

UNIVERSITAT DE GIRONA



MEDICAL IMAGE REGISTRATION AND ITS APPLICATION

---

## Rigid Image Registration

---

*Authors:*

Ali Berrada  
Ama Katseena Yawson

*Supervisors:*

Rafael Garcia  
Nuno Gracias

December 17, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Overview . . . . .	2
1.2	Problem Definition . . . . .	2
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Overview . . . . .	2
2.2	Lowe's SIFT vs Third Party Implementation . . . . .	3
2.3	Image Pairs Registration by Estimating Homography . . . . .	4
2.3.1	Euclidean . . . . .	4
2.3.2	Similarity . . . . .	4
2.3.3	Affine . . . . .	5
2.3.4	Projective . . . . .	5
2.3.5	Reprojection Error . . . . .	6
2.3.6	Homography for Medical Images . . . . .	6
2.4	Improvement to the Registration Accuracy . . . . .	7
2.4.1	RANSAC . . . . .	7
2.4.2	Data Normalization . . . . .	8
<b>3</b>	<b>Conclusion</b>	<b>9</b>

## List of Figures

1	Illustration of Medical Image registration . . . . .	2
2	Feature extraction using the Lowe's and VLFeat algorithm using retina images . . . . .	3
3	Feature extraction using the Lowe's and VLFeat algorithm using skin images . . . . .	3
4	Illustration of the homography transformation . . . . .	4
5	Visualization of the registration results using overlay in different color channels . . . . .	5
6	Plot of features and their corresponding matches . . . . .	6
7	Registration results of the retina images using overlay in different color channels . . . . .	7
8	Registration results of the skin images using overlay in different color channels . . . . .	7
9	Registration results of the retina images after applying the Ransac algorithm . . . . .	8
10	Registration results of the skin images after applying the Ransac algorithm . . . . .	8

## List of Tables

1	Mean reprojection error for the synthetic images . . . . .	6
2	Mean reprojection error for the medical images . . . . .	7
3	Mean reprojection error for the medical images after applying the Ransac algorithm . . . . .	8
4	Mean reprojection error for the medical images after applying Ransac and data normalization. . . . .	9

# 1 Introduction

## 1.1 Overview

As shown in Figure 1, image registration can be defined as the process for finding the correspondence of features between images taken at different times or using different imaging modalities (medical images). This image processing technique is a key factor in medical image analysis. The correspondences can be used to alter the appearance by applying transforms such as rotation, translation, scaling and shearing. The most intuitive use of registration in the medical domain is to correct for different patient positions between scans [1].

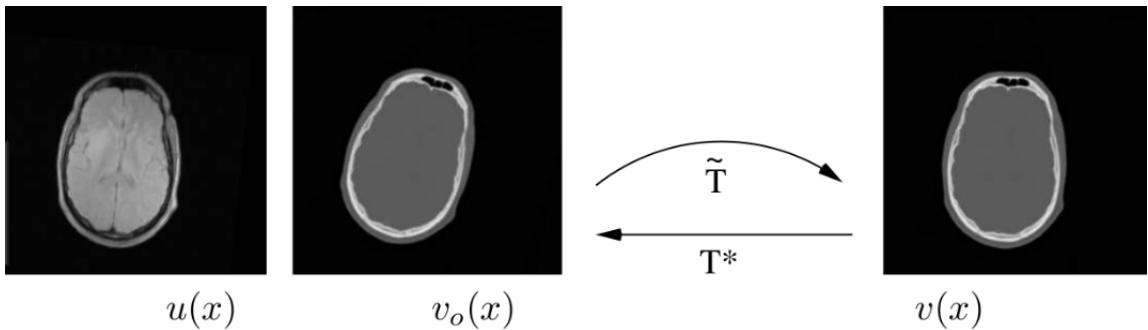


Figure 1: Illustration of Medical Image registration

## 1.2 Problem Definition

In this exercise, the underlying objective were as follows:

- To understand and become familiar with the SIFT detector and descriptor.
- To compare the different implementation of the SIFT algorithm. That is, Lowe's and VLFeat implementation.
- To implement a planar transformation (Homography) to register both artificial and medical images.
- To improve the planar transformation by applying RANSAC and data normalization to the pipeline.

# 2 Implementation

## 2.1 Overview

All implementation were carried out using MATLAB R2018b. The datasets provided in the intranet of the course were used in carrying out the various steps. The implementation details are explained in the next subsections.

## 2.2 Lowe's SIFT vs Third Party Implementation

The Scale-Invariant Feature Transform (SIFT) is made up of a feature detector and a feature descriptor. The feature detectors are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint [2]. The descriptor on the other hand associates to the regions a signature which identifies their appearance compactly and robustly [3]. Before beginning the activities of this coursework, we read the Lowe's sift implementation to understand how the algorithm works. Using the provided link in the lab guide, the siftDemoV4.zip package (version 4, July 2015) was downloaded and tested on the images in the 'DataSet00.zip'.

Following the same procedure, the third party implementation (VLFeat) was also downloaded and tested on the same set of images. Unlike the Lowe's implementation, many features were extracted in the VLFeat implementation of the sift algorithm. To visualize the similarities between these two approaches, we tuned the parameters of VLFeat implementation to replicate the results of the Lowe's implementation. We tuned parameters such as the peak thresh and the edge thresh. The peak thresh filters peaks of the DoG scale space that are too small and the edge thresh removes peaks of the DoG scale space whose curvature are too small. After tuning these parameters, we still couldn't replicate the Lowe's implementation and hence we randomly selected a total number of features to display. Other parameters tested were the number of octaves and the number of levels per octave. The results of features extracted from each implementation for retina and skin images are displayed in Figure 2 and Figure 3 respectively. One significant observation made from both implementation was that, some features extracted in Lowe's implementation were also extracted by the VLFeat implementation after the random selection.

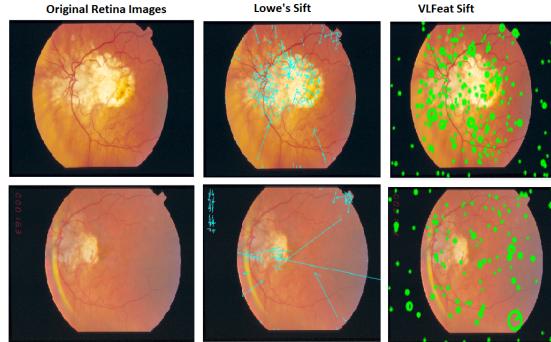


Figure 2: Feature extraction using the Lowe's and VLFeat algorithm using retina images

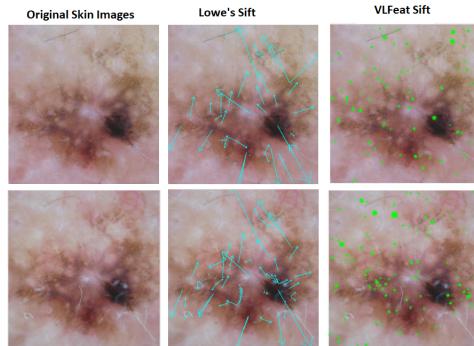


Figure 3: Feature extraction using the Lowe's and VLFeat algorithm using skin images

## 2.3 Image Pairs Registration by Estimating Homography

Homography can be defined as a perspective transformation of a plane, that is, a reprojection of a plane from one camera into a different camera view. This has many practical applications in fields such as image rectification, image registration, image stitching, etc.

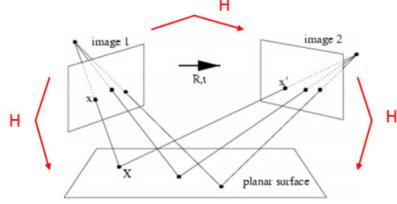


Figure 4: Illustration of the homography transformation

In this subsection, a (3 x 3) homography matrix for each transform was estimated. We implemented a Matlab function called 'computeHomography()' which takes as input two ( $N \times 2$ ) matrices and the model type. The two matrices represent features and their corresponding matches while the model represent the transformation type which includes euclidean, similarity, affine and projective. Using 'Dataset01', we estimated the homography for each model type using the 'Features.mat' file. For all the features and their corresponding points, we solved the system of equation using the linear relation as shown in Equation 1.

$$Ax = b \quad (1)$$

where  $A$  is matrix representing the features,  $b$  is vector representing the corresponding matches of the features and  $x$  are the unknown parameters of the model. For each model, the number of unknown  $x$  were fewer than the matrix  $A$  and hence, the least squares approach was used in solving the system of equation. Depending on the model type, the number of degrees of freedom (DOF) varied. The details of each model are explained below.

### 2.3.1 Euclidean

The euclidean homography matrix (Equation 2) has 3 degrees of freedom which includes translation in x and y axis followed by rotation. This transformation type is non-linear and hence in finding the 3 parameters, we initially determined the similarity homography matrix. From the similiarity homography matrix, we found the angle of rotation and then adjust the matrix again using the correct corresponding parameters.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{vmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

### 2.3.2 Similarity

The similarity homography matrix (Equation 3) is likened to the euclidean transform. The difference between these transforms is that the similarity transform has a scale factor multiplying each of the rotation component and hence there are 4 factors which includes translation in x

and y axis, rotation and scaling.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{vmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{vmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

### 2.3.3 Affine

The affine homography matrix (Equation 4) has 6 degrees of freedom. The transformations that makes up this transform includes translation in x and y axis, rotation, scaling and shearing in x and y.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{vmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

### 2.3.4 Projective

The projective homography matrix (Equation 5) has 8 degrees of freedom.

$$\begin{bmatrix} kx' \\ ky' \\ k \end{bmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{vmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$

The developed Matlab function was used to estimate the homography of each of the models using 'Features.mat' file. After obtaining the homography matrix for each model, we transformed the reference image using Matlab 'imwarp()' function to verify the accuracy of the homography matrix. For qualitative evaluation, the transformed images were overlayed on the moving images using red (moving image) and green (transformed image) channel. The results obtained for all synthetic images are shown in Figure 7.

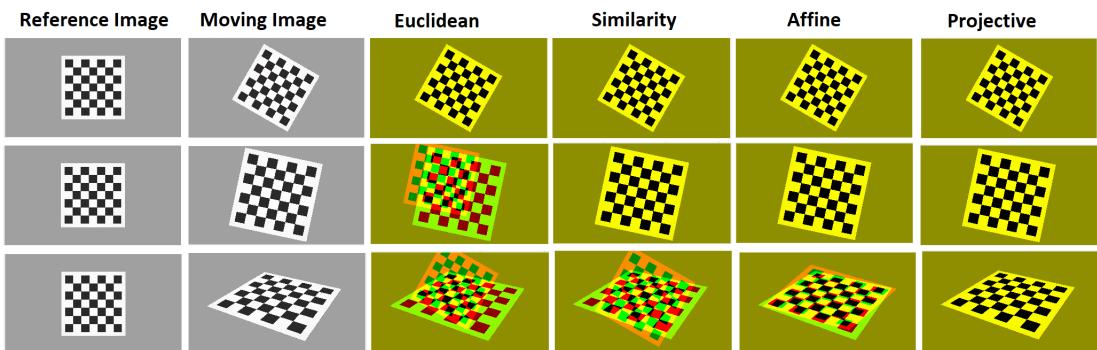


Figure 5: Visualization of the registration results using overlay in different color channels

By visually inspecting all the transforms in Figure 7, the results obtained for the projective transform in all the moving image cases were more accurate than the other transforms and hence we deduced that the higher the number of degree of freedom, the more accurate the registration results.

### 2.3.5 Reprojection Error

The quantitative evaluation was made using the reprojection error. The reprojection error was estimated using both the forward and backward transformation. The mathematical equation used in the computing this factor is shown in Equation 6.

$$Error_i = d(Hu_i, u'_i) + d(u_i, H^{-1}u'_i) \quad (6)$$

where  $H$  is the homography matrix,  $u_i$  is the input reference features,  $u'_i$  is the transformed features,  $H^{-1}$  is the inverse of homography and  $d$  is the distance. The mean reprojection error measured for all the transforms are shown in Table 1.

Features	Euclidean	Similarity	Affine	Projective
01	0.69	0.46	0.46	0.46
02	526.02	0.36	0.36	0.35
03	379.11	201.66	57.64	0.55

Table 1: Mean reprojection error for the synthetic images

As shown in Table 1, the projective transform has the lowest error for all the 3 case. Quantitative results coupled with the qualitative results proves that, the higher the number of degrees of freedom, the more accurate the registration results.

### 2.3.6 Homography for Medical Images

Using the images in 'Dataset00', we extracted features and their corresponding matches from the skin and retina images using the 'match()' function. We added some few lines of code to the original function to obtained the features and their corresponding matches since the function returned only the number of matches. Another observation made while using the 'match()' function on the medical images was that, the number features and the corresponding matches were very few. To solve this issue, we adjusted the contrast of the original images using the Matlab 'adaptHisteq()' function to obtain more features and corresponding matches. An illustration of matching with and without contrast adjustment are shown in Figure 6 below.

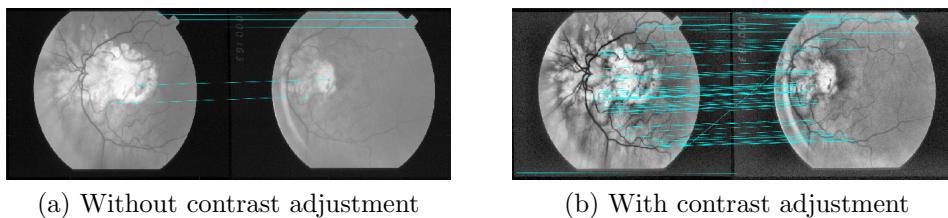


Figure 6: Plot of features and their corresponding matches

From the list of features and corresponding matches, we estimated the homography for each transform. The results obtained are displayed in Figure 7 and Figure 8.

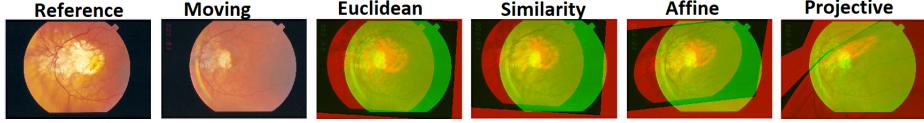


Figure 7: Registration results of the retina images using overlay in different color channels

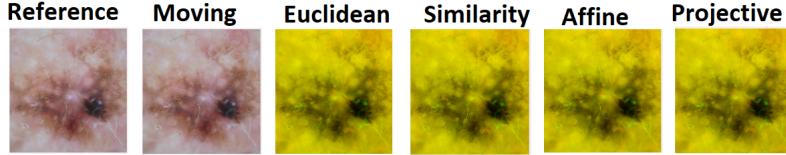


Figure 8: Registration results of the skin images using overlay in different color channels

Unlike the results obtained with the synthetic images, there were significant misalignment both visually and quantitatively in the estimated homography. The errors for each transform are recorded in Table 2 below.

Image Type	Euclidean	Similarity	Affine	Projective
Skin	9.92	9.35	9.30	9.43
Retina	70.21	51.24	64.63	572.47

Table 2: Mean reprojection error for the medical images

With the medical images, the transformation that produced minimum mean reprojection error for skin image was the affine transform while the transformation that produced the minimum error for retina image was the similarity transform. This misalignment in the estimated homography matrix could be attributed to the presence of outliers in the features and corresponding matches extracted as shown in Figure 6b.

## 2.4 Improvement to the Registration Accuracy

As a way of reducing the reprojection error, we introduced the concept of RANSAC and data normalization in our pipeline.

### 2.4.1 RANSAC

RANSAC stands for Random Sampling Consensus. This algorithm is an iterative process which leads to robust estimation of homography matrix by rejecting outliers in the dataset. In estimating the homography matrix using this algorithm, random sample are selected from the dataset. These randomly sampled points are used to fit a model and the best model with the maximum consensus is selected to compute the homography matrix.

We implemented a new function called 'computeHomographyRANSAC()' which infuses the Ransac algorithm. The function takes as input 2 ( $N \times 2$ ) matrix, the model type and a boolean which specifies whether the input data should be normalized or not. The function returns the inlier features with their corresponding matches and the homography matrix. To fully evaluate the impact of the Ransac algorithm, we set the input 'dataNormalized' to 0. In our implementation of the Ransac algorithm the minimum number of randomly selected varied depending on the model type. ie : euclidean(2), similarity(2), affine(3), projective(4). We

adopted the adaptive Ransac algorithm. The number of iteration was set to infinity. Also we initialized the probability of outliers to the worst case which is 1, probability of inliers as 0.99 and set a threshold of 20. These initialization were updated after every iteration until the loop converges. For the randomly selected samples, we fit a model using the 'computeHomography()' function and the error associated with each selected sample is determined. From the computed error, we determine whether or not the samples are inliers or not. The results obtained using the Ransac algorithm are shown shown in Figure 9 and Figure 10 below.

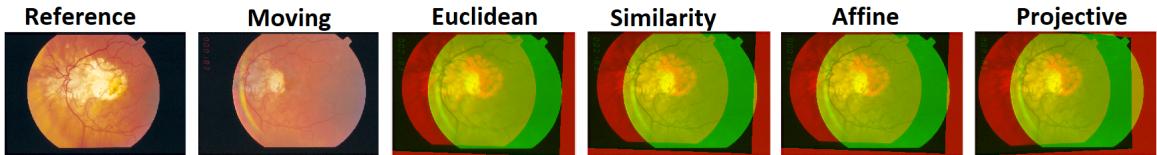


Figure 9: Registration results of the retina images after applying the Ransac algorithm

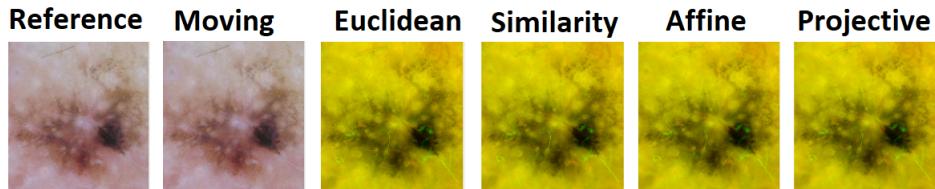


Figure 10: Registration results of the skin images after applying the Ransac algorithm

Image Type	Euclidean	Similarity	Affine	Projective
Skin	6.97	5.76	5.43	5.64
Retina	15.29	2.51	2.18	5.06

Table 3: Mean reprojection error for the medical images after applying the Ransac algorithm

By comparing the results of Figure 7 and Figure 9, the overlay of Figure 9 was more accurate especially with the affine and projective transform after applying the Ransac algorithm. This results concords with mean reprojection error as shown in Table 3. From the mean reprojection error as shown in Table 3, it can be deduced that the removal of outliers from the data samples enabled a more accurate estimation of the homography matrix.

#### 2.4.2 Data Normalization

Another important factor we considered to increase the accuracy of the estimated homography matrix was normalizing the data set to appropriately condition them to increase its numerical stability. We developed a function known as 'normalizeData()'. The input of this function is the data set to be normalized and returns transform  $T$  and the normalized data set. In implementing this function, we translated the centroid point of the data set to the origin and scaled them such that the average distance from the origin was  $\sqrt{2}$ . The same procedure was repeated for the corresponding matches and the transform  $T'$ . Once we obtained the normalized features with the corresponding matches, we compute the homography matrix using the 'computeHomographyRANSAC()' function. The proceeding step was denormalizing using Equation 7.

$$H = T'^{-1} \hat{H} T \quad (7)$$

where  $T'^{-1}$  is inverse transform of the matches,  $\hat{H}$  is the normalized homography matrix,  $T$  is the transform of the features and  $H$  is the denormalized homography. This factor helped in reducing the mean reprojection error in some transforms. The results after applying this algorithm are recorded in Table 4.

<b>Image Type</b>	<b>Euclidean</b>	<b>Similarity</b>	<b>Affine</b>	<b>Projective</b>
Skin	6.29	6.36	5.71	5.63
Retina	2.55	2.51	2.18	2.01

Table 4: Mean reprojection error for the medical images after applying Ransac and data normalization.

Comparing the results of Table 3 and 4, there was significant reduction in the mean reprojection error for the euclidean and projective transforms thanks to data normalization. The mean reprojection error was reduced from 15.29 to 2.55 in the retina image using the euclidean transform. Likewise, the mean reprojection error using the projective transform was also reduced from 5.06 to 2.01.

### 3 Conclusion

Image registration is an important procedure to compare and integrate images obtained at different times. The goal of this lab exercise was to become familiar with the sift features and descriptor from the different implementation and the estimation of planar transformation. Using the sift algorithm, we extracted features and their corresponding matches using the medical images. The planar transformation was estimated considering the euclidean, similarity, affine and projective transformation. From the results obtained for each of the transforms, we observed qualitatively and quantitatively that the higher the number of degree of freedom, the better the registration results. As a way of improving the registration results, we included the Ransac outlier rejection and data normalization to our implementation. We observed a drastic reduction in the mean reprojection error. To conclude, this lab exercise has helped us in building a clearer understanding of the basics of image registration.

## References

- [1] William R Crum, Thomas Hartkens, and DLG Hill. Non-rigid image registration: theory and practice. *The British journal of radiology*, 77(suppl\_2):S140–S153, 2004.
- [2] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [3] Andrea Vedaldi and Brian Fulkerson. Vlfeat tutorial.  
<http://www.vlfeat.org/overview/sift.html?fbclid=IwAR2Lqe9Ak7mCz72bXfQghKVebgEYphmEYsnNYxHgmo0CR4An0v0suBiCP80>. Accessed: 2018-12-06.