Chapter 1

INTRODUCTION

1.1 IMAGE REGISTRATION

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It geometrically aligns two images—the reference and sensed images. The present differences between images are introduced due to different imaging conditions. Image registration is a crucial step in all image analysis tasks in which the final information is gained from the combination of various data sources like in image fusion, change detection, and multichannel image restoration. Typically, registration is required in remote sensing (multispectral classification, environmental monitoring, change detection, image mosaicking, weather forecasting, creating super-resolution images, integrating information into geographic information systems (GIS)), in medicine (combining computer tomography (CT) and NMR data to obtain more complete information about the patient, monitoring tumour growth, treatment verification, comparison of the patient's data with anatomical atlases), in cartography (map updating), and in computer vision (target localization, automatic quality control), to name a few. Registration is necessary in order to be able to compare or integrate the data obtained from these different measurements.

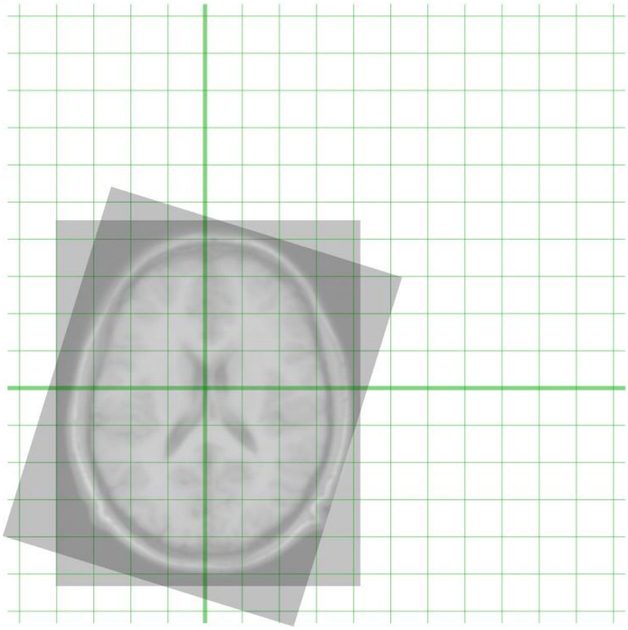

Fig 1.1: Registration of two MRI images of the brain

Many number of algorithms are designed and developed based on the application requirements of image registration techniques which includes mathematical models also. Image registration algorithms are broadly divided into ten classifications:

- Intensity-based vs feature-based
- Transformation models
- Transformations of coordinates via the law of function composition rather than addition
- Spatial vs frequency domain methods
- Mono-modality vs multi-modality methods
- Automatic vs interactive methods
- Similarity measures for image registration

1.Intensity-based vs feature-based:

Image alignment algorithms of image registration process are classified into intensity-based and feature-based. In this process, one of the images is referred to as the reference or source, and the other respective images are referred as the target or sensed or subject images. The image registration is obtained through spatially registering the target images to align with the reference images. In the intensity-based methods, it is compared intensity patterns in images through correlation metrics.

Intensity-based methods register entire images or sub-images. If sub-images are registered, centres of corresponding sub images are treated as corresponding feature points.

In feature-based methods, finds the correspondence between image features such as points, lines, and contours. Feature-based methods establish a correspondence between a number of distinct points in images especially. Knowing the correspondence between a number of points in images, a geometrical transformation is then determined to map the target image to the reference images, which indicates point-by-point correspondence is established between the reference and target images.

2. Transformation models:

The transformation model defines how one image can be deformed to match another; it characterizes the type and number of possible deformations. The most well-known example is the rigid or affine transformation that can be described very compactly by between 6 (3 translations and 3 rotations) and 12 (6 + 3 scaling's + 3 shears) parameters for a whole image. These parameters are applied to a vector locating a point in an image to find its location in another image.

The transformation model serves two purposes; first it controls how image features can be moved relative to one another to improve the image similarity and second it interpolates between those features where there is no useable information. Transformations used in non-rigid registration range from smooth regional variation described by a small number of parameters to dense displacement fields defined at each voxel.

3. Spatial vs frequency domain methods:

Spatial methods operate in the frequency domain for matching intensity features of images patterns. The algorithms developed for feature matching is a superficial to traditional techniques of image registration.

When the number of control points exceeds the minimum required to define the appropriate transformation model, iterative Random sample consensus algorithm, and be used to robustly estimate the parameters for registration of the images.

The algorithms developed to find the spatial transformation parameters between source and target images in the image registration process falls into spatial frequency domain models. These models use frequency transformations such as translation, rotation, and scaling in frequency domain.

The phase correlation method is used to a pair of images in order to produce a third image which contains a single peak in the frequency domain. The location of this peak corresponds to the relative translation between source and target images. The phase correlation method is resilient to noise, occlusions, and other defects typical of medical or satellite images, unlike many spatial domain algorithms.

In addition, the phase correlation uses the fast fourier transform methodology to compute the cross-correlation between the two images that generally results large performance gains. This method is extended to determine rotation and scaling differences between two images by first converting the images into log-polar coordinates. Due to efficient properties of Fourier transform, the rotation and scaling parameters are determined in a manner invariant to translation.

4. Mono-modality and multi-modality:

Another classification is made between mono-modality and multi-modality methods. In single modality model the images are acquired by the same scanner/sensor, while in multi-modality registration methods the images acquired by different scanners/sensors and/or at different times.

Multi-modality registration methods have many applications in medical image analysis. The medical images that are obtained from dissimilar scanners are used for medical diagnosis. The examples include registration of brain CT/MRI images or whole body PET/CT images for a tumour localization or defect in any other human body/part diagnosis purpose. The registration of contrast-enhanced versus non-contrast-enhanced in CT images for segmentation of specific parts of the anatomy, and registration of ultrasound and CT images for prostate localization in radiotherapy is another application.

5.Automatic vs interactive methods:

The algorithms developed to provide the level of automation are classified into (i) manual, (ii) semiautomatic or interactive and (iii) automatic models of image registration methods. The tools developed for manual alignment of source and target images are called as manual models. Semi-automatic or interactive models needs user to verify the correspondence of registration. Interactive methods reduce user bias by performing certain key operations automatically and help the user to guide for the registration. Semi-automatic methods perform more of the registration steps automatically but depend on the user to verify the correctness of image registration process. Algorithms that are developed for not to allow any user interaction of any sort, and performs all registration steps automatically.

6. Similarity measures for image registration:

Intensity approaches match intensity patterns in each image using mathematical or statistical criteria. They define a measure of intensity similarity between the source and the target and adjust the transformation until the similarity measure is maximized. They assume that the images will be most similar at the correct registration.

Measures of similarity have included squared differences in intensities, correlation coefficient, measures based on optical flow, and information-theoretic measures such as mutual information. The simplest similarity measure is the sum of squared differences, which assumes that the images are identical at registration except for (Gaussian) noise.

The correlation coefficient assumes that corresponding intensities in the images have a linear relationship. These two similarity measures are suitable for mono-modal registration where the intensity characteristics are very similar in the images.

For multi-modal registration, similarity measures have been developed, which define weaker relationships between intensities to reflect the different intensity characteristics of different imaging modalities.

The correlation ratio assumes that corresponding intensities are functionally related at registration and information-theoretic measures like mutual information assume only that a probabilistic relationship between voxel intensities is maximized at registration.

Voxel similarity measure	Comment
Sum of Squared Differences $\mathrm{SSD} = \frac{1}{N}\sum_{X}(T(\mathbf{x}) - S(t(\mathbf{x})))^2$	Registered images differ only by Gaussian noise. Sensitive to small number of voxels that have very large intensity differences. Only for mono-modal image registration
Correlation coefficient $\mathrm{CC} = \frac{\sum_{\mathbf{x}}(T(\mathbf{x}) - \bar{T})\cdot(S(t(\mathbf{x})) - \bar{S})}{\sqrt{\sum_{\mathbf{x}}(T(\mathbf{x}) - \bar{T})^2 \cdot \sum_{\mathbf{x}}(S(t(\mathbf{x})) - \bar{S})^2}}$	Registered images have linear intensity relationship and objects of interest are in the field of view of both images. Segmentation of interesting features often necessary. Only for single-modal image registration
Correlation ratio $\eta = 1 - \frac{1}{N\sigma^2}\sum_i N_i \sigma_i^2$	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 the target image $\sigma^2 = \frac{1}{N}\sum_{\mathrm{overlap}\,\mathrm{x}} S(\mathbf{x})^2 - m^2, m = \frac{1}{N}\sum_{\mathrm{overlap}\,\mathrm{x}} S(\mathbf{x})$ $\sigma_{\mathrm{i}}^2 = \frac{1}{N_i}\sum_{\mathbf{x}:T(\mathbf{x})=i} S(\mathbf{x})^2 - m_i^2, m_i = \frac{1}{N_i}\sum_{\mathbf{x}:T(\mathbf{x})=i} S(\mathbf{x})$
Mutual information $\mathrm{MI} = H_T + H_S - H_{TS}$	Assumes only a probabilistic relationship between intensities. Defined in terms of entropies of the intensity distribution $H_T = -\sum_i P_i \log P_i, H_S = -\sum_j Q_j \log Q_j$ and $H_{TS} = -\sum_{i,j} p_{ij} \log p_{ij}$ where P (Q)=probability of intensity I (J) occurring in target (source) and p_{ij}=joint probability of both occurring at the same place
Normalized mutual information $\mathrm{NMI} = \frac{H_T + H_S}{H_{TS}}$	Proposed to minimize the overlap problem seen occasionally with mutual information

Fig 1.2: Common image similarity measures used in registration.

- Here 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

1.2 MUTUAL INFORMATION

Mutual information is an information theory measure of the statistical dependence between two random variables or the amount of information that one variable contains about the other. It can be qualitatively considered as a measure of how well one image explains the other. The most commonly used measure of information in image processing is the Shannon Wiener entropy measure. Given m events occurring with probabilities p1…….pn the Shannon entropy is defined as

$$H = -\sum_{i=1}^{m} P_i \log P_i \qquad \text{Eq 1.1}$$

It is a measure of uncertainty or dispersion of the probabilities of events. For an image the entropy is calculated from the image intensity histogram in which the probabilities are the histogram entries.

It will have a maximum value if all symbols have equal probability of occurring, minimum value of zero if the probability of one symbol occurring is 1 and the probability of all the others occurring is zero. In image registration since there are two images joint entropy will have to be also considered. Joint entropy measures the amount of information we have in the two images combined. The Joint entropy H (I, J) can be calculated using the joint histogram of two images. If the images are totally unrelated, then the joint entropy will be the sum of the entropies of the individual images.

The more similar the images are, the lower the joint entropy compared with the sum of the individual entropies.

$$H(A, B) \leq H(A) + H(B) \qquad \text{Eq 1.2}$$

As the images become misaligned, dispersion of their joint histogram increases. Therefore, registration of two images can be accomplished by minimizing the joint entropy of the images, but mutual information is a better criterion as marginal entropies H (I) and H (J) are taken into account.

$$MI(A,B) = H(A) + H(B) - H(A,B) \qquad \text{Eq 1.3}$$

The optimal transformation can be gained by maximizing mutual information of the two images. So if the images are of the same object, when they are correctly registered, corresponding pixels in the two images will be of the same anatomical or pathological structure. Normalized measure of mutual information is defined as follows:

$$NMI(A, B) = \sqrt{(2-(2*H(A,B)/(H(A)+H(B)))} \qquad \text{Eq 1.4}$$

Normalized mutual information has been shown to be more robust for intermodality registration than standard mutual information.

1.3 GENETIC ALGORITHM

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions.

Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

1.4 APPLICATIONS OF IMAGE REGISTRATION

1. Computer vision: Computer vision is a field that includes methods for acquiring, processing, analysing, and understanding images. Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception.

2. Medical Imaging: Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention.

Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities.

3. Automatic Target Detection: Automatic target recognition (ATR), is the ability for an algorithm or device to recognize targets or objects based on data obtained from sensors. Target recognition was initially done by using an audible representation of the received signal, where a trained operator who would decipher that sound to classify the target illuminated by the radar.

1.5 OBJECTIVES OF THE PROJECT

The Project is mainly focused on image registration of non-rigid images and is to determine an optimal transformation in such a way that the transformed template image becomes similar to the reference image as much as possible. To align two test images after performing operations such as translation, rotation, scaling and shearing for non-rigid objects and identify the maximum mutual information between the images.

We use pixel based method for the project where first we find the entropy of the image then find the histogram, after which we find the joint histogram, then we find the entropy after which we use that to find the normalized mutual information.

- To align two test images after performing operations such as translation for rigid objects.
- For non-rigid objects we implement algorithms that learn by themselves and are able to identify distinct patterns in the images.

1.6 METHODOLOGY

1. Feature detection: Pixel’s intensity is taken as a parameter to find the normalised mutual information which serves as a correlation metric to find the similarity/dissimilarity between the source and template images.

2. Feature matching: A genetic algorithm is implemented to simulate a set of operations on the template image to find the maximum normalised mutual information.

Each operation has a different set of values and thus, we get different transformations each of which have a different normalised mutual information value.

3. Transform model estimation: The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated. The parameters of the mapping functions are computed by means of the established feature correspondence.

4. Image re-sampling and transformation: The sensed image is transformed by means of the mapping functions. Image values in non-integer coordinates are computed by the appropriate interpolation technique.

Chapter 2

LITERATURE SURVEY

Image registration is an active field of research in image processing. Many methods can still be considered for image processing in general and image analysis in particular. The need and requirements in all fields is unimaginable provided that the designed algorithms & methods are validated properly.

- **"Medical Image Registration based on SURF detector"; P. V. Lukashevich, B. A. Zalesky, and S. V. Ablameyko, 2011,**

  Registration in the area of interest is accomplished by searching for a promising transformation, which would combine the points of interest areas of the two images in the best way possible. To calculate the optimum perspective transformation, the method of registering based on a SURF detector, which has proved its validity, is used.

  Drawbacks:The rate of the implementation of the method is quite high even though it is performed using a script interpreter.

- **Roshini V S, DR K Revathy, Using Normalized Mutual Information and JointEntropy as Metrics for Registration of Images, pp 2005-2008,**

  Image similarity measures include Cross-Correlation, Mutual Information, Mean-square difference and Ratio Image Uniformity. Mutual information is an information theory measure of the statistical dependence between two random variables or the amount of information that one variable contains about the other. It can be qualitatively considered as a measure of how well one image explains the other. Registration of two images can be accomplished by minimizing the joint entropy of the images, but mutual information is a better criterion.

  Drawbacks:Some multistate methods and generic algorithms, may prove to be less suitable for optimization of the mutual information measure, because they can move outside the capture range.

- **"Image registration methods: a survey", Barbara Zitova, Jan Flusser, ,2003,**
  A survey of the classical and up-to-date registration methods, classifying them according to their nature as well as according to the four major registration steps.

  Drawbacks:Registration of images with complex nonlinear and local distortions, multimodal registration, and belong to the most challenging tasks.

- **"Classification of Image Registration Techniques and Algorithms in Digital Image Processing – A Research Survey", SindhuMadhuri G, 2014,**
  This survey emphasizes Image Registration as the most essential part of panoramic image generation & creation, where applications and uses are unimaginable for researchers longing to invent & implement alternative image registration methods from general to specific to complex applications.

  Drawbacks: Although the proper verification methods are known in most cases, and coarsely laid for most applications the painstaking work of conducting the many experiments involved is only now starting due to its computational complexity.

- **"Non-rigid image registration: theory and practice", W R CRUM, DPhil, T HARTKENS, PhD and D L G HILL, PhD, 2004,**
  The current state of the art of non-rigid registration to put on-going research in context and to highlight current and future clinical applications that might benefit from this technology. The philosophy and motivation underlying non-rigid registration is discussed and a guide to common terminology is presented. The core components of registration systems are described and outstanding issues of validity and validation are confronted.

  Drawbacks:The most sophisticated non-rigid registration algorithms frequently take many hours to register images, which makes them unsuitable for interactive use on image analysis workstations or on scanner consoles.

Chapter 3

NEED FOR THE SYSTEM

3.1 EXISTING SYSTEM

- **NiftyReg:** This project, developed at University college London, contains programs to perform rigid, affine and non-linear registration of nifty or analyse images. Two versions of the algorithms are included, a CPU- and a GPU- (using CUDA) based implementation.

- **Elastix:** Elastix is open source software, based on the well-known insight segmentation and registration toolkit(ITK). The software consists of a collection of algorithms that are commonly used to solve(medical) image registration problems. The modular design of elastix allows the user to quickly configure, test, and compare different registration methods for a specific application. A command-line interface enables automated processing of large numbers of data sets, by means of scripting.

- **ANTS:** The ANTS package is designed to enable researchers with advanced tools for brain and image mapping. Many of the ANTS registration tools are diffeomorphic, but deformation transformations are available. Unique components of ANTS include multivariate similarity metrics, landmark guidance, the ability to use label images to guide the mapping and both greedy and space-time optimal implementations of diffeomorphisms. The symmetric normalization (SyN) strategy is a part of the ANTS toolkit as it directly manipulated free form deformation(DMFFD).

3.2 PROPOSED SYSTEM

An overview of proposed system:

The goal of this project is to align two images based on the mutual information that we use as a correlation metric and we iterate the process using a genetic algorithm to yield maximum normalized mutual information

To achieve this, we use the following steps:

- Accept the two image path and validate them
- Pass them as parameters to align.py module and convert them to grayscale
- Find the initial NMI(Normalized Mutual Information) between the two images
- Perform a set of 10 operations on the template image and find the NMI in each case to produce 10 children
- Find the highest NMI among them, perform the transformation on the template image and consider it as a parent for the next iteration. The values of translation and rotation are noted each time a transformation is noted.
- Repeat this for several iterations until maximum NMI is obtained and finally display the aligned image on a separate window along with the values of translation and rotation that were done to the template image to serve as an offset to be registered with the source image.

Step one: Accept file names and validate them:

The very first step is to accept the file location of the two modules and verify if they exist. If they do, they are passed on to the **'align.py'**otherwise an error is prompted.

Below is the code for the test case and the alerts associated with them.

```
try:
        subprocess.call([sys.executable, 'align.py', fileName1,fileName2])
exceptNameError as e:
        QtGui.QMessageBox.information(self, "Image Registration","Please select a file
        first.")
else:
        return
```

```
if fileName2:
	image = QtGui.QImage(fileName2)
ifimage.isNull():
	QtGui.QMessageBox.information(self, "Image Registration","Please load an image file.")
return
```

Step two: Pass them as parameters to the 'align.py' module and convert them to grayscale:

We convert our image to grayscale in order to make our results more accurate as we will only look at the greyscale entities and not the RGB thus, significantly ignoring the noise that is the colours present in each image. Rather we only see the intensity values of each pixel. Below we see the code snippets to convert each to grayscale after opening the images.

```
image1 = cv2.imread(sys.argv[1],0)
image2 = cv2.imread(sys.argv[2],0)
```

Step three: Find the initial NMI:

We invoke the following function in order to find the NMI between two images.

```
defcalc_entropy(image):
	entropy = 0
	size = image.size
	hist = cv2.calcHist([image],[0],None,[16],[0,256])
	fori in range(0,15):
		p = hist[i]/size
		if p!=0:
			entropy-=p*math.log(p,2)
	return entropy
```

```
defcalc_joint_histogram(image1,image2):
	joint_entropy = 0
	joint_mat = np.ndarray((16,16))
	joint_mat.fill(0)
	size = image1.size
	fori in range(0,image1.shape[0]):
		for j in range(0,image1.shape[1]):
			inten1 = image1[i][j]/16
			inten2 = image2[i][j]/16
			joint_mat[inten1][inten2]+=1
	returnjoint_mat,size

defcalc_joint_entropy(joint_mat,size):
	joint_entropy = 0
	fori in range(0,15):
		for j in range(0,15):
			p = joint_mat[i][j]/size
			if p!=0:
				joint_entropy-=p*math.log(p,2)
	returnjoint_entropy

defcalc_mi(image1,image2):
	entropy1 = calc_entropy(image1)
	entropy2 = calc_entropy(image2)
	joint_mat,size = calc_joint_histogram(image1,image2)
	joint_entropy = calc_joint_entropy(joint_mat,size)
	mi = math.sqrt(2-(2*joint_entropy/(entropy1+entropy2)))
	return mi
```

Step four: Perform a set of 10 operations on the template image:

Using a series of random functions, we assign 10 set of random values initially for translation and rotation whose values we pass as parameters to each operation. In which the following code snippet, we observe that each operation has values i.e., (x, y, theta) which serve as parameter to perform basic set of operations on the image.

After the first iteration, we perform 10 more operations which are offset to each of the (x, y, theta) values. In each case, we use the cal_mi() function to find the NMI after each operation.

```
parent[1][0]=randint(5,10)
parent[1][1]=0
parent[1][2]=0
parent[2][0]=randint(-10,-5)
parent[2][1]=0
parent[2][2]=0
parent[3][0]=randint(-10,-5)
parent[3][1]=randint(-10,-5)
parent[3][2]=randint(-10,-5)
parent[4][0]=randint(5,10)
parent[4][1]=randint(5,10)
parent[4][2]=randint(5,10)
parent[5][0]=randint(-10,-5)
parent[5][1]=randint(-10,-5)
parent[5][2]=0
parent[6][0]=randint(5,10)
parent[6][1]=randint(5,10)
parent[6][2]=0
parent[7][0]=randint(-10,-5)
parent[7][1]=randint(5,10)
parent[7][2]=0
parent[8][0]=randint(5,10)
parent[8][1]=randint(-10,-5)
parent[8][2]=0
parent[9][0]=0
```

```
parent[9][1]=0
parent[9][2]=randint(-10,-5)
parent[0][0]=0
parent[0][1]=0
parent[0][2]=randint(5,10)
```

Step five: Find the highest NMI among the 10 children and consider the child with the highest NMI as parent for the next iteration:

Observe the following code snippet, we see that unless the NMI obtained from an operation is not greater than our NMI, we discussed the operation. Otherwise, we take into consideration the operation and transform the template image which becomes more and more similar to the source image.

```
iftemp_mi<mi:
        image3 = calc_rotation(image2,parent[i][2],cols,rows)
temp_mi = calc_mi(image1,image3)
iftemp_mi>mi:
mi = temp_mi
final_image = image3
final_x = parent[i][0]
final_y = parent[i][1]
final_angle = parent[i][2]
else:
mi = temp_mi
final_image = image3
final_x = parent[i][0]
final_y = parent[i][1]
final_angle = parent[i][2]
        image3 = calc_rotation(image3,parent[i][2],cols,rows)
temp_mi = calc_mi(image1,image3)
```

```
iftemp_mi>mi:
mi = temp_mi
final_image = image3
final_x+= parent[i][0]
final_y+= parent[i][1]
final_angle+= parent[i][2]
```

Step six: Repeat until maximum NMI is obtained:

Depending on the level of accuracy you are looking for adjust the value of the variable ‘limit’. But setting the value to a high number, does not yield a significant change in NMI and it is also time consuming as each iteration does 10 operations. Finally display the aligned image after using absdiff() to find the difference between the source and newly transformed image in a separate window and the values of NMI offset values of translation and rotation i.e.,(x, y) and theta.

Chapter 4

SYSTEM REQUIREMENTS
AND
SPECIFICATION

4.1 HARDWARE RQUIREMENTS

- Intel Core 2 Duo Processor
- CPU minimum 1.5 GHZ
- Minimum 2 GB of RAM
- Mouse
- Keyboard

4.2 SOFTWARE REQUIREMENTS

- Operating system: Ubuntu
- OpenCV on Python platform
- Qt4 on Python Platform for GUI

4.3 FUNCTIONAL REQUIREMENTS

- The goal of using genetic algorithm is to obtain a better solution in less number of iterations compare to any other brute force method.
- The normalised mutual information value of the two images is expected to be high to get a better alignment of the images.
- For a best solution, the normalised mutual information value should be 1.
- The system must be compliant with all types of readable images which may have RGB or gray scale.
- The system must be capable of displaying an output or an appropriate message for all possible set of images
- The system should be invariant to the change in colour of the images

- The system must provide information about the offset values that transform the template image to overlay closely with the source image.

4.4 NON-FUNCTIONAL REQUIREMENTS

- The system should get accurate and efficient solution by performing less number of iterations.
- The approach of registration and alignment should be cost efficient since no additional hardware and software are required.
- The system must be self-explanatory and must appeal to all users from beginners to experts
- The system must be robust so as to report errors when sufficient input is not met.
- The system should not be configured every time a new session begins.

Chapter 5

PROJECT ACTIVITY CHART

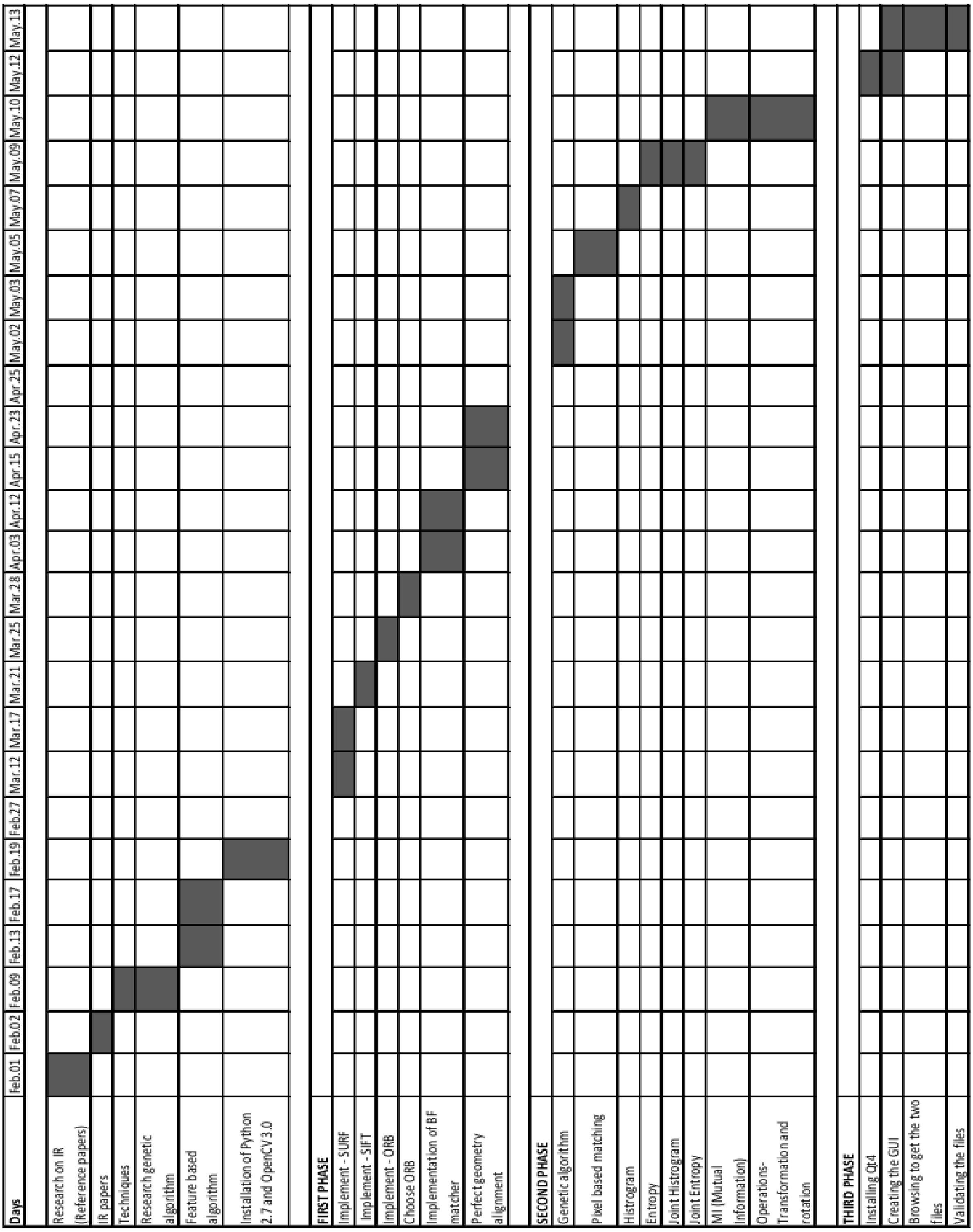

Fig 5.1: Gantt chart

Chapter 6

SYSTEM DESIGN

6.1 HIGH LEVEL DESIGN

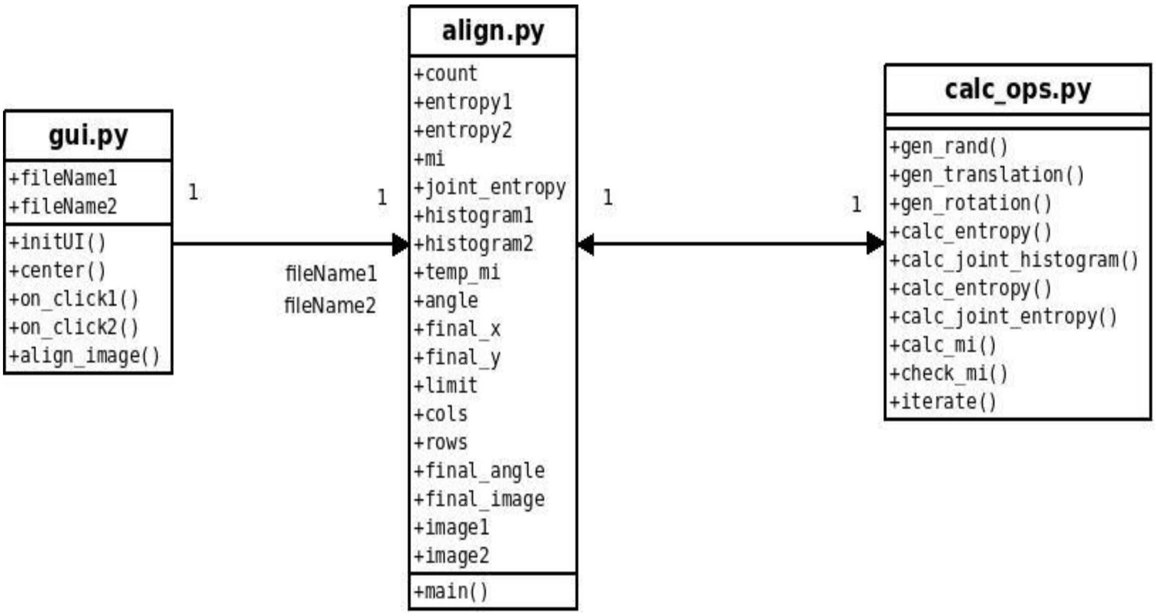

Fig 6.1: UML Class Diagram

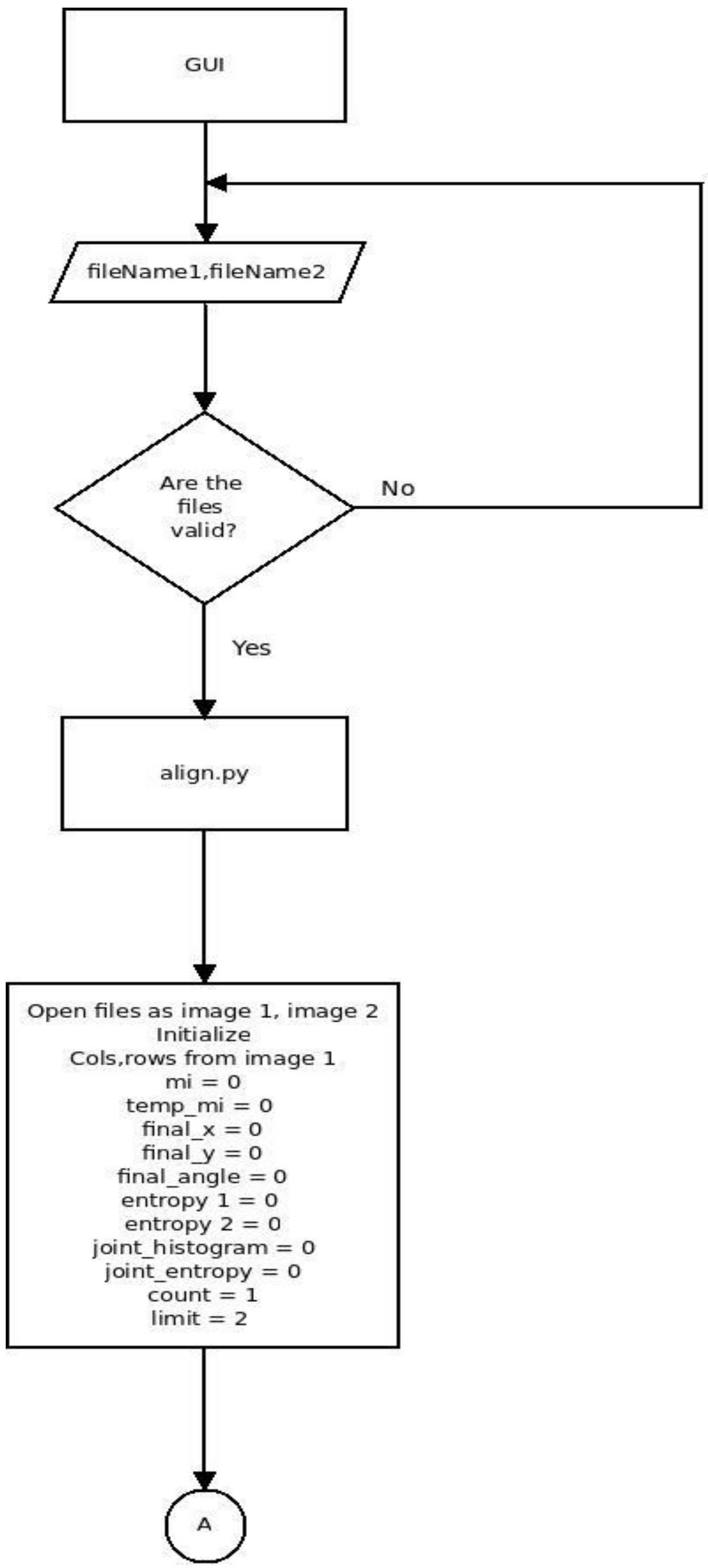

Fig 6.2: Flowchart

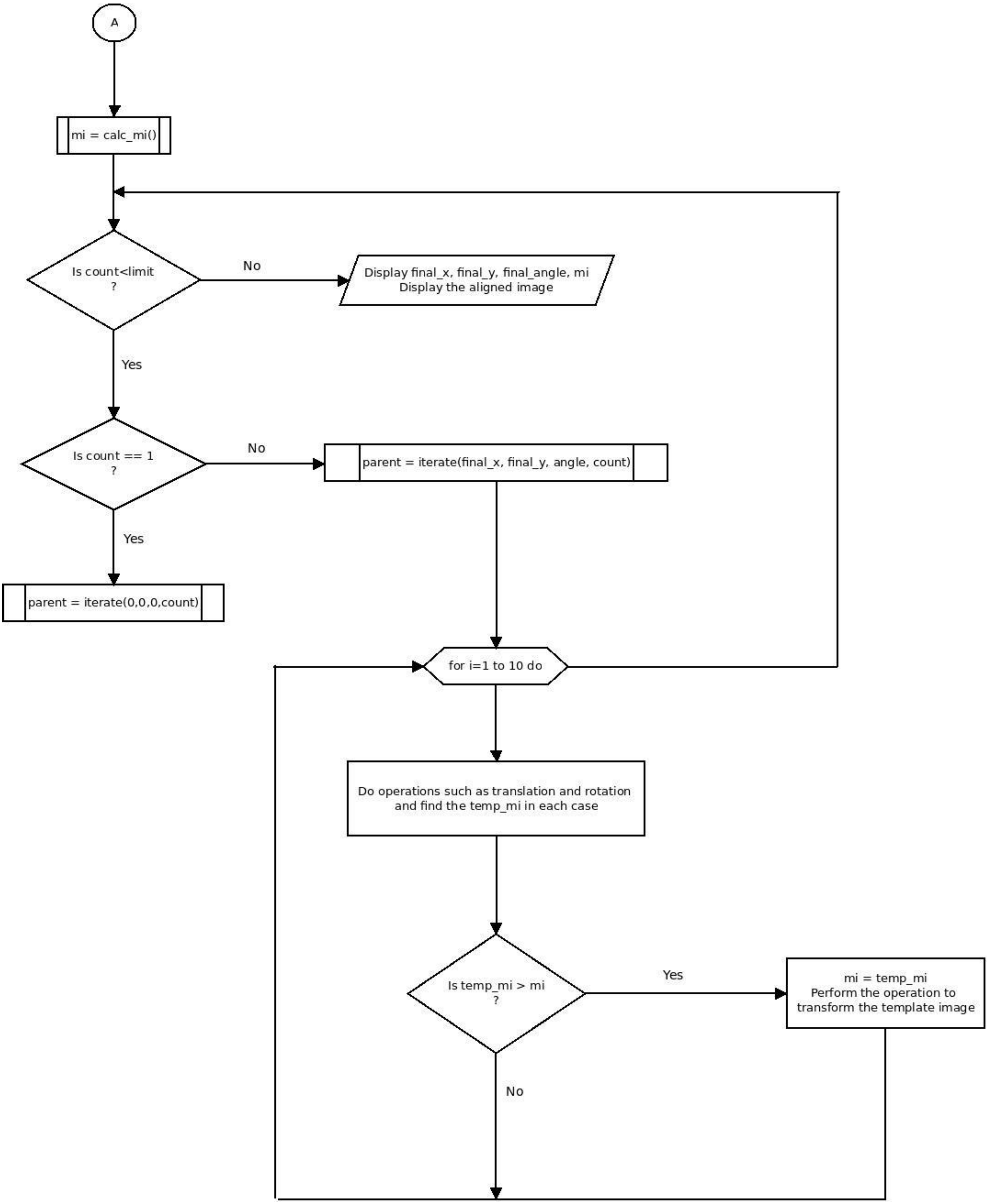

Fig 6.3: Flowchart continued

6.2 LOW LEVEL DESIGN

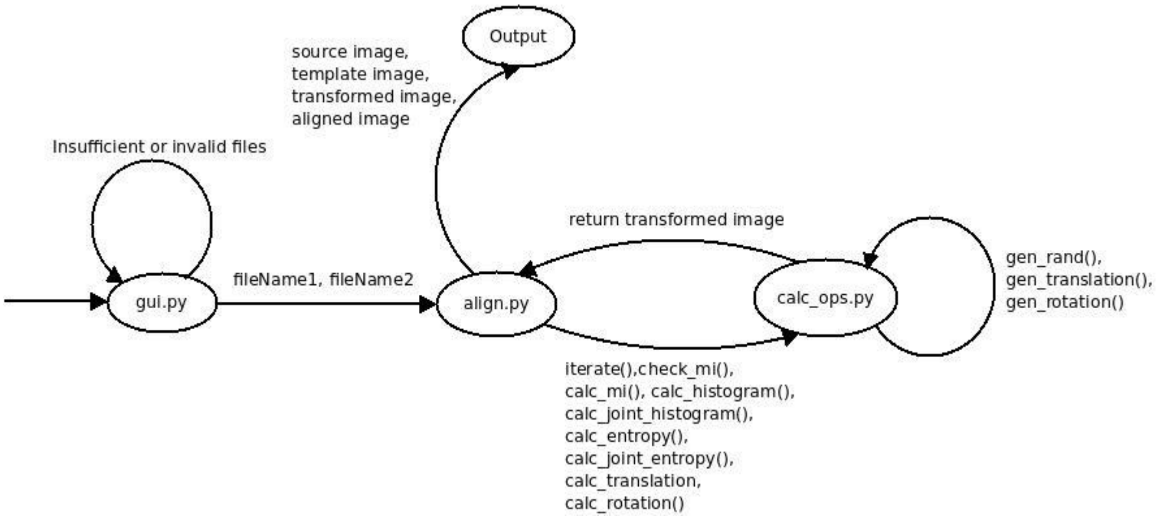

Fig 6.4: Low Level Design

The above diagram is the data flow diagram of the system it shows as to how the transition takes place from one module to another.

Chapter 7

SYSTEM IMPLEMENTATION

7.1 TOOLS USED

Operating System

Ubuntu: It is a software that manages computer hardware and software resources and provides common services for computer system. Ubuntu is a debian -based open source (with exceptions) Linux operating system and distribution.

Libraries

A library is a collection of precompiled routines that a program can use during software development. These routines, also called modules are stored in objects format and is included within programs whenever required.

OpenCV (Open Source Computer Vision Library): It is an open source machine learning software library. The library comprises of more than 2500 optimized algorithms, which includes identifying objects,stitching images together, grayscaling and binarization, finding contours and segmenting an image etc. It has C++, python,java and MATLAB interfaces and support Windows, Linux,Android and Mac OS.

Numpy: Numpy is a fundamental package for scientific computing using Python. It contains a powerful N-dimensional array object.Efficient multi-dimensional container of generic data of arbitrary data types.Useful linear algebra, fourier transform and random number capabilities on arrays.

Random: This module implements pseudo-random number generators for various distributions. Almost all module functions depend on the basic function random(), which generates a random float uniformly in the semi-open range [0.0, 1.0).

Math: This module is always available. It provides access to the mathematical functions defined by the C standard.These functions cannot be used with complex numbers; use the functions of the same name from the cmath module if you require support for complex numbers.

Language

Python:A programming language is a formal constructed language designed to communicate instructions to a computer. Python is high-level, interpreted, dynamic programming language.Python features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Text Editor

Atom: Atom is a tool that can be used to customize anything but also use productively with features like builtin package manager, smart auto completion, file system browser, multi panes apart from the features in any other editor.

gedit: is the default text editor of the GNOME desktop environment and part of the GNOME Core Applications. Designed as a general-purpose text editor, gedit emphasizes simplicity and ease of use, with a clean and simpleGUI, according to the philosophy of the GNOME project. It includes tools for editing source code and structured text such asmarkup languages.

7.2 MODULE WISE DESCRIPTION

7.2.1 Align Module

align.py:

Input:The source and template image paths.

Output:The aligned image

Description:This module is the core component of the project. Based on the input two image pathsthat we obtained, we perform crucial operations such as determining the shape,i.e.,the height and width of the two images, scaling them if they are not equal. We alsofind their histograms, joint histograms, entropy, joint entropy and finally the mutual informationbetween them.

gen_rand():

Input:A minimum and maximum value both of which are integers.

Output:A random integer.

Usage:Used to generate a random integer between the minimum and maximum value specified.

gen_translation():

Input:N/A

Output:Two random integers

Usage:Used to generate the (x,y) co-ordinates for translation, both which lie between the values -100 and +50.

gen_rotation():

Input:N/A

Output:An integer value between the range -60 and 60.

Usage:Used to generate a theta value or angle of rotation that we use to rotate our template image.

calc_histogram():

Input:An image which is in grayscale form.

Output:A histogram in form of a 2D array (i.e.,matrix)

Usage:Used to generate the histogram of the image which take as an input in order to plotits histogram in the form of a 2D Numpy Array.

calc_entropy():

Input:An image which is in grayscale form.

Output:A floating point number.

Usage:Used calculate the Shannon Entropy of the provided image.

calc_joint_histogram():

Input:Two images which are in grayscale form.

Output:A 2D Numpy array and an array of 2 integers.

Usage:Used to generate the joint histogram of the two images and thus, plot their jointhistogram through which we find their joint entropy and we also return the number of pixels in the source image.

calc_joint_entropy():

Input:A 2D Numpy array and an array of 2 integers

Output:

Usage: Given the joint histogram and the number of pixels, we calculate the joint entropy and return its value.

calc_mi():

Input: Two images

Output: A floating point number

Usage: Used to calculate the NMI between the two images. This done by plotting their histogram and Joint histogram through which we obtain their Shannon Entropy and Joint Entropy and thus, we using the formula, we find the Normalised Mutual Information.

check_mi():

Input:

- A floating point number as current nmi
- A floating point number as temporary nmi
- The template image
- x value
- y value
- Theta value

Output: A floating point number that gives NMI

Usage: We check if the temporary NMI is greater than our current NMI.

If it is greater, we perform a translation and rotation operation on the template image using the x,y and theta value that was passed into the function.

iterate():

Input:

- x value
- y value
- theta value
- number of current iterations
- number of columns in the source image
- number of rows in the source image

Output: A 10*3 2D Numpy Array which contains 10set of values for operations to be performed to generate 10 children.

Usage: Using random function, we initialise a 10*3 matrix each row containing different values for (x,y,theta) which serve as parameters that are to be applied to transform the template image.

cv2.imread():

Input: Image's file location

Output: A matrix of pixel values of the image

Usage: Used to open the specified image as a matrix whose cell specify the pixel intensity values. Can optionally convert the image to grayscale or keep it in RGB.

cv2.imshow():

Input: A string which serves as title of the image and a variable which holds the image.

Output: Displays the image on a window.

Usage: Used to display an image on the screen.

np.zeros():

Input: Columns, rows and number of channels in the image as well as the type of image in bits (1- for grayscale and 3-for RGB)

Output: Creates an empty image of the specified size and stores it in a variable

Usage: Used to create a blank image filled with zeroes and store it in a variable

absdiff():

Input: Two images

Output: An image containing the binary diff between them.

Usage: Used to find the binary difference between two images, in our case, to align the two images.

waitKey():

Input: An integer

Output: Waits until a certain key is passed.

Usage: An integer is provided and depending on the status code, it displays output until the status code is met with.

7.2.2 GUI Module

gui.py

Input: The source and reference images are taken.

Output: The final aligned image is shown on the screen.

Description: It has two browse texts which has a browse button next to it for selection of the images. After the images have been selected and path is shown on the screen, the algorithm is run by pressing a particular button.

QtGui.QPushButton():

Input: Name of widget.

Output: Shows a box with the widgets name

Usage: To create widgets with their names on.

QtGui.QLineEdit():

Input: N/A

Output: Show a plain box.

Usage: Used to add text or type text in the particular area.

QtGui.QGridLayout():

Input: N/A

Output: Setting in grid layout

Usage: Setting a grid layout to arrange the widgets.

grid.addWidget():

Input: The widget to bet added and the row and column position

Output: Adding the widgets.

Usage: Adding the widgets in a proper order.

QtGui.QFileDialog.getOpenFileName():

Input: The present directory

Output: A dialog box pops up to select the required files

Usage: A dialog box opens up to select the required files for aligning them

QtCore.QFileInfo():

Input: This takes the pathname

Output: The file name present at the end it separates it.

Usage: Used to separate the pathname from the filename.

QtGui.QMessageBox.infomation():

Input: N/A

Output: Shows a pop box displaying an error message.

Usage: Used to display error messages.

QtGui.QIimage():

Input: Takes in the filename

Output: Checks whether the given file is an image or not.

Usage: Used to check filename whether it is an image file

subprocess.call():

Input: Takes in the filename to be run along with the filenames

Output: Runs any script with set of programs

Usage: Used to call another script from within the present code itself.

Chapter 8

EXPERIMENTAL RESULTS

Below are the four results of how when two images are selected and different operations are performed on them how the outputs are and there is around an 80% plus accuracy.

The below screenshot shows the basic GUI for selection of images and for aligning the images.

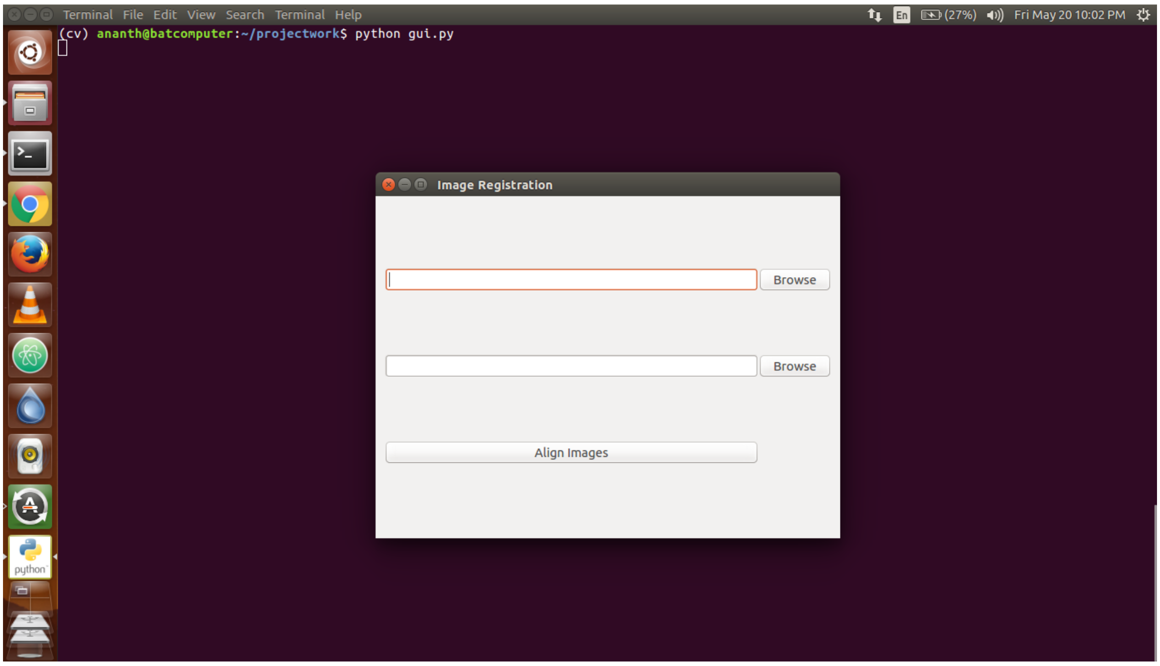

Fig 8.1: On start of GUI

Below is the screenshot of when the two image files are selected and their path names are being shown.

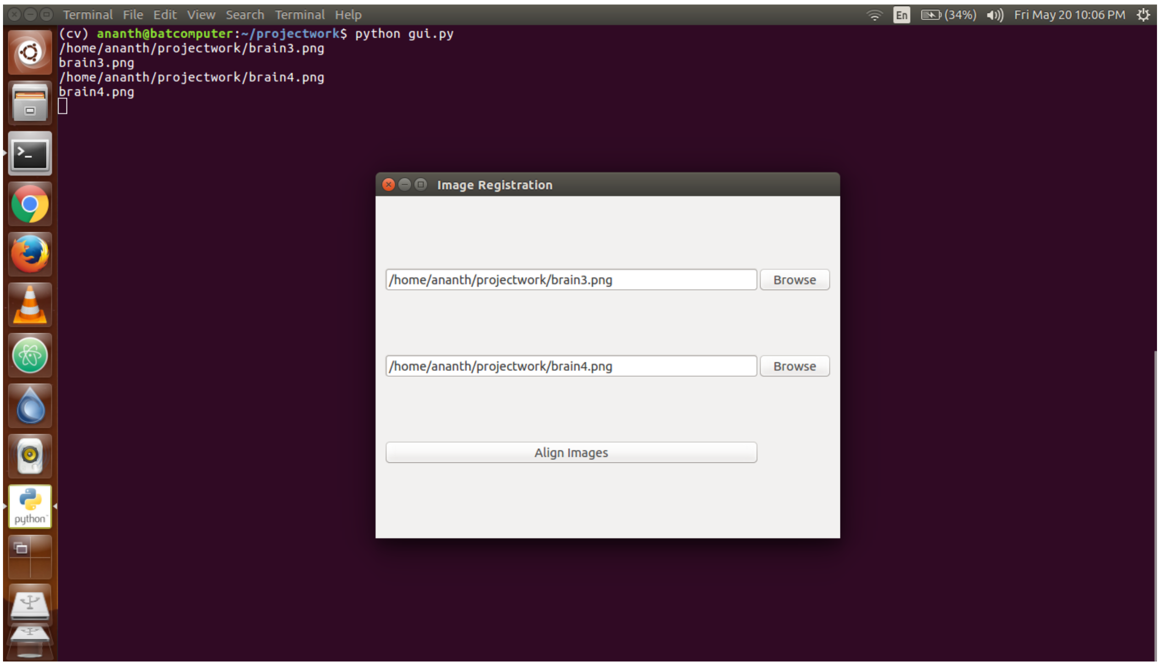

Fig 8.2: When the files are selected

Below is the screenshot of when the two images are aligned together and the four windows show the source image, template image, transformed image if any transformation is done, and the final aligned image.

Fig 8.3: Output 1

Similarly below is the screenshot of the two different images that are aligned together and the four windows show the source image, template image , transformed image if any transformation is done, and the final aligned image.

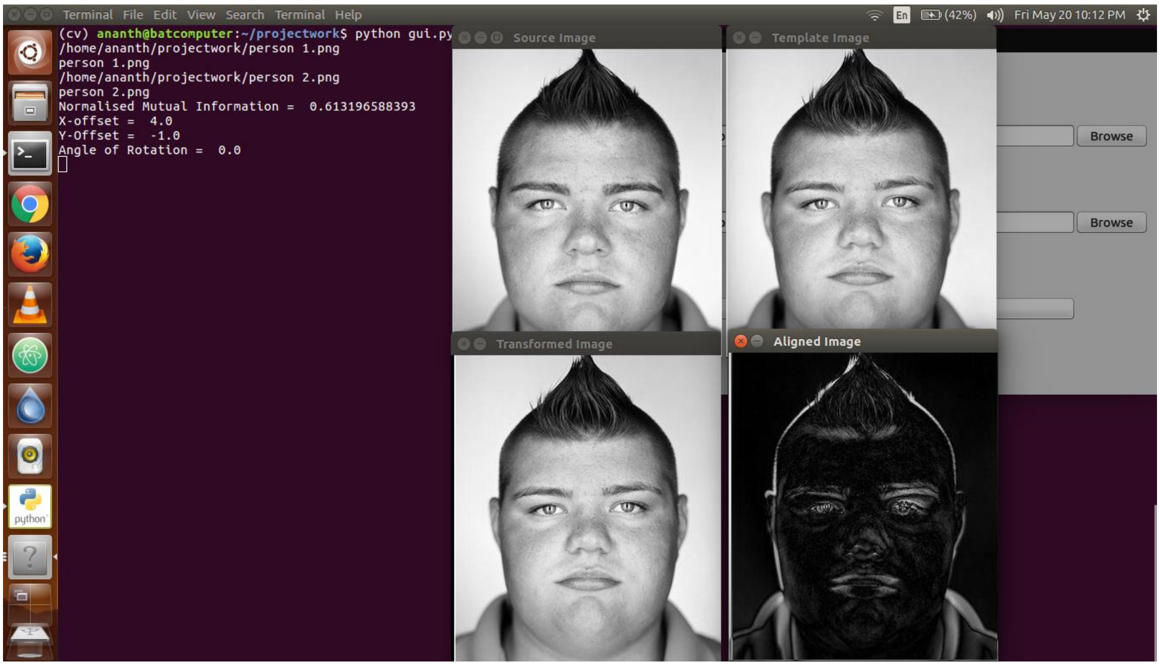

Fig 8.4: Output 2

Chapter 9

SYSTEM TESTING

Validation usually means showing that a registration algorithm applied to typical data in a given application consistently succeeds with a maximum (or average) error acceptable for the application. For geometric approaches a real-world error can be computed, which for landmark methods expresses the distance between corresponding landmarks post-registration. For rigid-registration this form of error analysis has been studied intensively and it has been found that an average target registration error for the whole volume can be estimated from knowledge of the landmark positions. Such an analysis is not generally possible for non-rigid techniques so although the error at landmarks can be established, the error in other parts of the volume is dependent on the transformation model and must be estimated using other means.

In intensity-based approaches the registration itself, usually cannot inform the user of success or failure, as the image similarity measure is not related to real-world error in a simple way. For these problems, validation is usually performed by making additional measurements post registration or showing that an algorithm performs as desired on pairs of test images for which the transformation is known. One common approach is to identify corresponding landmarks or regions independently of the registration process and establish how well the registration brings them into alignment. Pre- and post-contrast images subject to known deformation were generated and used to validate a B-spline based non-rigid registration. Of course in many applications the true point-to-point correspondence can never be known and may not even exist (e.g. inter subject brain registration).

Various kinds of consistency test are also used in validation; the most common are establishing that registration of source to target produces the same alignment as from target to source (this is commonly not the case for non-rigid registration) or that for three images, A, B, C, registration of CRA gives the same result as CRB compounded with BRA. It is important to carefully pose the registration task in application specific terms that make use of available information in the image and prior knowledge.

In most applications, careful visual inspection remains the first and most important validation check available for previously unseen data.

The different types of test cases are:

1. No files have been selected and the Align Images button is pressed:

Test Case No.	1
Input	Two images
Actual Output	Aligned Images
Output	Select a file
Remark	Failure

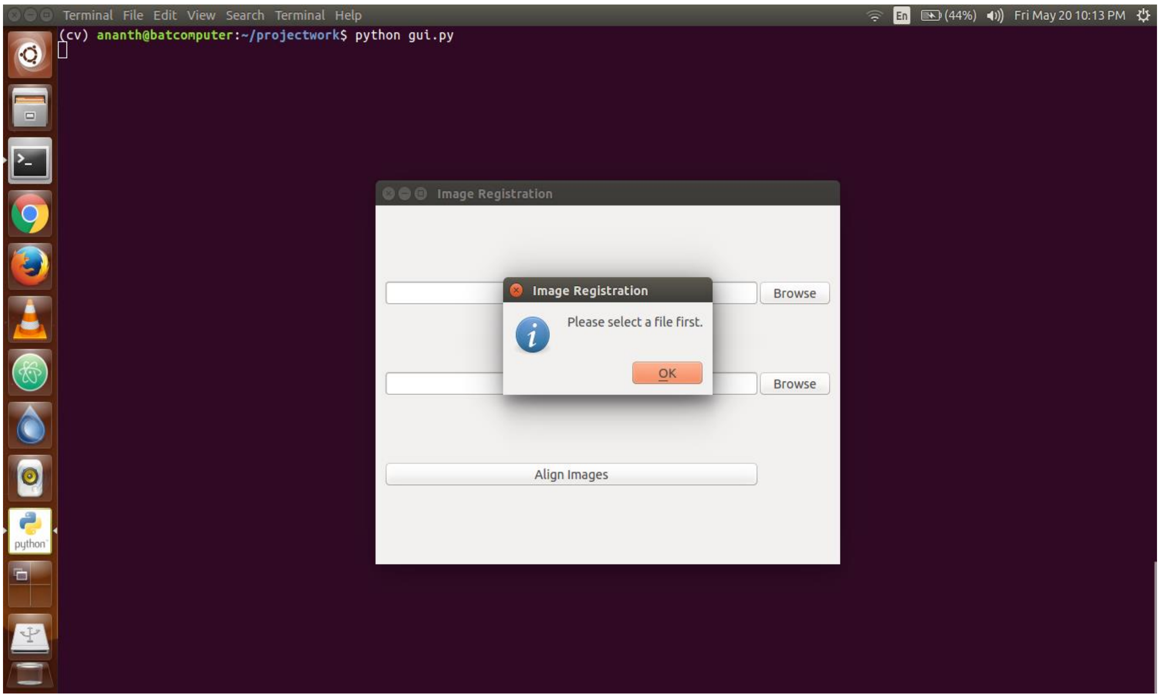

Fig 9.1: Empty files selection

2. If one of the file selected is not an image type (.jpg, .jpeg, .png):

Test Case No.	2
Input	Two images
Actual Output	Aligned Images
Output	Select an Image file
Remark	Failure

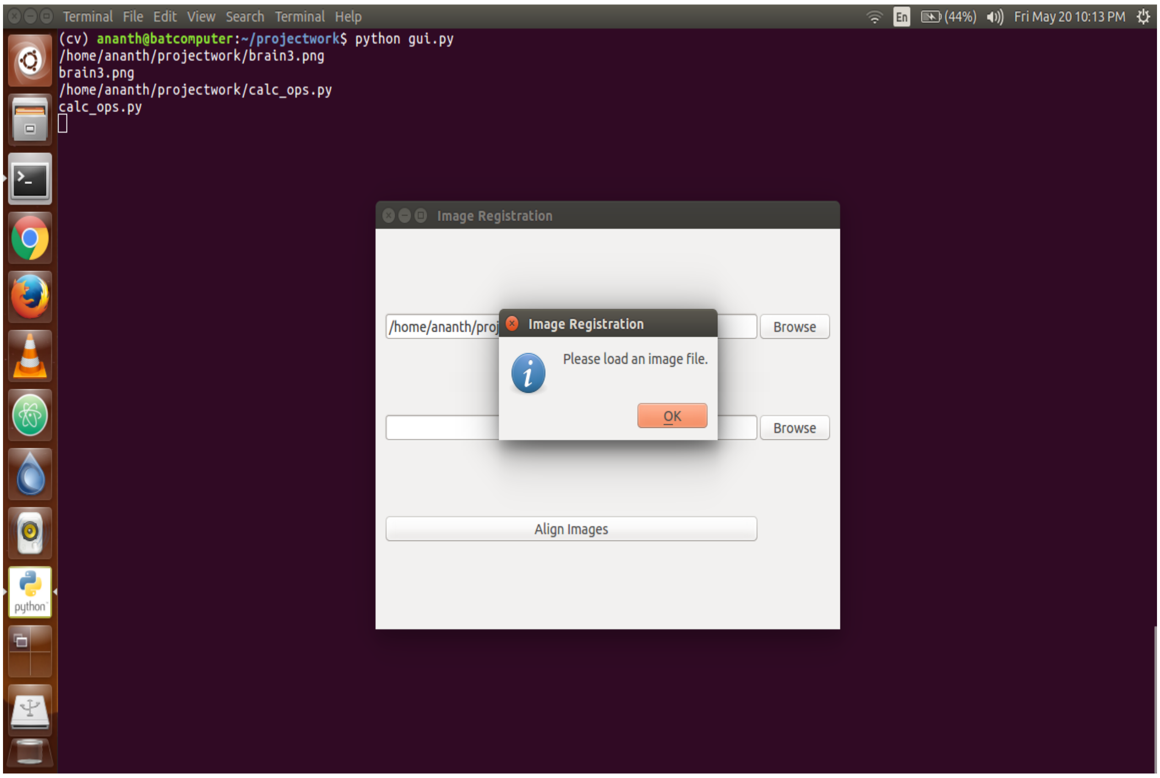

Fig 9.2: Invalid files selection

3. If both the selected files are same:

Test Case No.	3
Input	Two images
Actual Output	Aligned Images
Output	Select an Image file
Remark	Success

Fig 9.3: Perfectly Identical Images

CONCLUSION AND FUTURE WORK

CONCLUSION

Image registration is one of the most important tasks when integrating and analysing information from various sources. It is a key stage in image fusion, change detection, super-resolution imaging, and in building image information systems, among others.Using genetic algorithm and intensity-based metrics such as histogram and entropy, we try to maximize the Normalized Mutual Information through which the chances of image registration increase drastically.

Although the speed of computers has been growing, the need to decrease the computational time of methods persists. The complexity of methods as well as the size of data still grows (the higher resolution, higher dimensionality, larger size of scanned areas). Moreover, the demand for higher robustness and accuracy of the registration usually enforces solutions utilizing the iterations or backtracking, which also produces increase of computational complexity of the method.

FUTURE WORK

In the future, the idea of an ultimate registration method, able to recognize the type of given task and to decide by itself about the most appropriate solution, can motivate the development of expert systems. They will be based on the combination of various approaches, looking for consensus of particular results.

BIBLIOGRAPHY

[1] SindhuMadhuriG.,Classification of Image Registration Techniques and Algorithms in Digital Image Processing – A Research Survey, 2014

[2] P. V. Lukashevich, B. A. Zalesky, and S. V. Ablameyko, Medical Image Registration Based on SURF detector, 2011

[3] Medha V. Wyawahare, Dr. Pradeep M. Patil, and Hemant K. Abhyankar, Image Registration Techniques: An overview, 2009

[4] Roshini V S, DR K Revathy, Using Normalized Mutual Information and JointEntropy as Metrics for Registration of Images, pp 2005-2008

[5] ArdeshirGoshtasby: 2-D and 3-D Image Registration for Medical, Remote Sensing, and Industrial Applications, Wiley Press, 2005.

[6] W R CRUM, DPhil, T HARTKENS, PhD and D L G HILL, PhD, Non-rigid imageregistration: theory and practice, 2004

[7] Barbara Zitova, Jan Flusser, Image registration methods: a survey,2003

[8] Christensen GE, Johnson HJ, Invertibility and transitivity analysis for nonrigid image registration. J Electron Imaging, 2003

[9] Fitzpatrick JM, West JB. The distribution of target registration error in rigid-body point-based registration. IEEE Trans Med Imaging, 2001

[10] Image registration: https://en.wikipedia.org/wiki/Image_registration

[11] OpenCV: https://en.wikipedia.org/wiki/OpenCV