

Assignment 2

Medical Image Analysis

Azalea Alothmani

December 18, 2021

1 Introduction

Landmark-based image registration is performed during this assignment. In addition, rigid image registration using Euclidean transformations, and Procrustes alignment using similarity transformation are considered.

1.1 Image Registration

Image registration is a process performed in order to overlying two images that have been taken from the same scene at different time (multi-temporal registration), different sensors (multi-modal registration) and/or different viewpoints. The purpose of image registration is to compare multiple images of the same scene by aligning them spatially onto one single framework or into the same coordinate system. There are various method to implement image registration between a reference image and a sensed image.

The two image can be represented as I_1 for the reference image and $_2I$ for the sensed image, which is obtained if the scene in I_1 is subjected to a transformation or mapping $T(x)$ at a particular point(pixel) x is defined as

$$y = T(x)$$

$$T(x) \equiv y(x, \omega)$$

where ω defined a set of parameters associated to the specific transformation used. For rigid transformation these parameters are known as R -rotation and t -transformation. For similarity transformation. The transformed image can be defined as $T * I_2$, which is called the *pullback* of I_2 by the transformation T . The purpose is to find the mapping T that satisfy the condition such that $T * I_2$ is approximately similar to I_1 . This achieved by first finding corresponding keypoints in both images in order to estimate the optimal mapping using **Procrustes** analysis.

1.2 Scale Invariant Feature Transform (SIFT)

Landmark-based image registration is based on defining a number of keypoints and the associated descriptors, (landmarks), on the same locations in both the reference and sensed image. The keypoints are generated using **Scale Invariant Feature Transform (SIFT)**, which is a method for detecting interest keypoints in the both the reference gray scale image and sensed image. The sensed image is transformed

after associating each keypoint of interest in the reference image with the corresponding keypoint in the sensed image.

The keypoints are local extrema in space and they defined interesting points in the image. The keypoints are represented by the feature vectors containing the keypoints' characteristics. These feature vectors are called **descriptors** containing gradient statistics from a local neighbourhood around each interest point. The descriptors are invariant to translations, rotations and scaling transformations. The keypoint together with its associated descriptor is called a landmark.

1.3 Matching

After performing SIFT on the two images, a number of keypoints are found and can be denoted as $N = (x_i, y_i)$ for $i = 1, \dots, N$. x_i corresponds to the keypoints found in the reference image, while y_i are the keypoints found in the reference image such that y_i corresponds to x_i up to a small error ϵ_i . The optimal transformation or mapping is the one that minimizes the sum of the squares errors:

$$\min \sum_{i=1}^N \|\epsilon_i\|^2$$

In order to find the optimal mapping, some steps need to be performed. After finding the keypoints $x_i = 1, \dots, N_1$ for image I_1 , $y_i = 1, \dots, N_2$ for image I_2 , and the associated descriptors, a preliminary matching of pairs of keypoints using the descriptors is performed. For each keypoint k^1 in I_1 we look for a good match that satisfies the following condition: the true or good match is found if the keypoint k^2 in I_2 is significantly closer to the keypoint k^1 than the other keypoints in I_2 . The found matches might contain a percentage of outliers or false matches. In order to find these outliers **RANSAC** algorithm can be performed.

1.3.1 Random Sample Consensus (RANSAC)

RANSAC is an algorithm used to randomly select a small number of preliminary matches from a set of preliminary matches. The set of randomly selected matches is then used to estimate a model e.i. a Procrustes alignment. All the keypoints found in the reference image are transformed into the sensed image I_2 . A preliminary matching of a pair of keypoints at index i , x_i and y_i , is considered as an inlier if the transformed keypoint is within a selected threshold, **threshold**, of the keypoint it has matched to in the preliminary matching. The number of inliers is stored and the procedure is repeated to finally choose the model that has the highest number of inliers. The matches of the best inliers are then used to estimate the final model using Procrustes alignment for similarity transformation and Euclidean alignment for rigid transformation.

1.3.2 Procrustes and Euclidean Alignment

Orthogonal Procrustes is statistical shape analysis method performed to find out the optimal rotation R^* , transformation t^* and/or scaling s^* parameters that give the optimal orthogonal linear mapping. After for example performing similarity transformation or rigid transformation, a comparison between the best transformation shape or model y_i achieved previously and the model with the model achieved with the

currently estimated parameters, R^* , t^* and s^* . The aim is to obtain a model that minimizing the Euclidean distance between the two compared models. In other words the following condition need to be satisfied:

$$\min \sum_{i=1}^N \|y_i - t - Rx_i\|^2$$

The alignment first involves computing the *centroid* or the mean of the coordinate of each randomly chosen sample

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i \right)$$

The next step is to translate the the mean of the coordinates to the origin

$$\tilde{x}_i = x_i - \bar{x} \quad \text{and} \quad \tilde{y}_i = y_i - \bar{y}$$

Let $H = \sum_{i=1}^N \tilde{y}_i \tilde{x}_i^T$ and use SVD to get homography matrix $H = UDV^\top$.

Then in 2D

$$\begin{aligned} R_* &= U \operatorname{diag}(1, \det(UV^T)) V^T \\ s_* &= \frac{\sum_{i=1}^N \tilde{y}_i^T R_* \tilde{x}_i}{\sum_{i=1}^N \|\tilde{x}_i\|^2}, \\ t_* &= \bar{y} - R_* \bar{x} \end{aligned}$$

2 Methods

During this assignment Landmark-based registration methods; both rigid/Euclidean registration and similarity transformation are studied using two data sets consist of a collection of image pairs each. Collection 1 conssist of 19 pairs of microscopy images histopathological material obtained from radical prostatectomy. Each image pair consists of two consecutive sections of thickness 4 μm from these cores. One slice was stained with hematoxolin-eosin (HE), which looksred or pink, and the other slice with p63AMACR, which appears blue or purple. The shape of the tissue samples are similar but not identical. Euclidean transformation (translation + rotation) is used to study this collection since the the image are acquired with the same magnification.

Collection 2 consists of 12 pairs of images. Each pair consists of an image with 10x magnification and stained with HE and another image stained with fluorescent marker using time-resolved fluorescence microscopy (TRF), the image is 40x magnified. Similarity transformation is used to study this collection due to the different in magnification.

2.1 Image Preprocessing

The images of the two collections have been converted to grayscale images and single precision using Matlab build-in functions. The images in both collections have been downscaled with factor of 0.5 to reduce computational effort. For the TMR images in collection 2, in addition to rescaling, histogram equalization and image inversion has been performed on these images. Besides, the pixel intensity range of the images has been normalized in order to ensure that all the images are between 0 and 255.

As a first step in image registration, we need to load the data pairs and preprocess them. I always loaded the two images with the respective file names and then resized them by 0.5 in order to reduce the computational effort. Afterwards, I converted them to grayscale images and converted them to single precision.

2.2 Feature Detection and Extraction

After performing preprocessing on the data sets, the SIFT keypoints and the associated descriptors in each pair of images are computed using VLFeat command `vl_sift()`. The preliminary matchings of pairs of keypoints are computed using `vl_ubcmatch()` function. The "true" matches among all preliminary matchings are found using RANSAC procedure. First, all the preliminary matches are stored in the variable S , The sample size is denoted with the parameter n which corresponds to the number of samples we will use in each iteration to build the model. Model complexity or degrees of freedom is of the same size as the minimum sample size that we randomly pick from S . For rigid transformation in two dimensions, we have 3 degrees of freedom. For similarity transformation, we have 4 degrees of freedom. A threshold for goodness of fit, $t = 20$, is used to evaluate if the keypoints are within the the threshold for goodness of model fit. The procedure is iterative and performed under 1000 times.

For each iteration, a sample of size n is picked randomly from the set S . This is done by first finding the indices of the n samples and the corresponding sample descriptors. The x and y coordinates of each descriptor from the two images, the reference and the sensed image, are calculated.

2.3 Image Transformation and Warping

The next step is perform transformation alignment, where we first calculated the centroids of the samples, `sample1` from the reference image and `sample2` from the sensed image. Homography matrix H is then calculated using singular value decomposition (SVD) for decomposition of the matrix to its constituent parts in order to calculate estimate the U and V vectors, which are used to calculate R^* , t^* and s^* for similarity transformation. Only R^* and t^* are calculated for rigid transformation.

2.4 Feature Matching and RANSAC

The performance is then checked by i.e. checking for outliers. To do that, we need to store all the samples S and then get all possible matching pairs' coordinates. The transformation is then calculated with the R^* , t^* and s^* parameters before apply it to all possible matches using the build-in function `transformPointsForward()`, let's call it `proj`. The difference vectors between `sample2` and `proj` coordinates are calculated. We iterate then through the difference vectors to find the true matches. The number of inliers is calculated (Elements in S fitting estimated model within threshold t). The chosen sample and number of inliers are stored if better than previous one.

The images are then aligned by fit the model to the inliers of best samples found in the previous step. Procrustes alignment is then performed on these samples in order to find best R^* , t^* (rigid) and s^* (similarity). Finally, the best Rs , ts and ss parameters are stored and saved. In order to display the registered image/images, the transformation matrix is calculated and applied to the first image. The registered image and the fixed image are then showed o top of each other.

3 Results

After performing the algorithms described above, the resulting images are displayed, and the magnitude of transformation, scale factors and the rotation are achieved.

3.1 Collection 1

The images below show the resulting rigid transformation for some of image pairs:

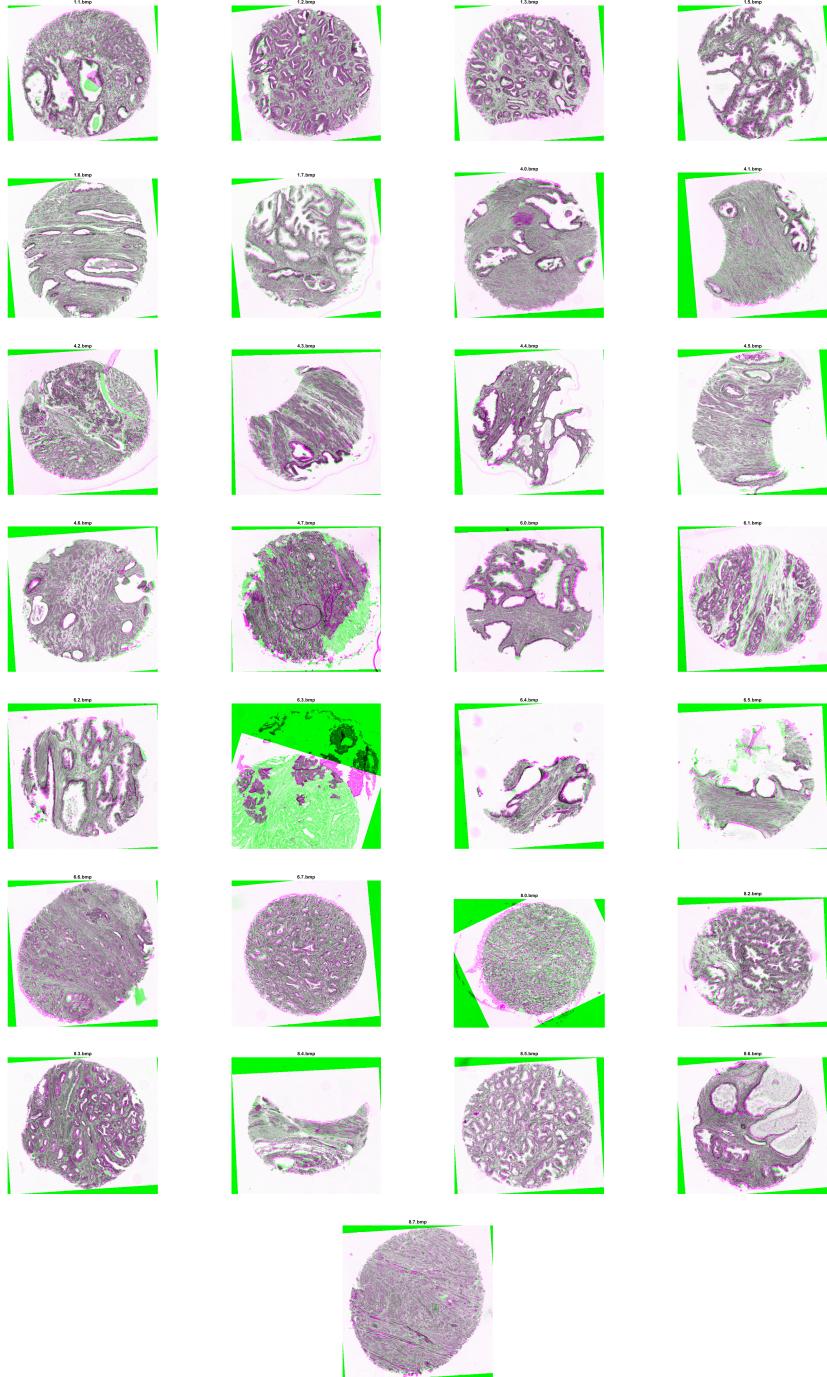


Figure 1: Images from collection 1 after performing rigid transformation.

The magnitudes of translation vectors for the 29 image pairs can be read directly from the calculated t^* , the results are shown in Fig.1. The angles can be calculated from the rotational matrices as shown in tab.2.

Translation x	Translation y
-12.9439	9.9063
-16.5249	27.6729
-21.2900	13.1220
-17.2385	25.3063
-20.6628	30.7014
-25.8984	26.5797
-31.6671	17.8058
27.4461	25.1243
-11.4592	23.3814
0.4077	24.4181
-1.0814	25.1352
-30.8457	32.7580
-6.3600	43.8181
-6.1758	12.7959
-11.1086	24.2315
5.6322	11.3656
-3.9502	33.4996
651.8465	871.1070
13.8083	24.7576
-19.9235	35.7633
-5.8697	24.7528
-52.2305	26.8501
-28.2110	152.3703
-34.9972	30.7214
-22.7211	22.9151
0.5557	66.6713
-19.0364	37.9759
-10.5735	31.7342
-27.8326	35.9273

Table 1: The magnitudes of translation vectors for the 29 image pairs in collection 1

The rotation angles for the 29 image pairs:

Rotational Angle
-4.0616
-6.4211
-4.6385
-5.5088
-5.5837
-6.6871
-4.3426
-5.0201
-3.8005
-1.6472
-3.5103
-4.8509
-4.1041
-1.0273
-1.9806
-2.9897
-2.6689
17.3852
-4.3804
-2.7441
-4.1066
-5.3409
-25.4297
-2.5173
-3.8238
-2.8617
-3.6306
-4.7707
-3.7153

Table 2: Rotation angles for images in collection 1

3.2 Collection 2

The images below show the resulting similarity transformation:

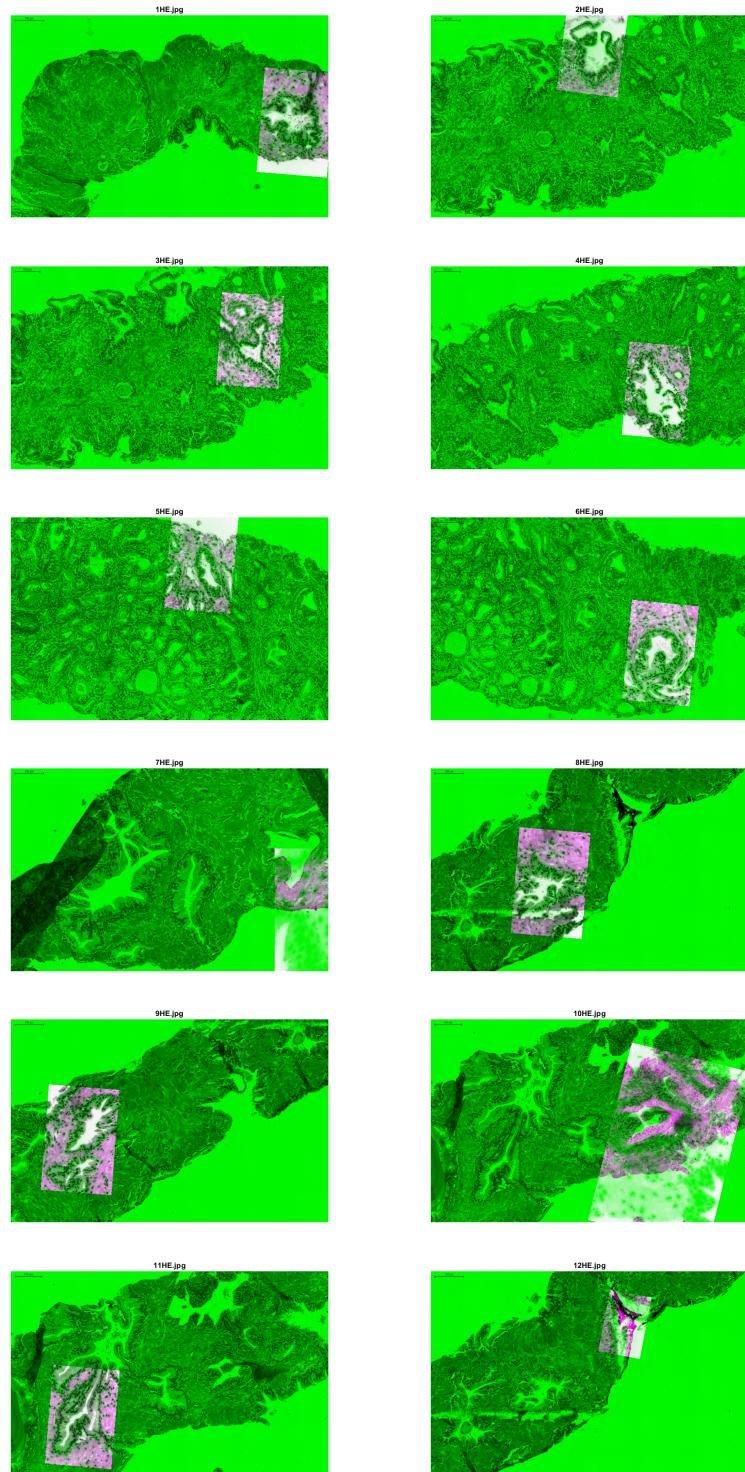


Figure 2: The resulting images after performing similarity transformation on the images from collection 2

The rotation angles of the image pairs in collection 2:

Rotational Angle
94.7377
94.9517
93.8730
94.6299
97.1380
93.5579
94.8085
94.9374
95.6974
0
94.7869
94.4814

Table 3: Rotation angles for the 12 pairs of images in collection 2

The magnitudes of translations for 12 image pairs in collection 2

Image1	Image 2
635.8753	156.2554
263.0262	238.9615
588.2709	-59.9736
781.7308	139.2790
477.1834	-19.0743
658.4987	71.2247
629.3859	193.1040
549.6104	-6.5094
646.2133	213.4761
634.9691	191.4475
385.0427	155.2421
262.5735	169.2384

Table 4: Magnitudes of translations for the 12 pairs of images in collection 2

The scale factors for the image pairs in collection 2

0.4348
0.4369
0.5653
0.4625
0.4011
0.4082
0.4108
0.4380
0.4412
1
0.4668
0.4641

3.3 Visual Assessment

3.3.1 Collection 1

Summary of visual assessment perform on collection 1 is shown in the table below. The rigid transformation seems to perform good on almost all the images in collection 1, expect for few images. After performing visual assessment for all the pairs of images, I found out that the algorithm works well. However, for some image pairs the scale seems to be a bit different making it challenging to achieve a perfect image registration. Here, similarity transformation could be a better choice.

Image	Performance
1.1	Very good
1.2	Very good
1.3	Very good
1.5	Very good
1.6	Very good
1.7	Very good
4.0	Very good
4.1	Very good
4.2	Very good
4.3	Good
4.4	Very good
4.5	Very good
4.6	Very good
4.7	Okay
6.0	Good
6.1	Good
6.2	Good
6.3	Completely Wrong
6.4	Good
6.5	Okay
6.6	Okay
6.7	Good
8.0	Probably Bad
8.2	Good
8.3	Very good
8.4	Good
8.5	Very good
8.6	Good
8.7	Very good

3.3.2 Collection 2

Summary of visual assessment perform on collection 2 is shown below:

Image	Performance
1	Very good
2	Very good
3	Very good
4	Very good
5	Very good
6	Very good
7	Completely wrong
8	Very good
9	Very good
10	Very good
11	Very good
12	Completely wrong

4 Conclusions and Discussion

Performing the algorithm described above performed in general well on all image pairs from both collection 1 and collection 2. The evaluation is performed by virtual inspection of the images. Gray areas correspond to areas that have similar intensities, while magenta and green areas show places where one image is brighter than the other. However, the transformation performed poor on image "6.3.bmp" from collection 1. This might be due to the fact the HE image of the pair (6.3) is over-stained in comparison to the other image, making it difficult to perform good alignment. Furthermore, the rotation is image '8.0' is slightly off, as it can be seen in Fig.?? and from the rotational angle in Tab2. This might depends on how the images are stained. The HE image seems to have been uniformly stained while in the p63AMACR image there are some missing parts. This can give a false interpretation of the cell shape, making it difficult to have a good rotation.

In addition, the alignment method performed poor on "7HE.jpg". The algorithm might need to be improved in order to solve this issue. Some somethings have been tried to encounter the issue but without any improvement.