Medical Image Registration and Applications

Lab on

Rigid Image Registration

Rafael Garcia, Nuno Gracias

Lab sessions: 2
Required effort: 8 hours
Programming platform: Matlab

Version: 2

Universitat de Girona

1. Objective

The aim of this lab is to develop competences regarding teamwork and problem solving. By developing the proposed activity you will also become familiar with SIFT and planar transformations for rigid image registration: how to extract invariant features, how to describe them, how to match them and how to use them to compute a homography. Most importantly, this should give you some feeling about the strengths and weaknesses of local feature-based approaches.

It should be noted that before you start this activity, you should read Lowe's paper on SIFT:

> *David G. Lowe,* ***"Distinctive image features from scale-invariant keypoints,"*** *International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.*

The paper can be downloaded from here.

This lab requires working in teams of two (exceptionally three) students. The labs will be organized on the first lab session by the professor who acts as lab instructor.

Students will have to:

1. **Test** Lowe's implementation on a set of images showing some skin lesions and compare the obtained results with the provided by a third party library.
2. **Register** the images pairs corresponding to the same skin lesion using Lowe's implementation to detect and match features, and implementing different motion models by estimating homography matrices.
3. **Improve** the registration accuracy by means of data normalization for the homography estimation.

2. Test Lowe's SIFT vs Third Party Implementation

David Lowe, the researcher who proposed SIFT, has available on his website the binaries of his feature extraction (DoG) and description (SIFT) algorithms ("SIFT demo program"). The package also includes both Matlab® and C code to visualize the features extracted from an image and to find matches between features found in two different images.

However, in this lab session, we are not only going to study the behavior of Lowe's implementation of SIFT, but also to compare it with a third party library, as well as to learn how to use matched features to rigidly register two images.

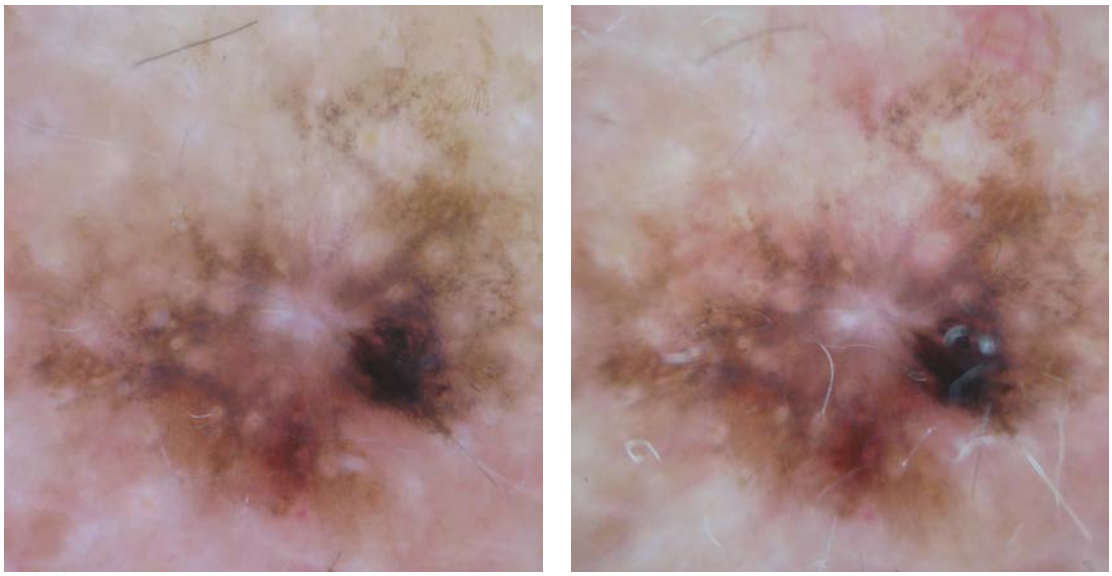
The datasests that will be used consist of image pairs depicting the same tissue area but acquired at different times, or pairs of images covering complementary areas. Registering the images will enable further studies, such as the evaluation of the skin lesion evolution over time, or building a photomosaic.

Step 1

Download the package siftDemoV4.zip "SIFT demo program (Version 4, July 2005)" from Lowe's web page (Windows only).

Step 2

Download and unpack the image datasets stored in the intranet of the course (file **DataSet00.zip**). An example of one of the available image pairs can be seen in **Figure 1**.



(a) (b)

Figure 1: **(a) Skin lesion imaged at t_1 with a Canon DSLR camera and (b) same lesion imaged at t_2 with a Nikon DSLR camera. Both images were taking using the same Dermlite dermoscope.**

Step 3

Unpack the **siftDemo.zip** file and run in Matlab the **sift.m** function, after appropriately converting the available input images to PGM format. Get used to the parameters of the function and display some detection results.

Step 4

Browse the web to find third party implementation of SIFT, preferably one which can be used from Matlab, in order to make both versions easy to compare. One of the available and well-known Matlab implementations is VLFeat, but any other can be used. Get used to the library interface and try to replicate the results obtained by Lowe's binary.

Questions

Display side-by-side the results obtained by both implementations. Are they identical, or do they show some differences?

Is it possible to replicate the results from Lowe's implementation with the third party one by tuning some parameterization? If not, what could be the reasons for these behavior differences?

Report the tested parameters and the impact of those on the obtained results.

3. Register Image Pairs Estimating Homography

Introduction

Once a set of features have been detected on both images of a given pair, those can be associated according to the similarity of their descriptors in what is known as the matching step. This association will mainly rely on computing the distance between the descriptor vectors of each feature to determine the best candidate to represent the same feature in both images.

When this association has been done, the location of these feature pairs (or matchings) can be used to estimate a homography matrix, describing the transformation of the plane containing them between the two views.

Nevertheless, during the matching process, some feature descriptors can be missassociated, due to different factors affecting the appearance of the scene, which can lead to wrong correspondences, also known as *outliers*, which will disturb a correct homography computation. Consequently, using an outlier rejection strategy, such as RANSAC, is a required step.

Step 1

Implement a Matlab function that, given two Nx2 matrices storing the coordinates of features and correspondences ([x, y] and [x', y']), and a string specifying a transformation model (Euclidean, Similarity, Affine and Projective), computes the homography matrix that describes the planar transformation. The homography matrix has to be computed by solving the equation systems presented in the lecture notes. The implemented function will have the following shape: **H = computeHomography(Features, Matches, Model)**.

Step 2

Unpack the **DataSet01.zip** file available in the intranet of the course. Use the feature coordinates stored in the **Features.mat** file to compute the homographies defining the transformation between the reference image **00.png** and the other three, that is, **01.png**, **02.png** and **03.png**. Identify the motion model that better fits the transformation of each image pair. Use the Matlab function **imwarp** to map and later visualize the result of the registration. This step, based on synthetic data, will allow verifying that the homography matrix for the different motion models has been correctly computed.

Questions

What are the motion models that better fit the planar transformation between the reference image and the rest? Visualize the registration results using overlay in different color channels (red for the reference image and green for the other, for instance). Did you notice any benefit or drawback using motion models with more degrees of freedom than the theoretically needed in some image pairs? In order to evaluate the accuracy of the homography matrices computation, use the reprojection error as error measure.

Step 3

Generate the lists of features and correspondences for the images in **DataSet00**, using the correspondences suggested by the **match.m** function, compute the corresponding homography and display the results using overlay.

Questions

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

4. Improving the Registration Accuracy

Introduction

During the matching process, some descriptors can be miss associated, due to different factors affecting the appearance of the scene, which can lead to wrong matches, also known as *outliers*. These wrong correspondences will prevent an accurate homography computation. Consequently, using an outlier rejection strategy, such as RANSAC, is a required step.

Aside from the outlier rejection, appropriately conditioning the data to increase its numerically stability, using data normalization, is another strategy to take into account.

Step 1

Improve the Matlab function that computes the homopgraphy matrix adding RANSAC outlier rejection to the pipeline. The new function will have the shape: **H = computeHomographyRANSAC(Features, Matches, Model)**.

Step 2

Compute the homography matrices for the previously tested image pairs, and compare the results before and after the use of RANSAC. Visualize the results using overlay and compute the corresponding reprojection error.

Step 3

Implement a data normalization method to better condition the data before the homography computation. A small literature review (or web browsing) might be required. Repeat again the previous experiment using the normalized data, computing also the reprojection error.

Questions

Are the results obtained with the RANSAC-based homography estimation more accurate than the previously obtained? The use of data normalization helps reducing the reprojection error?

5. Methodology

You will carry out this activity in groups of two. You need to decide how you are going to address this activity and how are you going to solve the problem. The whole group should meet and report regularly to the professor.

6. Deliverable

At the end of the activity, every group should give a report and deliver the code that has been developed. The report will consist in a brief introduction to the problem, and a section corresponding to each of the steps described on this document. All the questions should be explicitly answered. The report will be preferably done using single column layout, in order to allow an easier placement of the images and figures that you will produce during this exercise.