In addition, the proposed method in this work extend the use of the original image filter so that it can deal with images with different modalities. Another improvement is that the proposed eye detection method provides a result verification and exception handling mechanism through adult interpupillary distance checking and an interactive

GUI corrector.

The implementation of the proposed method is a registered, open-source, and cross-platform landmark constellation detection software that is suitable for large-scale, multisite imaging study [20]. It is fully automated yet provides user a convenient way to interact with the software by a GUI corrector and lots of arguments for parameter setting. The software advocates the use of in-place resampling image filter for general medical image processing task to effectively reduce unnecessary interpolation errors emanated from traditional resample image filters. The software also provides great interoperability with popular image processing software such as *3DSlicer* [32]. Information as the aligned NIfTI image, aligning transform, output landmark list *fcsv* file in either input space or *acpc*-aligned space, and the output *MRML* scene file, *etc.* can be transmitted between the proposed software and *3DSlicer* without any difficulty.

5.2 Future work

- Add confidence factor into account for each landmark in the kernel spline transform for atlas construction.
- Construct atlases from much larger datasets (1000 subjects).
- Combine the information of images with different modalities to improve the detection result as a landmark might be more salient in certain modality than in others.
- Perform validation of (a) landmark rotation-robustness and (b) registration result using *3D* Procrustes analysis

APPENDIX A ABBREVIATIONS AND NOTATIONS

A.1 Conventions

x scalar variable

X scalar constant

\mathbf{x} vector variable

$\mathbf{x}.\mathcal{L}$ left-right component of $3D$ point landmark location \mathbf{x}

$\mathbf{x}.\mathcal{P}$ posterior-anterior component of $3D$ point landmark location \mathbf{x}

$\mathbf{x}.\mathcal{S}$ superior-inferior component of $3D$ point landmark location \mathbf{x}

\mathbf{X} matrix variable

\mathbf{X}' transpose matrix of \mathbf{X}

\mathbf{X}^{-1} inverse matrix of \mathbf{X}

$f()$ function returning scalar

$\mathbf{f}()$ function returning vector

$\mathbf{F}()$ function returning matrix

A.2 Terms and abbreviations

LPS left-to-right, posterior-to-anterior, and superior-to-inferior physical coordinate system commonly used in the application of medical imaging

PCA principal component analysis

PRHT probabilistic radial Hough transform

TPS thin plate spline

PDF probability distribution function

CDF cumulative distribution function

LME-EPCA linear model estimation using evolutionary PCA

EPCA landmark landmark estimated by LME-EPCA algorithm

MSP anatomical mid-sagittal plane that is a vertical plane and contains anterior commissure and posterior commissure

EMSP estimated mid-sagittal plane that maximizes reflective correlation of the brain image in left-right direction

***acpc*-aligned space** a physical space of a brain image with aligned *ac*, *pc*, and *MPJ*

A.3 Landmark definitions

ac centroid of anterior commissure

pc superior aspect of posterior commissure

MPJ midbrain pontine junction

4VN fourth ventricle notch

CM centroid of head mass

LE centroid of the left eye

RE centroid of the right eye

CEC center of left and right eye centroids

aq-4V the junction of cerebral aqueduct and the fourth ventricle

BPons basal pons

genu genu in anterior corpus callosum

rostrum the pointed tip of rostrum of the corpus

optic chiasm centroid of optic chiasm

left/right ventricular head the most anterior cerebrospinal fluid (CSF) in left-
/right ventricular nucleus

A.4 Notations

r_t radius of the target circle/sphere to be detected

r_v radius of the voting circle/sphere

α ratio relating to r_i and r_p

\mathbf{p} arbitrary pixel (location)

\mathbf{p}_0 pixel on the rim/surface of circle/sphere

\mathbf{u}_0 center (mean) of $f_{HT}()$

σ variance (of Gaussian distribution)

$f_{HT}(\mathbf{p}(\rho), \beta, \sigma)$ probabilistic radial Hough transform probability distribution function
(Gaussian) of any location \mathbf{p} with variance σ and contributed by \mathbf{p}_0

$F_{HT}(\mathbf{p}(\rho), \beta, \sigma)$ probabilistic radial Hough transform cumulative distribution function

β auxiliary ratio used in the expression of $F_{HT}()$

θ angle of \mathbf{p}_0 in polar coordinate system

ρ radius of \mathbf{p} in polar coordinate system

ϕ angle of \mathbf{p} in polar coordinate system

$\gamma(\rho, \beta, \sigma)$ exponential term of $F_{HT}()$

$I_0(x)$ modified Bessel function of first kind

K number of dimensions of the landmarks

B number of base landmarks estimated by methods other than LME-EPCA algorithm

T number of landmarks estimated by LME-EPCA algorithm

N number of training datasets

r_i number of desired principal components in i th iteration of LME-EPCA algorithm

x^j j th observation of variable x in the discussion of LME-EPCA algorithm

\mathbf{c}_i linear model coefficient vector in terms of least squares with respect to i th iteration in LME-EPCA algorithm

\mathbf{W} principal component matrix

\mathbf{w}_i i th largest principal component

APPENDIX B

NOTE ON PROBABILISTIC RADIAL HOUGH TRANSFORM

This note will show a sufficient condition for circle detection using probabilistic radial Hough transform (PRHT). An analog can be made for sphere detection. A notational reference for this appendix can be found in **Appendix C**.

Assertion: To sufficiently detect the circle using PRHT, the ratio between the voting circle radius r_v and the target circle radius r_t should satisfy

$$\alpha := \frac{|r_v - r_t|}{r_t} \leq 0.1 \quad (\text{B.1})$$

Proof: By the property of PRHT, each pixel location \mathbf{p}_0 on the rim of the circle will contribute a Gaussian probability distribution function (PDF) along its gradient vector \vec{g} of magnitude. Let \mathbf{u}_0 be $\mathbf{p}_0 + \vec{g}(\mathbf{p}_0)$, the PDF f_{HT} of any location \mathbf{p} contributed by a point on the rim of target input circle \mathbf{p}_0 can be expressed as

$$f_{HT}(\mathbf{p}, \sigma, \mathbf{u}_0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{p} - \mathbf{u}_0)'(\mathbf{p} - \mathbf{u}_0)}{2\sigma^2}\right) \quad (\text{B.2})$$

$$r_v = 3\sigma \quad (\text{B.3})$$

As shown in **Section 2.7**, the locus of the centers of the PDFs is a smaller circle co-centered with the target circle and with a radius of $|r_v - r_t|$. An illustration showing the geometry of PRHT is given in **Fig. B.1**.

Actually, the voting circle radius r_v can be either greater or less than the target circle radius. For simplicity, the following analysis is based on the condition that $r_v < r_t$. A similar analysis can be performed when $r_v > r_t$. To simplify the expression of future equations and with **Equ. B.3** and **B.1**, β is defined as follows

$$\beta := \frac{\alpha r_t}{\sigma} = \frac{3\alpha}{1 - \alpha} \quad (\text{B.4})$$

Without loss of generality, if the center of the target circle is set to the origin of the parameter space, then in a polar coordinate system, for each location $\mathbf{p}(\rho, \phi)$, each pixel on the rim of the target circle contributes a PDF that centered at $\mathbf{p}_0(|r_v - r_t|, \theta)$

can be expressed as

$$f_{HT}(\mathbf{p}(\rho, \phi), \sigma, \mathbf{p}_0(|r_v - r_t|, \theta)) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\rho \cos \phi - |r_v - r_t| \cos \theta)^2 + (\rho \sin \phi - |r_v - r_t| \sin \theta)^2}{2\sigma^2}\right) \quad (\text{B.5})$$

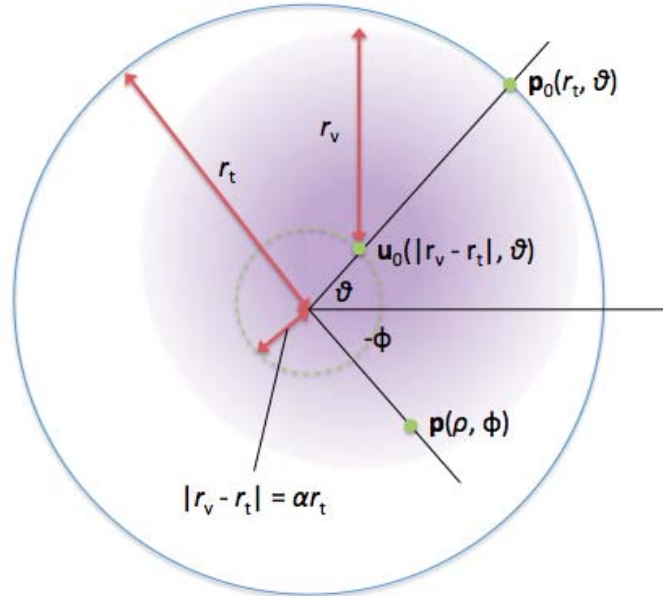


Figure B.1: The geometry of probabilistic radial Hough transform. The aim is to find the center of the target circle with radius r_t in input space (*i.e.* the big blue circle). Each pixel \mathbf{p}_0 on the rim of the target circle contributes a Gaussian probability distribution function f_{HT} centered at \mathbf{u}_0 that has the same direction as gradient vector of \mathbf{p}_0 and a magnitude of voting radius r_v . The locus of the center of the PDFs is a smaller circle (in green and dashed) and is co-centered with the target circle and with a radius of $|r_v - r_t|$. The note in this appendix will try to analysis the detection correctness when r_v is different from r_t , by first finding the expression of the cumulative distribution function F_{HT} for any location $\mathbf{p}(\rho, \phi)$.

By using the definition of α in **Equ.** B.1, β in **Equ.** B.4, and f_{HT} in **Equ.** B.5, the cumulative distribution function (CDF) of any location \mathbf{p} can be expressed as follows,

$$F_{HT}(\mathbf{p}, \beta, \sigma) = \int_0^{2\pi} f_{HT}(\mathbf{p}(\rho, \phi), \sigma, \mathbf{p}_0(|r_v - r_t|, \theta)) |r_v - r_t| d\theta = \gamma(\rho, \beta, \sigma) \int_0^{2\pi} \exp\left(\frac{\rho\beta}{\sigma} (\cos \phi \cos \theta + \sin \phi \sin \theta)\right) d\theta \quad (\text{B.6})$$

$$\gamma(\rho, \beta, \sigma) := \frac{\beta}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\rho^2}{\sigma^2} + \beta^2\right)\right) \quad (\text{B.7})$$

The expression of F_{HT} can be further simplified by using an integral formula as mentioned in [33],

$$\int_0^{2\pi} e^{x \cos \theta + y \sin \theta} d\theta = 2\pi I_0(\sqrt{x^2 + y^2}) \quad (\text{B.8})$$

Where I_0 is the modified Bessel function of first kind.

Finally, the cumulative distribution function F_{HT} of PRHT for any location $\mathbf{p}(\rho, \phi)$ can be expressed as

$$F_{HT}(\mathbf{p}(\rho), \beta, \sigma) = \gamma(\rho, \beta, \sigma) \times 2\pi I_0\left(\frac{\rho\beta}{\sigma}\right) \quad (\text{B.9})$$

To prove the assertion, considering the smoothness of F_{HT} and the fact that the CDF value outside the PDF center locus circle (as shown as a dashed green circle in **Fig. B.1**) is less than the value inside the locus circle, it is equivalent to show that

“ $\alpha < 0.1$ is a sufficient condition for $F|_{\rho=0}$ to be the maximum of F_{HT} , $\rho \in [0, \alpha r_t]$ ”.

In this range, $\frac{\rho\beta}{\sigma} \rightarrow 0$ if $\alpha < 0.1$. $I_0(x)$ is monotonic and $I_0(x) \rightarrow 1$ if $x \rightarrow 0$. For example, if $\rho = 0$, then $I_0(0) = 1$; if $\rho = \alpha r_t$, $\alpha = 0.1$, $I_0\left(\frac{\rho\beta}{\sigma}\right) = I_0(\beta^2) = I_0\left(\frac{1}{9}\right) \approx 1.0030888$. Observing B.9, as γ is always monotonically decreasing with respect to ρ , F_{HT} will have a maximum at $\rho = 0$. On the contrary, as α grows, $I_0\left(\frac{\rho\beta}{\sigma}\right)|_{\rho=\alpha r_t}$ becomes much larger than $I_0(0)$. For example, if $\alpha = 0.5$,

$$\frac{F_{HT}(\rho, \beta, \sigma)|_{\rho=0, \beta=3}}{F_{HT}(\rho, \beta, \sigma)|_{\rho=1, \beta=3}} = \frac{1}{\exp(-\frac{9}{2}) \times I_0(9)} \approx \frac{1}{12.15} \quad // \quad (\text{B.10})$$

In summary, the cumulative distribution function $F_{HT}(\rho, \beta, \sigma)$ of probabilistic circular Hough transform is a product of a exponential term $\gamma(\rho, \beta, \sigma)$ and a modified Bessel function of first kind term. When α is small enough (*e.g.* $\alpha \leq 0.1$), F_{HT} is dominated by the exponential term so that $F_{HT}()$ always has the largest value at the center of the target circle; if α is large (*e.g.* $\alpha = 0.5$), $F_{HT}()$ is dominated by the Bessel term so that the target circle center may not be at the location associated with the largest cumulative value. To detect circle/sphere using probabilistic radial Hough transform, ensure that the ratio of the radius r_t of target object in input space and the tentative radius r_v of the object in parameter space is close to one. ■

APPENDIX C
NOTATIONAL REFERENCE FOR APPENDIX B

Notation	Definition	Relationship
r_t	Radius of the target circle/sphere to be detected	/
r_v	Radius of the voting circle/sphere	$r_v < r_t$ for most of the discussion
α	Ratio relating to r_t and r_v	$\alpha = r_v - r_t / r_t$
\mathbf{p}	Arbitrary pixel (location)	/
\mathbf{p}_0	Pixel on the rim/surface of circle/sphere	/
\mathbf{u}_0	Center (mean) of $f_{\text{HT}}()$	Equ. B.2
σ	Variance of Gaussian distribution	Related to $f_{\text{HT}}()$
$f_{\text{HT}}(\mathbf{p}(\rho), \beta, \sigma)$	Probabilistic radial Hough transform probability distribution function (Gaussian) of any location \mathbf{p} with variance σ and contributed by \mathbf{p}_0	Equ. B.5
$F_{\text{HT}}(\mathbf{p}(\rho), \beta, \sigma)$	Probabilistic radial Hough transform cumulative distribution function	Equ. B.6, B.9
$\exp(x)$	Exponential function of scalar x	Equivalent to e^x
β	Auxiliary ratio	Equ. B.4
θ	Angle of \mathbf{p}_0 in polar coordinate system	/
ρ	Radius of \mathbf{p} in polar coordinate system	/
ϕ	Angle of \mathbf{p} in polar coordinate system	/
$\gamma(\rho, \beta, \sigma)$	Exponential term of $F_{\text{HT}}()$	Equ. B.7
$I_0(x)$	Modified Bessel function of first kind	Component of $F_{\text{HT}}()$

APPENDIX D
NOTATIONAL REFERENCE FOR LME-EPCA ALGORITHM
(SECTION 3.3)

Notation	Definition	Dimension	Relationship
K	Landmark dimensions	1-by-1	Scalar constant
N	Number of training datasets	1-by-1	Scalar constant
B	Number of base landmarks	1-by-1	Scalar constant
T	Number of EPCA landmarks	1-by-1	Scalar constant
i	Iteration index/EPCA landmark index	1-by-1	$1 \leq i \leq T$
j	Training dataset index	1-by-1	$1 \leq j \leq N$
k	Landmark index	1-by-1	$1 \leq k \leq B + T$
\mathbf{x}_k^j	Landmarks in j th training datasets. Base landmarks: $\{\mathbf{x}_k \mid k = 1, 2, \dots, B\}$ EPCA landmarks: $\{\mathbf{x}_k \mid k = B+1, B+2, \dots, B+T\}$	K -by-1	/
\mathbf{X}_i	Matrix containing all landmarks processing before the current landmark in training datasets in i th iteration	$(B + i - 2)K$ -by- N	Composed of $\{\mathbf{x}_k^j \mid j = 1, 2, \dots, N; k = 1, 2, \dots, i - 1\}$
\mathbf{s}_i	Vector demeaning \mathbf{X}_i in i th iteration; output training parameters in Training phase-1	K -by-1	$\sum_{j=1}^N \sum_{k=2}^{B+i-1} (\mathbf{x}_k^j - \mathbf{x}_1^j)$
\mathbf{I}_{s_i}	$(B + i - 2)$ -by- N block matrix having all elements of \mathbf{s}_i	$(B + i - 2)K$ -by- N	/
\mathbf{W}_i	principal component matrix of \mathbf{X}_i in i th iteration	$(B + i - 2)K$ -by- $(B + i - 2)K$	$eigenvectors(\mathbf{X}_i \mathbf{X}_i')$
\mathbf{w}_i^j	The j th principal in i th iteration	$(B + i - 2)K$ -by-1	/
r_i	Number of desired principal components in i th iteration	1-by-1	$r_i = \text{rank}(\mathbf{X}_i)$
$\mathbf{W}_i^{r_i}$	First r_i principal components in	$(B + i - 2)K$ -by- r_i	$[\mathbf{w}_i^1, \mathbf{w}_i^2, \dots, \mathbf{w}_i^{r_i}]$

	<i>i</i> th iteration		
\mathbf{C}_i	Linear model parameter in <i>i</i> th iteration	r_i -by- K	/
\mathbf{Y}_i	A matrix having current EPCA landmark in all training datasets in <i>i</i> th iteration	N -by- K	$\mathbf{X}_i' \mathbf{W}_i^{r_i} \mathbf{c}_i$
$\hat{\mathbf{C}}_i$	Optimal linear model parameter in least squares term	r_i -by- K	About $\mathbf{X}_i, \mathbf{W}_i^{r_i}, \mathbf{Y}_i$
\mathbf{M}_i	Output training parameter in Training phase-2	$(B + i - 2)K$ -by- K	$\mathbf{W}_i^{r_i} \hat{\mathbf{C}}_i$
\mathbf{X}_i^t	A matrix containing all known landmarks in test dataset in <i>i</i> th iteration	$(B + i - 2)K$ -by-1	Composed of $\{\mathbf{x}_k^t \mid k = 1, 2, \dots, i - 1\}$
\mathbf{x}_k^t	<i>k</i> th landmark in test datasets Base landmarks: $\{\mathbf{x}_k^t \mid k = 1, 2, \dots, B\}$ EPCA landmarks: $\{\mathbf{x}_k^t \mid k = B+1, B+2, \dots, B+T\}$	K -by-1	$\mathbf{x}_k^t + (\mathbf{X}_k^t)' \mathbf{m}_i$

APPENDIX E

NOTE ON PRINCIPAL COMPONENT ANALYSIS

A brief introduction of PCA can be found in **Subsection 2.2**.

Inspired by Alpaydin's proof in his book [34], a proof is rederived for the first principal component, and a proof for the rest of the principal components is also provided by mathematical induction.

First to show that data \mathbf{X} projected on the largest principal component \mathbf{w}_1 has largest variance.

By definition:

$$\mathbf{w}_1 = \operatorname{argmax}_{\mathbf{w}} [\operatorname{var}(\mathbf{w}^T \mathbf{X})] \quad (\text{E.11})$$

$$\begin{aligned} &= \operatorname{argmax}_{\mathbf{w}} E[\mathbf{w}^T \mathbf{X} - E(\mathbf{w}^T \mathbf{X})][\mathbf{w}^T \mathbf{X} - E(\mathbf{w}^T \mathbf{X})]^T \\ &= \operatorname{argmax}_{\mathbf{w}} E(\mathbf{w}^T \mathbf{X})(\mathbf{w}^T \mathbf{X})^T \\ &= \operatorname{argmax}_{\mathbf{w}} \mathbf{w}^T \Sigma \mathbf{w} \end{aligned} \quad (\text{E.12})$$

$$\mathbf{w}_1^T \mathbf{w}_1 = 1 \quad (\text{E.13})$$

Where $E()$ is the expectation of variable. Σ is the self-covariance of matrix \mathbf{X} that is defined as $E(\mathbf{X}\mathbf{X}^T)$. \mathbf{X} is required to have zero mean.

To solve this problem, rewrite it to the Lagrange form, then take the derivative with respect to \mathbf{w} , and setting it equal to zero:

$$f_1(\mathbf{w}) := \mathbf{w}^T \Sigma \mathbf{w} - \alpha_1 (\mathbf{w}^T \mathbf{w} - 1) \quad (\text{E.14})$$

$$\frac{d}{d\mathbf{w}} f_1(\mathbf{w}) = 2\mathbf{w}^T \Sigma - 2\alpha_1 \mathbf{w}^T \quad (\text{E.15})$$

$$\mathbf{w}_1 = \operatorname{argmax}_{\mathbf{w}} f_1(\mathbf{w}) \quad (\text{E.16})$$

$$= \operatorname{root}\left(\frac{d}{d\mathbf{w}} f_1(\mathbf{w})\right)\Big|_{\mathbf{w}=\mathbf{w}_1} = 0 \quad (\text{E.17})$$

$$\Sigma \mathbf{w}_1 = \alpha_1 \mathbf{w}_1 \quad (\text{E.18})$$

It is already known that \mathbf{w}_1 is an eigenvector of Σ from **Equ. E.18**. Also the goal is to maximize $\mathbf{w}_1^T \Sigma \mathbf{w}_1$ from **Equ. E.12**. Finally combined with \mathbf{w}_1 's unity

condition (**Equ.** E.13), the goal is to maximize

$$\mathbf{w}_1^T \Sigma \mathbf{w}_1 = \mathbf{w}_1^T \alpha_1 \mathbf{w}_1 = \alpha_1 \quad (\text{E.19})$$

This indicates that \mathbf{w}_1 is the eigenvector corresponding to the largest eigenvalue α_1 of Σ .

Next to show that data projected on the i -th largest principal component \mathbf{w}_i , $\forall i \in \{2, 3, \dots, r\}$ has the i -th largest variance.

Before the proof, the uncorrelated definition and unity definition between \mathbf{w}_i and \mathbf{w}_j , $\forall i, j \in \{1, 2, \dots, r\}$ should be reinstated:

$$\mathbf{w}_i^T \mathbf{w}_j = 0, \forall i \neq j \quad (\text{E.20})$$

$$\mathbf{w}_i^T \mathbf{w}_j = 1, i = j \quad (\text{E.21})$$

Given \mathbf{w}_i , $\forall i \in \{1, 2, \dots, n\}$, $n < r$, the eigenvector corresponding to the i -th largest eigenvalue, for the $(n + 1)$ -th principal component \mathbf{w}_{n+1} , it maximize $\mathbf{w}_{n+1}^T \Sigma \mathbf{w}_{n+1}$ and it has an additional constraint by **Equations** E.20 that it should be different from \mathbf{w}_i , $\forall i \in \{1, 2, \dots, n\}$, $n < r$. To summarize and write in the Lagrange form

$$f_{n+1}(\mathbf{w}) := \mathbf{w}^T \Sigma \mathbf{w} - \alpha_{n+1}(\mathbf{w}^T \mathbf{w} - 1) - \sum_{i=1}^n \beta_i \mathbf{w}^T \mathbf{w}_i \quad (\text{E.22})$$

$$\frac{d}{d\mathbf{w}} f_{n+1}(\mathbf{w}) = 2\mathbf{w}^T \Sigma - 2\alpha_{n+1} \mathbf{w}^T - \sum_{i=1}^n \beta_i \mathbf{w}_i^T \quad (\text{E.23})$$

$$\mathbf{w}_{n+1} = \operatorname{argmax}_{\mathbf{w}} f_{n+1}(\mathbf{w}) \quad (\text{E.24})$$

$$= \operatorname{root}\left(\frac{d}{d\mathbf{w}} f_{n+1}(\mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}_{n+1}} = 0\right) \quad (\text{E.25})$$

$$2\mathbf{w}_{n+1}^T \Sigma = 2\alpha_{n+1} \mathbf{w}_{n+1}^T - \sum_{i=1}^n \beta_i \mathbf{w}_i^T \quad (\text{E.26})$$

By postmultiplying \mathbf{w}_i , $\forall i \in \{1, 2, \dots, n\}$ on the transposed version of **Equ.** E.26 and note Σ is diagonal, then use **Equ.** E.20 again β_i satisfies

$$\beta_i = 0, \forall i \in \{1, 2, \dots, n\} \quad (\text{E.27})$$

By using the transposed version of **Equations** E.26 and E.27 it is true that

$$\Sigma \mathbf{w}_{n+1} = \alpha_{n+1} \mathbf{w}_{n+1} \quad (\text{E.28})$$

Equ. E.28 tells that \mathbf{w}_{n+1} is an eigenvector of Σ . Also recall that the goal is to

maximize $\mathbf{w}_{n+1}^T \Sigma \mathbf{w}_{n+1}$. Finally combined with \mathbf{w}_{n+1} 's unity condition (**Equ.** E.21) and the fact that \mathbf{w}_i 's are uncorrelated with each other (**Equ.** E.20), the goal is to maximize

$$\mathbf{w}_{n+1}^T \Sigma \mathbf{w}_{n+1} = \mathbf{w}_{n+1}^T \alpha_{n+1} \mathbf{w}_{n+1} = \alpha_{n+1} \quad (\text{E.29})$$

As $\mathbf{w}_i, \forall i \in \{1, 2, \dots, n\}, n < r$ is the eigenvector corresponding to the i -th largest eigenvalue, \mathbf{w}_{n+1} can only be the eigenvector corresponding to the $(n + 1)$ -th largest eigenvalue α_{n+1} of Σ . ■

APPENDIX F CONSTELLATION PROFILES

Selected notable brain landmarks are defined in this appendix. Some of them are midbrain and point-typed landmarks including *ac*, *pc*, *MPJ*, *4VN*, *genu*, *rostrum*, *optic chiasm*, *BPons* and, *aq-4V*, *etc.*; others are off-midbrain and quasi-landmarks including *LE*, *RE*, *left/right ventricular head*, *etc.* The definitions of the landmarks can be found either in the figure captions in this appendix or in **Appendix A**.

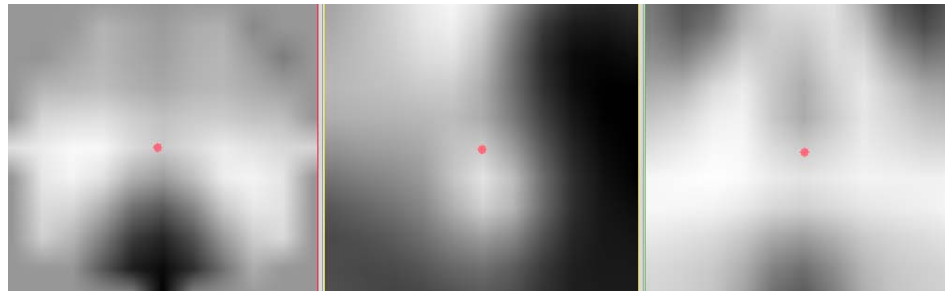
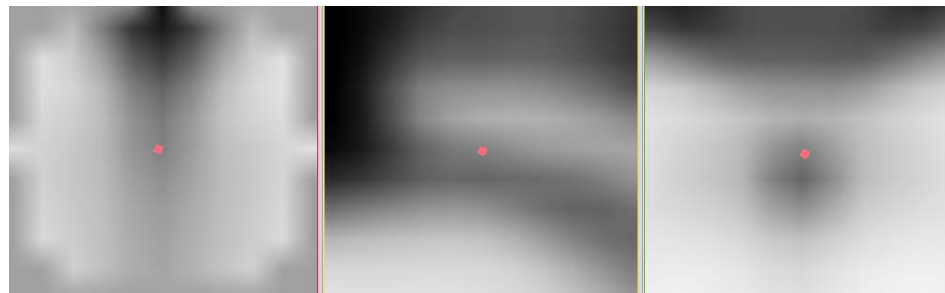
(a) Location for *ac*(b) Template for *ac*(c) Location for *pc*(d) Template for *pc*

Figure F.1: Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the centroid of anterior commissure (*ac*) and the superior aspect of posterior commissure (*pc*)

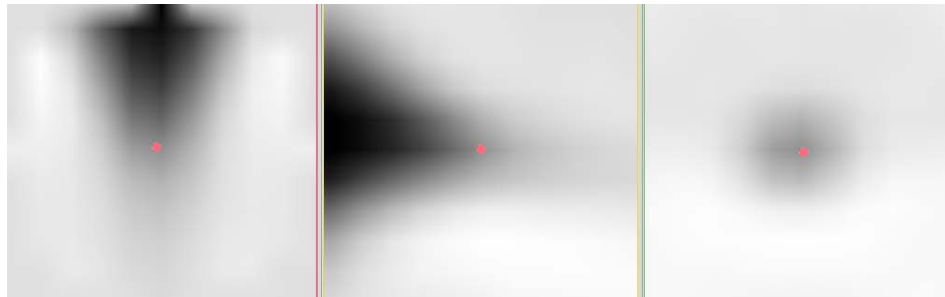
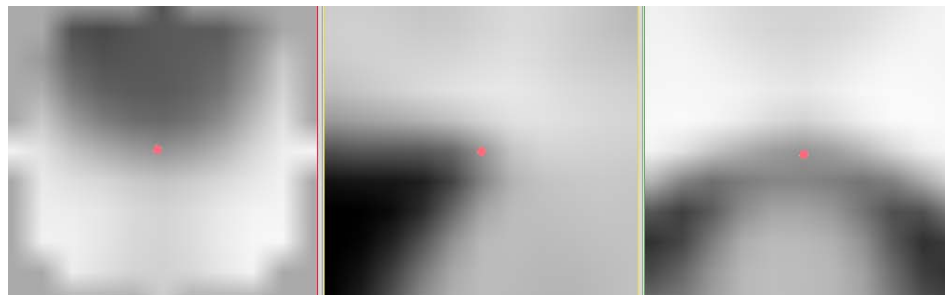
(a) Location for *MPJ*(b) Template for *MPJ*(c) Location for *4VN*(d) Template for *4VN*

Figure F.2: Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the midbrain pontine junction (*MPJ*) and the fourth ventricle notch (*4VN*)

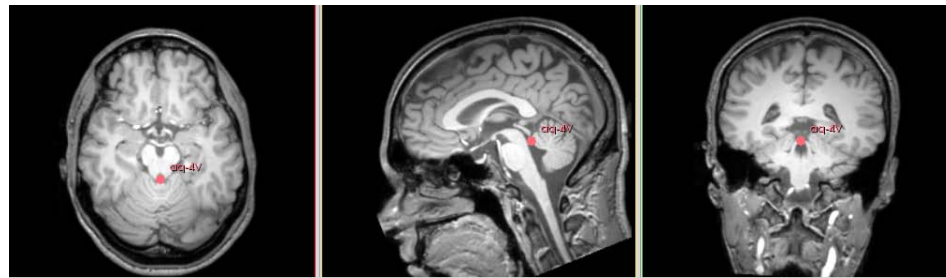
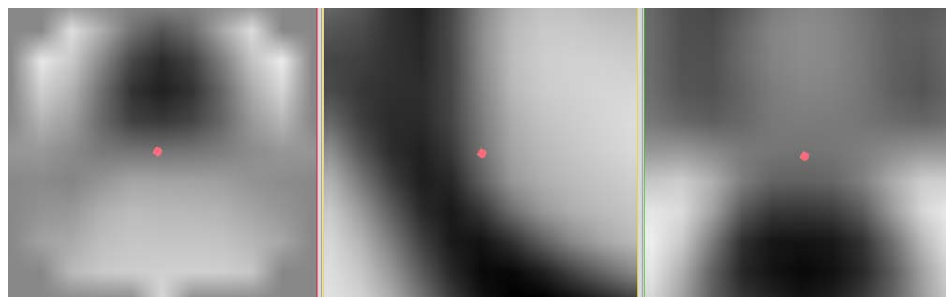
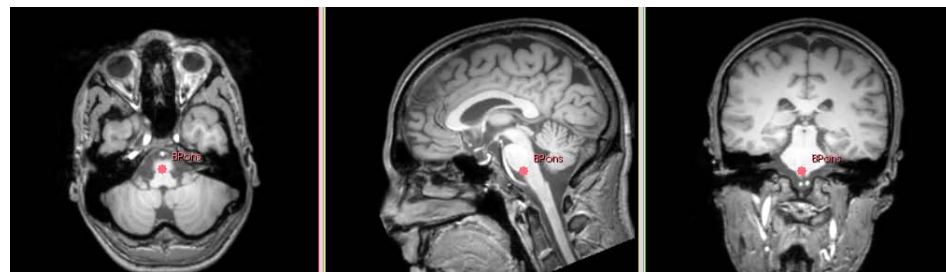
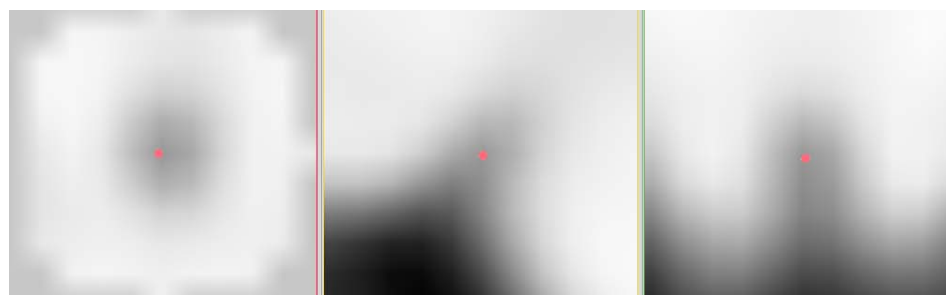
(a) Location for *aq-4V*(b) Template for *aq-4V*(c) Location for *BPons*(d) Template for *BPons*

Figure F.3: Typical acpc-aligned, T_1 -weighted central slices showing the junction of cerebral aqueduct and the fourth ventricle (*aq-4V*) and basal pons (*BPons*)

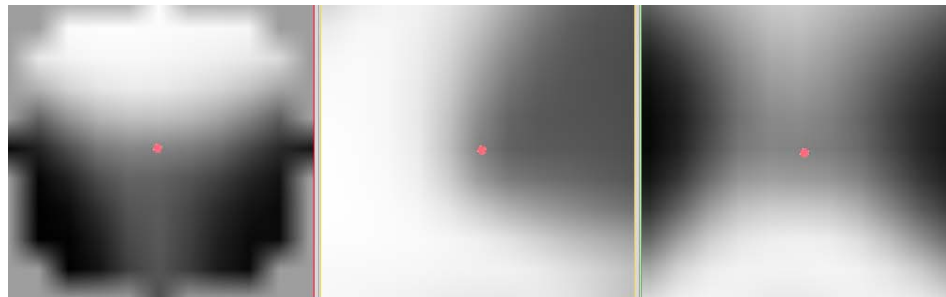
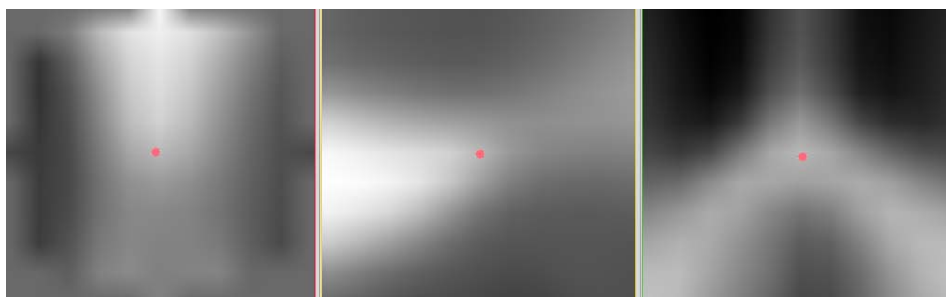
(a) Location for *genu*(b) Template for *genu*(c) Location for *rostrum*(d) Template for *rostrum*

Figure F.4: Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for genu in anterior corpus callosum (*genu*) and the pointed tip of rostrum of the corpus (*rostrum*)

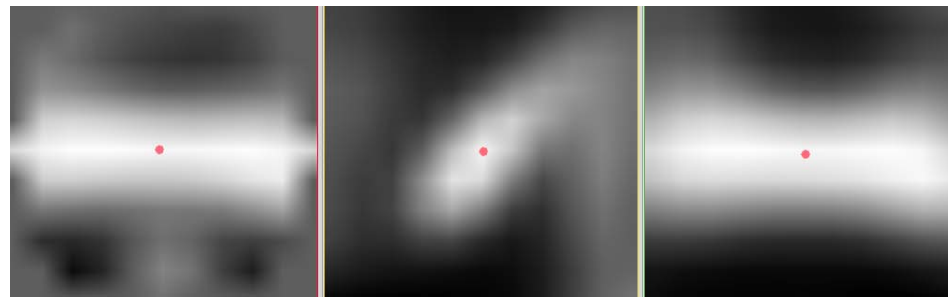
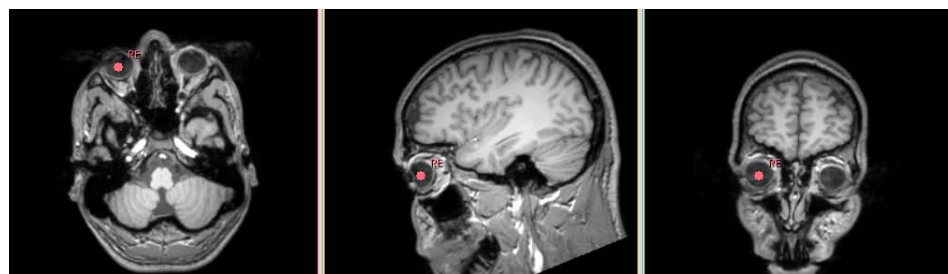
(a) Location for *optic_chiasm*(b) Template for *optic_chiasm*(c) Location for *LE*(d) Location for *RE*

Figure F.5: Typical *apc*-aligned, T_1 -weighted central slices showing the constellation profile for centroid of optic chiasm (*optic_chiasm*) and the centroid of the left/right eye (*LE/RE*)

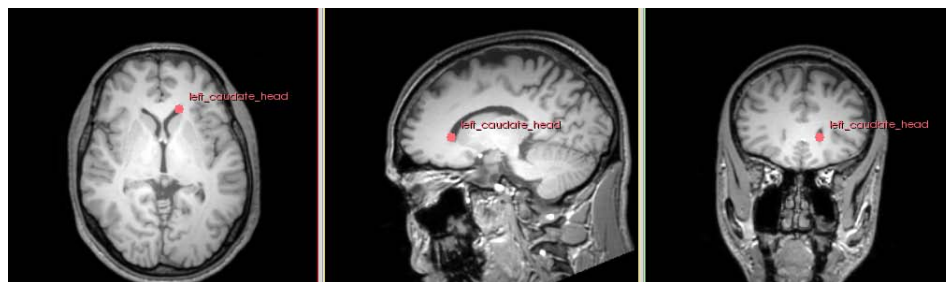
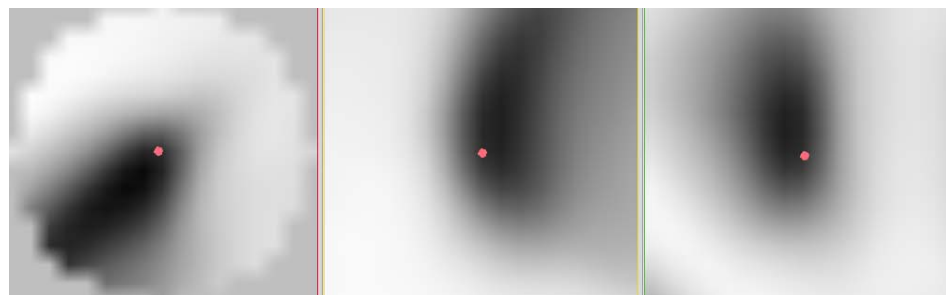
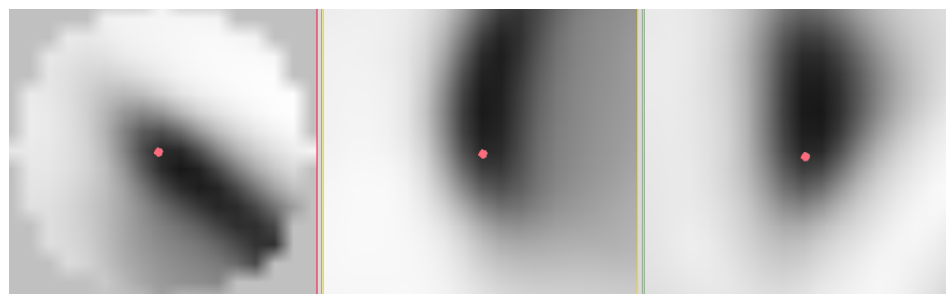
(a) Location for *left_ventricular_head*(b) Template for *left_ventricular_head*(c) Location for *right_ventricular_head*(d) Template for *right_ventricular_head*

Figure F.6: Typical acpc-aligned, T_1 -weighted central slices showing the constellation profile for the most anterior CSF in left/right ventricular nucleus (*left/right_ventricular_head*)

APPENDIX G

BRAINS CONSTELLATION MODELER MANUAL

The main usage of *BRAINS* constellation modeler is to build a template model as well as a linear estimation model from training datasets for *BRAINS* constellation detector (in **Appendix H**). The tool belongs to the *BRAINSConstellationDetector* project [20] that is one of a series in the *BRAINS* tool suite registered at *NITRC* [35]. The models can be simply considered as some statistical information extracted from the training datasets. This appendix will discuss the typical usage of the tool by how to accomplish the following common tasks:

- How to generate a template model for customized landmarks
- How to generate a linear estimation model for customized landmarks

Users can skip this manual if they prefer to use default model files. We have already created models for dozens of landmarks (in **Appendix F**).

G.1 Generate a template model for customized landmarks

If users plan to create their own template model, first note that the tool to achieve this goal is command line-based, so the basic usage looks like this:

```
1 ${BINARY_DIR}/./BRAINSConstellationModeler \
  OPTION1 [ARGUMENT1] [OPTION2 [ARGUMENT2] ...]
```

Here, "[]" means that the component enclosed is optional. For more usage information, type the following command in a terminal:

```
${BINARY_DIR}/./BRAINSConstellationModeler -h
```

The main input for the modeler is a few *NIfTI-1* formatted [29] image files to be trained and some files containing the landmarks locations of the images. Users can

input this important information by specifying the filename of a list that contains the filename of each image and landmark file to be used for training. The corresponding option is *-i* or *--inputTrainingList*. Note the training list file should be in a simple document format such as *txt*. Users need to enter three categories of information in the file. They are

- General training parameters
- Template size
- Image and landmark filenames

The definition of the training list file is summarized as follows:

```

1 numberOfDatasets searchBoxDims resolutionUnits
  initialRotationAngle initialRotationStep numRotationSteps
3
  landmarkName_1 templateRadius templateHeight
5 landmarkName_2 templateRadius templateHeight
  ...
7 END

9 trainingImageFilename_1
  traininglandmarkFilename_1
11 END
  trainingImageFilename_2
13 traininglandmarkFilename_2
  END
15 ...

```

The landmark file is supported by *NA-MIC* [36], and can be created and modified by some popular medical image viewer with *Fiducials* module such as the *3DSlicer* software [32]. An example of how it may look like is shown in the following listing:

```

1 # Fiducial List file /ACPCModel/data/test.fcsv
  # version = 2
3 # name = test.fcsv
  # numPoints = 19
5 # symbolScale = 5
  # symbolType = 13
7 # visibility = 1
  # textScale = 4.5
9 # color = 0.4,1,1

```

```

# selectedColor = 1,0.5,0.5
11 # opacity = 1
# ambient = 0
13 # diffuse = 1
# specular = 0
15 # power = 1
# locked = 0
17 # numberingScheme = 0
# columns = label , x , y , z , sel , vis
19 AC, -127.032, -128.216, 125.942, 1, 1
PC, -125.988, -155.846, 125.756, 1, 1
21 MPJ, -126.098, -152.606, 113.279, 1, 1
...

```

The other compulsory argument for generating a user-specified template model file is to indicate the desired output template model filename by using the `--outputModel` option.

G.2 Generate a linear estimation model for customized landmarks

The linear estimation model builds the linear morphometric relationship among landmarks by applying EPCA (in **Subsection 3.3**) to their instances in training datasets. Along with the local search module, this model is used to estimate the search centers for newly introduced or customized landmarks iteratively. To build a linear estimation model for customized landmarks, user should modify two files, named *load_landmarks.m* and *train_LM_EPCA.m* in the training model source directory. To ensure a successful training, the modeler requires the accurate information of six mandatory landmarks. They are *ac*, *pc*, *MPJ*, *4VN*, *LE*, and *RE*. The definition of the landmarks can be found in **Appendix F**.

The *load_landmarks.m* is a *Matlab* [37] script file that tells the modeler that landmarks should be taken into account. An example of the script is shown in the following listing:

```

%% Initialization
2 AC = [];
PC = [];

```

```

4 ...
  % Add any customized landmarks here
6
  %% Add landmarks location info from training datasets
8 % This following line will try to add a new
  % AC landmark (-127.164,-127.69,127.285)
10 AC = [AC; -127.164, -127.69, 127.285];
  AC = [AC; -127.032, -128.216, 125.942];
12 ...
  PC = [PC; -126.642, -155.162, 126.514];
14 PC = [PC; -125.988, -155.846, 125.756];
  ...
16 % Add any customized landmarks location info here

18 %% Transform landmarks from RAS space to LPS space
  %if the landmark location is obtained by 3DSlicer
20 AC(:,1) = -AC(:,1);
  AC(:,2) = -AC(:,2);
22 PC(:,1) = -PC(:,1);
  PC(:,2) = -PC(:,2);
24 ...
  % Transform any customized landmarks here

```

The *train_LM_EPCA.m* script is used to compute the mapping of principal components of any landmark space and, find the optimal linear model parameters in terms of least squares. An example of the script is shown in the following listing:

```

1 %% Load landmarks in training datasets
  load_landmarks
3
  %% Variables & constants declaration
5 dim = 3; %3D points
  numNewLandmarks = 13; % number of customized landmarks
7 numDatasets = size(AC,1);
  numPCs = 10; % number of principal components we need
9 newLandmarks = cell(numNewLandmarks,1); % new landmarks cell
  newLandmarks{1} = aq_4V;
11 newLandmarks{2} = genu;
  ...
13 % Add your customized landmarks here

15 newLandmarksName{1} = 'aq-4V';
  newLandmarksName{2} = 'genu';
17 ...
  % Add the names of your customized landmarks here

```

After finishing the editing of the two script files, in *Matlab* directly run script

in terminal:

```
train.LM_EPCA
```

The resulted linear estimation model will be written into a file named *LME_EPCA.txt*. The last step is to update the linear estimation model in the *llsModel.txt* file in the $\{\text{SOURCE_DIR}\}$ and the $\{\text{BINARY_DIR}\}$ directories by the new parameters that are created in the *LME_EPCA.txt* file.

APPENDIX H

BRAINS CONSTELLATION DETECTOR MANUAL

The *BRAINS* constellation detector has two identities: both as an image registration tool and as a constellation detector for human brain MR scans. The tool belongs to the *BRAINSConstellationDetector* project [20] that is one of a series in the *BRAINS* tool suite registered at *NITRC* [35]. This appendix will discuss the typical usage of the tool by how to accomplish the following common tasks:

- How to generate a *NIfTI-1*-formatted [29] acpc-aligned image
- How to generate an *ITK*-recognized [26] registration transformation
- How to generate a *NA-MIC*-supported [36] landmark file
- How to tune the parameters of the detector
- How to handle the failure of automated detection

The tool is a command line-based application, so the basic usage looks like this:

```
1 ${BINARY_DIR}/./BRAINSConstellationDetector \
  OPTION1 [ARGUMENT1] [OPTION2 [ARGUMENT2] ...]
```

Here, ‘‘[]’’ means that the component enclosed is optional. For more usage information, type the following command in a terminal:

```
`${BINARY_DIR}/./BRAINSConstellationDetector -h
```

Some practical use cases are also provided in a tutorial webpage [38] that is hosted by *NITRC*. Users might want to look at them before dealing with their own tasks.

H.1 Generate a NIfTI-1-formatted acpc-aligned image

The detector works with *NIfTI-1* data format that is proposed by Neuroimaging Informatics Technology Initiative (NIfTI) [29]. Providing the filename of a *NIfTI-1*-formatted brain image with arbitrary image direction, spacing, and origin by using the *-i* or *--inputVolume* option, the detector is able to generate an acpc-aligned image automatically. Users can use the *--outputResampledVolume* option to obtain an isotropic and resampled aligned image, or they can use the *-o* or *--outputVolume* option to obtain an unresampled but direction cosine-modified image that has identical voxel contents as the input image has, and the same physical representation as the resampled image in the LPS coordinate system 2.4.3 (*i.e.* in the physical space). Consistently choosing to apply the latter option can effectively reduce the resampling errors throughout a large image-processing pipeline. Its idea has already been adapted to many other *BRAINS* software. In addition, due to the large volume of a *3D* medical image, the resulted output image is compressed, usually featured by a *.nii.gz* filename extension.

H.2 Generate an ITK-recognized registration transformation

Users can obtain the resulted versor transformation from the output aligned image (*i.e.* the moving image) to the original input image (*i.e.* the fixed image) by using the *-t* or *--outputTransform* option. The resulted registration transformation will be written to a file with a *.mat* filename extension that is recognized by *ITK* [26] as well as *3DSlicer* [32].

H.3 Generate a NA-MIC-supported landmark list file

The resulted landmark list file is supported by National Alliance for Medical Image Computing (*NA-MIC*) [36] and is readable from the *Fiducials* module in its *3DSlicer* software [32]. The landmark list file is identified by a *.fcsv* filename extension. Users can use the `--outputLandmarksInInputSpace` option to generate a landmark list file in the input space where the input image lies. Alternatively, they can specify the output landmark list file for the *acpc*-aligned image by using the `--outputLandmarksInACPCAlignedSpace` option as long as the aligned image is requested as well.

H.4 Tune the parameters of the detector

The default parameters of the program are stored in an XML file along with the source files. Of course, a preset parameter set won't fit for all situations. In response to the problem, users are provided with the convenience to modify various parameters of the tool with proper options and arguments in command line. For instance, users can:

Specify a different template model file: We provide three kinds of built-in models: T_1 -weighted, T_2 -weighted, and PD. They are files ending with a *.mdl* filename extension, and can be found in `${TestData.DIR}`. The default model is set to the T_1 -weighted model. Users are free to switch from different kinds of models, or even use their own. To specify a different model file, users can use the `--inputTemplateModel` option. Also note that if the target input is a T_2 -weighted or PD dataset, users are responsible to set the Hough eye detector mode to 0 accordingly by specifying the option and argument like this:
`--houghEyeDetectorMode 0;`

Specify a different linear estimation model file: Users can specify a different

linear estimation model file by the `--inputLLSModel` option. The linear estimation model file is stored in the `${TestData.DIR}` directory, and named as `llsModel.txt`. The definition of the model content is described in **Appendix G**.

Apply different interpolation methods: Users can apply different interpolation methods for the application. The available options are *Linear*, *NearestNeighbor*, *BSpline*, or *WindowedSinc*. The *Linear* interpolation method is set as default. Users can use the command option `--interpolationMode` to change the interpolation methods.

H.5 Handle the failure of automated detection

The constellation detector is designed in a way so that it can take the users' feedback as a hint for its own processing. Users can interact with the detector by applying related options and providing corresponding arguments. For instance, users can:

Provide user-specified landmarks information: When the detector is about to estimate a certain landmark, it will first try to fetch the corresponding information from users. It will simply skip the estimation phase for that landmark and use the right information provided by the user directly as long as it finds one in the user-specified landmark list file. Users can specify the landmark list file by using the option `--inputLandmarksEMSP`. Note the landmarks should be in the *EMSP*-aligned space. The detector requires this because many landmarks especially those who are midline points are much easier to pick, and the resulted landmarks are considered to be more accurate in the *EMSP*-aligned space. Users can request the detector to initial a pair of landmark list file and a resampled input image in the *EMSP*-aligned space for later manual correction use by setting the “*write debugging images level*” option argument be

greater than one. The resulted landmark list file is named as *EMSP.fcsv* and the resampled image is named as *EMSP.nii.gz*. Both of them can be found in the $\{\text{DEBUG_IMAGE_DIR}\}$ directory that can be set through proper option and argument. After correcting the landmark locations with some handy GUI tools such as the *BRAINS* Constellation corrector that will be discussed in **Appendix I**, users can feed the corrected *EMSP.fcsv* landmark list file to the detector by the *--inputLandmarksEMSP* option.

Set write debugging image level and debugging image directory: The “*write debugging image level*” option controls the amount of exposure for debugging information. The default level is set to 0 that means no debugging images would be written to disk. Users can specify the debugging image level by using the option *--writedebuggingImagesLevel*. Similarly, users can use the option *--resultsDir* to specify the path where the debugging images should be written. The two options will be very handy when the user wants to correct some estimation result or interact with the detector.

Notify the detector about the failure of eye detection: Although the eye detector has some simple and innate mechanism to check if the detected eye centers are reasonable (as discussed in **Section 3.2.3**), sometimes it still may fail. In this case users can notify the failure of the eye detection by specifying the option *--forceHoughEyeDetectorReportFailure*. This will force the detector to terminate the program after it has written to disk a pair of empty landmark list file and a resampled input image in the *EMSP*-aligned space for later manual correction. The resulted landmark list file is named as *EMSP.fcsv* and the resampled image is named as *EMSP.nii.gz*. Both of them can be found in the $\{\text{DEBUG_IMAGE_DIR}\}$ directory. User can load the two files and manually

specify the correct eye center locations by the proposed GUI tool that is introduced in **Appendix I**. Finally, user may want to use the technique that is described in **Section H.5** to use the corrected eye centers information in the detection process.

APPENDIX I

BRAINS CONSTELLATION GUI CORRECTOR MANUAL

The *BRAINS* constellation GUI corrector is a lite medical image viewer that provides the convenience for the users to correct landmark locations in a visual and efficient way. The tool was developed within the *Qt* framework [39] and the slice viewers are rendered with the help of the *VTK* [40] APIs. It belongs to the *BRAIN-SConstellationDetector* project [20] that is one of a series in the *BRAINS* tool suite registered at *NITRC* [35]. This appendix will discuss the typical usage of the tool by how to accomplish the following common tasks:

- How to load a *NIfTI-1*-formatted [29] image and a *NA-MIC*-supported [36] landmark list file
- How to generate a corrected landmark list file
- How to interact with the GUI tool

The GUI tool is invoked from a terminal, so the basic usage looks like this:

```
1 ${BINARY_DIR}/./BRAINSConstellationDetectorGUI \
  OPTION1 [ARGUMENT1] [OPTION2 [ARGUMENT2] ...]
```

Here, "[]" means that the component enclosed is optional. For more usage information, type the following command in a terminal:

```
${BINARY_DIR}/./BRAINSConstellationDetectorGUI -h
```

Some practical use cases are also provided in a tutorial page [38] hosted by *NITRC*. Users might want to look at them before dealing with their own tasks.

I.1 Load a NIfTI-1-formatted image and a NAMI-supported landmark list file

Users can specify the filename of an input *NIfTI-1*-formatted brain image by using the *-i* or *--inputVolume* option. Similarly, they can use the *-l* or *--inputLandmarks* option to tell the corrector that landmark list file is associated with the input image. Note the input image and input landmarks in the landmark list file should all be in the *EMSP*-aligned space because many landmarks especially those who are midline points are much easier to pick, and the resulted landmarks are considered to be more accurate in the *EMSP*-aligned space. The constellation detector has an option to generate those files automatically for users. For details on how to generate these two files, please look at **Appendix H.5**.

I.2 Generate a corrected landmark list file

This task can be achieved either by specifying the filename of the list file in command line with an option of *-o* or *--outputLandmarks*, or users can take the advantage of the **Save** button in the GUI tool. For example, if the output landmark list filename is specified in advance, clicking on the **Save** button will save the landmark information to the file with the specified filename. Else, if the input landmark list filename is specified, clicking on the **Save** button will result in saving the changes to the original landmark list file. Else, clicking on the **Save** button will trigger the same dialog as clicking on the **Save As** button. Detailed information on how to interact with the GUI tool will be discussed right away in the next section.

I.3 Interact with the GUI tool

The GUI tool is mainly composed of an image volume information module, a landmark list module, and a slice viewer module containing three viewers with axial, sagittal, and coronal views respectively. The modules are arranged as follows:

- The image information is shown at the left upper side;
- The landmarks list is shown at the left lower side;
- Three different viewers are located at the right upper side;
- The slider bars for each viewer are just below the viewers;
- All the buttons are at the bottom of the window.

A screenshot of the corrector is shown in **Fig. I.1**. Users can perform the following typical operations on landmark(s):

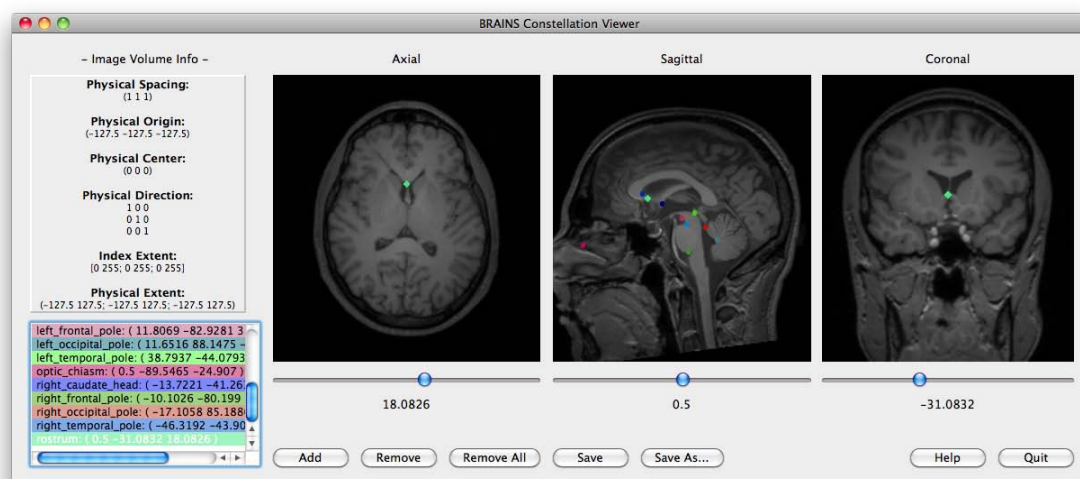


Figure I.1: A screenshot for *BRAINS* constellation GUI corrector

Adding a new landmark: Click the **Add** button at the bottom of the tool.

Selecting a landmark: Click on the landmark in the label list.

Moving a landmark: Select the landmark, move either one of the slider bar to find a proper slice, then click at a location in the corresponding viewer to move the landmark to the current place. Adjust the 3D location of the landmark with the help of different viewers.

Zooming in/out a viewer: Hover the mouse on the viewer, then right click then drag up/down to zoom in/out about the center.

Removing a landmark: Double click on the landmark in the label list, or select the landmark, then click on the **Remove** button at the bottom of the GUI. Click the **Accept** button when a follow-up dialog appears.

Others: Other operations such as **Remove All** landmarks, **Save** landmarks, etc can be achieved by clicking proper buttons at the bottom of the GUI. These basic operations can also be viewed inside the GUI corrector by clicking on the **Help** button.

APPENDIX J

SOFTWARE USAGE FOR ITK RESAMPLE IN-PLACE IMAGE FILTER

The filter is written in C++, and is depended on the *ITK* [26] framework.

First include the header file by

```
1 #include "itkResampleInPlaceImageFilter.h"
```

Define the resample in-place image filter by

```
1 typedef itk::ResampleInPlaceImageFilter
   < ImageType, ImageType > FilterType;
```

Set up the filter and process the input volume

```
FilterType::Pointer filter = FilterType::New();
2 filter->SetInputImage( inputImage );
  filter->SetRigidTransform( transform );
4 filter->Update();
ImagePointer outputImage = filter->GetOutput();
```

REFERENCES

- [1] F. Bookstein. *Morphometric tools for landmark data geometry and biology, second edition*. Cambridge Univ. Press, New York, 1997.
- [2] J. Maintz and M. Viergever. A survey of medical image registration. *Med. Imag. Anal.*, 2(1):1–36, 1998.
- [3] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Trans. Med. Imag.*, 20(6), Jun. 2001.
- [4] F. Bookstein. Four metrics for image variation. *Proc. 11th Int. Conf. Information Processing Med. Imag.*, 363:227–240, 1991.
- [5] A. Babak and H. Alvin. Model-based automatic detection of the anterior and posterior commissures on mri scans. *NeuroImage*, 46:677–682, 2009.
- [6] L. V erard, P. Allain, J. Trav ere, J. Baron, and D. Bloyet. Fully automatic identification of the ac and pc landmarks on brain mri using scene analysis. *IEEE Trans. Med. Imag.*, 16:610–616, 1997.
- [7] H. Abdi and L. Williams. *Principal Component Analysis*. Wiley Interdisciplinary Reviews: Computational Statistics, 2. <http://wires.wiley.com/WileyCDA/>, 2010.
- [8] I. Jolliffe. *Principal Component Analysis, second edition*. Springer, NY, 2002.
- [9] R. Davies, C. Twining, T. Cootes, and C. Taylor. Building 3-d statistical shape models by direct optimization. *IEEE Trans. Med. Imag.*, 29(4), Apr. 2010.
- [10] P. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, Apr. 1936.
- [11] A. Toga. *Brain Warping*. Academic Press, New York, 1998.
- [12] B. Zitova and J. Flusser. Image registration methods: a survey. *Imag. Vis. Comput.*, 21:977–1000, 2003.
- [13] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys.*, 30:292–304, 1979.
- [14] G. Nielson. Scattered data modeling. *IEEE Comput. Graphics Appl.*, 13:60–70, Jan. 1993.

- [15] M. Davis, A. Khotanzad, D. Flamig, and S Harms. A physics-based coordinate transformation for 3-d image matching. *IEEE Trans. Med. Imag.*, 16(3), Jun. 1997.
- [16] K. Johnson. Neuroimaging primer. <http://www.med.harvard.edu/AANLIB/hms1.html>.
- [17] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognit.*, 132(2):111–122, Apr. 1981.
- [18] K. Mosaliganti, A. Gelas, P. Cowgill, and S. Megason. An optimized n-dimensional hough filter for detecting spherical image objects. *The Insight Journal*, July-December 2009.
- [19] Britannica Inc. *Encyclopedia Britannica Macropedia: Sensory Reception*. Britannica Inc., USA, 1997.
- [20] Nitrc wikipe for brains constellation detector. <http://www.nitrc.org/plugins/mwiki/index.php?title=brainscdetector:MainPage>.
- [21] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, SMC-9(1), Jan. 1979.
- [22] M. Powell. An efficient method of finding the minimum of a function of several variables without calculating derivatives. *Computer J.*, 7:155–162, 1964.
- [23] R. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, Englewood-Cliffs, N.J., 1973.
- [24] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C: The art of scientific computing*. Cambridge University Press, 1992.
- [25] N. Dodgson. Variation and extrema of human interpupillary distance. *Proc. SPIE*, 5291:36–46, 2004.
- [26] Homepage for insight segmentation and registration toolkit (itk). <http://www.itk.org/Insight/>.
- [27] itkresampleinplaceimagefilter in insight journal. <http://www.insight-journal.org/browse/publication/770>.
- [28] I. Grachev, D. Berdichevsky, S. Rauch, S. Heckers, D. Kennedy, V. Caviness, and N. Alpert. A method for assessing the accuracy of intersubject registration of the human brain using anatomic landmarks. *NeuroImage*, 9:250–268, 1999.

- [29] Homepage for nifti, the neuroimaging informatics technology initiative. <http://nifti.nimh.nih.gov/>.
- [30] Hui Zhang. Homepage for the spatial normalization and atlas construction toolkit (dti-tk). <http://www.nitrc.org/projects/dtitk>.
- [31] H. Johnson and G. Christensen. Consistent landmark and intensity-based image registration. *IEEE Trans. Med. Imag.*, 21(5), May 2002.
- [32] Homepage for 3d slicer. <http://www.slicer.org/>.
- [33] Lists of integrals on wikipedia. http://en.wikipedia.org/wiki/Lists_of_integrals.
- [34] E. Alpaydin. *Introduction to Machine Learning, second edition*. The MIT Press. Cambridge, Massachusetts. London, England, 2010.
- [35] Homepage for brains software. <http://www.nitrc.org/projects/brains/>.
- [36] Homepage for na-mic, the national alliance for medical image computing. <http://www.na-mic.org/>.
- [37] Homepage for matlab. <http://www.mathworks.com/>.
- [38] Tutorial page for brains constellation detector. <http://www.nitrc.org/plugins/mwiki/index.php/brainscdetector:Tutorial>.
- [39] Homepage for qt. <http://qt.nokia.com/>.
- [40] Homepage for visualization toolkit (vtk). <http://www.vtk.org/>.