

A Study of Efficiency, Accuracy, and Robustness in Intensity-Based Rigid Image Registration

by

Lin Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2008

© Lin Xu 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Image registration is widely used in different areas nowadays. Usually, the efficiency, accuracy, and robustness in the registration process are concerned in applications. This thesis studies these issues by presenting an efficient intensity-based mono-modality rigid 2D-3D image registration method and constructing a novel mathematical model for intensity-based multi-modality rigid image registration.

For mono-modality image registration, an algorithm is developed using RapidMind Multi-core Development Platform (RapidMind) to exploit the highly parallel multi-core architecture of graphics processing units (GPUs). A parallel ray casting algorithm is used to generate the digitally reconstructed radiographs (DRRs) to efficiently reduce the complexity of DRR construction. The optimization problem in the registration process is solved by the Gauss-Newton method. To fully exploit the multi-core parallelism, almost the entire registration process is implemented in parallel by RapidMind on GPUs. The implementation of the major computation steps is discussed. Numerical results are presented to demonstrate the efficiency of the new method.

For multi-modality image registration, a new model for computing mutual information functions is devised in order to remove the artifacts in the functions and in turn smooth the functions so that optimization methods can converge to the optimal solutions accurately and efficiently. With the motivation originating from the objective to harmonize the discrepancy between the image presentation and the mutual information definition in previous models, the new model computes the mutual information function using both the continuous image function representation and the mutual information definition for continuous random variables. Its implementation and complexity are discussed and compared with other models. The mutual information computed using the new model appears quite smooth compared with the functions computed by others. Numerical experiments demonstrate the accuracy and efficiency of optimization methods in the case that the new model is used. Furthermore, the robustness of the new model is also verified.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Justin Wan, for his warm guidance during my research and life at the University of Waterloo. The critical thinking and rigorous academic writing training offered by him will be rewarding for me for ever. His eternal energy and enthusiasm in academic research will motivate me all the time.

My deepest gratitude also goes to my thesis committee members, Yuying Li and Jeffery Orchard, for their time and energy on this thesis. I also want to appreciate their support during the process of my research. Without their help, it is not possible for me to finish this thesis smoothly.

Last but not least, I feel a deep sense of gratitude for my family and friends for their unflagging care and encouragement. Thanks to my parents and my girlfriend, Xiaowen, for their endless love and support. Thanks to my past and present colleagues in Scientific Computing Laboratory for providing me with a friendly and stimulating working environment.

Dedication

This is dedicated to my parents, Shuguo Xu and Hui Zhang.

Contents

List of Tables	ix
List of Figures	xii
1 Introduction	1
2 Image Registration	6
2.1 General Description	6
2.2 Classification	7
2.2.1 Registration Bases	7
2.2.2 Image Dimensions	7
2.2.3 Transformation Types	8
2.2.4 Modalities	8
2.3 Mathematical Formulation	10
2.3.1 Image Functions	10
2.3.2 Optimization	11
2.3.3 Similarity Measures	11
3 GPU-based Acceleration	13
3.1 2D-3D Image Registration	13
3.2 Gauss-Newton Method	16
3.3 Parallel Computing	17
3.3.1 Graphics Processing Units	18

3.3.2	RapidMind Multi-Core Development Platform	19
3.4	Numerical Results	20
3.4.1	Synthetic Data	21
3.4.2	Clinical Data	24
4	Multi-modality Image Registration	26
4.1	Mutual Information	26
4.1.1	Discrete Cases	27
4.1.2	Continuous Cases	29
4.1.3	Mutual Information for Image Registration	30
4.2	Probability Distributions	31
4.2.1	Interpolation-based Models	32
4.2.2	Partial Volume Models	36
4.3	Interpolation Artifacts	39
4.4	Optimization Methods	43
5	Continuous Models	47
5.1	Motivation	47
5.2	One Dimensional Image Registration	48
5.2.1	Image Functions	48
5.2.2	Individual Probability Density Functions	49
5.2.3	Joint Probability Density Functions	53
5.2.4	Individual Entropies	57
5.2.5	Joint Entropies and Mutual Information	59
5.2.6	Complexity Analysis	60
5.2.7	Examples	61
5.3	Two Dimensional Image Registration	65
5.3.1	Translation	66
5.3.2	Rotation	69
5.3.3	Complexity Analysis	69
5.4	Relationship with Partial Volume Models	70

6	Numerical Results for Continuous Models	72
6.1	Images	72
6.2	Smoothness	72
6.2.1	Image Modalities	75
6.2.2	Image Resolution	75
6.3	Sub-pixel Accuracy	80
6.4	Optimization Performance	80
6.5	Numbers of Strips	86
6.6	Numbers of Cells	87
7	Conclusions	89
	References	90

List of Tables

3.1	Average CPU times for each iteration of the image registration process.	23
3.2	Total CPU times for the entire image registration process.	25
6.1	The trust region performance for different models with different initial guesses (the optimal solutions = $(0, 0)$).	84
6.2	The Nelder-Mead performance for different models with different initial guesses (the optimal solutions = $(0, 0)$).	85
6.3	Optimization performance for the continuous model (the optimal solutions = $(0, 0, 0)$).	86
6.4	Error analysis.	87

List of Figures

2.1	T1- (upper left), T2- (upper right), PD- (lower left) weighted MRIs and CT (lower right).	9
3.1	The iteration scheme of 2D-3D medical image registration.	14
3.2	The framework of DRR construction.	15
3.3	The schematic of the ray casting algorithm.	16
3.4	NVIDIA GeForce 8800 GTX.	19
3.5	The architecture of RapidMind. Copyright 2008 RapidMind Inc. All rights reserved.	20
3.6	The initial guess of the DRR (left) and simulated target image using the transformation parameter set \mathbf{s} as $(5^\circ, 5^\circ, 5^\circ, 0.1, 0.1, 0.1)^T$ (right).	22
3.7	A tripod fracture of a skull	24
4.1	The relationship between the information of X and Y .	29
4.2	The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) using the histogramming model, as one of the interpolation-based models.	34
4.3	The interpolated intensity value in the transformed template image may cross the boundaries of bins and lead to discontinuous change of the probabilities.	35
4.4	The flow chart of interpolation-based models.	37
4.5	A pixel in the transformed template image is located among four pixels in the target image.	39

4.6	The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) using the partial volume model.	40
4.7	Mutual information using the histogramming model with the numbers of bins as 64 (upper) and 32 (lower) for both the target and the transformed template images. Although the mutual information function gets smoother with the increase of bin sizes, the interpolation artifacts can still be observed.	44
4.8	Mutual information using the partial volume model with the numbers of bins as 64 (upper) and 32 (lower) for both the target and the transformed template images. Although the mutual information function gets smoother with the increase of bin sizes, the interpolation artifacts can still be observed.	45
5.1	The target image in the case that $I_{i-1}^g < I_i^g$	50
5.2	The target image in the case that $I_{i-1}^g > I_i^g$	51
5.3	The target image in the case that $I_{i-1}^g = I_i^g$	51
5.4	Images in the case that $K_{i-1}^g \neq K_i^g$ or $K_{i-1}^f \neq K_i^f$	54
5.5	The domain of the joint PDF where $p_i^{fg} \neq 0$ is denoted by \mathcal{C}_i in the case that $K_{i-1}^g \neq K_i^g$ or $K_{i-1}^f \neq K_i^f$	55
5.6	Images in the case that $K_{i-1}^g = K_i^g$ and $K_{i-1}^f = K_i^f$	55
5.7	The point where p_i^{fg} is a Dirac delta function in the case that $K_{i-1}^g = K_i^g$ and $K_{i-1}^f = K_i^f$	56
5.8	An example of an image function.	61
5.9	The PDF associated with the image in each subinterval.	62
5.10	The PDF associated with the whole image.	63
5.11	The PDF is approximated as a constant function in each cell.	64
5.12	The approximate probabilities associated with each bin in the PDF domain with respect to the horizontal translation.	64
5.13	The pixels in the target and the transformed template images generally divide the whole image domain into sub-domains.	67
5.14	Each sub-domain is divided into strips with equal width.	68

5.15	A box domain of the transformed template image is generally located among four adjacent box domains of the target image.	71
6.1	The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) with respect to the horizontal translation.	73
6.2	The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) with respect to the vertical translation.	74
6.3	The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the histogramming model.	76
6.4	The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the partial volume model.	76
6.5	The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the continuous model.	77
6.6	The mutual information taking the images in the resolution of 256×256 (upper), 128×128 (middle), and 64×64 (lower) using the histogramming model.	78
6.7	The mutual information taking the images in the resolution of 256×256 (upper), 128×128 (middle), and 64×64 (lower) using the continuous model.	79
6.8	The mutual information with respect to the horizontal translation using the histogramming model (upper), the partial volume model (middle), and the continuous model (lower).	81
6.9	The mutual information computed using 4×4 cells (upper), 8×8 cells (middle), and 16×16 cells (lower).	88

Chapter 1

Introduction

Nowadays, digital images have been widely used to convey information as a kind of media. In the field of medicine, physicians often utilize digital images to collect clear information about some parts of the patients' bodies which are not externally visible. This helps physicians analyze certain kinds of disease and make future treatments. In the realm of earth remote sensing, some characters of the earth, which are not directly observable for humans, are obtained by generating digital images using sensors positioned in the air or in earth orbit. Those images are important data for future analysis (e.g., weather report). Very often, to receive comprehensive information of an huge object, or compare information of the same object at different times, many different images are generated from the same or different sensor devices. To combine different information of the same object from different images, images are required to be aligned in the same coordinates. This process is known as image registration.

Image registration is helpful and sometimes required in practice. Because of its importance, different people have paid and are paying considerable attention to that with different emphases: physicians focus on different impacts of different anatomical regions to the performance of image registration; statisticians desire to obtain statistical characters from images to facilitate registration processes; the practicality attracts a lot of engineers to build stable and easy-to-use products for conducting image registration; applied mathematicians and computer scientists concentrate on the mathematical models and algorithms behind image registration processes.

Image registration is broad and can be classified into different categories based on different criteria. Based on whether deformation of images is allowed, image

registration can be classified into rigid image registration and non-rigid image registration. In rigid image registration, images are considered as rigid objects and no deformation of images is allowed. Because of their rigid characters, the transformations of the images are restricted to rotation and spatial translation. In comparison, non-rigid image registration treats images as soft objects and deformation of images is allowed. In this thesis, we mainly focus on rigid image registration.

Image registration in the case that images are generated from the same sensor device with the same physical parameters is called mono-modality image registration. In this case, one image is the same as, or similar to, others when they are aligned. This kind of image registration problems has been solved well by standard methods. However, for sequential programs, computational complexity dramatically increases when one or more than one image is in high dimensions. For example, in 2D-3D image registration, the target image is in 2D and the template image is in 3D. In this kind of registration problems, volume rendering and the optimization processes are normally computationally intensive and require a significant amount of time. We consider parallel computing as a way to improve the efficiency.

Hardware acceleration methods using hundreds or even thousands of processors have been studied and employed for image registration [6]. Though it has been reported with success in many cases, the physical size of large computer clusters limits their use in clinical applications. Multi-core processing has become an attractive alternative in the past few years. The multi-core architectures exploit hardware parallelism on the chipsets while maintaining a small form factor. Ohara *et al.* [18] implement mutual-information-based multi-resolution registration algorithms on Cell Broadband Engines. However, it is only for 3D-3D registration, and the random sampling strategy is used due to the memory limit of Cell Broadband Engines. Chisu [5] investigates both graphics processing unit (GPU) and central processing unit (CPU) acceleration techniques for 2D-3D rigid image registration. The author applies GPUs for solving the optimization problems in registration processes, but still use CPUs for the volume rendering computation. Based on our experiments, the CPU time for the volume rendering computation is a significant part in the registration process; see Section 3.4. Hence, an efficient GPU computation for volume rendering becomes crucial for optimal efficient performance.

In this thesis, a parallel 2D-3D image registration method is developed using GPUs implemented by RapidMind Multi-core Development Platform as a way to improve the efficiency. In contrast with the previous work [5, 18], our efficient image registration method [32] performs almost the entire 2D-3D registration process

using GPUs to fully exploit the multi-core parallel efficiency. Numerical results are also included to verify this point.

Multi-modality image registration refers to the case that the images for registration are generated from different sensor devices (e.g., X-rays and magnetic resonance imaging (MRI)) or the same sensor device with different physical parameters (e.g., T1-weighted MRI, T2-weighted MRI, and proton density (PD)-weighted MRI). Comparing to mono-modality image registration, multi-modality image registration is a challenging task since there is no direct relationship in pixel intensity values between images. Mutual information, which first originates from information theory and statistics, was introduced to the field of multi-modality image registration as the similarity measure by Viola and Wells [28] and Collignon, Maes, Delaere, Vandermeulen, Suetens, and Marchal [8] independently in 1995. In mutual information, image intensity values are treated as random variables and statistical characters of the intensity values are exploited in the similarity measure to conduct image registration. The optimal transformation solutions are computed by maximizing the mutual information. In this process, the evaluation of mutual information requires the computation of probability density functions (PDFs) of intensity values in images. However, because only discrete intensity values can be acquired, PDFs can only be estimated using discrete probability distributions. Viola *et al.* and Collignon *et al.* provide different approaches to compute probability distributions as ways to estimate PDFs. The mutual information is then computed by exploiting the probability distributions. Viola *et al.* devise a histogramming model. Although it is straightforward and computationally efficient, the resulting mutual information is not a smooth function of the transformation parameters (the first derivative is not continuous), which hampers the convergence to the optimal solutions of optimization methods. Collignon *et al.* apply the Parzen windowing strategy and the model is qualitatively better than, though not as computationally efficient as, the model Viola *et al.* provide. However, the mutual information is still not smooth. Maes, Collignon, Vandermeulen, Marchal, and Suetens [14] introduce the partial volume model and the mutual information is generally much smoother than before. However, when all or most of the pixel locations in the transformed template image are aligned with the pixel locations in the target image during the transformation, the mutual information function becomes non-smooth. This kind of phenomenon is known as interpolation artifacts and it can hamper the performance of optimization methods. In fact, this phenomenon also exists in the previous two models. Chen and Varshney [3] suggest a generalized partial volume model using a way similar to B-spline interpolation in order to spread the weight of image

pixel intensity values, as an extension of the partial volume model. Albeit this model helps reduce the interpolation artifacts, the interpolation artifacts cannot be totally removed. Other suggestions for reducing interpolation artifacts include resizing the pixel size [27], jittering or blurring images [27], and balancing the size of bins for computing probability distributions [10, 26]. However, none of them has fundamentally solved the problem. Any derivative-based optimization may easily get stuck in one of those artifacts due to the non-smoothness. Because of that reason, derivative-free methods, such as simplex methods [3, 4], Powell’s method [8], simulated annealing [34], and genetic algorithms [33], are used in order to obtain optimal solutions. However, it is not guaranteed that those optimization methods always converge to the optimal solutions; failure cases can still occur based on our experience.

This thesis specifically focuses on analyzing the reason why interpolation artifacts occur and devising a new model to totally remove the interpolation artifacts and in turn smooth the mutual information. After realizing that the artifacts happen because discrete interpolated intensity values are extracted from the images for computing the mutual information functions based on discrete random variables, we suggest constructing continuous image functions, using all the information of those continuous functions to compute the PDFs analytically, and then computing the mutual information based on continuous random variables. Although similar ideas have also been mentioned in [11] and [22], no concrete proposal has been suggested due to the computational difficulty to evaluate the image functions in continuous domains. To avoid that bottleneck, we assume the image functions to be simple interpolants for observable pixel values. In this case, computing mutual information functions numerically using analytical PDFs for 1D images is possible. The resulting mutual information functions appear smooth without any artifact. However, computing mutual information numerically using the analytical PDFs for higher dimensional (2D or higher) images is challenging because the corresponding analytical PDFs are more difficult to define. Alternatively, we reduce the task for computing PDFs for higher dimensional images to the one for computing several PDFs for 1D images. We evaluate the mutual information functions and no artifact is observable. To verify the benefit of the smoothness of the mutual information functions computed in our new model, a trust region optimization method (a derivative-based method) and the Nelder-Mead method (a derivative-free method) are applied to the new model. Fast convergence and high accuracy are obtained in the experiments. The robustness of the model is verified at the same time.

The thesis is organized as follows. Chapter 2 includes an overview of image

registration problems. Following a general description, different categories of image registration classified using different criteria are introduced. In addition, mathematical formulation is constructed by describing image registration as an optimization problem. Chapter 3 shows a way to improve the efficiency using GPU-based parallel computing. The 2D-3D image registration problem is introduced. The parallel computing is recognized as a way to accelerate the registration process. Numerical experiments are also included to verify this point. Chapter 4 discusses multi-modality image registration. Mutual information, as a different similarity measure from those in mono-modality image registration, is defined. Usually, mutual information functions are defined as the ones of the probability distributions of image intensity values. Different approaches to compute the probability distributions are shown based on different models. The interpolation artifact phenomenon in those models is then demonstrated and analyzed. Some optimization methods which are used for solving the registration problems are also described. Chapter 5 presents a new model to solve the interpolation artifact problems arising from previous models. The motivation of constructing the new model is explained. A concrete process of constructing the new model is described in the case that the images are in one or two dimensions. The complexity of the new model is also analyzed. Moreover, we discuss the relationship between the partial volume model and our continuous model. Chapter 6 contains numerical experiments for verifying the advantages of the new model by comparing it with other models. The mutual information functions turn out to be quite smooth using our new model. At the same time, the robustness of the new model is verified by changing the modalities and resolution of images. By applying our new model to compute the mutual information, it can be observed that optimization methods converge to the optimal solutions both efficiently and accurately. The numerical error analysis and the effect of different parameters in our model are also examined. Finally, conclusions and future work are comprised in Chapter 7.

Chapter 2

Image Registration

The objective of image registration is to align one image with another using the optimal transformation in order to combine information from both images. Image registration has been widely and successfully used in a lot of fields, such as medical applications [15] and remote sensing [21]. In this chapter, we start with a general description of image registration from the perspective of its applications, followed by its classification with different criteria. Finally, we introduce the mathematical criteria which are used to find the optimal transformation parameter sets.

2.1 General Description

Image registration is used to align a pair of images in the same coordinate system in order to get comprehensive information from different images. For example, physicians make clinical plans and decisions by comparing a medical image including an organ of a particular patient with another medical image including a normal one. However, the anatomical information in those two images may not be necessarily aligned. It is necessary to transform one image to align with the other one so that the difference between the anatomical information can be easily observed. Image registration is also widely used in the field of remote sensing. By aligning two images of the earth information at the same location but taken at different times, it is easy to detect and further analyze the change.

In image registration, we assume one image is fixed and transform another one to align with the fixed one. The image whose location is fixed is called the target image. The image which is transformed to aligned with the target one is called the template image.

2.2 Classification

Image registration can be classified into different categories based on different aspects. For different types of registration, different registration methods are applied. A fully description of the classification is included in [15]. Here, we only focus on four aspects: registration bases, image dimensions, transformation types, and modalities.

2.2.1 Registration Bases

Image registration can be conducted by extracting different information from images. Based on the kind of information, image registration can be classified into landmark-based and intensity-based methods.

In landmark-based image registration, features of the objects (points, curvatures, gradients, etc.) in images are extracted for registration. The choice of landmarks highly depends on the shape of the objects in images. Thus, locating features precisely is difficult. Also, preprocessing of images, such as image segmentation, is often needed before the registration, which may affect the robustness of registration. In comparison, intensity-based image registration only use intensity values to perform registration. Although it generally requires more computation than landmark-based image registration, intensity-based image registration is considered more robust because no preprocessing of images is required. In this thesis, we exclusively focus on intensity-based image registration.

2.2.2 Image Dimensions

The dimensions of the target and the template images are usually the same. Common types in this category include 2D-2D and 3D-3D image registration, which indicate that the target and the template images are both in two dimensions or three dimensions, respectively.

The dimensions of the target and the template images can also be different. For example, in 2D-3D image registration, the target image is in two dimensions and the template image is in three dimensions. Registering those two images requires transforming the three dimensional template image, including mapping a 3D data volume onto a 2D image, to align with the 2D target image. It will be discussed in Chapter 3 in more details.

2.2.3 Transformation Types

Image registration can be classified into rigid and non-rigid image registration. In rigid image registration, we assume template images are rigid bodies, and only rotation and translation are included in the transformation parameter sets. For example, we consider three degrees of freedom when template images are in two dimensions (rotation through one axis and translation in two dimensions), and six degrees of freedom when template images are in three dimensions (rotation through three axes and translation in three dimensions). In non-rigid image registration, in addition to rigid transformations, deformable (e.g., affine, projective, curved, etc.) transformations are also considered, which requires much more degrees of freedom than rigid image registration.

Transformation types essentially depend on the characters of objects in the images. If the attributes of objects are rigid (e.g., bones) or almost rigid (e.g., kidneys), rigid image registration is preferred. Conversely, if the attributes of objects indicate that corresponding objects can be deformed (e.g., livers), it is more suitable to perform non-rigid image registration. Although non-rigid image registration is more general, the computational complexity is high due to its high degrees of freedom. To reduce the computational complexity, rigid image registration is usually performed first to approximately align the images. Afterwards, non-rigid registration is implemented to get more accurate solutions. In this thesis, we focus on how to perform rigid image registration efficiently and accurately.

2.2.4 Modalities

Image registration can also be classified into mono-modality and multi-modality depending on image modalities. If the target and the template images are produced by the same sensor with the same physical parameters, this kind of registration is called mono-modality image registration. Multi-modality image registration refers to the case that the target and the template images are produced by different sensors or the same sensor with different physical parameters. Figure 2.1 shows an example of medical images in different modalities (T1-, T2-, PD-weighted MRIs and CT). Those images are from Retrospective Image Registration Evaluation Project¹ and

¹ The images and the standard transformations were provided as part of the project, “Retrospective Image Registration Evaluation”, National Institutes of Health, Project Number 8R01EB002124-03, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.

provided by courtesy of Professor Jeffery Orchard from the School of Computer Science at the University of Waterloo. They are aligned using the gold-standard transformations and further re-sampled.

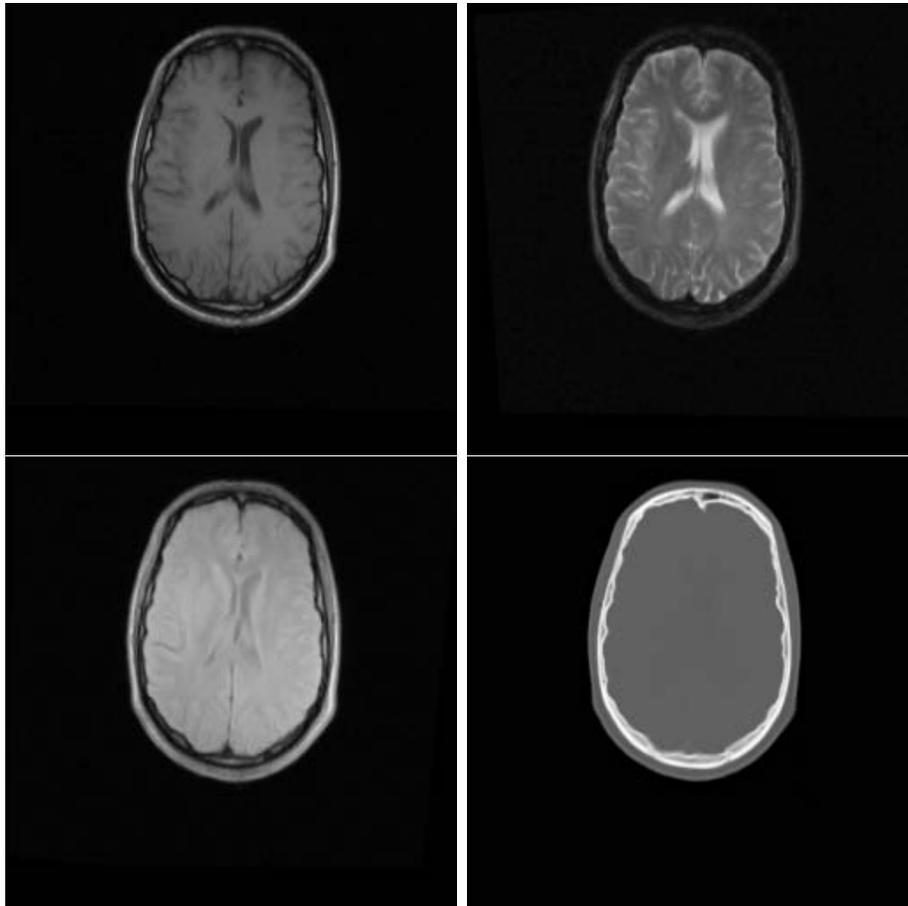


Figure 2.1: T1- (upper left), T2- (upper right), PD- (lower left) weighted MRIs and CT (lower right).

In mono-modality image registration, the target and the template images have the same or similar intensity values when they are registered. However, in multi-modality image registration, different sensors or different physical parameters result in different intensity values between the target and the template images. Unlike mono-modality image registration, direct relationship between the image intensity values may not be easily found for registration.

2.3 Mathematical Formulation

Image registration can be formulated as a mathematical problem. In this section, we start with the mathematical description of image functions. We then indicate that solving an image registration problem is equivalent to solving an optimization problem. Also, different similarity measures for optimization are discussed.

2.3.1 Image Functions

An image refers to a finite set of digital value vectors located at a set of rectangle grids (pixels) in a Cartesian coordinate system. We can formulate the relationship as a map

$$I_d : \mathcal{A} \mapsto \mathbb{N}^n,$$

where

$$\mathcal{A} = \{\mathbf{x}_i\} \subsetneq \mathbb{R}^m.$$

\mathbf{x}_i denotes the discrete grid positions in \mathbb{R}^m , where i represents the indices of the pixel locations, and m represents the dimensions of images (e.g., $m = 2$ denotes a 2D image; $m = 3$ denotes a 3D image). Each mapping value vector $I_d(\mathbf{x}_i)$ represents the vector of the color intensity values in \mathbb{N}^n , where n denotes the number of the color channels. For example, an image in the RGB color space consists of three color intensity values at each pixel location, in which case $n = 3$. In a grey image, on the other hand, there is only one pixel intensity at each pixel location; i.e., $n = 1$. We mainly consider grey images in this thesis. Intensity values normally range in a subset of \mathbb{N} . For example, intensity values range in the integers in the range of $[0, 255]$ for grey images encoded in 8 bits.

Although only intensity values at pixel locations can be observed, mathematically, we often consider images as functions defined not just at the pixel locations but also on the whole image area with the observable intensity values as the sample values of that function. More precisely, an image can be defined as a function

$$I : \mathbb{R}^m \mapsto \mathbb{R}, \tag{2.1}$$

satisfying $I(\mathbf{x}_i) = I_d(\mathbf{x}_i)$. For simplicity, we denote the pixel value at \mathbf{x}_i by I_i . Note that in practice, only $I(\mathbf{x}_i) = I_i$ are given. The image function $I(\mathbf{x})$ is usually constructed by interpolation. The pre-image of I is denoted by I^{-1} , representing the domain of the image.

In this thesis, $I^g(\mathbf{x})$ and $I^f(\mathbf{x})$ denote the target image and the template image, respectively.

2.3.2 Optimization

Let ϕ be the transformation with a parameter set \mathbf{s} . The transformed template image can be written as

$$\phi(I^f; \mathbf{s})(\mathbf{x}).$$

The image registration problem is to find an optimal parameter set \mathbf{s} such that the difference between the target and the transformed template images is minimized. Mathematically, the registration problem can be formulated as an unconstrained minimization problem

$$\mathbf{s} = \operatorname{argmin} \Gamma(I^g(\mathbf{x}), \phi(I^f; \mathbf{s})(\mathbf{x})),$$

where Γ is a measure of the difference between the two images. An alternative is to consider the registration problem as the one to maximize the similarity between the target and the transformed template images. The formulation thus becomes an unconstrained maximization problem

$$\mathbf{s} = \operatorname{argmax} \Psi(I^g(\mathbf{x}), \phi(I^f; \mathbf{s})(\mathbf{x})),$$

where Ψ is a measure of the similarity between the two images.

2.3.3 Similarity Measures

Let the set \mathcal{I} represent the indices where the corresponding pixel locations are in the intersection of domains of the target and the transformed template images, denoted by

$$\mathcal{I} \stackrel{\text{def}}{=} \left\{ \mathbf{i} \mid \mathbf{x}_{\mathbf{i}} \in I^{g^{-1}} \cap \phi(I^f; \mathbf{s})^{-1} \right\}.$$

In mono-modality image registration, the images for registration have the same or close intensity maps. A commonly used similarity measure is the sum of squared differences (SSD):

$$\Gamma_{\text{SSD}} = \sum_{\mathbf{i} \in \mathcal{I}} (\phi(I^f; \mathbf{s})_{\mathbf{i}} - I_{\mathbf{i}}^g)^2. \quad (2.2)$$

It is the l^2 norm of the pixel value differences between the target and the transformed template images. SSD has the advantage that it is also the optimum measure even if those two images differ by a Gaussian noise [29].

Alternatively, the similarity measure can be chosen as the sum of absolute differences (SAD):

$$\Gamma_{\text{SAD}} = \sum_{\mathbf{i} \in \mathcal{I}} |\phi(I^f; \mathbf{s})_{\mathbf{i}} - I_{\mathbf{i}}^g|.$$

It is the l^1 norm of the pixel value differences between the target and the transformed template images.

If intensity values of the two images are not identical but linearly related when they are aligned (e.g., two images are generated from the same sensor with the same physical parameters but different numbers of bits are encoded), another kind of similarity measure called correlation coefficient (CC) can be applied. It is defined as

$$\Psi_{\text{CC}} = \frac{\sum_{\mathbf{i} \in \mathcal{I}} (I_{\mathbf{i}}^g - \bar{I}_{\mathbf{i}}^g) (\phi(I^f; \mathbf{s})_{\mathbf{i}} - \bar{\phi}(I^f; \mathbf{s})_{\mathbf{i}})}{\sqrt{\sum_{\mathbf{i} \in \mathcal{I}} (I_{\mathbf{i}}^g - \bar{I}_{\mathbf{i}}^g)^2 \sum_{\mathbf{i} \in \mathcal{I}} (\phi(I^f; \mathbf{s})_{\mathbf{i}} - \bar{\phi}(I^f; \mathbf{s})_{\mathbf{i}})^2}},$$

where $\bar{I}_{\mathbf{i}}^g$ and $\bar{\phi}(I^f; \mathbf{s})_{\mathbf{i}}$ denote the mean of the intensity values in the target image and the mean of the intensity values in the transformed template image, respectively. Different from SSD and SAD, CC registers two images by maximizing the objective function, assuming intensity values of the two images are positively related. On the contrary, in the case that intensity values of the two images are negatively related, we can register the two images by minimizing the objective function.

For multi-modality image registration, because there is no explicit relationship between intensity values in the target and the transformed template images, similarity measures, such as SAD, SSD, and CC, cannot be applied in this scenario. Other similarity measures are used instead, and mutual information [8, 28] is most commonly used among them. In this similarity measure, intensity values in the images for registration are treated as random variables. Mutual information is then computed using their probabilities. It will be made more precise in Chapter 4.

Similarity measures normally depend on the image intensity values in the overlap region of the target and the transformed template images. However, determining the set \mathcal{I} during the registration process is quite complicated. To simplify the registration process, in this thesis, we assume the template image is periodic at the region of the target image. We further assume the image intensity values applied to similarity measures are always at the pixel locations of the target image.

Chapter 3

GPU-based Acceleration

2D-3D image registration is playing important roles in many medical image applications, such as radiation therapies [30] and computer-assisted surgeries [24, 25]. However, this kind of registration is computationally intensive, especially due to 3D volume rendering involved in the whole registration procedure. We begin the chapter with an introduction to 2D-3D mono-modality rigid image registration, and then describe the Gauss-Newton method, which is used to solve the 2D-3D image registration. Noticing that this problem can be accelerated using parallel computing on multi-core hardware, we also address how we implement the Gauss-Newton method on graphics processing units (GPUs) using RapidMind Multi-core Development Platform (RapidMind) to speed up the procedure. Numerical experiments are included at the end of this chapter to show the improvement of the efficiency.

3.1 2D-3D Image Registration

In 2D-3D image registration, the target image is in 2D and the template image is in 3D. The objective of 2D-3D image registration is to transform the 3D template image to align with the 2D target image. This kind of registration is widely used in a lot of medical applications. In computer assisted surgeries, to locate the instruments in patients' bodies, physicians often use 2D X-ray images collected from the sensors attached to the instruments to adjust the positions of the instruments. Technically, the 2D images collected by the sensors are called portal images. Although portal images help physicians determine the positions of the instruments to some degree, it is really difficult to precisely locate the positions of the instruments in patients' 3D bodies using 2D images. In order to solve the problem, a 3D CT data volume

for a patient’s body is often collected before a surgery. During the surgery, it is transformed to align with the portal images in order to get a precise position of the instrument. Similarly, in radiation therapies, due to the harm of radiation, cancers are supposed to receive more radiation than healthy tissues. 2D-3D image registration is used to precisely adjust the focus of the radiation by aligning a pre-acquired 3D CT data volume of the tissues with 2D portal images collected by a sensor attached to the beam collimator.

Different from other image registration where the target and the template images are in the same dimensions, a 2D-3D image registration process involves volume rendering from a 3D image to a 2D image. After rotated and translated, the 3D volume is rendered into a 2D image which is expected to be close to the portal image. Technically, the generated 2D image is called a digitally reconstructed radiograph (DRR). Figure 3.1 shows the schematic of the 2D-3D image registration procedure. With an initial guess of the transformation, the DRR is computed. By comparing it with the portal image based on some similarity measure, new transformation parameters are generated to transform the 3D image in the next iteration. The process is iterated till the solutions converge.

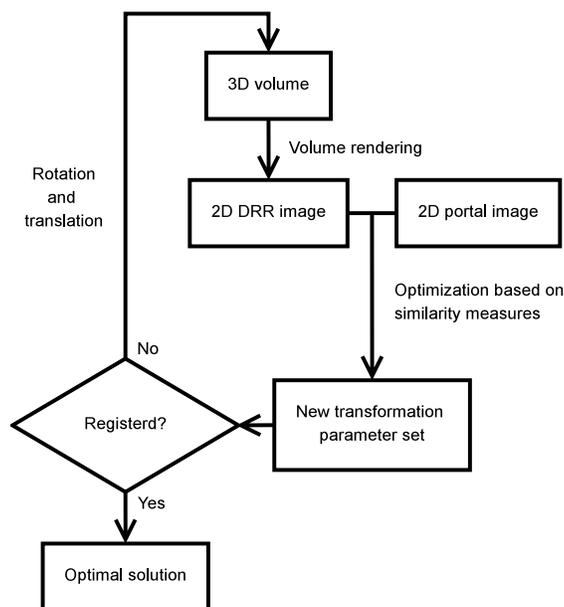


Figure 3.1: The iteration scheme of 2D-3D medical image registration.

Volume rendering is an essential step to compute DRRs. The real mechanism for volume rendering is complex since the effect of attenuation need to be considered. For simplicity, we assume the volume rendering refers to a perspective projection from a 3D volume onto a 2D plane, as is shown in Figure 3.2.

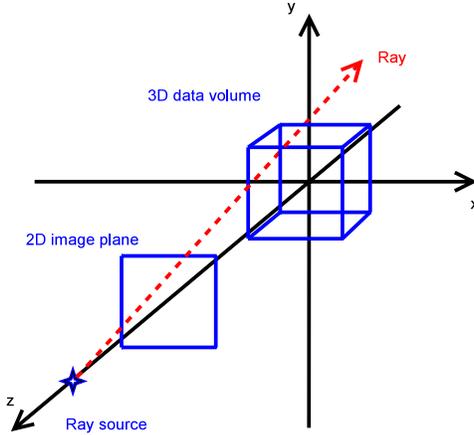


Figure 3.2: The framework of DRR construction.

Algorithms for volume rendering include ray casting [20] and the shear-warp factorization [12]. Although efficient, the shear-warp factorization makes artifacts in the resulting DRRs under certain circumstances and thus is not suitable for accurate image registration [12]. In comparison, ray casting is more accurate and simple. Thus, it is chosen as the volume rendering algorithm in our experiments.

Figure 3.3 shows the mechanism of a ray casting algorithm. Each intensity value in the DRR is computed as follows: a ray emitted from the ray source passes through a specific pixel location in the DRR, and eventually passes through the 3D volume. In principle, the pixel value on the DRR should be calculated as the accumulated intensity values of the 3D volume along the ray. However, it is quite expensive to determine the line integral along the ray. Instead, the ray casting algorithm selects m equidistant samples which are the intercepts of the ray with m concentric spheres with different radii covering the 3D volume. The intensity value of the corresponding pixel on DRR is computed as the sum of the intensity values of those m samples, which are determined by interpolation based on voxels in the 3D volume. In principle, the centers of those concentric spheres are assumed to be located at the ray source. For intensity values on other pixels of the DRR, we use the same strategy and use the same spheres to determine the samples. Normally, those spheres need to cover the possible positions of the 3D volume to achieve accurate solutions.

The complexity of the ray casting algorithm is $O(N^2m)$, assuming the resolution of the DRR is $N \times N$. Although simple, the volume rendering process can be computationally intensive, especially when DRRs are in high resolution.

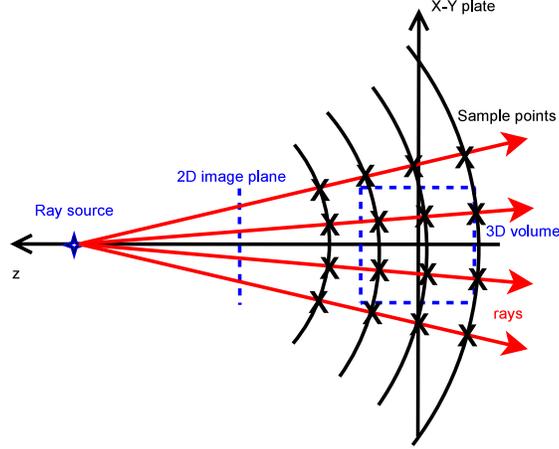


Figure 3.3: The schematic of the ray casting algorithm.

3.2 Gauss-Newton Method

The Gauss-Newton method is one of the most popular optimization methods for mono-modality rigid image registration. Sum of squared differences (2.2) is often used as the similarity measure in this method.

Let I^g and I^f be the portal image and the 3D data volume, respectively. Also, we denote the DRR by $\phi(I^f; \mathbf{s})$, where \mathbf{s} is the transformation parameter set. Precisely, $\mathbf{s} = (\theta_x, \theta_y, \theta_z, \tau_x, \tau_y, \tau_z)^T$, where θ_x , θ_y , and θ_z are the rotation angles in the three axes, and τ_x , τ_y , and τ_z are the translation in the x , y , and z directions, respectively.

Equation (2.2) is a nonlinear least squares problem. Let symbol \mathbf{s}^n denote the transformation parameter set at the iteration n , and $\Delta \mathbf{s}^{n+1} = \mathbf{s}^{n+1} - \mathbf{s}^n$ denote the change of solutions between iteration n and $n + 1$. For simplicity, we omit I^f in $\phi(I^f; \mathbf{s})$. By Taylor series expansion, at pixel (i, j) , $\phi(\mathbf{s}^{n+1})_{i,j}$ can be written as

$$\phi(\mathbf{s}^{n+1})_{i,j} = \phi(\mathbf{s}^n)_{i,j} + \nabla \phi(\mathbf{s}^n)_{i,j}^T \cdot \Delta \mathbf{s}^{n+1} + \dots, \quad (3.1)$$

where $\nabla \phi(\mathbf{s}^n)_{i,j} = \left(\frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \theta_x}, \frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \theta_y}, \frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \theta_z}, \frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \tau_x}, \frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \tau_y}, \frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \tau_z} \right)^T$ is the gradient of $\phi(\mathbf{s}^n)_{i,j}$. By substituting (3.1) into (2.2), the optimization problem can be approximated by

$$\min_{\Delta \mathbf{s}^{n+1}} \sum_{i,j} \left(\phi(\mathbf{s}^n)_{i,j} - I_{i,j}^g + \nabla \phi(\mathbf{s}^n)_{i,j}^T \cdot \Delta \mathbf{s}^{n+1} \right)^2.$$

This is a linear least squares problem which can be rewritten in the matrix form as

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,$$

where $A = \nabla\phi(\mathbf{s}^n) = N^2 \times 6$ matrix, $\mathbf{x} = \Delta\mathbf{s}^{n+1}$, and $\mathbf{b} = I^g - \phi(\mathbf{s}^n) = N^2 \times 1$ vector, assuming the resolution of the DRR is $N \times N$. The linear least squares problem can then be solved by the normal equation

$$A^T A \mathbf{x} = A^T \mathbf{b} \quad (3.2)$$

using Gaussian elimination on a 6×6 linear system.

Once $\mathbf{x} = \Delta\mathbf{s}^{n+1}$ is determined, it can be used to update the orientation of the 3D volume. The procedure is repeated until $\{\mathbf{s}^n\}$ have converged to an accurate solution.

The computational complexity is linear to the number of pixels in the DRRs and portal images. In the case that the DRRs or portal images are in high resolution, the complexity can be large.

3.3 Parallel Computing

The 2D-3D image registration is often used in clinical treatments, and the efficiency of the registration process is important. However, the large complexity in the algorithms, especially the DRR construction, results in considerable delay. Parallel computing is considered as a way to reduce the complexity and accelerate the registration process.

In each iteration, the first step of the registration process is constructing the DRR. Note rotating and translating the 3D data volume is equivalent to rotating and translating the sample points by opposite values. Taking the solution of $\Delta\mathbf{s}^n$ in the previous iteration, the sample points are rotated and translated. Because we only consider rigid transformations, all positions of the sample points are applied to the same linear transformation, and their transformed positions can be computed in parallel. Given the intensity value and the position of each voxel in the 3D data volume, the interpolation of the intensity values at the sample points can be done in parallel. Considering each sample point in each sphere is from a pixel point in the DRR, the sample points in each sphere can be constructed as a 2D array. Since each sphere has the same arrangement of points, the entire sample points can be formed as a 3D array. Finally, the interpolated intensity values at the sample points corresponding to a ray is summed up as an intensity value in the DRRs. This is equivalent to project the 3D array, whose elements represent the pixel values at the sample points, into a 2D array. That can also be parallelized on each simulated ray for constructing DRRs.

After the DRR is computed, the second step is using the Gauss-Newton method to acquire a new solution. This procedure can also be parallelized. In order to compute the matrix A in (3.2), the gradient vector at each pixel (i, j) is approximated by finite difference. For example,

$$\frac{\partial \phi(\mathbf{s}^n)_{i,j}}{\partial \theta_x} \approx \frac{\phi(\mathbf{s}^n + \Delta \mathbf{s}_{\theta_x})_{i,j} - \phi(\mathbf{s}^n)_{i,j}}{\Delta \theta_x}, \quad (3.3)$$

where $\Delta \mathbf{s}_{\theta_x} = (\Delta \theta_x, 0, 0, 0, 0, 0)^T$. $\phi(\theta_x^n + \Delta \theta_x)$ is obtained from the DRR by changing the rotation angle θ_x^n to $\theta_x^n + \Delta \theta_x$ but keeping the other parameters fixed. The parallel DRR construction has been explained above. Then, the subtraction and division in (3.3) are easily computed in parallel. Other derivatives can also be computed in parallel in similar ways. Finally, we have six 2D arrays of $\frac{\partial \phi}{\partial \theta_x}, \dots, \frac{\partial \phi}{\partial \tau_z}$ which correspond to the six columns of A . To compute each entry in $A^T A$ in the normal equation, we take two derivative arrays, multiply the corresponding elements in parallel, and then use a reduction operation to compute the global sum. Regarding the right-hand side, \mathbf{b} represents the difference between I^g and $\phi(\mathbf{s}^n)$, which can be easily computed in parallel. $A^T \mathbf{b}$ can be computed in parallel in a similar manner as $A^T A$. However, to solve the final 6×6 linear system in parallel is not that obvious. Fortunately, the scale of the system is so minimal that the computational time for this part can almost be neglected in the whole registration process. This is the only part that parallel computing is not applied in the registration procedure.

3.3.1 Graphics Processing Units

Multi-processing is a commonly used technique for parallel computing. In a multi-processing system, more than one central processing unit (CPU) in one computer system is exploited to complete different parts of one task at the same time. In this kind of system, a task is first divided into several parts. Each part is assigned to each processor and then implemented sequentially. Data computed by each sub-task need to communicate with others if necessary. Finally, the data computed by different processors are gathered together.

Multi-processing is powerful. However, it normally takes large space and thus is not suitable for clinical environment. Also, maintaining such a giant system is not easy.

Different from multiprocessing systems, multi-core architectures refer to the case that more than one independent core is built in the same integrated circuit.

Compared to multi-processing systems, multi-core processors have a much smaller form factor. Multi-core CPUs, cell processors, and GPUs are all in this category. Considering modern GPUs normally have more than 100 cores which are more than the other two, GPUs are used as the underlying hardware to implement the parallel 2D-3D image registration in this thesis.

GPUs are a kind of multi-core devices for graphics rendering for personal computers (PCs), workstations or game consoles. Due to their highly parallel architectures, GPUs are widely used for parallel stream computing nowadays. Figure 3.4 shows the GPU of NVIDIA GeForce 8800 GTX, which is exploited in the experiments.



Figure 3.4: NVIDIA GeForce 8800 GTX.

3.3.2 RapidMind Multi-Core Development Platform

To make full use of the cores in a GPU, several GPU programming languages, such as OpenGL [31], Brook for GPUs [1], and Cg [16], have been designed in the past few years. However, those languages normally require programmers to have substantial knowledge of the underlying architectures such as vertex shader, rasterizer, texture maps, etc.

The RapidMind Multi-core Development Platform¹ makes GPU programming more convenient. As shown in Figure 3.5, the platform acts as a median layer between the applications and the multi-core hardware (e.g., GPUs). It essentially abstracts the hardware complexity from the users and makes GPU programming

¹<http://www.rapidmind.com>

much easier than before. In RapidMind, the parallel processing arithmetic is performed as array operations. Parallel processing is achieved by performing operations on all the elements in all the arrays concurrently. Also, it can automatically optimize the code to further improve the efficiency of the performance.

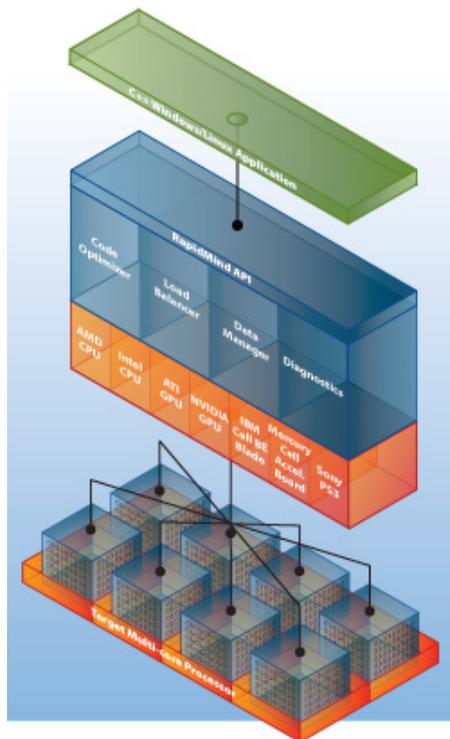


Figure 3.5: The architecture of RapidMind. Copyright 2008 RapidMind Inc. All rights reserved.

We use RapidMind to implement the registration process in parallel on a GPU for improving the efficiency. All the procedures, except for solving a 6×6 linear system, are executed in a GPU in parallel in each iteration. After the linear system is computed, we transfer the matrix $A^T A$ and right-hand side $A^T \mathbf{b}$ to a CPU and solve the linear equations using the Gauss elimination method in order to obtain a new update for the transformation parameter set. Afterwards, we transfer the new solution from the CPU back to the GPU for the next iteration.

3.4 Numerical Results

Both synthetic and clinical data are used to demonstrate the efficiency of the registration process in parallel using RapidMind on GPUs. For both of the data, two

kinds of experiments, including the regular C++ language on CPUs and the RapidMind code on GPUs, are executed on the same registration problem. The C++ code is used to test the CPU efficiency, and RapidMind code is used to test the GPU efficiency. Their times are recorded for comparison. All the experiments are performed on a standard PC running with Ubuntu 7.04 Linux operating system, Intel processor with 3GHz, 1GB memory, and a graphics card of NVIDIA GeForce 8800 GTX.

The common parameters for all the experiments are fixed for consistency. As shown in Figure 3.2, the center of the 3D volume is placed in the center of the 3D coordinates. The initial guess of the transformation is set as $\mathbf{s} = (0, 0, 0, 0, 0, 0)^T$. The center of the 2D projection plane is located on the z axis, and the plane is perpendicular to z axis. The ray source is also located on the z axis. In the experiments, the positions of the ray source and the center of the projection plane are chosen as $2.5 \times L_z$ and $1.5 \times L_z$, respectively, where L_z is the length of the 3D volume in the z direction. Regarding DRR construction, we adopt ray casting algorithm explained in Section 3.1. The radii of those concentric spheres arranging from L_z to $4 \times L_z$ seem sufficient to cover the possible positions of the 3D volume.

The portal images are synthesized by constructing DRRs with known transformation parameter sets for the 3D volume. Also, we assume the resolution of DRRs is the same as that of corresponding target 2D images.

3.4.1 Synthetic Data

Synthetic data are used to test the acceleration regarding the same registration problem but with different resolution images. An advantage of using synthetic data is that we can generate image volumes with different resolution without affecting the image quality. In the experiments, the 3D volume is chosen as a white cube (e.g., all intensity values in the 3D volume are 255 for 8-bit grey images). For consistency, the size of the cube is scaled to $[0, 1] \times [0, 1] \times [0, 1]$ for all the resolution. We construct the initial DRRs without any rotation or translation. The portal images are simulated using the transformation parameter set \mathbf{s} as $(5^\circ, 5^\circ, 5^\circ, 0.1, 0.1, 0.1)^T$ in all of the experiments. We also keep the resolution of the white cube, the resolution of the DRR, and the resolution of the portal image consistent with one another in each experiment. For instance, assuming the resolution of the white cube is set as $64 \times 64 \times 64$, the resolution of the DRR and the portal image are also set as 64×64 . Figure 3.6 (left) shows the initial guess of the DRR and Figure 3.6 (right) shows the simulated target image.



Figure 3.6: The initial guess of the DRR (left) and simulated target image using the transformation parameter set \mathbf{s} as $(5^\circ, 5^\circ, 5^\circ, 0.1, 0.1, 0.1)^T$ (right).

Table 3.1 shows the CPU times for registration using images with different resolution. The 3D volume resolution varies from $16 \times 16 \times 16$ to $128 \times 128 \times 128$. Each total CPU time is subdivided into two CPU times corresponding to the time for DRR computation (DRR), which is included in both DRR construction and linearized optimization problem, and the time for the rest non-DRR computation (Non-DRR). Note that the times are per iteration. The stopping criteria are $\|\Delta \mathbf{s}^n\|_2 < 0.1$. The total numbers of iterations are also shown.

As the 3D resolution increases, the CPU time for the regular C++ code increases with a factor of around 10. While the per iteration time is less than 1 second for the lower resolution dataset, it increases rapidly to over 50 seconds for the higher resolution dataset ($128 \times 128 \times 128$). The CPU time for the RapidMind code increases much slower; the per iteration time is less than 1 second for all datasets. For the higher resolution dataset ($128 \times 128 \times 128$), the total registration time for RapidMind is around 2 seconds, whereas for regular C++, it is almost 6 minutes.

By comparing the DRR times and non-DRR times in the column of regular C++, we note that the former is around 2-4 times longer than the latter. Because DRR computing dominates the CPU time in the entire registration process, it is important to use efficient parallel algorithms for DRR computation to improve the efficiency of the performance. Also, it is easy to implement the algorithms using RapidMind on GPUs.

We have also tried different parameters to construct DRRs and simulated the portal images. Similar timing results are achieved.

Resolution	Time (sec)	Regurlar C++	RapidMind
$16 \times 16 \times 16$	DRR	0.012	0.029
	Non-DRR	0.003	0.109
	Total	0.015	0.138
	Iteration	6	6
Resolution	Time (sec)	Regular C++	RapidMind
$32 \times 32 \times 32$	DRR	0.140	0.047
	Non-DRR	0.060	0.139
	Total	0.200	0.186
	Iteration	5	5
Resolution	Time (sec)	Regular C++	RapidMind
$64 \times 64 \times 64$	DRR	3.990	0.091
	Non-DRR	1.318	0.171
	Total	5.308	0.262
	Iteration	5	5
Resolution	Time (sec)	Regular C++	RapidMind
$128 \times 128 \times 128$	DRR	40.331	0.152
	Non-DRR	16.388	0.248
	Total	56.720	0.400
	Iteration	6	6

Table 3.1: Average CPU times for each iteration of the image registration process.

3.4.2 Clinical Data

We also demonstrate the capability of RapidMind on GPUs using clinical data. The experimental image volume is a real 3D CT volume from the database of the Department of Radiology at the University of Iowa, which shows a tripod fracture of a skull²; see Figure 3.7.

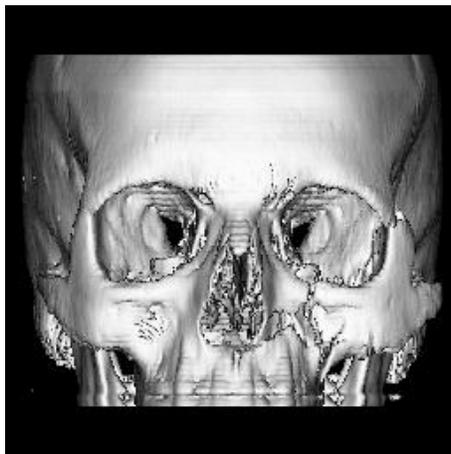


Figure 3.7: A tripod fracture of a skull

The resolution of the image is $256 \times 256 \times 203$ with each voxel dimension as $0.7\text{mm} \times 0.7\text{mm} \times 2\text{mm}$. This 3D volume is quite large and we only use the central chunk in the first 100 slices. The resulting resolution is $128 \times 128 \times 100$ for our numerical experiments. The portal images are also synthesized similar to the first experiment, but we use different parameters, which are shown in the first column of Table 3.2. The parameters are still $\mathbf{0}$ for constructing the initial DRRs. The CPU times for the entire image registration process are also shown in Table 3.2. The stopping criteria are chosen as $\|\Delta\mathbf{s}\|_2 < 0.8$. The resulting relative errors are less than 0.012 for all of our experiments by comparing the final numerical solutions with the corresponding ground truths.

Similar to the synthetic experiments, the RapidMind implementation accelerates the registration process by a factor of over 100 for clinical images.

²<http://www.radiology.uiowa.edu/downloads/>

Portal image parameters	Time (sec)	Regular C++	RapidMind
Rotation: (2°, 2°, 2°)	DRR	123.27	0.62
	Non-DRR	44.89	1.13
Translation: (2mm, 2mm, 2mm)	Total	168.16	1.75
	Iteration	5	5
Portal image parameters	Time (sec)	Regular C++	RapidMind
Rotation: (4°, 4°, 4°)	DRR	191.94	0.98
	Non-DRR	71.36	1.72
Translation: (4mm, 4mm, 4mm)	Total	263.30	2.70
	Iteration	8	8
Portal image parameters	Time (sec)	Regular C++	RapidMind
Rotation: (6°, 6°, 6°)	DRR	267.96	1.32
	Non-DRR	103.43	2.33
Translation: (6mm, 6mm, 6mm)	Total	371.39	3.65
	Iteration	11	11

Table 3.2: Total CPU times for the entire image registration process.

Chapter 4

Multi-modality Image Registration

In this chapter, mutual information, which is one of the similarity measures for multi-modality image registration, is discussed. The definition of the mutual information is first introduced, followed by different models for computing it. The optimal translations can normally be obtained by maximizing the mutual information functions. However, non-smoothness and interpolation artifacts, which are derived from the models, hamper standard optimization processes. We discuss those issues for different models. Optimization methods suitable for those models are also discussed.

4.1 Mutual Information

In multi-modality image registration, since there is no explicit relationship between intensity values in the target and the template images, similarity measures cannot use the intensity values directly. Instead, the statistical properties of intensity values are exploited and used in similarity measures.

In the perspective of statistics, the image intensity value in one image can be considered as a random variable. Entropy [9] is defined as a function of its probability distribution if the random variable is discrete, or its probability density function (PDF) if the random variable is continuous. It is a measure of the randomness of the random variable. This term originally arises from classical thermodynamics as a measure of randomness of molecules in a system. Recently, it has been widely used in statistics, chemical thermodynamics, statistical mechanics, quantum mechanics,

astrophysics, information theory, and many other areas. In the area of information theory, Shannon entropy [23] is commonly used. Also, it has also become a popular similarity measure in image registration in the last decade.

4.1.1 Discrete Cases

Given a discrete random variable X , let $\{p_i^X\}_{i=1}^{N^X}$ represent its probability distribution, assuming there are N^X possible observations. We denote the entropy of X by the symbol H^X , which is defined as

$$H^X \stackrel{\text{def}}{=} - \sum_{i=1}^{N^X} p_i^X \log p_i^X.$$

Note that in the case $p_i^X = 0$, we simply assign $p_i^X \log p_i^X = 0$.

The joint entropy is a term describing the relationship between two random variables. It is essentially a function of the joint probability distribution. Let Y be another discrete random variable with the probability distribution $\{p_j^Y\}_{j=1}^{N^Y}$, where N^Y denotes the number of possible observations. Let the joint probability distribution of X and Y be $p_{i,j}^{XY}$, where $i = 1, \dots, N^X$ and $j = 1, \dots, N^Y$. The joint entropy of these two random variables H^{XY} is defined as

$$H^{XY} \stackrel{\text{def}}{=} - \sum_{i=1}^{N^X} \sum_{j=1}^{N^Y} p_{i,j}^{XY} \log p_{i,j}^{XY}. \quad (4.1)$$

The joint entropy characterizes how related two random variables are, or how random the joint distribution is. The more related the two random variables are, the lower the joint entropy is, and vice versa. This can be verified from two examples. In the first example, the two random variables, X and Y , are independent; i.e.,

$$p_{i,j}^{XY} = p_i^X \cdot p_j^Y \quad i = 1, \dots, N^X; j = 1, \dots, N^Y.$$

The joint entropy is computed by

$$\begin{aligned}
H^{XY} &= - \sum_{i=1}^{N^X} \sum_{j=1}^{N^Y} p_{i,j}^{XY} \log p_{i,j}^{XY} \\
&= - \sum_{i=1}^{N^X} \sum_{j=1}^{N^Y} p_i^X p_j^Y \log (p_i^X p_j^Y) \\
&= - \sum_{i=1}^{N^X} \sum_{j=1}^{N^Y} p_i^X p_j^Y \log p_i^X - \sum_{i=1}^{N^X} \sum_{j=1}^{N^Y} p_i^X p_j^Y \log p_j^Y \\
&= - \sum_{i=1}^{N^X} p_i^X \log p_i^X - \sum_{j=1}^{N^Y} p_j^Y \log p_j^Y \\
&= H^X + H^Y,
\end{aligned}$$

where H^X and H^Y represent the individual entropies for X and Y , respectively.

In another example, suppose X and Y have the relationship $X = Y$. We have

$$p_{i,j}^{XY} = \begin{cases} p_i^X = p_j^Y & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, the joint entropy will be

$$H^{XY} = H^X = H^Y.$$

Comparing to the first example where the two random variables are independent of each other, after the relationship $X = Y$ is specified, the two random variables are more related than before. Accordingly, the joint entropy also decreases .

The relationship between the information of X and Y can be expressed using the diagram shown in Figure 4.1. The information of X is represented using the left circle, and its area denotes the entropy of X ; the information of Y is demonstrated using the right circle, and its area denotes the entropy of Y . The area of the union of the two circles represents the joint entropy of X and Y . When the two random variables are less related, the two circles will fall apart. The union of the two circles will be larger, and the joint entropy will be higher (e.g., $H^X + H^Y$ when X and Y are independent). In the contrary, when the two random variables are more related, the two circles will get closer. The union of the two circles will be smaller, and the joint entropy will be lower (e.g., H^X or H^Y when $X = Y$).

Mutual information is a measure of the reduction of the uncertainty of one variable given the knowledge of the other one. Let the symbol M represent the

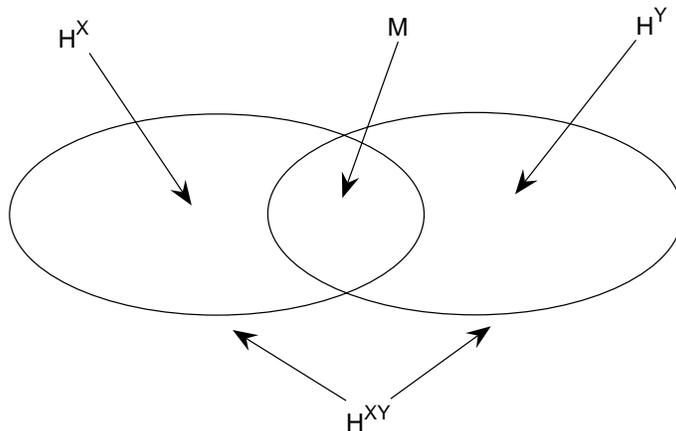


Figure 4.1: The relationship between the information of X and Y .

mutual information. It is defined as

$$M \stackrel{\text{def}}{=} H^X + H^Y - H^{XY}. \quad (4.2)$$

In Figure 4.1, mutual information is denoted by the intersection of the two circles representing individual entropies. When the two random variables are more related, the two circles are closer, the measure of the intersection is larger, and the mutual information is higher. When the two random variables are less related, the two circles spread apart. The measure of the intersection decreases, and so does the mutual information.

4.1.2 Continuous Cases

Shannon entropy can easily be extended for continuous random variables. As a counterpart, let \tilde{X} and \tilde{Y} be two continuous random variables associated with the PDFs of p^X and p^Y respectively, and their joint PDF be p^{XY} . The individual entropy of \tilde{X} is denoted by \tilde{H}^X , which is defined as

$$\tilde{H}^X \stackrel{\text{def}}{=} - \int_{-\infty}^{+\infty} p^X(x) \log p^X(x) dx. \quad (4.3)$$

Similarly, the individual entropy \tilde{H}^Y of \tilde{Y} is defined as

$$\tilde{H}^Y \stackrel{\text{def}}{=} - \int_{-\infty}^{+\infty} p^Y(y) \log p^Y(y) dy. \quad (4.4)$$

The joint entropy \tilde{H}^{XY} is defined as

$$\tilde{H}^{XY} \stackrel{\text{def}}{=} - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p^{XY}(x, y) \log p^{XY}(x, y) dx dy.$$

Accordingly, the mutual information \tilde{M} is defined as

$$\tilde{M} \stackrel{\text{def}}{=} \tilde{H}^X + \tilde{H}^Y - \tilde{H}^{XY}.$$

They have similar interpretations with those in the discrete case.

4.1.3 Mutual Information for Image Registration

By (2.1), the image intensity value can be assumed to be a continuous function with respect to the image coordinates. In this case, the image intensity value is treated as a continuous random variable, and the observable pixel intensity values are treated as observations of the random variable. Shannon entropy can be used here to represent how much information an image has.

For multi-modality image registration, the target image intensity value I^g and the transformed template image intensity value $\phi(I^f; \mathbf{s})$ are assumed to be different random variables associated with different distributions. The PDFs for the target image and the transformed template image are represented by p^g and p^f , respectively. Their entropies are represented by \tilde{H}^g and \tilde{H}^f .

In multi-modality image registration, let \tilde{H}^{fg} be the joint entropy of the two images, which can be considered as the similarity measure. The target image intensity value I^g and transformed template image intensity value $\phi(I^f; \mathbf{s})$ are considered as two random variables. Since the transformed template image is a function with respect to the transformation \mathbf{s} , \tilde{H}^{fg} is also a function of \mathbf{s} . When the two images are aligned, since the two random variables are most related, \tilde{H}^{fg} obtains its minimum. Otherwise, while the two images fall apart, \tilde{H}^{fg} will increase. Thus, the optimal transformation parameter set for registering two images can be obtained by minimizing the joint entropy.

Mutual information \tilde{M} has a similar interpretation to the joint entropy. We can find the optimal transformation parameter set to align two images by maximizing the mutual information. However, in literature, it is preferred using mutual information. The reason is because there may be an incorrect global minimum for the joint entropy in certain cases. Considering the case that two images have only background regions overlapped, if the region of interest is defined as the intersection of the two images, the joint entropy will attain its global minimum, which is not the desired solution [10, 19]. In our case, the region of interest is defined as the region of the target image. The template image is assumed to be periodic in the transformation process. The joint entropy is less possible to have the same

problem. Though, we still choose mutual information as the similarity measure in this thesis.

Although image intensity values can be treated as continuous random variables, the observable images are always discrete arrays, and only finite observations of the underlying continuous image functions can be obtained. Thus, it is difficult to attain the individual and joint PDFs of image intensity values. As a result, finite observations of intensity values lead people to use entropies and mutual information based on discrete random variables for image registration. As a counterpart, in the mutual information definition based on discrete random variables, we denote H^g and H^f by the individual entropies for the target image and the transformed template image respectively, H^{fg} by the joint entropy, and M by the mutual information.

4.2 Probability Distributions

Note that mutual information depends on the individual entropies and the joint entropy. Individual entropies depend on individual probability distributions, and the joint entropy depends on the joint probability distribution. Thus, mutual information is determined by how probability distributions are constructed.

Formally, let $[0, I_{\max}]$ be the range of the intensity values of an image (e.g., $I_{\max} = 255$ for an image encoded in 8 bits). The possible range is further divided into subintervals, or bins,

$$B_i \stackrel{\text{def}}{=} [J_{i-1}, J_i) \quad i = 1, \dots, N_B - 1,$$

$$B_{N_B} \stackrel{\text{def}}{=} [J_{N_B-1}, J_{N_B}],$$

where $J_i = ih_B$, $h_B = \frac{I_{\max}}{N_B}$, and N_B denotes the number of bins. The individual probability distribution is constructed by assigning appropriate probabilities p_i to the corresponding bins B_i where $i = 1, \dots, N_B$. The basic idea is that p_i represents the proportion of the pixels located in each bin B_i .

The joint probability distribution can be constructed in a similar way. Let $\{B_i^g\}_{i=1}^{N_B^g}$ be the bin partition for the target image and $\{B_j^f\}_{j=1}^{N_B^f}$ be the bin partition for the transformed template image, where N_B^g and N_B^f denote the number of bins for the target image and the transformed template image, respectively. A natural bin partition approach for the joint probability distribution can be defined as

$$B_{i,j}^{fg} \stackrel{\text{def}}{=} \left\{ (\alpha, \beta) \mid \alpha \in B_i^g, \beta \in B_j^f \right\} \quad i = 1, \dots, N_B^g; j = 1, \dots, N_B^f.$$

The corresponding joint probability distribution is denoted by $p_{i,j}^{fg}$. It is used to quantify the measure of the pixels located in each corresponding bin.

The individual and joint probability distributions can be computed differently using different models and further exploited to compute the mutual information. We will mainly discuss different models in this section.

4.2.1 Interpolation-based Models

One kind of models is called interpolation-based models. In this class of models, the calculation of mutual information requires the evaluation of intensity values located at the same fixed positions for both the target and the transformed template images. For simplicity, evaluation positions are chosen as the pixel locations in the target image. Thus, for the target image, observable intensity values are directly used for calculating the individual entropy H^g and the joint entropy H^{fg} . For the transformed template image, the pixel locations may not generally be aligned with the pixel locations of the target image. However, pixel pairs of the target image and the transformed template image are necessary in order to compute the joint probability distribution. Intensity values need to be assigned at the pixel locations of the target image. The interpolation-based models compute the intensity values using interpolation strategies.

Having the pixel pairs, the simplest way to calculate probability distributions is histogramming. The individual and joint probabilities associated with each bin are evaluated by counting the frequency of the intensities whose values are located at the corresponding bin. We further normalize the frequencies so that the sums of the normalized frequencies are equal to ones, which is consistent with the definition of probability distributions. Mathematically, suppose the number of pixels in the images is N and let $\mathbf{1}_{\mathcal{S}}(\cdot)$ denote the characteristic function of set \mathcal{S} which is defined as

$$\mathbf{1}_{\mathcal{S}}(\alpha) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \alpha \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$$

The individual probability distribution for the target image is then computed by

$$p_i^g = \frac{1}{N} \sum_{k=1}^N \mathbf{1}_{B_i^g}(I_k^g) \quad i = 1, \dots, N_B^g,$$

where I_k^g represents each pixel intensity value in the target image. The individual

probability distribution for the transformed template image is computed by

$$p_j^f(\mathbf{s}) = \frac{1}{N} \sum_{k=1}^N \mathbf{1}_{B_j^f}(\phi(I^f; \mathbf{s})_k) \quad j = 1, \dots, N_B^f, \quad (4.5)$$

where $\phi(I^f; \mathbf{s})_k$ denotes each interpolated pixel intensity value in the transformed template image. Finally, the joint probability distribution is defined as

$$p_{i,j}^{fg}(\mathbf{s}) = \frac{1}{N} \sum_{k=1}^N \mathbf{1}_{B_{i,j}^{fg}}(I_k^g, \phi(I^f; \mathbf{s})_k) \quad i = 1, \dots, N_B^g; j = 1, \dots, N_B^f. \quad (4.6)$$

Figure 4.2 shows the individual entropy for the target image, the individual entropy for the transformed template image, the joint entropy, and the mutual information with respect to the horizontal translation using the histogramming model. The target image is chosen as the T1-weighted MRI; the template image is chosen as the CT. Both of the images are shown in Figure 2.1. The number of bins are set by $N_B^g = N_B^f = 128$. The interpolation strategy is chosen as bilinear interpolation in the processes of image transformations. Note that the two images are roughly aligned. It can be observed that the mutual information obtains its maximum when the translation is near 0. However, we may also observe that the mutual information is not smooth. Notice that the characteristic function $\mathbf{1}_{\mathcal{S}}(\cdot)$ is a discontinuous function. By (4.5) and (4.6), $p_i^f(\mathbf{s})$ and $p_{i,j}^{fg}(\mathbf{s})$ are in turn discontinuous functions.

It can also be illustrated using an image plot shown in Figure 4.3. Suppose we only consider the case for 1D images. The transformation parameter set \mathbf{s} only contains the translation τ in one direction. Two adjacent pixels of the transformed template image are shown as the circle symbols when $\tau = \tau_0$. The interpolated intensity value at x_k is $\phi(I^f; \tau_0)_k$. In this case, it falls into the bin B_i^f . In the case that we translate the transformed template image a little further to the right, the two new pixel locations are in turn shifted to the right, as shown by the square symbols. Then the interpolated value at x_k becomes $\phi(I^f; \tau_0 + \Delta\tau)_k$, which now falls into the bin B_{i-1}^f . In this process, the frequency of pixels in the bin B_{i-1}^f increases by 1 and the frequency of the pixels in the bin B_i^f decreases by 1, leading to a discontinuous change in the probabilities. As a result, the entropies and mutual information are also not smooth.

In order to smooth the mutual information function, an intuitive approach is substituting the characteristic function for other functions that are smoother. One example is called Parzen windowing. Let

$$J_{i-\frac{1}{2}}^g \stackrel{\text{def}}{=} \frac{1}{2} (J_{i-1}^g + J_i^g)$$

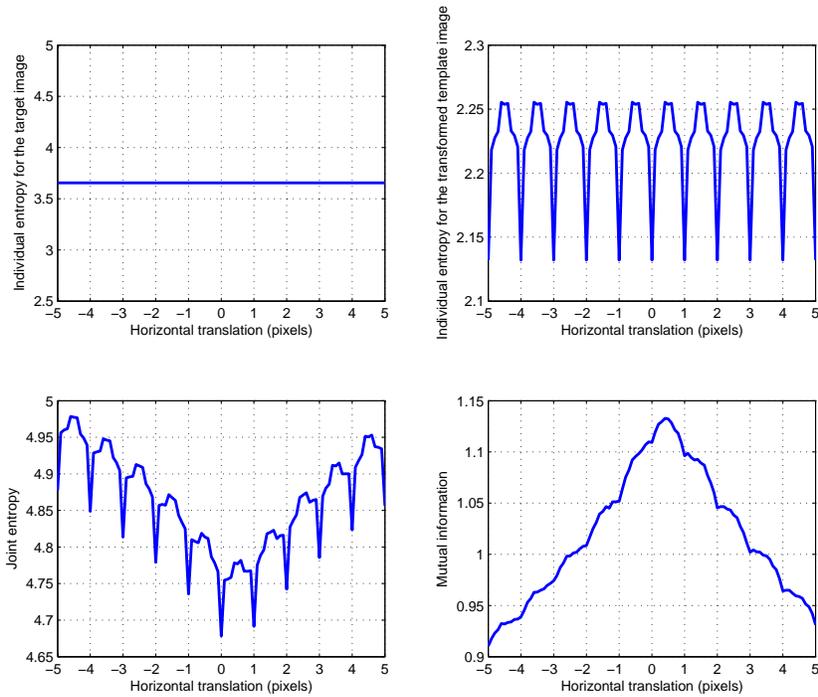


Figure 4.2: The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) using the histogramming model, as one of the interpolation-based models.

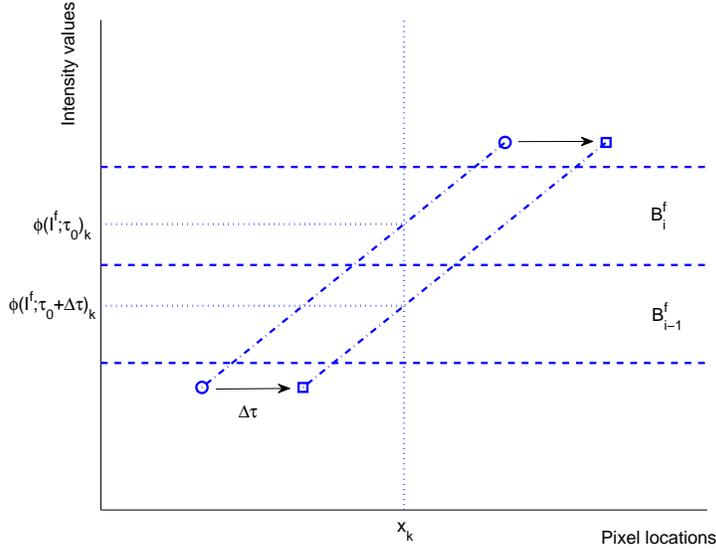


Figure 4.3: The interpolated intensity value in the transformed template image may cross the boundaries of bins and lead to discontinuous change of the probabilities.

and

$$J_{j-\frac{1}{2}}^f \stackrel{\text{def}}{=} \frac{1}{2} \left(J_{j-1}^f + J_j^f \right)$$

be the central intensity values at each bin for the target image and the transformed template image, respectively. Let $\mathcal{W}(\cdot)$ be the kernel function which can be used to substitute the characteristic function. Probability distributions are then calculated as

$$p_i^g = \frac{1}{\sum_m p_m^g} \sum_{k=1}^N \mathcal{W} \left(I_k^g - J_{i-\frac{1}{2}}^g \right) \quad i = 1, \dots, N_B^g,$$

$$p_j^f(\mathbf{s}) = \frac{1}{\sum_n p_n^f(\mathbf{s})} \sum_{k=1}^N \mathcal{W} \left(\phi(I^f; \mathbf{s})_k - J_{j-\frac{1}{2}}^f \right) \quad j = 1, \dots, N_B^f, \quad (4.7)$$

and

$$p_{i,j}^{fg}(\mathbf{s}) = \frac{1}{\sum_{m,n} p_{m,n}^{fg}(\mathbf{s})} \sum_{k=1}^N \mathcal{W} \left((I_k^g, \phi(I^f; \mathbf{s})_k) - (J_{i-\frac{1}{2}}^g, J_{j-\frac{1}{2}}^f) \right)$$

$$i = 1, \dots, N_B^g; j = 1, \dots, N_B^f. \quad (4.8)$$

The kernel function $\mathcal{W}(\cdot)$ is often chosen to satisfy

1. $\mathcal{W} \in \mathcal{C}^0$,
2. $\int_{-\infty}^{+\infty} \mathcal{W}(\alpha) d\alpha = 1$.

The first condition is to guarantee that mutual information is at least continuous. The second one is to guarantee that it is consistent with the idea to construct the probability distributions. Common weighting functions \mathcal{W} include Gaussian functions, double exponential functions, and splines. The resulting mutual information functions are then continuous. However, they are still not smooth enough. Note that in (4.7) and (4.8), the normalization factors $1/\sum_n p_n^f(\mathbf{s})$ and $1/\sum_{m,n} p_{m,n}^{fg}(\mathbf{s})$ depend on the transformations and thus may not be smooth. Accordingly, $p_j^f(\mathbf{s})$ and $p_{i,j}^{fg}(\mathbf{s})$ are not smooth and so are H^f and H^{fg} . Finally, the mutual information is in turn not smooth. Some special kernel functions such as B-spline functions [2] can avoid the normalization factors depending on transformations. However, there will still be interpolation artifacts phenomenon; see Section 4.3.

Since interpolation for the transformed template image is necessary in the transformation process to determine the pixel intensity value pairs, we call this kind of models as interpolation-based models. The flow chart of interpolation-based models are shown in Figure 4.4. Because interpolation is involved, from the perspective of the transformed template image, the sample locations change with respect to the transformations. Thus, the individual probability distribution and joint probability distribution will change with respect to the transformation. That is why the individual entropy for the transformed template image will be affected by the transformation, as shown in Figure 4.2. However, conceptually, the individual entropy for the transformed template image should not depend on transformations since the underlying image function and hence its intensity values do not change under rigid transformations.

4.2.2 Partial Volume Models

To solve the problem that the individual entropy depends on the transformation, the partial volume model [14] is introduced. Similar to the interpolation-based models, the partial volume model uses the intensity values of the target image to compute the individual probability distribution and the individual entropy for the target image in the same way as the histogramming model. However, different from the interpolation-based models, the individual probability distribution and the individual entropy for the transformed template image are also computed by taking the intensity values of the transformed template image at its own pixel locations in the same way as the individual probability distribution and the individual entropy for the target image. In this case, its intensity values do not depend on transformations and hence no interpolation is required. Thus, p_i^f and H^f is independent of the

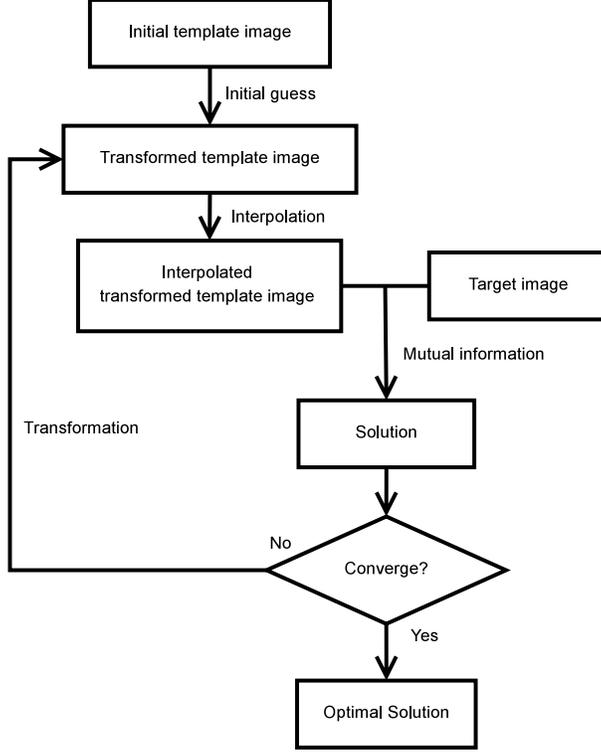


Figure 4.4: The flow chart of interpolation-based models.

transformation parameters. However, since the pixel locations in the transformed template image may not necessarily be aligned with those in the target image, we do not have the intensity pairs directly. The joint probability distribution in turn cannot be calculated directly. Other strategies need to be explored.

To explain how the joint entropy is calculated, we assume the target image and the template image are both in 2D for simplicity. When the template image is transformed, from another perspective, the transformation can be considered as the one applied to the coordinates of the image functions instead of the intensity values. After applying the transformation, the template image becomes $I^f(\xi(x, y))$, where (x, y) represents the coordinates of the image functions and $\xi(\cdot)$ denotes the transformation applied to the coordinates. Let (x_k^f, y_l^f) , $k = 1, \dots, N_x$ and $l = 1, \dots, N_y$, be the pixel locations in the template image, where $N_x \times N_y$ denotes the resolution of the template image. They become $\xi(x_k^f, y_l^f)$ after the transformation. In general, $\xi(x_k^f, y_l^f)$ do not align with the pixel locations in the target image. Instead, each pixel $\xi(x_k^f, y_l^f)$ generally is located in the box domain determined by four adjacent pixels in the target image. Let the four adjacent pixels be located at $(x_{k_0}^g, y_{l_0}^g)$, $(x_{k_1}^g, y_{l_0}^g)$, $(x_{k_0}^g, y_{l_1}^g)$, $(x_{k_1}^g, y_{l_1}^g)$, as shown in Figure 4.5.

We assume the size of every pixel is $h_x \times h_y$ and further denote

$$\begin{aligned}
(\Delta x, \Delta y) &\stackrel{\text{def}}{=} \xi \left(x_k^f, y_l^f \right) - \left(x_{k_0}^g, y_{l_0}^g \right), \\
\omega_{0,0} &\stackrel{\text{def}}{=} \frac{\Delta x \Delta y}{h_x h_y}, \\
\omega_{1,0} &\stackrel{\text{def}}{=} \frac{(h_x - \Delta x) \Delta y}{h_x h_y}, \\
\omega_{0,1} &\stackrel{\text{def}}{=} \frac{\Delta x (h_y - \Delta y)}{h_x h_y}, \\
\omega_{1,1} &\stackrel{\text{def}}{=} \frac{(h_x - \Delta x) (h_y - \Delta y)}{h_x h_y}.
\end{aligned}$$

The partial volume model computes the joint probability distribution by histogramming the intensity vector pairs of

$$\begin{aligned}
&\left(I^g \left(x_{k_0}^g, y_{l_0}^g \right), I^f \left(\xi \left(x_k^f, y_l^f \right) \right) \right), \\
&\left(I^g \left(x_{k_1}^g, y_{l_0}^g \right), I^f \left(\xi \left(x_k^f, y_l^f \right) \right) \right), \\
&\left(I^g \left(x_{k_0}^g, y_{l_1}^g \right), I^f \left(\xi \left(x_k^f, y_l^f \right) \right) \right), \\
&\left(I^g \left(x_{k_1}^g, y_{l_1}^g \right), I^f \left(\xi \left(x_k^f, y_l^f \right) \right) \right).
\end{aligned}$$

with the weights $\omega_{1,1}$, $\omega_{0,1}$, $\omega_{1,0}$, and $\omega_{0,0}$ respectively for each pixel in the transformed template image. More precisely, we compute the joint probability distribution by

$$\begin{aligned}
p_{i,j}^{fg}(\mathbf{s}) &= \frac{1}{N} \sum_{k=1}^{N_x} \sum_{l=1}^{N_y} \left(\omega_{1,1} \mathbf{1}_{B_{i,j}^{fg}} \left(I \left(x_{k_0}^g, y_{l_0}^g \right), I \left(\xi \left(x_k^f, y_l^f \right) \right) \right) + \right. \\
&\quad \omega_{0,1} \mathbf{1}_{B_{i,j}^{fg}} \left(I \left(x_{k_1}^g, y_{l_0}^g \right), I \left(\xi \left(x_k^f, y_l^f \right) \right) \right) + \quad (4.9) \\
&\quad \omega_{1,0} \mathbf{1}_{B_{i,j}^{fg}} \left(I \left(x_{k_0}^g, y_{l_1}^g \right), I \left(\xi \left(x_k^f, y_l^f \right) \right) \right) + \\
&\quad \left. \omega_{0,0} \mathbf{1}_{B_{i,j}^{fg}} \left(I \left(x_{k_1}^g, y_{l_1}^g \right), I \left(\xi \left(x_k^f, y_l^f \right) \right) \right) \right) \\
&\quad i = 1, \dots, N_B^g; j = 1, \dots, N_B^f,
\end{aligned}$$

where $N = N_x \times N_y$ represents the total number of pixels.

The idea of this model is to distribute the weight of each pair of pixels into different bins for constructing the joint probability distribution so that the individual entropies are independent of the transformations, the joint probability distributions change more smoothly with respect to the transformation, and the joint entropy and the mutual information are in turn smoothed. Figure 4.6 shows the individual

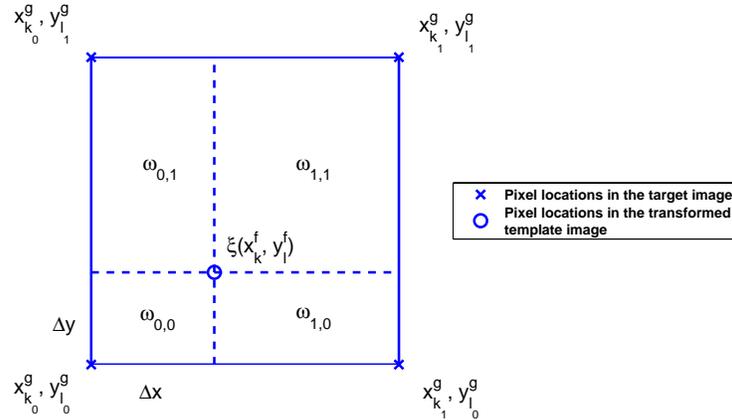


Figure 4.5: A pixel in the transformed template image is located among four pixels in the target image.

entropy for the target image, the individual entropy for the transformed template image, the joint entropy, and the mutual information with respect to the horizontal translation using the partial volume model. The images and parameters for the experiment are the same with the ones for the histogramming model. It can be clearly observed that the individual entropies for both the target and the transformed template image are independent of the transformation. Also, the mutual information attains its maximum value when the two images are roughly aligned.

4.3 Interpolation Artifacts

Figure 4.2 and Figure 4.6 show an interesting phenomenon. Assuming only spatial translation is considered in the transformation parameter sets, the mutual information functions are not differentiable at the integer pixel translation. In other words, during the transformation process, whenever all or most of the pixel locations in the transformed template image are aligned with those in the target image, mutual information is not smooth and an artifact will happen. Similar observations have been reported by others [2, 3, 4, 10, 11, 13, 19, 22, 27, 33].

The reason of this phenomenon varies in different models. In order to analyze the reason, we assume the two images are originally aligned, the pixel sizes of the target and the template images are the same, and only the translation in one axis (i.e., either horizontal or vertical translation) is considered. Let τ be the translation in the transformation parameter set in one direction and h be the pixel size in the corresponding direction.

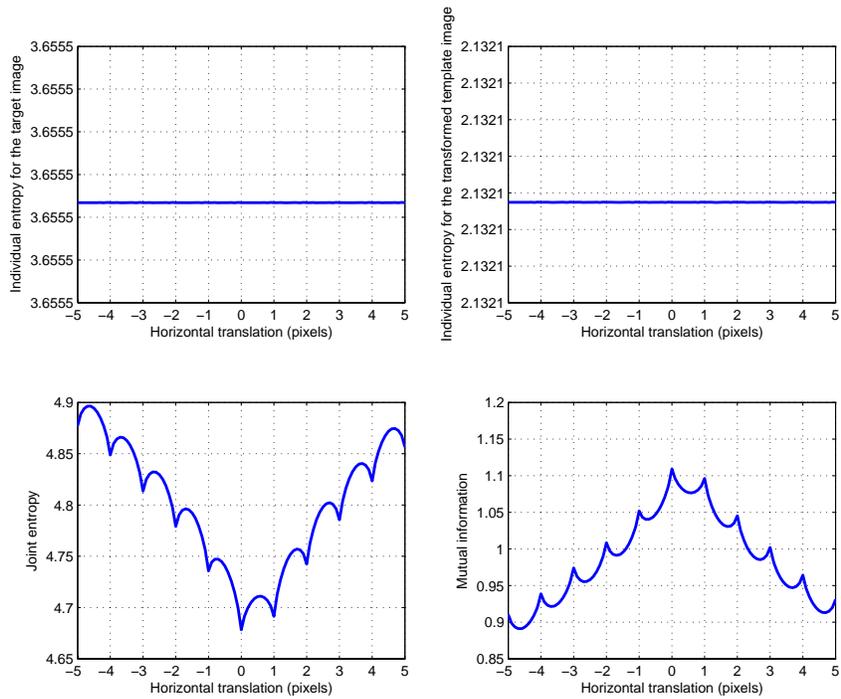


Figure 4.6: The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) using the partial volume model.

In interpolation-based models, the interpolated intensity values in the transformed template image change periodically with respect to the translation τ (every pixel interval h). That leads to the artifacts in the individual entropy for the transformed template image, the joint entropy, and in turn the mutual information at the integer pixel translation.

In the partial volume model, the artifact phenomenon can be explained by a mathematical analysis. Suppose τ changes from $\tau_0 = n_0 h$ to $\tau_1 = (n_0 + 1)h$ for some integer n_0 . By (4.1), we can rewrite the joint entropy as

$$H^{fg} = - \sum_{i,j} \gamma \left(p_{i,j}^{fg}(\tau) \right),$$

where

$$\gamma(\alpha) = \alpha \log \alpha.$$

By (4.9), it is known that $p_{i,j}^{fg}(\tau)$ is a linear function in τ . Let $p_{i,j}^0 = p_{i,j}^{fg}(\tau_0)$ and $p_{i,j}^1 = p_{i,j}^{fg}(\tau_1)$. Then $p_{i,j}^{fg}(\tau)$ can be written as

$$p_{i,j}^{fg}(\tau) = \frac{\tau_1 - \tau}{h} p_{i,j}^0 + \frac{\tau - \tau_0}{h} p_{i,j}^1 \quad \tau \in [\tau_0, \tau_1].$$

The first derivative of $\gamma(p_{i,j}^{fg}(\tau))$ can be written as

$$\begin{aligned} \frac{d\gamma \left(p_{i,j}^{fg}(\tau) \right)}{d\tau} &= \frac{d\gamma \left(p_{i,j}^{fg} \right)}{dp_{i,j}^{fg}} \cdot \frac{dp_{i,j}^{fg}(\tau)}{d\tau} \\ &= \frac{1}{h} \left(\log p_{i,j}^{fg}(\tau) + \frac{1}{\ln 2} \right) (p_{i,j}^1 - p_{i,j}^0). \end{aligned}$$

That results in the first derivative of H^{fg} to be

$$\frac{dH^{fg}}{d\tau} = -\frac{1}{h} \sum_{i,j} \left(\log p_{i,j}^{fg}(\tau) + \frac{1}{\ln 2} \right) (p_{i,j}^1 - p_{i,j}^0).$$

Let $\tau_2 = (n_0 + 2)h$ and $p_{i,j}^2 = p_{i,j}^{fg}(\tau_2)$. We have

$$\begin{aligned} \lim_{\tau \rightarrow \tau_1^-} \frac{dH^{fg}(\tau)}{d\tau} &= -\frac{1}{h} \sum_{i,j} \left(\log p_{i,j}^1 + \frac{1}{\ln 2} \right) (p_{i,j}^1 - p_{i,j}^0), \\ \lim_{\tau \rightarrow \tau_1^+} \frac{dH^{fg}(\tau)}{d\tau} &= -\frac{1}{h} \sum_{i,j} \left(\log p_{i,j}^1 + \frac{1}{\ln 2} \right) (p_{i,j}^2 - p_{i,j}^1). \end{aligned}$$

In general,

$$\lim_{\tau \rightarrow \tau_1^-} \frac{dH^{fg}(\tau)}{d\tau} \neq \lim_{\tau \rightarrow \tau_1^+} \frac{dH^{fg}(\tau)}{d\tau}.$$

Therefore, $\frac{dH^{fg}}{d\tau}$ is not continuous and thus H^{fg} is not smooth at the point $\tau = \tau_1$. We further examine the second derivative of $\gamma(p_{i,j}^{fg})$ in the interval of (τ_0, τ_1) . In the case that $p_{i,j}^0 \neq 0$ or $p_{i,j}^1 \neq 0$,

$$p_{i,j}^{fg}(\tau) \neq 0 \quad \forall \tau \in (\tau_0, \tau_1).$$

We then have

$$\frac{d^2\gamma(p_{i,j}^{fg}(\tau))}{d\tau^2} = \frac{1}{\ln 2} \cdot \frac{(p_{i,j}^1 - p_{i,j}^0)^2}{h^2 p_{i,j}^{fg}(\tau)} \geq 0.$$

Accordingly, $\gamma(p_{i,j}^{fg})$ is a convex function. In the case that $p_{i,j}^0 = p_{i,j}^1 = 0$,

$$p_{i,j}^{fg}(\tau) = 0.$$

It is obviously a convex function. Therefore, $\gamma(p_{i,j}^{fg})$ is a convex function in (τ_0, τ_1) . By the definition of convex functions,

$$\begin{aligned} \gamma(p_{i,j}^{fg}(\tau)) &= \gamma\left(\frac{\tau_1 - \tau}{h} p_{i,j}^0 + \frac{\tau - \tau_0}{h} p_{i,j}^1\right) \\ &\leq \frac{\tau_1 - \tau}{h} \gamma(p_{i,j}^0) + \frac{\tau - \tau_0}{h} \gamma(p_{i,j}^1) \quad \forall \tau \in [\tau_0, \tau_1]. \end{aligned}$$

Consequently,

$$\begin{aligned} H^{fg}(\tau) &= - \sum_{i,j} \gamma(p_{i,j}^{fg}(\tau)) \\ &\geq \frac{\tau_1 - \tau}{h} \left(- \sum_{i,j} \gamma(p_{i,j}^0) \right) + \frac{\tau - \tau_0}{h} \left(- \sum_{i,j} \gamma(p_{i,j}^1) \right) \\ &= \frac{\tau_1 - \tau}{h} H^{fg}(\tau_0) + \frac{\tau - \tau_0}{h} H^{fg}(\tau_1). \end{aligned}$$

That proves the joint entropy is a concave function in the interval (τ_0, τ_1) . Similarly, we can also prove that the joint entropy is a concave function in the interval of (τ_1, τ_2) . Note that the joint entropy is not continuous at τ_1 . Therefore, the joint entropy is not smooth and there is often a local minimum at τ_1 . Because the individual entropy for the target image and the individual entropy for the transformed template image are both independent of τ , mutual information only depends on the joint entropy: the shape of mutual information should be exactly the same as $-H^{fg}$ except for a constant difference. Thus, the mutual information will have an artifact at τ_1 . The result can be extended when $\tau = nh$, $n \in \mathbb{Z}$. The mutual information is therefore a piecewise convex function but not smooth at the points where the pixel locations in the template image are all aligned with those of the target image.

We refer to this phenomenon as interpolation artifacts. Increasing the sizes of the bins for computing probability distributions can help smooth mutual information functions. However, the artifacts are hard to be removed totally. Figure 4.7 and Figure 4.8 show the mutual information functions with the number of bins as 64 and 32 for both the target and the transformed template images using the histogramming model and the partial volume model, respectively. The image pairs for registration are the same as the ones used in Figure 4.2 and Figure 4.6. Comparing those plots together with Figure 4.2 and Figure 4.6, it can be observed that the mutual information functions get smoother with the increase of the bin sizes, or decrease of the numbers of bins. However, artifacts are hard to remove and can still be observed. Furthermore, on the other hand, making bin size too large will also decrease the accuracy of mutual information [10].

Other approaches for reducing interpolation artifacts include using a generalized partial volume model [3], resizing the pixel size [27], and jittering or blurring images [27]. However, interpolation artifacts have never been totally removed.

4.4 Optimization Methods

Artifacts hamper the performance of optimization methods. Most of the derivative-based optimization methods cannot be applied to these models because the mutual information functions are not smooth. Some derivative-free optimization methods such as simplex methods [3, 4], Powell’s method [8], simulated annealing [34], and genetic algorithms [33], occasionally succeed. Up to now, there is no optimization method that can totally solve this kind of problems.

There are two different optimization methods applied on the mutual information functions in this thesis. One is a trust region method for non-linear optimization problems [7], which is a derivative-based method. It is provided by courtesy of Professor Yuying Li from the School of Computer Science at the University of Waterloo. In this method, a local optimum is found based on the gradient and curvature information of the objective function. At each iteration, the method approximates the function in a certain trust region as a local quadratic function and further finds its optimum. This method requires objective functions to be smooth enough to guarantee quick convergence to optimal solutions. In this method, the gradient information and Hessian matrix information of the objective function are required, and they are approximated using the finite difference method in our approach, see Section 6.4.

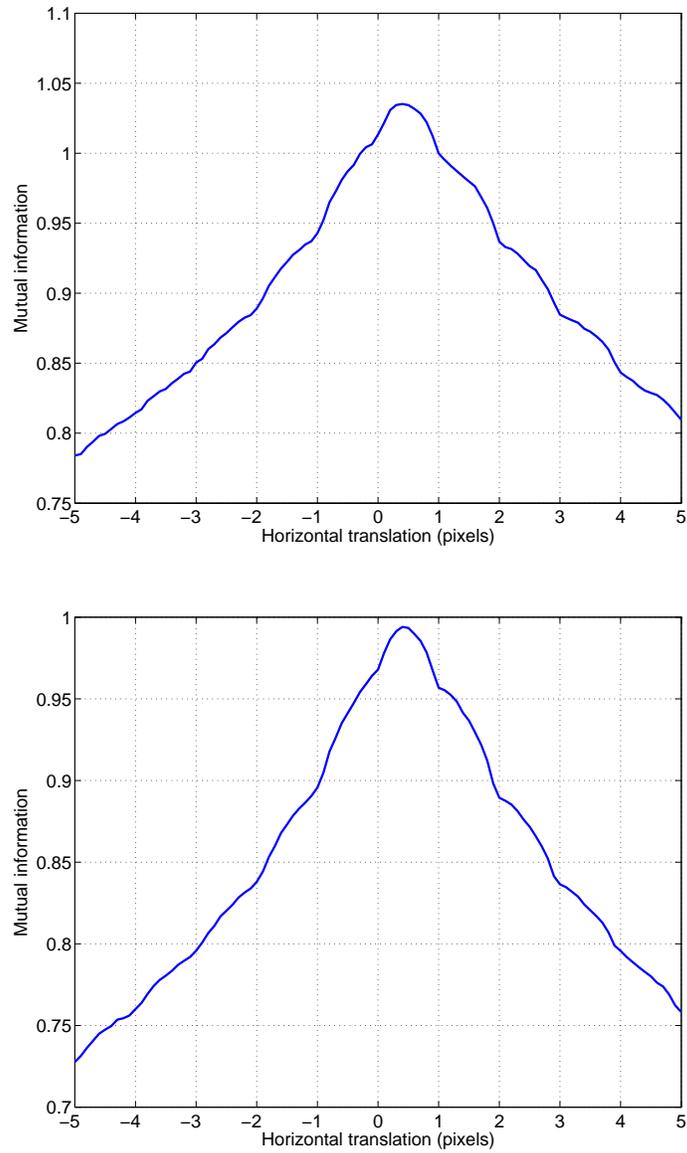


Figure 4.7: Mutual information using the histogramming model with the numbers of bins as 64 (upper) and 32 (lower) for both the target and the transformed template images. Although the mutual information function gets smoother with the increase of bin sizes, the interpolation artifacts can still be observed.

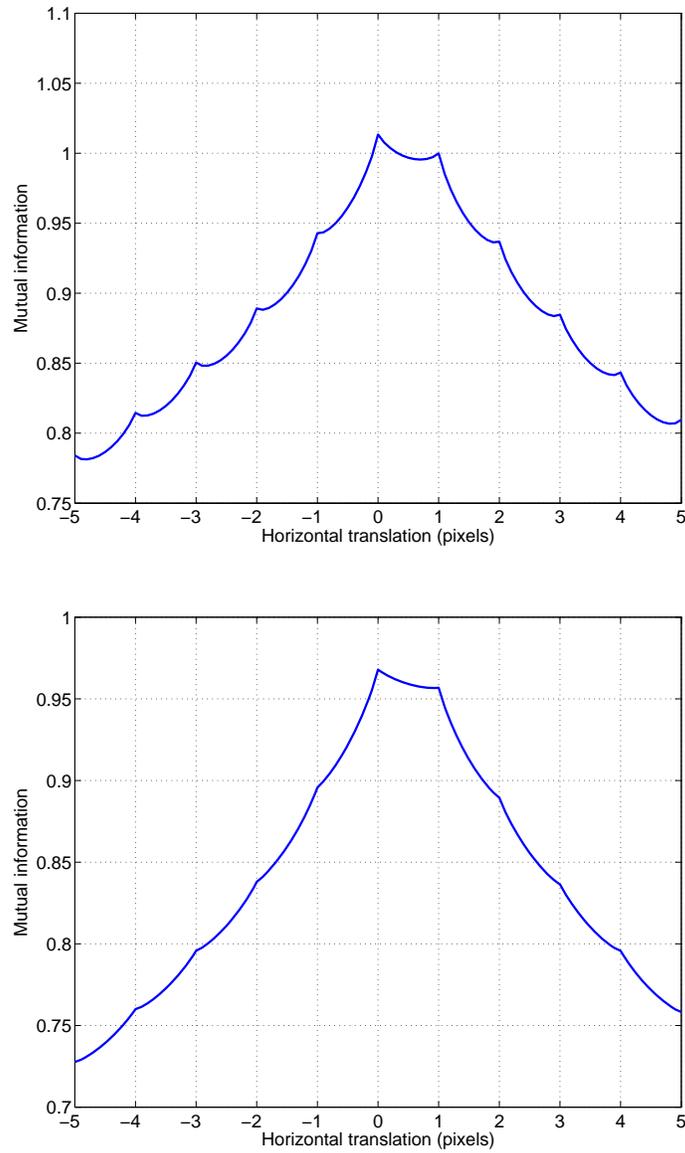


Figure 4.8: Mutual information using the partial volume model with the numbers of bins as 64 (upper) and 32 (lower) for both the target and the transformed template images. Although the mutual information function gets smoother with the increase of bin sizes, the interpolation artifacts can still be observed.

The other one is the Nelder-Mead method [17], which is a derivative-free method. This method defines a simplex in the domain of the objective function. In each iteration, it tests the function values at the test points arranged by the simplex, and replaces a part of the test points with new test points. The simplex is expected to move to one of the local maxima eventually.

Chapter 5

Continuous Models

Artifacts in mutual information functions prevent standard optimization methods converging to the optimal transformation solutions for aligning images. Even if some methods can attain the optimal solutions in some scenarios, the convergence is normally slow, and the sub-pixel accuracy is not easy to obtain. To increase the efficiency as well as the accuracy of registration, it is necessary to remove the artifacts. Previous approaches can help smooth mutual information functions, but none of them can totally remove artifacts. In this chapter, a novel model is proposed in order to remove the artifacts and further improve the efficiency and accuracy of the optimization methods after the new model is applied.

We start from the motivation, followed by the detailed explanation of the new model. Finally, we discuss the relationship between the partial volume model and our new model.

5.1 Motivation

The fundamental reason why mutual information is not smooth in the interpolation-based models and the partial volume model is because images are treated as continuous functions so that the use of interpolation can make sense, but the formulas used for computing probability distributions, entropies, and mutual information are all based on discrete random variables for image intensity values. This discrepancy eventually leads to the artifacts in mutual information functions. In contrast, our model treats image intensity values as continuous random variables and exploits the formulas based on continuous random variables to compute the PDFs, entropies,

and mutual information. Our idea is to directly address the issue of the discrepancy. Since both images and formulas are based on continuous random variables, the new model is called a continuous model.

5.2 One Dimensional Image Registration

We first explain the continuous model in the case that both the target image and the template image are in 1D.

5.2.1 Image Functions

Let the resolution of both the target image and the template image be N , and the 1D arrays $\{I_i^g\}_{i=0}^{N-1}$ and $\{I_j^f\}_{j=0}^{N-1}$ be the observable intensity values of the target image and the template image, respectively. To model an image function defined in a continuous domain from the observable pixels, one of the simplest ways is using linear interpolation. Note that other interpolation strategies can also be used. Also, for easy exposition, we assume the images are periodic and continuous. Ghost points I_N^g and I_N^f are added for the target image and the template image, respectively, satisfying

$$\begin{aligned} I_N^g &= I_0^g, \\ I_N^f &= I_0^f. \end{aligned}$$

Let $\{x_i\}_{i=0}^N$ be the pixel locations for the images, satisfying $x_i = ih$, where h is the pixel size. We denote the functions $I^g(\cdot)$ and $I^f(\cdot)$ as the linear interpolants for $\{(x_i, I_i^g)\}_{i=0}^N$ and $\{(x_j, I_j^f)\}_{j=0}^N$, respectively. Formally, $I^g(\cdot)$ satisfies

$$I^g(x_i) = I_i^g \quad i = 0, \dots, N,$$

and $I^f(\cdot)$ satisfies

$$I^f(x_j) = I_j^f \quad j = 0, \dots, N.$$

In each subinterval $[x_{k-1}, x_k]$, $k = 1, \dots, N$, the target image I^g and the template image I^f are linear functions; i.e., $\forall x \in [x_{k-1}, x_k]$,

$$\begin{aligned} I^g(x) &= \frac{x_k - x}{h} I_{k-1}^g + \frac{x - x_{k-1}}{h} I_k^g, \\ I^f(x) &= \frac{x_k - x}{h} I_{k-1}^f + \frac{x - x_{k-1}}{h} I_k^f. \end{aligned}$$

In the 1D case, there is only one element, translation τ , in the transformation parameter set \mathbf{s} . Thus, the transformed template image can be simply rewritten as

$$\phi(I^f; \mathbf{s}) = \phi(I^f; \tau).$$

Also, translating an image is equivalent to translating the corresponding 1D image function. Mathematically, $\forall \alpha \in \mathbb{R}$, the transformed template image has the property

$$\phi(I^f; \tau)(x) = \phi(I^f; \tau + \alpha)(x + \alpha). \quad (5.1)$$

5.2.2 Individual Probability Density Functions

We first construct the individual PDF for the target image I^g . By assumptions, I^g is a piecewise linear function. For simplicity, we normalize the image length as 1, and hence the pixel size $h = \frac{1}{N}$. In order to construct the PDF for the target image, it is natural to first consider the construction of the PDF in each subinterval where I^g is a simple linear function.

In the subinterval $[x_{i-1}, x_i)$ where $i \in \{1, \dots, N\}$, we first discuss the case that $I_{i-1}^g < I_i^g$, as shown in Figure 5.1. In the histogramming models, the probability in each bin is quantified using the normalized frequency of the pixels whose intensity values are in that bin. As a counterpart, in the continuous case, for any $\alpha \in \mathbb{R}$, the cumulative distribution function, $F_i^g(\alpha)$, which is formally defined as the probability of intensity value being in the interval $(-\infty, \alpha]$, should be equal to the proportion of the measure of the image domain whose intensity value is in that subinterval over the length of the subinterval $[x_{i-1}, x_i)$. Since the image function is linear in the subinterval $[x_{i-1}, x_i)$, F_i^g should also be a linear function in the subinterval $[I_{i-1}^g, I_i^g)$. Also, obviously, $F_i^g(\alpha) = 0$ when $\alpha < I_{i-1}^g$ and $F_i^g = 1$ when $\alpha \geq I_i^g$. Formally,

$$F_i^g(\alpha) \stackrel{\text{def}}{=} \mathbf{P} \{i^g \in (-\infty, \alpha]\} = \begin{cases} 0 & \text{if } \alpha \in (-\infty, I_{i-1}^g), \\ \frac{\alpha - I_{i-1}^g}{I_i^g - I_{i-1}^g} & \text{if } \alpha \in [I_{i-1}^g, I_i^g), \\ 1 & \text{if } \alpha \in [I_i^g, +\infty), \end{cases} \quad (5.2)$$

where i^g denotes the observation of I^g . Since F_i^g is almost everywhere differentiable, its first derivative can be used as the PDF. Let p_i^g be the PDF corresponding to

the interval $[x_{i-1}, x_i)$. It can be defined as

$$p_i^g(\alpha) = \frac{d}{d\alpha} F_i^g(\alpha) = \begin{cases} 0 & \text{if } \alpha \in (-\infty, I_{i-1}^g), \\ \frac{1}{I_i^g - I_{i-1}^g} & \text{if } \alpha \in [I_{i-1}^g, I_i^g), \\ 0 & \text{if } \alpha \in [I_i^g, +\infty). \end{cases} \quad (5.3)$$

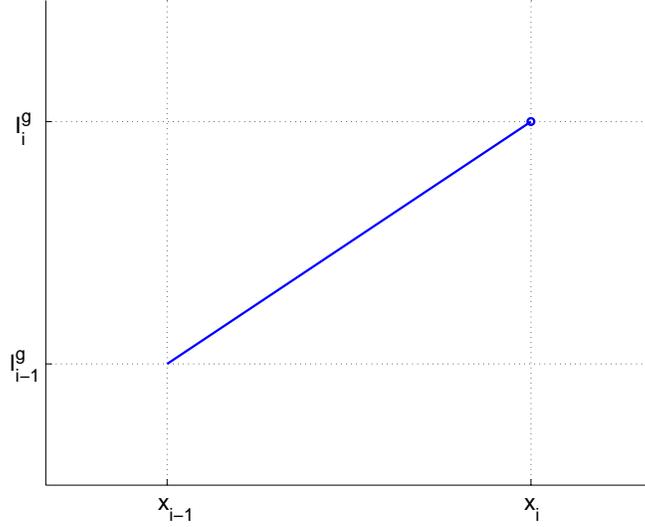


Figure 5.1: The target image in the case that $I_{i-1}^g < I_i^g$.

In the case that $I_{i-1}^g > I_i^g$, as shown in Figure 5.2, following the similar idea, p_i^g can be defined as

$$p_i^g(\alpha) = \frac{d}{d\alpha} F_i^g(\alpha) = \begin{cases} 0 & \text{if } \alpha \in (-\infty, I_i^g], \\ \frac{1}{I_{i-1}^g - I_i^g} & \text{if } \alpha \in (I_i^g, I_{i-1}^g], \\ 0 & \text{if } \alpha \in (I_{i-1}^g, +\infty). \end{cases} \quad (5.4)$$

A special case needed to be considered is $I_{i-1}^g = I_i^g$, as shown in Figure 5.3. By the analysis above, it is clear that

$$p_i^g(\alpha) = 0 \quad \forall \alpha \neq I_i^g.$$

By the definition of PDFs, p_i^g should also satisfy

$$\int_{-\infty}^{+\infty} p_i^g(\alpha) d\alpha = 1.$$

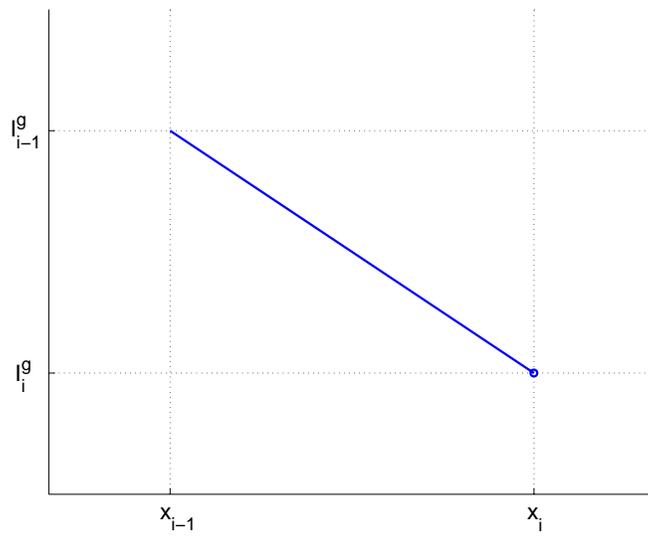


Figure 5.2: The target image in the case that $I_{i-1}^g > I_i^g$.

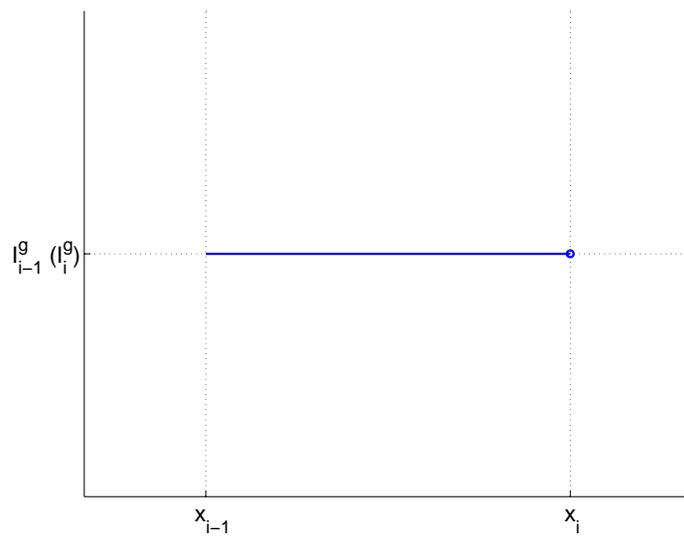


Figure 5.3: The target image in the case that $I_{i-1}^g = I_i^g$.

Considering p_i^g has similar properties as the Dirac delta function $\delta(\cdot)$, we model it as

$$p_i^g(\alpha) = \delta(\alpha - I_i^g).$$

Since all of the subintervals have the same length, p_i^g should contribute to the PDF for the target image equally. Formally, let the PDF for the target image be p^g . It is defined as the normalized summation of the PDFs corresponding to each subinterval $[x_{i-1}, x_i)$. Combining those three different cases, let

$$\begin{aligned}\mathcal{J}^g &\stackrel{\text{def}}{=} \{j | I_{j-1}^g < I_j^g\}, \\ \mathcal{K}^g &\stackrel{\text{def}}{=} \{k | I_{k-1}^g > I_k^g\}, \\ \mathcal{L}^g &\stackrel{\text{def}}{=} \{l | I_{l-1}^g = I_l^g\}.\end{aligned}$$

Then p^g can finally be written as

$$\begin{aligned}p^g(\alpha) &= \frac{1}{N} \sum_{i=1}^N p_i^g(\alpha) \\ &= \frac{1}{N} \left(\sum_{j \in \mathcal{J}^g} \frac{\mathbf{1}_{[I_{j-1}^g, I_j^g)}(\alpha)}{I_j^g - I_{j-1}^g} + \sum_{k \in \mathcal{K}^g} \frac{\mathbf{1}_{(I_k^g, I_{k-1}^g]}(\alpha)}{I_{k-1}^g - I_k^g} + \sum_{l \in \mathcal{L}^g} \delta(\alpha - I_l^g) \right),\end{aligned}\tag{5.5}$$

where $\frac{1}{N}$ is a normalization factor. Note that $p^g(\alpha)$ is a function combined with piecewise constant functions and Dirac delta functions. In Section 5.2.7, an example is shown to illustrate how the individual PDF is constructed.

We model the individual PDF for the transformed template image similarly. Let p^f be the individual PDF for the transformed template image. Following the same idea for constructing the individual PDF for the target image, the individual PDF for the transformed template image can be written as

$$p^f(\alpha; \tau) = \frac{1}{N} \sum_{i=1}^N p_i^f(\alpha; \tau),\tag{5.6}$$

where $p_i^f(\alpha; \tau)$ denotes the individual PDF for the transformed template image corresponding to the subinterval $[x_{i-1} + \tau, x_i + \tau)$. Here, α denotes the variable of the image intensity value and τ denotes the translation parameter.

By (5.1), $\forall \tau \neq 0$, the transformed template image $\phi(I^f; \tau)$ can be written as

$$\phi(I^f; \tau)(x) = \phi(I^f; 0)(x - \tau) = I^f(x - \tau).$$

Therefore, $\phi(I^f; \tau)$ can be treated as a shifted function of I^f . By the idea for constructing PDFs, it is clear that the shifted function generates the same PDF as

the initial one so that $p_i^f(\alpha; \tau)$ and $p_i^f(\alpha; 0)$ are the same. Thus, the PDF in (5.6) is independent of the transformation, and the PDF for the transformed template image is the same as the PDF for the template image. We omit the parameter τ , and simply denote the individual PDF for the transformed template image as $p^f(\alpha)$, which is exactly the PDF for the template image.

Similar to the case for the target image, let

$$\begin{aligned}\mathcal{J}^f &\stackrel{\text{def}}{=} \{j | I_{j-1}^f < I_j^f\}, \\ \mathcal{K}^f &\stackrel{\text{def}}{=} \{k | I_{k-1}^f > I_k^f\}, \\ \mathcal{L}^f &\stackrel{\text{def}}{=} \{l | I_{l-1}^f = I_l^f\}.\end{aligned}$$

The PDF for the transformed template image can be written as

$$\begin{aligned}p^f(\alpha) &= \frac{1}{N} \sum_{i=1}^N p_i^f(\alpha) \\ &= \frac{1}{N} \left(\sum_{j \in \mathcal{J}^f} \frac{\mathbf{1}_{[I_{j-1}^f, I_j^f)}(\alpha)}{I_j^f - I_{j-1}^f} + \sum_{k \in \mathcal{K}^f} \frac{\mathbf{1}_{(I_k^f, I_{k-1}^f]}(\alpha)}{I_{k-1}^f - I_k^f} + \sum_{l \in \mathcal{L}^f} \delta(\alpha - I_l^f) \right).\end{aligned}\tag{5.7}$$

5.2.3 Joint Probability Density Functions

To model the joint PDF, the target and the transformed template images need to be considered together. The target image $I^g(x)$ is a piecewise linear function determined by the data points $\{x_i, I_i^g\}$, $i = 0, \dots, N$, and the transformed template image $\phi(I^f; \tau)$ is also a piecewise linear function. Since we assume the template image is periodic for simplicity, the function is determined by the node points $\{(x_i + \tau) \pmod{1}\}$, $i = 0, \dots, N - 1$. Let $\Delta h = \tau \pmod{h}$. Then $\{(x_i + \tau) \pmod{1}\}$ can be simplified as $\{x_i + \Delta h\}$, $i = 0, \dots, N - 1$. Let

$$z_i \stackrel{\text{def}}{=} \begin{cases} x_{\frac{i}{2}} & \text{if } i \text{ is even,} \\ x_{\frac{i-1}{2}} + \Delta h & \text{if } i \text{ is odd.} \end{cases}$$

$\{z_i\}_{i=0}^{2N}$ is the union of the set of the node points in the target image and the set of the node points in the transformed template image. In each subinterval $[z_{i-1}, z_i)$, both the target image and the transformed template image are linear functions. Considering the joint PDF in each of those subintervals can finally facilitate us to construct the joint PDF.

For simplicity, we denote $I^g(z_i) = K_i^g$ and $\phi(I^f; \tau)(z_i) = K_i^f$. We first discuss the case that $K_{i-1}^g \neq K_i^g$ or $K_{i-1}^f \neq K_i^f$, as shown in Figure 5.4. In the domain

of the joint PDF, let \mathcal{C}_i be the segment determined by the end points (K_{i-1}^g, K_{i-1}^f) and (K_i^g, K_i^f) , which is defined as

$$\mathcal{C}_i \stackrel{\text{def}}{=} \{(\alpha, \beta) \mid t \in [0, 1)\},$$

where

$$\begin{aligned}\alpha(t) &= (K_i^g - K_{i-1}^g)t + K_{i-1}^g, \\ \beta(t) &= (K_i^f - K_{i-1}^f)t + K_{i-1}^f;\end{aligned}$$

see Figure 5.5. α denotes the intensity value of the target image and β denotes the intensity value of the transformed template image. Similar to the idea for constructing the individual PDF, the joint PDF p_i^{fg} associated with the images in the subinterval $[z_{i-1}, z_i)$ can be modeled as

$$p_i^{fg}(\alpha, \beta) = \begin{cases} \frac{1}{\|(K_i^g, K_i^f) - (K_{i-1}^g, K_{i-1}^f)\|_2} & \text{if } (\alpha, \beta) \in \mathcal{C}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (5.8)$$

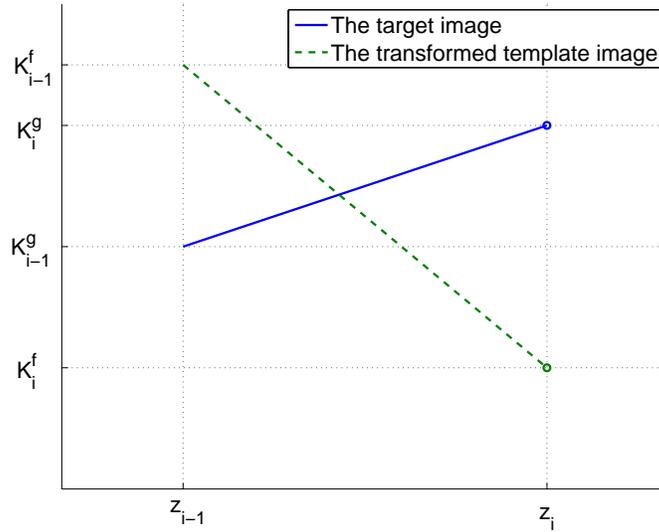


Figure 5.4: Images in the case that $K_{i-1}^g \neq K_i^g$ or $K_{i-1}^f \neq K_i^f$.

In the case that $K_{i-1}^g = K_i^g$ and $K_{i-1}^f = K_i^f$ as shown in Figure 5.6, p_i^{fg} can be modeled as

$$p_i^{fg}(\alpha, \beta) = \begin{cases} \delta(\alpha - K_i^g, \beta - K_i^f) & \text{if } (\alpha, \beta) = (K_i^g, K_i^f), \\ 0 & \text{otherwise.} \end{cases} \quad (5.9)$$

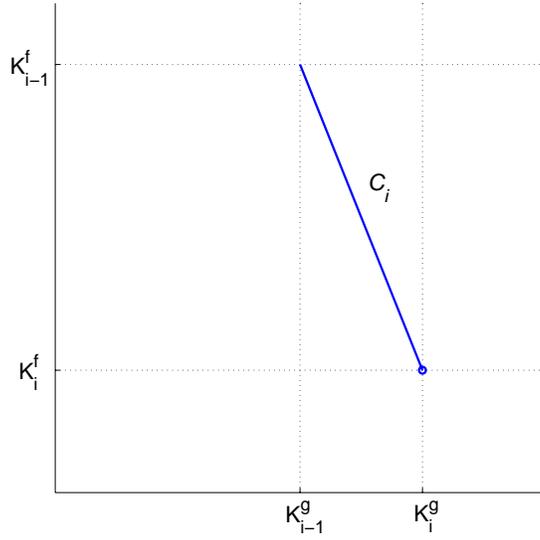


Figure 5.5: The domain of the joint PDF where $p_i^{fg} \neq 0$ is denoted by \mathcal{C}_i in the case that $K_{i-1}^g \neq K_i^g$ or $K_{i-1}^f \neq K_i^f$.

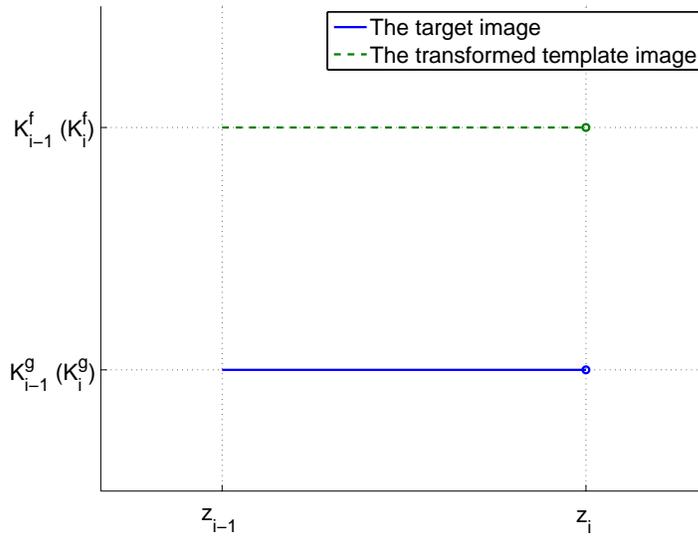


Figure 5.6: Images in the case that $K_{i-1}^g = K_i^g$ and $K_{i-1}^f = K_i^f$.

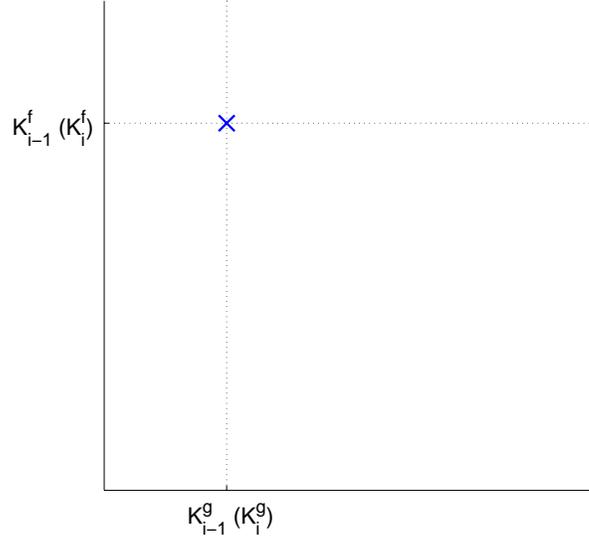


Figure 5.7: The point where p_i^{fg} is a Dirac delta function in the case that $K_{i-1}^g = K_i^g$ and $K_{i-1}^f = K_i^f$.

Figure 5.7 shows the point where the joint PDF is a Dirac delta function in the domain of the joint PDF.

The joint PDF is modeled by combining each p_i^{fg} . Considering the probability contributed by the images in each subinterval should be proportional to its own length, the joint PDF is defined as the normalized summation of p_i^{fg} with the weight proportional to the length of $[z_{i-1}, z_i)$. Considering the two cases in (5.8) and (5.9), let

$$\begin{aligned} \mathcal{U} &\stackrel{\text{def}}{=} \{u | K_{u-1}^g \neq K_u^g \text{ or } K_{u-1}^f \neq K_u^f\}, \\ \mathcal{V} &\stackrel{\text{def}}{=} \{v | K_{v-1}^g = K_v^g \text{ and } K_{v-1}^f = K_v^f\}. \end{aligned} \quad (5.10)$$

The joint PDF is finally defined as

$$\begin{aligned} p^{fg}(\alpha, \beta) &= \sum_{i=1}^{2N} \omega_i p_i^{fg}(\alpha, \beta) \\ &= \sum_{u \in \mathcal{U}} \omega_u \frac{\mathbf{1}_{\mathcal{C}_u}(\alpha, \beta)}{\|(K_u^g, K_u^f) - (K_{u-1}^g, K_{u-1}^f)\|_2} + \sum_{v \in \mathcal{V}} \omega_v \delta(\alpha - K_v^g, \beta - K_v^f), \end{aligned}$$

where

$$\omega_i = z_i - z_{i-1} = \begin{cases} \Delta h & \text{if } i \text{ is odd,} \\ h - \Delta h & \text{if } i \text{ is even.} \end{cases} \quad (5.11)$$

It can easily be verified that the projections of the joint PDF are the individual PDFs for the target and the transformed template image, which is consistent with the property of the joint PDF.

5.2.4 Individual Entropies

We now explain the computation of individual entropies. By definition, the entropies are computed by (4.3) and (4.4). Although individual PDFs for images can be modeled analytically by (5.5) and (5.7), the entropies are hard to compute analytically in general. Instead, we compute the entropies numerically. However, since the points where the PDFs are not continuous are a lot and the discontinuities are hard to be determined, standard numerical integration methods based on function evaluation of an interpolating polynomial, such as integration of an interpolating polynomial, composite integration, and Gaussian integration, would not be applicable here. Other strategies need to be exploited.

We first explain how the individual entropy for the target image is computed. Let $[0, I_{\max}^g]$ be the range of the intensity value of the target image. The PDF is possibly non-zero on that interval. To facilitate the integral computation, we divide the interval $[0, I_{\max}^g]$ into equal subintervals, or cells,

$$\begin{aligned} D_i^g &\stackrel{\text{def}}{=} [L_{i-1}^g, L_i^g) \quad i = 1, \dots, N_D^g - 1, \\ D_{N_D^g}^g &\stackrel{\text{def}}{=} [L_{N_D^g-1}^g, L_{N_D^g}^g], \end{aligned}$$

where $L_i^g = ih_D^g$, $h_D^g = \frac{I_{\max}^g}{N_D^g}$, and N_D^g is the number of cells. The individual entropy can be written as the summation of the integration in each cell; i.e.,

$$\begin{aligned} \tilde{H}^g &= - \int_{-\infty}^{+\infty} p^g(\alpha) \log p^g(\alpha) d\alpha \\ &= - \sum_{i=1}^{N_D^g} \int_{D_i^g} p^g(\alpha) \log p^g(\alpha) d\alpha. \end{aligned} \tag{5.12}$$

Considering the integrand $p^g(\alpha) \log p^g(\alpha)$ is not a continuous function, the integration in each cell is still quite difficult to compute analytically. Instead, we approximate the PDF as a constant function in each cell D_i^g with the value q_i^g computed by

$$q_i^g \stackrel{\text{def}}{=} \frac{1}{h_D^g} \int_{D_i^g} p^g(\alpha) d\alpha.$$

There are two advantages to approximate the PDF in each cell in this way. First, the PDF is approximated as a piecewise constant function, and so is $p^g \log p^g$. That

facilitates the approximation to the entropy in (5.12). Second, it is easy to verify that the integration of the approximate PDF over the whole domain is 1, which is consistent with the definition of the PDF.

By (5.5), q_i^g can be rewritten as

$$\begin{aligned} q_i^g &= \frac{1}{Nh_D^g} \int_{D_i^g} \sum_{k=1}^N p_k^g(\alpha) d\alpha \\ &= \frac{1}{Nh_D^g} \sum_{k=1}^N \int_{D_i^g} p_k^g(\alpha) d\alpha. \end{aligned} \quad (5.13)$$

Note that $\int_{D_i^g} p_k^g(\alpha) d\alpha$ is easy to compute since $p_k^g(\alpha)$ is either a constant function or a Dirac delta function.

After q_i^g is computed, the integrand can be approximated by

$$p^g(\alpha) \log p^g(\alpha) \approx q_i^g \log q_i^g \quad \forall \alpha \in D_i^g.$$

By (5.12), we can finally approximate the entropy by

$$\tilde{H}^g \approx \bar{H}^g \stackrel{\text{def}}{=} -h_D^g \sum_{i=1}^{N_D^g} q_i^g \log q_i^g. \quad (5.14)$$

There is an example shown in Section 5.2.7 to demonstrate how the individual PDF is computed.

For the transformed template image, we use the same idea to approximate its entropy \tilde{H}^f . By dividing the range of the intensity value of the template image $[0, I_{\max}^f]$ into cells

$$\begin{aligned} D_i^f &\stackrel{\text{def}}{=} [L_{i-1}^f, L_i^f), \quad i = 1, \dots, N_D^f - 1, \\ D_{N_D^f}^f &\stackrel{\text{def}}{=} [L_{N_D^f-1}^f, L_{N_D^f}^f], \end{aligned}$$

where $L_i^f = ih_D^f$, $h_D^f = \frac{I_{\max}^f}{N_D^f}$, and N_D^f is the number of cells. We approximate the PDF in each cell as a constant function with the value q_i^f , denoted by

$$q_i^f \stackrel{\text{def}}{=} \frac{1}{Nh_D^f} \sum_{k=1}^N \int_{D_i^f} p_k^f(\alpha) d\alpha. \quad (5.15)$$

The entropy is finally approximated by

$$\tilde{H}^f \approx \bar{H}^f \stackrel{\text{def}}{=} -h_D^f \sum_{i=1}^{N_D^f} q_i^f \log q_i^f. \quad (5.16)$$

Note that both the PDF for the target image p^g and the PDF for the transformed template image p^f are independent of the transformation. Thus, by (5.13) and (5.15), both q_i^g and q_i^f are independent of the transformation, and hence \bar{H}^g and \bar{H}^f are also independent of the transformation.

5.2.5 Joint Entropies and Mutual Information

Similar to individual entropies, we compute the joint entropy numerically. Let the joint PDF be defined over the domain

$$\Omega \stackrel{\text{def}}{=} \{(\alpha, \beta) \mid \alpha \in [0, I_{\max}^g], \beta \in [0, I_{\max}^f]\}.$$

By using the partition scheme for computing individual entropies, it is natural to divide Ω into cells

$$D_{i,j}^{fg} \stackrel{\text{def}}{=} \{(\alpha, \beta) \mid \alpha \in [K_{i-1}^g, K_i^g), \beta \in [K_{j-1}^f, K_j^f)\}.$$

The joint entropy can then be written as

$$\begin{aligned} \tilde{H}^{fg} &= - \iint_{\Omega} p^{fg}(\alpha, \beta) \log p^{fg}(\alpha, \beta) d\alpha d\beta \\ &= - \sum_{i=1}^{N_D^g} \sum_{j=1}^{N_D^f} \iint_{D_{i,j}^{fg}} p^{fg}(\alpha, \beta) \log p^{fg}(\alpha, \beta) d\alpha d\beta. \end{aligned}$$

Similar to the individual case, in each cell $D_{i,j}^{fg}$, we approximate the joint entropy by a constant function with the value $\bar{q}_{i,j}^{fg}$ computed by

$$\bar{q}_{i,j}^{fg} \stackrel{\text{def}}{=} \frac{1}{h_D^g h_D^f} \iint_{D_{i,j}^{fg}} p^{fg}(\alpha, \beta) d\alpha d\beta. \quad (5.17)$$

Note that the joint PDF p^{fg} is composed by a set of Dirac delta functions and a set of constant functions defined in segments which are zero measure subsets of Ω . Using the formula (5.17) will incur the loss of the contribution of piecewise constant functions to the joint entropy. To avoid that, we modify the formula (5.17) by substituting the double integral for the line integral where the joint PDFs associated with subintervals are not Dirac delta functions but still keep the double integral for Dirac delta functions, denoted by

$$\begin{aligned} q_{i,j}^{fg} &\stackrel{\text{def}}{=} \frac{1}{h_D^g h_D^f} \left(\int_{C_k \cap D_{i,j}^{fg}} \sum_{u \in \mathcal{U}} \omega_u p_u^{fg}(t) dt + \iint_{D_{i,j}^{fg}} \sum_{v \in \mathcal{V}} \omega_v p_v^{fg}(\alpha, \beta) d\alpha d\beta \right) \\ &= \frac{1}{h_D^g h_D^f} \left(\sum_{u \in \mathcal{U}} \omega_u \int_{C_k \cap D_{i,j}^{fg}} p_u^{fg}(t) dt + \sum_{v \in \mathcal{V}} \omega_v \iint_{D_{i,j}^{fg}} p_v^{fg}(\alpha, \beta) d\alpha d\beta \right), \end{aligned} \quad (5.18)$$

where \mathcal{U} and \mathcal{V} are defined in (5.10), and ω_u and ω_v are defined in (5.11). Since p_u^{fg} is constant functions and p_v^{fg} is Dirac delta functions, $\int_{C_k \cap D_{i,j}^{fg}} p_u^{fg}(t) dt$ and $\iint_{D_{i,j}^{fg}} p_v^{fg}(\alpha, \beta) d\alpha d\beta$ are easy to compute. Finally, $q_{i,j}^{fg}$ can be easily computed.

Let \bar{H}^{fg} be the approximate joint entropy. \tilde{H}^{fg} is then approximated by

$$\begin{aligned} \tilde{H}^{fg} &\approx \bar{H}^{fg} \stackrel{\text{def}}{=} - \sum_{i=1}^{N_D^g} \sum_{j=1}^{N_D^f} \iint_{D_{i,j}^{fg}} q_{i,j}^{fg} \log q_{i,j}^{fg} d\alpha d\beta \\ &= -h^g h^f \sum_{i=1}^{N_D^g} \sum_{j=1}^{N_D^f} q_{i,j}^{fg} \log q_{i,j}^{fg}. \end{aligned} \quad (5.19)$$

Finally, the mutual information is computed numerically using the approximate entropies \bar{H}^g , \bar{H}^f , and \bar{H}^{fg} . Let \bar{M} be the approximate mutual information. It is computed by

$$\bar{M} = \bar{H}^g + \bar{H}^f - \bar{H}^{fg}. \quad (5.20)$$

An example shown in Section 5.2.7 demonstrate the constant value of the approximate PDF in each bin $q_{i,j}^{fg}$ is a smooth function with respect to the translation τ . By (5.19), \tilde{H}^{fg} is in turn a smooth function. Also, considering \bar{H}^g and \bar{H}^f are constant functions, by (5.20), the approximate mutual information \bar{M} is finally a smooth function. The plots shown in Section 6.2 verify this point.

5.2.6 Complexity Analysis

By (5.13), the complexity for approximating the individual PDF for the target image as constant functions in cells is $O(\tilde{N}N)$, where \tilde{N} denotes the average number of cells where the PDF is nonzero corresponding to each pixel subinterval. Generally, \tilde{N} is around 2 or 3 on average if the image is not noisy. Then, by (5.14), with the knowledge of q_i^g , the complexity for computing the individual entropy is $O(N_D^g)$. Thus, the complexity for computing the individual entropy for the target image is $O(N_D^g + \tilde{N}N)$. Considering $N \gg N_D^g$ generally, the complexity is $O(\tilde{N}N)$.

Similarly, by (5.15) and (5.16), the complexity for computing the individual entropy for the transformed template image is also $O(\tilde{N}N)$.

Regarding the joint entropy, by (5.18), it is known that the complexity for computing every $q_{i,j}^{fg}$ is $O(\tilde{N}N)$. Then, by (5.19), it can be analyzed that, with the knowledge of $q_{i,j}^{fg}$, the complexity for computing the joint entropy is $O(N_D^g N_D^f)$.

Combining the two steps, the complexity for computing the joint entropy is $O(\tilde{N}N + N_D^g N_D^f)$. Considering $N \gg N_D^g N_D^f$, the complexity is $O(\tilde{N}N)$.

Finally, by (5.20), the complexity for computing the mutual information is also $O(\tilde{N}N)$.

It can be analyzed that the complexity of the histogramming model and the partial volume are both in $O(N + N_B^g N_B^f)$. Considering $N \sim N_B^g N_B^f$ in practice, the complexity of our new model is about the same with the histogramming model and the partial volume model.

5.2.7 Examples

An example is shown to clearly illustrate how the individual entropies are computed using our model. Suppose we have a continuous target image function shown in Figure 5.8 which is obtained from linear interpolation described in Section 5.2.1. For simplicity, we normalize the possible pixel value range as $[0, 1]$.

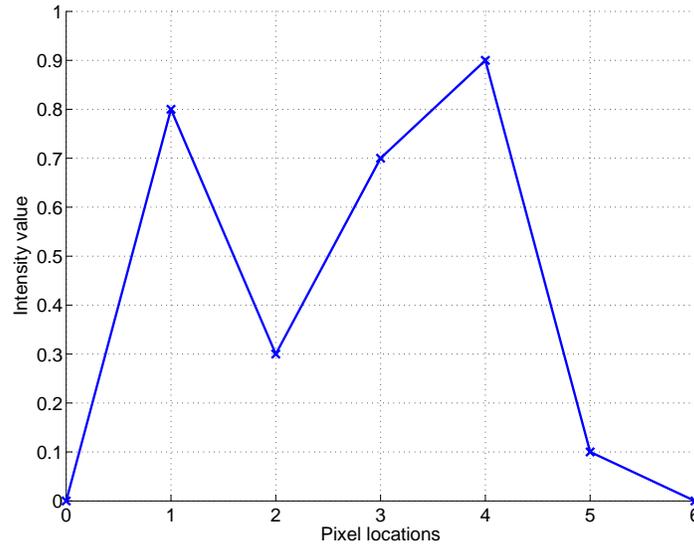


Figure 5.8: An example of an image function.

As explained in Section 5.2.2, we consider the PDF associated with the image in each pixel subinterval. Consider the image from pixel 0 to pixel 1. The intensity value changes linearly from 0 to 0.8. By (5.3), p_1^g is a piecewise constant function. It is nonzero over the intensity value range $[0, 0.8)$. The value of the constant is given by (5.3), which is 1.25. Similarly, p_2^g, \dots, p_6^g are constructed by either (5.3) or (5.4). They are shown in Figure 5.9.

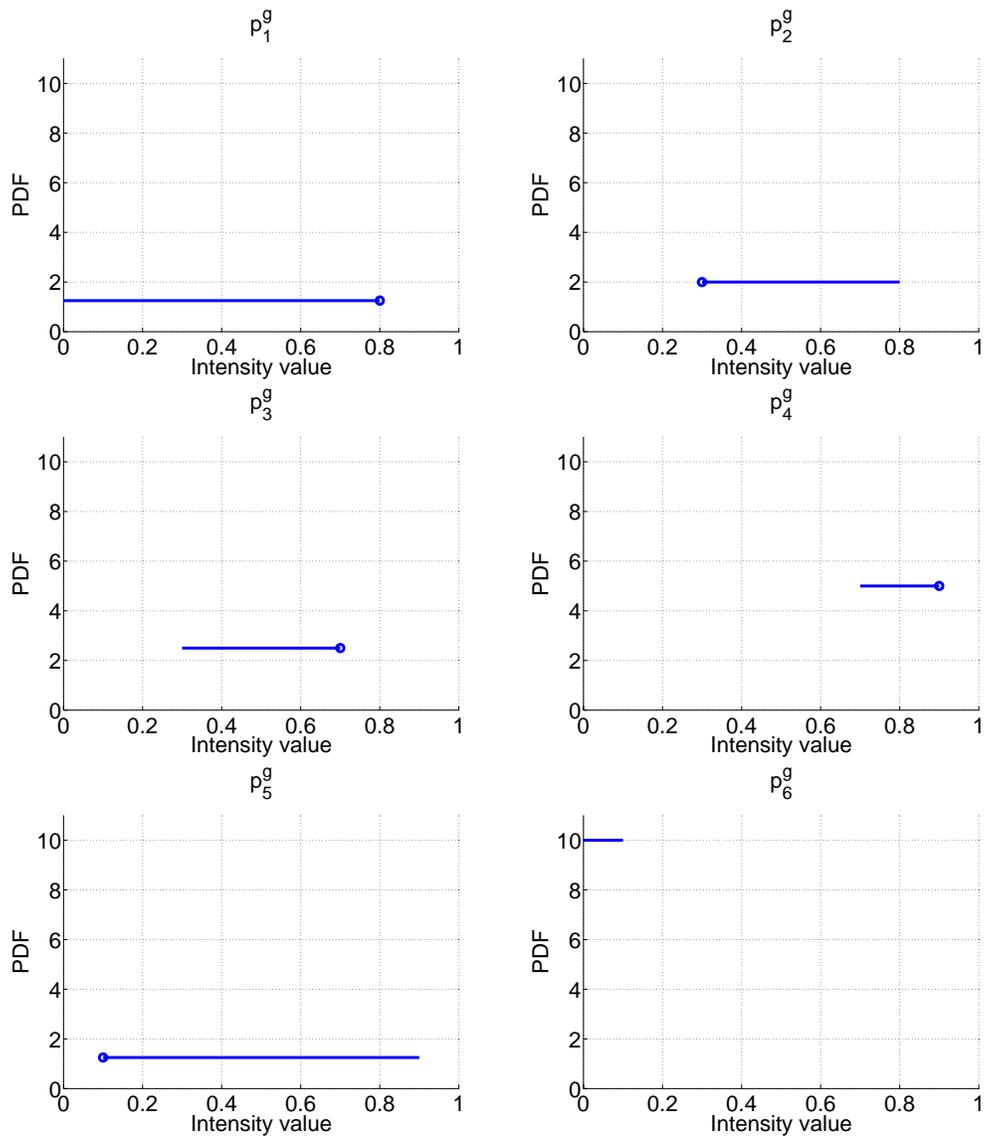


Figure 5.9: The PDF associated with the image in each subinterval.

By (5.5), the PDF p^g associated with the whole image is defined as the normalized summation of those PDFs p_i^g , $i = 1, \dots, 6$, as shown in Figure 5.10.

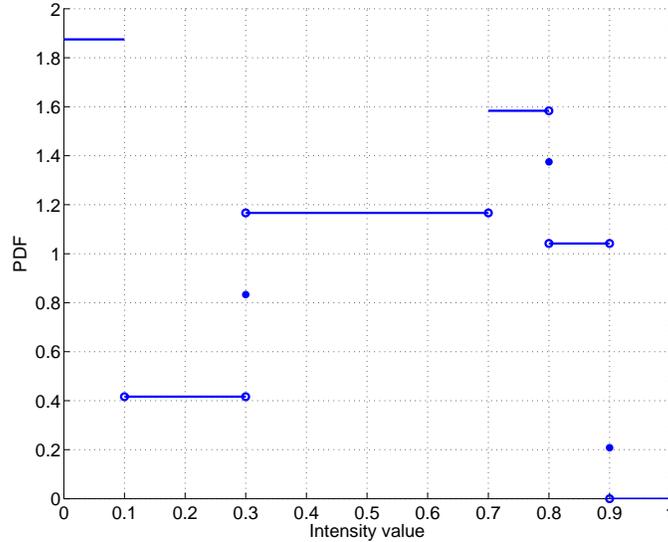


Figure 5.10: The PDF associated with the whole image.

In general, it is difficult to determine the value of p^g at a specific intensity value. Thus, we approximate the PDF as a piecewise constant function by (5.13). Figure 5.11 shows the piecewise constant approximate PDF with only two cells. In this case, $q_1^g \approx 1.008$ and $q_2^g \approx 0.992$. The individual entropy can then easily be computed using (5.14).

As shown in Section 5.2.5, we approximate the joint PDF p^{fg} in each cell $D_{i,j}^{fg}$ defined in the PDF domain as a constant function $q_{i,j}^{fg}$ by (5.18). We can consider $q_{i,j}^{fg}$ as a function with respect to the translation τ . Another example is to demonstrate the smoothness of the function $q_{i,j}^{fg}(\tau)$. We take the T1-weighted MRI as the target image and the CT as the template image, which are shown in Figure 2.1. For simplicity, we assume there are 4×4 cells in the joint PDF domain and the plots of $q_{i,j}^{fg}(\tau)$, $i = 1, \dots, 4$, $j = 1, \dots, 4$, are shown in Figure 5.12.

As can be observed, each $q_{i,j}^{fg}(\tau)$ changes smoothly with respect to translation. It illustrates why the joint entropy \bar{H}^{fg} and hence the mutual information \bar{M} will also be smooth; see Section 6.2.

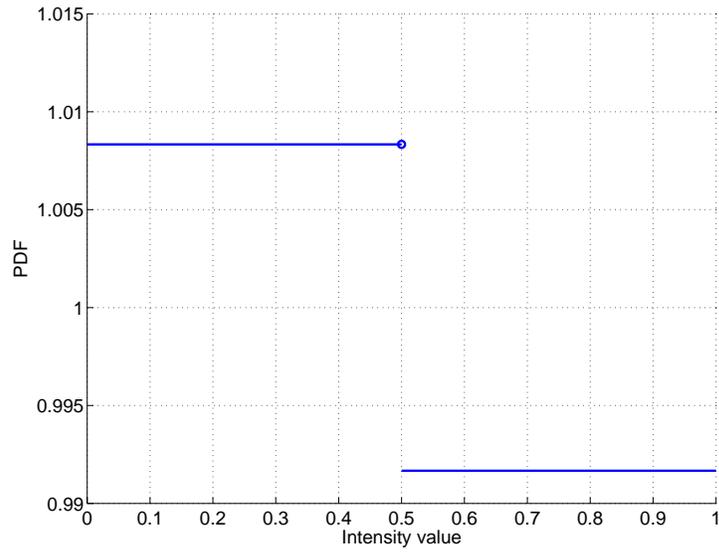


Figure 5.11: The PDF is approximated as a constant function in each cell.

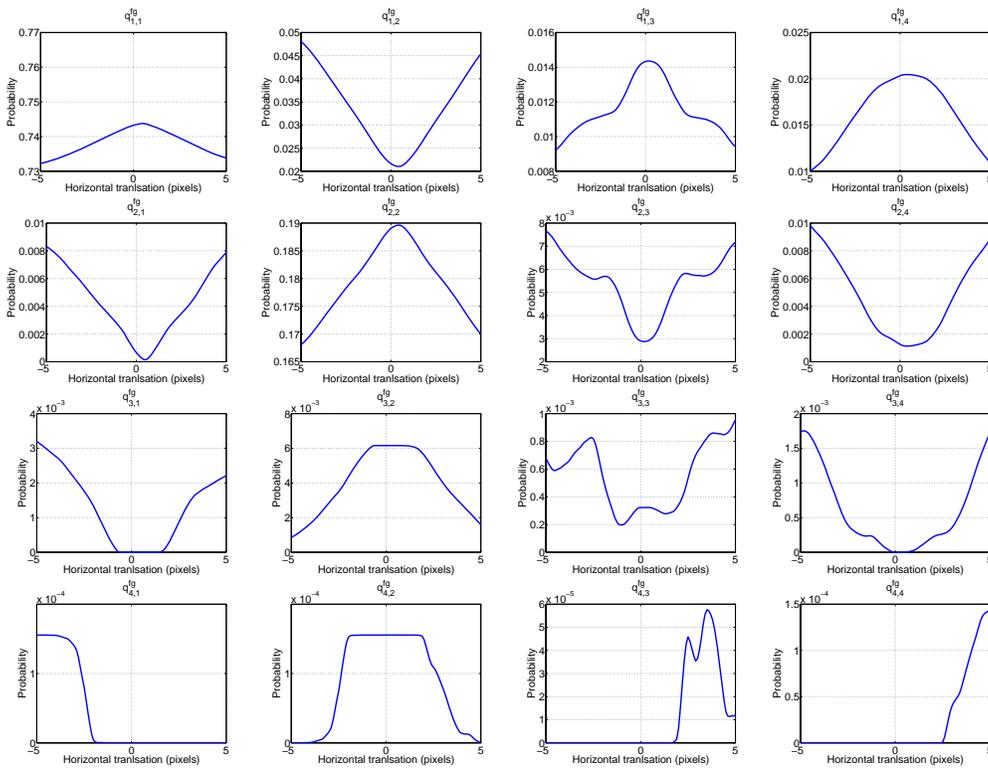


Figure 5.12: The approximate probabilities associated with each bin in the PDF domain with respect to the horizontal translation.

5.3 Two Dimensional Image Registration

Based on the same motivation for 1D images, given the target image I^g and the template image I^f which are both in 2D, the mutual information can also be computed numerically.

Continuous functions can be modeled using the observable pixel intensity values. Let the observable intensity values be $\{I_{i,j}^g\}$ and $\{I_{i,j}^f\}$ for the target image and the template image, respectively, where $i = 0, \dots, N_x - 1$ and $j = 0, \dots, N_y - 1$. $N_x \times N_y$ denotes the resolution of the target image and the template image with the pixel size of $h_x \times h_y$. For consistency with the assumption that images are continuous and periodic in our model, we add ghost points at the boundaries of images. For example, for the target image, extra elements of $\{I_{i,N_y}^g\}$ where $i = 0, \dots, N_x - 1$ and $\{I_{N_x,j}^g\}$ where $j = 0, \dots, N_y$, are added to the 2D image array, satisfying $I_{i,N_y}^g = I_{i,0}^g$ and $I_{N_x,j}^g = I_{0,j}^g$. For the template image, the same strategy can be applied. Interpolation is used for the data points $\{(x_i, y_j), I_{i,j}^g\}$ and $\{(x_i, y_j), I_{i,j}^f\}$ to construct the continuous target and template images, respectively. The continuous model uses bilinear interpolation to construct the continuous functions, though other interpolation strategies are also possible to be used. Formally, $\forall(x, y)$ such that $x \in [x_{i-1}, x_i]$ and $y \in [y_{j-1}, y_j]$, the intensity value at the position (x, y) in the target image is determined by

$$I^g(x, y) = \omega_{1,1}I_{i-1,j-1}^g + \omega_{0,1}I_{i,j-1}^g + \omega_{1,0}I_{i-1,j}^g + \omega_{0,0}I_{i,j}^g,$$

where

$$\begin{aligned}\omega_{1,1} &= \frac{(h_x - \Delta x)(h_y - \Delta y)}{h_x h_y}, \\ \omega_{0,1} &= \frac{\Delta x(h_y - \Delta y)}{h_x h_y}, \\ \omega_{1,0} &= \frac{(h_x - \Delta x)\Delta y}{h_x h_y}, \\ \omega_{0,0} &= \frac{\Delta x \Delta y}{h_x h_y}, \\ \Delta x &= x - x_{i-1}, \\ \Delta y &= y - y_{i-1}.\end{aligned}$$

The template image can be constructed using the same bilinear interpolation strategy.

Unfortunately, unlike the case for 1D images, the analytical PDFs for 2D images

are not easy to compute. Alternatively, we reduce the problem for 2D images to a problem for 1D images.

5.3.1 Translation

In general, the transformation of images in a 2D domain includes the rotation through the axis perpendicular to the image plane as well as the translation in x and y axes. For simplicity, we first discuss the case that only translation is considered.

In this case, the transformation function $\xi(\cdot)$ defined in Section 4.2.2 can be simplified as

$$\xi(x, y) = (x + \tau_x, y + \tau_y),$$

where τ_x denotes the translation horizontally and τ_y denotes the translation vertically. The node points $\{(x_i, y_j)\}$ will be translated to $\{(x_i + \tau_x, y_j + \tau_y)\}$. We have the assumption that images are periodic to avoid the case that images may be translated out of boundaries. $\{(x_i + \tau_x, y_j + \tau_y)\}$ can be simplified as $\{(x_i + \Delta h_x, y_j + \Delta h_y)\}$, where $\Delta h_x = \tau_x \pmod{h_x}$ and $\Delta h_y = \tau_y \pmod{h_y}$.

To easily construct the PDFs, following the similar idea in the 1D case, we divide the images domain into sub-domains and then construct the PDFs for each sub-domain. Naturally, the image domain can be divided using the node points in both the target image and the transformed template image. Formally, let

$$s_i \stackrel{\text{def}}{=} \begin{cases} x_{\frac{i}{2}} & \text{if } i \text{ is even,} \\ x_{\frac{i-1}{2}} + \Delta h_x & \text{if } i \text{ is odd.} \end{cases}$$

$$t_j \stackrel{\text{def}}{=} \begin{cases} y_{\frac{j}{2}} & \text{if } j \text{ is even,} \\ y_{\frac{j-1}{2}} + \Delta h_y & \text{if } j \text{ is odd.} \end{cases}$$

$\{(s_i, t_j)\}$ is the union of the node points in both the target image and the transformed template image. As shown in Figure 5.13, those node points generally divide the whole image domain into sub-domains

$$E_{i,j} \stackrel{\text{def}}{=} \{(x, y) \mid x \in [s_{i-1}, s_i), y \in [t_{j-1}, t_j)\}.$$

In each sub-domain, PDFs are still not easy to compute analytically. Alternatively, we divide each $E_{i,j}$ into several horizontal strips with equal width and

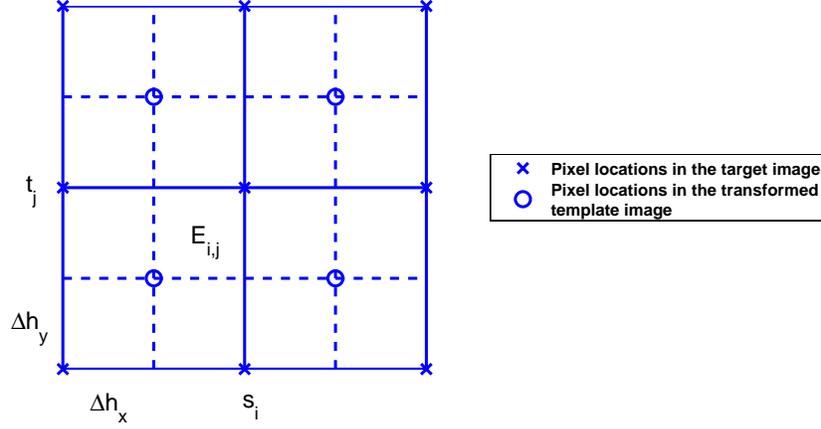


Figure 5.13: The pixels in the target and the transformed template images generally divide the whole image domain into sub-domains.

approximate image functions as linear 1D image function in each strip. Formally, assuming the number of strips are fixed and denoted as N_S , let

$$t_j^k \stackrel{\text{def}}{=} t_{j-1} + h_S k \quad k = 0, \dots, N_S,$$

where $h_S = \frac{t_j - t_{j-1}}{N_S}$. As shown in Figure 5.14, $E_{i,j}$ is divided into strips $S_{i,j}^k$ defined by

$$S_{i,j}^k \stackrel{\text{def}}{=} \{(x, y) \mid x \in [s_{i-1}, s_i], y \in [t_j^{k-1}, t_j^k]\} \quad k = 1, \dots, N_S.$$

Note that I^g and ϕ are linear both horizontally and vertically. However, when h_S is small enough (i.e., N_S is large enough), the image functions in each strip can be approximated as functions which are linear horizontally and constant vertically. For example, the target image in the strip $S_{i,j}^k$ can be approximated by

$$I^g(x, y) \approx \bar{I}_{i,j}^{gk}(x, y) \stackrel{\text{def}}{=} \frac{s_i - x}{s_i - s_{i-1}} I^g\left(s_{i-1}, \frac{t_j^{k-1} + t_j^k}{2}\right) + \frac{x - s_{i-1}}{s_i - s_{i-1}} I^g\left(s_i, \frac{t_j^{k-1} + t_j^k}{2}\right) \quad \forall (x, y) \in S_{i,j}^k. \quad (5.21)$$

The transformed template image in each strip can be approximated by the same strategy.

Let the approximate individual PDF for the target image in $S_{i,j}^k$ be $\bar{p}_{i,j}^{gk}$. Considering (5.21) is constant vertically, it is the same as the PDF for the equivalent linear 1D function, which has been discussed in Section 5.2.2. The approximate PDF in each sub-domain $E_{i,j}$ is computed by combining the approximate PDFs in each strip. Since we assume the strips have equal width, the approximate PDF

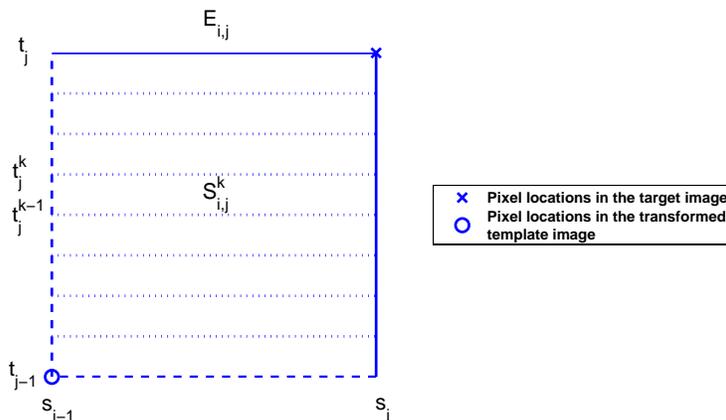


Figure 5.14: Each sub-domain is divided into strips with equal width.

in each strip contributes to the approximate PDF in each sub-domain $E_{i,j}$ equally. Let the symbol $p_{i,j}^g$ be the PDF associated with the sub-domain $E_{i,j}$ and the symbol $\bar{p}_{i,j}^g$ be the approximate PDF associated with the same sub-domain. We have

$$p_{i,j}^g \approx \bar{p}_{i,j}^g \stackrel{\text{def}}{=} \frac{1}{N_S} \sum_{k=1}^{N_S} \bar{p}_{i,j}^{gk}.$$

Finally, let p^g be the PDF for the target image I^g . It can be approximated by combining the approximate PDFs in each sub-domain with the weight proportional to its own area; i.e.,

$$p^g \approx \bar{p}^g \stackrel{\text{def}}{=} \frac{(s_i - s_{i-1})(t_j - t_{j-1})}{N_x h_x N_y h_y} \sum_{i=1}^{2N_x} \sum_{j=1}^{2N_y} \bar{p}_{i,j}^g.$$

The PDF for the transformed template image and the joint PDF can be approximated using similar strategies. Since we finally translate the 2D image problems into 1D image problems, entropies and mutual information can be computed as the same ways described in the 1D case.

Note when only horizontal translation is considered in the transformation, it is exactly the same with the model applied for 1D images and the individual entropies are independent of the translation. However, when vertical translation is also considered, the individual entropies for the target image and the template image are not independent of the translation anymore because the positions of the strips change with the translation. However, the errors are so minimal that they will not qualitatively affect the behaviour of mutual information; see Section 6.2.

5.3.2 Rotation

When rotation is also considered in the transformation parameters, to analyze PDFs is quite difficult. However, since rotation normally does not make the final mutual information non-smooth in the interpolation-based models [11], we separate the transformation into rotation and translation. When rotation is first performed, we use interpolation to evaluate the pixel intensity values of the transformed template image, which is the same as the case in the interpolation-based models. Then, we only consider translation in the transformation parameters and apply the continuous model to compute the mutual information. The experiments shown in Chapter 6 verify this point.

5.3.3 Complexity Analysis

In the 2D case, the computation of the mutual information consists two steps. When rotation is first considered in the transformation parameters, the complexity for interpolating the transformed template image is $O(N_x N_y)$.

Afterwards, each sub-domain is divided into strips and computing the PDFs corresponding to each strip is treated as a 1D image problem. Since both the target image and the transformed template image are linear functions in each strip, according to the analysis of the complexity in the 1D case, the complexity for computing the PDFs numerically corresponding to each strip is $O(\tilde{N})$ where \tilde{N} is defined in Section 5.2.6 and it is normally 2 or 3. Combining the PDFs in each strip in each sub-domain, we can compute the PDFs numerically with the complexity $O(\tilde{N} N_S N_x N_y)$. Given the knowledge of the approximate PDFs, entropies and finally the mutual information are computed with the complexity $O(N_D^g N_D^f)$. The total complexity in this step is $O(\tilde{N} N_S N_x N_y + N_D^g N_D^f)$. Considering that $N_x N_y \gg N_D^g N_D^f$ in general, the complexity is $O(\tilde{N} N_S N_x N_y)$.

Combining the two steps, the final complexity is $O(\tilde{N} N_S N_x N_y)$. The complexity of the histogramming model and the partial volume model are both $O(N_x N_y + N_B^g N_B^f)$. Considering that N_S is not that large ($N_S = 32$ for most of our experiments) and $N_x N_y \sim N_B^g N_B^f$, the complexity of our new model is comparable to the other two.

5.4 Relationship with Partial Volume Models

Assuming nearest-neighbour interpolation is exploited to construct the continuous image functions, the partial volume model can be interpreted as a variant of our new model when only translation is considered in the transformation parameter set.

Following the same idea in the continuous model, the nearest neighbour interpolation can be interpreted as a piecewise constant image function defined in a continuous domain. Formally, for each pixel location (x_i, y_j) , let the box domain

$$F_{i,j} \stackrel{\text{def}}{=} \left\{ (x, y) \mid x \in \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right), y \in \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}} \right) \right\},$$

where

$$\begin{aligned} x_{i-\frac{1}{2}} &\stackrel{\text{def}}{=} \frac{1}{2} (x_{i-1} + x_i), \\ x_{i+\frac{1}{2}} &\stackrel{\text{def}}{=} \frac{1}{2} (x_i + x_{i+1}), \\ y_{j-\frac{1}{2}} &\stackrel{\text{def}}{=} \frac{1}{2} (y_{j-1} + y_j), \\ y_{j+\frac{1}{2}} &\stackrel{\text{def}}{=} \frac{1}{2} (y_j + y_{j+1}). \end{aligned}$$

Image intensity values in the domain $F_{i,j}$ are assumed to be the same as the one at the pixel location (x_i, y_j) . That can be interpreted as a constant function over the box domain $F_{i,j}$.

Regarding the individual PDFs, since both the target and the transformed template images are constant functions in each of their own box domains, according to the way constructing PDFs in our new model, the constant function in each box domain corresponds to a Dirac delta function in the PDF, and finally contributes to a cell with the same weight as the constant functions in other box domains. This is equivalent to the case in the partial volume model: each pixel contributes to some bin with equal weight in order to construct the individual probability distribution.

Regarding the joint PDF, considering four adjacent box domains composed of $F_{i-1,j-1}$, $F_{i,j-1}$, $F_{i-1,j}$, and $F_{i,j}$, if only translation is considered in the transformation parameters, the union of the four domains generally covers a box domain of the transformed template image (Figure 5.15), which can be denoted by

$$F' \stackrel{\text{def}}{=} \left\{ (x, y) \mid \exists k, l, \text{ s.t. } x \in \left[x_{k-\frac{1}{2}} + \tau_x, x_{k+\frac{1}{2}} + \tau_x \right), y \in \left[y_{l-\frac{1}{2}} + \tau_y, y_{l+\frac{1}{2}} + \tau_y \right) \right\},$$

where τ_x and τ_y represent the horizontal and vertical translation, respectively.

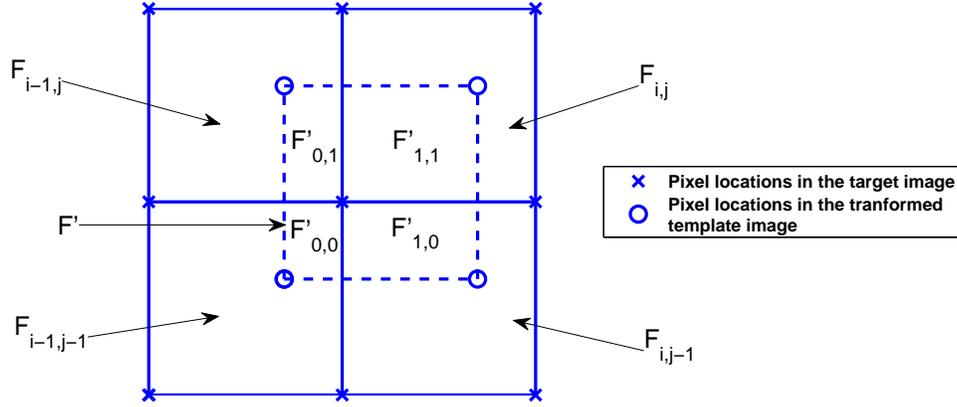


Figure 5.15: A box domain of the transformed template image is generally located among four adjacent box domains of the target image.

Note that the target image intensity value in the domain $F_{i-1,j-1}$, $F_{i,j-1}$, $F_{i-1,j}$, and $F_{i,j}$ are constants. When the template image are translated, its intensity value is also constant in the domain F' . To construct the joint PDF associated with the domain of F' , we can analyze it in each of the sub-domains where both of the target image and the transformed template image are constants, denoted by

$$\begin{aligned}
 F'_{0,0} &\stackrel{\text{def}}{=} F' \cap F_{i-1,j-1}, \\
 F'_{1,0} &\stackrel{\text{def}}{=} F' \cap F_{i,j-1}, \\
 F'_{0,1} &\stackrel{\text{def}}{=} F' \cap F_{i-1,j}, \\
 F'_{1,1} &\stackrel{\text{def}}{=} F' \cap F_{i,j}.
 \end{aligned}$$

They are also shown in Figure 5.15.

In each of the sub-domains, since both of the target and transformed template images are constants, the joint PDF associated with each sub-domain (i.e., $F'_{0,0}$, $F'_{1,0}$, $F'_{0,1}$, and $F'_{1,1}$) is a Dirac delta function. According to the continuous model, it will finally contribute to some cell in the PDF domain with the weight proportional to its area. It is equivalent to assigning the weight of the pixel $(x_k + \tau_x, y_l + \tau_y)$ to the bins associated with the pixel intensity value pairs of itself and its neighbours with the same manner in the partial volume model. Similar analysis can also be performed for other box domains in the transformed template image.

After the PDFs have been computed, by comparing (4.1) and (5.19), it can be noticed that the entropies and finally the mutual information in the continuous model are computed similarly to the ones in the partial volume model.

Chapter 6

Numerical Results for Continuous Models

In this chapter, numerical experiments are performed to compare the continuous model and the other models. For different clinical images, smooth mutual information functions can be observed based on the new model. Besides, by applying standard optimization methods, including a trust region method and the Nelder-Mead method, to the model, both efficiency and accuracy of the registration problem are validated. At the same time, the robustness of the model is also demonstrated.

6.1 Images

The images for the experiments are a set of the T1-weighted MRI, T2-weighted MRI, PD-weighted MRI, and CT shown in Figure 2.1. Those images are extracted from the same patient. They are aligned using the gold-standard transformations and further re-sampled. All of the images are encoded in 8 bits. The resolution is 256×256 for all the images.

6.2 Smoothness

Taking the T1-weighted MRI as the target image and the CT as the template image, Figure 6.1 and Figure 6.2 show the individual entropies, the joint entropy, and the mutual information with respect to the horizontal and vertical translation using the continuous model, respectively. The cell numbers for estimating PDFs are 8

for both the target and the transformed template images. The number of strips is chosen as 32 for each sub-domain. We refer them as the default parameters in the following experiments.

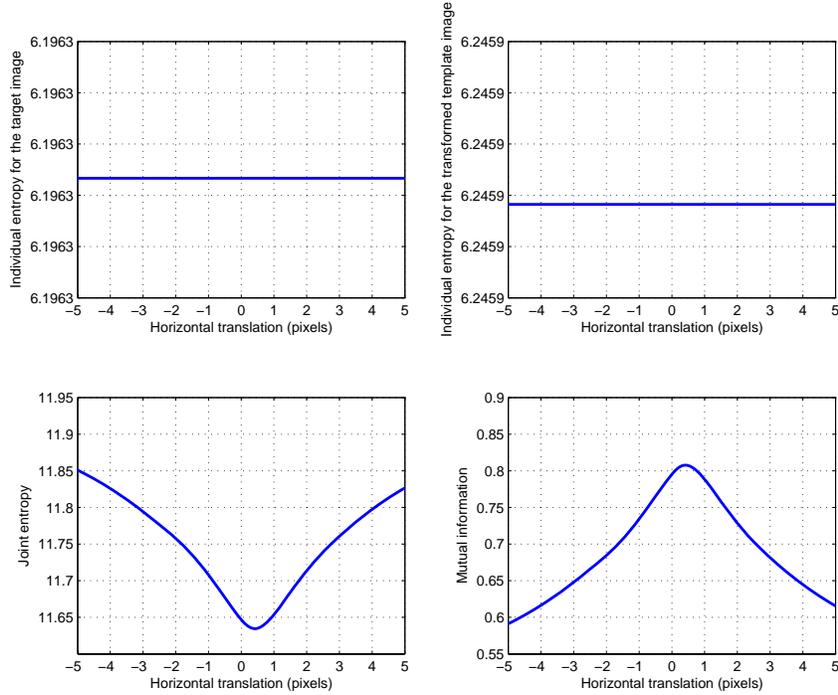


Figure 6.1: The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) with respect to the horizontal translation.

Compared with the mutual information functions computed using the histogramming model and the partial volume model shown in Figure 4.2 and Figure 4.6, the mutual information functions computed using our continuous model (Figure 6.1 and Figure 6.2) are quite smooth and no artifact is observed.

Comparing Figure 6.1 and Figure 6.2, it is noticed that individual entropies are both constant functions with respect to the horizontal translation but it is not the case in the individual entropies with respect to the vertical translation. That is because, according to the analysis in Section 5.3.1, when only horizontal translation is considered in the transformation parameter set, the 2D image problem is equivalent to the 1D image problem, and the individual entropies should be independent of the transformation. In comparison, entropies depend on the vertical translation as shown in Figure 6.2. However, the magnitude of the fluctuation in the

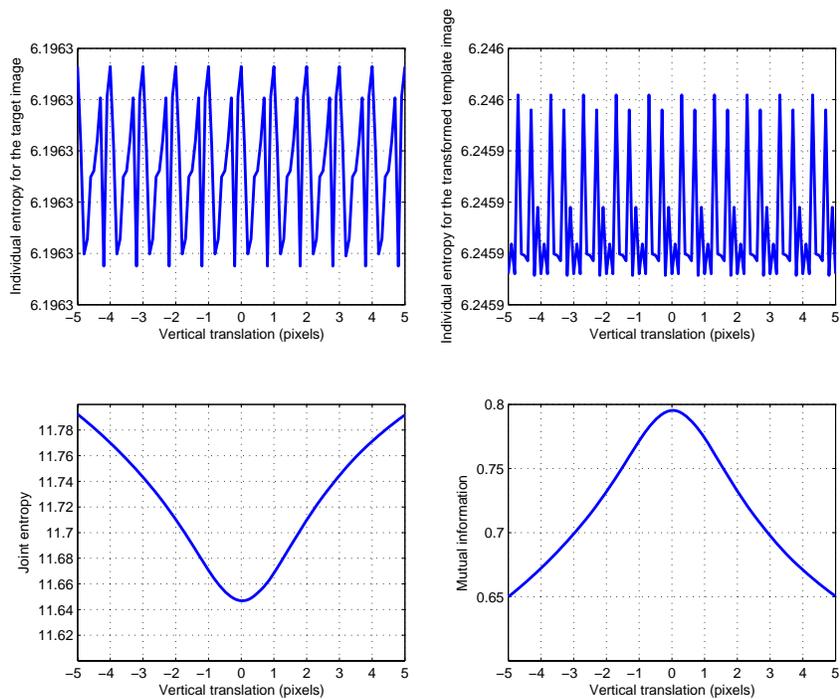


Figure 6.2: The individual entropy for the target image (upper left), the individual entropy for the transformed template image (upper right), the joint entropy (lower left), and the mutual information (lower right) with respect to the vertical translation.

entropies is so small (i.e., 10^{-5}) that it does not qualitatively affect the smoothness of the mutual information. It can be observed that the mutual information function in Figure 6.2 is visibly as smooth as the one in Figure 6.1.

6.2.1 Image Modalities

Based on our observation, when the image modalities of the two images for registration are close (e.g., one is a T1-weighted MRI and the other is a T2-weighted MRI), the mutual information computed using the histogramming model or the partial volume model is smoother than the cases that the modalities of the two images differ a lot (e.g., one image is an MRI and other one is a CT).

Figure 6.3 and Figure 6.4 show the mutual information functions with respect to the horizontal translation using the histogramming model and the partial volume model by taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image. The bin numbers are 128 and 128 for the target and the template images respectively, which are the same as the ones used in Figure 4.2 and Figure 4.6. By comparing Figure 6.3, Figure 6.4, Figure 4.2, and Figure 4.6, it can be observed that, when the image modalities are close (e.g., the T1-weighted and T2-weighted MRI pair), the mutual information can be smoother than the cases that the modalities of the two images differ a lot (e.g., the T1-weighted MRI and CT pair).

In comparison, the mutual information functions computed using our continuous model are consistently smooth. Figure 6.5 plots the mutual information with respect to the horizontal translation with the default parameters except taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image. Note that both Figure 6.5 and Figure 6.1 present smooth mutual information functions, which demonstrates the robustness of our model.

6.2.2 Image Resolution

We also notice that the image resolution affects the smoothness of mutual information functions using the histogramming model. By downsampling the images using nearest-neighbour interpolation from 256×256 to 64×64 , Figure 6.6 shows the mutual information functions with respect to the horizontal translation using the histogramming model. It can be observed that the mutual information becomes less smoother when the resolution decreases. That is because less image intensity

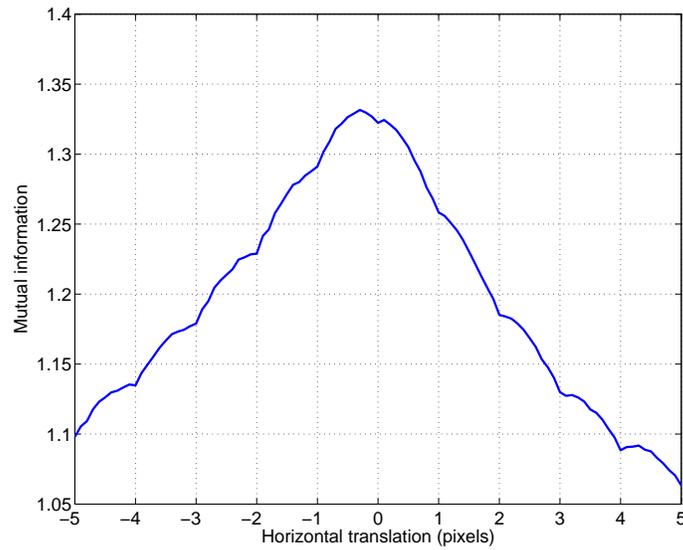


Figure 6.3: The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the histogramming model.

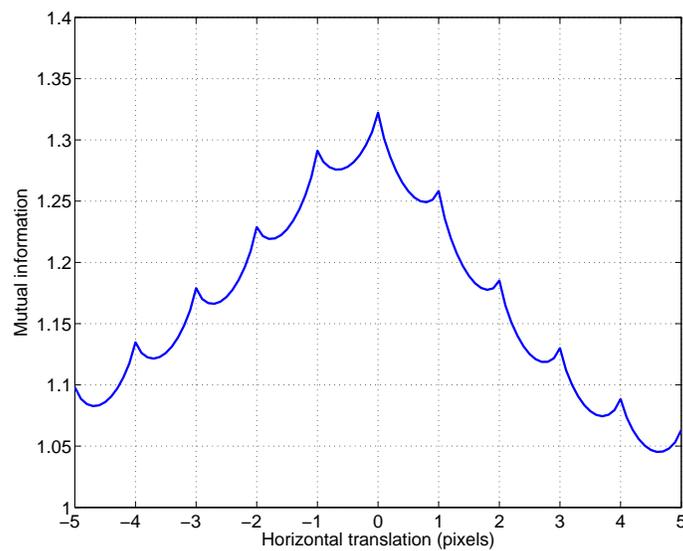


Figure 6.4: The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the partial volume model.

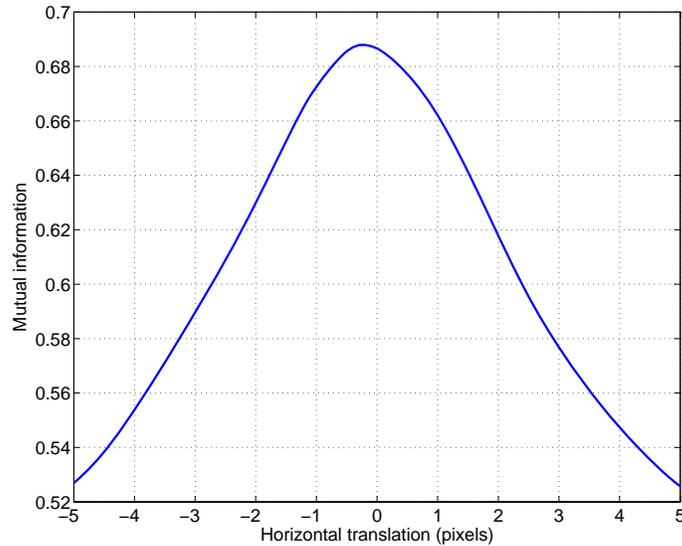


Figure 6.5: The mutual information taking the T1-weighted MRI as the target image and the T2-weighted MRI as the template image using the continuous model.

values fail representing the statistical behaviour of the probability distributions. In comparison, the continuous model is more robust: it is smooth no matter the resolution decreases or not. Figure 6.7 demonstrates that the mutual information is always smooth even if the image resolution decreases from 256×256 to 64×64 .

This is very useful when multi-resolution schemes [26] are used in order to efficiently solve the image registration problem. In multi-resolution schemes, images for registration are first downsampled as images in lower resolution. Registration is first performed using those images in lower resolution. After computing a solution close to the real optimal solution, we increase the resolution of images and implement another image registration process with the initial guess as the final solution in the last scenario which is referred as the first level. The procedure is performed iteratively until the resolution increases to the original resolution of images. Since image resolution is relatively lower in the first few levels, the registration process can be accelerated in some degree.

In practice, when images for registration are in lower resolution, the non-smoothness that can be observed in Figure 6.6 hamper the performance of the multi-resolution strategy because the artifacts prevent the numerical optimization methods from converging to the optimal solution. In comparison, our model turns out to be quite robust: the mutual information functions maintain smoothness even though images are in lower resolution (Figure 6.7). This facilitates the optimiza-

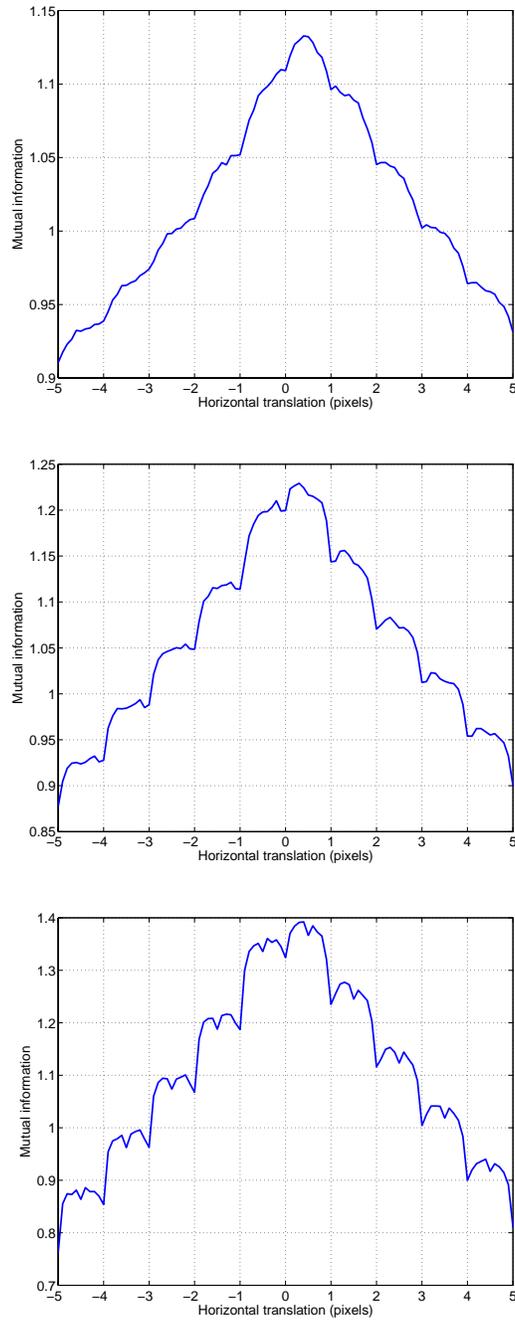


Figure 6.6: The mutual information taking the images in the resolution of 256×256 (upper), 128×128 (middle), and 64×64 (lower) using the histogramming model.

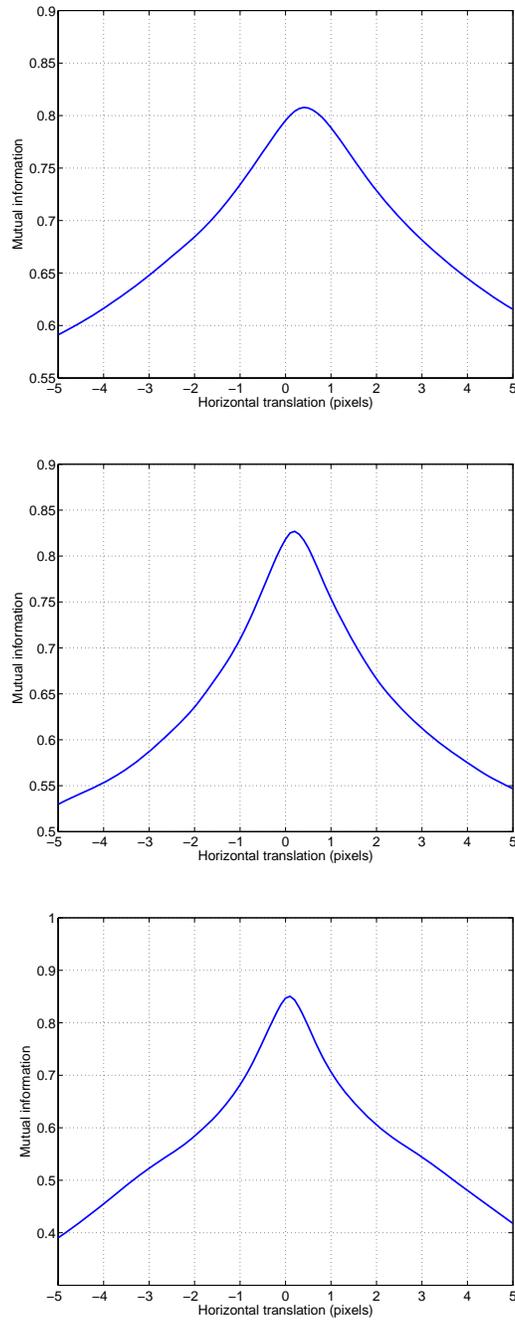


Figure 6.7: The mutual information taking the images in the resolution of 256×256 (upper), 128×128 (middle), and 64×64 (lower) using the continuous model.

tion methods to converge to a better initial guess for the next level and finally accelerates the whole registration process.

6.3 Sub-pixel Accuracy

The objective of this section is to test the sub-pixel accuracy when different models are used for optimization methods. We manually shift the template image by 0.8 pixels horizontally and re-sample the shifted template image using bilinear interpolation. Figure 6.8 shows the mutual information plots using different models. The numbers of bins are 128 for both the target and the transformed template images in the histogramming and partial volume models; the numbers of cells are 8 for both the target and transformed template images and the number of strips is 32 for each sub-domain in the continuous model.

In Figure 6.8, the solid lines indicate the mutual information functions taking the original T1-weighted MRI as the target image and the CT as the template image; the dash lines indicate the mutual information functions taking the original T1-weighted MRI as the target image and the shifted CT as the template image. We can observe that both the histogramming model and the continuous model can detect the sub-pixel accuracy, showing that the global maximum is also shifted by 0.8 pixel to the left. The partial volume model fails doing that since the global maximum always happens at integer pixel translation according to the analysis in Section 4.3. Thus, optimization methods can never detect the sub-pixel accuracy in this case.

6.4 Optimization Performance

To illustrate the benefit of the smoothness of the mutual information using the continuous model, two different optimization methods mentioned in Section 4.4, a trust region method for non-linear optimization problems and the Nelder-Mead method, are applied on the mutual information functions computed using different models.

In the trust region method, the gradient and the Hessian matrix information of the mutual information function are required for efficient convergence. Central difference method is exploited to approximate them. More specifically, let the

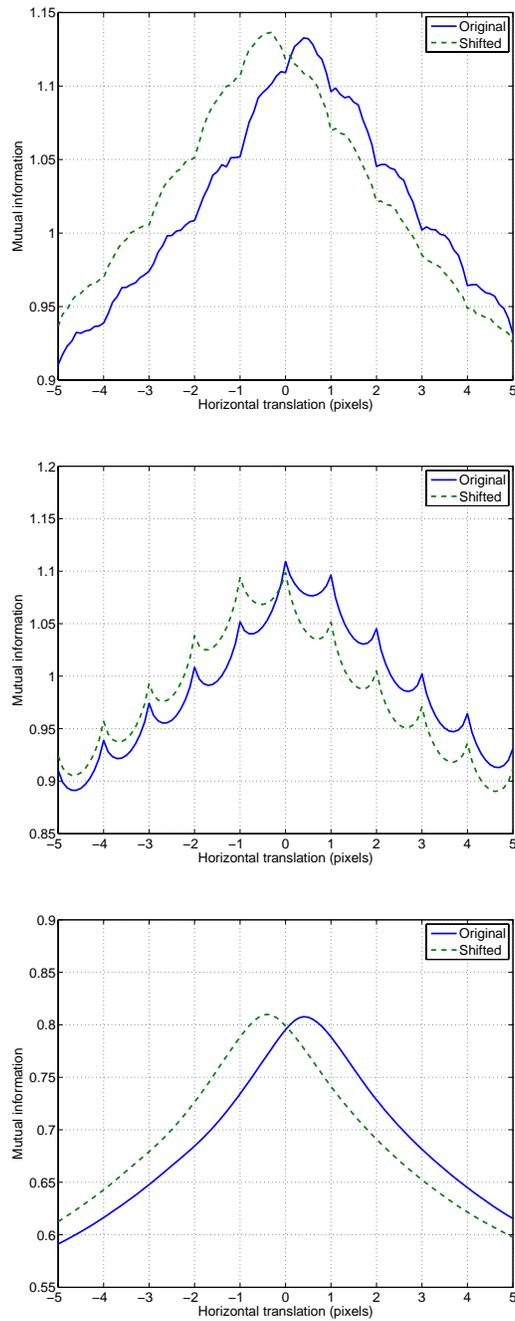


Figure 6.8: The mutual information with respect to the horizontal translation using the histogramming model (upper), the partial volume model (middle), and the continuous model (lower).

transformation parameter set be

$$\mathbf{s} = (\theta, \tau_x, \tau_y),$$

where θ denotes the rotation, τ_x denotes the horizontal translation, and τ_y denotes the vertical translation. In order to compute the gradient of the mutual information function, the first-order partial derivatives are then approximated by the central difference method. For example,

$$\frac{\partial M(\mathbf{s})}{\partial \theta} \approx \frac{M(\mathbf{s} + \Delta \mathbf{s}_\theta) - M(\mathbf{s} - \Delta \mathbf{s}_\theta)}{2\Delta \theta},$$

where $\Delta \mathbf{s}_\theta = (\Delta \theta, 0, 0)$. $M(\mathbf{s} + \Delta \mathbf{s}_\theta)$ and $M(\mathbf{s} - \Delta \mathbf{s}_\theta)$ are obtained by changing the rotation from the angle θ to $\theta + \Delta \theta$ and $\theta - \Delta \theta$, respectively, but keeping the other parameters fixed. Other derivatives can be approximated in similar ways by choosing appropriate $\Delta \tau_x$ and $\Delta \tau_y$ as the steps in the horizontal and vertical translation. After $\frac{\partial M(\mathbf{s})}{\partial \theta}$, $\frac{\partial M(\mathbf{s})}{\partial \tau_x}$, and $\frac{\partial M(\mathbf{s})}{\partial \tau_y}$ are approximated, the gradient of the mutual information function can then be formed as

$$\nabla M(\mathbf{s}) = \left(\frac{\partial M(\mathbf{s})}{\partial \theta}, \frac{\partial M(\mathbf{s})}{\partial \tau_x}, \frac{\partial M(\mathbf{s})}{\partial \tau_y} \right)^T.$$

In order to compute the Hessian matrix, the second-order partial derivatives of the mutual information function can be approximated using the central difference method twice. For example,

$$\begin{aligned} \frac{\partial^2 M(\mathbf{s})}{\partial \theta \partial \tau_x} &\approx \frac{\partial}{\partial \theta} \left(\frac{M(\mathbf{s} + \Delta \mathbf{s}_{\tau_x}) - M(\mathbf{s} - \Delta \mathbf{s}_{\tau_x})}{2\Delta \tau_x} \right) \\ &\approx \frac{(M_{1,1} - M_{0,1}) - (M_{1,0} - M_{0,0})}{4\Delta \theta \Delta \tau_x}, \end{aligned}$$

where

$$\begin{aligned} M_{1,1} &= M(\mathbf{s} + \Delta \mathbf{s}_\theta + \Delta \mathbf{s}_{\tau_x}), \\ M_{0,1} &= M(\mathbf{s} - \Delta \mathbf{s}_\theta + \Delta \mathbf{s}_{\tau_x}), \\ M_{1,0} &= M(\mathbf{s} + \Delta \mathbf{s}_\theta - \Delta \mathbf{s}_{\tau_x}), \\ M_{0,0} &= M(\mathbf{s} - \Delta \mathbf{s}_\theta - \Delta \mathbf{s}_{\tau_x}), \\ \Delta \mathbf{s}_{\tau_x} &= (0, \Delta \tau_x, 0). \end{aligned}$$

After all the second-order partial derivatives of the mutual information function are approximated, the Hessian matrix can then be formed as

$$H(M) = \begin{bmatrix} \frac{\partial^2 M}{\partial \theta^2} & \frac{\partial^2 M}{\partial \theta \partial \tau_x} & \frac{\partial^2 M}{\partial \theta \partial \tau_y} \\ \frac{\partial^2 M}{\partial \tau_x \partial \theta} & \frac{\partial^2 M}{\partial \tau_x^2} & \frac{\partial^2 M}{\partial \tau_x \partial \tau_y} \\ \frac{\partial^2 M}{\partial \tau_y \partial \theta} & \frac{\partial^2 M}{\partial \tau_y \partial \tau_x} & \frac{\partial^2 M}{\partial \tau_y^2} \end{bmatrix}.$$

In our experiments, we choose $\Delta\theta = 0.1$ radian and $\Delta\tau_x = \Delta\tau_y = 0.1$ pixel as the default parameters.

Since the interpolation artifacts mainly happen when spatial translation is considered in the transformation parameter set, to verify the benefit of the smoothness of the mutual information functions computed using the continuous model and compare them with the ones computed by other models, we first assume the transformation parameters only include pure translation.

Table 6.1 shows the performance of the trust region method on different models. The mutual information is computed using the T1-weighted MRI as the target image and the CT as the template image. The images are sampled from the original images using nearest-neighbour interpolation with the resolution as 128×128 . Different initial guesses are chosen in order to test the performance of the method. The stopping criteria of the method are chosen as 10^{-4} .

In the table, the first column denotes different initial guesses. The first element in each vector represents the translation in the horizontal axis; the second one represents the translation in vertical axis. Both of them are in the units of pixels. Different models are employed to evaluate the mutual information, including the histogramming model (HM), the partial volume model (PV), and our continuous model (CM). The number of iterations (IT), the gradient count (GC), and the computed solution are also shown in the table as the performance of the optimization methods. Last column indicates whether the trust region method converges to the optimal solutions successfully. Note that the optimal solutions are approximately $(0, 0)$.

For the histogramming model, since the objective functions are not smooth, the trust region method easily gets stuck at wrong computed solutions due to non-smoothness. In all of our experiments, it is noticed that solutions generally stop at some points around the initial guesses. For the partial volume model, it is known that the mutual information is smooth except for the interpolation artifacts. The method easily converges to one of those artifacts. In Experiment 1, since the initial guess is a local maximum, the computed solution does not move. In the other two, it can be observed that the solutions converge to interpolation artifacts nearest to the initial guesses. In comparison, since the objective functions are globally smooth in our continuous model, the trust region method always quickly converges to the optimal solutions $(0, 0)$ based on our experiments.

Table 6.2 shows the performance of the Nelder-Mead method on different models. The images used for registration are exactly the same with the ones in the

<i>Experiment 1</i>					
Initial guess	Model	IT	GC	Computed solution	Success?
(10, 12)	HM	10	7	(9.6257, 11.3601)	NO
	PV	3	1	(10, 12)	NO
	CM	17	15	(0.1897, -0.0222)	YES
<i>Experiment 2</i>					
Initial guess	Model	IT	GC	Computed solution	Success?
(8.3, -10.2)	HM	3	2	(8.2994, -10.2010)	NO
	PV	7	5	(7.9931, -9.9798)	NO
	CM	18	15	(0.1882, -0.0212)	YES
<i>Experiment 3</i>					
Initial guess	Model	IT	GC	Computed solution	Success?
(-7.3, -5.8)	HM	5	2	(-7.3033, -5.7567)	NO
	PV	5	4	(-6.9915, -5.9885)	NO
	CM	18	16	(0.1897, -0.0222)	YES

Table 6.1: The trust region performance for different models with different initial guesses (the optimal solutions = (0, 0)).

<i>Experiment 1</i>					
Initial guess	Model	IT	FC	Computed solution	Success?
(10, 12)	HM	44	95	(10.3842, 0.2621)	NO
	PV	32	61	(10, 12)	NO
	CM	50	95	(0.1882, -0.0185)	YES
<i>Experiment 2</i>					
Initial guess	Model	IT	FC	Computed solution	Success?
(8.3, -10.2)	HM	49	103	(0.2607, 0.2245)	YES
	PV	48	92	$(-0.1473, 0.2259) \times 10^{-4}$	YES
	CM	49	92	(0.1881, -0.0185)	YES
<i>Experiment 3</i>					
Initial guess	Model	IT	FC	Computed solution	Success?
(-7.3, -5.8)	HM	32	79	(0.6719, -6.5746)	NO
	PV	35	66	(-7.0000, -6.0000)	NO
	CM	56	105	(0.1882, -0.0185)	YES

Table 6.2: The Nelder-Mead performance for different models with different initial guesses (the optimal solutions = (0, 0)).

former experiment. The stopping criteria of the optimization method are chosen as 10^{-4} . The description of the table is almost the same as Table 6.1 except substituting the column indicating the gradient count (GC) for the function count (FC). Compared with the trust region method, the Nelder-Mead method has more capability to skip the artifacts. For the histogramming model, based on our experiments, the Nelder-Mead method may converge to the optimal solutions. However, we still notice that it fails doing that sometimes (e.g., Experiment 1 and Experiment 3). For the partial volume model, similarly to the case in the histogramming model, some failed cases can be observed (e.g., Experiment 1 and Experiment 3). In contrast, for the continuous model, the globally smoothness of the mutual information facilitate the method to converge to the global maximum. Based on our experiments, the Nelder-Mead method always converges to the optimal solutions efficiently.

Table 6.3 shows the performance of different optimization methods when ro-

<i>Experiment 1</i>				
Optimization method	IT	GC	Computed solution	Success?
Trust Region	13	11	(0.0139, 0.4007, -0.0524)	YES
<i>Experiment 2</i>				
Optimization method	IT	FC	Optimal solution	Success?
Nelder-Mead	117	207	(0.0110, 0.3312, -0.0479)	YES

Table 6.3: Optimization performance for the continuous model (the optimal solutions = $(0, 0, 0)$).

tation is also included in the transformation parameter sets using our continuous model. The image data and stopping criteria are chosen as the same as previous experiments. The initial guesses of the positions of the transformed template image are chosen as 5° for the rotation and 5 pixels for both the horizontal and vertical translation. In the column indicating the computed solution, the first element presents the rotation in the unit of radians, the second one indicates the translation in the horizontal axis, and the last one indicates the translation in the vertical axis. Both of the horizontal translation and vertical translation are in the units of pixels. From the table, it can be observed that methods converge to the optimal solutions $(0, 0, 0)$ efficiently.

6.5 Numbers of Strips

When the target image and the template image are both in 2D, we convert the 2D image registration problem into 1D image registration problems. Regarding the vertical translation, it will involve some computational errors in PDFs and finally in entropies and mutual information. The errors depend on the number of strips in each sub-domain. In theory, the more strips we use, the more accurate the computed mutual information will be, and vice versa. To analyze the errors, taking the images used in Section 6.4 for registration, we sample the mutual information with respect to horizontal translation at some fixed points. Then, we flip both of the images over along their diagonals and sample the mutual information with respect to vertical translation at the same fixed points. The former are the analytical solutions of the mutual information and the latter represent the numerical solutions.

Number of strips	Mean of the magnitude of the errors
8	7.2929×10^{-5}
16	3.3264×10^{-5}
32	2.1528×10^{-5}

Table 6.4: Error analysis.

Table 6.4 shows the mean of the magnitude of the differences between them. It can be obviously noticed that the errors decrease while the number of strips increases. When the number of strips in each sub-domain goes to infinity, the mutual information will be exactly the analytical solution for the 2D image problem. However, too many strips will make the image registration more computationally intensive. In practice, we need to seek a balance between the accuracy and efficiency. Based on our experience, making the number of strips as 32 is large enough to facilitate the fast convergence in optimization methods.

6.6 Numbers of Cells

In theory, since we use numerical schemes to compute the mutual information, the more cells we use for computing PDFs, the more the numerical mutual information is close to the analytical solution. On the other hand, by the complexity analysis in Section 5.3.3, more cells will make the computation quite intensive. Figure 6.9 shows the mutual information plots for different numbers of cells for estimating PDFs by fixing the number of strips as 32 in each sub-domain in image functions. It can be noticed that the mutual information is still smooth even if the number of cells decreases to 4×4 . Considering the smoothness and accuracy are not sensitive to the number of cells based on our experience, setting the number of cells as 8×8 is enough for us to quickly find accurate solutions in smooth mutual information functions.

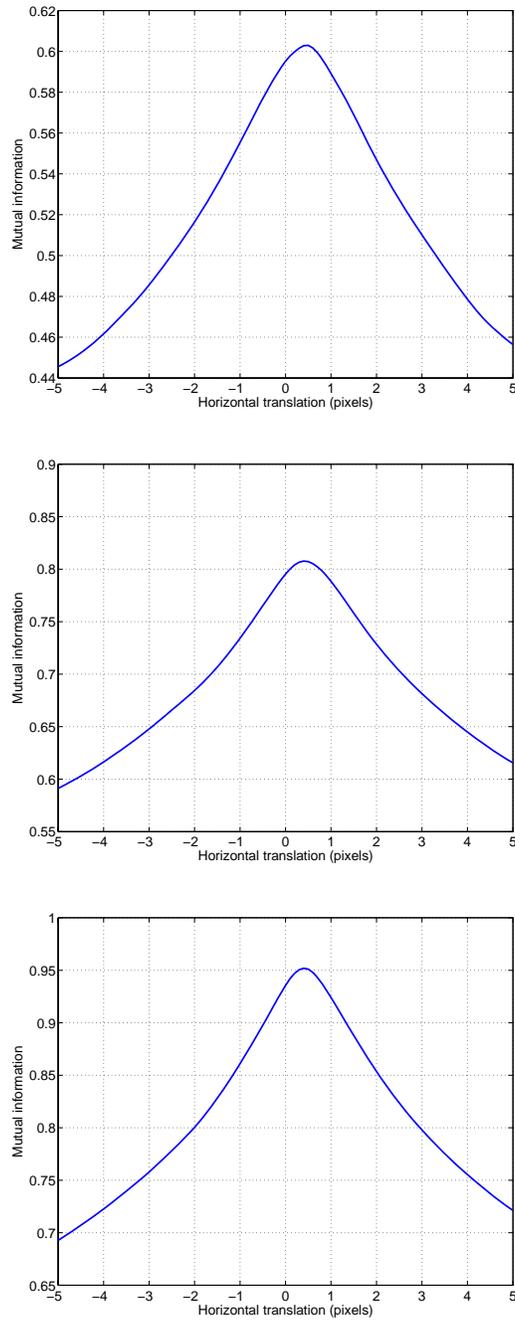


Figure 6.9: The mutual information computed using 4×4 cells (upper), 8×8 cells (middle), and 16×16 cells (lower).

Chapter 7

Conclusions

This thesis presents an efficient parallel 2D-3D mono-modality rigid image registration method which is amenable for GPU processing. Also, it provides a new robust model for computing smooth mutual information functions in order to facilitate the accuracy and efficiency in the multi-modality image registration processes.

Regarding the 2D-3D mono-modality image registration, we have implemented an algorithm using the RapidMind Multi-core Development Platform to exploit the intensive parallelism of GPUs. Numerical experiments using both artificial and clinical image data show that our method is much faster than regular CPU processing. For the clinical image dataset, it only takes around 3 seconds for performing the 2D-3D image registration and in turn accelerates the image registration process by about 100 times.

Regarding the multi-modality image registration, a new model is devised in order to smooth the mutual information functions. We have analyzed the underlying reason of the occurrence of artifacts. Different from others, our model directly removes the discrepancy between the image representation and the mutual information formula and finally makes the mutual information quite smooth. Moreover, using our new model, mutual information can be computed with a comparable complexity with other models. Numerical experiments show that standard optimization methods can converge to the optimal solutions much more accurately and efficiently after the new model is exploited. At the same time, the robustness of the new model has been validated.

Future work can be made in the following aspects. Firstly, besides rigid image registration, RapidMind is possible to be applied in the non-rigid image registration problems for developing efficient parallel algorithms. Secondly, implement-

ing our new model in parallel on GPUs using RapidMind can further accelerate the multi-modality image registration processes without losing accuracy. Thirdly, multi-resolution schemes can be applied to the new model and further increase the efficiency. Fourthly, assuming 2D images can be modeled as special continuous functions so that PDFs can be computed analytically, the mutual information will be computed more accurately and the computational complexity can be even more reduced. Lastly, extending the new model to 3D image registration problems can make our new model more widely applicable.

References

- [1] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, and Pat Hanrahan. Brook for GPUs: stream computing on graphics hardware. *ACM Transactions on Graphics (TOG)*, 23(3):777 – 786, 2004. 19
- [2] Hua-mei Chen and Pramod K. Varshney. Registration of multimodal brain images: some experimental results. In Belur V. Dasarathy, editor, *Proceedings of the SPIE International Conference on Sensor Fusion: Architectures, Algorithms, and Applications VI*, volume 4731, pages 122 – 133, April 2002. 36, 39
- [3] Hua-Mei Chen and Pramod K. Varshney. Mutual information-based CT-MR brain image registration using generalized partial volume joint histogram estimation. *IEEE Transactions on Medical Imaging*, 22(9):1111 – 1119, September 2003. 3, 4, 39, 43
- [4] Hua-Mei Chen, Pramod K. Varshney, and Manoj K. Arora. Performance of mutual information similarity measure for registration of multitemporal remote sensing images. *IEEE transactions on Geoscience and Remote Sensing*, 41(11):2445 – 2454, November 2003. 4, 39, 43
- [5] Razvan Chisu. Techniques for accelerating intensity-based rigid image registration. Master’s thesis, Technische Universität München, January 2005. 2
- [6] Nikos Chrisochoides, Andriy Fedorov, Andriy Kot, Neculai Archip, Peter Black, Olivier Clatz, Alexandra Golby, Ron Kikinis, and Simon K. Warfield. Toward real-time image guided neurosurgery using distributed and grid computing. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, 2006. 2

- [7] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418 – 445, May 1996. 43
- [8] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multimodality medical image registration using information theory. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Information Processing in Medical Imaging*, pages 263–274. Kluwer Academic Publishers, 1995. 3, 4, 12, 43
- [9] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., second edition, 2006. 26
- [10] Geoffrey Egnal and Kostas Daniilidis. Image registration using mutual information. Technical Report MS-CIS-00-05, University of Pennsylvania, 2000. 4, 30, 39, 43
- [11] Jim Xiuquan Ji, Hao Pan, and Zhi-Pei Liang. Further analysis of interpolation effects in mutual information-based image registration. *IEEE Transactions on Medical Imaging*, 22(9):1131 – 1140, September 2003. 4, 39, 69
- [12] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451 – 458, New York, NY, USA, 1994. ACM. 15
- [13] Jundong Liu and Junhong Liu. Artifacts reduction in mutual information-based image registration using prior information. In *Proceedings of IEEE 2003 International Conference on Image Processing*, volume 1, pages I – 1113 – 1116, September 2003. 39
- [14] Frederik Maes, André Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16:187–198, April 1997. 3, 36
- [15] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36, 1998. 6, 7
- [16] William R. Mark, R. Steven Glanville, Kurt Akeley, and Mark J. Kilgard. Cg: a system for programming graphics hardware in a C-like language. In

- International Conference on Computer Graphics and Interactive Techniques*, pages 896–907, New York, NY, USA, 2003. ACM. 19
- [17] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308 – 313, 1965. 46
- [18] Moriyoshi Ohara, Hangu Yeo, Frank Savino, Giridharan Iyengar, Leiguang Gong, Hiroshi Inoue, Hideaki Komatsu, Vadim Sheinin, Shahrokh Daijavad, and Bradley Erickson. Real-time mutual-information-based linear registration on the cell broadband engine processor. In *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 33–36, 2007. 2
- [19] Josien P.W. Pluim, J.B. Antoine Maintz, and Max A. Viergever. Mutual information matching and interpolation artefacts. In Kenneth M. Hanson, editor, *SPIE Medical Imaging 1999: Image Processing*, volume 3661, pages 56 – 65, May 1999. 30, 39
- [20] Harvey Ray, Hanspeter Pfister, Deborah Silver, and Todd A. Cook. Ray casting architectures for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):210 – 223, 1999. 15
- [21] Robert A. Schowengerdt. *Remote sensing : models and methods for image processing*. Elsevier Academic Press, third edition, 2007. 6
- [22] Mika Seppä. Continuous sampling in mutual-information registration. *IEEE Transactions on Image Processing*, 17(5):823 – 826, May 2008. 4, 39
- [23] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379 – 423/623 – 656, July/October 1948. 27
- [24] Renata Smolíková, Mark P. Wachowiak, and Maria Drangova. Registration of fast cine cardiac MR slices to 3D preprocedural images: toward real-time registration for MRI-guided procedures. In *Medical Imaging 2004: Image Processing*, pages 1195 – 1205, 2004. 13
- [25] Renata Smolíková-Wachowiak and Mark P. Wachowiak and Aaron Fenster and Maria Drangova. Registration of two-dimensional cardiac images to preprocedural three-dimensional images for interventional applications. *Journal of Magnetic Resonance Imaging*, 22(2), 2005. 13
- [26] Philippe Thévenaz and Michael Unser. Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image processing*, 9(12):2083 – 2099, December 2000. 4, 77

- [27] Jeffrey Tsao. Interpolation artifacts in multimodality image registration based on maximization of mutual information. *IEEE Transactions on Medical Imaging*, 22(7):854 – 864, July 2003. 4, 39, 43
- [28] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997. 3, 12
- [29] Paul A. Viola. *Alignment by Maximization of Mutual Information*. PhD thesis, Massachusetts Institute of Technology, 1995. 11
- [30] Wolfgang Wein. Intensity based rigid 2D-3D registration algorithms for radiation therapy. Master’s thesis, Technische Universität München, December 2003. 13
- [31] Mason Woo, Jackie Neider, and Tom Davis. *OpenGL Programming Guide*. Addison-Wesley, third edition, 1999. 19
- [32] Lin Xu and Justin W.L. Wan. Real-time intensity-based rigid 2D-3D medical image registration using RapidMind multi-core development platform. In *30th Annual International Conference of the IEEE Engineering in Medical and Biology Society*, pages 5382 – 5385, Vancouver, British Columbia, Canada, August 2008. 2
- [33] Hongying Zhang, Xiaozhou Zhou, Jizhou Sun, and Jiawan Zhang. A novel medical image registration method based on mutual information and genetic algorithm. In *Computer Graphics, Imaging and Vision: New Trends*, pages 221 – 226, 2005. 4, 39, 43
- [34] Maryam Zibaeifard and Mohammad Rahmati. An adaptive simulated annealing scheme for multi-modality medical image registration by maximization of mutual information. In *8th International Conference on Signal Processing*, volume 2, 2006. 4, 43