A Study on the Impact of Curvature on Intensity Based Non-Rigid Medical Image Registration

Submitted in partial fulfillment of the requirements

of the degree of

M.E. in Electronics and Tele-Communication Engineering

By

PRASENJIT KUMAR MUDI

Registration No. – 103738 of 2008-09

Examination Roll No. – M4ETC1608

Under the supervision of

Dr. Ananda Shankar Chowdhury

DEPARTMENT OF ELECTRONICS & TELE-CMMUNICATION ENGINEERING

JADAVPUR UNIVERSITY - KOLKATA

2016

This thesis is dedicated to the memory of my father Late Satrughana Mudi, whom I still miss every day.

CERTIFICATE

This is to certify that the thesis entitled "Study on impact of curvature in non-rigid image registration" has been carried out by Prasenjit Kumar Mudi (Examination Roll No. - M4ETC1608, Registration No. – 103738 of 2008-09) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master in Engineering in Electronics & Tele-Communication Engineering.

Supervisor

Dr. Ananda Shankar Chowdhury

Associate Professor

Department of Electronics &

Tele-Communication Engineering

Jadavpur University

Kolkata-700032

Prof. Palaniandavar Venkateswaran

Professor & Head of the Department

Department of Electronics &

Tele-Communication Engineering

Jadavpur Universty

Kolkata-700032

Prof. Sivaji Bandyopadhyay

Dean, Faculty of Engineering & Technology

Jadavpur University

Kolkata-700032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby approved as creditable study of an engineering subject and presented in a manner satisfactory to warrant its acceptance as pre-requisite to obtain the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for which it is submitted.

Committee on Final Examination

For the evaluation of the Thesis.

Signature of the Examiner

____

Signature of the Supervisor

____

Declaration

I declare that this written submission represents my ideas in my own words and where ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that, I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea /data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Prasenjit Kumar Mudi Date………………..

(Signature)

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Dr Ananda Shankar Chowdhury, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

I am highly grateful to Professor Palaniandavar Venkateswaran, Head of the Department, Electronics and Telecommunication Engineering Department, Jadavpur University, who is kind enough to provide me with all the necessary facilities to carry out this project.

I would like to thank all the members of our group "Imaging, Vision and Pattern Recognition (IVPR)" including PhD scholar Mr. Vijay Gangapure, PhD scholar Rukmini Roy, my friend Ranodip Das for their useful support from time to time. I would like to specially thank Mr. Susmit Nanda, MTech 2015 passed out, for helping me in my thesis in various ways whenever necessary. I shall never forget the contributions given by my undergraduate batch mates Rounak Roy, Sumon Bose, Arka Mukherjee and Bhaskar Sen. It has been a wonderful and challenging experience for me to work in this group.

The Department of Electronics & Tele-Communication Engineering, Jadavpur University, has provided the support and equipment I have needed to produce and complete my thesis and the AICTE (All India Council for Technical Education) has funded my studies.

Finally, I thank my mother and elder brother for supporting me throughout all my studies starting from school to Masters Degree.

Contents

3. Impact of curvature on non-rigid registration

4. Conclusions and Future Directions

List of Tables

List of Figures

Abstract

Problem of non-rigid registration has gained much prominence in the area of bio-medical image analysis. Recently, such a non-rigid registration problem has been modeled as a global optimization problem and was solved using graph cuts. Graph cuts based solution is shown to improve the accuracy of the registration results over other existing approaches. Incorporation of curvature information in graph cuts based non-rigid bio-medical image registration has not been explored considerably. In this thesis, we have undertaken a study of showing the impact of curvature in an intensity-based non-rigid registration process. At first, we discuss how Mean and Gaussian curvatures can be extracted from the brain MRI and retinal images and show appropriate results. Based on these curvature values, every pixel in the source and the target images is labeled by one of the eight local surface primitives. Then, every pair of corresponding pixels, already registered using graph cut based on intensity information only, is checked to see whether their local surface primitives also match. If the local surface primitives do not match, then a displacement map is generated for every such unmatched pixel in source image. The map essentially indicates the local displacement such a source pixel should undergo within a small neighborhood to achieve best curvature-wise matching with the corresponding target pixel. Experimental results clearly corroborate to the fact that intensity based registration cannot completely guarantee curvature-wise best matching.

Chapter 1

INTRODUCTION

This chapter provides an overall outline of this thesis. Curvature of an image is introduced first in the section 1.2. In section 1.3, the concept of rigid and non-rigid registration has been discussed. Finally, the chapter is concluded with an overview of the organization of the thesis in section 1.4.

1.1. Motivation

The motivation behind the thesis is to analyze whether intensity based non-rigid registration can ensure best curvature-wise matching as well. The eventual goal is to achieve better registration with both intensity and curvature information. Non-rigid registration has gained much prominence in biomedical image analysis. In this thesis, we work with brain MRI (Fig. 1.1) and retinal images (Fig 1.2).

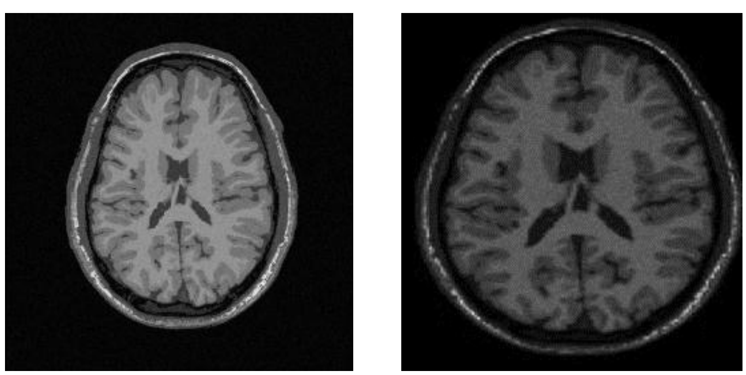


Fig.1.1. Brain MRI: Datasets

Brain MRI data are downloaded from Brainweb [1]. The retinal images are multi-temporal colored images, captured at different times by using a ZEISS FF 450plus IR Fundus Camera with VISUPAC/System 451, which is connected to a JVC digital camera. The images were taken at the same visit or at different dates from the patients at the Kentucky Lions Eye Center.

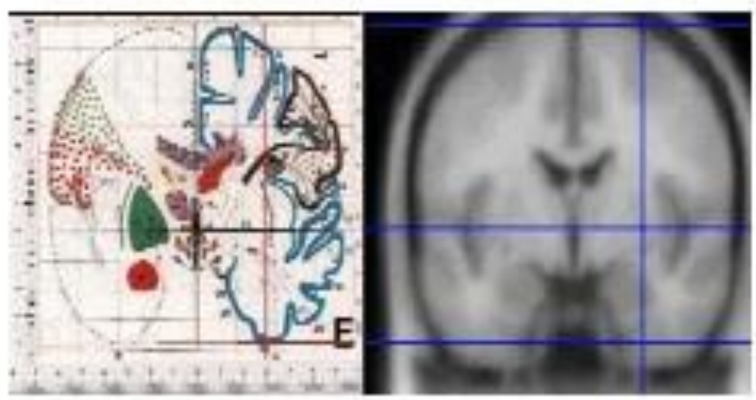


Fig.1.2. Brain atlas of MRI

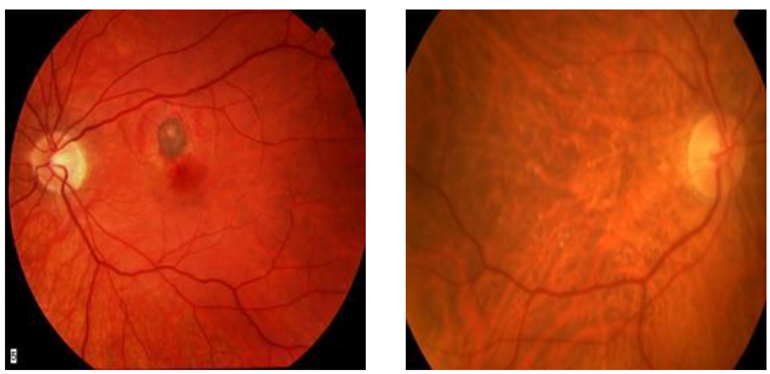


Fig.1.3. Retinal colored images

1.1. Concept of curvature

The problem of image registration arises when images of one object are taken for example at different times, from different perspectives, and/or different imaging devices. The fundamental goal is to combine information and to integrate useful data obtained from the separate images. However, due to spatial distortion of the object under consideration, for example, introduced in motion, the same object in different images is not directly comparable. The idea of image registration is to geometrically transform images in order to compensate for these distortions.

In particular in medical domain, the image registration method has a wide area of applications. To study about the evolution of pathology of a patient, or to take full advantage of the complementary information coming from multimodal imagery [2], image registration is very helpful. This registration makes the diagnosis of a patient easier for a doctor. As for example, to detect brain tumors, doctors often get misdirected after observing scanned brain slice images obtained from imaging devices. Such wrong results can lead to wrong treatments for the patients. Detection of such brain tumors demands very accurate registration which can be possible if curvature information is introduced in existing intensity based greater level accurate Graph-cuts based non-rigid image registration [3]. In other case, diabetic retinopathy, a diabetic eye disease, which occurs to the retina due to diabetes, can eventually lead to blindness. Detection of such disease by analyzing retinal fundus images also requires high accurate image registration [4].

From differential geometry, the second order derivative of a curvilinear function $y = f(x)$ i.e. $\frac{d^2y}{dx^2}$ represents the curvature of a point on a surface.

Considering the diagram in Fig. 1.1, at each point on the curve, the direction of tangent bends at a certain rate in different directions. A measure of this rate of bending is called the curvature and is defined as:

$$\kappa = \left|\frac{f''(x)}{(1+[f'(x)]^2)^{3/2}}\right| \tag{1.1}$$

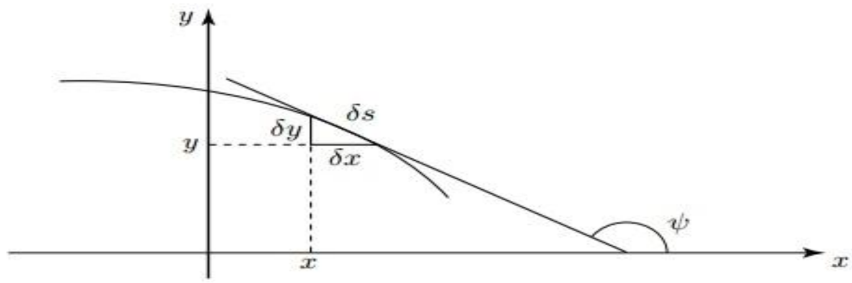


Fig.1.4. Definition of curvature

Now, the curvature information of any pixel in a digital image can be mathematically represented by two principle curvatures, namely, the Mean curvature and the Gaussian curvature. For image registration with non-rigid geometry, two corresponding pixels in the source and the target images will be more accurately registered if curvature information is also considered with their intensity maps.

1.2. Non-rigid registration

Non-rigid image registration plays an important role in bio-medical image analysis. It becomes a challenging problem due to its high degrees of freedoms and inherent requirement of smoothness. Image registration is a process for determining the correspondence of features between images collected at different times or using different imaging modalities. It is mostly used in correction for different patient positions between scans [5].

Image registration has been modeled as an energy minimization problem [5, 6, 7]. This optimization was done by using Graph-Cuts algorithms via alpha-expansions [8, 9, 10]. The dissimilarity measure used in the energy function of this graph-cuts based method can incorporate either SAD (Sum of Absolute Difference), SSD (Sum of Squared Difference) or MI (Mutual Information) [10]. All of these techniques are used to improve the accuracy of the registration outputs as per high requirement in patient diagnosis. One objective is to further improve registration accuracy of such registration methods [11]. Another objective is to incorporate higher order term in the energy function representing the surface properties of registered images. Though there can be other motives, we shall focus on the second objective.

Generally, image registration is of two types: Rigid registration and Non-Rigid registration. In rigid registration, images are assumed to be of objects that simply need to be rotated and/or translated with respect to one another to achieve correspondence.

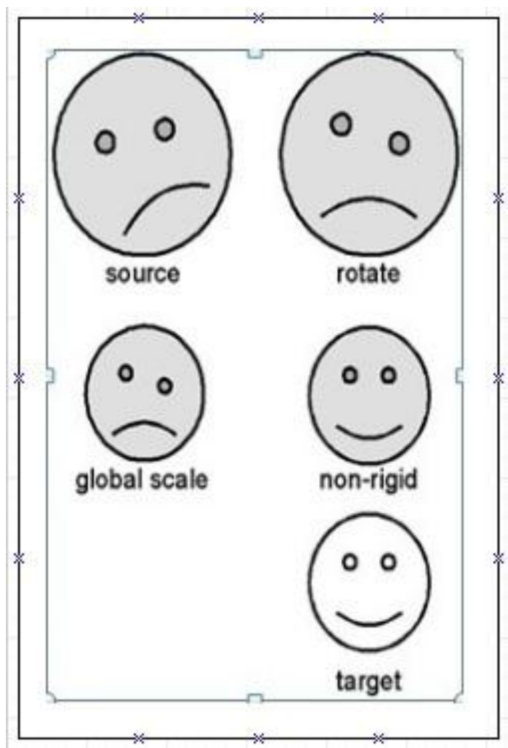


Fig.1.5. Example for Rigid vs. Non-rigid registration

In sharp contrast, non-rigid image registration cannot achieve correspondence between structures in two images, without some localized stretching of the images, either through biological differences or image acquisition or both.

1.3. Thesis organization

The thesis is organized into following three chapters: In chapter 2, detailed discussion of curvature extraction process is depicted. The chapter also discussed the concept local surface primitive class of a pixel in image. In chapter 3, a brief discussion about intensity based registration has been introduced. We show the limitations of intensity based registration. A curvature based method has been proposed for this solution. Curvature based local displacement map is generated for fine tuning of already registered images based on only intensity information. The thesis is concluded in Chapter 4 with potential directions of future research.

References:

[1] B. Aubert-Broche, A. Evans et al, "A new improved version of the realistic digital brain phantom," *NeuroImage, vol. 32 (1), pp. 138-145, 2006*.

[2] Maintz JBA, VandenElsen PA, Viergever MA, "Evaluation of ridge seeking operators for multimodality medical image matching," *IEEE Trans Pattern Anal 1996;18:353–65*.

[3] A. S. Chowdhury, R. Roy, S. Bose, F. Khalifa, A. Elnakib, A. El-Baz, "Non-rigid Biomedical Image Registration Using Graph Cuts With a Novel Data Term," *Proc. Ninth IEEE Int'l Symp. on Biomedical Imaging (ISBI), Barcelona, Spain, 2012*.

[4] D. Usher, M. Dumskyj, M. Himaga, T. H. Williamson, S. Nussey, and J. Boyce, "Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening," *Diabetic Medicine, vol. 21, pp.84-90, 2003.*

[5] D. Rueckert, "Non-rigid registration: Techniques and applications," *Medical Image Registration, CRC Press, 2001.*

[6] W R Crum, T Hartkens, and D L G Hill, "Non-rigid image registration: theory and practice," *British J. Radiology, vol. 77, S140-S153, 2004.*

[7] V. Kolmogorov and R. Zabih, "What Energy Functions can be Minimized via Graph Cuts?," *IEEE Trans, Pattern Anal. Mach. Intel. Vol. 26(2), 147-159, 2004.*

[8] T.W.H. Tang and A.C.S. Chung, "Non-rigid image registration using graph-cuts," *MICCAI LNCS, vol. 4791, pp.916-924, 2007.*

[9] Yuri Boycov, G. F. Lea, "Graph Cuts and Efficient N-D Segementation," *Int'l Journal of Computer Vision 70 (2), 109-131, 2006.*

[10] Y. Boykov and Vladimir Kolmogorov, “An Experimental Comparison of Min-Cut/ Max-Flow Algorithms for Energy Minimization in Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124-1137, Sept. 2004.*

[11] R.W.K. So and A.C.S. Chung, “Non-rigid Image Registration By Using Graph Cuts with Mutual Information,” *IEEE ICIP, pp.4429-4432, 2010.*

[12] P. Kohli, L. Ladicky and P.H.S. Torr, “Robust Higher Order Potentials for Enforcing: Label Consistency,” *Int. J. Computer Vision, vol. 82,302-324, 2009.*

Chapter 2

CURVATURE EXTRACTION

In this chapter detail discussions have been provided for the process of curvature extraction from a digital image. The chapter starts with fundamental concepts of curvature brought from differential geometry in mathematics. Then it is discussed how a pixel of a digital image behaves according to its local surface property or its curvature property. These discussions are depicted in section 2.1. Section 2.2 gives brief idea about local surface class primitives of a pixel. In the last section of the chapter, some graphical results are shown to demonstrate the process of extraction of curvature information from a given image.

2.1. Definition of curvature

It has already been introduced that curvature of an image surface is also very important information in image registration process. Earlier, it has been successfully incorporated to detect some surface related features as ridges, valleys, thin nets, crest lines etc. from digital images [1]. Such features are characterized by points of maximal curvature on the image surface. Likewise, in image registration, if it is possible to model the local surface behavior of every pixel of an image, then it is necessary to align two images with respect to their local surface behavior along with their intensities, to yield more accurate registration.

At any point on a surface, the vector that is at right angle to the surface (tangent at that point on the surface) is called normal vector on that point. Planes containing the normal vector are called normal planes. The intersection of a normal plane and the surface will form a curve called a normal section and the curvature of this curve is called the normal curvature (see Fig. 2.1). For most of the points on the surface, different normal sections will have different curvatures [2]. It demonstrates that curvatures can be represented to characterize a point on any surface.

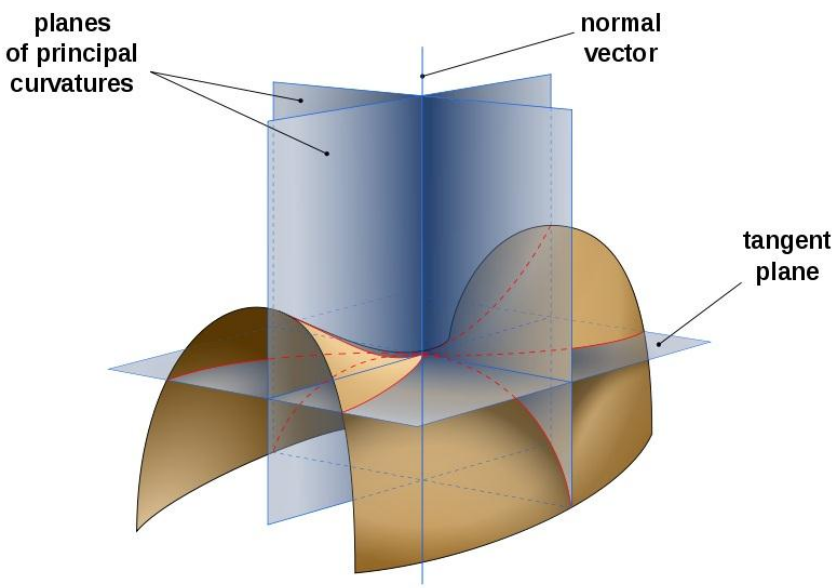


Fig. 2.1. Planes of principal curvatures

From the fundamental differential geometry, we know that, second order derivative of a curvilinear function, $y = f(x)$ i.e. $\frac{d^2y}{dx^2}$ determines the curvature of a point on a surface[3, 4, 5].

Consider the following diagram:

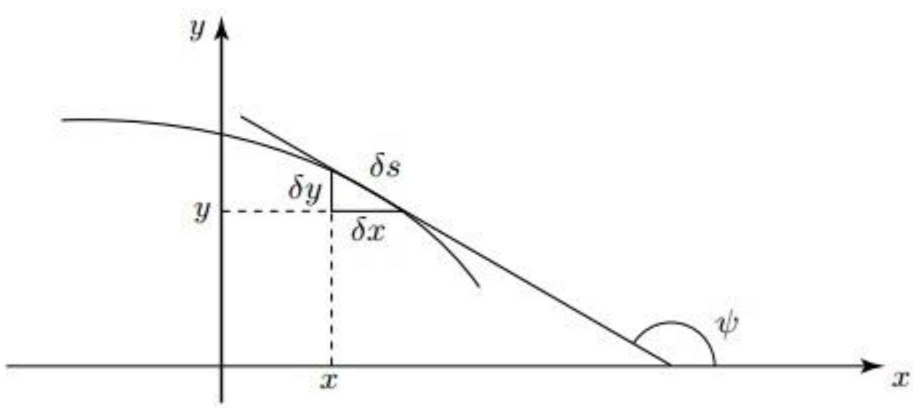


Fig.2.2. Definition of curvature

At each point on a curve, with equation y = f(x), the tangent line turns at a certain rate (Fig. 2.2). A measure of this rate of turning is the curvature:

$$\kappa = \left|\frac{f''(x)}{(1+[f'(x)]^2)^{3/2}}\right| \qquad (2.1)$$

2.1.1. Principal curvature

The differential df of the Gauss map f can be used to define a type of extrinsic curvature, known as the Shape Operator. At a point on the surface, the eigenvalues of the shape operator are called principal curvatures of the surface and the eigenvectors are the corresponding principal directions. These eigenvalues and the eigenvectors at each point thus determine how the surface bends by different amounts in different directions at each point.

Curvature information [13] of any point on a surface can be mathematically modeled with two matrices: first fundamental form matrix and second fundamental form matrix.

2.1.2. First fundamental form matrix

$$I(x,y) = < x, y >$$

$$= x^T \begin{pmatrix} E & F \\ F & G \end{pmatrix} y \qquad (2.2)$$

Where E, F, G are the coefficients of first fundamental form matrix, I.

Illustrative example:

Consider a unit sphere represented as:

$$X(u,v) = \begin{pmatrix} \cos u \sin v \\ \sin u \sin v \\ \cos v \end{pmatrix} \qquad ; (u,v) \in [0,2\pi] \times [0,\pi] \qquad (2.3)$$

Now, $X_u = \frac{\partial X}{\partial u}$

$$or,\ X_u = \begin{pmatrix} -\sin u \sin v \\ \cos u \sin v \\ 0 \end{pmatrix} \tag{2.4}$$

And $X_v = \frac{\partial X}{\partial v}$

$$or,\ X_v = \begin{pmatrix} \cos u \cos v \\ sin\, u \cos v \\ -\sin v \end{pmatrix} \tag{2.5}$$

$$E = X_u . X_u = sin^2 v \tag{2.6}$$

$$F = X_u . X_v = 0 \tag{2.7}$$

$$G = X_v . X_v = 1 \tag{2.8}$$

So, $I = \begin{pmatrix} E & F \\ F & G \end{pmatrix}$ so that $|I| = EG - F^2$ (2.9)

Where E, F & G are given by eq.(2.6) – eq.(2.8).

2.1.3. Second fundamental form matrix:

Consider r = r (u, v) be a regular parameterization of a surface in R^3.

Unit normal vector, $\hat{n} = \frac{r_u \times r_v}{|r_u \times r_v|}$

$$L = r_{uu} . n\ ; \quad r_{uu} = \frac{r_u}{\partial u} \tag{2.10}$$

$$M = r_{uv} . n\ ; \quad r_{uv} = \frac{r_u}{\partial v} \tag{2.11}$$

$$N = r_{vv} . n\ ; \quad r_{vv} = \frac{r_v}{\partial v} \tag{2.12}$$

where L, M, N are the coefficients of second fundamental form matrix, II.

So, $II = \begin{pmatrix} L & M \\ M & N \end{pmatrix}$ so that $|II| = LN - M^2$ (2.13)

A quadratic equation is formed as:

$$|I|k_n{}^2 - (EN + LG - 2FM)k_n + |II| = 0 \quad ; \qquad n = 1, 2 \qquad (2.14)$$

where k_1, k_2 are the two principal curvatures of a point on a surface [12].

2.1.4. Mean Curvature

Mean curvature of a point is given by,

$H = (K_1 + K_2)/2$; where K_1, K_2 are the roots of eq. (2.14).

2.1.5. Gaussian Curvature

Gaussian curvature of a point is defined as:

$K = K_1K_2$; where K_1, K_2 are the roots of eq. (2.14).

These values (H, K) provide the complete curvature information of a point on a surface.

In this way, for 2D and 3D digital images, every pixel on images can also be characterized by its local principal curvatures: Mean and Gaussian curvatures (H, K) [6]. Here, for non-rigid image registration, (H, K) curvature extraction is done for Brain MRI images and colored Retinal images results. For further details of curvatures please refer [7, 8, 9, 10].

2.2. Surface Primitives

From section 2.1, we have learned to extract the Mean and Gaussian curvature of every pixel of an image. Now those curvature values of every pixel determines how much a local surface of that pixel, defined in earlier section 2.1, is bent or curved along with its neighbor pixels. Based on the (H, K) values whether H-K value is positive or negative or zero, a point or pixel on an image surface can also be characterized by some local surface primitives or labels as given in Table 2.1. There are three cases of H-value whether it is zero or positive or negative and same cases for K-value as well. So there will be total 8 local surface class labels and a 'no' surface class label or local surface primitives corresponding to 9 possible combinations of positivity, negativity and zero of (H, K) values taken together [6].

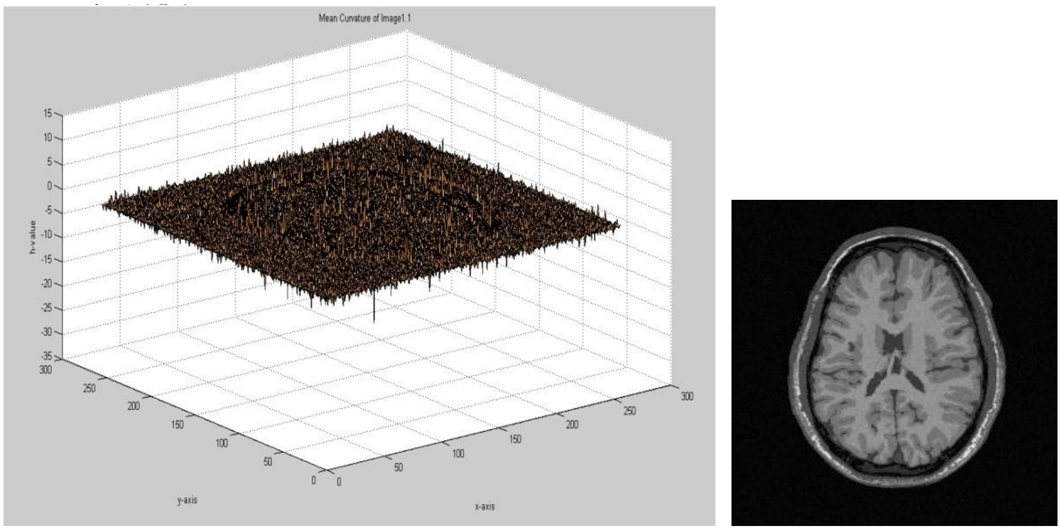
	H<0	H=0	H>0
K<0	Saddle Ridge	Minimal Surface	Saddle Valley
K=0	Ridge Surface	Flat surface	Valley Surface
K>0	Peak Surface	None	Pit Surface

Table 2.1: Local Surface Primitives based on (H, K) [12].

From table 2.1, one can have necessary information about the local geometry of a pixel. Therefore, in non-rigid image registration, two corresponding pixels between source and target images will be curvature-wise aligned if their local surface primitives are properly matched. It may happen that two corresponding pixels are matched intensity-wise but their local surface primitives are not the same.

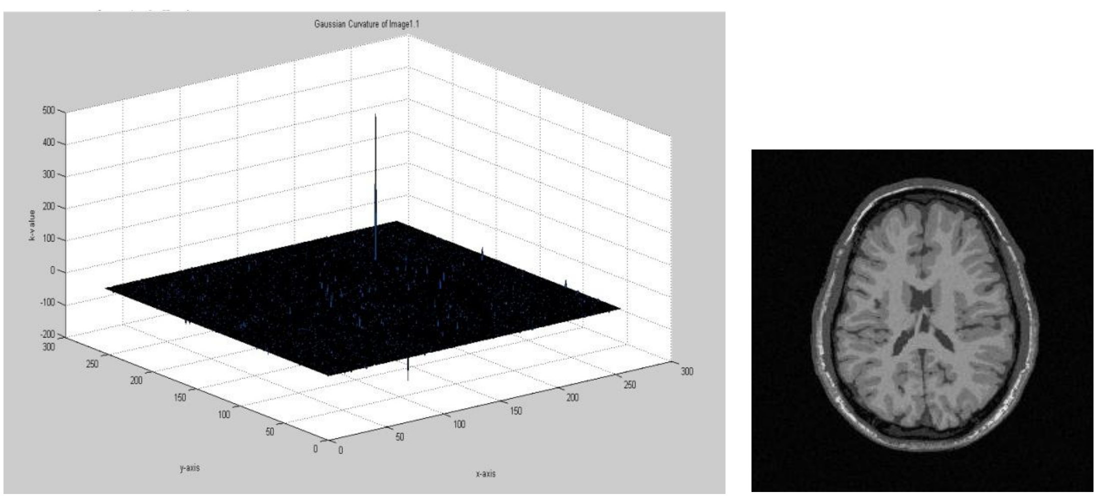
2.3. Some statistical results showing extraction of curvature from images

Datasets which are applied for curvature extraction are taken from Brain MR Images and Retinal Images. Mean and Gaussian curvature values of those images are calculated as descripted in section 2.2. Some results are shown below in Fig.2.3 - Fig. 2.10. From these graphs, we can see that, the mean curvatures are slightly noisy and random whereas the Gaussian curvatures are smoother in the neighborhood.



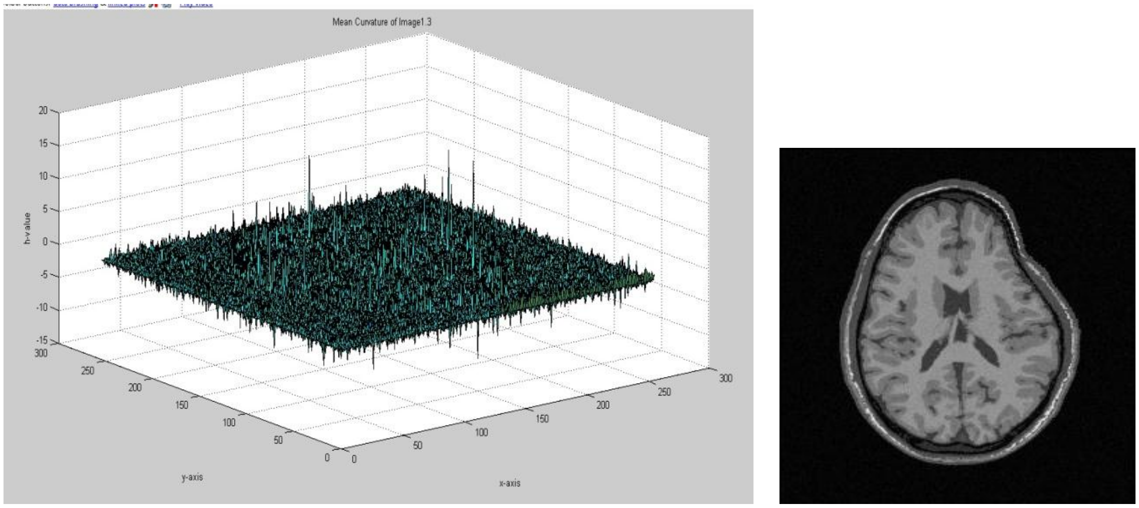
(a) **(b)**

Fig.2.3. (a) Mean Curvature plot; (b) Source Image of Brain MRI DataSet1



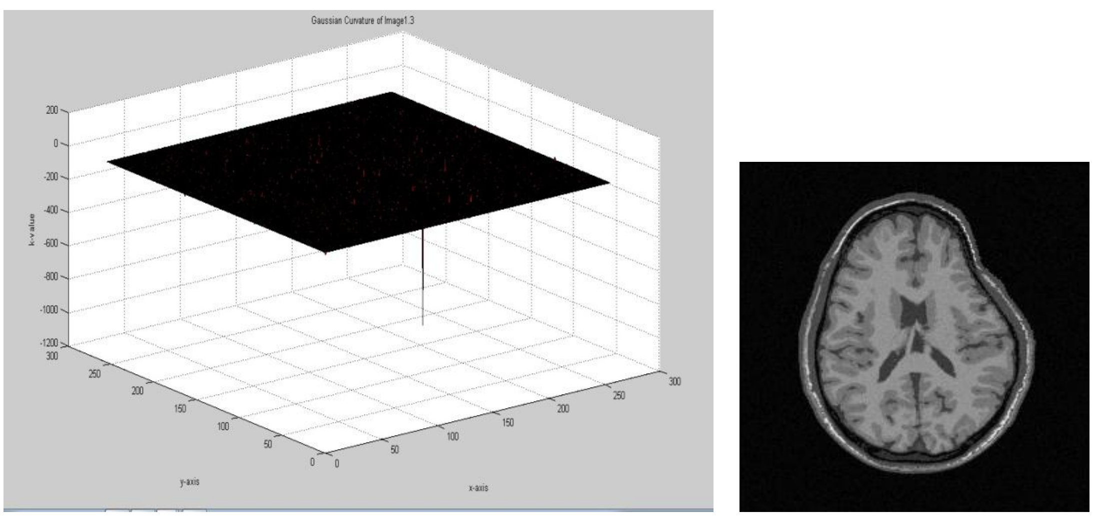
(a) **(b)**

Fig.2.4. (a) Gaussian Curvature plot; (b) Source Image of Brain MRI DataSet1



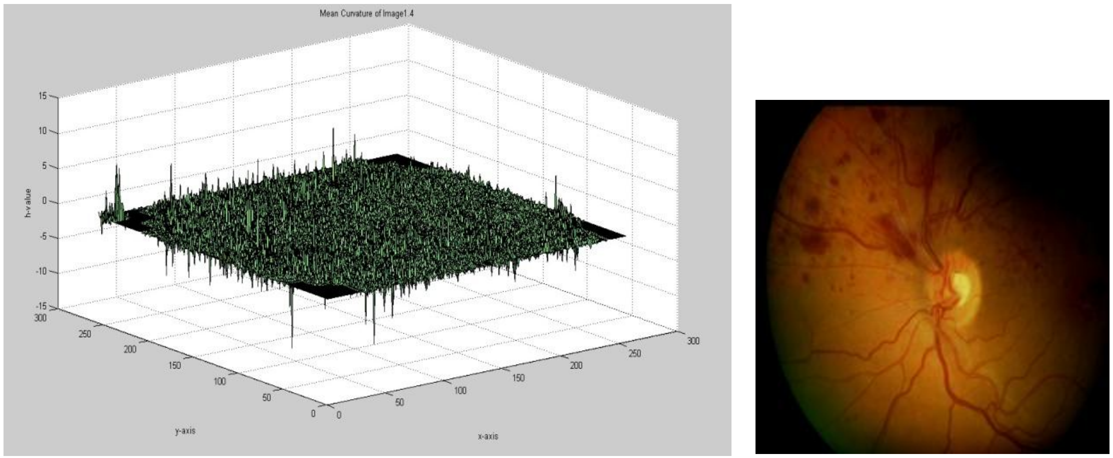
(a) **(b)**

Fig.2.5. (a) Mean Curvature plot; (b) Floating Image of Brain MRI DataSet3



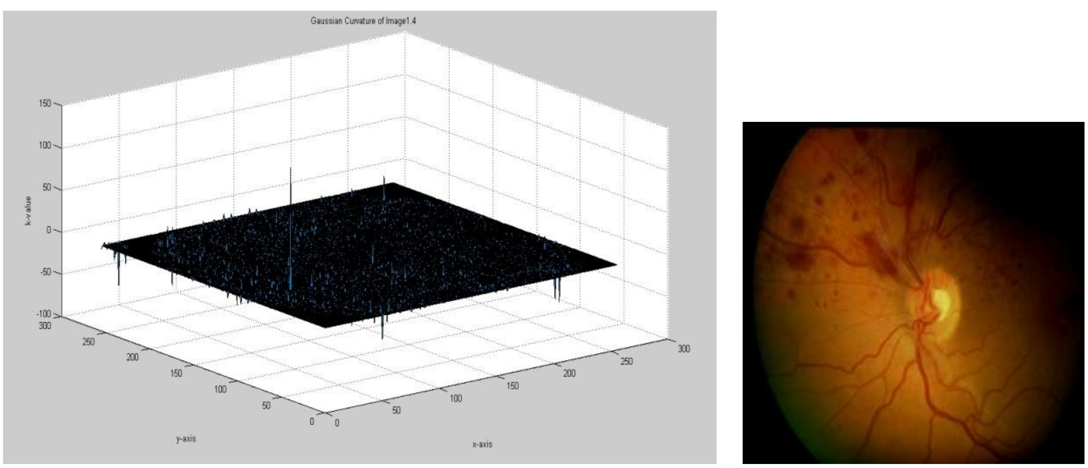
(a) **(b)**

Fig.2.6. (a) Gaussian Curvature plot; (b) Floating Image of Brain MRI DataSet3



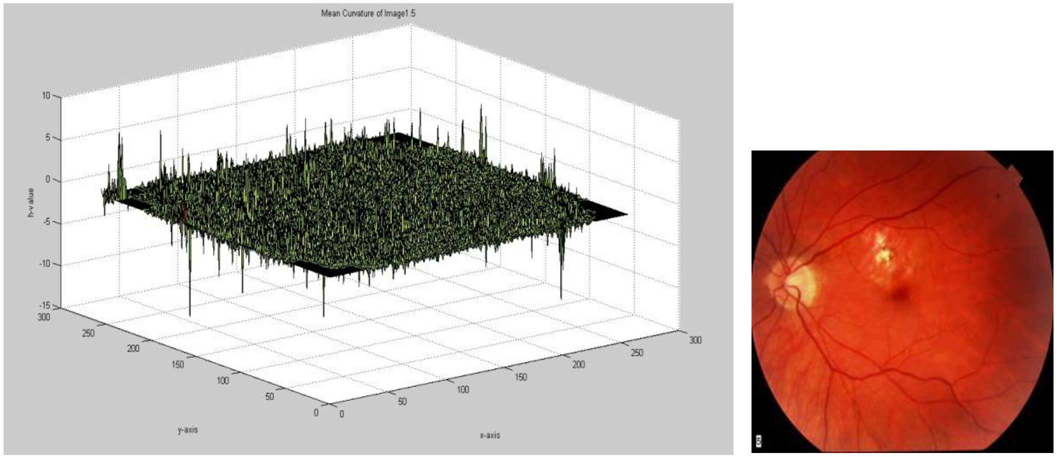
(a) **(b)**

Fig.2.7. (a) Mean Curvature plot; (b) Source Image of Retinal DataSet1



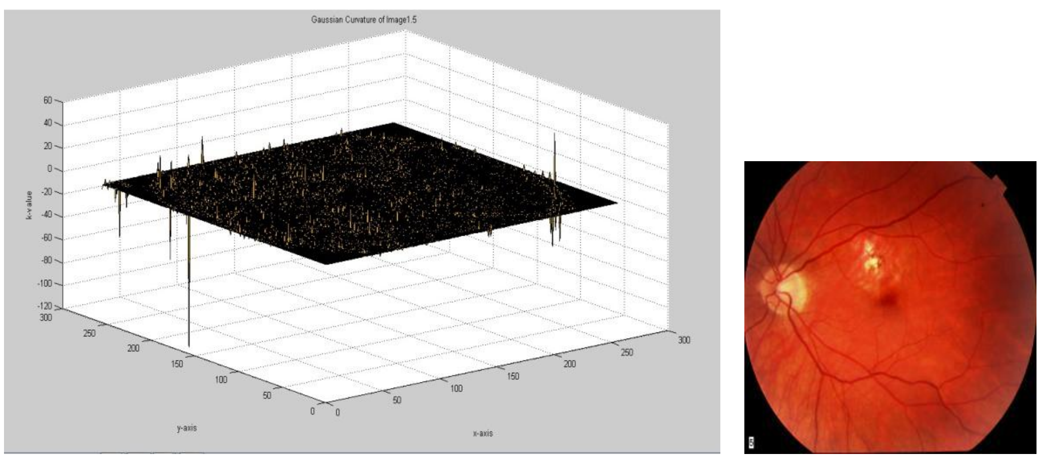
(a) **(b)**

Fig.2.8. (a) Gaussian Curvature plot; (b) Source Image of Retinal DataSet1



(a) **(b)**

Fig.2.9. (a) Mean Curvature plot; (b) Floating Image of Retinal DataSet2



(a) **(b)**

Fig.2.10. (a) Gaussian Curvature plot; (b) Floating Image of Retinal DataSet2

It can be observed that, the Mean curvature values range from -5 to +5 approximately whereas the Gaussian curvature values range from -20 to +20. This shows these values can have negative as well as positive values. If both the curvature values of a pixel become zero, then it can be deemed as locally flat surface.

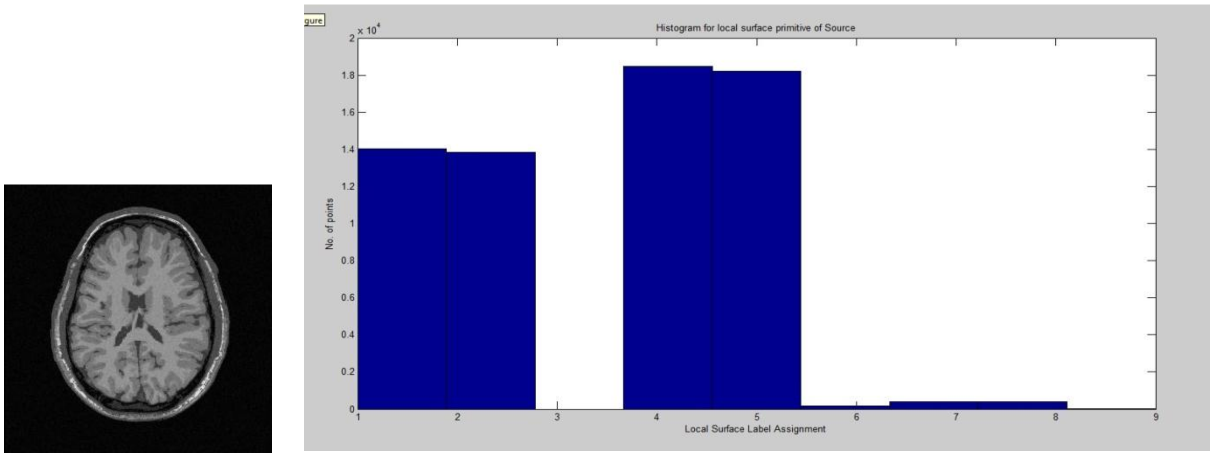
Some statistical histograms are shown below for above mentioned datasets. In each histogram, the no. of pixels belong to a certain surface class primitive is plotted along y-axis and label assignments are plotted along x-axis. The labels represent the following surface classes as given in Table 2.2.

Label Assignment	(H, K) values	Surface Class
1	$H>0$; $K>0$	Pit Surface
2	$H<0$; $K>0$	Peak Surface
3	$H=0$; $K>0$	None
4	$H>0$; $K<0$	Saddle Valley
5	$H<0$; $K<0$	Saddle Ridge
6	$H=0$; $K<0$	Minimal Surface
7	$H>0$; $K=0$	Valley Surface
8	$H<0$; $K=0$	Ridge Surface
9	$H=0$; $K=0$	Flat Surface

Table 2.2: Label Assignments based on Surface Classes

According to Table 2.2, the pixels which have $H>0$ and $K>0$ for example will belong to 'Pit surface' class and have to be assigned label '1'. Similarly, other pixels are assigned corresponding labels of their corresponding surface class belongingness based on their (H, K) values. In each histogram, shown below in Fig. 2.11 – Fig.2.14, most pixels belong to label '1', '3', '4' & '5' corresponding to 'Pit Surface', 'Peak Surface', 'Saddle Valley' & 'Saddle Ridge' respectively.

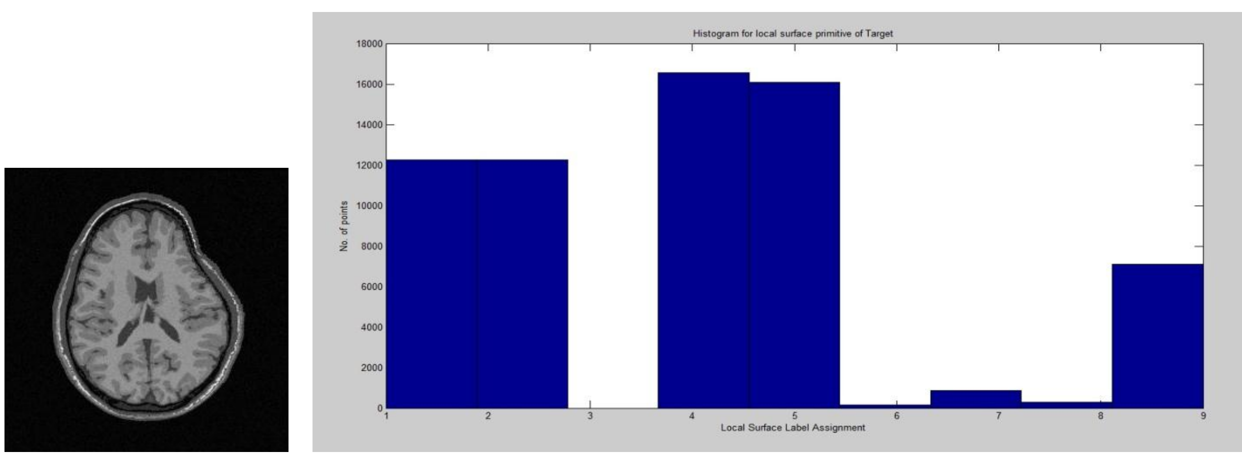
a. Source image of Brain MRI DataSet1 [11]



(a) **(b)**

Fig.2.11. (a) Source image of Brain MRI DataSet1[11] ; (b) Histogram showing no. of pixels assigned surface class labels for Fig.2.11. (a).

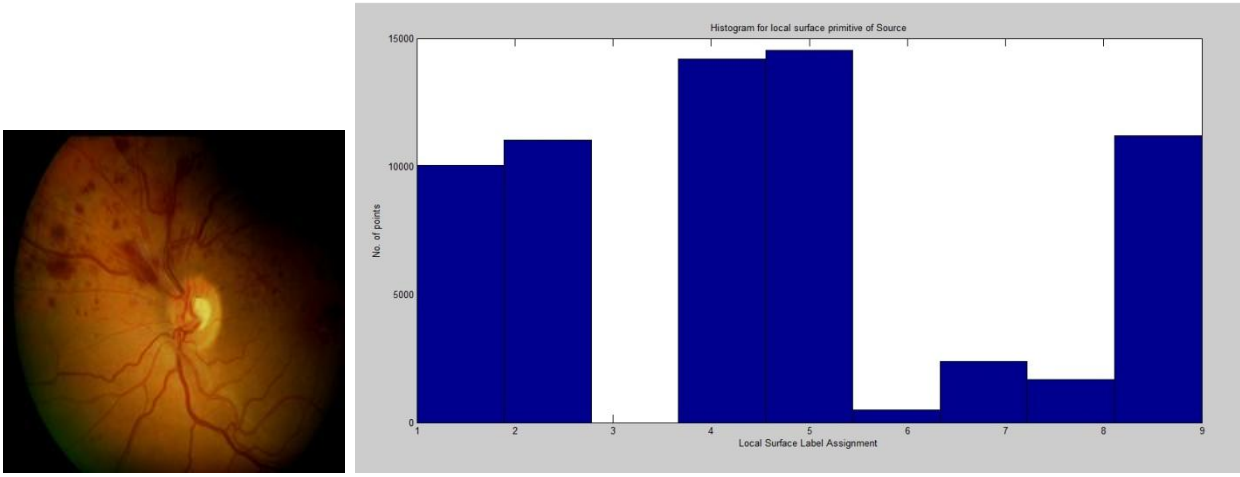
b. Floating image of Brain MRI DataSet3[11]



(a) **(b)**

Fig.2.12. (a) Floating image of Brain MRI DataSet3[11] ; (b) Histogram showing no. of pixels assigned surface class labels for Fig.2.12a.

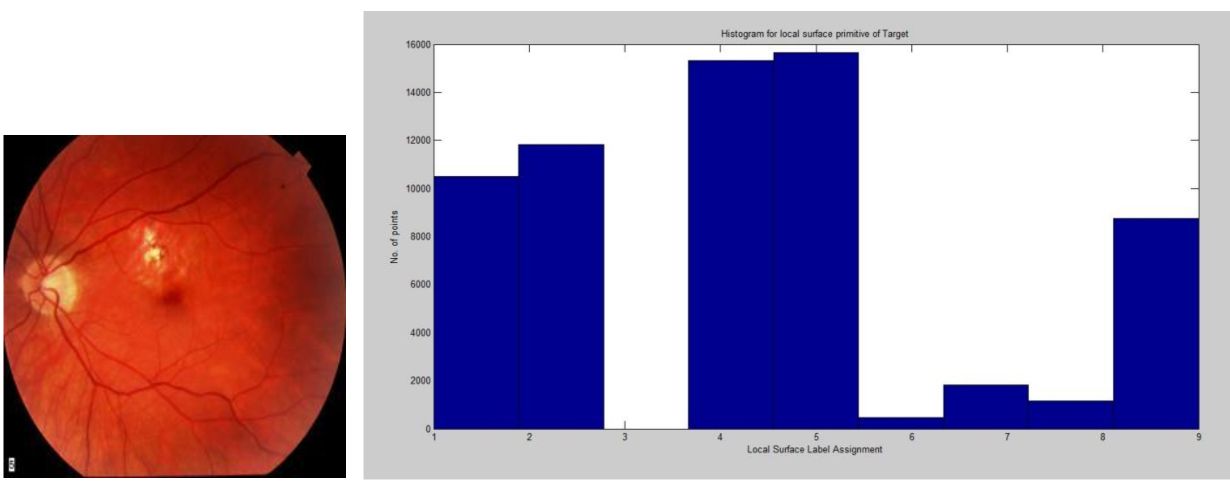
c. Source image of Retinal Fundus DataSet1



(a) **(b)**

Fig.2.13. (a) Source image of Retinal Fundus DataSet1; (b) Histogram showing no. of pixels assigned surface class labels for Fig.2.13. (a).

d. Target image of Retinal Fundus DataSet2



(a) **(b)**

Fig.2.14. (a) Floating image of Retinal Fundus DataSet2; (b) Histogram showing no. of pixels assigned surface class labels for Fig.2.14. (a).

References

[1] P. Kalra and S. Peleg (Eds.), "Computer Vision, Graphics and Image Processing," *ICVGIP 2006, LNCS 4338, pp. 228-239, 2006, © Springer-Verlag Berlin Heidelberg 2006.*

[2] Raussen, Martin. "Elementary Differential Geometry: Curves and Surfaces." *Lecture Notes. World Wide Web, http://www. math. auc. dk/raussen/mrteachgg. html* (1999).

[3] Pauly, Mark, Richard Keiser, Leif P. Kobbelt, and Markus Gross. "Shape modeling with point-sampled geometry." *ACM Transactions on Graphics (TOG)* 22, no. 3 (2003): 641-650.

[4] Stefan, Waner. "Introduction to differential geometry and General Relativity." (2005).

[5] Lai, Rongjie. "Computational differential geometry and intrinsic surface processing." PhD diss., University of California Los Angeles, 2010.

[6] A.S. Chowdhury, J. Burns, B. Sen, A. Mukherjee, J. Yao, R.M. Summers, "Detection of Pelvic Fractures using Graph Cuts and Curvatures," *Proc. Eighteenth IEEE Int'l Conf. on Image Processing (ICIP); Brussels, Belgium (2011), 1605-1608.*

[7] Lin Zeming, He Bingwei, "A curvature-based automatic registration algorithm for the scattered points," *Third International Conference on Measuring Technology and Mechatronics Automation*, 2011.

[8] Peter Dokladal, "Detection of thin, curvilinear structures: Advances, Algorithms and Implementations," *Image Processing. Universite Paris-Est, 2013.*

[9] Perciano, Talita, Roberto M. Cesar Jr, and Roberto Hirata Jr, "Detection of Thin and Ramified Structures in Images using Markov Random Fields and Perceptual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.*

[10] I. Bloch and R. Cesar, "Parameter estimation for ridge detection in images with thin structures," in Progress in Pattern Recognition, *Image Analysis, Computer Vision, and Applications, ser. Lecture Notes in Computer Science, Eds. Springer Berlin / Heidelberg, 2010, vol. 6419, pp. 386–393.*

[11] B. Aubert-Broche, A. Evans et al, “A new improved version of the realistic digital brain phantom,” *NeuroImage, vol. 32 (1), pp. 138-145, 2006*.

[12] Suk, Minsoo, and Suchendra M. Bhandarkar. *Three-dimensional object recognition from range images*. Springer Science & Business Media, 2012.

[13] Battiato, Sebastiano, Sabine Coquillart, Julien Pettré, Robert S. Laramee, Andreas Kerren, and José Braz, eds. *Computer Vision, Imaging and Computer Graphics-Theory and Applications: International Joint Conference, VISIGRAPP 2014, Lisbon, Portugal, January 5-8, 2014, Revised Selected Papers*. Vol. 550. Springer, 2016.

Chapter 3

IMPACT OF CURVATURE ON NON-RIGID REGISTRATION

In this chapter, detailed discussions are made on impact of curvature on non-rigid image registration. We first discuss intensity based non-rigid image registration in section 3.1. In section 3.2, the limitations of intensity based image registration are presented showing the importance of curvature in image registration. Then a method is proposed as possible solution to overcome the limitation of intensity based registration in section 3.3. In section 3.4, we show the experimental results obtained for datasets of Brain MRI images and Retinal images. The chapter concludes with quantitative and qualitative analysis of the results obtained in section 3.4.

3.1. Introduction:

Image registration is a process of spatial transformation of different sets of data into one coordinate system to find the relations between positions in one image with the corresponding positions in one or more other images. One of the scanned images is called as source image or reference image and the other images are called target images. Images can be taken from different sensors, different times, different depths and different viewpoints. The correspondences can be used to change the appearance-by rotating, translating, stretching etc. – of one image so it more closely resembles another so that the pair can be directly compared, combined or analyzed [13, 14].

An image registration algorithm can be decomposed into three components:

a) The similarity measure of how well two images match

b) The transformation model, which specifies the way in which the source image can be changed to match the target. A number of numerical parameters specify a particular instance of the transformation.

c) The optimization process that varies the parameters of the transformation model to maximize the matching criterion [15, 16].

3.2. Intensity based non-rigid registration

We now discuss intensity based image registration. This type of registration matches intensity patterns in each image using mathematical or statistical criteria [1]. They utilize a measure of intensity similarity between the source and target images and adjust the optimal transformation by maximizing the similarity measurement [2]. This results in most similar images after correct optimal registration [19, 20, 21].

3.2.1 Measures of similarity

Measures of similarity have included squared differences in intensities, correlation co-efficient, measures based on optical flow, and mutual information. Some of such measures [1] are detailed below:

(a) **Sum of squared differences (SSD)**

It is defined as:

$$SSD = \frac{1}{N}\sum_{X}(T(x) - S(t(x)))^2 \tag{3.1}$$

Registered images here differ only by Gaussian noise. It is sensitive to small number of voxels that have very large intensity differences. It is suitable for mono-modal image registration [1].

(b) **Correlation Co-efficient**

It is defined as:

$$CC = \frac{\sum_X (T(x) - \bar{T}.(S(t(x)) - \bar{S})}{\sqrt{\sum_X (T(x) - \bar{T})^2 . \sum_x (S(t(x)) - \bar{S})^2}} \tag{3.2}$$

Registered images have linearity intensity relationship and objects of interest are in the field of view of both images. Segmentation of interesting features is often necessary for registration using this kind of similarity measure. It is used in single-modal image registration.

(c) **Correlation ratio**

Correlation ratio is defined as:

$$\gamma = 1 - \frac{1}{N^2} \sum_i N_i {\sigma_i}^2 \tag{3.3}$$

The correlation ratio assumes a functional relationship between intensities. It can be defined in terms of sums and sums of squares of source voxels that correspond to a number N_i of iso-intense voxels in that target image.

(d) **Mutual information**

It is defined as:

$$MI = H_T + H_S - H_{TS} \tag{3.4}$$

It assumes only a probabilistic relationship between intensities. It is defined in terms of entropies of the intensity distribution [2].

(e) **Normalized Mutual Information**

It is defined as:

$$NMI = \frac{H_T + H_S}{H_{TS}} \tag{3.5}$$

It can be proposed to minimize the overlap problem seen occasionally with mutual information [17, 18].

In all these cases, T(X) is the intensity at a position x in an image.

S(t(x)) is the intensity at the corresponding point given by the current estimate of the transformation t(x).

N is the number of voxels in the region of overlap.

3.2.2. Optimization

Optimization refers to the manner in which the transformation is adjusted to improve the image similarity. A good optimizer is one that reliably and quickly finds the best possible transformation.

In non-rigid registration applications, choosing or designing an optimizer can be difficult because the more non-rigid (or flexible) the transformation model, the more parameters are generally required to describe it [3, 4]. For the optimizer this means that more time is required to make a parameter choice and that there is more chance of choosing a set of parameters, which result in a good image match [5, 6].

The method of image registration finds optimal transformation,T^*, which spatially matches a floating image, I_f,to the reference image, I_r, based on some measure of intensity dissimilarity, $C(I_r, T(I_f))$ where, I_r and I_f are intensities of floating image and reference image respectively. A registration problem can be mathematically defined as:

$$T^* = \arg\min_T C(I_r, T(I_f)) \tag{3.6}$$

Where $T(I_f)$ denotes the transformed floating image.

A regularization term is often added to Eq. (3.6) to ensure smooth transformations [7]. Thus, we can write:

$$T^* = \arg\min_T C(I_r, T(I_f)) + \lambda S(T) \tag{3.7}$$

The term $S(T)$ in Eq. (3.7) is called the smoothness penalty and λ is a constant representing the amount of penalty. In case of non-rigid transformation, the pixels in the floating image can move more freely. Therefore, a deformation vector field, **D,** is used to represent the transformation T. So, Eq. (3.2) can be written as:

$$D^* = \arg\min_D C(I_r, D(I_f)) + \lambda S(D) \tag{3.8}$$

There exist several problems with these choices for the data term. Firstly, displacement labels to the pixels in the floating image cannot be directly assigned from the above dissimilarity measures. Secondly, this way of choosing labels often fails to impose strict penalty in the data term for intensity mismatches. Thirdly, change in illumination between the floating and the reference image pair cannot be properly handled. The above goals can be achieved by using a novel function defined as below:

$$C(I_r, D(I_f)) = \lfloor \exp(\textstyle\sum_{x \epsilon X} \|I_r - D(I_f)\|) \rfloor \tag{3.9}$$

SAD (Sum of absolute difference) is directly used in eq. (3.9) as a measure of intensity dissimilarity. In Eq. (3.9), x denotes a pixel in the floating images X. Low displacement labels, which indicate more probable motions, need to be assigned to the pixels in the floating image where the intensity differences with the pixels in the reference image are small and vice-versa. Firstly, using Eq. (3.9), the labels can be directly assigned from the corresponding values of C. Secondly, the exponential function enforces strict penalty by assigning low labels in the data term to pixels in the floating image having small intensity difference with pixels in the reference image. The floor function is used to ensure integer labels. Since we want to assign lower labels (l) for better matches and for all l,$\lfloor l \rfloor \leq \lceil l \rceil$, a floor function is a better choice over a ceiling function. Table 3.1 illustrates the above concepts.

$\sum_{x \epsilon X} \lVert I_r - D(I_f) \rVert$	$C(I_r, D(I_f))$	Label
0	1	1
1	2	2
2	8	8
3	20	20

Table 3.1: Label Assignments from Eq. (3.9)

When $\sum_{x \epsilon X} \lVert I_r - D(I_f) \rVert$exceeds 3, we still assign the corresponding label to be 20. We restrict correspondences between pixels in the floating and reference images by imposing strict limits to intensity differences between them. Hence, this type of label assignment increases the registration accuracy. Notably, change in illumination between two images essentially causes a shift in their intensity patterns. Hence, in case of image pairs with considerable change of illumination, only large labels will be assigned even if the change in intensity due to motion is small. As a result of this incorrect label assignment, registration accuracy gets adversely affected. So the equation (4) is modified by incorporating a term to capture the changes in illumination between a floating and a reference image in the following manner:

$$C(I_r, D(I_f)) = \left\lfloor \exp\left(\sum_{x \epsilon X} \lVert I_r - D(I_f) \rVert - \partial(I_r, D(I_f)) \right) \right\rfloor \tag{3.10}$$

$$\text{Where } \partial(I_r, D(I_f)) = min(\sum_{x \epsilon X} \lVert I_r - D(I_f) \rVert) \tag{3.11}$$

Minimum SAD in equation (3.11) for each pixel in the reference image is calculated over its local neighborhood, and is used to establish better correspondence between the reference and the floating images. Using equations (3.10) and (3.11), we reduce the difference in intensity between the floating and the reference images caused by change of illumination. So, more accurate labels can be assigned and better registration accuracy can be achieved. The smoothness penalty is imposed based on the concept of label consistency using MRF [11] and is given by:

$$S(D) = \sum_{x \in X, y \in N(x)} \|D(x) - D(y)\| \tag{3.12}$$

Where *N(x)* denotes the neighborhood of the pixel *x*. The energy function E_f of the graph cuts based optimization is given by [11]:

$$E_f = \sum_{p \in P} D_p(f_p) + \sum_{p \in P, q \in N(p)} V_{pq}(f_p, f_q) \tag{3.13}$$

Where D_p is the data term, V_{pq} is the smoothness term and $f_p(f_q)$ denotes the intensity of the pixel p (q). The optimal transformation (D*) for non-rigid registration in Eq. (3.8) is similar to the graph cuts-based energy function (E_f) [22, 23] in Eq. (3.13).Graph cuts with α–expansion [8, 9] is employed to obtain optimal transformation through the assignment of multiple displacement labels. We directly apply the above equations for registration of brain images. For the colored retinal images, we treat red (R), green (G) and blue (B) components separately [10, 11]. So, three optimal transformations *DR **, *DG** and *DB** for the R, G and B components are obtained. Three separately registered images using *DR**, *DG** and *DB** are then generated. Finally, a test image is employed to obtain the composite non-rigidly registered image from these three components.

3.3. Shortcomings of intensity based registration

In the preceding section 3.1, it is shown how non-rigid image registration is being carried out based on intensity pattern between source and target images. But a very important shortcoming of intensity based non-rigid image registration is avoiding curvature information of a pixel in an image.

3.3.1. Registration accuracy

In earlier sections, we have seen the behavior of curvature information of every pixel in image. Matching of intensities between two images thus no longer gives us the most accurate registration results. The errors remained in intensity based registration can further also be rectified by incorporating the curvature information of every pixel in image.

Obviously, intensity based registration gives us maximum matching between two images. However, we have come across pixels, which are already registered intensity-wise, can differ in their curvature values. One possible reason of this behavior could be the lack of explicit use of curvature information in the intensity based registration process. It may happen in some cases that, two corresponding pixels are having, for an example, flat surface and ridge surface (based on local surface behavior of a pixel) but their intensities are same. That will lead us to improper registration results. So, curvature information should be incorporated along with intensity of each pixel in image during non-rigid image registration to enhance the registration accuracy.

3.3.3. Adaptive window size

We know that in non-rigid image registration, the pixels in floating image can move freely. So, a window of size 31×31 has been used to determine how much displacement of pixels take place through this window. This is used to make the registration more error free and accurate which is the high demand in biomedical applications. But the displacement of pixels is not uniform throughout the images. Therefore, an adaptive window size should be introduced to fix dynamically the size of window according to the displacements of pixels in different regions of images.

3.4. Possible improvements using curvature

As discussed in earlier sections, curvature information should be used to achieve better registration accuracy. We suggest a method to demonstrate the importance of curvature in non-rigid image registration. The schematic of the proposed method is shown below:

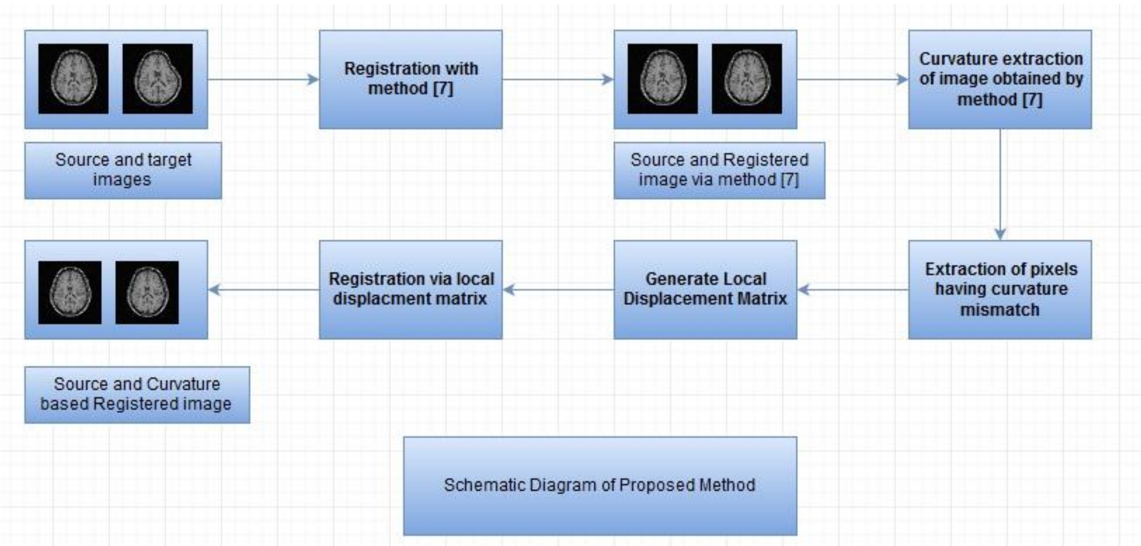


Fig.3.1. Schematic diagram of the proposed method

In this method, firstly, the registered image of the registration method [7] along with source image [7] is taken as input images. This registered image obtained from method [7] is taken as target image for further curvature based registration process. Mean and Gaussian curvatures of every corresponding pixels in source and target image is obtained as discussed earlier. Then local surface primitives are checked between corresponding pixels. If the surface primitive do not match, then a local displacement matrix is generated based on the Mean and the Gaussian curvature information for every pixel in its neighborhood. Based on this local displacement matrix, every pixel is displaced to its correct position. In this way, the target images are registered with the source image based on the curvature properties.

3.4.1. Curvature extraction of intensity based registered image

Firstly, a source image from Brain MRI [12] or Retinal image datasets and the corresponding registered image, obtained via non-rigid image registration method [7], as the target image are taken as inputs. Then Mean and Gaussian curvature values are obtained for every corresponding pixel in the source and the target images. The method of extraction of Mean and Gaussian curvature has already been discussed in Chapter 2. Those curvature values are denoted as (H, K) values where H = Mean curvature and K = Gaussian curvature. Some graphs showing the curvature extraction have been shown in Chapter 2, section 2.3.

3.4.2. Local displacement matrix generation

We have estimated (H, K) values of every corresponding pixel in target image.

a. Local surface primitive

Based on the Table 2.1 and the curvature values obtained, every pixel in target image are labeled according to their local surface primitives. Then these surface primitives between corresponding pixels are compared if they are same or not. If same then they belongs to same neighborhood and same surface primitive class. But if they are not same, then their curvature difference are compared which is discussed in the succeeding sub-section.

b. Curvature difference metric

When the corresponding pixels belong to same local surface primitive class, then a difference metric is used to determine curvature difference between each corresponding pixels in source and target images. The metric is given by:

$$\delta_{curvature} = \sqrt{(H_{source} - H_{target})^2 + (K_{source} - K_{target})^2} \qquad (3.14)$$

Where (H_{source},K_{source}) and (H_{target}, K_{target}) are corresponding (H, K) values of source and target image respectively.

c. Thresholding

The curvature difference metric thus give us curvature difference values between source and target images. Then thresholding of those curvatures is done with a chosen threshold value. The pixels below this threshold are the points of interest. Local displacement will have to generate for those interested points.

This curvature difference metric between two corresponding pixels is then checked if they are equal or not. If they are checked to be equal then these two corresponding pixels are properly aligned based on curvature. But if they differ in curvature metric, then a local first order neighborhood is chosen for each pixel.

d. Local displacement map

In the chosen local neighborhood of 8 neighbors, differences in curvature between the pixel in target image and its neighbors are calculated. So, a 3×3 matrix containing local curvature differences in the neighborhood is obtained.

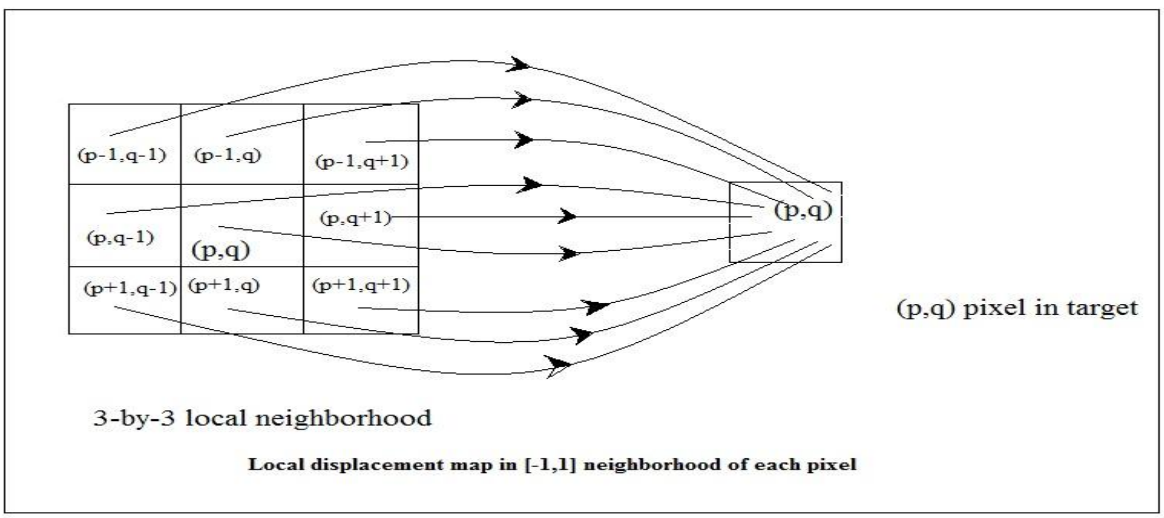


Fig.3.2. Schematic of Local displacement matrix generation

In the matrix, the position with minimum curvature difference is found. Now, if two corresponding pixels are found to have minimum curvature difference, then no displacement is required. Otherwise, the target pixel is displaced to the position with minimum curvature difference. This matrix so generated is called the local displacement matrix. The position where minimum difference is found will then represent the required local displacement of the pixel in the target image. We can observe that, more curvature difference for a pixel requires more local displacement whereas less curvature difference requires less local displacement for that pixel.

3.5. Experimental Results

The flowing table 3.2 demonstrates the comparative results between the method [7] and the proposed method. The number of pixels in registered image from method [7] which are matched with source based on curvature values is calculated. Then after creating local displacement matrix according to the proposed method, number of matched pixels based on curvature difference is obtained. So, the number of pixels, which are aligned based on curvature information, can be obtained. For Brain MRI datasets, maximum of 2430 pixels (3.7% of the image pixels) were not registered properly in method [7]. On other way, maximum of 34271 pixels (52.29% of image pixels) were not registered properly in method [7]. This no. of pixels can be properly aligned based on curvature.

3.5.1. Results for Brain MRI Datasets

a. Quantitative results

$\boldsymbol{N_{ISBI}}$ = No. of pixels matched with intensity after registered via intensity based registration method [7].

N_{LDM} = No. of pixels matched with curvature after creating local displacement matrix in the proposed curvature based method.

$N_{aligned}$ = No. of pixels aligned with local displacement map in the proposed method to improve registration accuracy.

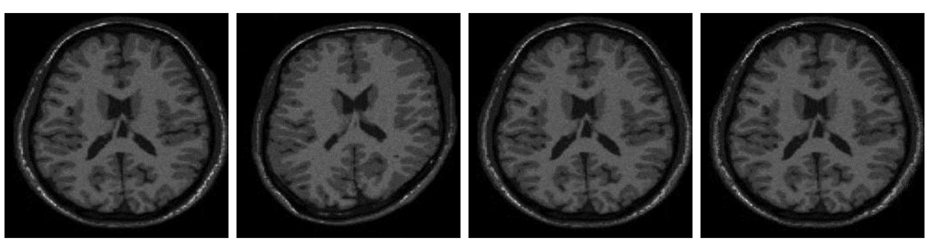
Datasets	N_{ISBI}	N_{LDM}	$N_{aligned}$
DataSet1	61018	63272	2430
DataSet2	*65529*	*65536*	*0*
DataSet3	*64592*	*64944*	*596*
DataSet4	*65271*	*65310*	*226*
DataSet5	*64764*	*64141*	*395*

Table 3.2: Comparison between method [7] & proposed method based on no. of pixels in Brain MRI [24] images registered with intensity & curvature

b. Qualitative results

The output registered images of our proposed method for Brain MRI datasets are shown and compared with their corresponding and floating image counterparts.

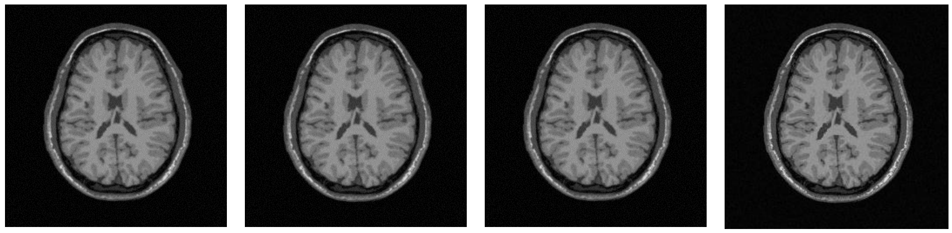
Dataset 1



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

Fig. 3.3: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

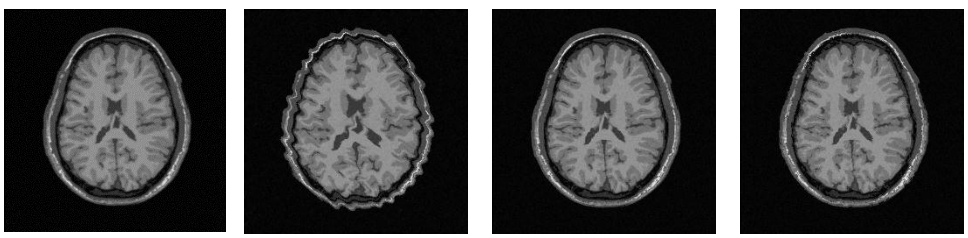
Dataset 2



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

Fig. 3.4: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

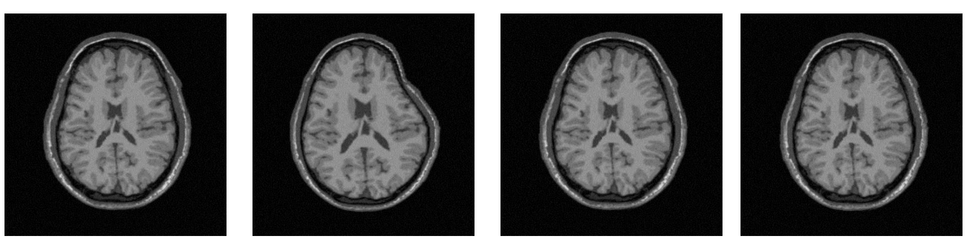
Dataset 3



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

Fig. 3.5: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

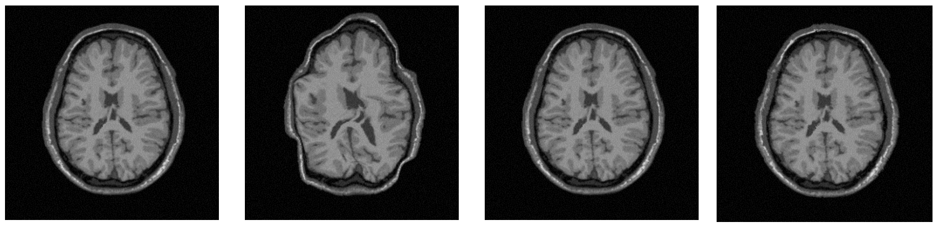
Dataset 4



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

Fig. 3.6: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

Dataset 5



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

Fig. 3.7: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

3.5.2. Results for Retinal colored images

a. Quantitative results

$\boldsymbol{N_{ISBI,R}}, \boldsymbol{N_{ISBI,G}}, \boldsymbol{N_{ISBI,B}}$ = No. of pixels matched with intensity after registered via intensity based registration method [7] for red, green and blue components of colored Retinal image respectively.

$\boldsymbol{N_{LDM,R}}, \boldsymbol{N_{LDM,G}}, \boldsymbol{N_{LDM,B}}$ = No. of pixels matched with curvature after creating local displacement matrix in the proposed curvature based method for red, green and blue components of colored Retinal image respectively.

$\boldsymbol{N_{ali,R}}, \boldsymbol{N_{ali,G}}, \boldsymbol{N_{ali,B}}$ = No. of pixels aligned with local displacement map in the proposed method red, green and blue components of colored Retinal image respectively to improve registration accuracy.

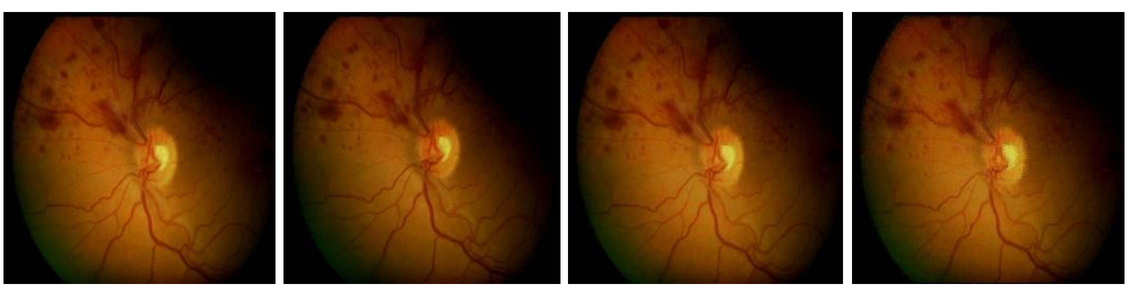
Datasets	$N_{ISBI,R}$	$N_{ISBI,G}$	$N_{ISBI,B}$	$N_{LDM,R}$	$N_{LDM,G}$	$N_{LDM,B}$	$N_{ali,R}$	$N_{ali,G}$	$N_{ali,B}$
DataSet1	55913	61144	49572	35982	38345	36875	16207	14792	25287
DataSet2	59902	53506	56148	62931	59968	61341	2705	5851	4418
DataSet3	37704	39369	57469	44558	42563	54715	23178	26923	12966
DataSet4	10574	34881	10392	32106	43716	32894	34342	22206	34271
DataSet5	20003	37683	12065	35192	44424	33818	30903	21705	32726

Table 3.3: Comparison of method [7] and proposed method based on no. of pixels in Retinal fundus images registered with intensity & curvature.

b. Qualitative results

The output registered images of our proposed method for Retinal image datasets are shown and compared with their corresponding and floating image counterparts.

Dataset 1



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

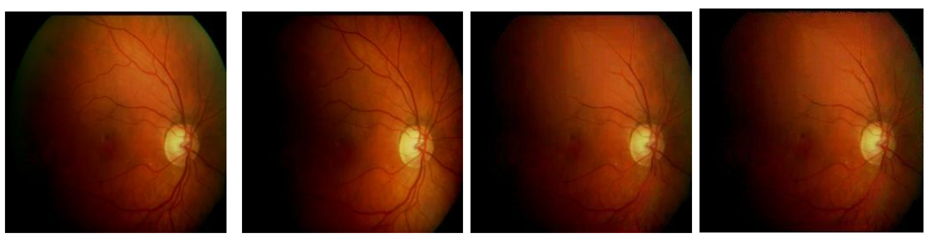
Fig. 3.8: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

Dataset 2

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

Fig. 3.9: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

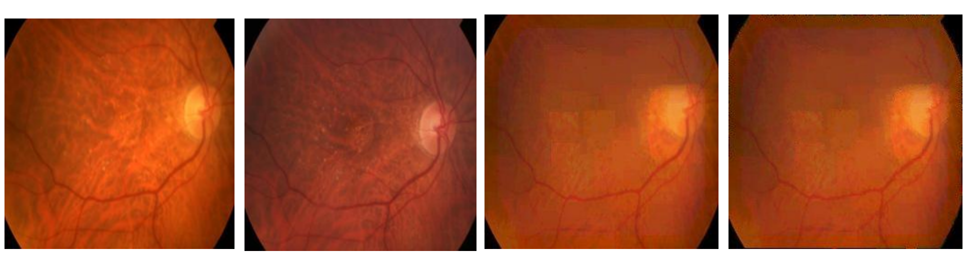
Dataset 3



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

Fig. 3.10: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

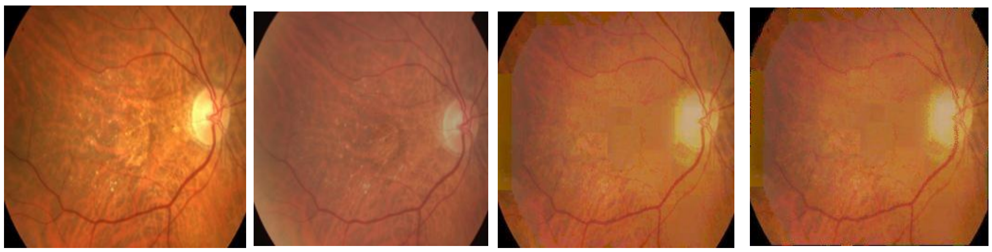
Dataset 4



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

Fig. 3.11: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

Dataset 5



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

Fig. 3.12: (a) Source image ; (b) Floating image ; (c) Registered image of method [7] (d) Registered image of proposed method.

Average execution times for registering a brain image and a retinal image are 4 min and 12 min respectively for our proposed method on a typical desktop with Intel (R) Core (TM) i3-3210 processor with a speed of 3.20 GHz and 4.00 GB of memory and 64-bit Operating System.

3.6. Analysis of Results

We now analyze the results obtained in the previous section. Both quantitative and qualitative analyses are included.

3.6.1. Quantitative analysis

From the quantitative analysis through Table 3.2.a and Table 3.2.b, we can observe that, there are some pixels which were not properly registered in the registration method [7]. Those pixels have to be properly registered with their curvature properties. That has been showed in the table how many pixels need to be aligned in our proposed method. This clearly demonstrates that our proposed method is supposed to give better results than the registration method [7] based on curvature information.

The output registered images for all the 5 datasets of Brain MRI images [24] and 5 datasets of Retinal colored images obtained from the proposed method has been shown in section 3.4.1 and section 3.4.2. These shows the qualitative results of the proposed method.

References

[1] W R Crum, T Hartkens, and D L G Hill, “Non-rigid image registration: theory and practice,” *British J. Radiology, vol. 77, S140-S153, 2004*.

[2] R.W.K. So and A.C.S. Chung, “Non-rigid Image Registration By Using Graph Cuts with Mutual Information,” *IEEE ICIP, pp.4429-4432, 2010.*

[3] Mori K, Deguchi D, Sugiyama J, Suenaga Y, Toriwaki J, Maurer CR, et al., “Tracking of a bronchoscope using epipolar geometry analysis and intensity-based image registration of real and virtual endoscopic image,” *Med Image Anal 2002;6:321–36.*

[4] Lester H, Arridge SR, “A survey of hierarchical non-linear medical image registration,” *Pattern Recognition 1999;32:129–49.*

[5] Fox NC, Crum WR, Scahill RI, Stevens JM, Janssen JC, Rossor MN, “Imaging of onset and progression of Alzheimer’s disease with voxel-compression mapping of serial MRI,” *Lancet 2001;358:201–5.*

[6] Besl PJ, McKay HD, “A method for registration of 3-D shapes,” *IEEE Pattern Analysis Machine Intelligence 1992;14:239–56.*

[7] A. S. Chowdhury, R. Roy, S. Bose, F. Khalifa, A. Elnakib, A. El-Baz, “Non-rigid Biomedical Image Registration Using Graph Cuts With a Novel Data Term,” *Proc. Ninth IEEE Int’l Symp. on Biomedical Imaging (ISBI), Barcelona, Spain, 2012*.

[8] Subsol G, Roberts N, Doran M, Thirion JP, Whitehouse GH. Automatic analysis of cerebral atrophy. *Magn Reson Imaging 1997;15:917–27.*

[9] V. Kolmogorov and R. Zabih, “What Energy Functions can be Minimized via Graph Cuts?,” *IEEE Trans. Pattern. Anal. Mach. Intel. vol. 26(2), 147–159, 2004.*

[10] Aylward SR, Jomier J, Weeks S, Bullitt E. Registration and analysis of vascular images. *Int’l J Computer Vision 2003;55:123–38*.

[11] Lucac R. and K. Plataniotis, Color Image Processing: Methods and Applications, CRC Press, Florida, USA, 2006.

[12] B. Aubert-Broche, A. Evans et al, “A new improved version of the realistic digital brain phantom,” NeuroImage, vol. 32 (1), pp. 138-145, 2006.

[13] D. Rueckert, “Non-rigid registration: Techniques and applications,” Medical Image Registration, CRC Press, 2001.

[14] J. Atif; CNRS, Orsay, France ; X. Ripoche ; C. Coussinet ; A. Osorio, “Non rigid medical image registration based on the maximization of quadratic mutual information,” *IEEE 29th Annual, Proceedings of Bioengineering Conference, 2003.*

[15] B. Fischer and J. Modersitzki, “ A unified approach to fast image registration and a new curvature based registration technique, ” *Linear Algebra Appl., vol. 380, pp. 107-*

124, 2004.

[16] F. Maes, D. Vandermeulen, and P. Suetens, “ Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information, ” *Med. Image Anal., vol. 3, no. 4, pp. 373–386, 1999*.

[17] P. Thévenaz and M. Unser, “ Optimization of mutual information for multiresolution image registration, ” *IEEE Trans. Image Process., vol. 9, no. 12, pp. 2083–2099, Dec.2000*.

[18] S. Klein, M. Staring and J. P. W. Pluim, “Evaluation of Optimization Methods for Non-rigid Medical Image Registration Using Mutual Information and B-Splines,” *IEEE Trans. On Image Processing, Vol. 16, No-12, 2007.*

[19] X. Pennec, P. Cachier and N. Ayache, “Tracking brain deformations in time sequences of 3D US images, ” *Pattern Recognit. Lett. , vol. 24, no. 4 – 5, pp. 801 – 813, 2003.*

[20] L. A. Brown, “A survey of image registration techniques,” *ACM Computing Surveys, Vol. 24, No-4, pp. 325-376, 1992*.

[21] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar-Jr., H. F. Jelinek, and M. J. Cree, “Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification,” *IEEE Trans. Med. Imag., vol. 25, pp. 1214–1222, 2006*.

[22] Yuri Boycov, G. F. Lea, “Graph Cuts and Efficient N-D Segementation,” *Int’l Journal of Computer Vision 70 (2), 109-131, 2006.*

[23] Y. Boykov and Vladimir Kolmogorov, “An Experimental Comparison of Min-Cut/ Max-Flow Algorithms for Energy Minimization in Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124-1137, Sept. 2004.*

[24] B. Aubert-Broche, A. Evans et al, “A new improved version of the realistic digital brain phantom,” *NeuroImage, vol. 32 (1), pp. 138-145, 2006*.

Chapter 4

Conclusions and Future Directions

In this chapter, the conclusions of the research work have been provided with key contributions of my work in this thesis. The probable future scopes of research are mentioned at the end of this chapter.

4.1. Conclusions:

The problem of non-rigid bio-medical image registration has become nowadays a very challenging problem due to its high degrees of freedoms and inherent requirement of smoothness. Due to high demand in high registration accuracy as for example in bio-medical applications such as brain tumor detection, the intensity based non-rigid image registration leave errors to some extent which are further need to be rectified. This thesis provides a direction towards possible improvement of intensity based non-rigid registration by incorporating curvature information of pixel in an image.

The proposed method introduces the fundamental concepts of Mean and Gaussian curvature of a pixel in image and its relation to the non-rigid registration. The method generates a Local Displacement Matrix (LDM) for every pixel in target image for fine tuning of intensity based registration based on Mean and Gaussian curvature information after labeling every pixel according to its local surface primitive class. A SSD (Sum of Squared Differences) metric has been used here for obtaining differences in curvature values between corresponding pixels in source and target images.

Experimental results, both in quantitative and qualitative way, demonstrate the importance of curvature in no-rigid intensity based image registration. The method shows the possibility of achieving better registration accuracy as well as less execution time.

4.2. Future directions:

The thesis proposed a method to demonstrate the importance of curvature in non-rigid image registration. This ensures that curvature should be undoubtedly incorporated in non-rigid image registration. Now, we have to focus on the compatible mathematical modeling of curvature information so that it can be incorporated inside the data term of the energy function required for Graph-Cuts based Optimization. In future, this optimization should have to be implemented using Graph-Cuts Optimization. After achieving this goal, other sub research objectives can also be future research direction such as creation of adaptive window size to enhance execution speed of the registration method.

Appendix A

Matlab Code Implementation for Histogram Generation

```
%-----------Matlab Code for Histogram generation showing curvature
%variation among pixels in source and target images-----------------------

a=imread('source1.jpg');
b=imread('target2.jpg');
aa=double(a);
bb=double(b);
c=zeros(256+30);
for i=16:271
    for j=16:271
        c(i,j)=bb(i-15,j-15);
    end
end
cc=zeros(256+30);
for i=16:271
    for j=16:271
        cc(i,j)=b(i-15,j-15);
    end
end

%---------------curvature computation of Source1------------

[hsource,ksource] = curvature_extraction(aa);

%---------------curvature computation of Registered----------

[htarget,ktarget] = curvature_extraction(c);

%----------------------labeling of source according curvature classes----
hksource_label = zeros(256);
for i = 1:256
    for j = 1:256
        if(hsource(i,j) > 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 1;
        else if(hsource(i,j) < 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 2;
        else if(hsource(i,j) == 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 3;
```

```
        else if(hsource(i,j) > 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 4;
        else if(hsource(i,j) < 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 5;
        else if(hsource(i,j) == 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 6;
        else if(hsource(i,j) > 0) && (ksource(i,j) == 0)
            hksource_label(i,j) = 7;
        else if(hsource(i,j) < 0) && (ksource(i,j) == 0)
            hksource_label(i,j) = 8;
            else
            hksource_label(i,j) = 9;
            end
            end
            end
            end
            end
            end
            end
        end
    end
end

%-----------------------labeling of target according curvature classes----
hktarget_label = zeros(256+2);
for i = 1:256
    for j = 1:256
        if(htarget(i,j) > 0) && (ktarget(i,j) > 0)
            hktarget_label(i+1,j+1) = 1;
        elseif(htarget(i,j) < 0) && (ktarget(i,j) > 0)
            hktarget_label(i+1,j+1) = 2;
        else if(htarget(i,j) == 0) && (ktarget(i,j) > 0)
            hktarget_label(i+1,j+1) = 3;
        else if(htarget(i,j) > 0) && (ktarget(i,j) < 0)
            hktarget_label(i+1,j+1) = 4;
        else if(htarget(i,j) < 0) && (ktarget(i,j) < 0)
            hktarget_label(i+1,j+1) = 5;
        else if(htarget(i,j) == 0) && (ktarget(i,j) < 0)
            hktarget_label(i+1,j+1) = 6;
        else if(htarget(i,j) > 0) && (ktarget(i,j) == 0)
            hktarget_label(i+1,j+1) = 7;
        else if(htarget(i,j) < 0) && (ktarget(i,j) == 0)
            hktarget_label(i+1,j+1) = 8;
            else
```

```
            hktarget_label(i+1,j+1) = 9;
            end
            end
                        %end
            end
            end
            end
            end
        end
    end
end

%---------------------Histogram generation---------------------------------
source_surface_label_matrix = zeros(1,65536);
target_surface_label_matrix = zeros(1,65536);
for i = 1:256
    for j = 1:256
        source_surface_label_matrix(1,256*(i-1)+j) = hksource_label(i,j);
        target_surface_label_matrix(1,256*(i-1)+j) = hktarget_label(i+1,j+1);
    end
end

figure; hist(source_surface_label_matrix,9);title('Histogram for local surface primitive of
Source')
xlabel('Local Surface Label Assignment');
ylabel('No. of points');
figure; hist(target_surface_label_matrix,9);title('Histogram for local surface primitive of
Target')
xlabel('Local Surface Label Assignment');
ylabel('No. of points');
```

Matlab Code Implementation for Curvature Extraction

```
%------------Matlab Code for Curvature Extraction--------------------------

function [h,k] = curvature_extraction(input_image)

x = 1:1:256;
y = 1:1:256;
Z(1:256,1:256) = 0;
[X,Y] = meshgrid(x,y);
for i = 1:256
   for j = 1:256
       Z(i,j) = input_image(i,j,1);
   end
end
k = gcurvature(X,Y,Z);
h = mcurvature(X,Y,Z);

%--------------Matlab Code for Gaussian Curvature extraction---------------

function gc = gcurvature(x,y,z)

[xu,xv]     =   gradient(x);
[xuu,xuv]   =   gradient(xu);
[xvu,xvv]   =   gradient(xv);

[yu,yv]     =   gradient(y);
[yuu,yuv]   =   gradient(yu);
[yvu,yvv]   =   gradient(yv);

[zu,zv]     =   gradient(z);
[zuu,zuv]   =   gradient(zu);
[zvu,zvv]   =   gradient(zv);

for i=1:(size(z,1))
   for j=1:(size(z,2))
       Xu        =   [xu(i,j) yu(i,j) zu(i,j)];
       Xv        =   [xv(i,j) yv(i,j) zv(i,j)];
       Xuu       =   [xuu(i,j) yuu(i,j) zuu(i,j)];
       Xuv       =   [xuv(i,j) yuv(i,j) zuv(i,j)];
       Xvv       =   [xvv(i,j) yvv(i,j) zvv(i,j)];
       E         =   dot(Xu,Xu);
       F         =   dot(Xu,Xv);
       G         =   dot(Xv,Xv);
```

```
        m          =  cross(Xu,Xv);
        n          =  m/sqrt(sum(m.*m));
        L          =  dot(Xuu,n);
        M          =  dot(Xuv,n);
        N          =  dot(Xvv,n);
        gc(i,j)    =  ((L*N)-M^2)/((E*G)-F^2);
    end
end

%-----------Matlab Code for Mean Curvature Extraction----------------------

function gm = mcurvature(x,y,z)

[xu,xv]     =  gradient(x);
[xuu,xuv]   =  gradient(xu);
[xvu,xvv]   =  gradient(xv);

[yu,yv]     =  gradient(y);
[yuu,yuv]   =  gradient(yu);
[yvu,yvv]   =  gradient(yv);

[zu,zv]     =  gradient(z);
[zuu,zuv]   =  gradient(zu);
[zvu,zvv]   =  gradient(zv);

for i=1:(size(z,1))
    for j=1:(size(z,2))
        Xu         =  [xu(i,j) yu(i,j) zu(i,j)];
        Xv         =  [xv(i,j) yv(i,j) zv(i,j)];
        Xuu        =  [xuu(i,j) yuu(i,j) zuu(i,j)];
        Xuv        =  [xuv(i,j) yuv(i,j) zuv(i,j)];
        Xvv        =  [xvv(i,j) yvv(i,j) zvv(i,j)];
        E          =  dot(Xu,Xu);
        F          =  dot(Xu,Xv);
        G          =  dot(Xv,Xv);
        m          =  cross(Xu,Xv);
        n          =  m/sqrt(sum(m.*m));
        L          =  dot(Xuu,n);
        M          =  dot(Xuv,n);
        N          =  dot(Xvv,n);
        gm(i,j)    =  ((E*N)+(G*L)-(2*F*M))/(2*(E*G)-F^2);
    end
end
```

Matlab Code Implementation for Proposed method

```
%-----Matlab Code for Calling the Proposed method function-----------------

expsumon;

proposed_method;


%---Matlab Code implementation of Intensity Based Non-rigid Image Registration
(function: expsumon) --------------

a=imread('source1.png');
b=imread('floating2.png');
%aa=20*double(a);
%bb=20*double(b);
aa=double(a);
bb=double(b);
c=zeros(256+30);
for i=16:271
   for j=16:271
       c(i,j)=bb(i-15,j-15);
   end
end
cc=zeros(256+30);
for i=16:271
   for j=16:271
       cc(i,j)=b(i-15,j-15);
   end
end

z=zeros(961,65536);
for p=1:256
   for q=1:256
       yy=zeros(31);
   for ii=-15:15
   for jj=-15:15
       yy(ii+16,jj+16)=abs(aa(p,q)-c(ii+p+15,jj+q+15));
   end
   end
   mn=min(min(yy));
k=zeros(31);
for i=-15:15
   for j=-15:15
```

```
    if((abs(aa(p,q)-c(i+p+15,j+q+15))-mn)<=3)
       k(i+16,j+16)=floor(exp(abs(aa(p,q)-c(i+p+15,j+q+15))-mn));
    else
      k(i+16,j+16)=20;
    end


   end
end
%for i=1:31
 %  for j=1:31
  %     if(k(i,j)>=.13)
   %         k(i,j)=floor(3.*rand(1));
    %    else
     %        k(i,j)=7+floor(3.*rand(1));
      % end
    %end
%end
k=k';
k=k(:);
z(:,256*(p-1)+q)=k;
   end
end
B=ones(65536,1);
B(1,1)=0;
s=spdiags(B,1,65536,65536);
pp=zeros(961,961);
for i=1:961
   for j=1:961
       pp(i,j)=12*abs(i-j);
   end
end


h = GCO_Create(65536,961);
GCO_SetDataCost(h,z);
GCO_SetSmoothCost(h,pp);
GCO_SetNeighbors(h,s);
GCO_Expansion(h);
ss=GCO_GetLabeling(h);
[E D S]=GCO_ComputeEnergy(h)


ss=ss';
```

```
d=zeros(1,65536);
for p=1:256
   for q=1:256
l=zeros(31);
for i=-15:15
   for j=-15:15
      l(i+16,j+16)=cc(i+p+15,j+q+15);

   end
end
l=l';
l=l(:);
l=l';
 d(256*(p-1)+q)=l(ss(256*(p-1)+q));
   end
end
sumon=zeros(256);
matchedpixs = 0;
for i=1:256
   for j=1:256
      sumon(i,j)=d(((i-1)*256)+j);
      if(a(i,j) == sumon(i,j))
         matchedpixs = matchedpixs + 1;
      end
   end
end
sumon=uint8(sumon);
figure;imshow(sumon),title('Sumon Registration')
diff=abs(a-sumon);
m=mean(mean(diff))
sd=std(std(double(diff)))
imwrite(sumon,'sumon_registered2_trial.png');


%---curvature error computation between source and registered-----
[source_mean_curv,source_gaussian_curv] = curvature_extraction(aa);
[registered_mean_curv,registered_gaussian_curv] = curvature_extraction(sumon);

mean_curvature_difference_sumon = (abs(source_mean_curv -
registered_mean_curv)).^2;
gaussian_curvature_difference_sumon = (abs(source_gaussian_curv -
registered_gaussian_curv)).^2;
curvature_error_sumon = sqrt(mean_curvature_difference_sumon +
gaussian_curvature_difference_sumon);
```

```
mean_curvature_error_sumon = mean(mean(curvature_error_sumon))
std_sumon = std(std(curvature_error_sumon))


% ---------------Matlab Code for Curvature based Non-rigid Image Registration  -----------


a=imread('source1.png');
b=imread('sumon_registered2_trial.png');
aa=double(a);
bb=double(b);

c=zeros(256+2);
for i=2:257
   for j=2:257
      c(i,j)=bb(i-1,j-1);
   end
end

%----------------curvature computation of Source1-------------------------
[hsource,ksource] = curvature_extraction(aa);

%----------------curvature computation of Registered4-------------
[htarget,ktarget] = curvature_extraction(bb);

%----------------Curvature Difference Matrix Generation--------------------
hdiff = zeros(256);
kdiff = zeros(256);
sc = zeros(256);
for p=1:256
   for q=1:256
      hdiff(p,q) = abs( hsource(p,q) - htarget(p,q) );
      kdiff(p,q) = abs( ksource(p,q) - ktarget(p,q) );
      sc(p,q) = hdiff(p,q)*hdiff(p,q) + kdiff(p,q)*kdiff(p,q);  %developing matrix based
on curvature difference metric
   end
end
sc = sqrt(sc); %square root of curvature difference matrix
meanval = mean(mean(sc));

%------------Thresolding of Curvature Difference Matrix with Mean-------------------------
-
upper_threshold = meanval + 10; %setting upper threshold
```

```
% th_l = meanval - 5; %setting lower threshold
correctpix = 0;
incorrectpix = 0;
matchedpix = 0;
for p =1:256
    for q =1:256
        if(sc(p,q) <= upper_threshold)
            correctpix = correctpix + 1;
        else
            incorrectpix = incorrectpix + 1;
            sc(p,q) = 0;    %these upper threshold pixels are discarded.
        end
        if(sc(p,q) == 0)
            matchedpix = matchedpix + 1;  % counting pixels matched with curvature
        end
    end
end

%------------------------Labeling of Source according to Curvature Class---
hksource_label = zeros(256);
for i = 1:256
    for j = 1:256
        if(hsource(i,j) > 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 1;
        elseif(hsource(i,j) < 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 2;
        elseif(hsource(i,j) == 0) && (ksource(i,j) > 0)
            hksource_label(i,j) = 3;
        elseif(hsource(i,j) > 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 4;
        elseif(hsource(i,j) < 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 5;
        elseif(hsource(i,j) == 0) && (ksource(i,j) < 0)
            hksource_label(i,j) = 6;
        elseif(hsource(i,j) > 0) && (ksource(i,j) == 0)
            hksource_label(i,j) = 7;
        elseif(hsource(i,j) < 0) && (ksource(i,j) == 0)
            hksource_label(i,j) = 8;
            else
            hksource_label(i,j) = 9;
        end
    end
end
%------------------------Labeling of Target according to Curvature Class---
```

```
hktarget_label = zeros(256+2);
for i = 1:256
   for j = 1:256
      if(htarget(i,j) > 0) && (ktarget(i,j) > 0)
         hktarget_label(i+1,j+1) = 1;
      elseif(htarget(i,j) < 0) && (ktarget(i,j) > 0)
         hktarget_label(i+1,j+1) = 2;
      elseif(htarget(i,j) == 0) && (ktarget(i,j) > 0)
         hktarget_label(i+1,j+1) = 3;
      elseif(htarget(i,j) > 0) && (ktarget(i,j) < 0)
         hktarget_label(i+1,j+1) = 4;
      elseif(htarget(i,j) < 0) && (ktarget(i,j) < 0)
         hktarget_label(i+1,j+1) = 5;
      elseif(htarget(i,j) == 0) && (ktarget(i,j) < 0)
         hktarget_label(i+1,j+1) = 6;
      elseif(htarget(i,j) > 0) && (ktarget(i,j) == 0)
         hktarget_label(i+1,j+1) = 7;
      elseif(htarget(i,j) < 0) && (ktarget(i,j) == 0)
         hktarget_label(i+1,j+1) = 8;
         else
         hktarget_label(i+1,j+1) = 9;
      end
   end
end


%----------------------Local displacement matrix-------------------------
neighbourhood_size = 9; % Local neighbourhood size = 9
hsource_copy = zeros(256+2);
ksource_copy = zeros(256+2);

local_neighbour_index = sqrt(neighbourhood_size); % local neighbourhood window is 3
by 3 matrix
local_curv_diff_map = zeros(local_neighbour_index);

for p = 1:156
   for q = 1:256
      hsource_copy(p+1,q+1) = hsource(p,q);   % h values of source are copied into
hsource_copy
      ksource_copy(p+1,q+1) = ksource(p,q);   % k values of source are copied into
ksource_copy
   end
end
```

```
displacement_row_position = zeros(1,65536);
displacement_col_position = zeros(1,65536);
aligned_pix = 0;

for p = 1:256
   for q = 1:256
      if(hktarget_label(p,q) ~= hksource_label(p,q))
         if(sc(p,q) ~= 0) % fine tuning i.e curvature alignment is done for unmatched pixels
         for i = -1:1
            for j = -1:1  % building the local curvature difference map of 3_by_3 matrix in the neighbourhood for each pixel in target image
               local_curv_diff_map(i+2,j+2) = sqrt( (hsource_copy(p+i+1,q+j+1)-htarget(p,q)).^2 + (ksource_copy(p+i+1,q+j+1)-ktarget(p,q)).^2 );
            end
         end
         min_local_diff = min(min(local_curv_diff_map)); % calculating minimum curvature difference in the neighbourhood of source pixel

         if(local_curv_diff_map(2,2) == min_local_diff) % checking whether corresponding points are having minimum curvature
            displacement_row_position(1,256*(p-1)+q) = i; % no displacement of those points having min curvature difference with their corresponding points
            displacement_col_position(1,256*(p-1)+q) = j;
            matchedpix = matchedpix + 1; % counting pixels matched with minimum curvature difference
         else
            for i = -1:1
               for j = -1:1
                  if(local_curv_diff_map(i+2,j+2) == min_local_diff) % searching for the position in the neighbourhood where the pixel finds minimum curvature difference
                     displacement_row_position(1,256*(p-1)+q) = i; % book keeping of pixel positions in the neighbourhood with minimum curvature difference
                     displacement_col_position(1,256*(p-1)+q) = j;
                     aligned_pix = aligned_pix + 1;
                  end
               end
            end
         end
         end
      end
   end
end
```

```
final_matched_pix = 0;
final_registration = zeros(256);
for p = 1:256
   for q = 1:256
      final_registration(p,q) = c( p+1+displacement_row_position(1,256*(p-1)+q),
q+1+displacement_col_position(1,256*(p-1)+q)) ;
   end
end

final_registration=uint8(final_registration);
imwrite(final_registration,'final_curvature_register2_trial.png');
sc = curvature_difference_metric_function;
for p = 1:256
   for q = 1:256
      if(sc(p,q) == 0)
         final_matched_pix = final_matched_pix + 1;  % total no of matched pixels after
this registration
      end
   end
end
figure;imshow(final_registration),title('Proposed Registration')

%----------------Calculate curvature error between source and registered
%image---

[registered_mean_curv,registered_gaussian_curv] =
curvature_extraction(final_registration);

mean_curvature_difference_proposed = (abs(hsource - registered_mean_curv)).^2;
gaussian_curvature_difference_proposed = (abs(ksource - registered_gaussian_curv)).^2;
curvature_error_proposed = sqrt(mean_curvature_difference_proposed +
gaussian_curvature_difference_proposed);

mean_curvature_error_proposed = mean(mean(curvature_error_proposed))
std_proposed = std(std(curvature_error_proposed))
```