Study of Non-Rigid Image Registration Algorithms

Di YANG
u4476533

A thesis submitted for the degree of
Master of Philosophy
School of Engineering,
College of Engineering and Computer Science,
The Australian National University

THE AUSTRALIAN NATIONAL UNIVERSITY

Declaration

This thesis is an original work. None of the work has been previously submitted by me for the purpose of obtaining a degree or diploma in any university or other tertiary education institution. To the best of my knowledge, this thesis does not contain material previously published by another person, except where due reference is made in the text.

Di Yang

DI YANG

17/05/2011

Research School of Information Science and Engineering
College of Engineering and Computer Science
The Australian National University

To my parents
who infused me with the first word and character.

To my wife who tells me the truth in love,
particularly when I do not want to hear it.

Acknowledgements

Writing a thesis is not an easy task at all. The development and refinement of this thesis have owed the great help from a lot of people that I feel sincerely grateful to.

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Hongdong LI, for his continuous support in the Master of Philosophy program. Dr. Hongdong LI was always there to listen and to give advice. He is most responsible for helping me complete the writing of this dissertation as well as the challenging research that lies behind it. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. Dr. Hongdong LI has been a friend and mentor. He taught me how to ask questions, express my ideas, write academic papers, made me a better programmer, had confidence in me when I doubted myself, and brought out the good ideas in me. Without his encouragement and constant guidance, I could not have finished this dissertation. He was always there to meet and talk about my ideas, to proofread and mark up my papers and chapters, and to ask me good questions to help me think through my problems (whether philosophical, analytical or computational). Thanks also to my co-supervisor Dr. Lei WANG, with whom I explored the ideas, discussed the details of experiments and studied knowledge of pattern recognition.

I must also give my heartfelt thanks to Prof.Marcus HUTTER, who as my co-supervisor guided my study of the practical implementation of theoretical knowledge to solve industrial problem such as vehicle damage detection through image. I'm grateful to have worked on such challenging projects and to have had such expert guidance. His vast knowledge, his view of the bigger picture and the humility to admit when he didn't know something have been the source of much inspiration. He also kindly offers me an opportunity to continue my further education in the field of engineering.

Besides my advisors, I would like to thank the rest members of research group

and people who gives helps in this dissertation: Dr. Mathieu Salzmann (Ecole Polytechnique Federale de Lausanne, Suisse), who provided precious warping cloth sequence data for my experiments, Prof. Wufan Chen, Prof. Jianhua MA and Prof. Zhentai LU (First Military Medical University), who supply me with double-echo MR image-pairs, Dr. Nianjun LIU, who gave insightful comments and reviewed my work on a very short notice, Mr. Nathan BREWER, who asked me good questions, discussed with me and explained research details for me, and Mr. Srimal JAYAWARDENA for his encouragement and hard questions.

Let me also say 'thank you' to the following people at ANU and NICTA: Ms. Sue VAN HAEFTEN, Ms. Elspeth DAVIES, and Mr. Peter SHEVCHENKO for helping me at any time, and solving any unsolvable problems concerning M.Phill. students. Mr. Jonathan PETERS and Ms. Deb PIOCH (HDR service), who dedicated their precious time and a drawer in the file cabinet for all my complicated records.

I am also greatly indebted to many teachers in the past: Prof. Wufan CHEN and Prof. Jianhua MA (First Military Medical University) for getting me interested in Design Methods and Theory and coming to the Australia. Prof. Bin LI and Prof. Gang YAN (First Military Medical University), for recommending me to participate China Undergraduate Mathematic Contest in Modeling, introducing me to MATLAB programs and concepts, and teaching me programming skills.

Many thanks to my friends, old and new, for being by my side whenever I need them: Yiming JI, Eric HOU, Brad YU, Xiangke WANG, Mingxu LI, Shi WANG, Xi YANG, Haifeng ZHAO and more.

Last, but not least, I thank my family: my parents, Huaixiang YANG, and Hang XIAO, for giving me life in the first place, for educating me with aspects from both arts and sciences, for unconditional support and encouragement to pursue my interests, even when the interests went beyond boundaries of language, field and geography, and my wife Shuyu GUO, for offering me her everlasting love, spiritual consolation, great care and concern. I love you.

Abstract

In this thesis, we study several non-rigid dense image registration algorithms, and a special attention is given to over-fitting and under-fitting problems, which have identified as two major causes to limit accuracy of non-rigid image registration.

In order to eliminate their impacts, two essential factors, leading to over-fitting and under-fitting, are specifically analysed in this thesis. They are ***ill-poseness*** and ***low expression power of geometric transformation***. After that, we propose three new algorithms in this thesis, namely **(a)** two-phase probabilistic second-order Demons Algorithm, **(b)** learning varying dimension radial basis functions (LVDRBF), and **(c)** Cross-cumulative residual entropy (CCRE) based LVDRBF.

- **Two-phase probabilistic second-order Demons Algorithm**

  In this algorithm, a divide-and-conquer strategy is presented to reduce effects of under-fitting and over-fitting through addressing *ill-poseness* and *low expression power of geometric transformation*.

  Firstly, *ill-poseness* is successfully solved with incorporating *priori* information into probabilistic second-order Demons Algorithm. After that, registration errors caused by *low expression of the rectangle free-form transformation* are rectified by a two-phase deformation strategy.

  This two-phase probabilistic Second-order Demons Algorithm [54] extends the applications of Demons Algorithm from homogenous region to texture-rich image.

- **Learning varying dimension radial basis functions (LVDRBF)**

  Different from divide-and-conquer strategy in two-phase probabilistic second-order Demons Algorithm, "learning varying dimension RBF" [53], can tackle *ill-poseness* and *low expression power of geometric transformation*, simultaneously.

To achieve this purpose, a matched data-pairing set is established via Lucas-Kanade-Tomasi feature tracking algorithm to be a latent control point library. Then, the "best" RBF transformation with the optimal number and the appropriate locations of control point is found by an iterative learning process based on this latent control point library, in order to eliminate impacts of *low expression of the rectangle free-form transformation*. Meanwhile, *ill-poseness* is solved by incorporating Bayesian framework into warping parameter estimation in this iterative learning process.

This new algorithm not only increases the accuracy of image registration, but also provides a strategy to establish the "best" RBF transformation and search for optimal corresponding warping parameters.

- **Cross-cumulative residual entropy (CCRE) based LVDRBF**

  Because the need for multi-modality image registration occurs in many applications including computer vision, remote sensing, and especially medical image processing, thus we present an improved "CCRE based learning varying dimension RBF" for multi-modality dense image registration. CCRE as a similarity measure not only contributes to the extension of application of learning varying dimension RBF from single-modality to multi-modality image registration, but also increases its robustness owing to property of CCRE.

Contents

Chapter 1

Introduction

Contents:

1.1 Overview of Image Registration

In this section, we will answer a set of questions concerning "image registration", such as what is image registration? why it is so fundamental? how to classify different types of image registration? what is the generic process flow of image registration?

Image registration is the process of transforming different sets of data into one coordinate system. Data here may be multiple photographs, taken from different modalities, from different times, or from different viewpoints. From an operational point of view, the inputs of registration are two images to be geo-

metrically registered, which are often referred to as the sensed image and the reference image. The output is a geometric transformation, which is a mathematical mapping from points in the sensed image to the corresponding points in the reference image. When all corresponding points are mapped together, the registration is successful. Thus, the determination of this correspondence is a crucial problem specific to the domain of objects being imaged.

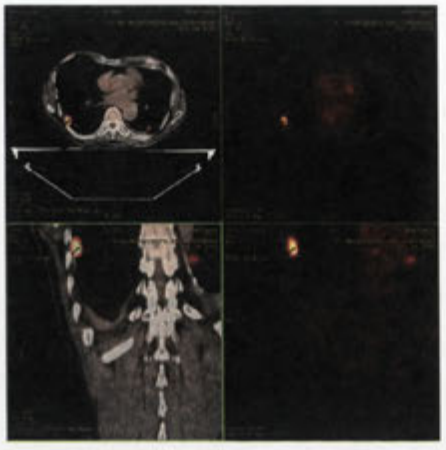
According to the database of the Institute of Scientific Information (ISI), in the last decade, more than 1000 papers have been published on the topic of image registration [57]. Why is image registration so important that attracts so much attention? This is because image registration is a fundamental step in almost all image analysis techniques (*e.g.* Object Tracking [29], Appearance Active Model-AAM [28] and Structure From Silhouette-SFS [14]). Additionally, image registration methods have been applied to several applications as mentioned previously and illustrated in Figure 1.1, namely medical imaging (*e.g.* PET/CT - combination between PET and CT data can assist doctors to monitor tumor growth and verify treatment effects [13]), remote sensing (*e.g.* image mosaicing is used to create super-resolution images, monitor and detect environmental change [12]), computer vision (*e.g.* SFM - finding structure from motion sequence, SFS - finding structure from silhouette [14] [15]), and robotics (*e.g.* Simultaneous Localisation And Mapping (SLAM) - using robots and autonomous vehicles to build up a map within an unknown environment or to update a map within a known environment while keeping track of their current location at same time [17]).

1.1.1 Classification of Image Registration

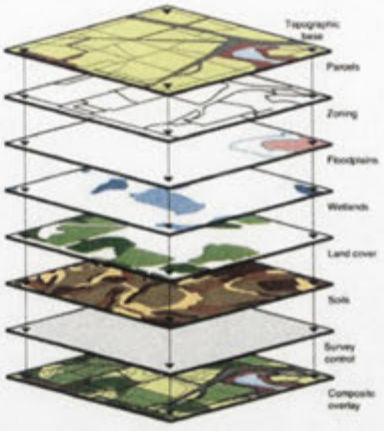
Image registration, as mentioned above, is widely used in remote sensing, medical imaging, computer vision and robotics. In general, its applications can be divided into three main groups based on the way of image acquisition:

- **Different viewpoints (multi-view analysis)**: Images of the same scene are acquired from different viewpoints. The aim of this kind of application is to gain a larger 2D view or a 3D representation of the scene with these aligned images from different viewpoints.

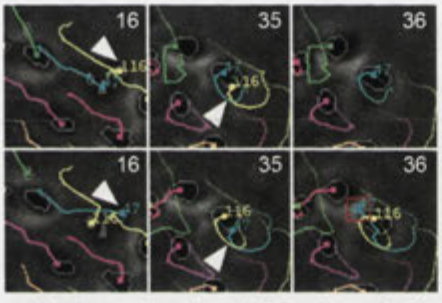
  This application has been extensively used in remote sensing domain, named as image mosaicing, which is to align different surveyed areas to create a super-resolution map (Figure 1.2-a). Additionally, this technique is also



(a) PET/CT Image Fusion

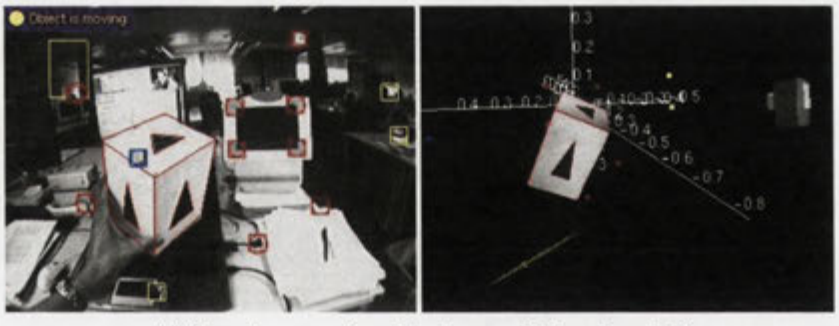


(b) GIS Information Overlapping



(c) Vision-based Cells Tracking [29]

(d) Appearance Active Model [28]



(e) Simultaneous Localisation and Mapping [17]

Figure 1.1: Application domains of image registration: (a). medical imaging-PET/CT used to diagnose tumor; (b). Geographical Information System-data overlapping (Town of Chapel Hill GIS Data and Services); (c). Computer Vision (object tracking)-Vision-based cells tracking [29]; (d). Computer Vision Appearance Active Model [28]; (e). Robotics-Simultaneous Localisation and Mapping [17].

applied in 3D reconstruction, for example, Structure From Silhouette-SFS [14], where the gathering estimated 3D localisation point clouds need a pre-alignment in order to reconstruct a reliable topological 3D model (Figure 1.2-b).

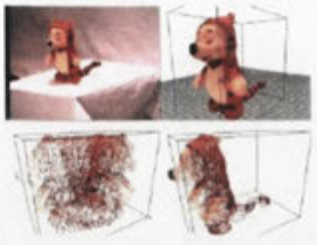
- **Different times (multi-temporal analysis)**: Images of the same scene are acquired at different time, often on regular basis, and possibly under different conditions. It is to find and evaluate differences in the scene which appeared between the consecutive image acquisitions.

  This technique is very important and already widely applied in the modern medical imaging area. For instance, in the image-guided radiation therapy (IGRT) [16], a patient's CT images in different time are registered and analysed to determine pathological trends of tumor and provide relevant information to guide future plan of radiation therapy (Figure 1.2-c). Another similar medical imaging technique is the digital subtraction angiography (DSA), which is able to obtain the clear visualisation of blood vessels in a bony or dense soft tissue environment through subtracting a registered pre-contrast image from the later image produced using contrast medium (Figure 1.2-d).

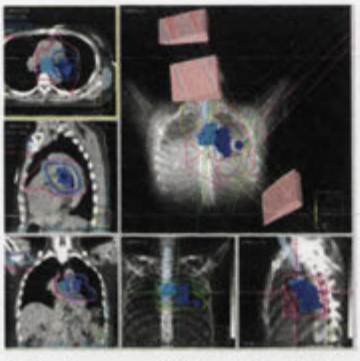
- **Different modalities (multi-modality analysis)**: Images of the same scene are acquired by different modalities, which is also called multi-modality image registration. The aim of this type of implement is to integrate the information obtained from different source streams in order to yield a scene representation with more details. Moreover, multi-modality image registration is able to provide useful information for either industry or medicine. Therefore, there are numerous multi-modality acquisition techniques.

  In medical imaging domain, the primary goal of this technique is to assist doctors to make the efficient and accurate diagnosis through combining different information of both pathology and anatomy, for example combined CT/PET scanner (CT providing human anatomy data, PET supplying human pathological data) (Figure 1.2-e). In industrial area, 3D model are usually required to align with a corresponding 2D image to make actual pose evaluation of object in 2D image (Figure 1.2-f).

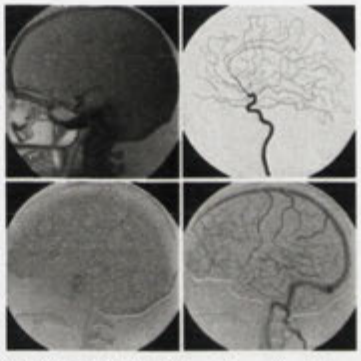
(a) GIS-Map Mosaicing



(b) Structure From Silhouette [14]



(c) Image-guiding Radiation Theraphy (IGRT)



(d) Digital Subtraction Angiography (DSA) of Brain

(e) Combined CT/PET image

(f) 3D-2D Vehicle pose recovery

Figure 1.2: Classification of image registration: 1st Multi-view analysis category: (a) Map Mosaicing (CVLab, University of Bologna); (b) Structure From Silhouette [14]. 2nd Multi-temporal analysis category: (c) Image-guiding Radiation Therapy (URL:www.wordpress.com); (d) Digital Subtraction Angiography. 3rd multi-modality analysis category: (e) combined CT/PET image (New England PET Imaging System); (f) 3D-2D vehicle pose recovery.

1.1.2 Generic Processing Flow

To further understand image registration, we illustrate generic processing flow of image registration in this section. It is difficult to propose an unified definition of image registration, due to numerous image registration techniques are developed for different purpose (*e.g.* GIS, Medicine, Robotics and etc.). Additionally, these techniques must take registration error tolerance, application-dependent data characteristics, and even the noise corruption into consideration. Nevertheless, most image registration methods contain the following four steps:

- **Step 1. Feature detection:** Some image registration methods directly begin with point matching process (*i.e.* Step 2) based on all pixels of the entire image, but others rely on feature detection as the first step.

  Feature detection (*e.g.* closed-boundary regions, contours, corners, etc.) can improve computational efficiency and also make the algorithm immune to noise. For further process, these features can be represented by their point representatives, *i.e.* so called "control points" in the literature.

- **Step 2. Point/Feature matching:** In this step, pixel correspondences will be established between the sensed image and the reference image through minimising or maximising a similarity measure function (*e.g.* minimising sum of squared errors (SSE) for single-modality registration [30]; maximising mutual information for multi-modality registration [50]). Additionally, the pixel correspondences will be fitted to a specific geometric transformation, which is used to represent the deformation between the two images.

- **Step 3. Geometric transformation estimation:** Geometric transformation estimation could be referred as the most important step of all steps. As mentioned previously, an estimated geometric transformation is the output of image registration, which is actually used to align the sensed image with the reference image.

  Generally, the geometric transformations can be divided into two classes, global transformations and free-form transformations, which will be reviewed in the later section. Furthermore, each kind of geometric transformations not only determine the scope of image registration, but also correspondingly affect the registration performance. This is the second reason that geometric transformation estimation is the most essential step.

- **Step 4. Image re-sampling and generation:** After the sensed image aligning with the reference image via a geometric transformation, pixel values in non-integer coordinates must be computed by an appropriate interpolation technique. The choice of re-sampling technique depends on the trade-off between the demanded accuracy of the interpolation and the computational complexity. Nearest-neighbour or bilinear interpolation are sufficient in most cases. However, some applications demand better methods.

1.1.3 Nature of Geometric Transformations

An appropriate geometric transformation function should be chosen according to a *priori* knowledge about the image acquisition process and expected image degradations. If this priory information is unavailable, the geometric transformation should be flexible enough to handle possible degradations which might appear. Additionally, the complexity of geometric transformations is usually indicated by ***degrees of freedom (dof)*** [1]

Geometric transformation is often grouped into two categories: global transformation and free-form transformation.

- **Global Transformation**

  Global transformation could be recognized as the most fundamental of the geometric transformation, which is able to be frequently defined by bivariate polynomials of low degrees. Global transformation is simple to specify, which contains several globe components, namely translation, scaling, shearing and rotation. Additionally, there are aslo some important specific globe geometric transformations, such as Euclidean transformation, similarity transformation and affine transformation, which are different combinations of some global compontents.

  - ***Global Components:***

    The translation is a two dimensional vector t that may be specified by giving its two coordinates t_x and t_y relative to a set of x and y

[1] ***degrees of freedom:*** degrees of freedom is equivalent to the number of independent parameters in geometric transformation, which is used in the formula to calculate these two values of each point.

Cartesian axes, as following:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{1.1}$$

Other global components (scaling, shearing and rotation) can be represented by matrices. In linear algebra, this geometric transformation is written as $T(\mathrm{x}) = M\mathrm{x}$, where M is called transformation matrix of $T(\cdot)$, for rotation by angle θ anticlockwise about the origin (denoted by R) is written as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{1.2}$$

For scaling (that is, enlarging or shrinking, denoting as S), we have:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{1.3}$$

For shear mapping (denoted by K), there are two possibilities, a shear parallel to the x axis and a share parallel to the y axis. The matrix form are:

$$\begin{array}{ll} \text{Parallel to axis } x: & \text{Parallel to axis } y: \\ \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & K \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, & \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ K & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \end{array} \tag{1.4}$$

- ***Euclidean Transformation:***

  This type global transformation contains rotation and translation. The globe transformation metric can be written as: $\mathrm{x}' = R\mathrm{x} + t$, where $\mathrm{x}' = \{x', y'\}^T$ denotes transformed coordinates of $\mathrm{x} = \{x, y\}^T$. This global transformation preserves size, straightness and parallelism of lines and angles between lines.

- ***Similarity Transformation:***

  This type global transformation consists of isotropic scaling, rotation and translation. Similar to Euclidean Transformation, the similarity transformation can be written as: $\mathrm{x}' = sR\mathrm{x} + t$, where s is a positive scalar, sometimes known as a dilation. This transformation preserves straightness, parallelism of lines and angles between lines.

- ***Affine Transformation:***

  Affine transformation contains all of global components, and can be written as: $\mathrm{x}' = Ax + t$, where transformation matrix $A = SRK$, a combination of scaling, rotation, and shearing. Additionally, there is no restriction on the elements a_{ij} of the matrix A.

  Affine transformation preserves the straightness and parallelism of lines, but it allows angles between line to change. It is an important global transformation when the image may have been skewed during acquisition, for example, when the CT gantry angle is incorrectly recorded.

  Moreover, affine transformation can also be described by means of homogenous transformation. In this representation, both A and t are folded into one 3×3 homogenous coordinate system M whose elements are defined as follows: $m_{ij} = a_{ij}$, where $(i = 1, 2,\ j = 1, 2)$; $m_{13} = t_x$ and $m_{23} = t_y$; rest of elements of M are 0, except $m_{33} = 1$. Then, homogenous transformation is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{1.5}$$

  The use of homogenous transformation does not produce any extra power or generality for global transformation, it does simplify notations.

- **Free-form Transformations**

  In addition to global transformations, free-form transformations are used to represent more complex local deformations between two images. Unlike global transformations, free-form transformations can hardly be determined by polynomials of low degrees.

  In this thesis, two types of free-form transformations will be discussed, namely rectangle free-form transformation and radial basis function (RBF) transformation.

  - ***Rectangle free-form transformation:***

    This method is called as rectangle free-form transformation, because an initialised rectangle mesh-grid is used to simulate the deformation

between two images via changing the locations of corresponding knots in the mesh grid. That is why this transformation is also widely named as mesh-grid transformation.

The simplest formula to describe principle of rectangle free-form transformation is a products of univariate polynomials in the components of rectangle knot coordinate $\mathrm{k}_{ij} = \{k_{ij}^x, k_{ij}^y\}^T$ [21] [31],

$$\mathrm{x}' = \sum_{ij}^{I,J} c_{ij} k_{ij}^x k_{ij}^y, \tag{1.6}$$

where $I = i + 1$ and $J = j + 1$, which consist of four nodes (k_{ij}, $\mathrm{k}_{i+1\,j}$, $\mathrm{k}_{i\,j+1}$ and $\mathrm{k}_{i+1\,j+1}$) of the i, j-th rectangle mesh. $c_{ij} = \{c_{ij}^x, c_{ij}^y\}^T$ is the two-element vector of coefficients for $i\,j$ term in this polynomial expression for any other non-node coordinate $\mathrm{x}' = \{x', y'\}^T$, located in the $i\,j$-th rectangle mesh.

Some modifications have been made in equation 1.6 [46] in order to construct an univariate polynomials, which is able to create more reliable transformation results than products of polynomials. The resulting transformations are defined by partitioning the space into a set of two-dimensional rectangles, each perpendicular to one of the Cartesian axes. Then, any other point x' within each $i\,j$-th rectangle can be presented as:

$$\mathrm{x}' = \sum_{ij}^{IJ} c_{ij} P^m(k_{ij}^x) P^m(k_{ij}^y), \tag{1.7}$$

where each $P^m(\cdot)$ is an univariate polynomial of degree m defined by over $i\,j$-th rectangle. By appropriate choice of the polynomial coefficients, it is possible to ensure that the polynomials join smoothly across the rectangle interfaces. That also means the m-th order univariate polynomial $P^m(\cdot)$ is continuously differentiable. Such special function defined piecewise by polynomials is called spline. The properties of spline have been extensively studied in approximation theory [44].

Commonly, m is set to 3 leading to a cubic spline, for which polynomial itself and its 1st and 2nd order derivatives are continuous. Additionally, the m-th order spline are often expressed with a set of convenient polynomial basis in same order, which is called B-spline [46]. Therefore, equation 1.7 is usually expressed with the expansion of the cubic

spline in terms of the B-spline as follows:

$$\mathrm{x}' = \sum_{i=i_0}^{I} \sum_{j=j_0}^{J} c_{ij} k_{ij}^{x} B_{i,3} \left(\frac{x - k_{ij}^{x}}{k_{Ij}^{x} - k_{i_0 j}^{x}} \right) k_{ij}^{y} B_{j3} \left(\frac{y - k_{ij}^{y}}{k_{iJ}^{y} - k_{ij_0}^{y}} \right), \quad (1.8)$$

where $k_{ij}^{x} B_{i3}(\cdot)$ and $k_{ij}^{y} B_{j3}(\cdot)$ are a single segment of the cubic B-spline formulation in x-direction and y-direction, respectively. (also see in [46])

The rectangle free-form transformation is simple and easy to implement, but its degrees of freedom in this entire transformation is rely on the number of knots of the initialised rectangle mesh-grid. In other words, the expression accuracy of a free-form transformation is decided by the size of each rectangle grid. This property will limit expression accuracy of a rectangle free-form transformation in describing of free-form deformations.

- ***Radial Basis Functions (RBF) transformations:***

  Another free-form transformation that has been widely used is Radial Basis Function (RBF) [11], with formulation as following:

$$\mathrm{x}' = A\mathrm{x} + t + \sum_{i}^{N} w_{i,j} \phi(\|\mathrm{x} - \mathrm{c_i}\|), \quad (1.9)$$

  where A is transformation matrix of affine transformation, t denotes translation transformation, $w_{i,j}$ is two-element vector coefficients of kernel function $\phi(r)$ ($r = \|\mathrm{x} - \mathrm{c_i}\|$), and c_i is control point. Unlike rectangle free-form transformation, these control points of RBF transformation are placed arbitrarily.

  The affine portion of equation 1.9 is a necessary part of RBF transformation. Without this component there may be no set of c_i that satisfies the equation at all N points to express some free-form transformation. With the affine part included, it is always possible, by means of enforcing a set of side conditions on the c_i to ensure that a solution exists for any arrangement of points.

  For form of RBF, there are a series of kernel functions can be chosen, namely Gaussian ($\exp(-\beta r^2)$ for some $\beta > 0$), multi-quadratic ($\sqrt{r^2 + \beta^2}$ for some $\beta > 0$), and some polynomial splines [18]. Whereas, in polynomial splines, there is a special case called thin-plate-spline with form of $r^2 \ln(r)$, which was originally called the surface spline [51].

This form was first proposed by Harder in 1972 [25] for designing aircraft wings, (their mathematical properties are explored by Duchon [32], who coined the term thin-plate-spline, and Meinguet [33]). The thin-plate splines were first employed to describe deformations within the two-dimensional plane by Goshtasby in 1988 [22], and first applied to image registration by Bookstein in 1989 [8]. But, why thin-plate-spline can be widely used as RBF to describe free-form transformation in 2-dimensional plane? This is because none of other kernel functions has compact support, which means a change in any control point has an affect on the transformation at all points. Whereas, thin-plate-spline kernel function with compact support have been employed to limit this effect to points in a neighborhood surrounding a given control point [37] [8].

1.2 Literature Review

Image registration is a crucial step in many computer vision tasks. In the last two decades, different algorithms have been presented. According to the similarity measure they used, these image registration algorithms can be divided into four categories, namely optic-flow based methods, information-theory based methods, correlation-like methods, and fourier methods [58]. In this thesis, we focus on the first two categories, *i.e.* optic-flow based methods and information-theory based methods.

1.2.1 Optic-flow based methods

The optic-flow based image registration methods can be traced back to Lucas and Kanade's work, published in 1988 [30]. This algorithm firstly employed optic-flow as motion force to map each point between the sensed image and the reference image. So far, numerous variants of Lucas-Kanade algorithm [26] [41] [55] [49] have been proposed.

Before addressing properties of two specific optic-flow based methods (Lucas-Kanade algorithm and Demons Algorithm), we shall firstly review the concept of optic-flow.

Optic-flow is used to describe the velocity field of each frame in a image sequence. For instance, in a 2D dimensional image sequence, a voxel at location (x, y, t) with intensity $I(x, y, t)$ will be moved by Δx, Δy in Δt time between the

two image frames, and the following optic flow equation is given as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

Then, assuming the movements are small, the image constraint at $I(x, y, t)$ is Taylor expanded, leading to

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \Delta x \frac{\partial I(\mathrm{x})}{\partial x} + \Delta y \frac{\partial I(\mathrm{x})}{\partial y} + \Delta t \frac{\partial I(\mathrm{x})}{\partial t}.$$

From these equations it follows that:

$$\Delta x \frac{\partial I(\mathrm{x})}{\partial x} + \Delta y \frac{\partial I(\mathrm{x})}{\partial y} + \Delta t \frac{\partial I(\mathrm{x})}{\partial t} = 0. \tag{1.10}$$

or

$$\frac{\Delta x}{\Delta t} \frac{\partial I(\mathrm{x})}{\partial x} + \frac{\Delta y}{\Delta t} \frac{\partial I(\mathrm{x})}{\partial y} + \frac{\Delta t}{\Delta t} \frac{\partial I(\mathrm{x})}{\partial t} = 0. \tag{1.11}$$

which results in

$$u \frac{\partial I(\mathrm{x})}{\partial x} + v \frac{\partial I(\mathrm{x})}{\partial y} + \frac{\partial I(\mathrm{x})}{\partial t} = 0. \tag{1.12}$$

In equation 1.12, $u = \frac{\Delta x}{\Delta t}$ and $v = \frac{\Delta y}{\Delta t}$ denote the velocity or optic-flow of $I(x, y, t)$ in the corresponding directions, respectively. Equation 1.12 can also be written in form of linear equations system as follows:

$$\left[\frac{\partial I(\mathrm{x})}{\partial x}, \frac{\partial I(\mathrm{x})}{\partial y}\right] \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{\partial I(\mathrm{x})}{\partial t}. \tag{1.13}$$

Obviously, this is a linear equation in two unknowns (*i.e.* u and v) and can not be solved. This obvious constraint of optic-flow is often known as the *aperture problem* and will be discussed with the other constraint in the following:

- ***Constraint 1***: In application of optic-flow algorithm, $\frac{\partial I}{\partial t}$ (*i.e.* the derivative of image at (x, y, t) with respected to time variance t) is hard to be actually calculated. Thus, the intensity differences between two image frames (*i.e.* $I(x, y, t + \Delta t) - I(x, y, t)$) could be used to approximately indicate $\frac{\partial I}{\partial t}$, when $I(x, y, t + \Delta t) - I(x, y, t)$ is small enough to support this approximation processing. In the image registration, due to this constraint, the intensity differences between two images can not be too large. Additionally, because of this constraints, optic-flow based image registration methods are only able to register images of the same modality.

- ***Constraint 2***: As mentioned before, this constraint is known as the *aperture problem* of optic-flow. The essence of this constraint is that one linear equation is insufficient to solve two unknown variables. Therefore, every optic-flow based image registration method must introduce additional conditions for estimating the optic-flow.

To address ***constraint 1***, Lucas-Kanade algorithm adopted iterative optimisation approach to estimate optic-flow between two images. Other approaches, like coarse-to-fine strategy, were presented in [9] to build up the pyramidal Lucas-Kanade algorithm.

To address ***constraint 2***, different optic-flow based image registration methods introduce different additional conditions. For example, Lucas-Kanade algorithm chooses smoothness condition, while Demons Algorithm chooses Maxwell's demons force as its additional condition.

- ***Lucas-Kanade Algorithm***

  As mentioned previously, the global smoothness condition is enforced to solve the actual optic-flow by Lucas-Kanade algorithm. The global smoothness condition is an additional constraint to ensure that every point in the sensed image shares the same specific global deformation. Because of the definition of this condition, Lucas- Kanade algorithm can not work well in the free-form image registration. For details, this algorithm will be discussed as a representative of global image registration approach in chapter 2.

- ***Demons Algorithm***

  Demons algorithm was originally invented by Thirion [49], in order to deal with homogenous free-form image registration. Different from Lucas-Kanade algorithm, Demons Algorithm adopted Maxwell's demons forces as an additional condition. The essence of Demons Algorithm is to apply Maxwell's demons force to the Horn and Schunck's optic-flow algorithm [26]. Then, This algorithm is to compute a regularised optic flow field through minimising an free energy function based on data term with Maxwell's demons force, encouraging smoothness of this optic-flow field.

  Thanks to employing Maxwell's demons force, Demons Algorithm shows an excellent performance in homogeneous free-form image registration, especially in medical image registration. In the application of medical image

registration, objects/images of interest (*e.g.* brain, liver, lung and other organs) often contain numerous homogenous (uniform) grey-level or colours, which are unable to be correctly aligned by ordinary optic-flow based image registration methods. Unfortunately, adoption of Maxwell's demons force will limit the alignment performance of this algorithm in texture-rich image registration. As a representative of free-form image registration, Demons Algorithm will be detailed in the next chapter.

1.2.2 Information-theory based methods

Information-theory based methods indicate a class of image registration approaches using information theory principle as the similarity measure, because information theory principle is a measure of statistical dependency between two data sets. Therefore, it is particularly suitable for registration of images from different modalities and has been widely used.

A famous information-theory based method is mutual information (MI) based image alignment, proposed by Viola and Wells in 1997 [50]. In this registration algorithm, mutual information is employed to measure correlation between two images of different modality. MI is maximised by using stochastic gradient descent optimisation method.

Similar to MI, cross-cumulative residual entropy (CCRE) [52] is a similarity measure based on cross-entropy [56] and will be applied by us in order to establish a multi-modality image registration in chapter 5.

1.3 Thesis Contribution

In this thesis, we study several non-rigid image registration algorithms and discuss the existing over-fitting and under-fitting problems in these algorithms. Then, we propose three improved non-rigid image registration algorithms to prevent the over-fitting and under-fitting problems, namely two-phase probabilistic second-order Demons Algorithm, learning varying dimension RBF, and learning varying dimension RBF with CCRE. There are three contributions presented for the three algorithms, respectively.

- **Contribution 1**: Through reviewing existing probabilistic image registration algorithms, we analyse the causes of the over-fitting and under-fitting problems for global and free-form image registration, respectively. For the

global image registration, ill-poseness significantly contributes to the over-fitting or under-fitting. In contrast, for the free-form image registration, low expression power of free-form transformations is the other factor that leads to the over-fitting and under-fitting problems, because the free-form transformation is too complicated for representation.

To eliminate effects of these two factors and prevent the over-fitting and under-fitting problems, we propose two-phase probabilistic second-order Demons Algorithm, which employs a divide-and-conquer strategy to resolve the ill-poseness and representative limitation, respectively. In the first phase, we utilize the probabilistic framework in warping parameter optimisation to overcome the ill-poseness. Then, in the second phase, another point-to-point matching process is adopted to correct errors of current rectangle free-form transformation. Through these two phases, we not only respectively resolve two factors for the over-fitting and under-fitting, but also increase the registration accuracy and make Demons Algorithm efficient with texture-rich image registration.

- **Contribution 2**: After proposing two-phase probabilistic second-order Demons Algorithm, we analyse two factors (*i.e.* ill-poseness and representative limitation) for the over-fitting and under-fitting. These two factors can be resolved by integrating a probabilistic framework and another point-to-point matching process. Yet, a generic and accurate free-form transformation model needs to be proposed to overcome the representative limitation of free-form transformation rather than adding a post-error-correction step.

  We study the RBF transformation and discuss advantages and disadvantages of this free-form transformation model. To perfectly present a free-form transformation, the control point selection becomes a crucial and challenging task in a specific RBF transformation establishment. This task contains the determination of the number and the locations of control points. Thus, we propose a new learning varying dimension RBF method to fulfil the control point selection, and then adaptively establish free-form transformations. To achieve this, Reversible Jump Markov Chain Monte Carlo (RJMCMC) [42] [23] is adopted to seek the optimal number and appropriate locations of these control points from a latent control points library, while warping parameters are estimated by Bayesian framework.

  To sum up, through establishing a learning varying dimension RBF, we

propose a synthesised algorithm to make an arithmetic representation of free-form transformation and estimate corresponding warping parameters at the same time.

- **Contribution 3**: Multi-modality image registration approach has been widely used in many application domains, especially in medical image processing. This is because integrating different functional images obtained from different modality can provide significant diagnostic information, assisting doctors to formulate timely and effective treatment.

  However, most recent research papers focused on analysis of similarity measurement, rather than establishing a complete multi-modality free-form image registration algorithm. Therefore, we extend our learning varying dimension RBF from single-modality to multi-modality domain by employing Cross-Cumulative Residual Entropy (CCRE) [52] [40] [3] as the similarity measure. Finally, we propose a multi-modality image registration algorithm for free-form transformations.

1.4 Thesis Structure

The rest parts of this thesis are organised as follows. In chapter 2, three major classical image registration algorithms will be reviewed, namely Lucas-Kanade algorithm [30], Demons Algorithm [49] and MI based image alignment [50]. Meanwhile, advantages and disadvantages of these three methods will also be discussed. Chapter 3 will illustrate how to overcome over-fitting (under-fitting) problem in Demons Algorithm and expand its application into texture-rich domain by incorporating Bayesian approach and two-phase deformation strategy. Chapter 4 will cover the design and construction of a new free-form image registration algorithm, named learning varying dimensional RBF. Theoretical and experimental details of this algorithm will provid in chapter 4 as well. After that, a new similarity measure for multi-modality image registration, named Cross-Cumulative Residual Entropy (CCRE), will be introduced in the beginning of chapter 5. Simulations and experiments will be conducted for comparisons between CCRE and MI in this chapter, which is used to prove the efficiency of CCRE. In the end of chapter 5, a multi-modality free-form image registration, named CCRE based learning varying dimension RBF, will be proposed. Meanwhile, relevant experimental results of this algorithm will be presented to verify its efficiency. Finally, chapter 6 will conclude this thesis.

Chapter 2

Classical Image Registration Algorithms

Contents:

2.1 Introduction

In this chapter, we will review three classical image registration algorithms, namely Lucas-Kanade algorithm, Demons Algorithm and Mutual Information (MI) based image alignment. Each of three algorithms is designed to solve global image registration, free-form image registration, and multi-modality image registration, respectively. Additionally, all of these three algorithms have brought significant impacts to image processing and computer vision.

The essence of Lucas-Kanade algorithm is the optic-flow based global image registration. Inspired by Lucas-Kanade algorithm, several variations of optic-flow based method [1] [47] have been developed, and these variations constitute optic-

flow based image registration category. That is why Lucas-Kanade algorithm has now become the foundation of optic-flow based image registration category and one of the the most widely applied techniques in computer vision. Applications of this type of methods have been reported in motion tracking, mosaic construction, medical image registration and face coding. Therefore, through reviewing Lucas-Kanade algorithm, we can better understand the mechanism and common limitations of the optic-flow based image registration category.

Demons Algorithm, which is an extension of Lucas-Kanade algorithm proposed by Thirion [49] , will be reviewed as a representative of free-form image registration method in this chapter. Through introducing Maxwell's demons force as an additional condition, this algorithm successfully achieves the homogenous region free-form registration, where this task has been frequently employed by several medical imaging techniques. Thus, this algorithm and its variants have been widely applied in medical image processing to assist diagnosis and treatment [24].

MI based image alignment, proposed by Viola and Wells [50], will be detailed as a representative of information-theory based image registration category. In Viola and Well's work, they firstly introduced mutual information as a similarity measure in order to achieve multi-modality image registration. Then, numerous approaches inspired by their idea have been proposed. Thus, as the foundation of information- theory based image registration category, the discussion of this MI based algorithm can assist us to understand mechanism and details of mutual information working in multi-modality image registration.

2.2 Lucas-Kanade Image Registration Algorithm

Lucas and Kanade (1981) [30] proposed a two-frame differential method for optic-flow estimation. Based on this work, Lucas-Kanade image registration algorithm was also created in that period. Briefly, Lucas-Kanade algorithm makes use of the spatial intensity gradient of the images to find a good match by using Newton-Raphson iteration. Before giving details of this algorithm, image registration task need to be formulated.

Image registration aims to align an image I with a template image T. $I(\mathrm{x})$ and $T(\mathrm{x})$ represent the intensity of these two images at coordinate x, respectively. Transformation (warping) function $W(\mathrm{x};\mathrm{p})$ maps a coordinate x to a new coordinate x$'$. Meanwhile, it is a function with respect to coordinate x and a set of

warping parameters p, where $\mathrm{p} = (p_1, \cdots, p_n)^T$ is a vector of parameters. For example, if translations are only considered in this situation, $W(\mathrm{x};\mathrm{p})$ will write as follows:

$$W(\mathrm{x};\mathrm{p}) = \begin{pmatrix} x + p_1 \\ y + p_2 \end{pmatrix}, \tag{2.1}$$

where parameter vector $\mathrm{p} = (p_1, p_2)^T$ indicates the displacement field. However, for undergoing a global transformation, affine transformation (mentioned in §1.1.3) is actually adopted in practice. Thus, warping function $W(\mathrm{x};\mathrm{p})$ becomes:

$$W(\mathrm{x};\mathrm{p}) = \begin{pmatrix} (1+p_1)\cdot x + p_2 \cdot y + p_3 \\ p_4 \cdot x + (1+p_5)\cdot y + p_6 \end{pmatrix} = \begin{pmatrix} 1+p_1, & p_2, & p_3 \\ p_4, & 1+p_5, & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{2.2}$$

where $\mathrm{p} = (p_1, p_2, \cdots, p_6)^T$.

After addressing image registration problem, the aim of Lucas-Kanade algorithm is to minimise sum of squared errors (SSE) between two images with respect to p. This procedure can be shown in the following:

$$\arg\min_{\mathrm{p}} \left\{ \sum_{\mathrm{x}} [I(W(\mathrm{x};\mathrm{p})) - T(\mathrm{x})]^2 \right\}, \tag{2.3}$$

where parameter vector p is updated with $\mathrm{p} \leftarrow \mathrm{p} + \Delta\mathrm{p}$. According to equation 2.3, Lucas-Kanade algorithm iteratively searches for the best p so that SSE between image I and template T is less than a pre-defined threshold ϵ_{E}, (*i.e.* $\sum_{\mathrm{x}} [I(W(\mathrm{x};\mathrm{p})) - T(\mathrm{x})]^2 \leq \epsilon_{\mathrm{E}}$).

The minimisation of equation 2.3 is done by performing a first order Taylor expansion at position x in the following:

$$\sum_{x} \left[I(W(\mathrm{x};\mathrm{p})) + \nabla I \frac{\partial W}{\partial \mathrm{p}} \Delta\mathrm{p} - T(\mathrm{x}) \right], \tag{2.4}$$

where ∇I is the gradient of image I, equation 2.4 starts with $\mathrm{p} = \vec{0}$, which means parameter vector $\vec{0}$ will maps pixels to themselves. Taking affine transformation as an example, Jacobian $\frac{\partial W}{\partial \mathrm{p}}$ will present as follows:

$$\frac{\partial W}{\partial \mathrm{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1}, \frac{\partial W_x}{\partial p_2}, \cdots, \frac{\partial W_x}{\partial p_6} \\ \frac{\partial W_y}{\partial p_1}, \frac{\partial W_y}{\partial p_2}, \cdots, \frac{\partial W_y}{\partial p_6} \end{pmatrix} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix}. \tag{2.5}$$

Differentiating and setting the result to zero gives:

$$\Delta\mathrm{p} = H^{-1} \sum_{\mathrm{x}} \left[\nabla I \frac{\partial W}{\partial \mathrm{p}} \right]^T [T(\mathrm{x}) - I(W(\mathrm{x};\mathrm{p}))], \tag{2.6}$$

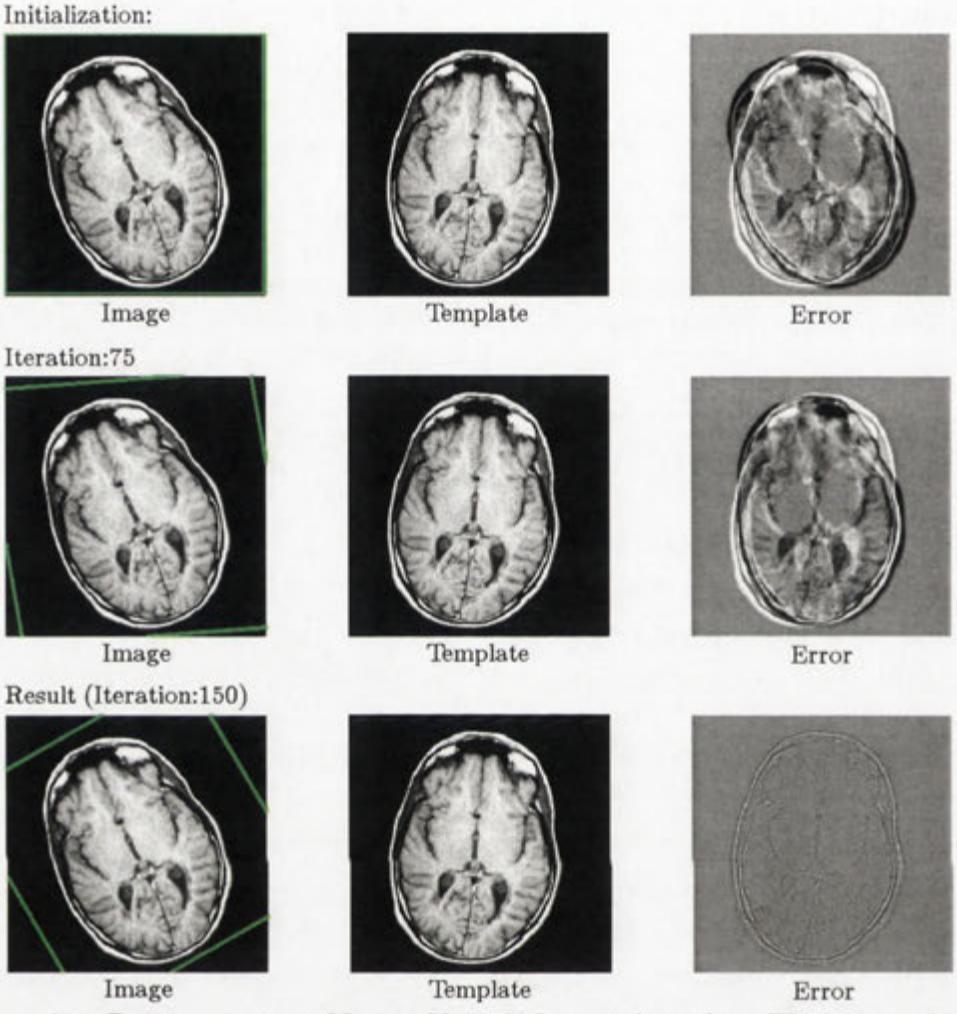


Figure 2.1: Demonstration of Lucas-Kanade Image algorithm: The 1st row is the initialisation of image I (Rotated I by 30 degree), where green-line illustrates warping status driven by the warping function $W(\mathrm{x};\mathrm{p})$, template T and error image E, obtained by $E(\mathrm{x}) = I(\mathrm{x}) - T(\mathrm{x}) + 128$; The 2nd row presents the alignment status of image I, template T, and error image, when Lucas-Kanade algorithm was running at the 75th step; The 3rd row shows the final registration result between image I and template T. In this case, the value of each pixel in error image E is obviously close to 128. The convergence came out, while algorithm was running at the 150th step.

where H is a $n \times n$ Gaussian-Newton approximation to the Hessian matrix:

$$H = \sum_{\mathrm{x}} \left[\nabla I \frac{\partial W}{\partial \mathrm{p}} \right]^T \left[\nabla I \frac{\partial W}{\partial \mathrm{p}} \right]. \tag{2.7}$$

Algorithm 2.1 Pseudocode of Lucas-Kanade Image Registration

Initialisation:

1. Define variance $\mathrm{p} = \vec{0}$
2. Calculate ∇I, the gradient of image I

while $(\| \sum_{\mathrm{x}} [I(W(\mathrm{x};\mathrm{p}+\Delta\mathrm{p})) - T(\mathrm{x})] \| \leq \epsilon_{\mathrm{E}})$ **do**

(1). Warp original image I with $W(\mathrm{x};\mathrm{p})$ to generate $I(W(\mathrm{x};\mathrm{p}))$

(2). Warp the gradient ∇I with $W(\mathrm{x};\mathrm{p})$

(3). Computer error image $[T(\mathrm{x}) - I(W(\mathrm{x};\mathrm{p}))]$

(4). Evaluate the Jacobian $\frac{\partial W}{\partial \mathrm{p}}$ at $(\mathrm{x};\mathrm{p})$

(5). Compute the steepest descent images $\nabla I \frac{\partial W}{\partial \mathrm{p}}$

(6). Calculate the Hessian matrix using equation 2.7

(7). Compute $\sum_{\mathrm{x}} \left[\nabla I \frac{\partial W}{\partial \mathrm{p}}\right]^T [T(\mathrm{x}) - I(W(\mathrm{x};\mathrm{p}))]$

(8). Compute $\Delta\mathrm{p}$ using equation 2.6

(9). Update warping parameters $\mathrm{p} \leftarrow \mathrm{p} + \Delta\mathrm{p}$

end while

Equations 2.6 and 2.7 express that parameters' update $\Delta\mathrm{p}$ are the steepest descent of parameters vector p multiplied by the inverse of Hessian matrix H. Typically, the criterion of convergence is SSE between between image I and template T need to be below a user specified threshold ϵ_{E}. However, for Lucas-Kanade algorithm, ∇I (the gradient of image I) is required to be computed with warping function $W(\mathrm{x};\mathrm{p})$ at every iteration of optimisation. Meanwhile, Jacobian matrix $\frac{\partial W}{\partial \mathrm{p}}$ must be calculated at each iteration too.

2.2.1 Inverse Compositional Algorithm

To accelerate the convergence of Lucas-Kanade algorithm, Baker and Matthews *et.al* (2002) [4] proposed a new strategy, named inverse compositional. This strategy provides a efficient optimisation formulation, which has become a popular method for image alignment and extensively applied by some other papers [35] [36].

The key of idea is to switch roles between image I and template T in equation 2.3, the rewritten loss function presents in the following:

$$\arg\min_{\Delta\mathrm{p}} \sum_{x} [T(W(\mathrm{x};\Delta\mathrm{p})) - I(W(\mathrm{x};\mathrm{p}))]^2, \tag{2.8}$$

where roles between image I and template T are reversed. Accordingly, the

Algorithm 2.2 Pseudocode of Inverse Compositional Algorithm

Initialisation:

1.Define variance $\mathrm{p} = \vec{0}$

2.Calculate ∇T, the gradient of image T

3.Evaluate the Jacobian $\frac{\partial W}{\partial \mathrm{p}}$ at $(\mathrm{x}; 0)$

4.Compute the steepest descent images $\nabla T \frac{\partial W}{\partial \mathrm{p}}$

5.Calculate the Hessian matrix using Equation 2.15

while $(\| \sum_{\mathrm{x}} [T(W(\mathrm{x}; \Delta\mathrm{p})) - I(W(\mathrm{x}; \mathrm{p}))] \| \leq \epsilon_{\mathrm{E}})$ **do**

(1). Warp original image I with $W(\mathrm{x}; \mathrm{p})$ to generate image $I(W(\mathrm{x}; \mathrm{p}))$.

(2). Computer error image $[I(W(\mathrm{x}; \mathrm{p})) - T(\mathrm{x})]$

(3). Compute $\sum_{\mathrm{x}} \left[\nabla T \frac{\partial W}{\partial \mathrm{p}}\right]^T [I(W(\mathrm{x}; \mathrm{p})) - T(\mathrm{x})]$

(4). Compute $\Delta\mathrm{p}$ using Equation 2.14

(5). Update $W(\mathrm{x}; \mathrm{p}) \leftarrow W(\mathrm{x}; \mathrm{p}) \circ W(\mathrm{x}; \Delta\mathrm{p})$

end while

update of warping function with $\Delta\mathrm{p}$ is transformed as follows:

$$W(\mathrm{x}; \mathrm{p})_{new} = W(\mathrm{x}; \mathrm{p})_{previous} \circ W(\mathrm{x}; \Delta\mathrm{p})^{-1}. \tag{2.9}$$

Inverse compositional algorithm iteratively solves an incremental warp $W(\mathrm{x}; \Delta\mathrm{p})$ rather than an additive update to parameters $\Delta\mathrm{p}$:.

$$W(\mathrm{x}; \mathrm{p}) \circ W(\mathrm{x}; \Delta\mathrm{p})^{-1} = \begin{pmatrix} (1 + p_1) \cdot \Delta x + p_2 \cdot \Delta y + p_3 \\ p_4 \cdot \Delta x + (1 + p_5) \cdot \Delta y + p_6 \end{pmatrix}, \tag{2.10}$$

where Δx and Δy are estimated by inverse of warping function $W(\mathrm{x}; \Delta\mathrm{p})$. Take affine transformation in consideration, inverse of $\Delta\mathrm{p}$ (indicated as $\mathrm{i}\Delta\mathrm{p}$) is represented as follows:

$$\frac{1}{(1 + \Delta p_1) \cdot (1 + \Delta p_5) - p_2 \cdot \Delta p_4} \begin{pmatrix} -\Delta p_1 - \Delta p_1 \cdot \Delta p_5 + \Delta p_4 \cdot \Delta p_2 \\ -\Delta p_2 \\ -\Delta p_3 - \Delta p_5 \cdot \Delta p_3 + \Delta p_2 \cdot \Delta p_6 \\ -\Delta p_3 \\ -\Delta p_5 - \Delta p_1 \cdot \Delta p_5 + \Delta p_2 \cdot \Delta p_4 \\ -\Delta p_6 - \Delta p_1 \cdot \Delta p_6 + \Delta p_4 \cdot \Delta p_3 \end{pmatrix}. \tag{2.11}$$

Then Δx and Δy can be computed with $\mathrm{i}\Delta\mathrm{p}$:

$$\begin{aligned} \Delta x &= (1 + \mathrm{i}\Delta p_1) \cdot x + \mathrm{i}\Delta p_2 \cdot y + \mathrm{i}\Delta p_3 \\ \Delta y &= \mathrm{i}\Delta p_4 \cdot x + (1 + \mathrm{i}\Delta p_5) \cdot y + \mathrm{i}\Delta p_6 \end{aligned} \tag{2.12}$$

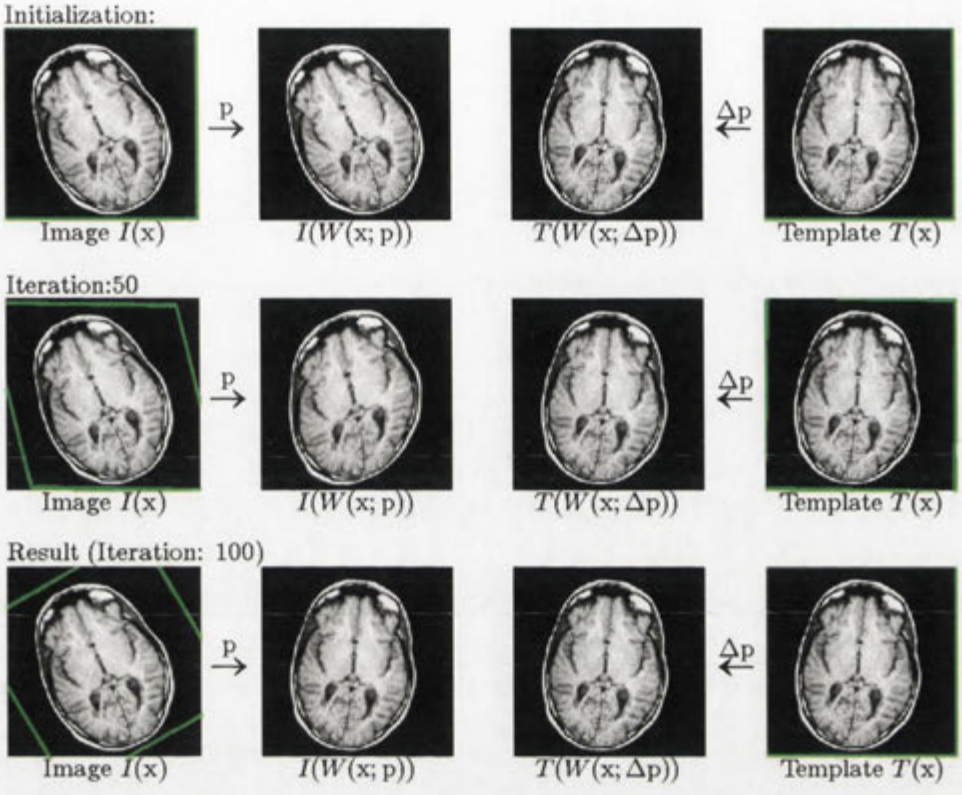


Figure 2.2: Demonstration of Inverse Compositional Algorithm: The 1st row is the initial situation of template T, warped template $T(W(\mathrm{x};\Delta\mathrm{p}))$ using $\Delta\mathrm{p}$, image I (Rotated I by 30 degree) and warped image $I(W(\mathrm{x};\mathrm{p}))$ using p, where green-line illustrates warping status with transformation functions $W(\mathrm{x};\mathrm{p})$ and $W(\mathrm{x};\Delta\mathrm{p})$; The 2-nd row is alignment state of Inverse Compositional Algorithm running at the 50th step; The 3rd row is the final registration result between image I and template T. The convergence came out, while algorithm is running at the 100th step.

The minimisation is also done by performing a first order Taylor expansion on equation 2.8 with respect to the warping parameter p:

$$\sum_{\mathrm{x}} \left[T(W(\mathrm{x};\vec{0})) + \nabla T \frac{\partial W}{\partial \mathrm{p}} \Delta\mathrm{p} - I(W(\mathrm{x};\mathrm{p})) \right]. \tag{2.13}$$

Assuming without loss of generality $W(\mathrm{x};0)$ is the identity warp. The solution of this least-squares problem is presented:

$$\Delta\mathrm{p} = H^{-1} \sum_{\mathrm{x}} \left[\nabla T \frac{\partial W}{\partial \mathrm{p}} \right]^{T} [I(W(\mathrm{x};\mathrm{p})) - T(\mathrm{x})] \tag{2.14}$$

where

$$H = \sum_{\mathrm{x}} \left[\nabla T \frac{\partial W}{\partial \mathrm{p}}\right]^T \left[\nabla T \frac{\partial W}{\partial \mathrm{p}}\right]. \tag{2.15}$$

Since nothing in the Hessian and Jacobian depend on $I(W(\mathrm{x};\mathrm{p}))$, they are constantly across iterations and able to be pre-computed. Due to these calculation cost reduction at every iteration, Inverse compositional algorithm significantly increases efficiency of the optimisation procedure.

2.3 Demons Algorithm

Demons Algorithm was proposed by Thirion *et. al.* [49] in 1998, and since then has become one of the most popular methods to solve free-form image registration problem. The essential idea of Demons Algorithm is inspired by Maxwell's demons force in the field of thermodynamics. This demons force has been incorporated into the conventional optic-flow algorithm (*e.g.* Lucas-Kanade [30] and Horn-Schunck [26]) as an additional condition in order to compute the displacement field for free-form transformation.

To understand the mechanism of Demons Algorithm, the displacement field of free-form transformation must be formulated at the beginning. Translation of each pixel is considered as the displacement field. According to equation 2.1, $W(\mathrm{x};\mathrm{p}(\mathrm{x}))$ denotes the displacement field between image I and template T, where $\mathrm{p}(\mathrm{x}) = (p_1(x), p_2(y))^T$ is a vector of the displacement parameters for pixel at coordinate x. Then we have $W(\mathrm{x};\mathrm{p}(\mathrm{x})) = (x + p_1(x), y + p_2(y))^T$.

Unfortunately, for optic-flow based algorithm, local displacement field $\Delta\mathrm{p}(\mathrm{x}) = (\Delta p_1(x),\ \Delta p_2(y))^T$ can not be directly calculated, due to *aperture problem* (*i.e. constraint 2* mentioned in §1.2.1). This *constraint 2* is recalled in the following:

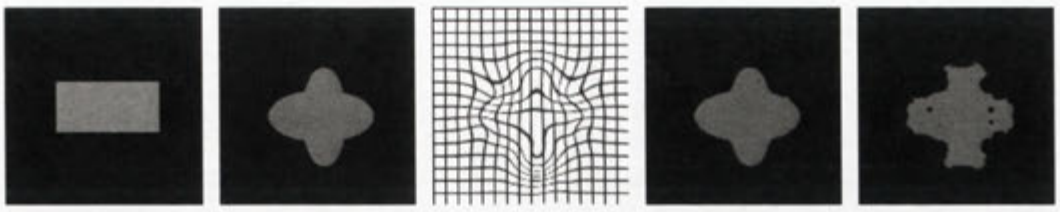
$$H(\mathrm{x})^{-1}\left[\nabla I(\mathrm{x})\right]^T\left[T(\mathrm{x}) - I(W(\mathrm{x};\mathrm{p}(\mathrm{x})))\right] = \mathrm{NaN} \quad \text{(Not A Number)}, \tag{2.16}$$

because

$$\det\left\{H(\mathrm{x})\right\} = \det\left\{\left[\nabla I(\mathrm{x})\right]^T\left[\nabla I(\mathrm{x})\right]\right\} = 0, \tag{2.17}$$

where $\det\{\cdot\}$ denotes the determinant of matrix. Apparently, in equation 2.16, $H(\mathrm{x})$ is a singular matrix, because its determinant equals to 0, which means $H(\mathrm{x})^{-1}$ does not exist. This is because one linear equation can not solve two unknown variables $\Delta p_1(x)$ and $\Delta p_2(y)$ of vector $\Delta\mathrm{p}(\mathrm{x})$.

To overcome *aperture problem*, instead of applying Hessian matrix $H(\mathrm{x})$, Demons Algorithm employs the norm of image gradient. Thus, the displacement



(a) Image (b) Template (c) Trans. Grid (d) Demons Alg. (e) Pyramid LK

Figure 2.3: Homogeneous region registration of "Rectangle-to-Star" sequence by 2nd-order Demons Algorithm and 3rd-order pyramid Lucas-Kanade: (a).Rectanale image, (b).Star image, (c).Regularisation displacement field, (d).2nd-order demons algorithm result, (e).3rd-order pyramid Lucas-Kanade algorithm result

field can be calculated as follows:

$$\Delta \mathrm{p}(\mathrm{x}) = \frac{[\nabla I(\mathrm{x})]^T [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]}{\|\nabla I(\mathrm{x})\|^2}. \tag{2.18}$$

However, equation 2.18 is unstable for small norm values of $\nabla I(\mathrm{x})$, which leads to infinite values for $\Delta p(\mathrm{x})$. Ideally, the expression should be close to zero for small $\nabla I(\mathrm{x})$. The solution is to add an extra term $\alpha [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]^2$ (α is usually set to 1) with the norm of image gradient.

$$\Delta \mathrm{p}(\mathrm{x}) = \frac{[\nabla I(\mathrm{x})]^T [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]}{\|\nabla I(\mathrm{x})\|^2 + \alpha [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]^2}. \tag{2.19}$$

Moreover, Demons Algorithm introduces Maxwell's demons force into equation 2.19 in order to maintain the smoothness of displacement field. The simplest Maxwell's demons force is to convolute a Gaussian smoothing kernel G_{σ^2} with current displacement field $\Delta \mathrm{p}(\mathrm{x})$.

$$\Delta \mathrm{p}(\mathrm{x}) = G_{\sigma^2} \otimes \left\{ \frac{[\nabla I(\mathrm{x})]^T [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]}{\|\nabla I(\mathrm{x})\|^2 + [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p}(\mathrm{x})))]^2} \right\}. \tag{2.20}$$

This procedure is also called regularisation, and it has been proven be efficient in the homogenous free-form image registration, such as brain CT, MRI, PET and etc.

In practice, displacement field is directly driven by some control points (*e.g.* rectangle mesh-grid nodes, contour-based features, or even classified shapes). Based on these different kinds of control points, Demons Algorithm can be divided into following three groups:

- **Demons I**: The geometric transformation relies on rectangle free-form transformation (in §1.2.1).

  In demons I, rectangle free-form transformation is applied as free-form transformation, where cubic B-splines is adopted to regularise displacement field and maintain its smoothness.

- **Demons II**: The geometric transformation depends on the contour features.

  In demons II, the transformation is controlled by contour features. Commonly, RBF (in §1.2.1) is applied to regularise displacement field.

- **Demons III**: The geometric transformation bases on segmentations.

  In demons III, free-form registration will be executed based on segmented images. The primary goal of Demons III is to seek the correspondences of different segmentations in these two images.

In this section, a synthetic free-form homogenous "Rectangle-to-Star" sequence is presented as a sample to demostrate the performance of Demons Algorithm, and this sequence will be also used in chapter 3 to examine performance of our method. Experimental result has been illustrated in Figure 2.3. Apparently, Demons Algorithm gives accurate result than that by pyramid Lucas-Kanade algorithm.

2.4 Mutual Information based Image Alignment

Multi-modality image registration is to register multiple images of same scene but obtained from different modalities. The aim of this registration is to integrate different information of same scene acquired by different source streams in order to gain more useful comprehensive information. That is why multi-modality image registration is widely required in several application domains, especially medical imaging.

For example, PET-CT is a multi-modality medical image technique, which combines a positron emission tomography (PET) and a x-ray computed tomography. Specifically, a functional image obtained by PET, depicting the spatial distribution of metabolic or biochemical activity in the body, can be precisely registered with an anatomic image obtained by CT scanning. Integrating different modality images can provide more useful and comprehensive information

than any images with single -modality. Therefore, PET-CT is capable of assisting doctors to locate tumor and examine pathological information at the same time.

Viola and Wells (1997) [50] proposed a multi-modulation image registration method based on Mutual information (MI). MI, originating from the information theory, is a measure of statistical dependency between two data sets. Therefore, it is particularly suitable for registering images of different modality. If $(x_1, x_2, \cdots, x_n)$ and $(y_1, y_2, \cdots, y_n)$ indicate two sets of N samples of two random variables x and y, the probability density functions of them are denoted as $p_{\mathrm{x}}(x)$, $p_{\mathrm{y}}(y)$ and given in the following:

$$p_{\mathrm{x}}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i), \quad p_{\mathrm{y}}(y) = \frac{1}{N}\sum_{i=1}^{N}\delta(y - y_i). \tag{2.21}$$

The joint probability density of these two random variables, indicated by $p(x, y)$, is presented as follows:

$$p(x, y) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i)\delta(y - y_i). \tag{2.22}$$

After defining the probability density and the joint probability density, MI can be calculated as follows:

$$\mathrm{E(x, y)} = \sum_{x\in \mathrm{x}}\sum_{y\in \mathrm{y}} p(x, y)\log_2\left(\frac{p(x, y)}{p_{\mathrm{x}}(x)p_{\mathrm{y}}(y)}\right). \tag{2.23}$$

Because MI is a quantity that is able to measure the mutual dependence of two variables x and y, larger value of MI means more significant correlation of these two random variables. In other words, smaller value of MI indicates these two random variables are more independent. If x and y are independent, so $p(x, y) = p_{\mathrm{x}}(x)p_{\mathrm{y}}(y)$. Then we have:

$$\log_2\left(\frac{p(x, y)}{p_{\mathrm{x}}(x)p_{\mathrm{y}}(y)}\right) = \log_2 1 = 0.$$

Because of above properties of mutual information, Viola *et.al.* employed it to measure the similarity between image I and template $\tilde{T}$ (I and $\tilde{T}$ are obtained in different modality imaging systems). Then, the goal of multi-modality image registration becomes maximising the MI between two images I and $\tilde{T}$ with Newton-Raphson gradient decent approach.

In probability density function equation 2.21, joint probability density function equation 2.22 and MI entropy equation 2.23, impulse function $\delta(\xi)$ is applied

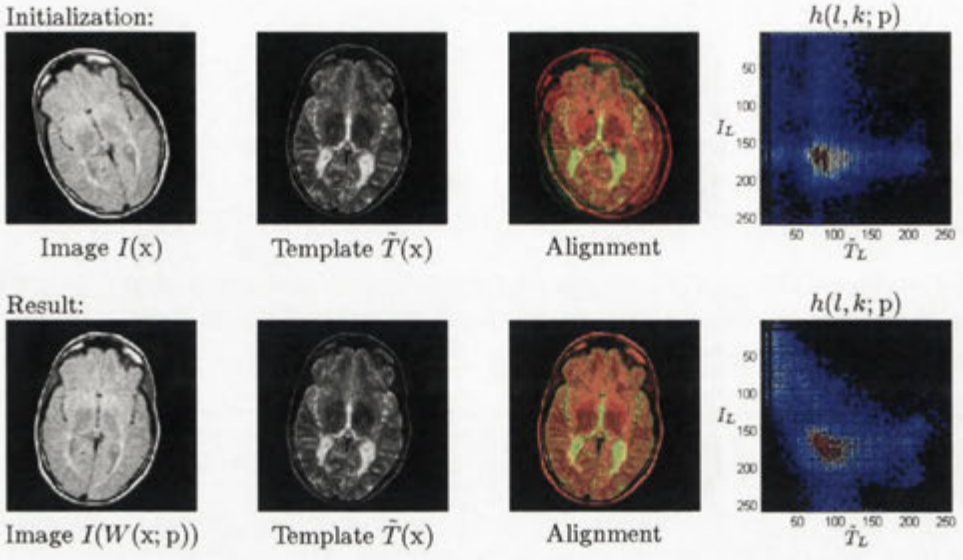


Figure 2.4: Demonstration of Image Alignment with Mutual Information: The 1st row is the initial situation of image $I(\mathrm{x})$ (rotated with 30 degree), the different modality template $\tilde{T}(\mathrm{x})$, the alignment statue and the joint histogram between image $I(\mathrm{x})$ and the template $\tilde{T}(\mathrm{x})$ (y and x coordinates denote discrete sets of intensities associated to image I and template $\hat{T}$, respectively) Mutual Information between them is: 0.7929bit; The 2nd row presents the transformed image $I(W(\mathrm{x};\mathrm{p}))$, the different modality template $\tilde{T}(\mathrm{x})$, the alignment statue and the joint histogram between aligned image $I(W(\mathrm{x};\mathrm{p}))$ and template $\tilde{T}(\mathrm{x})$. Mutual Information between them is: 1.3086bit.

to denote counts of sample in a random variable. According to discrete property of impulse function, MI formulation is transformed to be a discrete function. In other words, discontinuous MI formulation is non-differentiable. That is why Parzen Window is required in MI estimation.

2.4.1 Parzen Window

In statistics, Parzen window method is a non-parametric way of estimating probability density function of a random variable. If $p_x \sim\{x_1, x_2, \cdots, x_N\}$ and $p_y \sim\{y_1, y_2, \cdots, y_N\}$ are two sets of independent and identically-distributed samples of two random variables X and Y, respectively. Then the kernel density approximation

of their probability density functions are

$$\tilde{p}_{\mathrm{x}}(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{\omega((x-x_i)/\varepsilon_{\mathrm{x}})}{\varepsilon_{\mathrm{x}}}, \quad \tilde{p}_{\mathrm{y}}(y) = \frac{1}{N}\sum_{i=1}^{N}\frac{\omega((y-y_i)/\varepsilon_{\mathrm{y}})}{\varepsilon_{\mathrm{y}}}, \tag{2.24}$$

where $\omega(\xi)$ is some continuous kernel with unit integral ($\int_{-\infty}^{\infty}\omega(\xi)d\xi = 1$), ε_{x} and ε_{y} are smoothing parameters so called the bandwidth. They are strictly positive scaling factor controling the width of kernel $\omega(\xi)$. Quite often $\omega(\xi)$ is taken to be a B-spline kernel or a standard Gaussian kernel.

Image I and template $\tilde{T}$ are defined on a continuous domain x. $\mathrm{p} = (p_1, p_2, \cdots, p_n)$ indicates associated parameters of a geometric transformation (considering the affine transformation in this section,*i.e.* $n = 6$). Let I_L and $\tilde{T}_L$ be discrete sets of intensities associated with image I and template $\tilde{T}$. $\omega(\xi)$ denotes a separable kernel. Then, the kernel density approximation of joint histogram is given in the following:

$$h(l,k;\mathrm{p}) = \frac{1}{\varepsilon_I\varepsilon_T}\sum_{\mathrm{x}}\omega\left(\frac{l - I(W(\mathrm{x};\mathrm{p}))}{\varepsilon_I}\right)\omega\left(\frac{k-\tilde{T}(\mathrm{x})}{\varepsilon_T}\right), \tag{2.25}$$

where $l \in I_L$ and $k \in \tilde{T}_L$. ε_I and ε_T are width of $\omega(\xi)$ of image I and the template $\tilde{T}$, respectively. This joint histogram is proportional to joint probability density function p_h given by:

$$p_h(l,k;\mathrm{p}) = \alpha(\mathrm{p})h(l,k;\mathrm{p}), \tag{2.26}$$

where normalisation factor $\alpha(\mathrm{p})$:

$$\alpha(\mathrm{p}) = \frac{1}{\sum_{l\in I_L}\sum_{k\in\tilde{T}_L}h(l,k;\mathrm{p})}. \tag{2.27}$$

This normalisation factor takes up the role of factor $1/N$ in equation 2.22. Then, the marginal discrete probabilities and the histograms are given by

$$p_I(l;\mathrm{p}) = \alpha(\mathrm{p})h_I(l;\mathrm{p}) = \sum_{k\in\tilde{T}_L}p_h(l,k;\mathrm{p}), \tag{2.28}$$

$$p_{\tilde{T}}(k;\mathrm{p}) = \alpha(\mathrm{p})h_{\tilde{T}}(k;\mathrm{p}) = \sum_{l\in I_L}p_h(l,k;\mathrm{p}). \tag{2.29}$$

Consequently, MI between the warped image $I(W(\mathrm{x};\mathrm{p}))$ and template $\tilde{T}(\mathrm{x})$ is:

$$\mathrm{E}(I_L,\tilde{T}_L) = \sum_{l\in I_L}\sum_{k\in\tilde{T}_L}p_h(l,k;\mathrm{p})\log_2\left(\frac{p_h(l,k;\mathrm{p})}{p_I(l;\mathrm{p})p_{\tilde{T}}(k;\mathrm{p})}\right). \tag{2.30}$$

Warped image $I(W(\mathrm{x};\mathrm{p}))$ is correctly aligned with template T with parameter P, when $MI(I_L, \tilde{T}_L)$ is maximal. Because B-spline functions $\beta^n(x)$ have many good and interesting properties, it has been extensively chosen as the kernel function. Three-order B-spline $\beta^3(x)$ is selected as the kernel function of image I's histogram statistic, zero-order B-spline function $\beta^0(x)$ is employed to be the kernel function of template $\tilde{T}$'s histogram statistic. Then, joint probability density function (*i.e.* equation 2.25) between the warped image $I(W(\mathrm{x};\mathrm{p}))$ and template $\tilde{T}(\mathrm{x})$ can be re-written as:

$$p_h(l,k;\mathrm{p}) = \frac{\alpha(\mathrm{p})}{\varepsilon_I \varepsilon_T} \sum_{\mathrm{x}} \beta^3\left(\frac{l - I(W(\mathrm{x};\mathrm{p}))}{\varepsilon_I}\right) \beta^0\left(\frac{k - \tilde{T}(\mathrm{x})}{\varepsilon_T}\right). \tag{2.31}$$

2.4.2 Gradient of Mutual Information

After establishing MI function $\mathrm{E}(I_L, \tilde{T}_L)$ (*i.e.* equation 2.30), associated gradient $\nabla\mathrm{E}$ is provided as follows:

$$\nabla\mathrm{E} = \left[\frac{\partial \mathrm{E}}{\partial p_1}, \frac{\partial \mathrm{E}}{\partial p_2}, \cdots, \frac{\partial \mathrm{E}}{\partial p_n}\right]. \tag{2.32}$$

In general, a component of $\nabla\mathrm{E}$ is given by:

$$\begin{aligned}
\frac{\partial \mathrm{E}}{\partial \mathrm{p}} =& \alpha^2(\mathrm{p}) \sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial h(l,k;\mathrm{p})}{\alpha(\mathrm{p}) \partial \mathrm{p}} \log_2\left(\frac{e \cdot h(l,k;\mathrm{p})}{\alpha(\mathrm{p}) h_I(l;\mathrm{p}) h_{\tilde{T}}(k;\mathrm{p})}\right) \\
&+ h(l,k;\mathrm{p}) \log_2\left(\frac{e \cdot \alpha(\mathrm{p}) h_I(l;\mathrm{p}) h_{\tilde{T}}(k;\mathrm{p})}{h(l,k;\mathrm{p})}\right) \cdot \sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial h(l,k;\mathrm{p})}{\partial \mathrm{p}} \\
&- \frac{1}{\log_e(2)} \cdot \left(\frac{\partial h_I(l;\mathrm{p})}{\partial \mathrm{p}} h_{\tilde{T}}(k;\mathrm{p}) + \frac{\partial h_{\tilde{T}}(k;\mathrm{p})}{\partial \mathrm{p}} h_I(l;\mathrm{p})\right) \\
&\cdot \frac{h(l,k;\mathrm{p})}{\alpha(\mathrm{p}) h_I(l;\mathrm{p}) h_{\tilde{T}}(k;\mathrm{p})},
\end{aligned} \tag{2.33}$$

where e is the exponential constant. Above expression can be simplified, because $h_{\tilde{T}}$ and $\alpha(\mathrm{p})$ do not depend on p in this formulation of MI. Then, through selecting a 3rd order B-spline as the kernel, we have:

$$\frac{\partial \mathrm{E}}{\partial \mathrm{p}} = \sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial p_h(l,k;\mathrm{p})}{\partial \mathrm{p}} \log_2\left(\frac{p_h(l,k;\mathrm{p})}{p_I(l;\mathrm{p})}\right). \tag{2.34}$$

According to property of B-spline function, the gradient of this 3rd-order B-spline is given by:

$$\frac{\partial \beta^3(x)}{\partial x} = \beta^2(x + 1/2) - \beta^2(x - 1/2).$$

The gradient of joint probability density distribution can be represented:

$$\frac{\partial p_h(l,k;\mathrm{p})}{\partial \mathrm{p}} = \frac{\alpha(\mathrm{p})}{\varepsilon_I^2 \varepsilon_T} \sum_{\mathrm{x}} \beta^0 \left(\frac{k - \tilde{T}(\mathrm{x})}{\varepsilon_T} \right) \times \left[\beta^2 \left(\frac{l - I(W(\mathrm{x};\mathrm{p}))}{\varepsilon_I} - \frac{1}{2} \right) - \beta^2 \left(\frac{l - I(W(\mathrm{x};\mathrm{p}))}{\varepsilon_I} + \frac{1}{2} \right) \right] \nabla I \frac{\partial W}{\partial \mathrm{p}}. \tag{2.35}$$

In this formulation, $\frac{\partial W}{\partial \mathrm{p}}$ is the Jacobian matrix of warping function W as same as in equation 2.5.

2.4.3 Hessian matrix of Mutual Information

Let us define the matrix of the second-order gradient of $\mathrm{E}(I_L, \tilde{T}_L)$ as its hessian matrix $\nabla^2 \mathrm{E}$:

$$\nabla^2 \mathrm{E} = \begin{pmatrix} \frac{\partial^2 \mathrm{E}}{\partial p_1 \partial p_1}, & \frac{\partial^2 \mathrm{E}}{\partial p_1 \partial p_2}, & \cdots \\ \frac{\partial^2 \mathrm{E}}{\partial p_2 \partial p_1}, & \frac{\partial^2 \mathrm{E}}{\partial p_2 \partial p_2}, & \cdots \\ \vdots \ , & \vdots \ , & \ddots \end{pmatrix}. \tag{2.36}$$

Based on the same assumption in last section, the component of Hessian matrix can be determined by

$$\begin{aligned} \frac{\partial^2 \mathrm{E}}{\partial p_1 \partial p_2} = & - \left[\sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial^2 p_h(l,k;\mathrm{p})}{\partial p_1 \partial p_2} \log_2 \left(\frac{p_h(l,k;\mathrm{p})}{p_I(l)} \right) \right] \\ & + \frac{1}{\log_e(2)} \left[\sum_{l \in I_L} \frac{\partial p_I(l;\mathrm{p})}{\partial p_1} \frac{\partial p_I(l;\mathrm{p})}{\partial p_2} \frac{1}{p_I(l;\mathrm{p})} \right] \\ & - \frac{1}{\log_e(2)} \left[\sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial p_h(l,k;\mathrm{p})}{\partial p_1} \frac{\partial p_h(l,k;\mathrm{p})}{\partial p_2} \frac{1}{p_h(l,k;\mathrm{p})} \right]. \end{aligned} \tag{2.37}$$

The first term in equation 2.37 depends on the second-order gradient of the joint probability when a pair of registration parameters varies jointly. This term can be ignored when assuming p_h with respect to p. Then, when the first term in equation 2.37 is vanished, the remaining terms do still contribute, and Hessian will not globally vanish at idea registration. Finally, the simplified form of each component in hessian matrix is determined:

$$\begin{aligned} \frac{\partial^2 \mathrm{E}}{\partial p_1 \partial p_2} \approx & \frac{1}{\log_e(2)} \left[\sum_{l \in I_L} \frac{\partial p_I(l;\mathrm{p})}{\partial p_1} \frac{\partial p_I(l;\mathrm{p})}{\partial p_2} \frac{1}{p_I(l;\mathrm{p})} \right] \\ & - \frac{1}{\log_e(2)} \left[\sum_{l \in I_L} \sum_{k \in \tilde{T}_L} \frac{\partial p_h(l,k;\mathrm{p})}{\partial p_1} \frac{\partial p_h(l,k;\mathrm{p})}{\partial p_2} \frac{1}{p_h(l,k;\mathrm{p})} \right]. \end{aligned} \tag{2.38}$$

2.4.4 Gradient Decent Optimisation

The steepest gradient decent is a minimisation algorithm that can be described by:

$$\mathrm{p}^{k+1} = \mathrm{p}^{k} - \lambda \nabla \mathrm{E}^{k}. \tag{2.39}$$

The converge process may be very slow. A key problem is the determination of the appropriate scaling diagonal matrix λ.

The Quasi-Newton method can be described by

$$\mathrm{p}^{k+1} = \mathrm{p}^{k} - \left(\nabla^2 \mathrm{E}^{k}\right)^{-1} \nabla \mathrm{E}^{k}. \tag{2.40}$$

In image alignment by mutual information method, this optimisation is considered to be applied to search the maximal value of mutual information between image I and template $\tilde{T}$ with respect to p^{k}.

2.5 Discussion and Conclusion

In this chapter, three classical image registration methods have been reviewed. Lucas-Kanade was introduced as a representative of global image registration approaches; Demons algorithm was studied as a representative of free-form image registration methods; MI based Image alignment was discussed as a representative of multi-modality image registration algorithms.

Lucas-Kanade algorithm is an early proposed numerical image registration framework. By incorporating smoothness condition, optic-flow computation is applied to execute points/features matching. Then Newton-Raphson optimisation is employed to iteratively search for the best warping parameters of geometric transformation. Although incorporation of global smoothness condition makes actual optic-flow computation be possible, this condition also limits accuracy of Lucas-Kanade algorithm for free-form image registration, especially homogenous region free-form registration.

Applying Maxwell's demons force as an additional condition, Demons Algorithm successfully tackles *constraint 2* of optic-flow algorithm. Additionally, incorporation of Maxwell's demons force also makes Demons Algorithm become a very popular free-form homogenous image registration algorithm. This is because Maxwell's demons force is achieved by adding a regularisation into points/features

matching procedure. This regularisation increases smoothness of the displacement field of image I. This regularisation has been proven be efficient for free-form homogenous image registration. However, in the meantime, it will also limit registration accuracy of Demons Algorithm in application of the texture-rich image registration.

Adopting MI as a similarity measure, MI based image alignment successfully achieves multi-modality image registration purpose, and has been already widely used in many different applications. Although MI theory successfully solves multi-modality image alignment problem, there are still two major limitations:

1. The computation cost of mutual information between image $I(W(\mathrm{x};\mathrm{p}))$ and template $\tilde{T}(\mathrm{x})$ at each iteration is too much expensive.

2. The change of mutual information with respect to warping parameters is not very smooth, because the density functions are not regular. Irregular situation of mutual information can cause the optimisation procedure trapped into local minima, which leads to mis-alignment.

Chapter 3

Improved probabilistic Demons Algorithm

Contents:

3.1 The need for probabilistic image registration

Classical image registration algorithms, such as Lucas-Kanade algorithm, inverse compositional algorithm and demons algorithm, are often suffering from unstable convergence, especially when the number of warping parameters is more than three or four. For example, Jianbo Shi *et. al.* noticed this instability of convergence when using affine transformation in image registration [45] [27].

Actually, this issue of unstable convergence is caused by ill-posed inverse problem, lay in warping parameters estimation. Take Lucas-Kanade algorithm as an example, warping parameter estimation can be represented in the following with the same notations used in chapter 2.

Because Lucas-Kanade algorithm is based on optic-flow, it is recalled here, which has been previously discussed in §1.2.1.

For a 2D+t dimensional case (3D), a voxel at location (x, y, t) with intensity $I(x, y, t)$ will be moved by Δx, Δy and Δt between two frames. This image constraint can be represented as follows:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

Assuming the movement is small, this image constraint at $I(x, y, t)$ with Taylor series can be developed to get

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \Delta x \frac{\partial I(\mathrm{x})}{\partial x} + \Delta y \frac{\partial I(\mathrm{x})}{\partial y} + \Delta t \frac{\partial I(\mathrm{x})}{\partial t}.$$

From these equations it follows that:

$$\Delta x \frac{\partial I(\mathrm{x})}{\partial x} + \Delta y \frac{\partial I(\mathrm{x})}{\partial y} + \Delta t \frac{\partial I(\mathrm{x})}{\partial t} = 0. \tag{3.1}$$

If equation 3.1 is divided by Δt, $u = \frac{\Delta x}{\Delta t}$ and $v = \frac{\Delta y}{\Delta t}$ indicate velocities (*i.e.* optic- flow) in directions X and Y. Then, equation 3.1 can be rewritten in form of linear equations:

$$\left[\frac{\partial I(\mathrm{x})}{\partial x}, \frac{\partial I(\mathrm{x})}{\partial y}\right] \cdot \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{\partial I(\mathrm{x})}{\partial t}. \tag{3.2}$$

Taking into account of affine transformation with associated warping function $W(\mathrm{x}; \mathrm{p})$, equation 3.2 will be represented as:

$$\left[\nabla I(\mathrm{x}) \frac{\partial W}{\partial p_1}, \nabla I(\mathrm{x}) \frac{\partial W}{\partial p_2}, \cdots, \nabla I(\mathrm{x}) \frac{\partial W}{\partial p_6}\right] \cdot \mathrm{p} = -\frac{\partial I(\mathrm{x})}{\partial t}, \tag{3.3}$$

where $\mathrm{p} = (p_1, p_2, \cdots, p_6)^T$.

Because $\frac{\partial I(\mathrm{x})}{\partial t}$ in equation 3.3 can hardly be computed in practice, $[I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t)]$ is actually used to approximate $\frac{\partial I(\mathrm{x})}{\partial t}$, especially when differences between these two frames are small enough to satisfy *constraint 1* of optic-flow algorithms (in §1.2.1).

Unfortunately, image $I(\mathrm{x})$ and template $T(\mathrm{x})$ usually do not satisfy *constraint 1*. To solve this problem, iterative optimisation has been adopted to seek warping

parameters $\Delta \mathrm{p} = (\Delta p_1, \Delta p_2, \cdots, \Delta p_6)^T$ stepwisely, rather than directly estimating $\mathrm{p} = (p_1, p_2, \cdots, p_6)^T$.

Additionally, due to the *aperture problem* (*c.f. constraint 2*), equation 3.3 is unsolvable. In other words, the *aperture problem* can be roughly understood by the fact that one linear equation is unable to simultaneously solve six-unknown-parameter vector $\Delta \mathrm{p}$, without additional conditions.

Thanks to the smoothness condition, parameters $\Delta \mathrm{p}$ of affine transformation are able to be solved by Lucas-Kanade algorithm as follows.

$$\Delta \mathrm{p} = -\left[\sum_{\mathrm{x}} H(\mathrm{x})\right]^{-1} \sum_{\mathrm{x}} \nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} \left[T(\mathrm{x}) - I(\mathrm{x})\right], \tag{3.4}$$

where

$$\begin{aligned} H(\mathrm{x}) &= \left[\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}}\right]^T \left[\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}}\right] \\ &= \begin{pmatrix} \nabla I(\mathrm{x})^2 \left(\frac{\partial W}{\partial p_1}\right)^2, \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_1} \frac{\partial W}{\partial p_2}, \cdots, \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_1} \frac{\partial W}{\partial p_6} \\ \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_2} \frac{\partial W}{\partial p_1}, \nabla I(\mathrm{x})^2 \left(\frac{\partial W}{\partial p_2}\right)^2, \cdots, \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_2} \frac{\partial W}{\partial p_6} \\ \vdots \quad , \quad \vdots \quad , \ddots , \quad \vdots \\ \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_6} \frac{\partial W}{\partial p_1}, \nabla I(\mathrm{x})^2 \frac{\partial W}{\partial p_6} \frac{\partial W}{\partial p_2}, \cdots, \nabla I(\mathrm{x})^2 \left(\frac{\partial W}{\partial p_6}\right)^2 \end{pmatrix}. \end{aligned} \tag{3.5}$$

In equation 3.5, the smoothness condition is actually an assumption that all pixels in image $I(\mathrm{x})$ must share a same global transformation, and this smoothness condition has been expressed in form of summation.

3.1.1 Ill-poseness in image registration

Although, the smoothness condition makes Lucas-Kanade algorithm be able to overcome the *aperture problem*, it also causes the ill-posed problem in the optic-flow based image registration algorithms.

The ill-poseness concept was introduced by Hadamard. He indicated that mathematical models of physical phenomena should have the properties that:

1. A solution exists;
2. The solution is unique;
3. The solution depends continuously on the data, in some reasonable topology.

Any mathematical model that does not satisfy the above three properties is called ill-poseness problem. Because inverse problems are often ill-posed, they are called as ill-poseness as an abbreviation in this thesis). To some extent, according to

these criterions, the *aperture problem* is also an ill-poseness. Whereas, in global image registration, ill-poseness usually represents in modus of a over-determined system of linear equations, which means the solution of equation 3.5 is not unique.

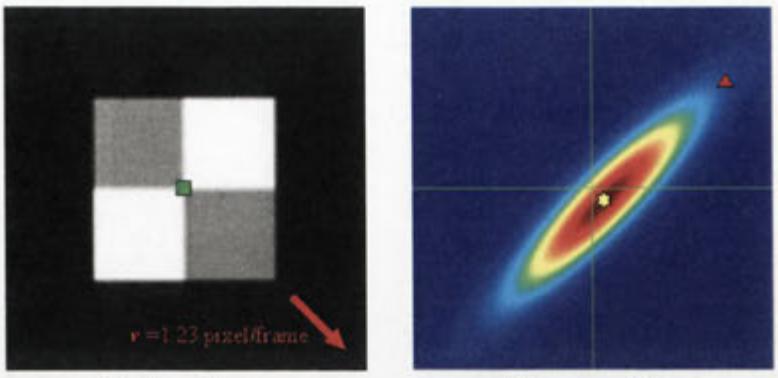
In practice, Lucas-Kanade algorithm applied an iterative optimizsation to estimate warping parameters. However, without extra constraints and with poor data, this iterative warping parameter estimation may yield some extremely unlike geometric transformations. In this respect, ill-poseness will cause unstable and unreliable convergence - inaccurate warping parameters. Then, these inaccurate warping parameters will certainly result the over-fitting and under-fitting, which severely limits the image registration performance.

3.2 Probabilistic Image Registration

Probabilistic image registration is established by incorporating a *priori* knowledge into the parameter estimation. One purpose of probabilistic image registration is to produce a smooth transition between fixed and fitted parameters. Because of ill-poseness, the estimation error for fitted parameters could be large and become unreliable. In this case, a *priori* knowledge will be adopted to conduct a reliable parameters estimation - a process known as regularisation.

Tikhonov regularisation [48] is the most commonly used regularisation method for solving ill-posed problems. Take an over-determined system of linear equations $M\mathrm{X} = \mathrm{b}$ as an example, linear least squares and seeking minima of residual $\|M\mathrm{X} - \mathrm{b}\|^2$ are the standard approaches to estimate unknown vector X, where M denotes the coefficient matrix, and $\|\cdot\|$ is the Euclidean norm. However, in the image registration, matrix M is ill-posed yielding a non-unique solution. In order to give preference to a reliable and stable solution, the regularisation term is inserted in the minimisation, and the residual becomes $\|M\mathrm{X} - \mathrm{b}\|^2 + \|\Gamma\mathrm{X}\|^2$, where Γ is called as Tikhonov matrix.

Although at first glance the choice of the solution to this regularised problem may look artificial, and indeed the matrix Γ seems rather arbitrary, the process can be justified from a Bayesian point of view. To overcome the ill-poseness problem, some additional assumptions must be incorporated in order to get a reliable and stable solution. Statistically, the assumption that taking X as a random variable with a multivariate normal distribution is commonly introduced as a *priori* knowledge. For simplicity, this independent and identically norm distribution is usually set as zero mean with standard deviation σ_{X}. Similarly,



(a) One Frame (b) Response System

Figure 3.1: (a): One frame of the synthetic drifting image sequence. The normal direction of motion in the synthetic drifting image sequence was down and to the right. The drifting speed in the norm direction was 1.23 pixel/frame. (b): This is the response system to evaluate accuracy of estimated norm velocities. The center of cross-line coincides with the ground truth norm velocity 1.23 pixel/frame. An planar interception of $p(\Delta \mathrm{p}|I(\mathrm{x}))$ is presented with mean value marked by a yellow hexagram. This yellow hexagram also denotes the estimated normal velocity, computed by probabilistic image registration (*i.e.* equation 3.9) over the small 3×3 green square patch near the center of image (a). The red upward-pointing triangle marks the computed normal velocity with original Lucas-Kanade algorithm (*i.e.* equation 3.5).

we take the errors in given data b to be also an independent normal distribution with zero mean and standard deviation σ_b. According to Bayesian theorem, the solution of Tikhonov regularisation, based on above assumptions, is the most reliable and stable solution given data b and the *priori* distribution of X. The Tikhonov matrix is then $\Gamma = \alpha \mathrm{I}$, where I is the identity matrix, and Tikhonov factor $\alpha = \sigma_b / \sigma_{\mathrm{X}}$. Finally, an explicit solution X is given by:

$$\mathrm{X} = \left(M^T M + \Gamma^T \Gamma\right)^{-1} M^T b. \tag{3.6}$$

After a brief introduction of Tikhonov regularisation, we will discuss how to adopt this Tikhonov regularisation to solve ill-poseness problems in Lucas-Kanade algorithm. In practice, the term of $[T(\mathrm{x}) - I(\mathrm{x})]$ may fail to satisfy *constraint 1* of optic-flow algorithms, because of changes in lighting or reflectance, or the presence of multiple motions. Additionally, in equation 3.3, there will be some errors in the derivative measurements of $\nabla I(\mathrm{x})$, due to camera and quantisation noise, aliasing, imprecision. Finally, estimation errors could yield in warping

parameters estimation, because of above two errors. To describe these errors, independent additive gaussian noise terms $n_1 \sim N(0, \sigma^2)$, $\mathrm{n}_2 \sim N(0, \Lambda_2)$ and $\mathrm{n}_3 \sim N(0, \Lambda_3)$ of dimensionality 1, 2 and n (the dof of geometric transformation, *e.g.* 6 for affine transformation) are applied to illustrate these types of uncertainty.

$$[\nabla I(\mathrm{x}) - \mathrm{n}_2] \frac{\partial W}{\partial \mathrm{p}} (\Delta \mathrm{p} - \mathrm{n}_3) = -[T(\mathrm{x}) - I(\mathrm{x})] + n_1. \tag{3.7}$$

Now, the goal is to compute an expression for the probability of the warping parameters $\Delta \mathrm{p}$ conditional on image $I(\mathrm{x})$ and template $T(\mathrm{x})$. For this purposes, a conditional probability based on image $I(\mathrm{x})$ and template $T(\mathrm{x})$ can be denoted with $p(\Delta \mathrm{p}|F)$, where $F = [T(\mathrm{x}) - I(\mathrm{x})]$. In order to write down the desired conditional probability, Bayesian rule has been switched the order of the arguments:

$$p(\Delta \mathrm{p}|F) = \frac{p(F|\Delta \mathrm{p})p(\Delta \mathrm{p})}{p(F)}. \tag{3.8}$$

The resulting distribution $p(\Delta \mathrm{p}|F)$ is gaussian, and the mean $\mu_{\Delta \mathrm{p}}$ and covariance matrix $\Lambda_{\Delta \mathrm{p}}$ can be derived using standard techniques:

$$\mu_{\Delta \mathrm{p}} = -\Lambda_{\Delta \mathrm{p}} \sum_{\mathrm{x}} \sigma_{\mathrm{x}}^{-2} \nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} [T(\mathrm{x}) - I(\mathrm{x})] \tag{3.9}$$

$$\Lambda_{\Delta \mathrm{p}} = \left[\sum_{\mathrm{x}} \sigma_{\mathrm{x}}^{-2} \left(\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} \right) \left(\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} \right)^{T} \right]^{-1}, \tag{3.10}$$

where

$$\sigma_{\mathrm{x}}^2 = \sigma^2 + \nabla I(\mathrm{x}) \Lambda_2 \nabla I(\mathrm{x})^T + \left[\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} \right] \Lambda_3 \left[\nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p}} \right]^T.$$

Then, the Maximum A Posteriori (MAP) estimate is simply the mean $\mu_{\Delta \mathrm{p}}$, since the distribution is gaussian.

Figure 3.1 illustrates the accuracy comparison of warping parameter estimation between probabilistic image registration and Lucas-Kanade algorithm at a single iteration. Figure 3.1-a is one frame of the synthetic drifting image sequence, where object is moving down and to the right with 1.23 pixel/frame through the sequence. Figure 3.1-b is the response system to evaluate accuracy of estimated drifting speeds produced by different algorithm, where the center of the cross-line coincides with the ground truth norm velocity (*i.e.* 1.23 pixel/frame). A planar interception of $p(\Delta \mathrm{p}|F)$ with mean value marked by a yellow hexagrams is presented in figure 3.1-b. Meanwhile, this yellow hexagram denotes the estimated

norm velocity over a green square patch computed by probabilistic image registration. Whereas, a red upward point triangle in figure 3.1-b is used to mark the norm velocity computed by Lucas-Kanade algorithm.

Apparently, the norm velocity yielded by probabilistic image registration is much closer to the ground truth than that computed by Lucas-Kanade algorithm. Through this evaluation experiment, probabilistic image registration is proven to significantly increases accuracy of stepwise warping parameters estimation at each iteration. Thanks to this improvement, probabilistic method can produce more reliable warping parameters.

3.2.1 Limitations of probabilistic image registration with Free-form Transformation

Free-form transformations have high degrees of freedom, which means the large number of warping parameters. Due to this high degrees of freedom, free-form transformations (*e.g.* rectangle free-form transformation and RBF transformation) have to be driven by sufficient scattered control points.

In 2D situation, each control point can provide two degrees of freedom in the directions of x and y. In fact, warping parameters of free-form transformation are used to generate motions of these control points, rather than directly expressing displacement of every pixel. After control points matching, the further point-to-point mapping processing would be executed via specific polynomials interpolation methods, *e.g.* cubic B-spline for rectangle free-form transformation and thin-plate-spline for RBF transformation.

In conclusion, a correct warping parameter estimation can generate an accurate measurement of displacement norms of control points. This accurate measurement is a necessary and sufficient condition for a well next point-to-point mapping processing, which is directly applied to create a dense free-form image registration.

In free-form image registration, warping parameter estimation errors are likely to be very large, because of ill-poseness in the complex over-determined linear system of equations. As described in previous section, the above errors caused by ill-poseness problem can be effectively corrected by probabilistic image registration (*i.e.* incorporating Tikhonov regularisation into the iterative warping parameter optimisation).

Unfortunately, in addition of errors yielded by ill-poseness, there is the other kind of errors in the warping parameter estimation of free-form transformation. In

free-form transformations, warping parameters do not directly control every pixel, but overmaster the displacement norms of scatted control points. Through these control points and corresponding polynomials interpolation, warping parameters can indirectly map every pixel between two images. However, these displacement norms of some control points have high probability of being fallaciously measured, because some control points may locate in featureless regions. Such kind of estimation errors can be interpreted as a limitation called "low expression power". This limitation can not be simply described by any zero-mean normal distribution. In other words, probabilistic image registration can do nothing to eliminate this limitation. Moreover, because of this errors, over-fitting and under-fitting will frequently occur and severely limit the performance.

In conclusion, by incorporating a *priori* knowledge into the iterative warping parameter estimation, probabilistic image registration can satisfactorily resolve ill-poseness problem. However, due to some incorrect measurements of displacement norms of control points, probabilistic method can not prevent the occurrence of over-fitting and under-fitting in free-form image registration.

3.3 Introduction to the Second-order Demons Algorithm

Before we establish two-phase probabilistic second-order demons algorithm, a brief introduction is required to the second-order Demons Algorithm.

3.3.1 Recall of Demons Algorithm

First of all, let us recall Demons Algorithm with the same denotation presented in chapter 2. In this section, Image $I(\mathrm{x})$ will deform to template $T(\mathrm{x})$ by a dense displacement field P. $W(\mathrm{x};\mathrm{p}(\mathrm{x}))$ denotes the warping function of a pixel at coordinate x in image $I(\mathrm{x})$. Similarly, $\mathrm{p}(\mathrm{x}) = (p_1(x), p_2(y))^T$ is two-element local displacement vector of pixel at coordinate x in image $I(\mathrm{x})$.

Thus the key task of Demons Algorithm is to seek an accurate dense displacement field $\mathrm{P} = \{\mathrm{p}(\mathrm{x})|\mathrm{x} \in \mathbf{R}\}$ that minimizes SSE between image $I(\mathrm{x})$ and template $T(\mathrm{x})$. Because of constraint 1 of optic-flow algorithms, in practice, a stepwise displacement field $\Delta\mathrm{P} = \{\Delta\mathrm{p}(\mathrm{x})|\mathrm{x} \in \mathbf{R}\}$ will be searched in each single iteration rather than directly seeking an accurate P. Therefore, the loss function

for Demons Algorithm can be represented as:

$$\arg\min_{\Delta \mathrm{p(x)}} \left\{ \sum_{\mathrm{x}} \left[I(W(\mathrm{x}; \mathrm{p(x)} + \Delta \mathrm{p(x)})) - T(\mathrm{x}) \right]^2 . \right\}, \tag{3.11}$$

where

$$\mathrm{p(x)} \leftarrow \mathrm{p(x)} + \Delta \mathrm{p(x)}.$$

Minimizing equation 3.11 is done by performing a first order Taylor expansion repected to $\Delta \mathrm{p(x)}$:

$$\arg\min_{\Delta \mathrm{p(x)}} \left\{ \sum_{\mathrm{x}} \left[I(W(\mathrm{x}; \mathrm{p(x)})) + \nabla I(\mathrm{x}) \frac{\partial W}{\partial \mathrm{p(x)}} \Delta \mathrm{p(x)} - T(\mathrm{x}) \right] \right\}, \tag{3.12}$$

where $\nabla I(\mathrm{x})$ is valued of gradient of image $I(\mathrm{x})$ at coordinate x. Additionally, the translation will be only applied to present local displacement of pixels in image $I(\mathrm{x})$. Therefore, apparently, for the translation, $\frac{\partial W}{\partial \mathrm{p(x)}}$ is always an 2×2 identity matrix. Equation 3.12 can be simplified as $[I(W(\mathrm{x}; \mathrm{p(x)})) + \nabla I(\mathrm{x}) \Delta \mathrm{p(x)} - T(\mathrm{x})]$. Equation 2.20 has been proposed to calculate stepwise $\Delta \mathrm{p(x)}$ at each iteration, which is recalled in the follows:

$$\Delta \mathrm{p(x)} = G_{\sigma^2} \otimes \left\{ \frac{[\nabla I(\mathrm{x})]^T [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p(x)}))]}{\|\nabla I(\mathrm{x})\|^2 + [T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p(x)}))]^2} \right\},$$

where extra term $[T(\mathrm{x}) - I(W(\mathrm{x}; \mathrm{p(x)}))]^2$ increases the stability of entire equation.

As discussed before, Demons Algorithm adopts Maxwell's demons force to help overcome the *aperture problem* (*i.e. constraint 2* of optic-flow algorithm). In above Demons Algorithm equation, Maxwell's demons force is presented by a Gaussian smoothing kernel G_{σ^2} with a variance of σ^2, where $\otimes$ indicates the convolution.

3.3.2 Concept of second-order Demons Algorithm

After recalling Demons Algorithm, concept of second-order Demons Algorithm, proposed by Pennec *et al.* [38] will be introduced here. As its name suggested, the second-order Demons Algorithm means minimisation of SSE is done by performing the second-order Taylor expansion expect to stepwise $\Delta \mathrm{p(x)}$, presenting as follows:

$$\arg\min_{\Delta \mathrm{p(x)}} \left\{ \sum_{\mathrm{x}} \left[I(W(\mathrm{x}; \mathrm{p(x)})) + \nabla I(\mathrm{x}) \Delta \mathrm{p(x)} + \frac{1}{2} \Delta \mathrm{p(x)}^T \nabla^2 I(\mathrm{x}) \Delta \mathrm{p(x)} - T(\mathrm{x}) \right] \right\}, \tag{3.13}$$

where $\nabla^2 I(\mathrm{x})$ indicates the second order gradient of image $I(\mathrm{x})$ at coordinate x. Then, minimization is actually executed by performing partial derivatives of equation 3.13 with respect to $\Delta\mathrm{p}(\mathrm{x})$, *i.e.*:

$$\arg\min_{\Delta\mathrm{p}(\mathrm{x})} \left[\nabla_{SSE}(\mathrm{x}) + H_{SSE}(\mathrm{x}) \cdot \Delta\mathrm{p}(\mathrm{x})\right], \tag{3.14}$$

where

$$\begin{aligned}
\nabla_{SSE}(\mathrm{x}) &= [\nabla I(\mathrm{x})]^T \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right], \\
H_{SSE}(\mathrm{x}) &= [\nabla I(\mathrm{x})]^T [\nabla I(\mathrm{x})] + \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right] \cdot \nabla^2 I(\mathrm{x}).
\end{aligned}$$

Assuming that $H_{SSE}(\mathrm{x})$ is positive definite, the minima is obtained for a null gradient:

$$\Delta\mathrm{p}(\mathrm{x}) = -H_{SSE}^{-1}(\mathrm{x}) \cdot \nabla_{SSE}(\mathrm{x}). \tag{3.15}$$

Then, a trace of H_{SSE} is applied to make a closest scalar matrix $\tilde{H}_{SSE}(\mathrm{x})$:

$$\tilde{H}_{SSE}(\mathrm{x}) = \frac{1}{n}\left\{\|\nabla I(\mathrm{x})\|^2 + \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right] \mathrm{Tr}(\nabla^2 I(\mathrm{x}))\right\} \cdot Id, \tag{3.16}$$

where n is the space dimension (2 for 2D and 3 for 3D), and Id is a 2×2 identity matrix. Adopting closest scalar matrix $\tilde{H}_{SSE}(\mathrm{x})$ in equation 3.15, the $\Delta\mathrm{p}(\mathrm{x})$ with regularisation step will come as follows:

$$\Delta\mathrm{p}(\mathrm{x}) = G_{\sigma^2} \otimes \left\{ \frac{n\,[\nabla I(\mathrm{x})]^T \left[T(\mathrm{x}) - I(W_{(X;D_X)})\right]}{\|\nabla I(\mathrm{x})\|^2 + \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right] \cdot \mathrm{Tr}(\nabla^2 I(\mathrm{x}))} \right\}. \tag{3.17}$$

Compared with the traditional Demons Algorithm, the second-order Taylor expansion gives an explanation of the extra term:

$$\alpha[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})]^2,$$

where α is a normalisation factor and simply set as 1 in Demons Algorithm (*i.e.* equation 2.20). In contrast, this normalisation factor α is defined in the second-order Demons Algorithm as follows:

$$\alpha = \frac{\mathrm{Tr}(\nabla^2 I(\mathrm{x}))}{\left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right]}.$$

This definition allows α to be adjusted adaptively at each iteration of warping parameter optimisation. On the contrary, α in traditional demons force is always defined and even bounded for $\alpha > 0$ by $\|\Delta\mathrm{p}(\mathrm{x})\| < 1/(2\sqrt{\alpha})$. However, nothing imposes that linearised loss function (*i.e.* equation 3.12) decreases with such

displacements. Whereas, the second order demons algorithm is based on the assumption that $\tilde{H}_{SSE}(\mathrm{x})$ is positive definite. If this denominator becomes close to zero, the second order Demons Algorithm could switch to another non-linear gradient descent method to minimise loss function (*i.e.* equation 3.14), such as Levenberg-Marquardt.

3.4 Two-Phase Probabilistic Second-order Demons Algorithm

Demons Algorithm is fast but falls short in terms of theoretical rigority, because it is based on intuitive ideas about free-form deformation. This is why it is difficult to predict when it fails and why. Moreover, due to the over-fitting and under-fitting, Demons Algorithm is unable to align two texture-rich images with a free-form transformation.

Therefore, the understanding of Demons Algorithm and its weakness opens many new research avenues. In §3.2.1, the over-fitting and under-fitting have been previously addressed by ill-poseness in warping parameter optimisation and low expression power of free-form transformation, respectively.

To solve ill-poseness in global image registration, Reid *et al.* [36] proposed a probabilistic image registration through applying the Bayesian rule method, which has been reviewed in §3.2. Therefore, we incorporate a *priori* knowledge into our improved Demons Algorithm to eliminate estimation errors caused by ill-poseness. Then, a smooth transition between fixed and fitted parameters can be achieved. When the errors for a particular fitted parameter is known to be large, then this parameter will be based more upon the prior information. To some extent, this Bayesian approach can decrease probability that the over-fitting and under-fitting occur in the warped image.

To tackle low expression power of free-form transformation, we propose a two-phase deformation strategy for free-form image registration. Through adding an extra point-to-point matching processing, the over-fitting and under-fitting caused by low expression power can be significantly removed.

In short, by integrating Bayesian approach with two-phase deformation strategy, our two-phase probabilistic second-order Demons Algorithm is very powerful and able to register even texture-rich images with free-form transformations.

3.4.1 Bayesian approach

Demons Algorithm iteratively proceeds through computing the loss function of image intensities via a linear approximation of stepwise warping parameters $\Delta \mathrm{p}(\mathrm{x})$:

$$z(\mathrm{x}) = [I(W(\mathrm{x}; \mathrm{p}(\mathrm{x}))) - T(\mathrm{x})] \approx \nabla I(\mathrm{x}) \cdot \Delta \mathrm{p}(\mathrm{x}). \tag{3.18}$$

We wish to calculate a stable and reliable estimate of $\Delta \mathrm{p}(\mathrm{x})$ by specifying uncertainty in the measurement equation, and *priori* information on $\Delta \mathrm{p}(\mathrm{x})$, and including them in the calculation. This estimation is the posterior distribution of $\Delta \mathrm{p}(\mathrm{x})$, given by Bayesian rule:

$$p(\Delta \mathrm{p}(\mathrm{x})|z(\mathrm{x})) = \frac{p(z(\mathrm{x})|\Delta \mathrm{p}(\mathrm{x}))p(\Delta \mathrm{p}(\mathrm{x}))}{\int_{-\infty}^{+\infty} p(z(\mathrm{x})|\Delta \mathrm{p}(\mathrm{x})p(\Delta \mathrm{p}(\mathrm{x}))d\Delta \mathrm{p}(\mathrm{x})}, \tag{3.19}$$

where $p(\Delta \mathrm{p}(\mathrm{x}))$ is the *priori* probability of the standard error in the fitted displacement field $\Delta \mathrm{p}(\mathrm{x})$ being true. $p(z(\mathrm{x})|\Delta \mathrm{p}(\mathrm{x})$ is the conditional probability, where z is observed given that $\Delta \mathrm{p}(\mathrm{x})$ is true and $p(\Delta \mathrm{p}(\mathrm{x})|z(\mathrm{x}))$ is the Baysien estimate of $\Delta \mathrm{p}(\mathrm{x})$ being true, given that measurement $z(\mathrm{x})$ has been made. The expression $\int_{-\infty}^{+\infty} p(z(\mathrm{x})|\Delta \mathrm{p}(\mathrm{x})p(\Delta \mathrm{p}(\mathrm{x}))d\Delta \mathrm{p}(\mathrm{x})$ is included so that the total probability of all possible outcomes is unity.

The maximum a posterior (MAP) estimate for the standard error of the fitted displacement field $\Delta \mathrm{p}(\mathrm{x})$ is the mode of $p(\Delta \mathrm{p}(\mathrm{x})|z(\mathrm{x}))$. For the purpose, $p(\Delta \mathrm{p}(\mathrm{x}))$ represents a known *priori* probability distribution from which the displacement field is drawn. $p(z(\mathrm{x})|\Delta \mathrm{p}(\mathrm{x}))$ is the likelihood of obtaining the displacement field given the data $z(\mathrm{x})$, and $p(\Delta \mathrm{p}(\mathrm{x})|z(\mathrm{x}))$ is the function to be maximised. The optimisation can be simplified by assuming that all probability distributions are multidimensional Gaussian, and can therefore be described by a mean vector and a covariance matrix.

When loss function is close to the minima, this optimisation becomes almost a linear problem. That allows us to exploit the estimated errors of the fitted displacement field $\Delta \mathrm{p}(\mathrm{x})$, which can be locally approximated by a multidimensional Gaussian distribution with covariance matrix $\Lambda_{\Delta \mathrm{p}(\mathrm{x})}$. We assume that the true standard error of displacement field is drawn from an underlying multidimensional Gaussian distribution of a known mean $\mu_{\Delta \mathrm{p}_0(\mathrm{x})}$ and a covariance $\Lambda_{\Delta \mathrm{p}_0(\mathrm{x})}$. By using the *priori* probability density function, we can obtain a better estimate of this true standard error of $\Delta \tilde{\mathrm{p}}(\mathrm{x})$ by taking a weighted average of $\mu_{\mathrm{p}_0(\mathrm{x})}$ $\Delta \mathrm{p}(\mathrm{x})$

$$\Delta \tilde{\mathrm{p}}(\mathrm{x}) = \left(\Lambda^{-1}_{\Delta \mathrm{p}_0(\mathrm{x})} + \Lambda^{-1}_{\Delta \mathrm{p}(\mathrm{x})}\right)^{-1} \left(\Lambda^{-1}_{\Delta \mathrm{p}_0(\mathrm{x})} \mu_{\Delta \mathrm{p}_0(\mathrm{x})} + \Lambda^{-1}_{\Delta \mathrm{p}(\mathrm{x})} \Delta \bar{\mathrm{p}}(\mathrm{x})\right), \tag{3.20}$$

where $\Delta\bar{\mathrm{p}}(\mathrm{x})$ is the sample mean of the standard errors of the fitted displacement field.

- **Estimation of covariance matrix $\Lambda_{\Delta\mathbf{p}(\mathbf{x})}$**

  In order to employ Bayesian approach, we must compute $\Lambda_{\Delta\mathrm{p}(\mathrm{x})}$, which is the estimated covariance matrix of the standard errors of the fitted displacement field. If the observations are independent, and each has unit standard deviation, $\Lambda_{\Delta\mathrm{p}(\mathrm{x})}$ is given by $H(\mathrm{x})$ in Demons Algorithm and H_{SSE} in second-order Demons Algorithm, respectively. Unfortunately, in practice, the standard deviation of the observations is unknown. Therefore, an assumption of its equal for all observations has been made, and estimated it from the standard differences parameter $z(\mathrm{x})$ between image $I(\mathrm{x})$ and template $T(\mathrm{x})$. A more accurate version of equation 3.19 includes terms representing uncertainty in the measurement:

$$z(\mathrm{x}) = [\nabla I(\mathrm{x}) - \varepsilon_2]\,[\Delta\mathrm{p}(\mathrm{x}) - \varepsilon_n] + \varepsilon_1, \tag{3.21}$$

  where terms ε_1, ε_2 and ε_n are zero mean Gaussian random variables of dimensionality 1, 2 and n, n is the number of parameter elements.

  ε_1 is in units of image intensity, and its variance should be set to the typical pixel intensity variance caused by noise. ε_2 not only represents the differences of aliasing effects between the image and the template, but also describes the errors of displacement field caused by an uncertainty estimation in gradient. The aliasing noise is represented as a small pixel offset, which is distributed by a 2D (usually isotropic) Gaussian. Meanwhile, the errors of displacement field is described by the deviation in pixel position caused by the linear approximation. ε_2 is 2D with units of pixels, and has components with a standard deviation similar to the average pixel location errors yielded by the differences in aliasing. Therefore, ε_2 could also be expressed as $\varepsilon_2 = \Omega_1 + \Omega_2$, where Ω_1 and Ω_2 are Gaussian random numbers with covariances Λ_{Ω_1} and Λ_{Ω_2} in order to illustrate the errors of aliasing and the errors of displacement field, respectively. The ε_n represents an uncertainty proportional of $\Delta\mathrm{p}(\mathrm{x})$ projected onto the intensity gradient. This error is closely related to the amount of the image smoothing, which is done before the image gradient calculation, named the intensity surface linearisation error.

  Then the likelihood function $p(z(\mathrm{x})|\Delta\mathrm{p}(\mathrm{x}))$ following from this uncertainty

model is:

$$p(z(\mathrm{x})|\Delta\mathrm{p}(\mathrm{x})) = \frac{1}{\sqrt{2\pi\sigma_z^2}}\exp\left\{-\frac{z(\mathrm{x}) - \nabla I(\mathrm{x})\Delta\mathrm{p}(\mathrm{x})}{2\sigma_z^2}\right\}, \tag{3.22}$$

where

$$\sigma_z^2 = \sigma_1^2 + [\nabla I(\mathrm{x})]\,\Lambda_2\,[\nabla I(\mathrm{x})]^T + \left[\nabla I(\mathrm{x})\frac{\partial W}{\partial \mathrm{p}(\mathrm{x})}\right]\Lambda_n\left[\nabla I(\mathrm{x})\frac{\partial W}{\partial \mathrm{p}(\mathrm{x})}\right]^T. \tag{3.23}$$

In equation 3.23, σ_1^2, Λ_2 and Λ_n are two variances and a covariance of ε_1, ε_2 and ε_n respectively. Then, covariance matrix $\Lambda_{\Delta\mathrm{p}(\mathrm{x})}$ in the second-order demons algorithm can now be estimated by:

$$\Lambda_{\Delta\mathrm{p}(\mathrm{x})} = \frac{\sigma_z^2}{n}\left\{\|\nabla I(\mathrm{x})\|^2 + [I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})]\,\mathrm{Tr}(\nabla^2 I(\mathrm{x}))\right\}\cdot\mathbf{Id}. \tag{3.24}$$

where **Id** denotes a 2×2 identity matrix.

- **Estimating $\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}$ and $\mu_{\Delta\mathrm{p}_0(\mathrm{x})}$**

  The *priori* distribution for $\Delta\mathrm{p}(\mathrm{x})$ (*i.e.* $p(\Delta\mathrm{p}(\mathrm{x}))$) with mean $\mu_{\Delta\mathrm{p}_0(\mathrm{x})}$ and covariance matrix $\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}$ can be estimated as:

$$\frac{1}{\sqrt{|2\pi\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}|}}\exp\left\{-\frac{1}{2}\left(\Delta\mathrm{p}(\mathrm{x}) - \mu_{\Delta\mathrm{p}_0(\mathrm{x})}\right)^T\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}^{-1}\left(\Delta\mathrm{p}_0(\mathrm{x}) - \mu_{\Delta\mathrm{p}_0(\mathrm{x})}\right)\right\}. \tag{3.25}$$

  This *priori* information (which is used to describe how likely each component of the errors of displacement field is to change) is useful for stabilising. Additionally, The estimation of covariance matrix $\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}$ in this *priori* distribution will determine the step size of iteration. When the $\Lambda_{\Delta\mathrm{p}_0(\mathrm{x})}$ is small, the convergence will speed up. Otherwise, the speed of convergence will decrease. For the estimation of $\mu_{\Delta\mathrm{p}_0(\mathrm{x})}$, the heuristic method has been chosen in the application. The value of $\mu_{\Delta\mathrm{p}_0(\mathrm{x})}$ is set at each iteration to the errors of displacement field, which represents the differences between current displacement field $\mathrm{p}^n(\mathrm{x})$ and initial estimation of $\mathrm{p}^{n-1}(\mathrm{x})$. But in the initialisation, $\mu_{\Delta\mathrm{p}_0(\mathrm{x})}$ will be set to zero.

- **Incorporating Bayesian approach into demons-based method**

  As mentioned previously, when the errors of displacement field estimation are closed to the minima, the registration problem is almost linear. Therefore, a *priori* information will be incorporated into the iterative optimissation scheme (*i.e.* steepest descent in demons-based algorithm) to calculate

second-order Demons Algorithm displacement field $\Delta\tilde{\mathrm{p}}(\mathrm{x})$ as follows:

$$\Delta\tilde{\mathrm{p}}(\mathrm{x}) = G_{\sigma^2} \otimes \frac{-\left\{\frac{\mu_{\Delta \mathrm{p}_0(\mathrm{x})}}{\mathrm{Tr}(\Lambda_{\Delta \mathrm{p}_0(\mathrm{x})})} + \sigma_z^{-2}\left[\nabla I(\mathrm{x})\right]^T \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right]\right\}}{\frac{1}{\Delta \mathrm{p}_0(\mathrm{x})} + \frac{\sigma_z^2}{n}\left\{\|\nabla I(\mathrm{x})\|^2 + \left[I(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right]\mathrm{Tr}(\nabla^2 I(\mathrm{x}))\right\}}. \tag{3.26}$$

3.4.2 Two-phase deformation strategy

As mentioned in the beginning of §3.4, after generating registration result by probabilistic second-order Demons Algorithm, a second-phase point-to-point matching processing will be used to justify registration errors, which is caused by low expression power.

Although probabilistic second-order Demons Algorithm significantly improves registration accuracy through successfully overcoming ill-poseness problem. Because of low expression power, the over-fitting and under-fitting still frequently occurs, and seriously limits registration performance.

In fact, the essence of this point-to-point matching processing is very simple. Every point in registration result $R(\mathrm{x})$ will iteratively match to template $T(\mathrm{x})$ again, with a non-regularised stepwise displacement field. This non-regularised stepwise displacement field is given in the following:

$$\Delta\tilde{\mathrm{p}}(\mathrm{x}) = \frac{-\left\{\frac{\mu_{\Delta \mathrm{p}_0(\mathrm{x})}}{\mathrm{Tr}(\Lambda_{\Delta \mathrm{p}_0(\mathrm{x})})} + \sigma_z^{-2}\left[\nabla R(\mathrm{x})\right]^T \left[R(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right]\right\}}{\frac{1}{\mathrm{Tr}(\Lambda_{\Delta \mathrm{p}(\mathrm{x})})} + \frac{\sigma_z^2}{n}\left\{\|\nabla R(\mathrm{x})\|^2 + \left[R(W(\mathrm{x};\mathrm{p}(\mathrm{x}))) - T(\mathrm{x})\right]\mathrm{Tr}(\nabla^2 R(\mathrm{x}))\right\}}. \tag{3.27}$$

Cooperating with probabilistic second-order Demons Algorithm, this two-phase deformation strategy can reliably prevent the occurrence of the over-fitting and under-fitting in free-form image registration.

That is why our two-phase probabilistic second-order Demons Algorithm is not only able to more accurately align homogeneous images with free-form deformation, but also can register texture-rich images with free-form deformation.

3.5 Experiment Results

We examine homogeneous image registration performance based on the "synthetic rectangle-to- star image sequence", which has been used to illustrate homogeneous image registration performance of Demons Algorithm in §2.3. To study texture-rich image registration performance, a "deformable cloth live sequence" and a

(a) (b) (c) (d)

Figure 3.2: Homogeneous registration comparison for the synthetic rectangle-to-star : (a). Result by Pyramids Lucas-Kanade, (b). Result by the traditional Demons Algorithm, (c). Result of the second-order Demons Algorithm, (d). Result of two-phase probabilistic second-order Demons Algorithm.

"back-yard image sequence" are chosen in this section, where the "deformable cloth live sequence" is offered by Mathieu Salzmann (EPFL), which have been used to test convex optimisation [43], and the "back-yard image sequence" is downloaded from optic-flow database (URL: http : //vision.middlebury.edu/flow/) which is builded by Baker *et al* [5].

All of these experiments are initialised with the same set of parameters. The rectangle free-form transformation is used to present the free-form transformation in the phase 1, where the rectangle size of control points is kept as 20×20. Meanwhile, for the comparison, this same rectangle free-form transformation will be applied in second-order Demons Algorithm and Demons Algorithm. To assess the quality of image registration results, we will calculate SSE between the registered image and the ground truth. Additionally, we will also compute MI them. Moreover, the registered result yielded by our approach will be compared with other results produced by Demons Algorithm, second-order Demons Algorithm and Pyramid Lucas-Kanade, respectively.

Figure 3.2 and Table 3.1 present the registered results obtained by different four registration methods. Obviously, second-order demons algorithm and our algorithm can provide more accurate registered images than the 4-level pyramid Lucas-Kanade algorithm and Demons Algorithm. Additionally, the registered image yielded by our approach is much smoother than that generated by second-order Demons Algorithm.

Figure 3.2-a shows the registered result aligned by the 4 level Pyramid Lucas-Kanade algorithm, proposed by Jean Yves Bouguet [9]. Although the coarse-to-fine strategy can partially improve registration accuracy and overcome the large motions between images, pyramid Lucas-Kanade algorithm still can hardly align

Method	Sum Squared Error		MI	
Two-phase Probabilistic 2nd Order Demons Algorithm	Phase 1	Phase 2	Phase 1	Phase 2
	4.3955	1.4142	1.9654	1.9988
2nd Order Demons Algorithm	4.4302		1.9595	
Original Demons Algorithm	14.7144		1.8797	
4 Level Pyramid Lucas-Kanade	18.9373		1.8570	

Table 3.1: Comparison for homologous region registration among two-Phase probabilistic 2nd order Demons Algorithm, 2nd order demons algorithm, original Demons Algorithm and 4 Level Pyramid Lucas-Kanade

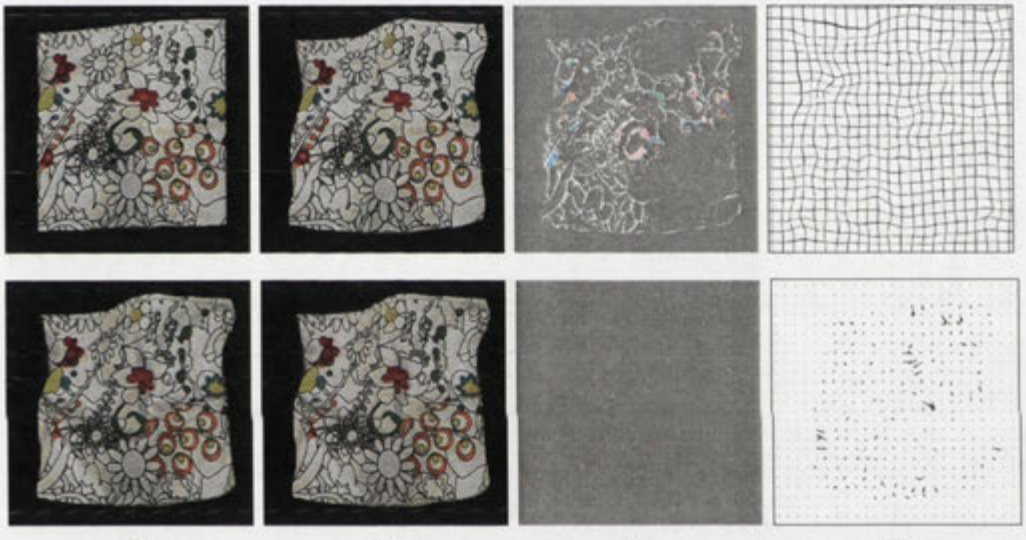


Figure 3.3: Deformable cloth live sequence registration procedure: from top to bottom, Column (a) image $I(\mathrm{x})$ and template $T(\mathrm{x})$; Column (b) includes alignment result after phase 1 registered and the final result of phase 2 registered; Column (c) present error image between the phase 1 registration and the template, and the final error image between phase 2 result and the template; Column (d) shows regularisation transformation mesh grid of phase 1 and displacement field of phase 2.

two homogeneous images (quality of registration: SSE: 18.9373; MI: 1.8570bit).

Figure 3.2-b is generated by Demons Algorithm. Compared with figure 3.2-a, registration performance of Demons Algorithm is better, due to the contributions of Maxwell's demons force (SSE: 14.7144 and MI: 1.8797bit).

In figure 3.2-c, second-order Demons Algorithm provides more reliable and

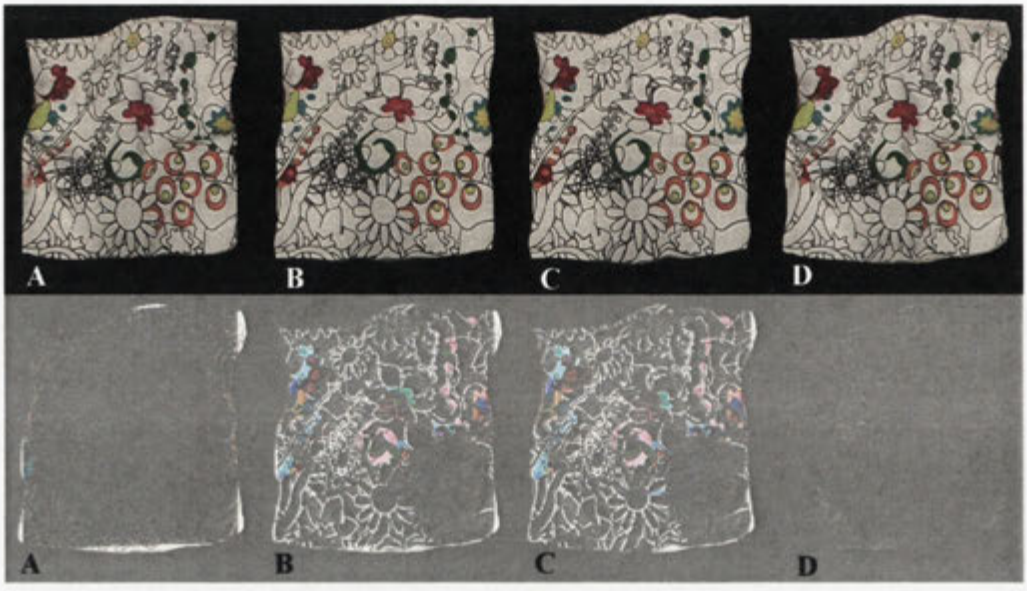


Figure 3.4: Top (A-D): Alignment results generated by four different registration algorithms: A. 4 Level Pyramid Lucas-Kanade, B. The original Demons Algorithm, C. Second order Demons Algorithm, D. Two phase probabilistic second order Demons Algorithm; Bottom (A-D): Registration interpolate errors of each corresponding algorithm.

Method	Sum Squared Error		MI	
Two-phase Probabilistic 2nd	Phase 1	Phase 2	Phase 1	Phase 2
Order Demons Algorithm	21.2557	6.6756	0.4104	1.0150
2nd Order Demons Algorithm	23.9358		0.4054	
Original Demons Algorithm	24.4735		0.3994	
4 Level Pyramid Lucas-Kanade	14.6105		0.9565	

Table 3.2: Comparison for cloth deformation live sequence registration among two-phase probabilistic second order Demons Algorithm, second order Demons Algorithm, original Demons Algorithm and 4 level pyramid Lucas-Kanade.

accurate registration result, where SSE is 4.4302 and MI rises to 1.9595bit.

Compared with above results, the registered result of the synthetic "rectangle-to-star sequence" is improved by two-phase probabilistic second-order Demons Algorithm, where SSE is 1.4142 and MI is 1.9988bit.

This experiment provides a solid evidence to support our two-phase probabilistic second order Demons Algorithm works well in homogeneous registration.

Then, we conduct a texture-rich image registration experiment by using the "deformable cloth live sequence", where the 144-th and 174-th frames are chosen. Figure 3.3 shows the registration procedure done by our approach. The top image in figure 3.3-a is the input image cropped from the 144-th frame. The bottom image in figure 3.3(a) is the template cropped from the 174-th frame. The top image and the bottom image in figure 3.3-b present the registered result of phase 1 and phase 2, respectively. Similarly, the top image and the bottom image in figure 3.3-c show the corresponding registration errors, respectively. In the top image in figure 3.3-c, we can find the outer contours between the input image and the template have been moved as close together as possible, while the texture registration errors (*i.e.* internal contour) is still large. The bottom image in figure 3.3-c shows the phase 2 (*i.e.* point-to-point matching processing) can magnificently reduce the registration errors in texture regions. Finally, the top and bottom images in figure 3.3-d are the regularised deformable mesh grid of phase 1 and the point-to-point displacement field of phase 2, respectively. The evaluation results of SSE and MI are represented in Table 3.2, the increased performance of our approach versus original and second order Demons Algorithm. Our approach improves both SSE measure and MI evaluation in each phase.

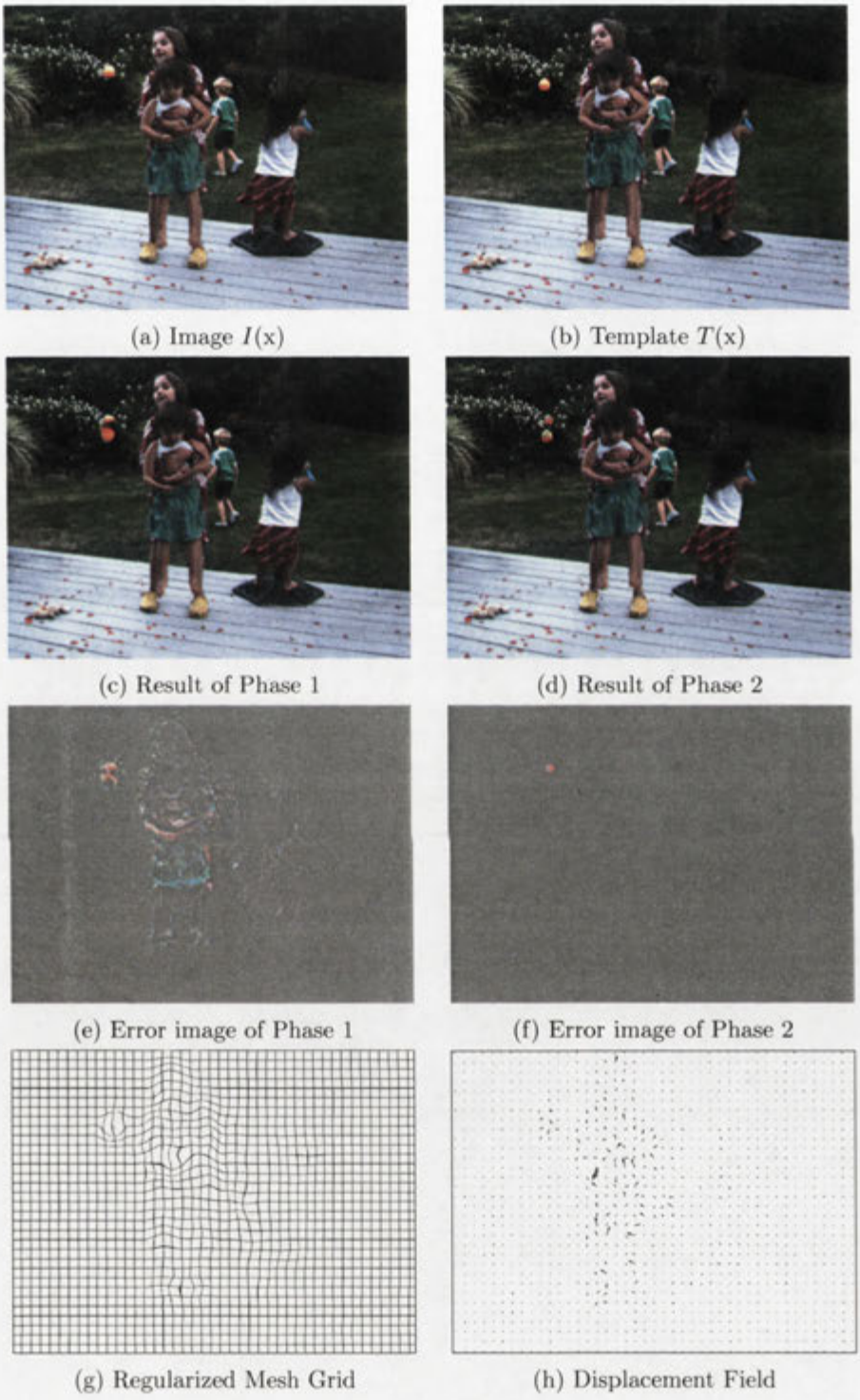
Additionally, the "back yard sequence" has been used to test our approach. Figure 3.5 illustrates this free-form image registration procedure between the 10-th frame (*i.e.* figure 3.5-a) and the 11-th frame (*i.e.* figure 3.5-b). The initial SSE and MI measures are 25.8704 and 0.9936bit, respectively. After phase 1 running, SSE and MI are getting into 11.7939 and 1.0525bit, respectively. Then, in phase 2, SSE and MI are turned to 3.5541 and 1.6831bit. This experiment proves that our algorithm is able to register general free-form images.

3.6 Conclusions

Incorporating *priori* knowledge into second-order Demons Algorithm and combining with two-phase deformation strategy, Demons Algorithm has been extended to the application domain of free-form homogeneous image registration to free-form texture-rich image registration. The phase 1 probabilistic second-order Demons Algorithm is able to move outer contours between the image and the template as close as possible. Then the phase 2 point-to-point matching processing is used to eliminate most texture registration errors between them. Integrating stabilising effect of prior knowledge and two-phase deformation strategy, the two-phase prob-

abilistic second-order Demons Algorithm does not only improve stabilisation and provide reliable convergence of warping parameters, but also extend the application domain of the traditional Demons Algorithm from free-form homogeneous image registration to free-form texture-rich image registration.

Our new algorithm achieves significant image registration accuracy in the free-form image alignment. However, the two phase probabilistic second order demons algorithm can not prevent the occurrence of over-fitting and under-fitting. To address this issue, a learning varying dimension RBF method will be proposed in the next chapter.



(a) Image $I(x)$ (b) Template $T(x)$

(c) Result of Phase 1 (d) Result of Phase 2

(e) Error image of Phase 1 (f) Error image of Phase 2

(g) Regularized Mesh Grid (h) Displacement Field

Figure 3.5: Backyard sequence registration procedure: (a) The image, (b) The Template, (c) Result of Phase 1, (d) Result of Phase 2, (e) Error image of phase 1, (f) Error image of Phase 2, (g) Regularised mesh grid, (h) Displacement Field.

Chapter 4

Learning Varying Dimension Radial Basis Functions

Contents:

4.1 Introduction

In chapter 3, we have discussed reasons for the occurrence of the over-fitting and under-fitting problems (*i.e.* issues that crucially limit the accuracy of image registration). To overcome such problem, we reviewed an existing probabilistic image registration method and proposed our two-phase probabilistic second-order demons algorithm for free-form image registration.

For global image registration, the over-fitting and under-fitting are caused by ill-poseness in warping parameter optimisation procedure, because the estima-

tion of warping parameters is actually based on an over-determined system of linear equations. Whereas, ill-poseness can be significantly eliminated by this probabilistic image registration.

Unfortunately, in free-form image registration, reasons for the occurrence of the over-fitting and under-fitting are more complex than that in global registration. Ill-poseness in warping parameter estimation is not the only reason contributing to the over-fitting and under-fitting in free-form registration. Because of the complexity, the over-fitting and under-fitting can also be caused by low expression power. Therefore, incorporating a *priori* knowledge can not resolve the mis-alignment caused by the over-fitting or under-fitting.

In this chapter, we will study how to establish an effective RBF transformation with a probabilistic warping parameters estimation to express the appropriate free-form deformation. Then, free-form image registration can prevent the over-fitting and under-fitting with this well estimated RBF transformation. To achieve this goal, the selection of control points has become a crucial yet challenging step.

4.1.1 Selection of control point

In rectangle free-form transformation, control points are four knots of each rectangle, and they have been uniformly distributed in the whole image. Therefore, control point selection is implicitly determined by size of the rectangle. This is because it actually determines the number and locations of control point. Then, these control points directly contribute to the expression power of rectangle free-form transformation. Rectangle free-form transformation has been widely applied, because it is simple to implement and easy to understand.

However, most general free-form deformations can not be effectively expressed by rectangle free-form transformation, because most of them are not uniform. Therefore, a set of uniformly distributed control points is insufficient to express the non-uniform deformations. Moreover, some control points may be located in certain featureless regions, which would cause low expression power. In addition to ill-poseness, this low expression power is the other reason contributing to the over-fitting and under-fitting in free-form image registration.

In contrast, control points of RBF transformation are selected manually and adaptively based on the image information. The locations and the number of control points will directly determine a specific RBF. Theoretically, with appropriate control points, RBF transformation can express any free-form transformation. However, in practice, how to select these control points is crucial and

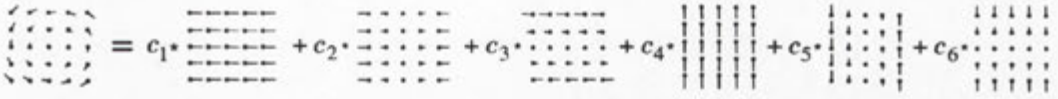


Figure 4.1: Illustration of affine optic-flow model of order six

challenging.

Recently, Bartoli and Zisserman *et. al.* presented a dynamic center insertion strategy [6] to help seek the appropriate control points. In this strategy, a latent control point library is established based on the sort of residual between images. Then, the minimal distance between adjacent control points is used as criterion to determine which potential point in the latent library should be inserted as a new control point. Meanwhile, a model selection criterion, *e.g.*Akaike Information Criterion (AIC), has been chosen as the stopping criterion in order to seek the 'best' number of control points.

However, taking the minimal distance between adjacent control points as selection criterion can not locate control points. This is because the minimal distance still need to be manually predefined. In other words, different minimal distance can produce different RBF transformation to express one free-form transformation.

After addressing the issue of control point selection, we will present a learning approach, which is used to determine the appropriate control points from the talent library. Then, through combing this learning approach with a probabilistic warping parameter estimation, a learning varying dimension RBF algorithm will be able to convincingly resolve the over-fitting and under-fitting in free-form image registration.

4.2 Recall of RBF Transformation

RBF transformation is concerned with the deformation expression as a regression of target variable Y on input set of covariates X, and give observations data pairings $\mathcal{D} = \{(\mathbf{y}_1, \mathbf{x}_1), (\mathbf{y}_2, \mathbf{x}_2), \ldots, (\mathbf{y}_n, \mathbf{x}_n)\}$. The regression curve or surface is assumed to be the conditional mean function $m(\mathbf{x}) = E[Y|X = \mathbf{x}]$ and the recorded observations Y corrupted by gaussian noise ϵ, so that we have

$$\mathbf{y}_i = m(\mathbf{x}_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.1}$$

where ϵ_i is an independent and identically distributed norm $\sim N(0, \sigma^2)$. In this section, we concern the approximation of $m(\mathbf{x})$ to be an estimated $\hat{m}(\mathbf{x})$ as follows:

$$\hat{m}(\mathbf{x}_i) = \sum_{j=1}^{K} w_j \phi_j(\mathbf{x}_i), \tag{4.2}$$

where $\{\phi_j(\mathbf{x})\}_{j=1,\ldots,n}$ represents a kernel function. w is the output coefficient, which is also considered as warping parameters.

RBF transformation has been proved as a popular approach for approximating of free-form transformation, and a review can be found in [39]. This free-form transformation is feed-forward model of RBF, where each kernel function is parameterised by the position vector μ of a control point, which is located in the $d-$dimensional covariance space X (*i.e.* 2D location vector of control points in free-form image registration). Conventionally, there are as many kernel functions as data points to be approximated with the position vectors. Output $m(\mathbf{x})$ of this RBF transformation could be rewritten as a linear combination of these kernel function responses and an extra low-order polynomial term as follows:

$$m(\mathbf{x}) = \sum_{j=1}^{K} w_j \phi_j(\|\mathbf{x} - \mu_j\|) + \sum_{m=0}^{p} a_m q_m(\mathbf{x}), \tag{4.3}$$

where $\|\cdot\|$ denotes a distance metric (*e.g.* Euclidean or Mahalanobis), and $q_m(\mathbf{x})$ represents a polynomial with degree m. Warping parameters w and a are calculated by least squares, where constraint $\{\sum_{i=1}^{N} w_j q_m(\mathbf{x}_i) = 0\}_{j=0,\ldots,p}$ is imposed to ensure the uniqueness of this solution.

In RBF transfromation, the low-order polynomial terms in equation 4.3 can be interpreted as an affine transformation, which has been used successfully for estimating and representing several global components and incorporated in as much literatures, *e.g.* [30]. Then, this low order polynomial term $\sum_{m=0}^{p} a_m q_m(\mathbf{x})$ could be rewritten as a displacement field $u(\mathbf{x})$ over coordinate $\mathbf{x} = (x, y)$:

$$u(\mathbf{x}) = \sum_{m=1}^{6} c_m \mathbf{b}_m(\mathbf{x}), \tag{4.4}$$

where $\{\mathbf{b}_m(\mathbf{x})\}_{m=1,\ldots,6}$ are six basis sets, while affine model required, encoding horizontal and vertical translation, shearing and scaling, respectively. $c = \{c_1, \ldots, c_6\}$ is the vector containing the scalar coefficients. Meanwhile, as mentioned in §1.2.1, the affine transformation is also able to be modelled by 2×2 matrix $\bar{A}$ and two-element vector $\mathbf{d} = (\mathbf{d}_x, \mathbf{d}_y)^T$. A better alternative is to enrich

the description of motion, that is to define a set of possible optic-flow functions $u(\mathbf{x}) = \{u^x(\mathbf{x}), u^y(\mathbf{x})\}^T$. Then, a pixel located in coordinate $\mathbf{x}$ in sensed image I is mapped to the corresponding coordinate $\mathbf{x}'$ in template T. This affine transformation can be summarised by the following equation:

$$u(\mathbf{x}) = \begin{pmatrix} u^x(\mathbf{x}) \\ u^y(\mathbf{x}) \end{pmatrix} = \bar{A}\mathbf{x} + \mathbf{d}. \tag{4.5}$$

RBF transformation defines a mapping from $\mathbb{R}^d$ to $\mathbb{R}$, where d is the dimension of data space X. A 2D displacement field $u(\mathbf{x})$ is defined by $\mathbb{R}^2 \to \mathbb{R}$ with a series of kernel functions $\{\phi_i(\|\mathbf{x} - \mu_i\|)\}_{i=1,\ldots,K}$. However, in image registration, the image and the template need be mapped from $\mathbb{R}^2 \to \mathbb{R}^2$. Therefore, the common way to construct this mapping m, is to stack 2 same sets of kernel functions with different coefficients ($\delta^x(\mathbf{x})$ and $\delta^y(\mathbf{x})$). Then the deformation between two images can be expressed by following equation:

$$m(\mathbf{x}) = \begin{pmatrix} \delta^x(\mathbf{x}) \\ \delta^y(\mathbf{x}) \end{pmatrix} = \bar{A}\mathbf{x} + \mathbf{d} + \sum_{j=1}^{K} \begin{pmatrix} w_j^x \\ w_j^y \end{pmatrix} \phi_j(\|\mathbf{x} - \mu_j\|), \tag{4.6}$$

where $\{\delta^x(\mathbf{x}), \delta^y(\mathbf{x})\}^T$ and $(w_j^x, w_j^y)^T$ represent encoding horizontal and vertical displacement fields and related coefficients, respectively.

The theory of radial basis functions indicates that many permissible forms of the kernel function $\{\phi_j(\|\mathbf{x} - \mu_j\|)\}_{j=1,\ldots,K}$ can be taken. In this chapter, thin-plate-spline $\phi(z) = z^2 \log z$ are chosen, which has an elegant algebra expressing the dependence of physical bending energy of a thin plate on points [8].

In free-form image registration, to efficiently resolve ill-posed problems and increase convergence stability, we need to remove the unnecessary degree of freedom. In neural networks community, it is to control complexity by the number of centers, which is less than the number of data points, which is advocated by Broomhead and Lowe [10]. The elimination of the dimension leads to the simple model with less variances but greater bias.

Improving convergence stability can also be achieved by adding a small, positive regularisation term Γ to the diagonal terms in the designed matrix of the least-squares solution of $\{w_j\}_{j=1,\ldots,K}$, which is known as Tikhonov regularisation [48] and has been discussed in probabilistic image registration. The larger value of Γ is, the smoother the warping parameters of RBF transformation will be.

These two approaches share a common motivation: to resolve ill-posed problems and increase the convergence stability of warping parameters estimation of

RBF transformation. However, these regularisation methods may be still too restrictive in many regions in image (*i.e.* over-fitting) and too loose in others (*i.e.* under-fitting). For example, when tracking a face, deformation of the jaw are much more likely than deformations of the nose. Moreover, as mentioned previously, ill-posed problems are not the only reason to cause over-fitting and under-fitting. Low expression power of transformation also contributes to the occurrence of the over-fitting and under-fitting. Therefore in the next section, we will propose a Bayesian learning framework of RBF transformation, which is able to resolve ill-posed problems and establish appropriate RBF transformation via appropriate control points.

4.3 Bayesian RBF with Varying Dimension

According to equation 4.2, these observation points between two images are fitted under data pairing $\mathcal{D}$:

$$\mathbf{y}_i = \bar{A}\mathbf{x}_i + \mathbf{d} + \sum_{j=1}^{K} \begin{pmatrix} w_j^x \\ w_j^y \end{pmatrix} \phi_j(\|\mathbf{x}_i - \mu_j\|) + \epsilon_i, \tag{4.7}$$

where ϵ_i is 2D zero-mean gaussian noise with covariance $\Lambda = \sigma^2\mathbf{Id}$, and $\mathbf{Id}$ is 2×2 identity matrix.

In our context, Bayesian framework is the calculation of a probability distribution on unknown warping parameters vector θ. Meanwhile, we also need to take into account of uncertainty transformation shifting among a set of latent RBF transformations $\mathcal{M}$ with varying dimensions. Thus, we make this free-form image registration explicit by writing the expectation as follows

$$E(Y|\mathbf{x}, \mathcal{D}) = \sum_{k=0}^{K} \int m(\mathbf{x}|\theta_k, M_k,)p(\theta_k|M_k, \mathcal{D})p(M_k|\mathcal{D})d\theta_k, \tag{4.8}$$

where $\mathcal{M} = \{M_0, \ldots, M_K\}$ is the set of latent RBF transformations entertained, while M_k is a defined RBF transformation with the determined locations and the determined number of control points $M_k = \{k, \mu_k\}$, and θ_k is to represent warping parameters w_k and affine parameters A_k. Posterior distribution $p(\theta_k, M_k|\mathcal{D})$ is given as a combination of likelihood and *priori*. Now, the aim of Bayesian radial basis functions of varying dimension is to make the Maximum A Posteriori

estimation (MAP).

$$p(\theta_k, M_k|\mathcal{D}) = \frac{p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k,)}{\int_{-\infty}^{+\infty} p(D|\theta_k, M_k)p(\theta_k, M_k)d\theta_k}$$

$$[M_k, \theta_k] \equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ p(M_k, \theta_k|\mathcal{D})$$

$$[M_k, \theta_k] \equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ \frac{p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k)}{\int_{-\infty}^{+\infty} p(D|\theta_k, M_k)p(\theta_k, M_k,)d\theta_k}$$

$$[M_k, \theta_k] \equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k), \tag{4.9}$$

where $p(\mathcal{D}|\theta_k, M_k)$ is the likelihood function and $p(\theta_k, M_k)$ is the *priori* density.

Assuming a normally distribution noise term in equation 4.7, we can obtain the following log-likelihood of the $p(\mathcal{D}|\theta_k, M_k)$, up to an additive constant,

$$\mathcal{L}[p(\mathcal{D}|\theta_k, M_k)] = -n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}[\mathbf{y}_i - m(\mathbf{x}_i; M_k, \theta_k)]^2. \tag{4.10}$$

Then, the location of each control point can be denoted by position vector μ_k. k indicates the number of control points. Now, any RBF transformation can be found from the latent RBF transformation, which is denoted as $\mathcal{M} = \{M_k|\mu_k, k\}$. After RBF transformation determined, the flexibility of this RBF transformation is controlled through a *priori* on θ_k. Typically, this *priori* will be a form of shrinkage *priori* $p(\theta_k|M_k) \sim N(\theta_k|\hat{\theta}_k, \sigma_m^2\mathbf{I})$ that penalizes large values. The *priori* is controlled through precision parameter σ_m^2, which shares the same smoothness effect provided by regularisation matrix Γ.

The analysis here accounts for the additional uncertainty present in the model selection of M_k. We place a proper *priori* over the entire set of latent RBF transformations library $\mathcal{M}$. Typically, a Poisson *priori* on k and an uniform *priori* on μ_k should be chosen, meanwhile setting the mean of this Poisson *priori* to be a small value penalises networks with a large number of control points, *e.g.* control points of a predefined rectangle free-form transformation overlapping the sensing image. However, this Poisson *priori* on k and uniform *priori* on μ_k will fail to concern the solution of ill-posed problems in warping parameters optimisation, which is already being achieved by the set a σ_m^2 on the output coefficients θ_k in the shrinkage *priori*. Therefore, we design a function Ψ_k integrating about k, μ_k and σ_m^2, which is readily computable and measures the amount of fitting that the model achieves. The details of function Ψ_k will be described in next subsection. Then a Gamma prior distribution is placed on Ψ_k, thus:

$$\begin{aligned} p(\theta_k, M_k) &= p(\theta_k, \mu_k, k) = p(\theta_k|M_k)p(\mu_k, k) \\ &= N(\theta_k|\hat{\theta}_k, \sigma_m^2\mathbf{I})\Gamma(\Psi_k|\alpha, \beta), \end{aligned} \tag{4.11}$$

where α and β are Gamma *priori* parameters, the mean of this Gamma distribution is α/β, and the variance is α/β^2. Additionally, in d-dimensions, setting the mean equal to $d+1$ indicates a preference for a linear fit. Then, $\hat{\theta}_k$ is the mean of conditional probability density $p(\theta_k|M_k)$, which will be reset at each iteration. However, at the beginning of the iteration, μ_k will be set to 0.

4.3.1 Function Ψ_k

Considering the typical system of linear equations, let ψ_k denote the coefficients matrix given by observation data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, while θ_k denotes the warping parameters of RBF transformation M_k. Then, this system of linear equations can be written as: $y = \psi_k \theta_k$. However, to overcome ill-posed problems, regularisation matrix $\sigma_m^2 \mathrm{I}$ is incorporated into coefficients matrix ψ_k. This regularisation procedure can be represented by a transformation matrix R, like $\hat{y} = Ry$. Then the least-square approximation should be:

$$\hat{\theta}_k = (\psi_k^T \psi_k + \sigma_m^2 \mathbf{I})^{-1} \psi_k^T y, \tag{4.12}$$

where σ_m^2 is the regularisation parameter mentioned in Tikhonov regularisation. Then, according to $\hat{y} = \psi_k \hat{\theta}_k$, $\hat{y}$ can be written as:

$$\hat{y} = \psi_k (\psi_k^T \psi_k + \sigma_m^2 \mathbf{I})^{-1} \psi_k^T y, \tag{4.13}$$

and hence

$$R = \psi_k (\psi_k^T \psi + \sigma_m^2 \mathbf{I})^{-1} \psi_k^T. \tag{4.14}$$

In accordance with Generalized Additive Model (GAM), we can define function Ψ_k as follows:

$$\Psi_k = tr(R). \tag{4.15}$$

This is the sum of the eigenvalues of R, which gives a measurement of amount of fitting that R the expected regularisation matrix for our model.

After all, the goal of this Bayesian RBF of varying dimension is to find appropriate RBF transformation (a RBF transformation with appropriate control points) and well estimated corresponding warping parameters, through making maximum a posteriori estimation. Unfortunately, those traditional fixed dimensional optimisation methods are not able to find maximum value of a posterior density equivalence $p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k)$, because the dimensions of this term will shift during optimisation. Thus, the Reversible Jump Markov Chain Monte Carlo (RJMCMC), presented by Green 1995 [23], will be applied to seek the maximum value of equation 4.9.

4.4 Introduction of RJMCMC

RJMCMC is a variation of traditional Markov Chain Monte Carlo (MCMC) method, thus MCMC method will be introduced at the beginning of this section.

4.4.1 MCMC

In fixed dimension situation, θ denotes a parameter vector belonging to a latent parameter library Θ. This parameter vector θ will be used in a possible probability distributions $p(\mathcal{D}|\theta)$ to express observations data pairing $\mathcal{D}$. Assuming $p(\theta)$ is a prior probability density describing θ, Bayesian formula will correspondingly give a posterior $\pi(\theta|\mathcal{D})$ in terms of likelihood and *priori*:

$$\pi(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}. \tag{4.16}$$

In traditional MCMC, a Markov Chain is constructed. In this Markov Chain, the latent parameter library Θ is as its state space. Whereas , $\pi(\theta|\mathcal{D})$ is its limiting stationary distribution. In the other words, this Markov Chain constructs an approach of sampling values from posterior distribution $\pi(\theta|\mathcal{D})$. Then Monte Carlo inference about θ is applied to make estimates of sample averages and densities.

Above Markov Chain can be described by a transition kernel procedure. This transition kernel will describe the state shifting procedure that each state θ_t in relevant probability distribution for the chain moves to the next state $\tilde{\theta}_{t+1}$ by step size $d\theta$, where t denotes the state number. For ease of exposition, it is assumed that the corresponding density function exists and denotes the transition density as $p(\theta_t, \tilde{\theta}_{t+1})$ for each distribution. MCMC methods produce the chain that are aperiodic, irreducible and fulfill a reversibility condition, also called the balance equation:

$$\pi(\theta_t|\mathcal{D})p(\theta_t, \tilde{\theta}_{t+1}) = \pi(\tilde{\theta}_{t+1}|\mathcal{D})p(\tilde{\theta}_{t+1}, \theta_t). \tag{4.17}$$

This equation means if $\pi(\theta_t)$ is the initial distribution of the starting state, the intensity of going from state θ_t to $\tilde{\theta}_{t+1}$ is the same as that of going from $\tilde{\theta}_{t+1}$ to θ_t. Meanwhile, the reversibility could directly lead to:

$$\int \pi(\theta_t|\mathcal{D})p(\theta_t, \tilde{\theta}_{t+1})d\theta = \pi(\tilde{\theta}_{t+1}|\mathcal{D}), \tag{4.18}$$

which means that $\pi(\theta_t|\mathcal{D})$ is in fact the stationary distribution of the chain. Then, we can use any sample from this chain as a random sample from distribution

$\pi(\theta_t|\mathcal{D})$. Therefore, according to Metropolis-Hastings algorithm, presented by Metropolis, Rosenbluth *et.al* (1953) [34], above described Markov Chain with transition density can be generated in the following:

$$p(\theta_t, \tilde{\theta}_{t+1}) = q(\theta_t, \tilde{\theta}_{t+1})\alpha(\theta_t, \tilde{\theta}_{t+1}). \tag{4.19}$$

For the proposal density $q(\theta^t, \tilde{\theta}^{t+1})$ is commonly chosen to be $N(0, \Sigma)$, where Σ is covariance matrix sharing the same dimensions with θ. Then, for acceptance probability $\alpha(\theta_k, \tilde{\theta}_{t+1})$ can be described in the following:

$$\alpha(\theta_t, \tilde{\theta}_{t+1}) = \mathbf{min}\left\{1, \frac{\pi(\tilde{\theta}_{t+1})q(\tilde{\theta}_{t+1}, \theta_t)}{\pi(\theta_t)q(\theta_t, \tilde{\theta}_{t+1})}\right\}. \tag{4.20}$$

If the proposed state $\tilde{\theta}_{t+1}$ was accepted, the $\tilde{\theta}_{t+1}$ is set to θ_{t+1} as the new state in the chain. Otherwise, the new state of the chain will be unchanged and stay in $\theta_{t+1} = \theta_t$. Now, the new state θ_{t+1} will form as the starting state for the next proposed state. This Metropolis-Hastings algorithm is iterated until a large number of samples have been drawn. Each updated state is just a function of previous state, and they are hence referred to as Markov Chains.

However, traditional MCMC method are restricted to densities of fixed dimensions. Whereas, the posteriori density $p(\theta_k, M_k|\mathcal{D})$ is a density of varying dimensions in equation 4.10. Therefore, traditional MCMC method can hardly be adopted to find the maximisation of the posteriori density. To extend traditional MCMC method to be available in densities of varying dimensions, the RJMCMC was developed.

4.4.2 RJMCMC

The key idea of reversible jump algorithm is to augment the usual proposal step in a conventional Metropolis-Hastings algorithm (*i.e.* equation 4.20) with a number of other possible shifting types surrounding a change in the dimensions of the densities.

In our RBF of varying dimensions, in addition to the possibility of moving within a latent warping parameters subspace Θ_k (*i.e.* Metropolis-Hastings algorithm, equation 4.20), the sampler can propose to switch dimension at each iteration, either up or down, by adding or removing a control point from the recent RBF transformation.

However, different from MCMC. in the RJMCMC method, the state space will be defined in the following:

$$E_k = \{M_k, \theta_k | M_k \in \mathcal{M}, \theta_k \in \Theta_k\}, \tag{4.21}$$

where $\mathcal{M}$ is latent RBF transformations library and Θ_k is the corresponding latent warping parameter subspace of a RBF transformation $M_k = \{k, \mu_k\}$. The dimension of θ_k is denoted by k.

To improve the convergence speed in free-form deformable image alignment, we implemented the birth and death proposition kernel, presented by Garcin *et.al.* [20]. The probability of attempting a birth or death step are respectively given by b_k and d_k, when the current RBF transformation has k control points. Additionally, $d_1 = 0$, $b_n = 0$, and $b_k = d_k = 0.1$ will be set for all other values of k, where n is the total number of the observation data pairing $\mathcal{D}$.

Commonly, the current state E_k (defined in equation 4.21) will jump between dimensions to form E_{k+1} with some random vector dE. We can generally use $E_{k+1} = E_k + dE$ to express this procedure. For a birth step, this vector dE is a datum vector drawn at random from those points in observation data pairing $\mathcal{D}$ that has not already become a control point. The jump move between dimensions is accepted with probability:

$$\alpha(E_k, E_{k+1}) = \mathbf{min}\left\{1, \frac{p(E_{k+1}|\mathcal{D})\tau_m(E_k)}{p(E_k|\mathcal{D})\tau_m(E_k)q(dE)}\left|\frac{\partial(E_{k+1})}{\partial(E_k, dE)}\right|\right\}, \tag{4.22}$$

where $\tau_m(E)$ is the probability of choosing a jump of type (*i.e.* the birth step or the death step), when the current RBF transformation is E_k. $q(dE)$ is the density function of dE. The final term, a Jacobian arises from the change of variables from (E_k, dE) to E_{k+1}. For the RBF transformation, the above accepting probability (*i.e.* equation 4.22) can be interpreted as **(likelihood ratio)×(prior ratio)×(proposal ratio)**.

- ***Likelihood Ratio***

  To illustrate this accepting probability, the likelihood ratio is $p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k)$, as given in equation 4.9.

- ***Prior Ratio***

  To describe the prior ratio, we take the birth step for example. In the birth step, we need consider the RBF transformation shifting from $E_k = \{M_k, \theta_k\}$ to $E_k = \{M_{k+1}, \theta_{k+1}\}$. Then, the prior ratio for the birth step is:

$$\frac{\Gamma(\Psi_{M_{k+1}}|\alpha, \beta)\,[(k+1)!(n-k+1)!/n!]\,p(\theta_{k+1}|M_{k+1})}{\Gamma(\Psi_{M_k})\,[k!(n-k)!/n!]\,p(\theta_k)|M_k}, \tag{4.23}$$

  where $\Gamma(\Psi_k)$ and $p(\theta_k|M_k)$ are taken from equation 4.11, and the term $k!(n-k)!/n!$ represents the probability of choosing the k-th control point from n data pairing points (*i.e.* observation data pairing $\mathcal{D}$).

Additionally, the prior ratio for the death step is just an inversion of the the prior ratio for the birth step:

$$\frac{\Gamma(\Psi_{M_k})\,[k!(n-k)!/n!]\,p(\theta_k)|M_k}{\Gamma(\Psi_{M_{k+1}}|\alpha,\beta)\,[(k+1)!(n-k+1)!/n!]\,p(\theta_{k+1}|M_{k+1})}. \tag{4.24}$$

- ***Proposal Ratio***

  Finally, the proposal ratio for the birth step can be given by:

$$\frac{d_{k+1}p(\theta_k|M_k)/(k+1)}{b_k p(\theta_{k+1}|M_{k+1}}\Big)/(n-k), \tag{4.25}$$

  where the full conditional of $p(\theta_k|M_{k+1})$ is used as the proposed distribution for θ_k, which is presented by warping parameters calculation in function Ψ_k (*i.e.* equation 4.12):

$$p(\theta_k|M_k) = N(\theta_k|\hat{\theta}_k, \sigma_m^2 \mathrm{I})$$

  where σ_m^2 is the prior precision for θ_k.

Now, the RJMCMC optimisation seeking the maximisation of the posteriori density $p(\mathcal{D}|\theta_k, M_k)p(\theta_k, M_k)$ has been established as above. The appropriate locations and the best number of control points can be automatically determined by RJMCMC optimisation. The detailed procedure of this iterative algorithm could be written in pseudo-code in algorithm 4.1. In algorithm 4.1, MOVE, BIRTH and DEATH steps are simple. MOVE selects a control point at random and resets its location vector to another datum drawn randomly from data set $\mathcal{D}$. BIRTH adds a new control point at a randomly selected point in the data set $\mathcal{D}$ that does not already contain one. DEATH selects just a control point at random and removes it.

4.5 Generation of Observations Data Pairing $\mathcal{D}$

We have established the whole RJMCMC algorithm to automatically estimate accurate warping parameters θ_k of an appropriate RBF transformation M_k for free-form image registration. However, this learning varying dimension RBF is established based on an assumption that observations data pairing $\mathcal{D}$ has been obtained. In this section, we will discuss how to generate this observations data pairing set from image I and template T.

Generation of observations data pairing set $\mathcal{D}$ is actually an image feature extraction and tracking problem. There are some existing well-studied feature

Algorithm 4.1 Pseudo-code of RJMCMC for Bayesian RBF transformation

Input: Problem equation 4.8, data pairings set $\mathcal{D}$

Output: Stationary distribution of Markov Chain: $p(\theta_k, M_k|\mathcal{D})$

Initialisation:

1. Define variance σ_m^2 from its prior probability: $\Gamma(\sigma_m^{-2}|10^{-3}, 10^{-3})$

2. Draw the out put coefficients θ by used the full conditionals of θ probability $\pi(\theta|\mathcal{D})$ (*i.e.* equation 4.16)

while do

(1). Draw an uniform random variable $dE \sim U(0, 1)$.

(2). Propose the next state of the chain as follow:

if $dE < b_k$ **then**

perform **BIRTH** step. $dE < b_k + d_k$

Perform **DEATH** step.

else

Perform **MOVE** step.

end if

(3). Redraw the coefficients θ' as initialisation step 2.

(4). Draw an uniform random variable $dE \sim U(0, 1)$.

if $dE < \alpha(E, E')$ (*i.e.* equation 4.23) **then**

Accept the proposed state.

else

Set the next state to be the current state.

end if

(5). Draw the variance σ_m^2 from its prior probability:
$\Gamma(\sigma_m^{-2}|10^{-3} + n/2, 10^{-3} + \varepsilon^2/2)$, where n is number of data points and ε^2 is the sum of squared residuals for the current model.

(6). Until convergence is assumed.

end while

(a) Unwarped Texture cloth (144) (b) Clair Speech sequence (55) (c) Heart (12)

(d) Warped Texture cloth (159) (e) Clair Speech sequence (57) (f) Heart (23)

Figure 4.2: Three different kinds of deformable surface: 1. (a) and (d) are the 144th and the 159th frame of the warping texture cloth with folding motion, respectively; 2. (b) and (e) are the 55th and the 57th frame in the Clair Speech Sequence, respectively, where contains the motion include lips motion and head pose change; 3. (c) and (f) are the 12th and the 23rd frame of the heart beating sequence. (The warping texture cloth sequence is provided by Mathieu Salzmann, EPFL; The Clair Speech Sequence and the heart beating sequence are downloaded from OSU/SAMPL Database-URL `http://sampl.ece.ohio-state.edu`)

extraction and tracking algorithms. In our learning varying dimension radial basis functions, Kanade-Lucas-Tomasi feature tracking method, has been adopted to generate the observations data pairing $\mathcal{D}$. Then, the warping parameters θ_k and RBF transformation M_k are learned from this observation data pairing $\mathcal{D}$ by RJMCMC. Therefore, the accuracy of this method is dependent of the accuracy of feature extraction and tracking. The more accurate features have been tracked, the better the free-form image registration we can get.

Because we have already estimated accurate warping parameters θ_k and established an appropriate RBF transformation M_k, then the free-form image registration has become a missing data points filling procedure. RBF transformation equation 4.6 with given parameters θ_k and M_k will be recalculated at each pixel of input image I to get dense displacement field.

4.6 Experiments

This section reports experimental results on three real motion sequences with different types of deformation.

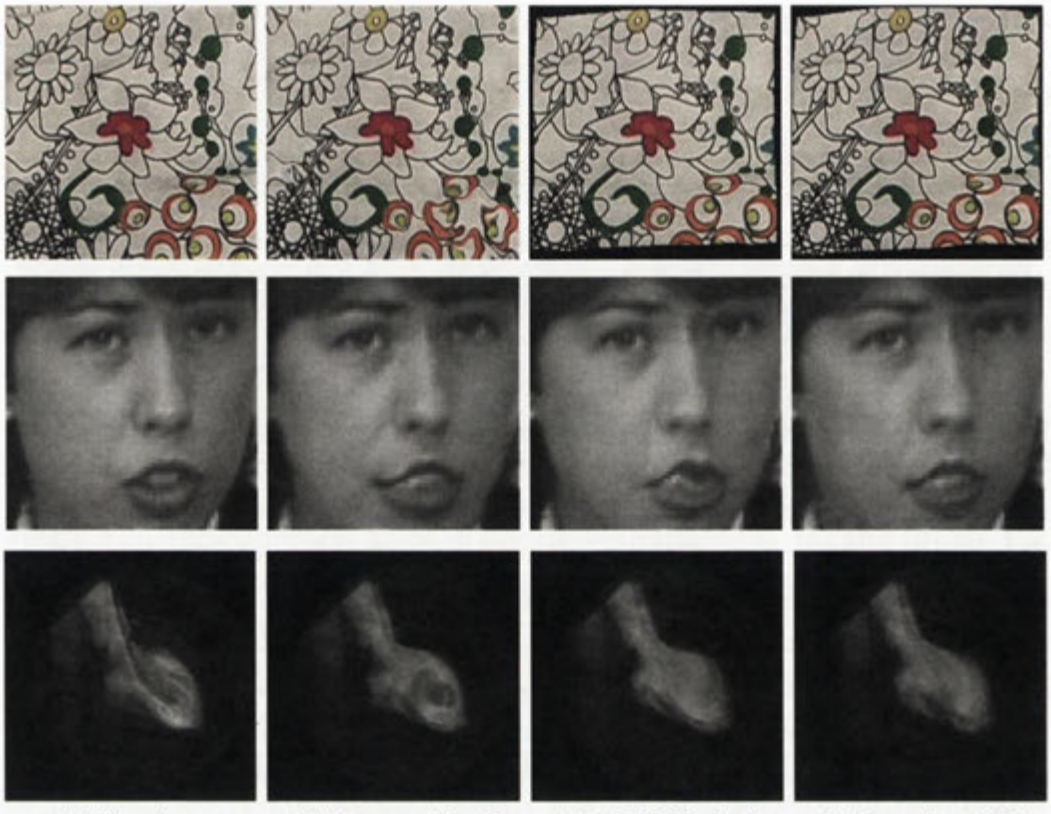
The warping cloth sequence, figure 4.2-a and 4.2-d, contains free-form deformation warped by hand, this sequence is provided by Salzmann. This sequence will be used to illustrate the performance of deformable texture-rich image alignment via our learning varying dimension RBF algorithm.

The Clair speech sequence, figure 4.2-b and 4.2-e, and the heart beating sequence, figure 4.2-c and 4.2-f are both downloaded from OSU/SAMPL Database. The movements in the Clair speech sequence are able to divided into the global motion of her head pose and the free-form deformation of her lips during speech. This sequence here is employed to evaluate the capability of our algorithm to resolve global and free-form deformations at the same time. The heart beating sequence contains numerous homogenous regions. Thus, this sequence can be applied to verify the homogenous image alignment performance of our algorithm.

The goal of these experiments is to validate learning varying dimension RBF approach and compared with three existing image registration algorithms, such as Demons Algorithm (*i.e.* Demon I in [49]) and Direct Estimation Radial Basis Mapping with Dynamic Centers Insertion (using the D.C.I. as abbreviation), presented by Bartoli *et.al.* in [6].

Our alignment results and quantitative evaluations in terms of SSE are presented in figure 4.3 and Table 4.1, respectively. Compared each algorithms in the first row of figure 4.3, we can find out learning varying dimension RBF can give more satisfied registration results than those yielded by others. It is easy to see that our approach is the best of three for warping cloth sequence. Meanwhile, we note there is a massive over-fitting appearing in the result generated by Demons I, due to Demons Algorithm focused on contours alignment by Maxwell demon force. Therefore, for complex texture-rich cloth folding motion, it can hardly handle this situation.

In the Clair speech sequence, the second row of figure 4.3, under-fitting and over-fitting appear in the lips motion area of Demon I and Direct Estimation Radial Basis Mapping with dynamic centers insertion method. But for head pose recovery, excellent performance has been given by above two methods. The reason for under-fitting in Demons I is caused by low expression power, because of featureless homogeneous regions in human face. Therefore, the local deformation, especially lips moving, is hardly described by Demons I. Although Direct



(a) Template (b) Demons Alg. **I** (c) D.C.I Method (d) Learning RBF

Figure 4.3: The templates and aligned images of all three approaches, three images in (a) column are templates of warping cloth, claire speech and heart beating, respectively. Images in (b) column are registered results of Demons I, as same, images in (c) and (d) columns are alignment results of dynamic centers insertion method and learning varying dimension RBF, respectively.

Sequence	Warping Cloth	Clair Speech	Heart Beating
Demons Algorithm I	23.37	8.12	11.18
Dynamic Centers Insertion	15.36	6.48	9.97
Learning Varying RBF	12.17	6.03	9.21

Table 4.1: Results of alignment error (measured by mean square error over 256 grey-levels).

Estimation Radial Basis Mapping method dynamically insert control points into image to solve low expression power problem, without quantitative determination of control points' density, it is also easy to cause the under-fitting with sparse

control points, and over-fitting under dense control points' distribution. Whereas, with incorporating *priori* information and learning framework, our approach can easily balance global transformation and local deformation by appropriate control points determination.

Finally, for the heart beating sequence, all of three deformable image alignment approach provide satisfied registered results. In this case, dynamic centers insertion and our algorithm can be interpreted as that they have degenerate to be equivalent to Demon II [49], because almost all of control points have been located near contours of ventricular and atrial. This is why all of these three algorithms make such similar alignment results in this sequence.

4.7 Conclusions

In this chapter, we have proposed a learning varying dimension RBF algorithm, which can solve the occurrence of over-fitting and under-fitting in the free-form image registration. This algorithm is able to simultaneously determine an appropriate RBF transformation as well as its best warping parameters, through integrating a learning processing from the observations data pairing with a probabilistic estimation. We will further extend our idea to multi-modality image registration. This will be detailed in next chapter - such issue has not been fully attended to in literatures.

Chapter 5

CCRE based Learning Varying Dimension RBF

Contents:

5.1 Introduction

In this chapter, learning varying dimension RBF is further extended to multi-modality free-form registration domain through incorporating a novel similarity measure - Cross-Cumulative Residual Entropy (CCRE). Learning varying dimension RBF formulation, proposed in chapter 4, is the method that aligns the image and the template by using an adaptive control points selection and a probabilistic warping parameter estimation [48]. Unfortunately, this algorithm is designed for single-modality registration.

Single-modality algorithms usually achieve image registration through minimising sum of squared errors (SSE) between two images. This is because SSE

has several advantages as a similarity measure. It is fast. It is simple to implement. It has a wide basin of convergence, which means make convergence easily. Its gradient is simple to derive, and it is well understood. Disadvantages of SSE include limited robustness to noise and variations in lighting conditions. Additionally, its wide basin of convergence can also make the result ambiguous. However, tracking multiple features and the use of models of appearance can significantly improve robustness, this is why our learning varying dimension RBF algorithm is able to partially robust to noise. In addition to above disadvantages, SSE has another crucial limitation that it can be only a similarity measure for single-modality image registration.

In multi-modality image registration, mutual information [50] is a popular similarity measure, which has been well described in chapter 2. Mutual information is applicable to multi-modality registration and robust to noise. However, mutual information also has a disadvantage limiting its performance as a similarity measure. The change of mutual information respecting to parameters is not smooth, because the density functions is not regular, even in the case that probability density function has been approximated by Parzen window. This irregular property can cause the optimisation frequently trapped into local minima, and this will lead to mis-alignment. To overcome this disadvantage, CCRE is applied as a similarity function as opposed to mutual information.

CCRE, developed by Wang and Vemuri [52], is based on a recently introduced information theoretic matching criterion to measure the similarity between two images. This new measure (dubbed cumulative residual entropy) unlike the well known Shannon entropy was shown to be consistently valid across discrete and continuous domains. Since it is based on cumulative distribution functions (CDFs) rather than probability densities. Compared with probability density functions, cumulative density functions are more regular and more robust in the presence of noise than probability densities.

5.2 Cross-Cumulative Residual Entropy

Wang and Vemuri [52] presented a novel registration criterion for multi-modality image registration. This criterion is based on an information theoretic measure called the cumulative residual entropy (CRE) [40], which is a measure of entropy defined using cumulative distributions. Let X indicate a random variable in R, and $F(\lambda) := P(|X| > \lambda)$ is the cumulative residual distribution. The cumulative

residual entropy of X, is defined as follows:

$$\varepsilon(X) = -\int_{R_+} F(\lambda) \log F(\lambda) d\lambda, \tag{5.1}$$

where $R_+ = (X \in R; X \geq 0)$. The essential idea in this definition is to use the cumulative distribution to replace the density function in Shannon's definition of entropy. The cumulative distribution function is regular, because it is defined in an integral form unlike the density function, which is defined as the derivative of the distribution. This definition also preserves the well established principle that the logarithm of the probability of an event should represent the information content in the event. CRE can be related to the well-known concept of mean residual life function, which is defined as:

$$m_F(t) = E(X - t | X \geq t) = \frac{\int_t^\infty F(X) dX}{F(t)}. \tag{5.2}$$

$m_F(t)$ is often used to measure departure from exponentiation. CRE can be shown to be the expectation of $m_F(t)$: $\varepsilon(X) = E[m_F(X)]$ [3]. Based on CRE, CCRE between two random variables is created, and it has been applied to solve multi-modality image alignment problem.

Now, we have sensed image $I(m(\mathrm{x}; M_k, \theta_k))$ and template $\tilde{T}(\mathrm{x})$, $M(\mathrm{x}; M_k, \theta_k)$ in equation 4.7 is a linear combination of the kernel functions in which $M_k = \{k, \mu_k\}$ is a defined RBF transformation including the number and the locations of kernel functions, and $\theta_k = \{w_k, A_k\}$ is used to represent kernel functions coefficients w_k and affine parameters A_k. Then, we define a similarity metric $S(I(m(\mathrm{x}); M_k, \theta_k))$ and maximise or minimise $S(\cdot)$ over RBF transformation $m(\mathrm{x})$. Mutual information for two random variables x and y can be defined like equation 2.23

$$\mathrm{E}(\mathrm{x}, \mathrm{y}) = \sum_{x \in \mathrm{x}} \sum_{y \in \mathrm{y}} p(x, y) \log_2 \left(\frac{p(x, y)}{p_{\mathrm{x}}(x) p_{\mathrm{y}}(y)} \right),$$

where $p(x, y)$ is the joint probability density function between random variables x and y. The reason for defining mutual information in terms of differential entropy as opposed to Shannon entropy is to facilitate the optimisation of mutual information with respect to warping parameters using any gradient based methods.

Whereas, the cross-CRE (CCRE) using CRE defined in equation 5.1 is defined as:

$$\mathcal{C}(X, Y) = \varepsilon(X) - E[\varepsilon(X/Y)]. \tag{5.3}$$

This quantity becomes a similarity criterion in image alignment problem. Consequently, image $I(m(\mathrm{x}; M_k, \theta_k))$ and template $\tilde{T}(\mathrm{x})$ have been incorporated into equation 5.3 RBF transformation $m(\mathrm{x}; M_k, \theta_k)$ describes the deformation, where θ_k is a set of warping parameters need to be determined. Now, the task of image registration becomes an optimisation problem. To align sensed image $I(m(\mathrm{x}; M_k, \theta_k))$ with template $\tilde{T}(\mathrm{x})$, we need seek a set of warping parameters M_k and θ_k, which maximises $\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))$ over the space of smooth transformations. This procedure is expressed in the following:

$$\arg\min_{(M_k, \theta_k)} \mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x})) \tag{5.4}$$

Then, the CCRE $\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))$ can be computed as follows:

$$\begin{aligned}
&\mathcal{C}\left(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x})\right) \\
&= \varepsilon(I(m(\mathrm{x}; M_k, \theta_k))) - \mathrm{E}[\varepsilon(I(m(\mathrm{x}; M_k, \theta_k))/\tilde{T}(\mathrm{x}))] \\
&= -\sum_{\lambda \in I_L} \left[\int_{\lambda}^{\infty} p_I(l; M_k, \theta_k) dl\right] \log_2 \left[\int_{\lambda}^{\infty} p_I(l; M_k, \theta_k) dl\right] \\
&\quad + \sum_{\lambda \in I_L} \sum_{s \in T_L} \left[\int_{\lambda}^{\infty} p_h(l, s; M_k, \theta_k) dl\right] \log_2 \left[\int_{\lambda}^{\infty} \frac{p_h(l, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} dl\right],
\end{aligned} \tag{5.5}$$

where I_L and $\tilde{T}_L$ are two sets of discrete intensities associated to sensed image $I(\mathrm{x})$ and template $\tilde{T}(\mathrm{x})$. Let $P_I(l > \lambda; M_k, \theta_k)$ and $P_h(l > \lambda, s; M_k, \theta_k)$ respectively denote $\int_{\lambda}^{\infty} p_I(l; M_k, \theta_k) dl$ and $\int_{\lambda}^{\infty} p_h(l, s; M_k, \theta_k) dl$, equation 5.5 can be simplified as

$$\begin{aligned}
&\mathcal{C}\left(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x})\right) \\
&= -\sum_{\lambda \in I_L} P_I(l > \lambda; M_k, \theta_k) \log_2 P_I(l > \lambda; M_k, \theta_k) \\
&\quad + \sum_{\lambda \in I_L} \sum_{s \in T_L} P_h(l > \lambda, s; M_k, \theta_k) \log_2 \frac{P_h(l > \lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} \\
&= -\sum_{\lambda \in I_L} \sum_{s \in T_L} P_h(l > \lambda, s; M_k, \theta_k) \log_2 P_I(l > \lambda; M_k, \theta_k) \\
&\quad + \sum_{\lambda \in I_L} \sum_{s \in T_L} P_h(l > \lambda, s; M_k, \theta_k) \log_2 \frac{P_h(l > \lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} \\
&= \sum_{\lambda \in I_L} \sum_{s \in T_L} P_h(l > \lambda, s; M_k, \theta_k) \left[\log_2 \frac{P_h(l > \lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} - \log_2 P_I(l > \lambda; M_k, \theta_k)\right] \\
&= \sum_{\lambda \in I_L} \sum_{s \in T_L} P_h(l > \lambda, s; M_k, \theta_k) \log_2 \frac{P_h(l > \lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s) P_I(l > \lambda; M_k, \theta_k)}
\end{aligned} \tag{5.6}$$

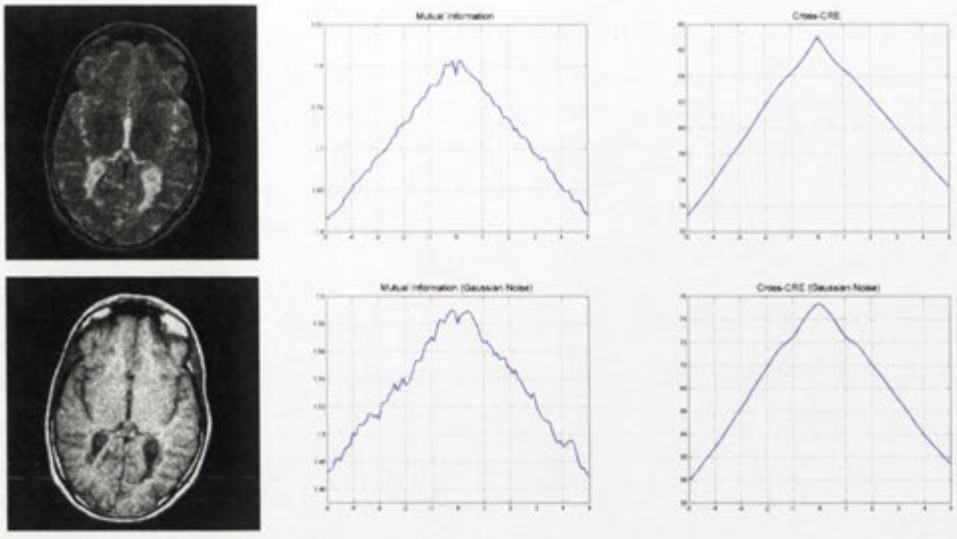


Figure 5.1: Plot of CCRE and mutual information for a multi-modality image registration between MR-T1 and MR-T2, where misalignment is generated by rotation of the MR-T1 image about an arbitrary axis in 3D. First column: MR-T1 and MR-T2 image pair distorted by zero-mean gaussian noise (Standard deviation 0.01). Second column: Plotted of different mutual information value over $-5°$ to $5°$. The top figure presented the misaligned image pair without gaussian noise distortion, the bottom one is result under distorted by gaussian noise. Third column: Plotted of different CCRE value over $-5°$ to $5°$. Top and bottom figures present results without gaussian noise and with gaussian noise, respectively

Compared with mutual information, there are three major advantages of CCRE as shown in figure 5.1. Firstly, the maxima value of CCRE appears at $0°$ of rotation either with or without gaussian noise distortion. Whereas, maxima value of mutual information appears closed to $0°$ of rotation, but disturbance occurs at exact $0°$ of rotation. Disturbance of mutual information is getting much severer under gaussian noise distortion. This property confirms that the CCRE measure is able to find optimum transformation parameters between two images. Secondly, the change of CCRE with respect to parameters is much more smoother than that of mutual information. This feature empirically validates that CCRE is more regular than mutual information. Theoretically, this justification stems from the fact that CCRE is based on cumulative density functions. Finally, in the same warping parameters range, CCRE shows much larger range of values than that of mutual information (CCRE: 72 bit to 87 bit within $-5°$ to $5°$; Mu-

tual Information: 1.62 bit to 1.81 bit). This property plays an important role in the numerical optimisation, since it would lead to a more stable numerical implementation by avoiding cancelation round off, which often plagues arithmetic operations with smaller numerical values.

5.2.1 Parzen Window for CCRE

In statistics, Parzen window method is a non-parametric way of estimating the probability density function of a random variable. Therefore, Parzen window technology is also implemented to estimate the cumulative residual function in CCRE.

In this section, a cubic and a zero-order B-spline window functions have been adopted as kernel functions in Parzen window. Then, the joint histogram between image $I(\mathrm{x})$ and template $\tilde{T}(\mathrm{x})$ can be rewritten as:

$$h(l,s;M_k,\theta_k)=\frac{1}{\varepsilon_I\varepsilon_T}\sum_{\mathrm{x}}\beta^3\left(\frac{l-I(m(\mathrm{x};M_k,\theta_k))}{\varepsilon_I}\right)\beta^0\left(\frac{s-\tilde{T}(\mathrm{x})}{\varepsilon_T}\right). \tag{5.7}$$

Consequently, the joint probability function is expressed based on $h(l,s;M_k,\theta_k)$

$$p_h(l,s;M_k,\theta_k)=\alpha(M_k,\theta_k)h(l,s;M_k,\theta_k), \tag{5.8}$$

where $\alpha(M_k,\theta_k)$ is a scalar coefficient to keep $\int_{-\infty}^{\infty}p_h(l,s;M_k,\theta_k)=1$, and it is defined in the following:

$$\alpha(M_k,\theta_k)=\frac{1}{\sum_{l\in I_L}\sum_{k\in T_L}h(l,s;M_k,\theta_k)}. \tag{5.9}$$

The fully form of joint probability density function is presented:

$$p_h(l,s;M_k,\theta_k)=\frac{\alpha(M_k,\theta_k)}{\varepsilon_I\varepsilon_T}\sum_{\mathrm{x}}\beta^3\left(\frac{l-I(m(\mathrm{x};M_k,\theta_k))}{\varepsilon_I}\right)\beta^0\left(\frac{s-\tilde{T}(\mathrm{x})}{\varepsilon_T}\right). \tag{5.10}$$

According to the definition of cumulative residual function, we have

$$\begin{aligned}
P_h(l>\lambda,s;M_k,\theta_k)&=\int_{\lambda}^{\infty}p_h(l,s;M_k,\theta_k)dl\\
&=\frac{\alpha(M_k,\theta_k)}{\varepsilon_I\varepsilon_T}\sum_{l=\lambda}^{\infty}\sum_{\mathrm{x}}\beta^3\left(\frac{l-I(m(\mathrm{x};M_k,\theta_k))}{\varepsilon_I}\right)\beta^0\left(\frac{s-\tilde{T}(\mathrm{x})}{\varepsilon_T}\right)\\
&=\frac{\alpha(M_k,\theta_k)}{\varepsilon_I\varepsilon_T}\sum_{\mathrm{x}}\int_{\lambda}^{\infty}\beta^3\left(\frac{l-I(m(\mathrm{x};M_k,\theta_k))}{\varepsilon_I}\right)dl\beta^0\left(\frac{s-\tilde{T}(\mathrm{x})}{\varepsilon_T}\right)
\end{aligned} \tag{5.11}$$

where the cubic B-spline kernel is an integrable piecewise function, which is expressed as follows:

$$\beta^3(x) = \begin{cases} (3|x|^3 - 6|x|^2 + 4)/6 & 0 \le |x| < 1 \\ (2 - |x|)^3/6 & 1 \le |x| < 2. \\ 0 & \text{elsewhere} \end{cases} \tag{5.12}$$

Then, the cumulative residual function of this cubic B-spline kernel is presented:

$$\begin{aligned} \Phi(\lambda) &= \int_{\lambda}^{\infty} \beta^3(x)dx \\ &= \begin{cases} 1 & x < -2 \\ 1 - \frac{(x+2)^4}{24} & -2 \le x < -1 \\ \frac{1}{2} - \frac{2}{3}x + \frac{x^3}{3} + \frac{x^4}{8} & -1 \le x < 0 \\ \frac{1}{2} - \frac{2}{3}x + \frac{x^3}{3} - \frac{x^4}{8} & 0 \le x < 1 \\ \frac{(x-2)^4}{24} & 1 \le x < 2 \\ 0 & x \ge 2 \end{cases} \end{aligned} \tag{5.13}$$

Figure 5.2 plots the differences among cubic B-spline, cumulative residual function, and cumulative density function. Now, with cumulative residual function $\Phi(x)$, the joint cumulative residual function between image $I(\mathrm{x})$ and template $\tilde{T}(\mathrm{x})$ is defined as follows:

$$P_h(l > \lambda, s; M_k, \theta_k) = \frac{\alpha(M_k, \theta_k)}{\varepsilon_I \varepsilon_T} \sum_{\mathrm{x}} \Phi\left(\frac{l - I(m(\mathrm{x}; M_k, \theta_k))}{\varepsilon_I}\right) \beta^0\left(\frac{s - \tilde{T}(\mathrm{x})}{\varepsilon_T}\right). \tag{5.14}$$

Using the same theory, cumulative residual function $P_I(l > \lambda; M_k, \theta_k)$ is:

$$\begin{aligned} & P_I(l > \lambda; M_k, \theta_k) \\ &= \frac{\alpha(M_k, \theta_k)}{\varepsilon_I \varepsilon_T} \sum_{\mathrm{x}} \sum_{s \in T_L} \Phi\left(\frac{l - I(m(\mathrm{x}; M_k, \theta_k))}{\varepsilon_I}\right) \beta^0\left(\frac{s - \tilde{T}(\mathrm{x})}{\varepsilon_T}\right) \\ &= \frac{\alpha(M_k, \theta_k)}{\varepsilon_I} \sum_{\mathrm{x}} \Phi\left(\frac{l - I(m(\mathrm{x}; M_k, \theta_k))}{\varepsilon_I}\right). \end{aligned} \tag{5.15}$$

5.2.2 Derivatives of CCRE

In learning varying dimension RBF, the free-form transformation will be determined with the appropriate locations and the best number of control points, *i.e.* $M_k = \{k, \mu_k\}$. Meanwhile, relevant warping parameters $\theta_k = \{w_k, A_k\}$ of this

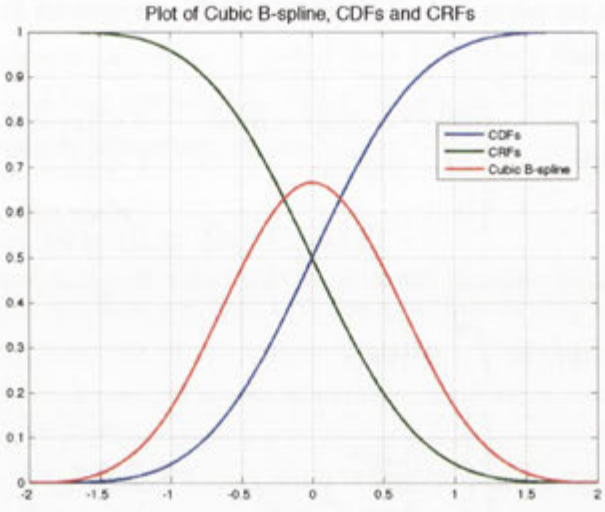


Figure 5.2: Differences among cubic B-spline (Red line), Cumulative Residual Function (CRFs) upon cubic B-spline (Green line) and Cumulative Density Function (CDFs) upon cubic B-spline (Blue line).

determined RBF need be determined via seeking the maxima of CCRE between image $I(\mathrm{x})$ and template $\tilde{T}(\mathrm{x})$. To achieve this purpose, the derivatives of CCRE are required.

- **Gradient of CCRE:**

  Let us define gradient $\nabla\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))$ as:

$$
\begin{aligned}
&\nabla\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x})) = \frac{\partial\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))}{\partial\theta_k} \\
&= -\sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial P_h(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k} \log_2 e \cdot P_I(l>\lambda; M_k, \theta_k) \\
&\quad + \sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial P_h(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k} \log_2 \frac{e\cdot P_h(l>\lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} \\
&= \sum_{\lambda\in I_L}\sum_{s\in T_L} \left[\log_2 \frac{e\cdot P_h(l>\lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)} - \log_2 e\cdot P_I(l>\lambda; M_k, \theta_k)\right] \\
&\quad \times \frac{\partial P(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k} \\
&= \sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial P(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k} \log_2 \frac{P(l>\lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s) P_I(l>\lambda; M_k, \theta_k)},
\end{aligned}
\tag{5.16}
$$

  where e is the exponential constant. We note the gradient formula of CCRE

is similar to CCRE formula itself. They share a same $\log_2(\cdot)$ term in their own formulas. This property will make efficient in the computation. This means only partial differential term need to be addressed. According to chain rule of differential computation, the partial differential term can be get:

$$\begin{aligned}
&\frac{\partial P(l > \lambda, s; M_k, \theta_k)}{\partial \theta_k} = \\
&\frac{\alpha(M_k, \theta_k)}{\varepsilon_I^2 \varepsilon_T} \sum_{\mathrm{x}} \beta^0 \left(\frac{s - \tilde{T}(\mathrm{x})}{\varepsilon_T} \right) \Phi' \left(\frac{l - I(m(\mathrm{x}; M_k, \theta_k))}{\varepsilon_I} \right) \\
&\times \left(-\frac{\partial I(t)}{\partial t}\Big|_{t=m(\mathrm{x};M_k,\theta_k)} \right) \frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k},
\end{aligned} \tag{5.17}$$

where $\Phi'(\lambda) = -\beta^3(\lambda)$. Then equation 5.17 is:

$$\begin{aligned}
&\frac{\partial P(l > \lambda, s; M_k, \theta_k)}{\partial \theta_k} = \\
&\frac{\alpha(M_k, \theta_k)}{\varepsilon_I^2 \varepsilon_T} \sum_{\mathrm{x}} \beta^0 \left(\frac{s - \tilde{T}(\mathrm{x})}{\varepsilon_T} \right) \beta^3 \left(\frac{l - I(m(\mathrm{x}; M_k, \theta_k))}{\varepsilon_I} \right) \\
&\times \left(\frac{\partial I(t)}{\partial t}\Big|_{t=m(\mathrm{x};M_k,\theta_k)} \right) \frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k},
\end{aligned} \tag{5.18}$$

where $\partial I(t)/\partial t$ is the gradient value of image $I(t)$ at position $t = m(\mathrm{x}; M_k, \theta_k)$, and $\partial m(\mathrm{x}; M_k, \theta_k)/\partial \theta_k$ is the Jacobian matrix upon the determined RBF transformation. Recalling equation 4.7 in chapter 4, RBF transformation is expressed as follows

$$m(\mathbf{x}) = \begin{pmatrix} \delta^x(\mathbf{x}) \\ \delta^y(\mathbf{x}) \end{pmatrix} = \bar{A}\mathbf{x} + \mathbf{d} + \sum_{j=1}^{K} \begin{pmatrix} w_j^x \\ w_j^y \end{pmatrix} \phi_j(\|\mathbf{x} - \mu_j\|).$$

Then $\partial m(\mathrm{x}; M_k, \theta_k)/\partial \theta_k$ is:

$$\frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k} = \begin{pmatrix} a_1, a_2, a_3, \phi_1(\|\mathbf{x} - \mu_1\|), \cdots, \phi_K(\|\mathbf{x} - \mu_K\|), & \mathbf{0}_{1\times(K+3)} \\ \mathbf{0}_{1\times(K+3)}, & a_4, a_5, a_6, \phi_1(\|\mathbf{x} - \mu_1\|), \cdots, \phi_K(\|\mathbf{x} - \mu_K\|). \end{pmatrix} \tag{5.19}$$

It is a $2 \times (2K + 6)$ matrix, where $a_1, a_2, \cdots, a_6$ are the parameters of affine transformation.

- **Hessian matrix of CCRE:**

Let us define the matrix of second derivative of $\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))$ as its hessian matrix $\nabla^2\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))$

$$\nabla^2\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x})) = \begin{pmatrix} \frac{\partial^2\mathcal{C}}{\partial\theta_1\partial\theta_1}, & \frac{\partial^2\mathcal{C}}{\partial\theta_1\partial\theta_2}, & \cdots \\ \frac{\partial^2\mathcal{C}}{\partial\theta_2\partial\theta_1}, & \frac{\partial^2\mathcal{C}}{\partial\theta_2\partial\theta_2}, & \cdots \\ \vdots \ , & \vdots \ , & \ddots \end{pmatrix}. \tag{5.20}$$

Hessian is obtained using chain and product rules:

$$\begin{aligned} &\frac{\partial^2\mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))}{\partial\theta_k^2} = \\ &-\left[\sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial^2 P_h(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k^2} \log_2\left(\frac{P_h(l>\lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)P_I(l>\lambda; M_k, \theta_k)}\right)\right] \\ &+\frac{1}{\log_e(2)}\left[\sum_{\lambda\in I_L}\left(\frac{\partial P_I(l>\lambda; M_k, \theta_k)}{\partial\theta_k}\right)^T \frac{\partial P_I(l>\lambda; M_k, \theta_k)}{\partial\theta_k}\frac{1}{P_I(l>\lambda; M_k, \theta_k)}\right] \\ &-\frac{1}{\log_e(2)}\left[\sum_{\lambda\in I_L}\sum_{s\in T_L}\left(\frac{\partial P_h(l, k; M_k, \theta_k)}{\partial\theta_k}\right)^T \frac{\partial P_h(l, s; M_k, \theta_k)}{\partial\theta_k}\frac{1}{P_h(l>\lambda, s; M_k, \theta_k)}\right]. \end{aligned} \tag{5.21}$$

In equation 5.21, based on the second order variation of the joint cumulative residual function, when a pair of registration parameters varies jointly, this term could be eliminated, which amounts to linearisation the variation of $P_h(l>\lambda, s; M_k, \theta_k)$ with respect to θ_k. Another motivation for ignoring the first term of equation 5.21 aries when one considers the situation at ideal registration of two dependent images. In this case, $P_h(l>\lambda, s; M_k, \theta_k) = p_{\tilde{T}}(s)P_I(l>\lambda; M_k, \theta_k)$. Then, the partition of unity condition implies:

$$\begin{aligned} &\sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial^2 P_h(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k^2} \log_2\left(\frac{P_h(l>\lambda, s; M_k, \theta_k)}{p_{\tilde{T}}(s)P_I(l>\lambda; M_k, \theta_k)}\right) \\ &= \sum_{\lambda\in I_L}\sum_{s\in T_L} \frac{\partial^2 P_h(l>\lambda, s; M_k, \theta_k)}{\partial\theta_k^2} \log_2(1) = 0. \end{aligned} \tag{5.22}$$

After first term of equation 5.21 vanished, the remain terms do still contribute, and the Hessian does not globally vanish at ideal situation. This is important to keep super-linear convergence near the optimum. Finally, the

Name	Update
Newton Decent	$p^{(t+1)} \leftarrow p^{(t)} - H^{-1}G$
Quasi-Newton Descent	$p^{(t+1)} \leftarrow p^{(t)} - \tilde{H}^{-1}G$
Steepest Descent	$p^{(t+1)} \leftarrow p^{(t)} - \delta G$
Levenberg Marquardt	$p^{(t+1)} \leftarrow p^{(t)} - ((1+\delta \mathrm{I})\tilde{H})^{-1}G$

Table 5.1: Updates for four Newton-type optimisation methods. (1 is a matrix of ones). Although not explicitly indicated, several δ^2 values may be tested.

following simplified form of CCRE Hessian matrix is adopted.

$$\frac{\partial^2 \mathcal{C}(I(m(\mathrm{x}; M_k, \theta_k)), \tilde{T}(\mathrm{x}))}{\partial \theta_k^2} \approx$$
$$\frac{1}{\log_e(2)} \left[\sum_{\lambda \in I_L} \left(\frac{\partial P_I(l > \lambda; M_k, \theta_k)}{\partial \theta_k} \right)^T \frac{\partial P_I(l > \lambda; M_k, \theta_k)}{\partial \theta_k} \frac{1}{P_I(l > \lambda; M_k, \theta_k)} \right]$$
$$- \frac{1}{\log_e(2)} \left[\sum_{\lambda \in I_L} \sum_{s \in T_L} \left(\frac{\partial P_h(l, k; M_k, \theta_k)}{\partial \theta_k} \right)^T \frac{\partial P_h(l, s; M_k, \theta_k)}{\partial \theta_k} \frac{1}{P_h(l > \lambda, s; M_k, \theta_k)} \right]. \tag{5.23}$$

Apparently, the every term needed in Hessian has been already pre-computed in the determination of the gradient. Thus, another consequence of ignoring the second order term in equation 5.21 is that Hessian $\nabla^2 \mathcal{C}(\cdot)$ comes at essentially no additional computational cost with respect to that of the gradient $\nabla \mathcal{C}(\cdot)$.

5.2.3 Maximal Optimisation for CCRE

Many optimisation algorithms exist, but a particular group so-called Newton-type method is most common and popular. These methods assume locally parabolic topology and seek the minimum using gradient information: $p^{(t+1)} \leftarrow p^{(t)} - H^{-1}(p^{(t)})G(p^{(t)})$. Here H is the Hessian of the loss function with respect to $p^{(t)}$, G is the gradient of loss function with respect to $p^{(t)}$, and t indicates the iteration number. There are several variants of standard Newton type optimisation, mentioned in the Table 5.1.

In table 5.1, the convergence to an optimum of these top three methods is not guaranteed: it may converge to a saddle point, which means at the same time a maximum for some parameters and a minimum for another parameters in the set θ_k. Even worse, it diverges from the desired solution when the problem is not

convex. In return, it is extremely efficient when the criterion is locally quadratic convex, for in this case it finds the optimum after a single criterion evaluation.

Levenberg Marquardt strategy is a convenient way to combine the advantages of the gradient method with those of the Newton method, preserving the efficiency of the latter when the conditions are nearly optimal, and robustness of the former when they are not. Depending on the value of δ, one can distinguish two extreme cases. When $\delta \to 0$, one sees that Levenberg Marquardt formula and Newton descent formula are identical. When $\delta \to \infty$, the diagonal terms of $(1 + \delta \mathrm{I})\tilde{H}$ dominate, and the formula degenerates to steepest descent method. Note that although the magnitude of the update is adapted to each component by the virtue of the normalising term H^{-1} are vanishingly small in the second case. Then δ can be adapted between these two extremes in order to achieve a good compromise between the efficiency but lack of robustness of the Newton-type approach, and the size of the steps of robust but generally inefficient gradient approach.

- **CCRE Summary:**

  The procedure of CCRE based registration algorithm with determined RBF transformation $M_k = \{\mu_k, k\}$ is presented as follows:

  1. For the current transformation model M_k and initialisation of coefficients θ_k^0 of this model M_k, interpolate the image by $I(m(\mathrm{x}; M_k, \theta_k^0))$. Calculate $P_h(l > \lambda, s; M_k, \theta_k)$, $P_I(l > \lambda; M_k, \theta_k)$, and $\frac{\partial P_h(l>\lambda,s;M_k,\theta_k)}{\partial(\theta_k|M_k)}$ using the equation 5.14, equation 5.15, and equation 5.18.

  2. Compute common term $\log_2(\frac{P_h(l>\lambda,s;M_k,\theta_k)}{p_{\hat{T}}(s)P_I(l>\lambda;M_k,\theta_k)})$ in both CCRE and the gradient of CCRE.

  3. Compute the gradient and hessian of the CCRE using equation 5.16 and equation 5.23. Then, adopting Levenberg Marquardt approach to get $\delta\theta_k^i$ of model M_k.

  4. Update warping coefficient by $\theta_k^{i+1} \leftarrow \theta_k^i + \delta\theta_k^i$, till find optimum coefficient $\hat{\theta}_k$ of RBF transformation M_k. Then, stop this registration process, otherwise go to **Step 1**.

5.3 CCRE based Learning Varying Dimension RBF

The above section provides us with a generic CCRE image registration processing flow. However, only CCRE is not adequate to resolve multi-modality free-form image registration, without an appropriate RBF transformation. Therefore, we consider to incorporate learning varying dimension RBF into CCRE to deal with this task.

However, to achieve this purpose, there are two major changes during learning varying dimension RBF implementing in the multi-modality image alignment:

1. In multi-modality case, Observation data pairing set $\mathcal{D}$ can not be pre-obtained by Kanade-Lucas-Tomasi feature tracking algorithm. Therefore, instead of this observation pairing data set, some feature points with high gradient values have been extracted from image $I(\mathrm{x})$ and to compose a latent control point library $\mathcal{D}_{cp}$.

2. Due to invalid data pairing set $\mathcal{D}$ for multi-modality image registration, the warping parameter estimation must be and only be based on CCRE similarity measure between image I and template T.

In conclusion, if we want to incorporate CCRE similarity measure into learning varying dimension RBF, there are several modifications need to be made.

Firstly, equation 4.9 will be changed with latent variable preference of probability rule. In order to simplify the expression, we have $\mathrm{y} = m(\mathrm{x}; M_k, \theta_k)$. Then, take account of uncertainty among RBF transformation set $\mathcal{M}$, the CCRE explicit by writing the expectation as:

$$E\left[\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))\right] = \sum_{k=0}^{K} \int_{-\infty}^{\infty} \mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x})) p(\mathrm{y}|M_k, \Delta\theta_k) p(\Delta\theta_k|M_k) p(M_k) d\Delta\theta_k, \tag{5.24}$$

where $M_k = \{k, \mu_k\}$ is a determined RBF transformation, and θ_k represents its coefficients w_k and A_k. Because the coefficients of M_k can not be directly obtained by closed-form least square linear equations, the iterative approach is employed to compute $\theta_k^{t+1} \leftarrow \theta_k^t + \Delta\theta_k$, where t indicates the number of iterations.

In singular-modality image registration, the aim of learning varying dimension RBF is to seek the MAP estimation based on training data pairing set $\mathcal{D}$, *i.e.*

equation 4.9:

$$
\begin{aligned}
p(M_k, \theta_k|\mathcal{D}) &= \frac{p(\mathcal{D}|M_k, \theta_k)p(M_k, \theta_k)}{\int_{-\infty}^{\infty} p(D|M_k, \theta_k)p(M_k, \theta_k)d\theta_k} \\
[M_k, \theta_k] &\equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ p(M_k, \theta_k|\mathcal{D}) \\
[M_k, \theta_k] &\equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ \frac{p(\mathcal{D}|M_k, \theta_k)p(M_k, \theta_k)}{\int_{-\infty}^{\infty} p(D|M_k, \theta_k)p(M_k, \theta_k)d\theta_k} \\
[M_k, \theta_k] &\equiv \underset{M_k \in \mathcal{M}, \theta_k \in \Theta_k}{\arg\max}\ p(\mathcal{D}|M_k, \theta_k)p(M_k, \theta_k).
\end{aligned}
$$

However, without generating training data pairing set $\mathcal{D}$, the aim of multi-modality is transformed based on CCRE measurement $\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))$ as following:

$$
\begin{aligned}
p(M_k, \Delta\theta_k|\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))) &= \frac{p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|M_k, \Delta\theta_k)p(M_k, \Delta\theta_k)}{\int_{-\infty}^{+\infty} p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|M_k, \Delta\theta_k)p(M_k, \Delta\theta_k)d\Delta\theta_k} \\
[M_k, \Delta\theta_k] &\equiv \underset{M_k \in \mathcal{M}, \Delta\theta_k \in \Delta\Theta_k}{\arg\max}\ p(M_k, \theta_k|\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))) \\
[M_k, \Delta\theta_k] &\equiv \underset{M_k \in \mathcal{M}, \Delta\theta_k \in \Delta\Theta_k}{\arg\max}\ \frac{p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|M_k, \Delta\theta_k)p(M_k, \Delta\theta_k)}{\int_{-\infty}^{+\infty} p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|M_k, \Delta\theta_k)p(M_k, \Delta\theta_k)d\Delta\theta_k} \\
[M_k, \Delta\theta_k] &\equiv \underset{M_k \in \mathcal{M}, \Delta\theta_k \in \Delta\Theta_k}{\arg\max}\ p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|M_k, \Delta\theta_k)p(M_k, \Delta\theta_k), \qquad (5.25)
\end{aligned}
$$

where $p(\Delta\theta_k, M_k)$ can be expressed as multiply between a gaussian *priori* distribution and a gamma prior based on $\Delta\Psi$, which is modification version of function Ψ, (*i.e.* equation).

$$
\begin{aligned}
p(\Delta\theta_k, M_k) &= p(\Delta\theta_k, \mu_k, k) = p(\Delta\theta_k|M_k)p(\mu_k, k) \\
&= N(\Delta\theta_k|\Delta\hat{\theta}_k, \sigma_\theta^2\mathbf{I})\Gamma(\Psi_{\hat{H}}|\alpha, \beta). \qquad (5.26)
\end{aligned}
$$

The value μ_k is set at each iteration to update coefficients $\Delta\theta_k$, which represents the differences between current coefficients θ_k^t and initial estimate of θ_k^0, set solely from the *priori* information. This ensures when θ_k is being updated, the *priori* estimation of θ_k is always effective for the set of starting θ_k^0. Additionally, α and β are Gamma parameters, the mean of the Gamma distribution is α/β, and variance is α/β^2. In d-dimensions, setting the mean equal to $d+1$ indicates a preference for a linear fit. Then the CCRE is:

$$
\begin{aligned}
\tilde{\mathrm{y}} &= m(\mathrm{x}; M_k, \theta_k^t + \Delta\theta_k) + m \\
\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x})) &= \mathcal{C}(I(\tilde{\mathrm{y}}), \tilde{T}(\mathrm{x})) + c, \qquad (5.27)
\end{aligned}
$$

where m and c are zero-mean gaussian vectors with variance Λ_m^2 and σ_c^2, respectively.

$$\Lambda_m^2 = \left[\frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k}\right] \sigma_\theta^2 \mathbf{I} \left[\frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k}\right]^T$$

$$\sigma_c^2 = \left[\frac{\partial I(t)}{\partial t}\right] \left[\frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k}\right] \sigma_\theta^2 \mathbf{I} \left[\frac{\partial m(\mathrm{x}; M_k, \theta_k)}{\partial \theta_k}\right]^T \left[\frac{\partial I(t)}{\partial t}\right]^T$$

where,

$$t = m(\mathrm{x}; M_k, \theta_k). \tag{5.28}$$

Then, we obtain the log-like maximum likelihood estimation entails minimising the following least-squares objective with respect to the unknowns:

$$\begin{aligned} &\mathcal{L}[p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|\Delta\theta_k, M_k)] = \ln p(\mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))|\Delta\theta_k, M_k) \\ &= -\ln(2\pi\sigma_c^2) - \frac{1}{2\sigma_c^2}\left[\mathcal{C}(I(\tilde{\mathrm{y}}), \tilde{T}(\mathrm{x})) - \mathcal{C}(I(\mathrm{y}), \tilde{T}(\mathrm{x}))\right]^2. \end{aligned} \tag{5.29}$$

- **Function $\Psi_{\hat{H}}$:**

  After that, the function $\Psi_{\hat{H}}$ need be redesigned. As mentioned in the CCRE section, update coefficients $\Delta\theta_k$ is computed like $\Delta\theta = \hat{H}^{-1}\nabla\mathcal{C}$, where $\hat{H} = (1 + \delta\mathrm{I})\nabla^2\mathcal{C}$. Then, the regularised least-square approximation should be:

$$\Delta\hat{\theta} = \left(\sigma_c^{-2}\hat{H} + \sigma_\theta^{-2}\mathbf{I}\right)^{-1}\left(\sigma_\theta^{-2}\mathbf{I}\mu_r + \sigma_c^{-2}\nabla\mathcal{C}\right). \tag{5.30}$$

  Inserting $\hat{\theta}_k$ into equation $\Delta\theta_k = \hat{H}^{-1}\nabla\mathcal{C}$, we can get:

$$\nabla\mathcal{C} = \hat{H}\left(\sigma_c^{-2}\hat{H} + \sigma_\theta^{-2}\mathbf{I}\right)^{-1}\left(\sigma_\theta^{-2}\mathbf{I}\mu_r + \sigma_c^{-2}\nabla\mathcal{C}\right),$$

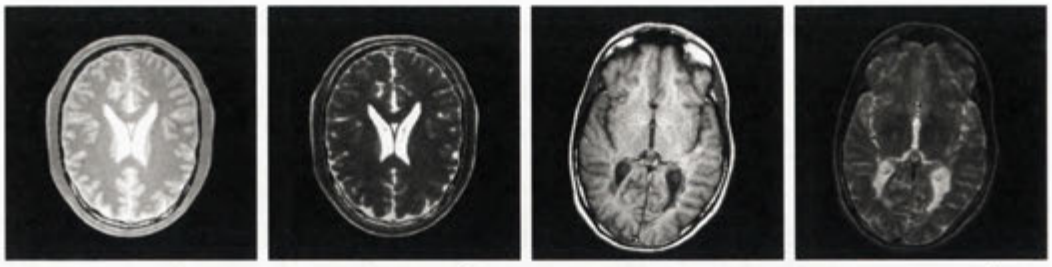
  where mean μ_k will be initially set as zero, during determination of RBF transformation. Therefore, the above equation can be simplified with elimination of μ_k term:

$$\nabla\mathcal{C} = \left(\sigma_c^{-2}\hat{H}\right)\left(\sigma_c^{-2}\hat{H} + \sigma_\theta^{-2}\mathbf{I}\right)^{-1}\nabla\mathcal{C},$$

  In accordance with generalized additive model (GAM), we can define function $\Psi_{\hat{H}}$ of RBFs, by using $R_{\hat{H}}$ to denote $(\sigma_c^{-2}\hat{H})(\sigma_c^{-2}\hat{H} + \sigma_\theta^2\mathbf{I})^{-1}$:

$$\Psi_{\hat{H}} = tr(R_{\hat{H}}). \tag{5.31}$$

  This is the sum of the eigenvalues of $R_{\hat{H}}$, which gives a measure of amount of fitting $R_{\hat{H}}$, the expected regularisation matrix for multi-modality learning varying dimension RBFs model.



(a) Proton-dens $I_1(\mathrm{x})$ (b) T2-weighted $\tilde{T}_1(\mathrm{x})$ (c) T1-weighted $I_2(\mathrm{x})$ (d) T2-weighted $\tilde{T}_2(\mathrm{x})$

Figure 5.3: Image-pairs of global transformation matrix recovery test

After modifying learning varying dimension RBFs to fit CCRE, the Reversible Jump Markov Chain Monte Carlo (RJMCMC), detailed described in section 4.5, is applied to determine transformation model parameter $M_k = \{\mu_k, k\}$. The pseudo-code of multi-modality learning varying dimension RBFs strategy is described in algorithm 5.1.

5.4 Experiments

Because multi-modality image registration is dominantly applied in medical image processing, in this section we choose two groups of neuroscience images to verify the performance of our CCRE based learning varying dimension RBF. These two groups of neuroscience images are provided by the southern medical university. Additionally, because computation of CCRE, its derivative and its second order terms are very expensive, stochastic image alignment strategy has been applied to reduce the cost of computation in the implementation of the CCRE. There are 49 15×15 patches extracted from image $I(\mathrm{x})$ as sub-image $sI(\mathrm{x})$, meanwhile same located patches are also extracted from template $\tilde{T}(\mathrm{x})$ as sub-template $s\tilde{T}(\mathrm{x})$. Then, entire learning varying dimension RBF is executed based on these sub-image and the sub-template.

In this section, there are two kinds of experiments will be executed, namely global transformation matrix recovery and free-form alignment evaluation.

- **Global Transformation Matrix Recovery**

  In this experiment, an initial 6 degree-of-freedom affine transformation matrix has been used to warp image $I(\mathrm{x})$. Then, we run CCRE based leaning varying RBF to find the correct alignment of two MR images, provided in figure 5.4. These two original images are components of a double-echo

Algorithm 5.1 Pseudo-code of RJMCMC for multi-modality learning varying dimension RBFs based on CCRE.

Input: Image $I(\mathrm{x})$ and template $\tilde{T}(\mathrm{x})$

Output: The model parameter M_k and coefficients θ_k

Initialisation:

1. Define variance σ_θ^2 from its prior probability: $\Gamma(\sigma_\theta^{-2}|10^{-3}, 10^{-3})$

2. Draw the out put coefficients θ by computed $\Delta\hat{\theta}$ equation 5.30 and Levenberg Marquardt optimisation, mentioned in section 5.5.

while do

(1). Draw a uniform random variable $dE \sim U(0,1)$.

(2). Propose the next state of the chain as follows:

if $dE < b_k$ **then**

perform **BIRTH** step. $dE < b_k + d_k$

Perform **DEATH** step.

else

Perform **MOVE** step.

end if

(3). Redraw the coefficients θ' as initialisation step 2.

(4). Draw an uniform random variable $dE \sim U(0,1)$.

if $dE < \alpha(E, E')$ (*i.e.* equation 4.23) **then**

Accept the proposed state.

else

Set the next state to be the current state.

end if

(5). raw the variance σ_θ^2 from its prior probability:

$\Gamma(\sigma_\theta^{-2}|10^{-3} + n/2, 10^{-3} + \varepsilon^2/2)$,where n is number of control points and ε^2 is the sum of squared residuals between current cross-cumulative residual entropy and previous value.

(6). Until convergence is assumed.

end while

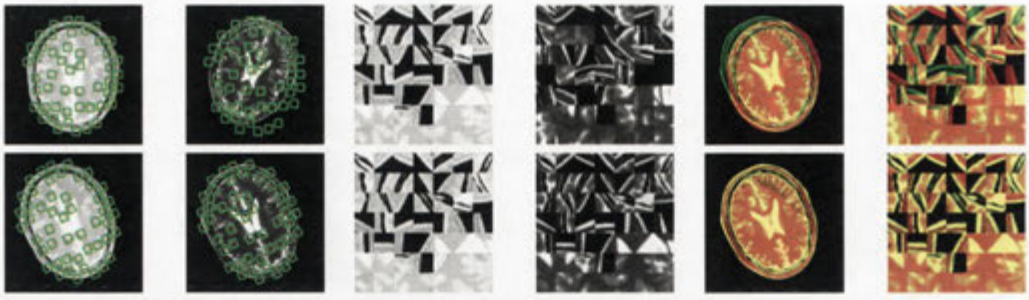
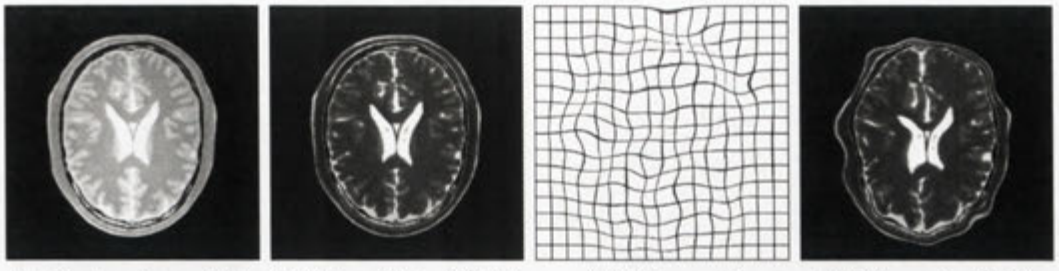


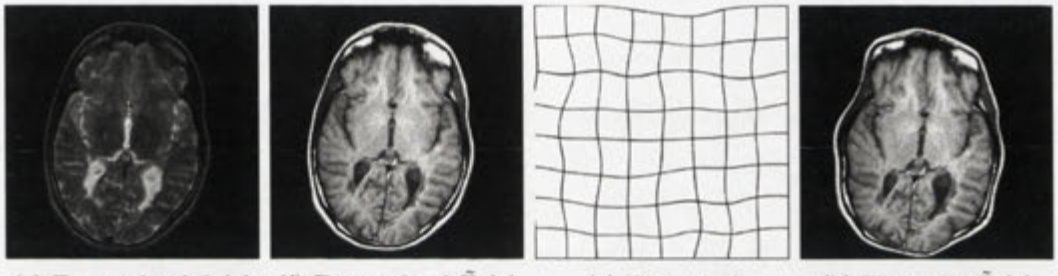
Figure 5.4: Global transformation matrix recovery test image-pairs result: The 1st row shows the situation of proton density image and T-2 weighted image initialised by affine transformation matrix 1 in table 5.2. The 2nd row presents the alignment result between them. Where 1st column contains image $I_1(\mathrm{x})$ and the template $\tilde{T}_1(\mathrm{x})$; the 2-nd column is sub-image $sI_1(\mathrm{x})$ and corresponding sub-template $s\tilde{T}_1(\mathrm{x})$, which are generated with 49 15×15 patches (Green squares indicate 49 patches in the in the image and the template); The 3-rd column consists of the alignment evaluation of both whole image pair ($I_1(\mathrm{x})$ and $\tilde{T}_1(\mathrm{x})$) and sub-image pair ($sI_1(\mathrm{x})$ and $s\tilde{T}_1(\mathrm{x})$).

Figure 5.5: Global transformation matrix recovery test image-pairs result: The 1st row shows the situation of proton density image and T-2 weighted image initialised by affine transformation matrix 2 in table 5.2. The 2nd row presents the alignment result between them. Where 1st column contains image $I_2(\mathrm{x})$ and the template $\tilde{T}_2(\mathrm{x})$; the 2-nd column is sub-image $sI(\mathrm{x})$ and corresponding sub-template $s\tilde{T}_2(\mathrm{x})$, which are generated with 49 15×15 patches (Green squares indicate 49 patches in the in the image and the template); The 3-rd column consists of the alignment evaluation of both whole image pair ($I_2(\mathrm{x})$ and $\tilde{T}_2(\mathrm{x})$) and sub-image pair ($sI_2(\mathrm{x})$ and $s\tilde{T}_2(\mathrm{x})$).

MR scan and obtained simultaneously by T1-weighted and T2-weighted, proton-density image and T2-weighted image, respectively.



(a) Proton-dens $I_1(x)$ (b) T2-weighted $\tilde{T}_1(x)$ (c) Warp-grid (d) Warped w$\tilde{T}_1(x)$



(e) T1-weighted $I_2(x)$ (f) T2-weighted $\tilde{T}_2(x)$ (g) Warp-grid (h) Warped w$\tilde{T}_2(x)$

Figure 5.6: Synthetic deformable warped image of free-form image alignment evaluation

No.	Affine Matrix	T1-T2	PD-T2
1	-0.0738 -0.4098 69.1023	-0.0739 -0.4097 69.0965	-0.0738 0.4098 69.1022
	0.3998 -0.1338 -26.8388	-0.3997 -0.1338 -26.8359	0.3998 -0.1338 -26.8383
2	-0.0002 -0.4319 -48.0457	-0.0002 -0.4319 -48.0417	-0.0003 0.4320 -48.0576
	-0.4419 -0.0602 71.4689	-0.4419 -0.0603 71.4831	-0.4421 0.0604 71.5088

Table 5.2: Results of alignment affine transformation matrix.

The experiment results is shown in table 5.2. From table 5.2, the affine transformation matrix recovery shows that our method can be used to accurately compute the parameters of global transformation. Compared with ground truth, the errors of estimations are really small. Figure 5.4 represents multi-modality global image registration procedure between proton density image $I_1(x)$ and T-2 weighted image $\tilde{T}_1(x)$, which is initialised by affine transformation matrix 1 in table 5.2. As mentioned before, stochastic image alignment strategy is applied to maximise CCRE with CCRE based learning-varying dimension RBF. The sub-image $sI_1(x)$ is generated with 49 15×15 patches extracted from image $I_1(x)$, while corresponding sub-template $s\tilde{T}_1(x)$ is extracted from template $\tilde{T}_1(x)$ with same located

patches. In figure 5.4, specifications of initialisation (Unit: **Bit**) MI:1.0182; CCRE:73.6405; MI of sub-image-pair:1.3983, CCRE:16.7582. While these specifications of result are MI:1.8584; CCRE:107.2888; MI of sub-image-pair:2.5047, CCRE:60.3360.

Figure 5.5 shows the multi-modality image registration procedure between T-1 weighted $I_2(\mathrm{x})$ and T-2 weighted image $\tilde{T}_2(\mathrm{x})$, which is initialised by affine transformation matrix 2 in table 5.3. In Figure 5.5, specifications of initialisation (Unit: **Bit**) MI:0.7219; CCRE:48.8317; MI of sub-image-pair:1.0211, CCRE:6.0029. While these specifications of result are MI:1.3579; CCRE:87.1345; MI of sub-image-pair:1.6715, CCRE:42.1710.

In global transformation case, each patch of sub-image $sI(\mathrm{x})$ will execute the same estimated affine transformation with entire image $I(\mathrm{x})$. This approach can fix the value of gradient matrix of each patch, to make it as an constant matrix and pre-computable out of iterations. Obviously, the differences between image $I_2(\mathrm{x})$ and initialized template $\tilde{T}_2(\mathrm{x})$ by affine transformation matrix 2 are much larger than that of affine that between image $I_1(\mathrm{x})$ and template $\tilde{T}_1(\mathrm{x})$. Figure 5.4 and the Figure 5.5 show the accuracy and robustness advantages of CCRE based learning varying dimension RBF.

- **Free-form Alignment Evaluation**

  In this section, we present the results of applying our CCRE based learning dimension RBF. The experimental results are presented for synthetic as real data. The two sets of experiment were done with synthetic free-form motion, which are shown in figure 5.6.

  Figure 5.7 and 5.8 represent the registration procedure between the proton density image $I_1(\mathrm{x})$ and warped T-2 weighted template $\mathrm{w}\tilde{T}_1(\mathrm{x})$, the T-1 weighted image $I_2(\mathrm{x})$ and warped T-2 weighted template $\mathrm{w}\tilde{T}_2(\mathrm{x})$, respectively.

  In figure 5.7, specifications of initialisation between image $I_1(\mathrm{x})$ and $\tilde{T}_1(\mathrm{x})$ (Unit: **Bit**) MI: 1.2109; CCRE: 82.6299; MI of sub-image-pair: 1.5675, CCRE: 21.7667. These specifications of result are MI: 1.8399; CCRE: 108.3385; MI of sub-image-pair: 2.6897, CCRE: 62.7284.

  In figure 5.8, specifications of initialisation between image $I_2(\mathrm{x})$ and $\tilde{T}_2(\mathrm{x})$ (Unit: **Bit**) MI: 0.9669; CCRE: 36.0361; MI of sub-image-pair: 1.2763, CCRE: 11.3373. These specifications of result are MI: 1.3702; CCRE: 48.5333; MI of sub-image-pair: 1.8012, CCRE: 23.4690.

Figure 5.7: The procedure of synthetic free-form transformation between the proton density image $I_1(\mathrm{x})$ and the warped T-2 weighted template $\mathrm{w}\tilde{T}_1(\mathrm{x})$: The 1st row shows the situation of proton density image and warped T-2 weighted image initialised by the synthetic warping grid in figure 5.6-c. The 2nd row presents the alignment result between them. Where 1st column contains image $I_1(\mathrm{x})$ and the template $\mathrm{w}\tilde{T}_2(\mathrm{x})$; the 2-nd column is sub-image $sI_1(\mathrm{x})$ and corresponding sub-template $s\mathrm{w}\tilde{T}_2(\mathrm{x})$, which are generated with 49 15×15 patches (Green squares indicate 49 patches in the in the image and the template); The 3-rd column consists of the alignment evaluation of both whole image pair ($I_1(\mathrm{x})$ and $\mathrm{w}\tilde{T}_2(\mathrm{x})$) and sub-image pair ($sI_1(\mathrm{x})$ and $s\mathrm{w}\tilde{T}_1(\mathrm{x})$).

Figure 5.8: The procedure of synthetic free-form transformation between the T-1 weighted image $I_2(\mathrm{x})$ and the warped T-2 weighted template $\mathrm{w}\tilde{T}_2(\mathrm{x})$: The 1st row shows the situation of proton density image and T-2 weighted image initialised by the synthetic warping grid in figure 5.6-g. The 2nd row presents the alignment result between them. Where 1st column contains image $I_2(\mathrm{x})$ and the template $\mathrm{w}\tilde{T}_2(\mathrm{x})$; the 2-nd column is sub-image $sI_2(\mathrm{x})$ and corresponding sub-template $s\mathrm{w}\tilde{T}_2(\mathrm{x})$, which are generated with 49 15×15 patches (Green squares indicate 49 patches in the in the image and the template); The 3-rd column consists of the alignment evaluation of both whole image pair ($I_2(\mathrm{x})$ and $\mathrm{w}\tilde{T}_2(\mathrm{x})$) and sub-image pair ($sI_2(\mathrm{x})$ and $s\mathrm{w}\tilde{T}_2(\mathrm{x})$).

5.5 Conclusions

By incorporating CCRE into learning varying dimension RBF as a similarity measure, we have presented this CCRE based learning RBF algorithm for multi-modality free-form image registration. Through the combination between CCRE and learning RBF, our algorithm inherits advantages from both. Incorporation of CCRE increases robustness of this multi-modality image registration algorithm and makes it more efficient to seek the optimal warping parameters. Meanwhile, adoption of learning varying dimension RBF can significantly eliminate over-fitting and under-fitting in free-form transformation and increase the accuracy of parameter estimation at each iteration of optimisation.

Chapter 6

Conclusion

The primary goal of this Master thesis research is to improve the accuracy of non-rigid dense image registration through alleviating the over-fitting and under-fitting issues in image registration.

After reviewing classical image registration algorithms in chapter 2, we respectively addressed causes for the over-fitting and under fitting in global image registration and the non-rigid registration in chapter 3.

In global image registration, the over-fitting and under-fitting are essentially caused by the ill-posed problems during the warping parameter estimation procedure. To satisfy two constraints of optic-flow estimation, the warping parameter estimation becomes an optimisation for an over-determined system of linear equations. Thus, least-square method is adopted in this warping parameter optimisation. However, due to the ill-poseness, warping parameter may be estimated incorrectly at each optimisation iteration. The unreliable iterative warping parameter will cause an unstable convergence. In other words, optimisation may be trapped in a local minimum. Furthermore, an unstable convergence will directly lead to the over-fitting or under-fitting.

To resolve the ill-posed problems, Tikhonov regularisation is commonly used as the first choice. This regularisation improves the conditioning of the least-square optimisation, thus enabling an explicit numerical solution. Through incorporating Tikhonov regularisation into traditional Lucas-Kanade image registration, the probabilistic image registration successfully overcomes the ill-poseness and increases stability of optimisation convergence, thus producing an accurate registration result for global image registration.

However, for non-rigid image registration, ill-poseness in warping parameter estimation is merely one reason to cause the over-fitting and under-fitting. The

limitation of non-rigid geometric transformation is the other cause of the occurrence of over-fitting and under-fitting. This limitation manifests as the non-rigid geometric transformation is incapable to produce a perfect non-rigid displacement field by a set of control points to express the real non-rigid transformation between the sensing image and the template. Displacements of some control points are estimated incorrectly, because they are located in the featureless regions. Accordingly, these incorrect control point displacements will produce some local misalignments. These local misalignments are caused by the over-fitting and under-fitting.

Therefore, if we want to prevent the occurrence of the over-fitting and under-fitting in the non-rigid image registration, the ill-poseness and the low expression power of certain transformations have to be resolved altogether.

- **Two-phase probabilistic second-order Demons Algorithm**

  In chapter 4, we first addressed above over-fitting and under-fitting in Demons Algorithm, and then provided our improved Demons Algorithm to solve the problem.

  For Demons Algorithm, it is observed that this algorithm is particularly successful in dealing with non-rigid registration with images having many homogenous regions. However it fails often in handling texture-rich non-rigid image registration. The limitation of Demons Algorithm in texture-rich non-rigid image registration illustrates that Demons Algorithm tends to over-fit the many spurious edges inside the texture-rich regions. This limitation will lead to erroneous thermodynamic "forces", thus making misalignments.

  To overcome these drawbacks, we adopted the divide-and-conquer strategy to resolve ill-posed problems and remove the limitation of geometric transformation, thus enabling our two-phase probabilistic second-order Demons Algorithm to increase performance. Firstly, incorporating prior information into the second-order demons is used to establish a probabilistic second-order demons algorithm to resolve ill-posed problems and reliably estimate warping parameters for the rectangle free-form transformation. Consequently, two-phase deformation strategy is utilised to overcome the limitation of the rectangle free-form transformation. By adding a dense displacement field estimation, our two-phase probabilistic second-order Demons Algorithm can recover the misalignment caused by incorrect

displacements of some control points. Additionally, thanks to preventing occurrence of the over-fitting and under-fitting, our two-phase second-order Demons Algorithm can not only deal with non-rigid image registration with homogenous regions, but also tackle the registration for texture-rich image.

- **Learning varying dimension RBF**

  The two-phase probabilistic second-order Demons Algorithm has provided a successful divide-and-conquer strategy to prevent the over-fitting and under-fitting issues. However, the above algorithm has not yet proposed a reliable non-rigid transformation to interpret deformation between the sensed image and the template.

  In order to develop a reliable non-rigid transformation model, we proposed learning non-rigid image registration based on radial basis transformation to avoid occurrence of the over-fitting and under-fitting. Traditional radial basis function transformation is mainly based on a fixed dimension parametric transformation model. Thanks to RBF, technically, this transformation is also able to express any non-rigid transformation with the appropriate number and locations of control points. Unfortunately, how to find these appropriate control points is a difficult task. Moreover, the over-fitting and under-fitting usually occur in the non-rigid image registration with inappropriate radial basis transformation.

  In order to provide a means to determine an appropriate radial basis transformation and prevent the over-fitting and under-fitting, we proposed an efficient learning varying dimension RBF. Different from the two-phase probabilistic second-order Demons Algorithm, learning varying dimension RBF simultaneously completes warping parameter estimation and radial basis transformation establishment. To resolve ill-posed problem, the Bayesian framework is applied in this image registration algorithm. Yet, the control points of this radial basis transformation will be inserted by a learning procedure from a latent control points set. Learning approach is capable of finding the best number and the locations of control points. This means that an appropriate non-rigid can be adaptively established to represent any non-rigid transformation by this way. Thus the limitation of non-rigid geometric transformation can be fundamentally overcome by an appropriate transformation. In short, this approach can establish a numerical transformation model to interpret any deformation between the sensing image and

the template.

- **CCRE based Learning varying dimension RBF**

  After successfully providing a non-rigid transformation model determination approach, we then integrate this transformation determination approach with a similarity measure named CCRE to deal with non-rigid multi-modality image registration.

  For multi-modality image registration, mutual information is the commonly used similarity measure. Mutual information utilises the joint probability distribution functions of intensities between the sensing image and the template and the independent probability distribution functions of intensities in each of them to measure the similarity between the sensing image and the template. The larger value of mutual information between the sensing image and the template means higher similarity between these two images. Therefore, the optimal warping parameter of transformation can be estimated through maximising mutual information. However, the change of mutual information corresponding to warping parameters is not smooth. This makes the multi-modality image registration based on mutual information usually trapped into local minima and fail to register the two images.

  To overcome this disadvantage of mutual information, CCRE was proposed. Similar to mutual information, the optimal warping parameters of transformation can be obtained via maximising CCRE. However, compared with mutual information, CCRE used cumulative distribution functions instead of probability distribution functions in mutual information. Because cumulative distribution function is smoother than probability distribution function. CCRE as a similarity measure can be optimised easily without some annoying ambiguities. Therefore, we choose this CCRE to integrate with learning varying RBF to successfully conduct non-rigid multi-modality image registration.

  Learning varying dimension RBF with CCRE can achieve non-rigid multi-modality image registration. However, the computation cost of CCRE is expensive, and it needs to be calculated at each iteration. Therefore, how to accelerate computation will be a critical problem in application. To further reduce the running time, exploiting the potentials of graphics processing units (GPUs) for speeding up multi-modality learning varying dimension RBF with CCRE is a possible future work [7] [19] [2].

Bibliography

[1] T. Amiaza, E. Lubetzkyb, and N. Kiryatia. Coarse to over-fine optical flow estimation. *Pattern Recognition*, pages 2496–2503, 2007.

[2] R.E. Ansorge, S.J. Sawiak, and G.B. Williams. Exceptionally fast non-linear 3d image registration using gpus. In *2009 IEEE NUCLEAR SCIENCE SYMPOSIUM and MEDICAL IMAGING CONFERENCE - IEEE-NSS/MIC 2009*, 2009.

[3] M. Asadi and Y. Zohrevand. On the dynamic cumulative residual entropy. *Journal of Statistical Planning and Inference*, 137:1931–1941, 2007.

[4] S. Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Report: CMU-RI-TR-02-16, Carnegie Mellon University, 2002.

[5] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, October 2007.

[6] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registrations. In *In proceedings of the 15th British Machine Vision Conference,*, volume 2, pages 899–908, London, UK, September 2004.

[7] D. Blythe. Rise of the graphics processor. In *the IEEE*, volume 96, pages 761–778, 2008.

[8] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[9] Jean Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Jean-Yves Bouguet, 2002.

[10] D.S. Broomhead and David Lowe. Multi-variable functional interpolation and adaptive networks. *Complex System*, 2:321–355, 1988.

[11] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations.* Cambridge University Press, 2003.

[12] David Capel. *Image mosaicing and super-resolution.* Springer, 2004.

[13] Simon R. Cherry. Multimodality imaging: Beyond pet/ct and spect/ct. *Nuclear Medicine*, 39:348–353, 2009.

[14] Kong Man Cheung, Simon Baker, and Takeo Kanadey. Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Computer Vision*, 62:221–247, 2005.

[15] Kong Man Cheung, Simon Baker, and Takeo Kanadey. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63:225–245, 2005.

[16] Peter H. Cossmann. Advances in image-guided radiotherapy - the future is in motion. *European Oncology Review*, July 2005.

[17] T Durrant-Whyte, H.; Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 13:99–110, 2006.

[18] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.

[19] F.Xu and K. Mueller. Accelerating popular tomographic. *IEEE transcations on Nuclear Science*, 52:654–663, 2005.

[20] L. Garcin, X. Descombes, H. Le Men, and J. Zerubia and. Building detection by markov object processes. In *Proceedings of 2001 IEEE International Conference on Image Processing*, volume 2, pages 565–568, Thessaloniki, Greece, October 2001.

[21] A. Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19:459466, 1986.

[22] A. Goshtasby. Registration of images with geometric distortions. *IEEE Trans. Geosci. Remote Sensing*, 26:60–64, 1988.

[23] Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[24] Alexandre Guimond, Alexis Roche, Nicholas Ayache, and Jean Meunier. Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. *IEEE Trans. on Medical Imaging,*, 20:58–69, 2007.

[25] R. L. Harder and R. N. Desmarais. Interpolation using surface splines. *J. Aircraft*, 9:189–191, 1972.

[26] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artifical Intelligence*, 17:185–203, 1981.

[27] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *International Conference on Computer Vision*, pages 684–689, Vancouver, Canada, July 2001.

[28] Seth C. Koterba, Simon Baker, Iain Matthews, Changbo Hu, Jing Xiao, Jeffrey Cohn, and Takeo Kanade. Multi-view aam fitting and camera calibration. In *Proc. International Conference on Computer Vision*, pages 511–518, October 2005.

[29] Kang Li, Mei Chen, and Takeo Kanade. Cell population tracking and lineage construction with spatiotemporal context. In *Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 295–302, 2007.

[30] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Journal on Computer Vision*, pages 674–679, 1981.

[31] G. Q. Maguire, M. E. Noz Jr., H. Rusinek, J. Jaeger, E.L. Kramer, J. J. Sanger, and G. Smith. Graphics applied to medical image registration. *IEEE Comput. Graph*, 11:20–29, March 1991.

[32] J. Meinguet. An intrinsic approach to multivariate spline interpolation at arbitrary points. *Polynomial and Spline Approximation*, 9:163–160, 1972.

[33] J. Meinguet. *Polynomial and Spline Approximation*, chapter An intrinsic approach to multivariate spline interpolation at arbitrary points, pages 163–190. D. Reidel Publishing Company, Boston, 1978.

[34] N. Metropolis, A.W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, June 1953.

[35] N. D. Molton, A. J. Davison, and I. D. Reid. Parameterisation and probability in image alignment. In *proceedings Asian Conference on Computer Vision*, Jeju, Korea, Jan 2004.

[36] Nicholas Molton, Andrew Davison, and Ian Reid. Parameterisation and probability in image alignment. OUEL Report: 2266/03, University of Oxford, 2003.

[37] G. M. Nielson. Scattered data modeling. *IEEE Comput. Graph. Appl.*, 13:60–70, 1993.

[38] X. Pennec, P. Cashier, and N. Ayache. Understanding the "demon's algorithm": 3d non-rigid registration by gradient descent. *MICCAI99*, pages 597–605, 1999.

[39] M. J. D. Powell. *Radial basis functions for multivariable interpolation: a review*, pages 143–167. Clarendon Press, New York, NY, USA, 1987.

[40] M. Rao, Y. Chen, B.C. Vemuri, and F. Wang. Culumative residual entropy, a new measure of information. *IEEE transcations on Information Theory*, 50(6):1220–1228, 2004.

[41] Xiaofeng Ren. Local grouping for optical flow. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[42] R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo Method.* John Wiley & Sons, New York, 2 edition, 2007.

[43] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.

[44] L. L. Schumaker. *Spline Functions: Basic Theory.* John Wiley & Sons, New York, 1981.

[45] Jianbo Shi and Carlo Tomasi. Good feature to track. *1994 IEEE Conference on Computer Vision and Pattern Recoginition*, pages 593–600, 1994.

[46] E. V. Shikin and A. I. Plis. *Handbook on Splines for the User.* CRC Press, New York, 1995.

[47] D. Sun, S. Roth, J.P. Lewis, and M. Black. Learning optical flow (srf-lfc). In *European Conference on Computer Vision-ECCV'2008*, 2008.

[48] A. Tarantola. *Inverse Problem Theory.* Society for Industrial and Applied Mathematics, 2004.

[49] J.P. Thirion. Image mathching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[50] P. Viola and W. Wells III. Alignment by maximization of mutual information. *IJCV97*, 24(2):137–154, 1997.

[51] G. Wahba. *Spline Models for Observational Data.* Philadelphia: Society for Industrial and Applied Mathematics, 1990.

[52] F. Wang and B.C. Vemuri. Non-rigid multi-modal image registration using cross-cumulative residual entropy. *International Journal of Computer Vision*, 74(2):0920–5691, 2007.

[53] Di YANG and Hongdong LI. Learning varying dimension radial basis functions for deformable image alignment. In *2009 IEEE International Conference on Computer Vision Workshop*, Kyoto, 2009.

[54] Di YANG and Hongdong LI. A probabilistic demons algorithm for texture-rich image registration. In *2009 IEEE International Conference on Image Processing*, Carlo, 2009.

[55] Ming Ye, Robert M. Haralick, and Linda G. Shapiro. Estimating piecewise-smooth optical flow with global matching and graduated optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1625–1630, 2003.

[56] Y.M.Zhu. Volume image registration by cross-entropy optimization. *IEEE Transactions on Medical Imaging*, 21:174–180, 2002.

[57] B. Zitová and J. Flisser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, June 2003.

[58] Barbara Zitov and Jan Flusser. Image registration methods: a survey. *Image Vision Computing*, 21:977–1000, 2003.