HOMEWORK 1 REGISTRATION

LEV KOLEZHUK, ROBERTO PAOLELLA

November 5, 2017

1 Introduction Part 1-SIFT

It is not possible to work with computer vision without considering the registration problem. One big help in this field has arrived in the 2004, with the publication of the article:
"Distinctive Image Features from Scale-Invariant Keypoints "
The aim of this homework is exploring the SIFT(scale invariant Feature Transform) algorithm. However, for better understanding the registration problem, it was decided to compare the result of SIFT with the result of third party implementation VLfeat. Original SIFT algorithm implementation is available online, but there is no possibility to modify the various parameters explained in the paper because of the fact that the source code remains undisclosed and is strongly patented. Instead, Vlfeat has some parameters which can be tuned depending on a specific case which, sometimes, may become a better option. One of the aims of this lab is to understand the meaning of the parameters, and to experience with their values in order to achieve best results in features detection.

1.1 SIFT

The original algorithm of Lowe has been applied on the different images of the dataset.In the figure 1 has been reported an image of the Dataset, Canon, and the respective keypoints obtained with the SIFT [Lowe, 2004] algorithm.

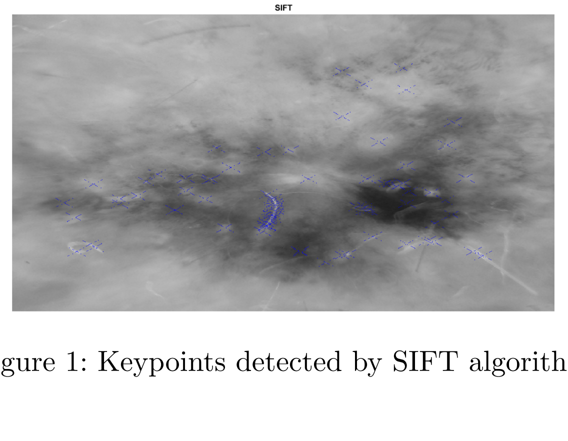


Figure 1: Keypoints detected by SIFT algorithm

1.2 Brief SIFT principle

As it has been already said, there is no possibility to tune the parameters mentioned in the paper.
The first step that SIFT has to do is to approximate keypoint locations, this is possible by using the DoG at the two nearby scales for looking the intensity change. Once we found the keypoints, the next step is to filter the less reliable ones.

1.3 Rejection of point with Low Contrast(peak threshold)

This is done by performing a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. Through this information the points with low contrast or along an edge, are rejected. Moreover, the keypoint location and scale can be interpolated for reaching greater accuracy. For this reason, to define the DoG function of keypoints in a small 3D neighborhood, it is used a second-order Taylor-series D. The Hessian and the derivative of D are approximated through the differences between neighboring sample points. To find the value of the extremum in the DoG's neighborhood the derivative of D has to be set to the value zero. All the extrema smaller than a threshold are discarded as "weak extrema" or "low contrast points".
If the magnitude of the intensity at the current pixel in the DoG image (that is being checked for minima/maxima) is less than a certain value, is rejected. Because we have subpixel keypoints we used the Taylor expansion to refine keypoints and we need to use it again in order to get the intensity value at subpixel locations. Again the keypoint will be reject if it's magnitude is less than a certain value.

1.4 Removal of high-contrast keypoints residing on edges (edge threshold)

Some key points reside on edges, hence the response to the DOG filter will be high. The SIFT paper uses the Hessian matrix around the keypoint. An important information about the the keypoint neighbours can be estimated by the eigenvalues of the Hessian. In effect, observing the eigenvalues, the maximal and minimal principal curvatures of the surface can be obtained. An edge will have high maximal curvature, but very low minimal curvature.
A keypoint which is on a corner (not an edge) will have high maximal and minimal curvature. For every keypoint SIFT provided a vector of features that characterize the region around it. For determining the correspondence between two image we need features that must not change when the scale position etc change in the image.For example the color of a pixel is not a good features to determine correspondences.
So SIFT provides features characterizing a salient point that remains invariant to the change in scale or rotation.
The main step of sift algorithm are

1. Determine the keypoint
2. Refine location and scale

3. Determine orientation
4. Determine descriptors.

In this first assignment the focus was put more in detector of the keypoint and not in the descriptor.
Several parameters in the SIFT algorithm are not clear such as: descriptor size, size of the region, various thresholds theoretical treatment

1.5 Test Lowe's SIFT vs Third Party Implementation

The comparison between SIFT and VLFEAT [Vedaldi and Fulkerson, 2008] was the first aim of the lab. Since the original SIFT implementation uses unknown parameters, it is of our interest understand which settings of VLFEAT implementation give similar result to SIFT. If we only consider the generated keypoints and their descriptors,the result of the comparison between the two algorithm, can be only analyzed qualitatively. We can try to tune VLFEAT algorithm for having the same number of SIFT keypoints, hence that means that many combinations of VLFEAT parameters has to be investigate for reach the same quantitative result [fig(graph)]. The parameters were tuned based on subjective evaluation of the qualitative by comparing the results of Lowe's SIFT implementation and the third party one. In VLFEAT case we should be interested in generating the keypoints in similar locations as in the original SIFT (edge points, points in the crucial and highly descriptive features of the object), which can partly be analyzed by the visual analysis. In order to achieve this goal was conducted an investigative analysis were the parameters of SIFT were spanned. In VLFEAT the modifiable parameters are:

- The peak threshold that filters peaks of the DoG scale space that are too small (in absolute value)
- The edge threshold that eliminates peaks of the DoG scale space whose curvature is too small (such peaks yield badly localized frames).

In the images 2 are shown the matches between Lowe's SIFT and VLFEAT implementation . It can be notified that in certain configuration the keypoints are generated in very similar locations, hence they will cause the same good quality of the result during the homography estimation, while in others configurations of the parameters the variations of the keypoints was bigger than the others.

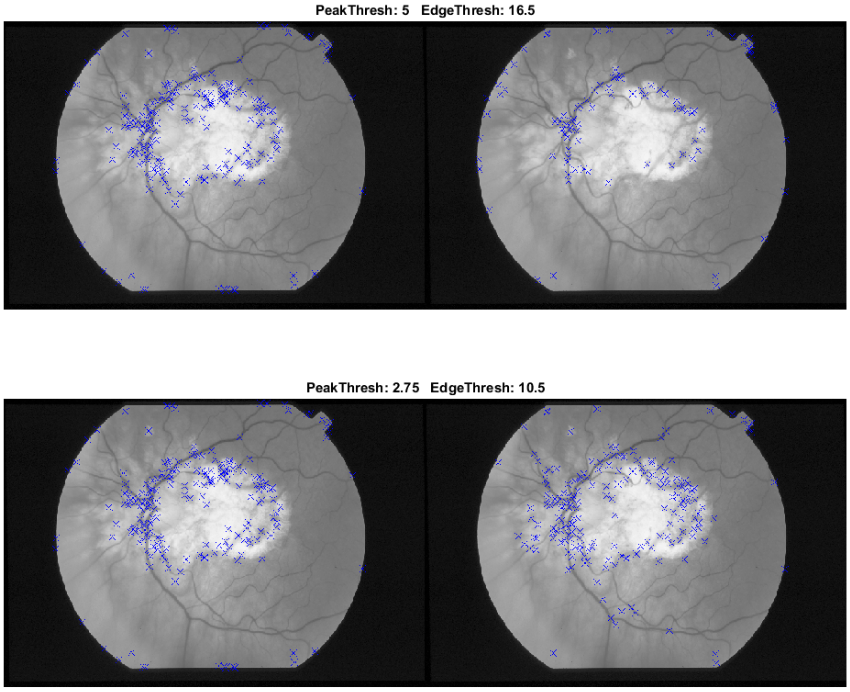

PeakThresh: 5 EdgeThresh: 16.5
PeakThresh: 2.75 EdgeThresh: 10.5

Figure 2: COMPARISON BETWEEN SIFT AND VLFEAT

In order to reach the same behavior of the original SIFT, the number of keypoints generated by VLFEAT were analyzed according with the variations of the two parameters edgethresold and peakthresold, In fig 3 is illustrated the result of this analysis

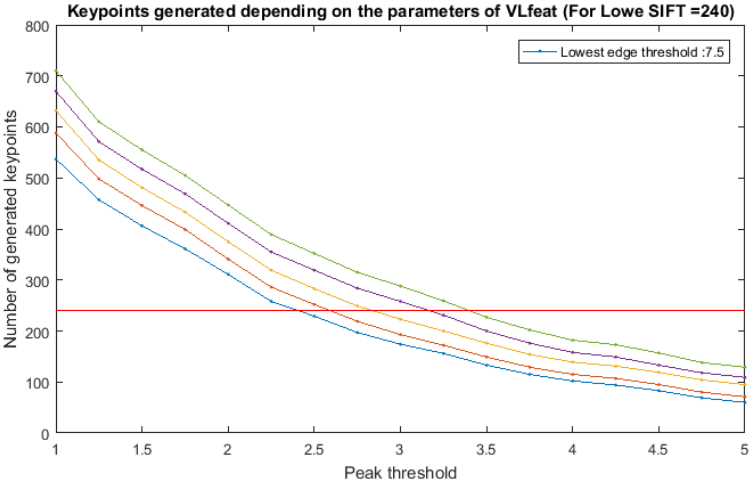


Figure 3: Keypoints generated by vlfeat

2 Homography computation

2.1 Introduction

Using SIFT or VLFEAT is possible detect features on images of a given pair .Same feature points can be represent in both image. In order to solve registration problems match the same features in different images is need. Matching process will rely on distance of the descriptors for compute the similarity between the features.When Features and Matches are known is possible estimate the parameters of the transform using the homography matrix.
Typically, homographies are estimated between images by finding feature correspondences in those images.

2.2 Brief homography computation principle

Homography is a relation between two different point, any two images of the same planar surface in space are related by homography. When homography matrix is applied to every pixel, the new image will be a warped version of the original image.According with the following theorem:
A mapping from P2 → P2 is a projectivity if and only if there exists a non singular $3X3$ *matrix H such that for any point in P 2 represented by vector x it is true that its mapped point equals* HX
Using a 3x3 homography matrix is possible compute the homography of the entire image. Although H is 8 degree of freedom matrix, by using some costrain is possible make it simpler. One good way to understand homographies is to put them into the context of other geometric transformations. The most common Homography Transformation are : Linear,Affine, Euclidean,Similarity,Projective.

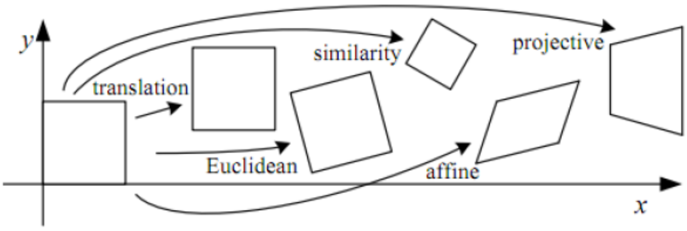


Figure 4: DIFFERENT HOMOGRAPHY

For representing the transformations with matrices has been used the homogeneous coordinate, this mean that each point it will be represented by vector compose as follow [x y 1]

2.2.1 Affine transformation

An affine transformation it is a composition of two rotations and two non-isotropic scalings. then affine transformation preserves collinearity and ratios of distances. The matrix H has 6DOF. Affine transformation is equivalent to the Parallel Plane Projection of source image on the destination image. For this reason, parallel lines remain parallel after mapping. Also the points with same distance are preserved after mapping.

AFFINE

$$\begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix}$$

2.2.2 Euclidean transformation

Euclidean transformation,is an affine transformation, that admit to compute translation, rotation and reflex,the corresponding matrix has 3DOF . Euclidean transformations preserve invariant size of the figure, they change only the figure position in the space. All Euclidean transformations are affine, but some affine transformations are not Euclidean.

EUCLIDEAN

$$\begin{bmatrix} a & -b & tx \\ b & a & ty \\ 0 & 0 & 1 \end{bmatrix}$$

2.2.3 Similarity transformation

Similarity transformation will preserve the shape of the object, hence it is a rigid motion transformations with a rescaling.In this case the matrix has 3 DOF.

SIMILARITY

$$\begin{bmatrix} a & -b & tx \\ b & a & ty \\ 0 & 0 & 1 \end{bmatrix}$$

2.2.4 Projective transformation

Finally we have the projective transformations that, as we already defined, has 2 DOF more than the affine.Projective transformations allow to do rotation, scaling, translation, shear and perspective projection. In the projective transformations two parallel lines can be transformed to two intersecting lines, this make the use of projective transformation so valuable.

PROJECTIVE

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$

2.3 Computation of the elements of the H matrix

To demonstrate how to compute the H matrix we will consider the general case that is the projective case.
We know from the theory that

$$\begin{bmatrix} kx_i \\ ky_i \\ k \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x'_i \\ y'_i \\ 1' \end{bmatrix} \tag{1}$$

$$\begin{cases} x_i = \frac{h_{11}*x'_i+h_{12}*y'_i+h_{13}}{h_{31}*x'_i+h_{32}*y'_i+h_{33}} \\ y_i = \frac{h_{21}*x'_i+h_{22}*y'_i+h_{23}}{h_{31}*x'_i+h_{32}*y'_i+h_{33}} \end{cases} \tag{2}$$

$$\begin{cases} x_i = h_{11} * x'_i + h_{12} * y'_i + h_{13} - h_{31} * x'_i x_i + h_{32} * x_i y'_i \\ x_i = h_{21} * x'_i + h_{22} * y'_i + h_{23} - h_{31} * y_i x'_i + h_{32} * y_i y'_i \end{cases} \tag{3}$$

$$\begin{bmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 & -x_1 x'_1 & -x_1 y'_1 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 & -y_1 x'_1 & -x_1 y'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n & y'_n & 1 & 0 & 0 & 0 & -x_n x'_n & -x_n y'_n \\ 0 & 0 & 0 & x'_n & y'_n & 1 & -y_n x'_n & -x_n y'_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} \tag{4}$$

It has to be notice that the result in the eq 1 is scaled by a factor k. So for having the correct point is needed to normalize by k.

3 Materials dataset

For this work, four synthetic images of a chessboard were provided. The first image was undistorted, hence was used like a reference, the other tree images were distorted with an unknown transformations. For each distorted images it has given 64 coordinates corresponding to 64 coordinate belonging to the reference image.All these features and matches points were store in a matlab file “Features.mat”.These coordinates were used to compute the homographies defining the transformation between the reference image and the other three.

4 Laboratory task

The first task of this lab was computing the different homography matrices for describing the various planar transformations of the given dataset. A matlab function "H = computeHomography(Features,Matches, Model)" has been implemented to address this task. In the system 4 it can be notice that are present eight unknowns and two equations.Hence a Least square problem has been defined for find a solution to the system .As a first approach the obtained H matrix was used for applying a backward transformation to the given matches points.Different re-projections errors were calculated in order to estimate the motion model that better fit the planar transformation between the reference and the rest of the image. After that the different backward transformations were applied to all the pixels of the image.This was done with the imwarp function.
For applying the transformation to the distorted image, it is not possible to use a forward transformation because some points will fall outside the space as well as some point of the new image-space could not find any arriving point from the distorted image. To guarantee a correct filling, H needs to be apply inversely.Hence, we apply the H matrix to its corresponding image inversely.
In order to have a visual result of the different transformations, the reference and the registered image, has been shown using overlay in different color channels. However the principal aim of this lab was applying the registration between real data, such as medical image of retina or skyn spot. Hence SIFT algorithm was applied to pair of unregistered real image in order to retrieve matches points. Afterwards the same procedure applied in the synthetic data was applied in this real data.

5 Results

In the first step were computed the different H matrices. Even if we considered four different transformations modality(euclidean, similarity, affine and projective) the resolution method applied was the same for all of them, the only differences was the number of unknowns. However in the first approach was notice fig 5 that, for the image distorted with a rescaling of the object, all the transformations lead to the same qualitative result. This result was unexpected because we know from the theory that euclidean transform doesn't allow the rescaling of the objects in the image.

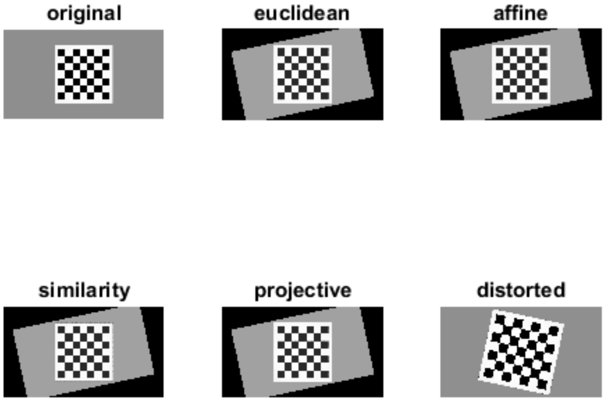


Figure 5: Wrong euclidean result

This problem was due to the fact that when we solve the system for the euclidean transformations we consider, the unknowns $a = S * sin(theta)$, like a single block without taking care of the scaling factor. So we compute an intrinsic normalization of the unknowns that has lead the result on a unwanted rescaling. For solving this problem we take in consideration that for an euclidean transformation the expected determinant is one, hence when the determinant was different from one, the matrix H was divided for a scaling factor. The result are shown in figure 6

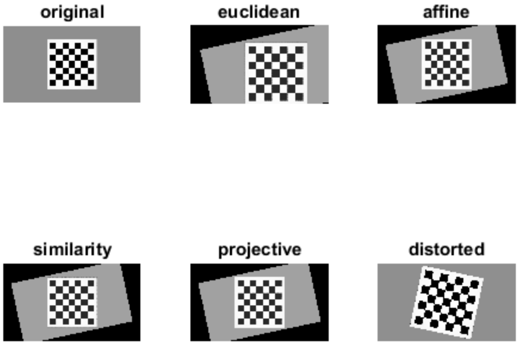


Figure 6: Corrected transformation results

As we already mentioned before some areas of the transformed image didn't correspond to any value of the backwards transformations (Black region Figure 7). The imwarp set this areas intensity to zero. The result of the visualization using overlay in different color channels r were affected by this, so we decided to fill the mentioned areas, and it was done by adding the parameter 'Fillvalue'. in the *imwarp* function.

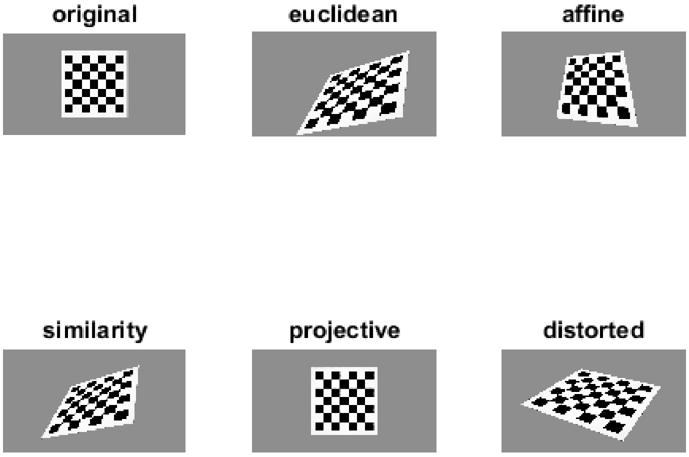


Figure 7: Final result image

In the image are shown fig. 8 the registration results for all the dataset overlay in different color channels:

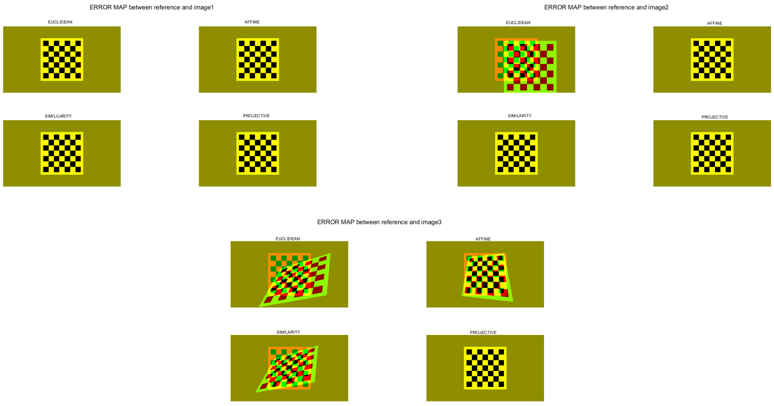

ERROR MAP between reference and image1
EUCLIDEAN
AFFINE
SIMILILARITY
PROJECTIVE
ERROR MAP between reference and image2
EUCLIDEAN
AFFINE
SIMILARITY
PROJECTIVE
ERROR MAP between reference and image3
EUCLIDEAN
AFFINE
SIMILARITY
PROJECTIVE

Figure 8: Registration results using overlay in different color channels

Compute the reprojection error was another requested task of this lab. For evaluating the error we decided to compute the H matrix with the minimum admissible amount of points, that for an 8 DOF H matrix, is 4. However when we estimated the projection we noticed that the H matrix was giving a wrong transformation. This problem was caused by the decrease of the rank of H, given by the linear dependence of the chosen points. Two solution can be adopted to solve this problem. 1)We can increase the number of points in order to ensure that the rank of the matrix is always sufficient for solving the system of equations or 2) in case we choose the minimum number of points, we must ensure that they are linearly independent.

What are the motion models that better fit the planar transformation between the reference image and the rest?
For better understanding which one are the motion models that better fit the planar transformation between the reference image and rest, all the possible planar transformation has been applied for the 64 given points in all the image of the dataset. The illustration may be found in the graph 12 9

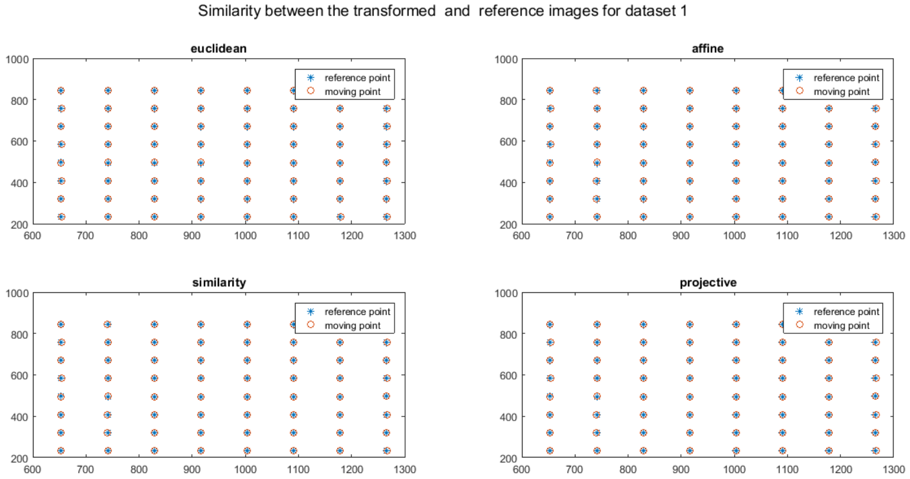


Figure 9: Registration results computed in the 64 points with a backward transformations

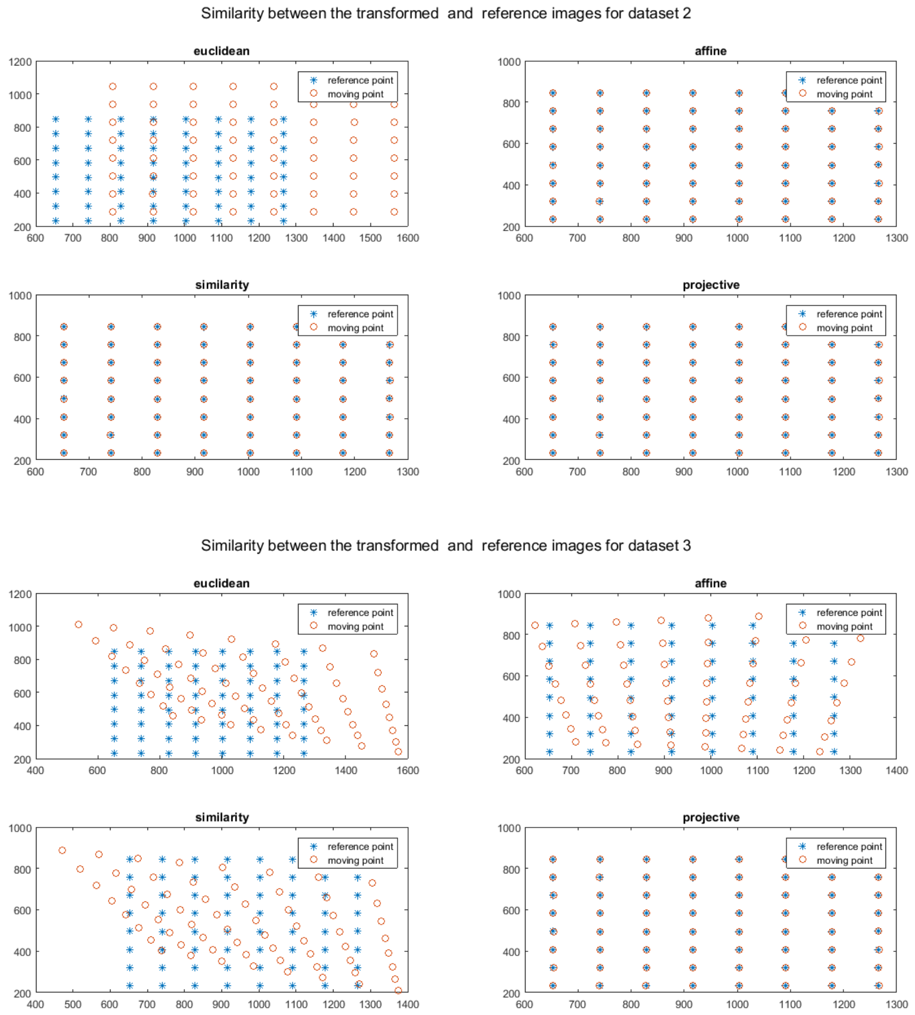


Figure 10: Registration results computed in the 64 points with a backward transformations

From the analysis conducted it has to be notice that, for the transformation between all the reference image and rest, the projective one has always reach good result. This was expected because the projective transformation has more DOF than the others. Regarding the other transformation (Euclidean, Affine, Similarity), it can be notice that for the image 1 all of them has achieve good result of registration. This happened because the image 1 was distorted with a rotation and translation. For manage this distortion it is needed a transformation matrix with just 3 DOF, but all the transformations that we consider had higher DOF. In the case 2, the image to register was distorted with a rotation translation and scaling; but we know from theory that the euclidean transformation is not able to rescale the image, for this reason the result registration that has

came out from euclidean transformation was not good.
Concerning the last case, the image 3 was disrupted with a spatial transformation. For dealing with such a distortion we need a 8 DOF matrix, thus the other transformation type cannot perform the alignment properly.

Did you notice any benefit or drawback using motion models with more degrees of freedom than the theoretically needed in some image pairs?
There is no benefit in using more DOF transformation than needed due to the fact that for an image distorted with a transformation of N DOF only NDOF transformation is required to register it back with the fixed image and any effort to compute the transformation with more than N DOF will result in obtaining the same transformation matrix, however with neutral components for every other DOF and will require more computation time and thus is irrational.
Nevertheless for the cases when one has no information regarding the distortion of the images, it is better to use a general transformation type, thus the one that has more degrees of freedom in order to ensure that any required transformation component may be computed.

The obtained homography matrices allow registering the image pairs accurately, or there are significant misalignments?

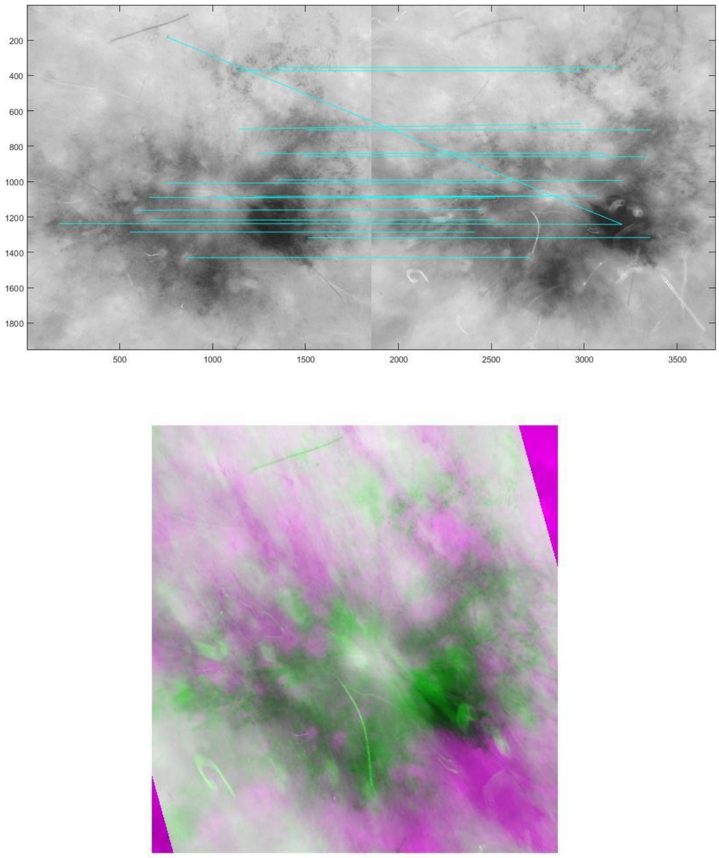


Figure 11: Registration of the real dataset

As experiments have proven, for non synthetic data, the homography transformation computed on the entire set of correspondences, transforms the image to an orientation, which is nowhere near the correct one. The cause of this misalignment becomes clear if one takes a closer look at the correspondences obtained from the SIFT algorithm. It is certain that some of the correspondences are totally irrelevant(such as the one represented by the diagonal line on the image Fig 11). There are also a few correspondences that in case of homography computation will cause image rotation by 180 degrees(inverted matches). This example introduces the need of an additional component in the registration system, that will reject the irrelevant correspondence matches between the key features of the images and will improve the quality of the registration by selecting correspondences, that produce a homography transformation that is as close to the true distortion of the image as possible.

5.1 Reprojection error

For reprojection error computation the direct transformations were also needed. The result of the direct transformations applied in the 64 given points for all the image are shown in fig:

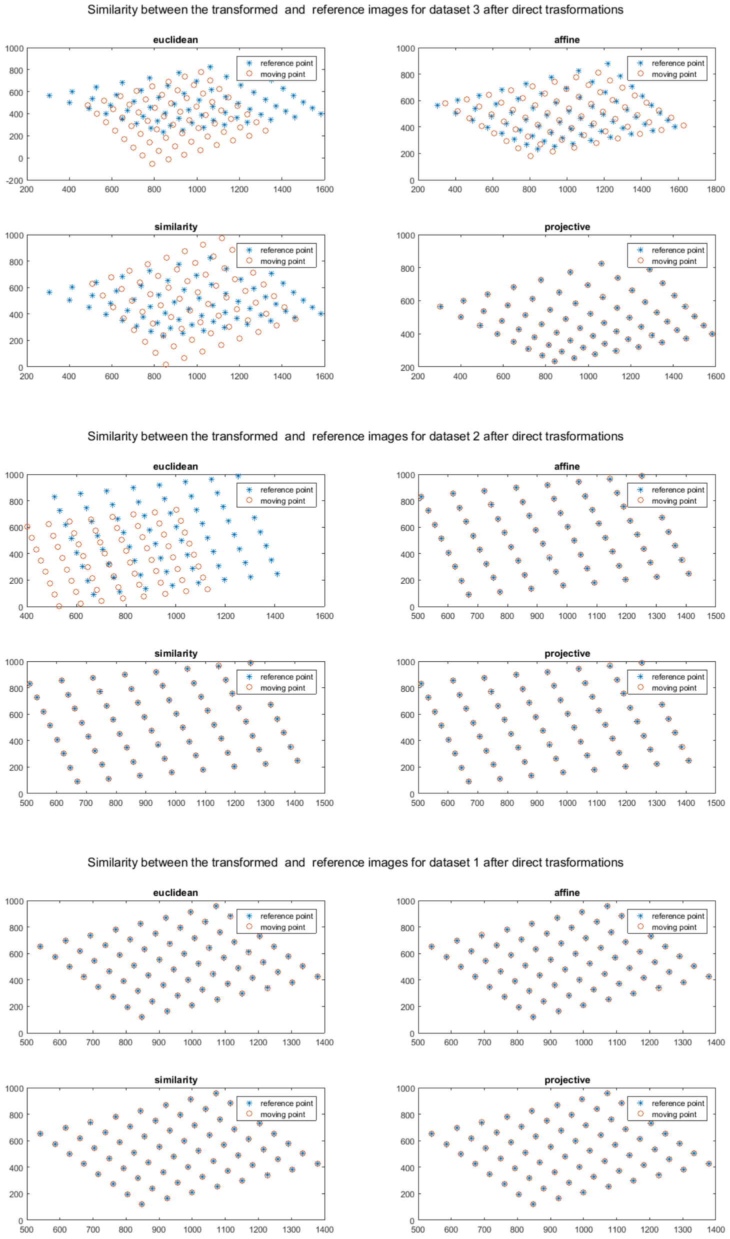


Figure 12: Registration results in the 64 points using direct transformations

After this transformation has been computed the reprojection error was estimated. In the table 1 are illustrated the reprojection errors obtained for each image (Column) for all the transformations (Row).

Table 1: Reprojection Error

	Img 1	Img 2	Img 3
euclidean	0,0108	$25474 * 10^9$	$79049 * 10^8$
affine	0,0030	0,0014	$59186 * 10^5$
similarity	0,0032	0,0016	$73467 * 10^7$
projective	0,0031	0,0009	0,0085

6 Random sample consensus correspondence rejection algorithm (RANSAC)

[Hartley and Zisserman, 2003] Independent of how perfect the SIFT algorithm was tuned, there is always a need of filtering the correspondences between key features of the moving and fixed images, that where obtained during its execution. This is due to the fact that the error of the matching key feature detection may never become zero, no matter how good the descriptor of SIFT or any other algorithm performs. This involves a need of building an additional component for the image registration task, that will perform rejection of irrelevant matches before computing the homographic transformation and aligning the images.

6.1 Introduction

Let the SIFT algorithm produce N correspondences between N feature points of the moving image and N matching points of the fixed. Depending on the task and the input data One may estimate the average percentage of outliers(irrelevant correspondences) by performing several experiments.

In order to build an algorithm that rejects this percentage of correspondences, we have to consider a justification measure for the determining if a certain point has to be considered an inlier or an outlier. It is clear, that by qualitative analysis of the matching points, the better correspondences can be determined, because it can be seen that one match brings a point of image A inside a certain object A_1 to an absolutely different object B_2 of image B, so if one tries to compute the transformation that aligns these two points, the objects A_1 and B_1, A_2 and B_2 will be misaligned, and thus these matching points have to be considered wrong. However, we have to know which corresponding matches are better by introducing some quantitative measure that describes their relevance.

6.2 Similarity measure

To quantitatively measure how close are two sets of points to each other (key points of the fixed image and the key points of the moving image) after the computed transformation in order to classify this transformation as good or bad, the best approach was to consider the reprojection error for quantifying the error in each of the two images involves estimating a “correction” for each correspondence. It gives the answer to the question how much it is necessary to correct the measurements in each of the two images in order to obtain a perfectly matched set of image points.

In our case, we are seeking a homography H and pairs of perfectly matched points x'_i and x_i that minimize the total error function

$$\sum_{i=1}^{n} d(x_i, x'_i)^2 + d(x'_i, x''_i)^2$$

where

$$x''_i = Hx'_i$$

Minimizing this cost function involves determining both H' and a set of subsidiary correspondences x'_i and x''_i. This estimation models, for example, the situation that measured correspondences x_i - x'_i arise from images of points on a world plane. We wish to estimate a point on the world plane X'_i from x_i - x'_i which is then reprojected to the estimated perfectly matched correspondence x'_i - x''_i. In our case as the d distance function the Euclidean distance was chosen.

6.3 Algorithm

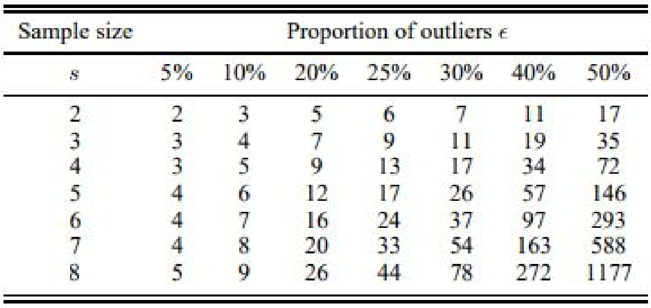
Random sample consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates.

The input of the algorithm is the whole set of key features of moving image A and the matching key features of fixed image B. The goal is to determine a subset of these correspondences, that produces the best fitting homography transformation for image alignment in terms of minimizing the reprojection similarity measure described in section 3.2.

The algorithm iteratively chooses constant size random subsets of matching points from the entire feature set of the input, computes the homographic transformation H according to this chosen subset. Afterwards the transformation H is applied on the entire set of the key features of the moving image and the similarity of the transformed set with the set of matching features of the fixed image is computed. The number of iterations has to be chosen to be big enough to ensure that all possible combinations of matches are covered during random selection because in theory the randomization function may not give two identical outputs while being executed a number of times less than the possible number of outputs.

Nevertheless, it is often computationally infeasible and unnecessary to try every possible sample. Instead the number of samples N is chosen sufficiently high to ensure with a probability, p, that at least one of the random samples of s points is free from outliers. Usually p is chosen at 0.99. Suppose w is the probability that any selected data point is an inlier, and thus $e = 1 - w$ is the probability that it is an outlier. Then at least N selections (each of s points) are required, where $(1 - w^s)^N = 1 - p$, so that

$$N = log(1 - p)/log(1 - (1 - e)^s).$$



Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Figure 13: The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers e.

In this way we iterate N times, selecting random subsets of feature matches and afterwards we aim to find the subsets, for which the smallest reprojection error has been obtained. It is possible to stop at that point and just choose the subset with the smallest corresponding reprojection error, nonetheless it would mean fixing the number of inliers to the size of the random subset. Most certainly it would produce a quite good result regarding the optimal transformation of the image, however the goal of RANSAC is to reject outliers, but not find the smallest subset of correspondences that produces the best transformation matrix.

This introduces the need of building a biggest subset of inlier correspondences, that produce a homography transformation, that results in a reprojection error, that lies below a given threshold.

This has been done by selecting a certain percentage of best correspondences as inliers, but only in the case if they produce a reprojection error smaller than a given value. If it is known that for certain image modalities usually a constant percentage of outliers appears, than there might be no need also to set the reprojection error threshold.

In our case the distance reprojection error threshold was chosen empirically by making experimental measures on our dataset, however it may also be computed automatically for the case when the error values have known to be of Gaussian distribution. Thus, if the mean of the error is 0 and the standard deviation is σ^2, for homography computation the distance error threshold should be set to $5.99\sigma^2$.

6.4 Normalization

Without normalization typical image points x'_i, x_i, that may be quite large due to a big size of the images, may cause to problems of arithmetic precision and accuracy during the operations with matrices, required to compute the homography transform. E.g. for points of order 10^4 the multiplication of two matrix elements will be of order 10^8. In the case of division by a large number, one may find the situation close to a singularity, etc. Thus it is always better to work with a normalized feature space, which is independent of the size of the images, that are being aligned.

The normalization is performed in the following steps:

1. Translating the feature space to the centroid of all features

2. Normalizing the features by their norm (so that a length of a point vector of any feature is less or equals to 1

However the step of normalization introduces also the need of converting the obtained homography transformation so that it may be applied for image alignment. The homography transformation matrix is invariant to the normalization except for its translation components. So in order to make the denormalization of the homography matrix, it is needed to take into account only this.

6.5 Results

The introduced outlier rejection algorithm RANSAC has proven to significantly improve the registration of the images. Bellow one can see the alignment of the images before and after applying the rejection algorithm. It is clear that without rejection the result of alignment is not satisfying due to some irrelevant correspondences. Obviously, the registration after RANSAC is much more accurate. In the example below, the RANSAC rejected 5 out of 19 correspondences, so the final homography matrix was computed with 14 correspondence matches.

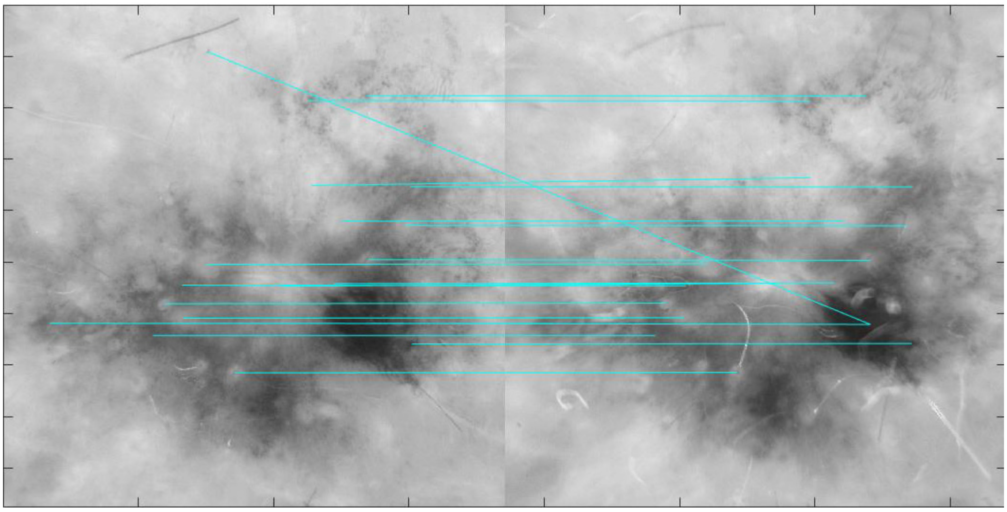


Figure 14: Correspondence matches between fixed and moving images of the skin resulting from Lowe's SIFT algorithm

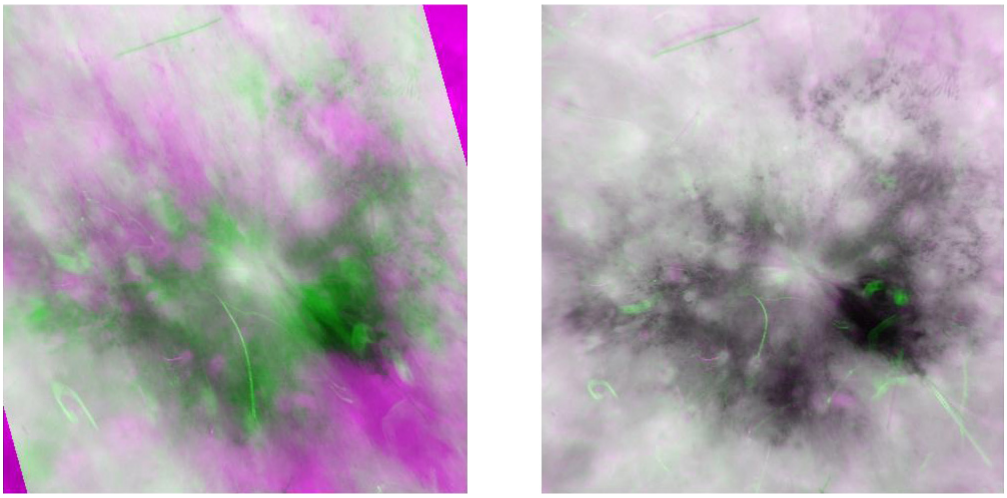


Figure 15: Difference between registration of images of skin without(left) and with(right) the application of RANSAC correspondence rejection algorithm

The RANSAC algorithm results in a significant reduction of the reprojection error (for this dataset the first alignment was nowhere near the correct transformation).

For the other provided dataset of retinal images the results where not that satisfying due to a big difference of the images as well as the fact that original Lowe SIFT provides only 5 matching correspondences between these images, 2 of which are incorrect and the other 3 are linearly dependent as one may see on the image bellow.

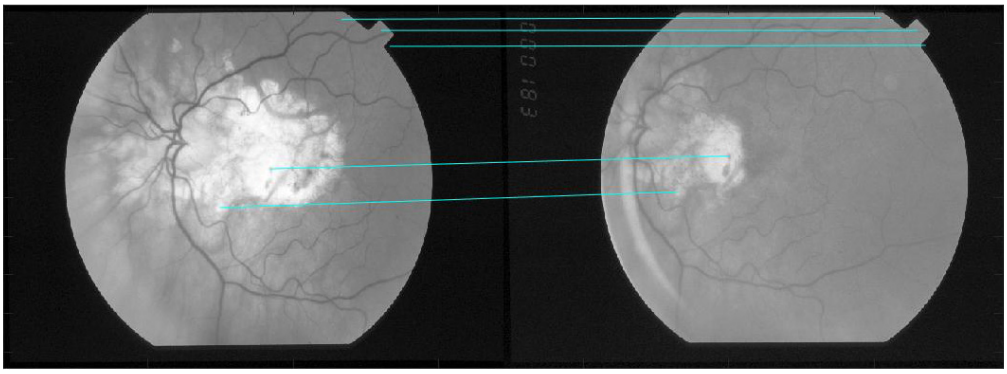


Figure 16: Correspondences between key features of the retinal images provided by Lowe's original SIFT algorithm

There is an issue that is being faced. In some cases, where only a few correspondences are found between images, there is a high chance of a big percentage of correspondences being irrelevant. Such as in the presented case the bottom two correspondences we consider to be wrong. This leads to having only 3 correspondences, some of which may also be linearly dependent. As we know from the previous chapters, that for homography computation in the most general (projective) case, 4 points are required, for the affine transformation - 3 points, for the similarity and euclidean transformations - a minimum of two points. Which leaves the idea of always using the most general(projective transformation) case for homography computation behind. So ideally, the algorithm has to reduce the DOF of the homography matrix in case of obtaining a number of correspondences lower than 4. However, if even the better correspondences are linearly dependent, we are left with an unsolvable case. Below one may find the result of registration before and after applying RANSAC using the similarity homography transformation.

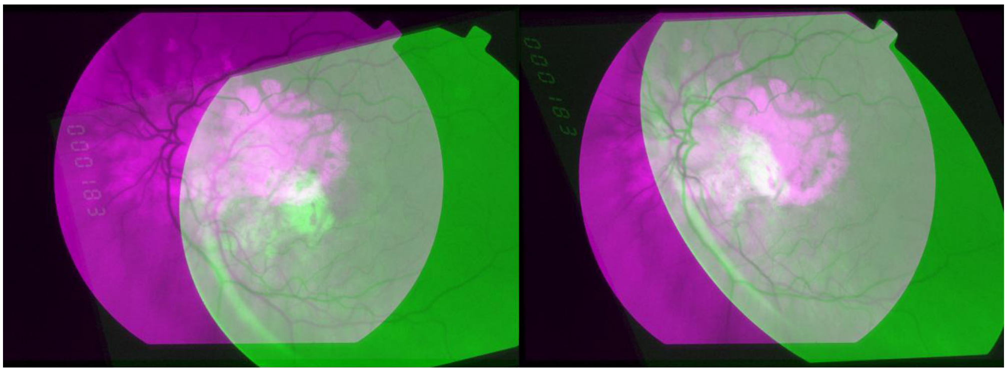


Figure 17: Difference between registration of retinal images without(left) and with(right) the application of RANSAC correspondence rejection algorithm

The normalization of features in our case did not result in any kind of quality of registration improvements, and the reprojection error remained the same, however it is worth noticing that the images, that were used for experiments are not of a very big size(1854x1953 pixels maximum), which may explain this phenomenon.

7 Conclusion

The original implementation of Lowe's SIFT algorithm was explored as well as its third party implementation by VLFEAT library. It became clear that it is not possible to precisely tune the VLFEAT implementation to match the results of the Lowe's, however this third party implementation gives much more flexibility in terms of tuning and for some cases may be crucially better than the original one.

The homographic transformations were computed for all images, as well as synthetic data. Four types of transformations were considered: euclidean, similarity, affine and projective. From the conducted experiments it was concluded that in most of the cases it is better to use the transformation with the greatest number of degrees of freedom, however for the case of a small generated number of correspondences, or a case of linear dependence of points it is necessary to decrease the number of the degrees of freedom and so use one of the more basic transformation types.

In order to deal with irrelevant correspondences the RANSAC outlier rejection algorithm was implemented. It have proven to work quite well for the dataset, that caused SIFT to produce a sufficient number of quality correspondences, however for the retinal images, the computed correspondences where not enough to produce a correct registration even after outlier rejection. This was caused by a small number of matches and the fact of linear dependence of most of them. In this case a good result could not even be produced by computing the most simple, euclidean transform.

Bibliography

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.