



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

**IMAGE STITCHING, COLOR CORRECTION
AND SUPERRESOLUTION OF A VIDEO
SEQUENCE FOR DERMATOLOGICAL
ANALYSIS**

Pau Ortega Garcia

Director: Simone Balocco
Realitzat a: Departament de Matemàtica Aplicada i Anàlisi
Barcelona, 22 de juny de 2017

Abstract

Skin cancer is the most common cancer on humans, some kinds of skin cancer can be deadly if not addressed correctly and in time. The diagnosis is usually stated based on the appearance of the lesions in the skin.

This paper describes how to apply stitching techniques to images extracted from a video acquired with a smartphone.

The method has been implemented in Matlab, and it has as objective to obtain a panoramic image of the patient skin. The goal of this project is to overcome to the resolution limitation of a typical smartphone, by applying super-resolution and colour correction techniques. The quality of the result is measured qualitatively using different methods at the merging of the images

The use of VLFeat library is required to develop and follow the steps explained in this paper, the compilation of this library and how to address any error in the installation process.

The panoramic image does not provide any diagnosis of the illness but is meant as a pre-processing step for an automatic tool aimed at helping the dermatologist to diagnose skin cancer.

Contents

1. Introduction	
1.1 Skin cancer	1
1.2 State of art	3
1.3 Project goal	5
1.4 Gantt Chart	8
2. Datasets	9
3. Methods	
3.1 Image registration	9
3.1.1 Feature based methods	11
3.1.1.a Harris-Stephens corners	11
3.1.1.b SIFT: Scale-invariant feature transform	13
3.1.2 Manual annotation	14
3.2 RANSAC	15
3.3 Stitching	17
3.4 Super-resolution	17
4. Development	
4.1 Naïve stitching approach	18
4.2 RANSAC for manual keypoint picking	19
4.3 VLFeat SIFT	20
4.4 Colour correction	21
4.4.a Plane of differences	21
4.4.b Histogram matching	23
4.5 Mosaicking	24
5. Results and discussion	
5.1 Harris-Stephens corner detection	25
5.2 SIFT: Scale-invariant feature transform	26
5.3 SIFT matching	27
5.4 RANSAC SIFT and Homography	28
5.5 RANSAC homography through manual annotation method ...	30
5.6 SIFT and manual keypoint annotation comparison	31
5.7 Mosaic display	33
5.8 Colour correction	36
5.8.a Planar colour correction	37
5.8.b Histogram matching	40
5.9 Super-resolution qualitative evaluation	41
6. Conclusions and future work	44
Annex	
A.1 VLFeat compilation	50
References	51

List of figures

- Figure 1: Types of skin cancer and how they usually look like
- Figure 2: ABCDE criteria for diagnosis
- Figure 3: On the right device and user interface for Metaoptima product, on the left a universal Dermlite smartphone device
- Figure 4: SkinVision app interface for different steps in the process
- Figure 5: Google Play Store applications for melanoma analysis
- Figure 6: iOS App Store applications for melanoma analysis
- Figure 7: Left image shows the Nokia 1020, right shows the Samsung K zoom smartphone
- Figure 8: Motorola's modular camera enhancer, the background device is a Moto Z which allows to have modular enhancements clipped to its back
- Figure 9: Smartphone sensor size comparison
- Figure 10: Camera sensor size comparison up to Full Frame size, on the figure above the largest would be on the bottom row here.
- Figure 11: Figure shows the frames of the video and the region they add over the initial frame i cropped by colour lines.
- Figure 12: A real-world object seen from two different perspectives map differently, parallel lines from each perspective do not remain parallel but they do not morph into curves
- Figure 13: Steps to the image registration process
- Figure 14: Auto-correlation principal curvature space-heavy lines give corner/edge/flat classification, fine lines are equi-response contours.
- Figure 15: Example of Harris features from Matlab's function.
- Figure 16: Strongest SIFT features in the image, with the scale on yellow and the direction on the green matrix
- Figure 17: This figure displays the model space M as a green surface (the locus for which $f_m(d; \cdot) = 0$). The yellow surfaces represent the boundaries for a datum to be considered an inlier (assuming that the distance function is the Euclidean distance, hence the smallest distance between any two points on the yellow surface and green surface is d). The structure of the green surface is both defined by the model M and by the parameter vector. The inliers, represented as blue dots, lie in between the two yellow "crusts".
- Figure 18: Image stitching example using a set of images of a cathedral to form an image showing the whole architectonic piece.
- Figures 19,20 and 21: Top left image shows the averaging of the two images, the boundary of the first image is clearly seen. On the top right, the difference between both images is displayed, there's a line displayed, that means it exists on one image but not in the other. On the bottom row the maximum value of each pixel is taken, so there's no faded regions

- Figure 22: VLFeat keypoints (blue) superimposed to D. Lowe's keypoints (red).
- Figure 23: Matching point deviation between Lowe's and VLFeat implementation from the centre of the feature
- Figure 24: Pipeline to creating the plane of intensity differences, this is needed to make an equalization of the colours on the target image due to the illumination difference between frames.
- Figure 25: On the tiles with an A will appear the values from the base image, whereas the ones with the B will be filled with values from the target image
- Figures 25, 26 and 27: Example of the steps to display a chequerboard mosaic.
- Figure 28: In this figure, the left side of I1 is stitched to the right side of I2, different blending algorithm results have been added to see the artefacts they produce.

1. Introduction

1.1 Skin cancer

Skin cancer is the most common type of cancer, developed due to an uncontrolled growth of abnormal skin cells between the superficial and the deeper layers of the skin. Each year in the U.S. over 5.4 million cases of non-melanoma skin cancer are treated in more than 3.3 million people (Skincancer.org, 2017) [1]. This cancer occurs when the DNA of skin cells is damaged and the repair mechanisms available fail to recover normal genetic information, leading to a mutation. When these mutations affect growth control, cells can grow rapidly forming malignant tumours. Nevertheless, some of suspicious moles turn out to be benign, therefore it is essential to identify and differentiate them from malign growths. Benign tumours grow slowly compared to malign and can be removed through a surgical operation.

The probability of developing metastasis (spreading and growing of malignant cancer cells in other regions) depends on the type of skin cancer. Melanoma is the most common type among humans, but basal cell carcinoma (BCC) and squamous cell carcinoma (SCC), Merkel cell skin cancer and Kaposi Sarcoma are also a threat to population, and all of them can show different symptoms.



Figure 1: Types of skin cancer and how they usually look like [1]

Diagnosis is usually based on the appearance of the lesion, but patient's clinical history is also considered. In order to confirm the disease and further investigate the specific case, a biopsy may be required.

BCC and SCC, are known collectively as non-melanoma skin cancer and are often caused by sun exposure, although several hereditary syndromes and genes are also associated with an increased risk of developing these cancers. While BCC is settled on the deepest layers of the dermis, SCC lays on the most external ones.

BCCs hardly ever metastasize beyond the original tumour place, but when it happens it becomes a serious life threat. SCCs may appear on all areas of the body including mucous membranes, but the usual spot where it appears is in areas where the exposure to sunlight is direct and sustained for extended periods of time.

Melanoma is one of the most dangerous skin cancer types because it naturally spreads to other parts of the body through the circulatory system and because it is the one which causes higher death rates. If diagnosed and treated in its initial stages, the chances of an effective treatment and cure increase. Stages are measured by thickness, depth of penetration and the probability for the melanoma to spread. Stage 0 tumours have not yet penetrated to layers under the skin, while stage 1 are around 1mm under the surface but are not ulcerated. This stage is considered an intermediate stage. Melanomas in stages 3 and 4 have metastasized to other parts of the body distant from the original tumour, at these stage tumours can be found in internal organs (Skincancer.org, 2017) [2].

Melanomas are characterized by asymmetry, border irregularity, colour variation, a diameter of more than 6 mm, and evolution. This is known as the ABCDE criteria for melanoma diagnosis [3].



Figure 2: ABCDE criteria for diagnosis [2]

Asymmetry is identified when there is a difference between the two halves of the mole when a line separating them is drawn. A malign mole has uneven edges. In melanoma, the mole does not stay in one plain colour, it has different shades of brown, or can even have black portions. Benign moles are usually smaller, but when a melanoma has been detected at an early stage it may still be small. When a mole changes in any way it is a symptom of great transcendence and medical assistance must be taken.

1.2 State of art

Melanoma diagnosis is performed by a physician through a visual inspection. This method is slow and patients often suffer long queues to get an appointment with the doctor. Therefore, it is interesting to use a software that accelerates the process by helping the doctor to visualize and analyse the evolution of melanomas. It can process the data generated by one or more cameras that portrait the moles.

On the one hand, some companies such as Metaoptima¹, have developed devices to increase the detail level on smartphone cameras to later analyse the information retrieved by it. Metaoptima's device improves the camera on the smartphone by giving it the ability to make optical zoom instead of the digital zoom included in most smartphone cameras, also by including some lights in the device help obtaining subdermal images. DermLite² also has a device to clip over smartphones cameras that enables patients to monitor moles and skin lesions at home and send quality images to their doctor to follow them up. The DermLite DL1 basic device enables patients to capture crystal-clear, polarized dermoscopic images of both pigmented and nonpigmented skin lesions. DermLite has developed an app to go along with their hardware, currently giving support to iOS and Android devices. The app lets a patient capture dermoscopic photos and assign a localization on a virtual mannequin.



Figure 3: On the right device and user interface for Metaoptima product, on the left a universal DermLite smartphone device. Images were extracted from the product's web site.

¹ <https://metaoptima.com/>

² <https://dermlite.com/>

On the other hand, other companies such as SkinVision³ aim to track the evolution of moles by only using the camera on the smartphone. Every time a mole is analysed the user gets feedback on the recommended actions to take, so users can track the evolution and when they have a doctor's appointment results can be easily shared.

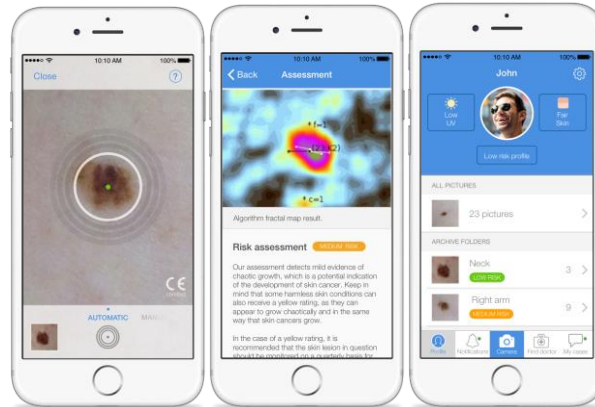


Figure 4: SkinVision app interface for different steps in the process. Images were extracted from the product's web site.

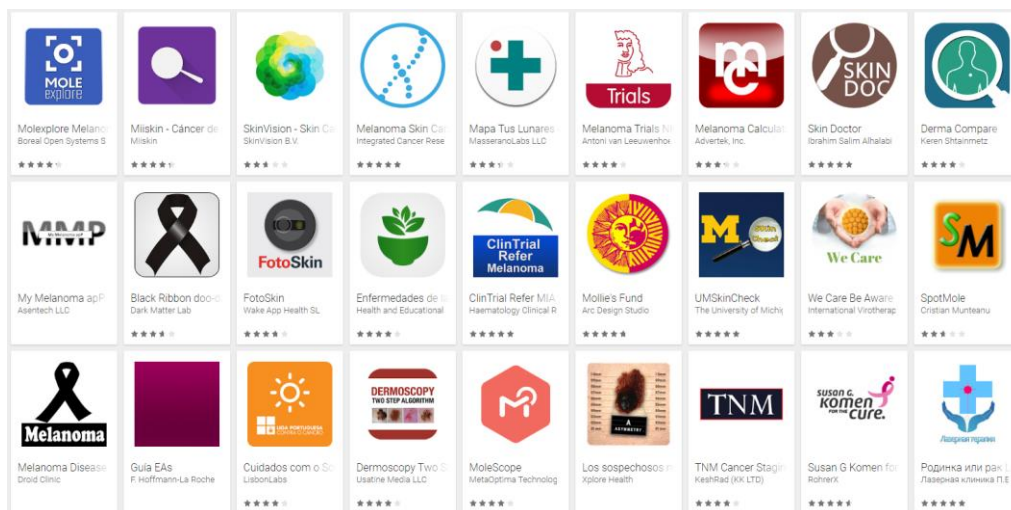


Figure 5: Google Play Store applications for melanoma analysis

³ <https://skinvision.com>

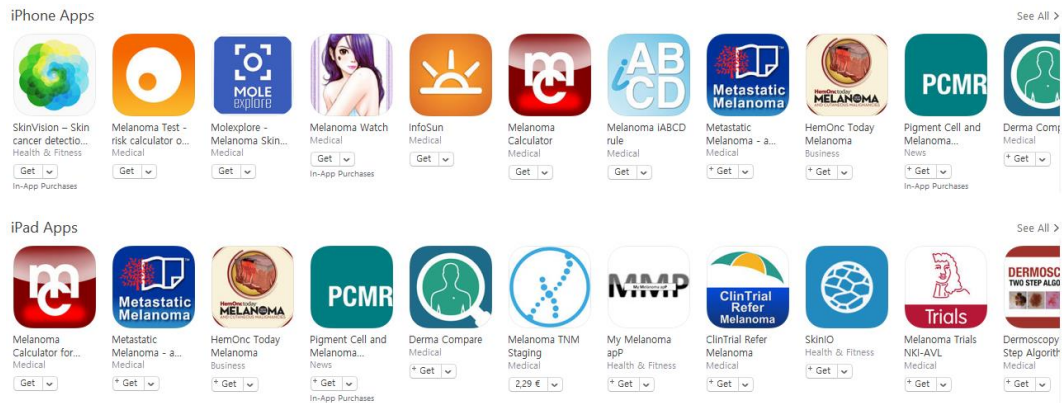


Figure 6: iOS App Store applications for melanoma analysis

The results given by all these devices and applications do not exclude the need of a doctor to check the proposed diagnosis by visualizing the moles and claim a valid statement on the final diagnosis.

1.3 Project goal

The fact that melanoma is the deadliest skin cancer becomes the main motivation of this project, which explores the possibility to track a method that analyses melanoma moles. What is more, it does not follow the previously presented applications, but is different from them because they base the diagnosis in just a single picture of the mole.

This project aims at stitching a sequence of images taken from video recording. By doing this, the ABCDE process is completed, because both the evolution of pre-existing moles and the appearance of new ones is monitored. What is more, not a single image of each mole is used, but the combination of different images and the following stitching process. It offers super-resolution methods to improve the quality of images, required because top rank smartphones cameras' sensors are tiny compared to the average DSLR cameras ones. Smartphones cannot carry bigger sensors because nowadays standards say that portability, size and thinness are the main priority when designing a new device. Some brands tried to make a difference including larger camera sensors to but failed. Other brands, such as Motorola, preferred to create modular cameras which could be attached to selected devices.



Figure 7: Left image shows the Nokia 1020, right shows the Samsung K zoom smartphone. Images were extracted from the product's web site.



Figure 8: Motorola's modular camera enhancer, the background device is a Moto Z which allows to have modular enhancements clipped to its back. Image was extracted from the product's web site.

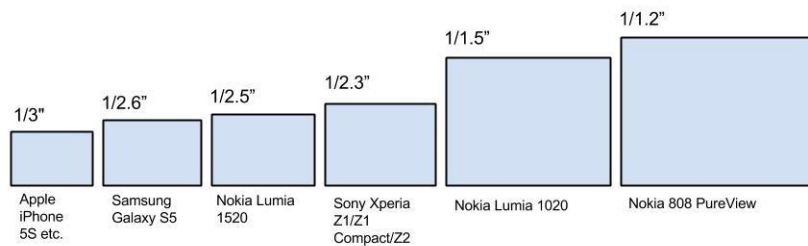


Figure 9: Smartphone sensor size comparison (Litchfield, 2017) [4].

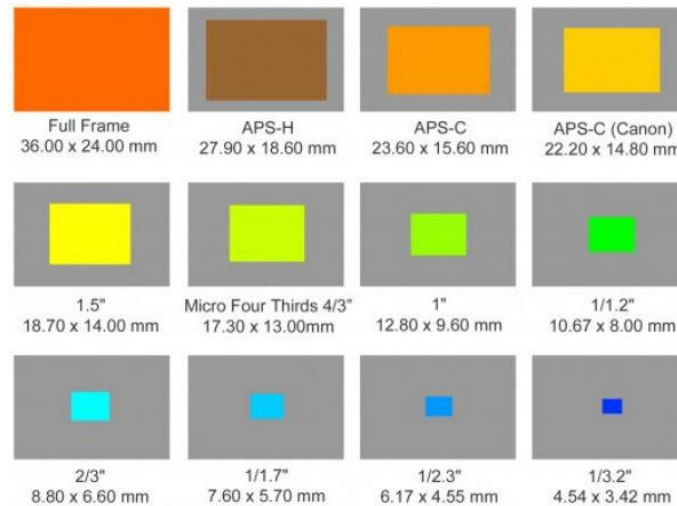


Figure 10: Camera sensor size comparison up to Full Frame size, on the figure above the largest would be on the bottom row here (Crisp, 2017) [5].

The smaller the sensor, the highest chances of getting blurry and noisy images, this issue is more noticeable in low light conditions. The size of the pixel is also relevant, as it gives the camera its resolution. The higher pixel density and the smaller pixel size offer the possibility to clearly represent smaller objects. Larger pixels receive more information, and the output contrasts all this information, while smaller pixels don't get so much information (N. Clark, 2016) [6].

This issue can be solved with a large aperture, which refers to the amount of light the lens allows to reach the sensor. Aperture is measured in an inverse scale, the highest the number the less light gets to the sensor. The problem with large apertures is that the images must be taken rather close to the subject, because if the aperture is set to large it will blur the background and if the subject is far from the camera it can also be blurred.

To sum up, a small sensor size combined with a large density of pixels and not optimal lighting settings makes that the chances of images captured with noise on them becomes large, what is called sensor noise. Quantization noise happens when analog values must be casted into digital values, thus making the image less accurate when compared to reality, and so it is different from the sensor noise.

Sensor noise is not always present in video or photograph sequences. Therefore, we can remove it by combining some images, obtaining accurate pixel intensity values. By combining the images with distinct levels of opacity in each layer, we can better define objects boundaries, or in this case moles, for later inspection.

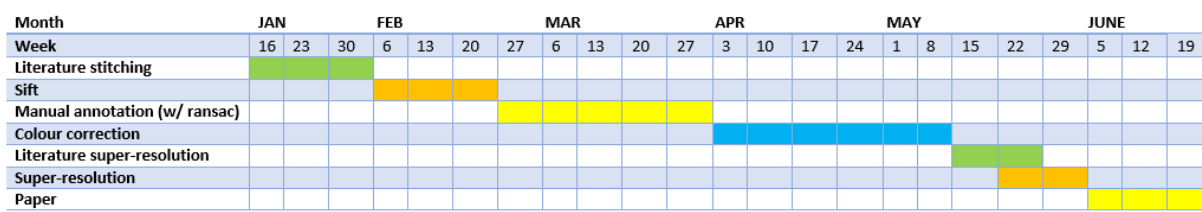
This project is part of a bigger project which will have the most common features from this kind of applications plus panoramic imaging.

The result of this study, hopefully will make the diagnostic process easier for doctors and reduce waiting lists. It does not mean that this project aims to be a diagnostic tool that makes the periodic appointments unnecessary. It is a prevention tool aiming to make people cautious and help them be aware of life threatening diseases by analysing in an inexpensive way the evolution or lack of evolution of the moles of their skin.

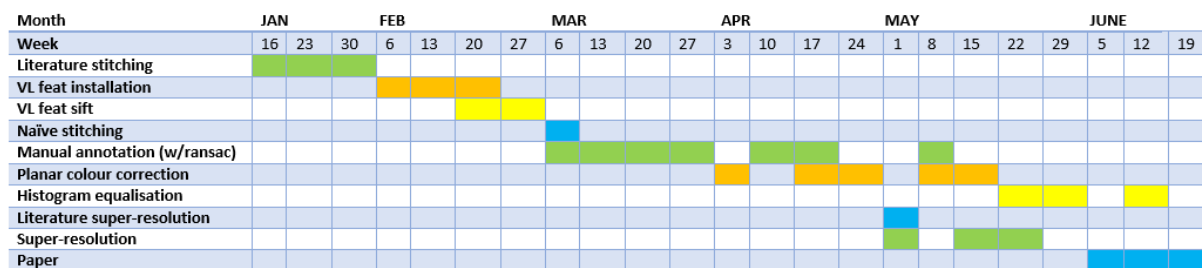
1.4 Gantt Chart

In these charts, the organisation and time of every task concerning the project has been described. There are two charts: The first shows how when the different parts of the project where supposed to take part, and the second one shows the real current evolution and the different tasks followed; The second shows the actual evolution and the tasks made in the project.

Ideal Gantt chart:



Real Gantt chart:



As it is seen in the real Gantt chart, some stages took more time than expected. The main delay took place prior to the development process, when compiling and installing the VLFeat library. When a task is considered done it shouldn't be worked on again, but some requirements in further experiments proved that there was a bug that had to be mended before continuing the process.

2. Datasets

For the realization of the project, several test datasets have been used. Most datasets consist of 720p video samples of the camera running along different arms. Every video has been split into frames, due to image redundancy (where no movement between frames is appreciated) some frames were removed and the results shown in this project take frames i intervals of 5, from 5 to 20 frames.

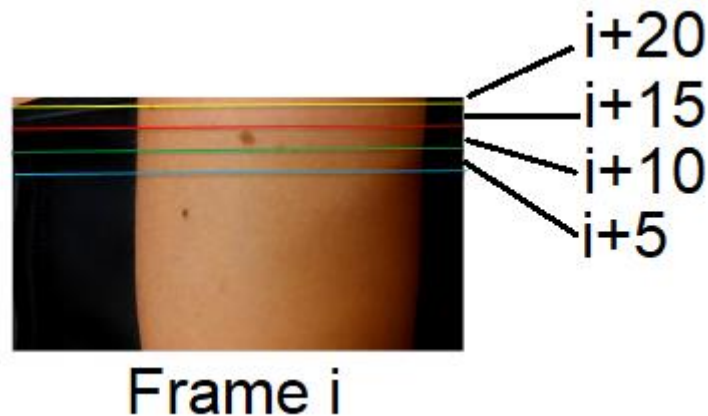


Figure 11: Figure shows the frames of the video and the region they add over the initial frame i cropped by colour lines.

At early stages of the project other datasets of non-arm images were used for testing, testing's such as Harris corners, SIFT and stitching.

3. Methods

3.1 Image registration

To stitch a pair of images with an overlapped area, a handful of matching points between the images is needed. Due to the nature of our datasets and issue we want to solve (stitching of image pairs extracted from a video sequence in which every frame has a different position and small angle variation), the transformation between the images is of a projective kind hence the homography is needed to produce the transformation between images. In this kind of transformations, a minimum of 4 matching points between the images is required so that the homography matrix can be computed. The homography matrix is a matrix which allows to take one location point from the base image and place it in its matching place of the second image, which does not mean all points from the first image are going to be in the second because due to the movement some points may fall out of bounds. Homography

matrix will always be a non-singular 3x3 matrix, what means that the determinant will not be zero.

$$\begin{pmatrix} wx' \\ wy' \\ w \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

p' \mathbf{H} p

Homography matrix is H where p is a known point, and p' is the projection of that point over the transformation provided by the homography.

The homography matrix maps lines to lines, so a pair or parallel lines will still be lines but they won't necessarily be parallel anymore after the transformation is done.

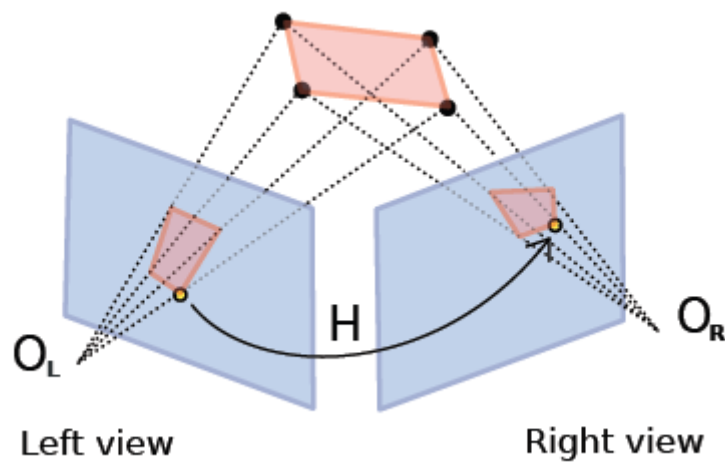


Figure 12: A real-world object seen from two different perspectives map differently, parallel lines from each perspective do not remain parallel but they do not morph into curves (Openmvg.readthedocs.io, 2017) [7].

This process can be used to align images made at different times or from different perspectives. The process of putting a set of points in correspondence with the points in a second image with the aim of aligning them is called image registration. Image registration is used in many fields, from arts with panoramic images to scientific purposes in research. Image registration algorithms used throughout the project can be classified in feature based, automatic methods and an interactive method.

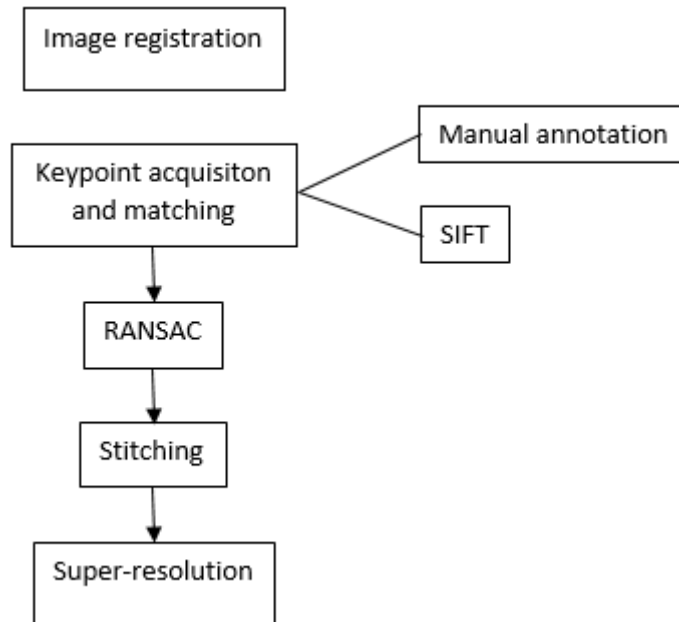


Figure 13: Steps to fulfill image registration process

3.1.1 Feature based methods

Within the feature based algorithms, the most popular ones are the Harris corners and Scale-invariant feature transform, (from now on SIFT). Feature based methods find correspondences between the base image to the target one looking for points, lines, changes in the gradient and many other details. Feature based methods are compared directly to intensity based methods which rely in the similarity of intensities from the images to be put in correspondence. They search for patterns from the base image inside the target, and the comparison is usually measured by the correlation from the pattern into the target patch.

3.1.1.a Harris-Stephens corners

The Harris corners method takes the corners and edges in the image as the features to look for. Harris is an improved version from Moravec's algorithm. This method is independent from illumination, rotation and scale. And is focused to track corners and edges in motion [8].

Moravec's algorithm considers a window in the image, said window can be either a squared or a gaussian window, and computes the changes on the intensity through shifting the window in several directions with a 45-degree variation from each direction. He states three relevant cases.

1. If the patch has no significant changes in intensity, shifting in any direction will return a minor change.
2. If the window lies over an edge, shifting along the edge will be a slight change, but the shifting in a direction perpendicular to the edge will turn into a significant change.
3. If the window is over a corner, shifting in any direction is going to result in a large change.

$$E(u, v) = \sum_{x,y} \underbrace{w(x, y)}_{\text{window function}} \underbrace{[I(x + u, y + v) - I(x, y)]}_{\text{shifted intensity} - \text{intensity}}^2$$

The changes found over the regions are computed by correlating the window to the image.

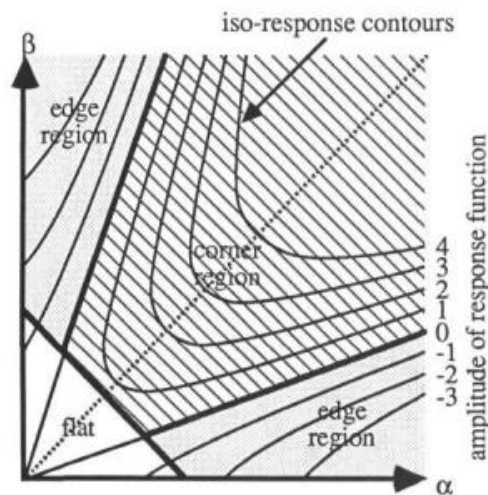


Figure 14: Auto-correlation principal curvature space-heavy lines give corner/edge/flat classification, fine lines are equi-response contours [8].

Harris Stephens uses the first derivative to search for the changes, what makes that in high textured areas will find many points whereas in flat areas hardly no points will be taken. This means that the points are going to be clustered in rich textured areas, instead of being even distributed along the image sample (Bennett and Lasenby, 2014) [9].

This could be a helpful method but due to the nature of our problem and the skin dataset features this method won't return many if any results.



Figure 15: Example of Harris features from Matlab's function

3.1.1.b SIFT: Scale-invariant feature transform

This method uses detectors in order to find similar blobs between the images to stitch. A blob is a region of a digital image in which some properties are constant or vary within a prescribed range of values. Blob detectors can identify matching areas in an image which are too smooth to be found by a corner detector.

SIFT was originally introduced by Lowe [10] as combination of a DoG interest region detector and a corresponding feature descriptor. SIFT features descriptor is invariant to image scaling, translation, and rotation and somewhat invariant to subtle light changes.

The main steps of Lowe's algorithm are the following:

1. Scale-space local extreme detection: using DoG detector interesting points are searched on all scales and positions of the image.
2. Accurate keypoint localization: the goal of this second step is to test the stability of the keypoints, analysing the contrast and evaluating the presence of edges or lines and interpolating the position to increase the precision.
3. Orientation assignment: this step is necessary to achieve the invariance of the feature's descriptor to image rotation. A direction, based on local property of the image, is given to each keypoint. The image gradient magnitude and orientation is sampled around the keypoint location using the region scale to select the level of Gaussian blur (i.e. the level of the Gaussian pyramid at which this computation is performed). An orientations histogram is built from gradient orientation. The histogram has 36 bins and covers the entire range of orientations and each sample of it has been weighted by its gradient

magnitude and by a circular Gaussian window with standard deviation equal to 1.5 times the current scale. To obtain the keypoint orientation the value of the histogram's highest peak is selected.

4. Keypoint descriptor: A descriptor must be assigned to each keypoint for the local image region that is as invariant as possible to change in illumination or 3D viewpoint. A gradient magnitude and orientation of neighbour points of selected keypoints are also computed. Samples are accumulated in orientation histograms summarizing the content of each region.

Each bin of the histogram contains the sum of the magnitude of gradient that are close to direction that it represents. The descriptor is formed from a vector containing the values of all the orientation histogram (bins' height).

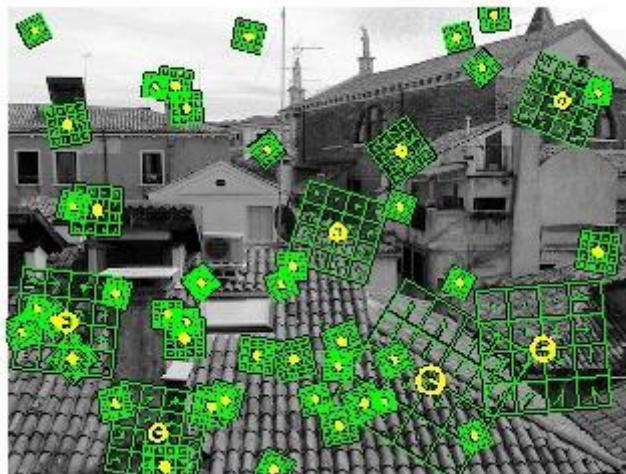


Figure 16: Strongest SIFT features in the image, with the scale on yellow and the direction on the green matrix (VLFeat.org, 2017) [11].

Features are stored inside a database to be compared with the ones of other images in order to give matching features. This matching is done using the Euclidean distance of the features vectors. Keypoints given by SIFT are highly distinctive which is useful to find exact matches for most keypoints, even though some false positives can be still found.

The number of features is relative to the image content, and various parameters such as the size of the area that features are computed within. Therefore, it is necessary to remove all false positives from the matching keypoints database. The process of removing these points is through the iterative algorithm called Random Sample Consensus (shortened as RANSAC from now on).

3.1.2 Manual annotation

For the manual annotation method, user must manually search for corresponding points among the images. The more accurate is the point picking the less distortion

will appear in the final mosaiced image. This process can easily be done within Matlab by selecting a point in a window using the available *figure* tool, as it has a tool which allow you to click a point (or as many as necessary points while holding the 'Alt' key in Windows distributions), and it will give you the location of the point, its values and which key was pressed to retrieve the coordinates. Once all the information is selected it is possible to export the data into a struct by right clicking and selecting the 'Export to Workspace' option in the menu. Another way to retrieve pixel location information within Matlab is *ginput* function, the main downside to this alternate method is that the *figure* it opens lacks zoom in and out options, therefore the accuracy for scaled down images is reduced.

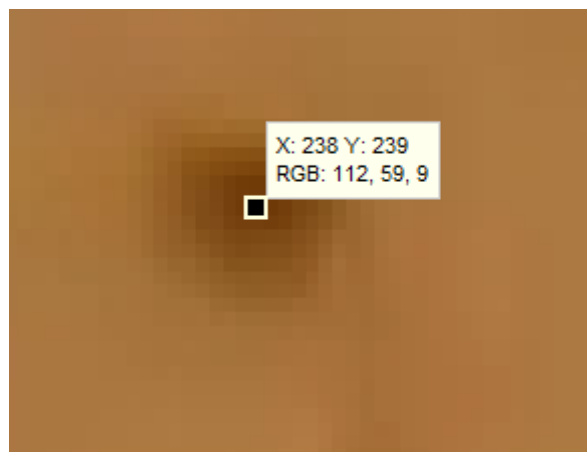


Figure 17: Matlab tool result for manual point annotation

3.2 RANSAC

As stated before, briefly RANSAC is an iterative method that analyses the data given, and estimates a group of inliers and outliers. Outliers are the false positives that must be removed, the difference between outliers and inliers is that the outliers do not fit the model. The method is highly resistant to noise while keeping mismatches away from the inlier group. It is a nondeterministic algorithm, as it finds a correct result only with a certain probability, that increases with the increasing of the possible iterations.

The algorithm has been published for the first time by Fischler and Bolles in 1981 [12].

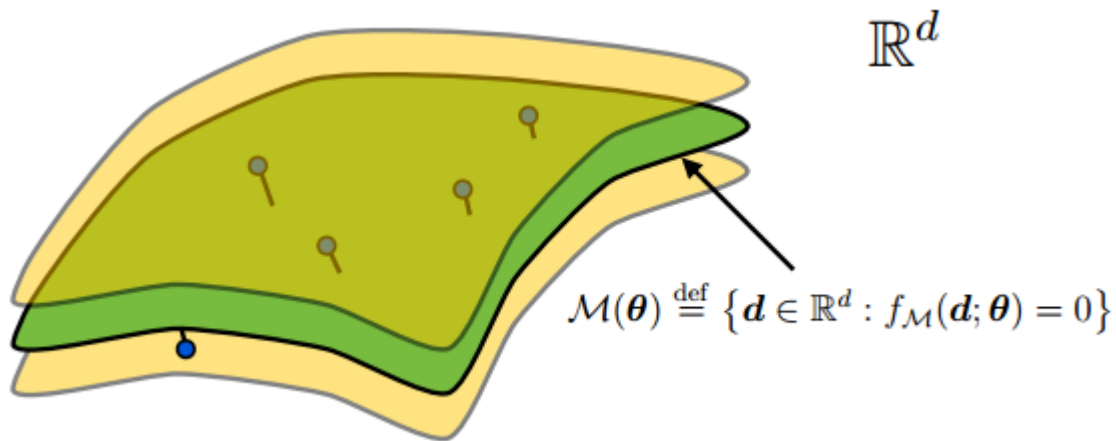


Figure 17: This figure displays the model space M as a green surface (the locus for which $f_M(d; \theta) = 0$). The yellow surfaces represent the boundaries for a datum to be considered an inlier (assuming that the distance function is the Euclidean distance, hence the smallest distance between any two points on the yellow surface and green surface is d). The structure of the green surface is both defined by the model M and by the parameter vector. The inliers, represented as blue dots, lie in between the two yellow "crusts" [13].

Although RANSAC works well with a large number of outliers, if the correct matches are fewer than 50% RANSAC will not provide a good set of inliers due to the real matching points are being outnumbered and its weight will not affect enough to make a proper transformation model. RANSAC is also time-consuming depending on how many matches are found.

The basic and general RANSAC algorithm takes at every iteration a different subset of the SIFT matches, these points cannot be repeated in the same subset but can appear in different iterations one at a time. The subset size depends on the problem, and it's the same of the necessary points to be able to calculate the transformation. These points are supposed to be inliers, with these values it is proceeded to find the transformation model, which is taken into testing. The points are transformed based in the model and compared to the position of the matching points, this comparison is done using the sum of squared differences. This returns a value that we do not know yet if is acceptable or not, hence a threshold value must be established. If the value found is larger than the threshold means the points have an outlier and these points don't give a proper transformation. Otherwise, if the value fits the threshold means the points matched and we save the points and how many of them fit. In following iterations, the best subset is the one that provides more matching points, so this is also compared in order to state the best transformation model. This means that a good threshold value is of great importance to get the most accurate model.

3.3 Stitching

Image stitching consists on creating an image of higher resolution from a set of images with overlapping areas. The stitching process deforms the images using affine or projective transformations. The resulting image is composed by images taken at different times, in the current dataset this time displacement is less than a second assuming the minimum frame rate that the eye is able to identify video instead of a succession of images which is 25 frames per second.

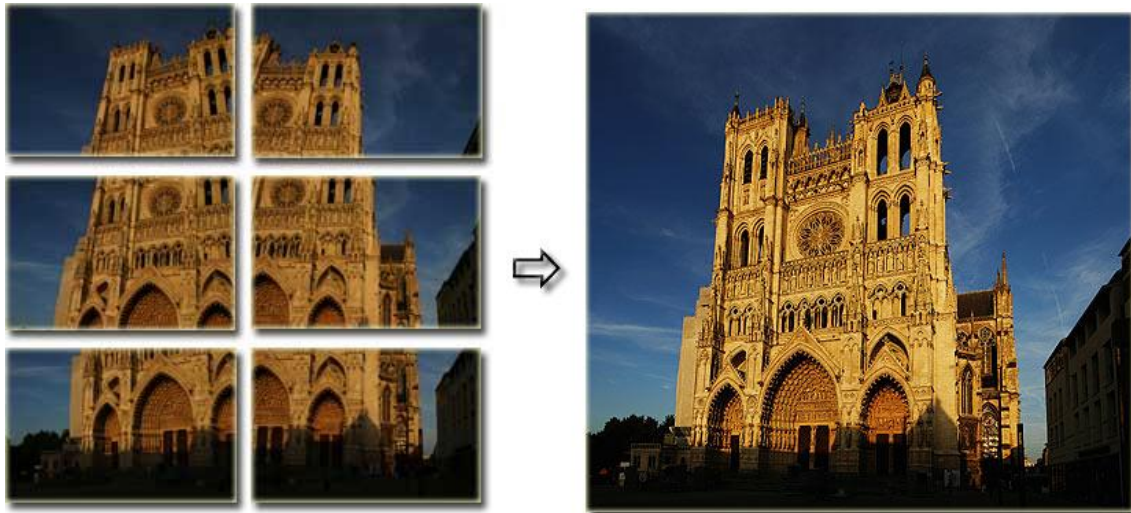


Figure 18: Image stitching example using a set of images of a cathedral to form an image showing the whole architectonic piece (Ptgui.com, 2017) [14].

For generic panoramic imaging, objects that appear in one of the frames but not on the rest, or if any object moves it will create artefacts in the resulting image.

3.4 Super-resolution

Super-resolution imaging is known as different methods that enhance the quality in digital images. In this project, these methods can be applied due to all the images produced by the video recording, there are at least 24 frames per second up to a thousand fps in most modern Sony smartphone devices (Singleton, 2017) [15], most of frames are really close to one another even for slow frame rates, so lots of images are produced in every sequence, and to make the alignment and mosaicking faster, not every frame is used, so this creates that most of the frames are not used and discarded. With the information lying on those frames, the result can be improved. The improved results can also be achieved with the frames selected for the image mosaicking.

The aim of super-resolution in the project is to define better moles boundaries, because this is a key factor of the illness analysis, while reducing the noise of the

image. This is achieved at the time of merging the images, giving priority to the base image over the transformed one.

4. Development

4.1 Naive stitching approach

The development process starts with a naive version to stitch a pair of images, the keypoints from these images are obtained manually. For this method only four points from each image are needed, these points must be selected with the minimum error in order to get the best result.

If all and only the necessary points to build a homography Matlab has a build-in function called *maketform*. This function creates a spatial transformation structure, the transformation can be affine, projective or customized by the user. Inside the structure the homography 3x3 matrix is found under `data.T`.

To apply the homography to the image the *imtransform* function is used, with this function the bounds of the transformed image are returned, this is used to build a holder image to display both images without any cropping. Now both reference and target images are transformed to fit the holder image. Both images must be inserted into the placeholder, to see if the matching is perfect both images can be averaged and check if there's any blurring, if it is, the deviation from the points is present and to give a better result choosing the maximum value from the pixel at each point can produce a better result. Another issue in the resulting image can be a visible edge for the images as this method does not blend the colour through the overlapped area of both images.





Figures 19, 20 and 21: Top left image shows the averaging of the two images, the boundary of the first image is clearly seen. On the top right, the difference between both images is displayed, there's a line displayed, that means it exists on one image but not in the other. On the bottom row, the maximum value of each pixel is taken, so there's no faded regions.

4.2 RANSAC for manual keypoint picking

This method requires an input of the matching points between the pair of images. It is supposed that the first image is the base and the following is the target.

To see the chances of getting a proper transformation, both images are shown with the matching points between them. This can be useful to see if there are outliers, and to make sure the input points are from the selected images from the dataset.

Next, a RANSAC algorithm must be developed, the number of iterations must be taken accordingly to the number of points given. To be sure every combination without repetition is computed, so no possible results are left behind. As the homography only needs 4 points to be computed, the random selection is going to be of 4 points at each iteration. To select these points at each iteration a subset of 4 points is chosen by slicing from the beginning a random permutation without repetition of the initial keypoint array from the base image. With these 4 points, the homography transformation to its matching point is computed.

To compute the homography, a $3 \times n$ matrix is made where n is the number of points it is getting. In this matrix, the first row will have the x coordinates of the base image points, the second row is going to be the y coordinate and the last is a row of ones. Under this matrix, a 3×4 matrix of zeros is appended and lastly, two rows in which the coordinate x from the base image are multiplied element wise by x and y for each row from the target image, the next and final row the coordinates x from the target image are placed.

For the next matrix, the upper 3×4 matrix is switched by the zero matrix. Then the coordinate y from the target image is multiplied element wise for the x and y from the

base image, this process will produce two rows first being the one multiplied for the x on the base and the other the y from the base. Lastly the y coordinates in the target form the last row.

These two 9x4 matrixes are placed side by side and computed using the singular value decomposition, this is calculated with the *svd* function Matlab has got built-in. From the return values, only the left singular vectors are needed. To get the homography the singular vectors must be sliced and reshaped to get a 3x3 matrix from the first element of the 9 singular vectors.

Once the homography is computed, this transformation matrix must be applied, to do this the points from the base image are multiplied for its corresponding row, the x coordinate is multiplied for the first row, the second for the y coordinates and the last one is a one's array multiplied for the third row. The final points are the resulting from the previous multiplications, divided by the result of multiplying the third row for the ones.

At this point, the position of the transformed points is known, now to be sure this is a valid homography, the error between these transformed points and the ones obtained manually must be calculated. For this, the sum of squared distances is computed. This distance must be compared with a threshold value big enough to give a solution after all iterations, but small enough that this solution is the correct one. From here two things can happen:

- The value is smaller than the threshold. This means the next point must be computed for the sum of squares differences. And this until it runs out of points
- If the value is smaller, a counter is increased, this will be compared with the actual maximum number of inliers found in previous iterations. If we have a higher number of inliers, the maximum will be updated and the points and homography saved, replacing the old ones.

At the end of this process, if the threshold meets the requirements an homography will be found. As the points of that make the homography are saved, these can be displayed among the ones discarded.

This RANSAC algorithm can be used with manually annotated points, but also with points from automatic algorithms.

4.3 VLFeat SIFT

VLFeat is an open source library which implements computer vision algorithms focused at extracting local features and matching between images.



Figure 22: VLFeat keypoints (blue) superimposed to D. Lowe's keypoints (red) [11].

Most of points are the same, even though some differences are visible, the following pie chart shows the percentage of points that match and the precision of matching per pixel.

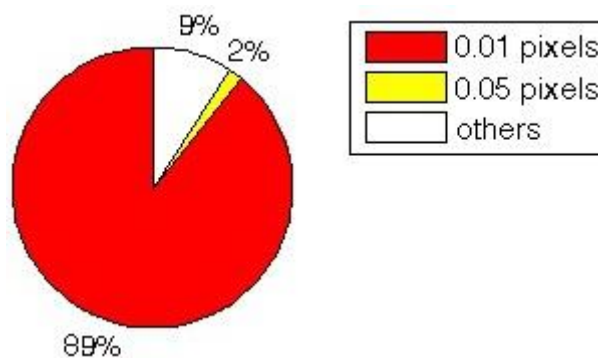


Figure 23: Matching point deviation between Lowe's and VLFeat implementation from the centre of the feature [11].

4.4 Colour Correction

4.4.a Plane of differences

At this point, the transformation required to stitch the images is known. In order to make the mosaiced image result seamless there are different methods that can be

used. In this paper, an approach is that if we know matching points between the images, therefore the sequence of images is taken from a video sample that moves linearly to a certain direction, so the colour and luminance values of base and target image must be close numbers. We make the hypothesis that the different lighting of the picture is applied uniformly to the image, hence the change in colour can be modelled as a linear interpolation along the image. For this reason, a plane fitting the colour difference between the two images is expected. At this point a function to compute a plane that best matches all the points in said plane is needed.

To fit a series of 3D points into the best fitting plane, the coordinates that define the plane must be stated. As for the x and y coordinates, the x and y location of the pixel in the image is taken, as for the z axis of the image the subtraction from the intensity value between the base and the target image point.

At first instance, the mean of all the points belongs to the plane, then the eigenvectors are calculated using the *eig* function, this will return a diagonal matrix of eigenvalues, and another matrix whose columns are filled with the right eigenvectors. From this matrix, the first row is the normal of the plane, and the rest represent the orthonormal basis of the plane.

Once the plane is defined by its normal and a point in the plane, the points in the target image are projected into the plane using the x and y coordinates as the pixel, and the z coordinate as the intensity value for the pixel in that channel or its single intensity in a grayscale image.

To get a better precision in this plane, knowing the homography, a huge set of points from the first image can be transformed to points into the second image, so at the time of scattering the differences, much more data can be displayed and taken into account to build a better precision plane solving some remaining luminance issues. Displaying the scattered points can show that what seems a plane in the space, is actually a paraboloid, or an uneven surface. This might not be perfect, as the points are randomly selected some of them can be transformed to noise points, and this would make the plane tilt in an inappropriate way. To get rid of the noise, what is noise must be defined, the intensity of each channel of an image goes from 0 to 255, anything above a difference of 40 intensity units, a bit above an eighth of the scale, is considered to be an enormous difference therefore noise.

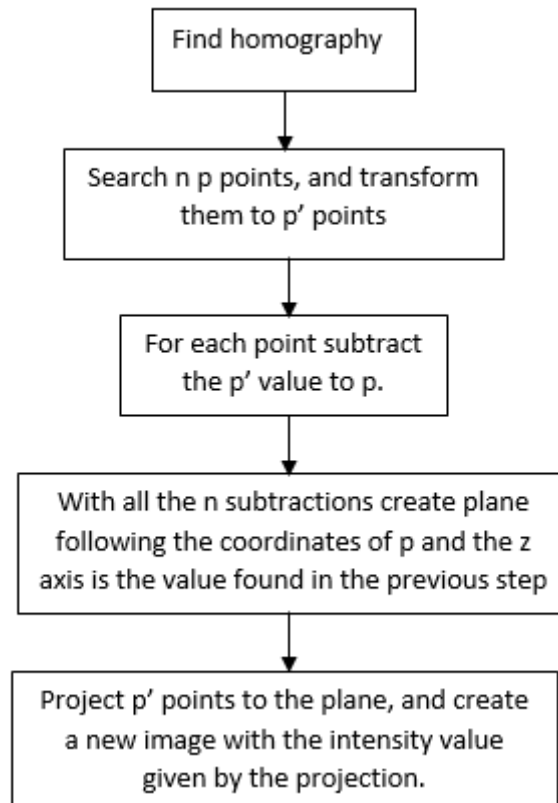


Figure 24: Pipeline to creating the plane of intensity differences, this is needed to make an equalization of the colours on the target image due to the illumination difference between frames

4.4.b Histogram matching

The former method may create a plane parallel to the ground, therefore only brightening or darkening the image by a few intensity points at all pixels. This would make a problem for some cases, such as non-planar surfaces, therefore another approach efficient must be made. At this point both images should have a rather similar histogram. So, histogram equalization is a method to disperse the values through a number of bins, being 256 bins in an image at total. That's not what the process is about, but to mat the former histogram to an ideal one, the ideal one is going to be the one from the base image.

In Matlab histogram matching can be developed with several functions, such as *histeq* that receives the target image as first parameter and as the second the ideal distribution. This method takes full RGB photos and due to RGB images nonlinear nature, this leaves huge artefacts through the image. Other Matlab's functions that allow to match histograms are *imhistmatch* for 2D images and *imhistmatchn* for nD images.

To reduce and remove artefacts, a linear representation model for images must be found, this can be HSV, which decomposes images in Hue, saturation, and luminance, this transformation to the image makes that each of the 3 channels is going to be at a double precision in a range from 0 to 1.

The algorithm to match the histogram goes as follows:

1. Compute histogram for both images, easily done with *imhist*.
2. Calculate the cumulative distribution function, this is made with using *cumsum* from the former histograms, and then dividing it by the total number of pixels at the image, the number of pixels can be known by the multiplication of the size of the image or using *numel* of the image.
3. The following step is to map the values from one histogram to the other, for each intensity level, on the scale 0-255, one value per iteration of the cumulative distribution function from the target image is subtracted to all the elements on the ideal cumulative distribution function, the ideal value is the index of the one that is 0 or closest to 0. This indexes for each level are saved in an array.
4. With the array, the values from the target image are mapped into the new distribution.

Resulting histogram can never be the same, but it will have similar values. Nevertheless, there are approaches that try to minimize the differences between the histograms. The main problem, is that several values, can map to only one value making the appearance of artefacts.

4.5 Mosaicking

To represent the mosaic of a couple of images, first the bounds of the resulting mosaic must be found. Within the bounds both images will be represented fully, no pixels will be found outside.

The base image, is laid down in a placeholder with the mosaic size, where the data that this image doesn't fill up it stores not a number values. The same process goes for the next image, the difference is that the transformation is applied. To merge both layers the not a number values must be replaced by a number that computes to a colour in the actual data type, for example 0 and these regions will be black.

To display the images in the form of a chequerboard, Matlab's *checkerboard* function is called, which computes a chequerboard with some parameters that make the board bigger than the image, the extra part is sliced off. Then the inverse of this

checkerboard has to be found and multiply the base image for the first checkerboard and the second for the other, now it's a matter of merging these two boards and the image will be displayed.

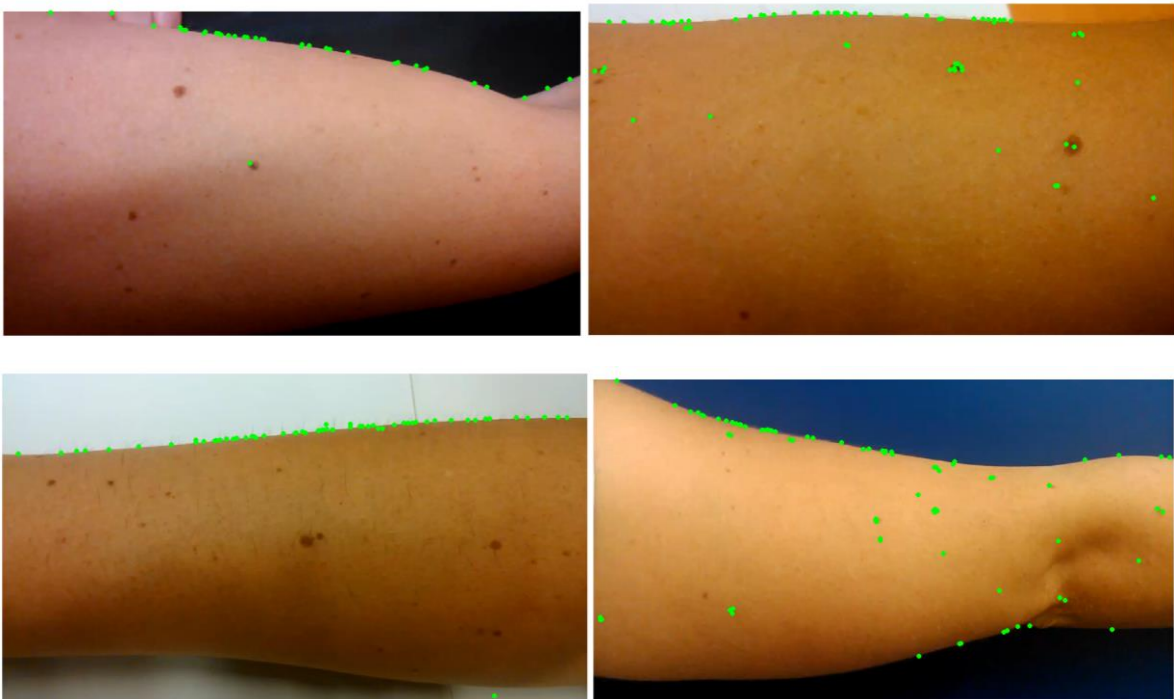
To see which region of the images change over the transformation, or to see the overlapping area, subtracting former placeholders and making it absolute values shows the differences between the images. At analysing this resulting mosaic, if a value is not black it means that the images differ at that point.

The first approach to super-resolution with the aim to reduce noise and make sharper looking images is by merging the layers with different opacity levels, this will also help the seam of the images fade. This is an easy process, by selecting the overlapped pixels, give a percentage of opacity to one layer, and the complementary to 100 to the other, so the result has full colour and the image is not faded through the overlapped area.

5. Results and discussion

5.1 Harris-Stephens corner detection

Despite this being one of the most used methods, theory says that in the datasets used, and the nature of this application might not be useful, but if enough points across both edges on an arm are found, an accurate homography can be found.

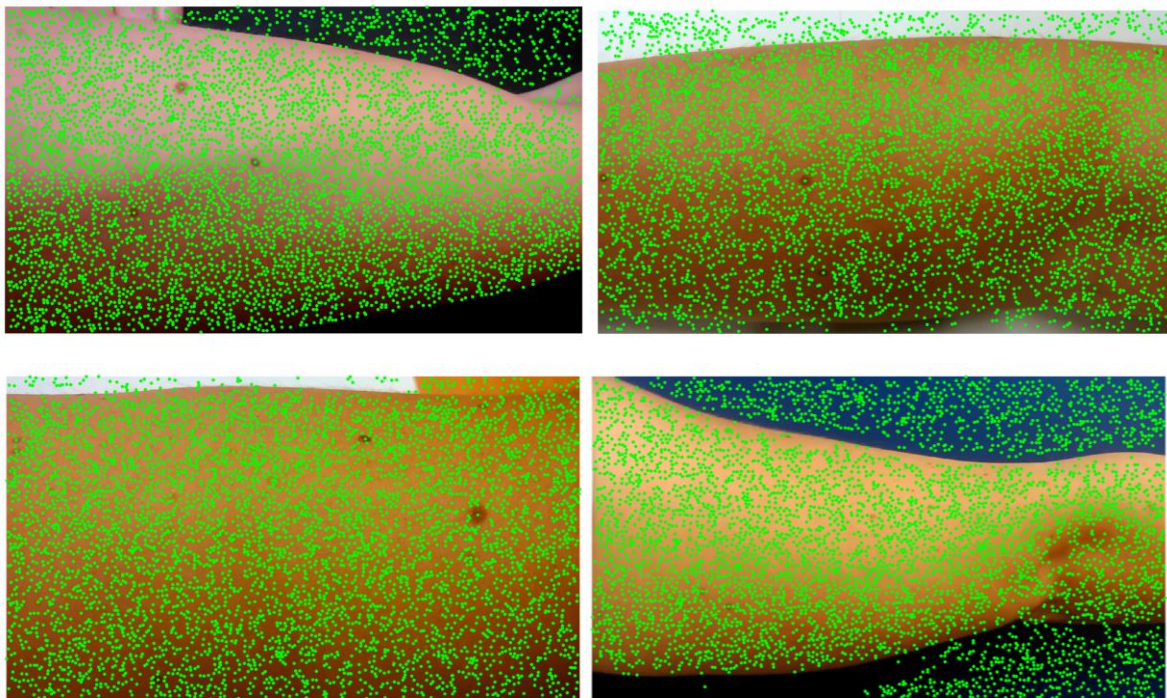


As it is seen, for half of the datasets points are found on the upper edge of the arm. It is strange that in the datasets that only finds keypoints on the upper edge it doesn't find any on the moles, but in other ones keypoints are found around the moles and sometimes even on plain skin.

To determine if this method is worth using, results must be compared to the most used one, SIFT.

5.2 SIFT: Scale-invariant feature transform

Following results come from using the SIFT function inside VLFeat library.

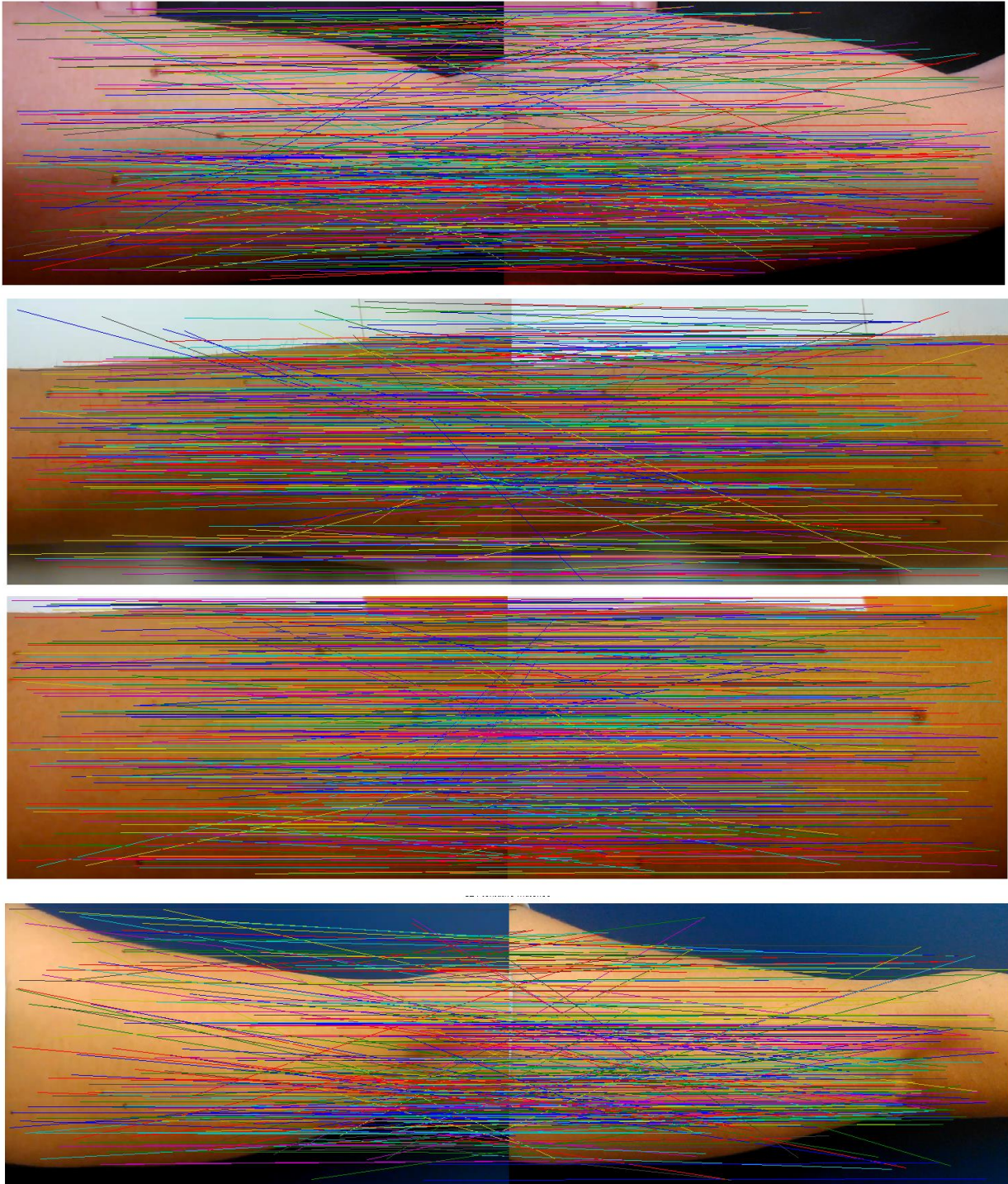


It is interesting, and worth mentioning that SIFT doesn't always look for the moles to find keypoints, SIFT works looking for blobs, small areas, and logic says that richer information on changes is going to be around the moles, this happens in the first 3 examples, where the biggest moles have a keypoint lying in the middle, but in the latest one, there are many small moles isolated from the keypoints found.

If we compare the results in this section with the ones found in the Harris-Stephens it is seen that Harris produces few keypoints, as we want the most accurate result achievable, from now on, if a manual input of keypoints must be compared it is going to be against the SIFT results.

5.3 SIFT matching

While trying to match two images, results will be fewer than running SIFT through the actual image. Following images show all the matches, including outliers.

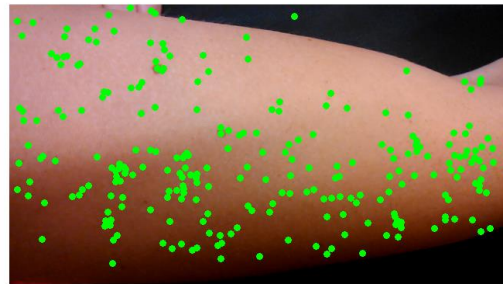
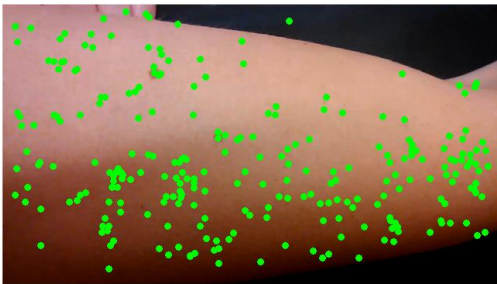


From these results, it is acknowledged that not in every situation SIFT select a mole as a relevant point to match. Also, the distribution of features goes through all the

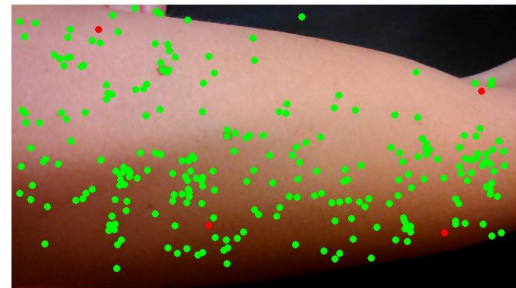
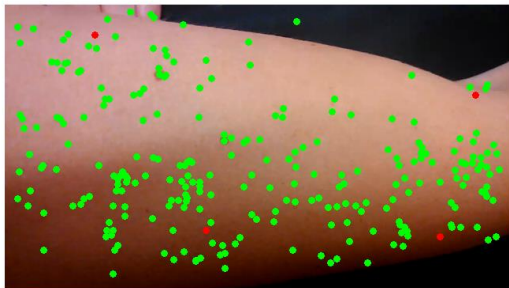
image, with this at a simple peek it is seen that there are plenty of mismatches, this is the time where RANSAC is needed to clean the image.

5.4 RANSAC SIFT and Homography

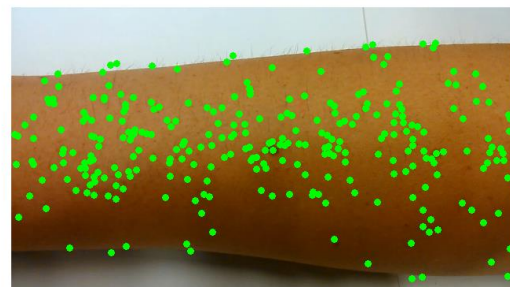
The mismatches found in former results must be removed. This is where RANSAC takes its place in the project, also RANSAC will find the best features to build the homography.



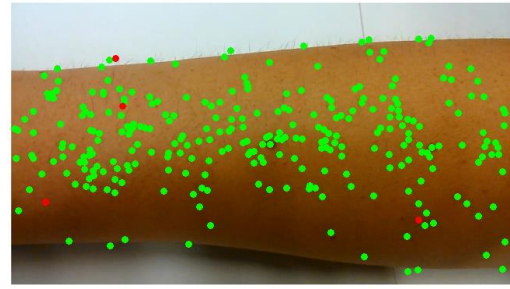
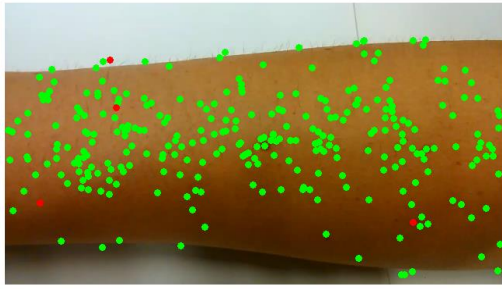
At first SIFT gave 414 matches, after applying RANSAC matches were reduced to 263 which is a 64% of the total.



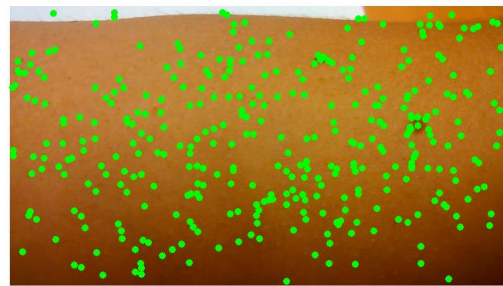
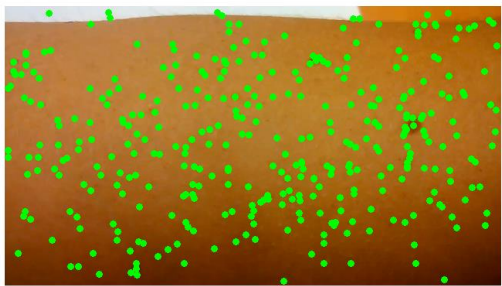
In this pair of images, in red the 4 points to create the homography are shown. It is interesting noting that not a single point from a mole was selected for this.



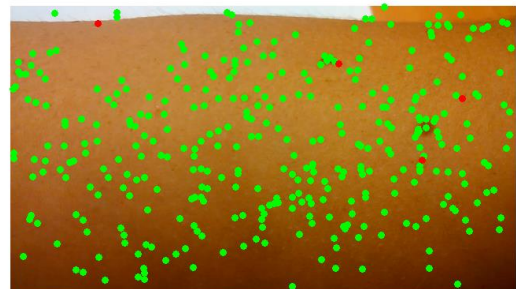
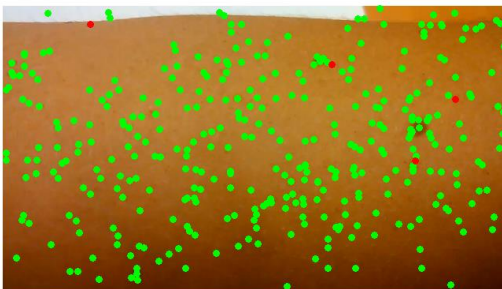
468 matches were initially found, after RANSAC this number is reduced to 306 which means it removed 35% of the total features.



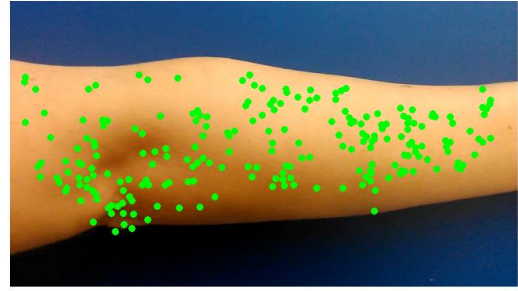
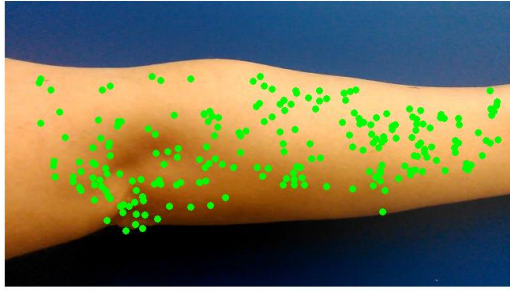
In this image, we see that the matches selected to be the homography are in the left-hand side of the image, but there is one on the other side of the image, this will increase the stability and cause less distortion for the transformation. This latest point lies over a mole, which means that homographies over moles central points are a possibility.



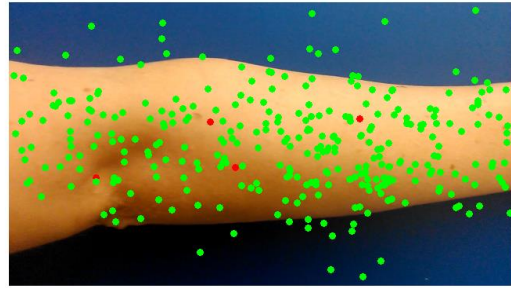
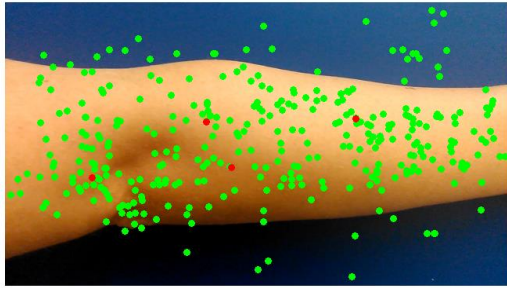
In this pair, 502 tentative matches were found but only 374 remained (75%).



In this image, the definitive RANSAC points for the homography are lined up, instead of being sparse. This will make an interesting comparison between the resulting image mosaic and the first result posted in this section which takes 4 points distant from the others.

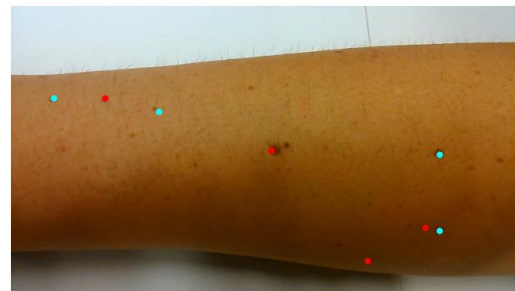
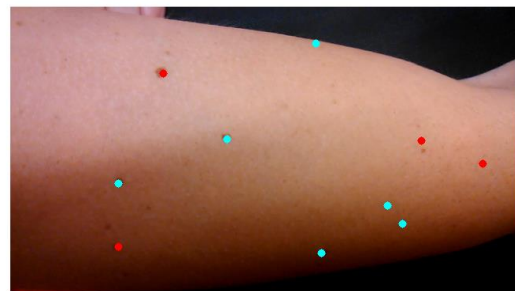
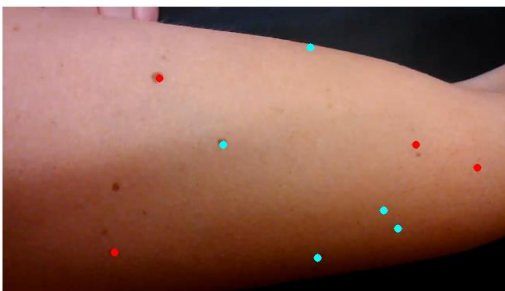


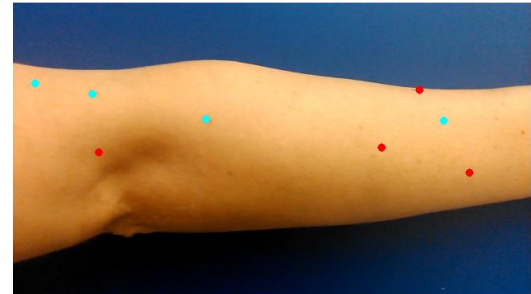
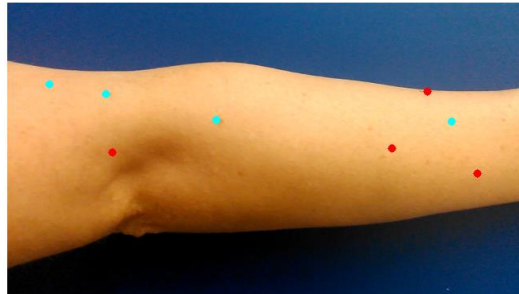
343 matches were found and 67% remained making 230 final matches.



This homography distribution looks like it is close together, but on the wide part of the image they are distant. It is going to be interesting to see if it is going to distort towards the upper and bottom edge of the arm.

5.5 RANSAC homography through manual annotation method





Red points are the ones selected for the homography, blue ones are the other points in the set given by the user. In some cases, the lack of moles makes that there are not many points to choose from, or the distribution from these can be clustered in any point.

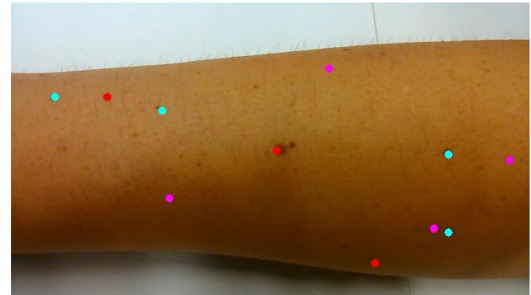
5.6 SIFT and manual keypoint annotation comparison

In cyan, there are the points that are manually selected from these the red shows the ones that are selected after RANSAC is ran. Lastly, the magenta ones are the features selected by VLFeat SIFT.

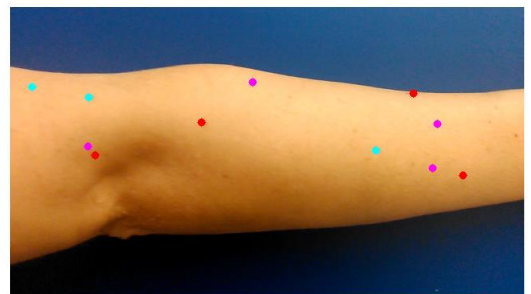
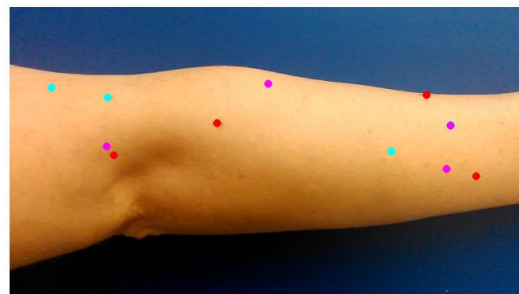


These two rows show the same data set, some red points are lacking because they fall behind hand selected points. This is good for the project and it's a symptom of accuracy when hand selecting the points.

It must be noted that SIFT is not going to give the same features every time it is run within the same sample images.



In this result, it is seen that VLFeat and manual annotation have one point in common, the points under the manual annotation fall in a line, while the VLFeat ones are better equispaced.



In these last distributions, even though results are close in distance, the transformation isn't the same, and it can make some deformations.

5.7 Mosaic display

To be sure the mosaiced image is properly transformed, it can be displayed in several ways, the most effective method to show if a mosaiced image is properly transformed is the chequerboard display methodology.

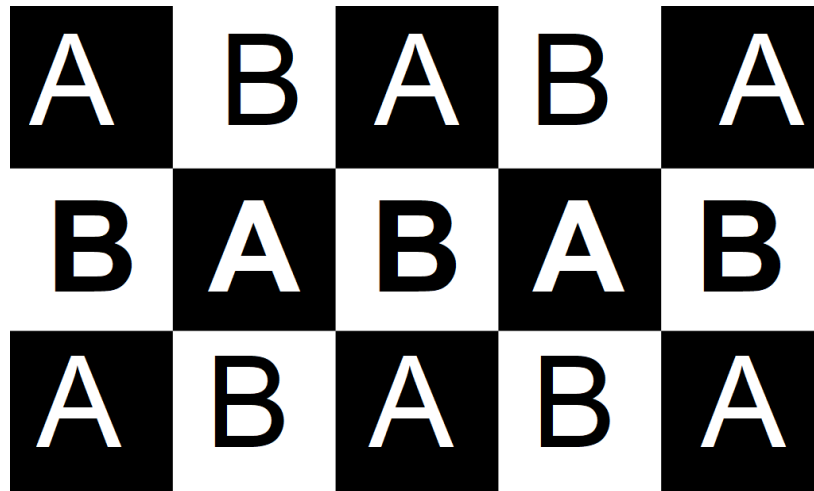
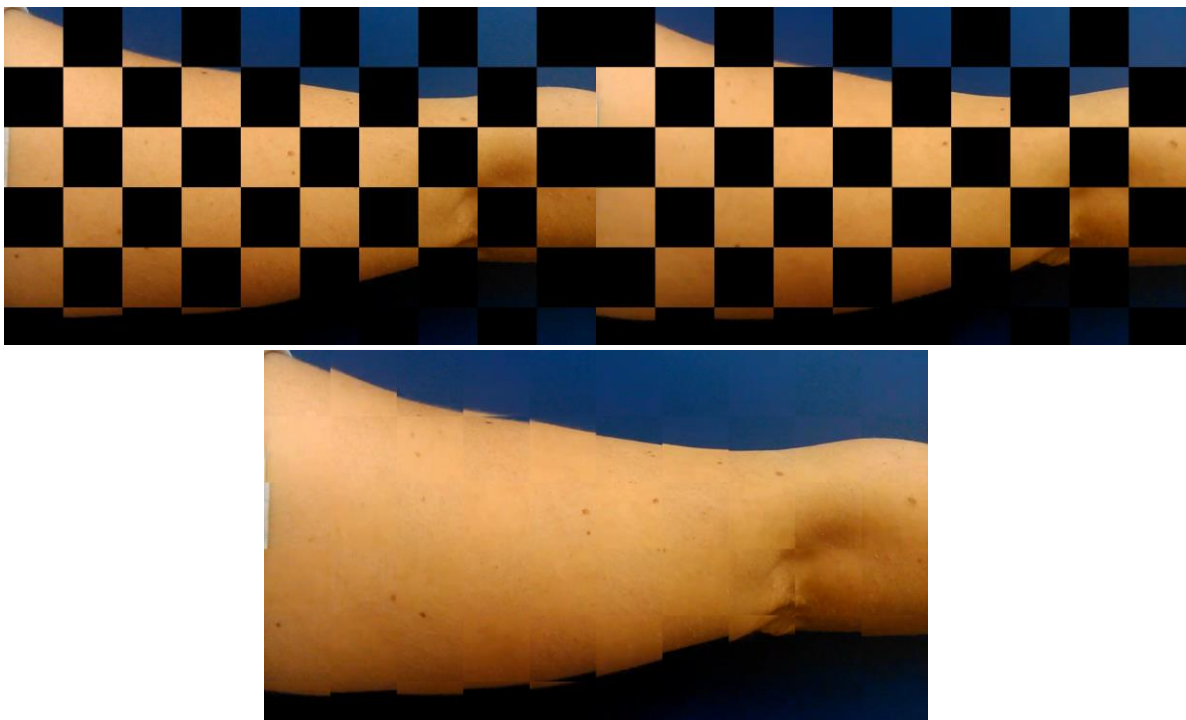


Figure 25: On the tiles with an A will appear the values from the base image, whereas the ones with the B will be filled with values from the target image



Figures 25, 26 and 27: Example of the steps to display a chequerboard mosaic

In the figure, a black and white chequerboard is shown, black boxes will show that portion of image from one image and the white regions will show that one from the other image. By doing this it can be appreciated if lines join perfectly through the images. This method can also show the difference between colour from both images as the seam dividing the cells can be seen if the intensity differences is great enough.









First, it must be acknowledged that the transformation is needed because there was some movement between the frames. This can be easily displayed with a chequerboard of the two images without any transformations on them.



On some cases, the movement is nearly null, but in other ones it is significant, we can see that the arm edges break at every tile. This distance can affect the accuracy of the homography, so when creating the mosaiced image it will be interesting describing the average distance in pixels there is from the features of the base image to the target one, this is calculated using the euclidean distance. For example, this will be useful in the bottom left image, where no movement is appreciated. This distance is calculated by taking the position of the keypoints of the base image, then the location of the keypoints in the second one and as it is known which point does any of them match the euclidean distance between them is computed, finally the distance is the mean from all the euclidean distances.

Now that it is clear that the adjustment is needed, let's apply the algorithms explained in the other section. And will compare the manually selected points with the results from the SIFT.

All the following examples have the image to transform 5 frames ahead of the base image, under the image the distance between keypoints is stated.

SIFT	Manual annotation
 <p>15px</p>	 <p>14px</p>
 <p>5px</p>	 <p>4 px</p>
 <p>3px</p>	 <p>2px</p>
 <p>41px</p>	 <p>41 px</p>

From these chequerboards, it is seen that the SIFT methods perform a better transformation than the keypoints selected by hand. Three conclusions can be extracted from these:

1. When the keypoint distance between frames is under 5 pixels, the hand selected points perform similar and no hiccups are seen along the edges.
2. The error from selecting the points in the two images might be big enough to make a difference, this is not likely to be so due to the images with low displacement perform same as SIFT.
3. SIFT takes points, and these not always match a mole, even less likely that the 4 points of SIFT lay on moles. So, SIFT has a better distribution finding keypoints, that will cause less distortion and better mosaics.

5.8 Colour Correction

When talking about image stitching, blending is a concept that comes along. Blending algorithms aim to merge the images without a visible seam and reduce the possible artefacts from moving objects (usually seen in artistic panoramic imaging). Blending algorithms merge the colour and find the optimal seam along the overlapping regions of the images. The main downside besides the artefacts is that the colour correction is made only to this area, so if an image has a whitening gradient from left to right, and the other image has an opposing gradient, the stitched result will be two darkened regions on the lateral edges while the central part is clear.

Input image I_1	Input image I_2	Feathering	Pyramid blending	Optimal Seam

Figure 28: In this figure, the left side of I_1 is stitched to the right side of I_2 , different blending algorithm results have been added to see the artefacts they produce [16].

This is not what we want to achieve, we want to transform the colour along all the image, so new approaches must be found.

5.8.a Planar colour correction

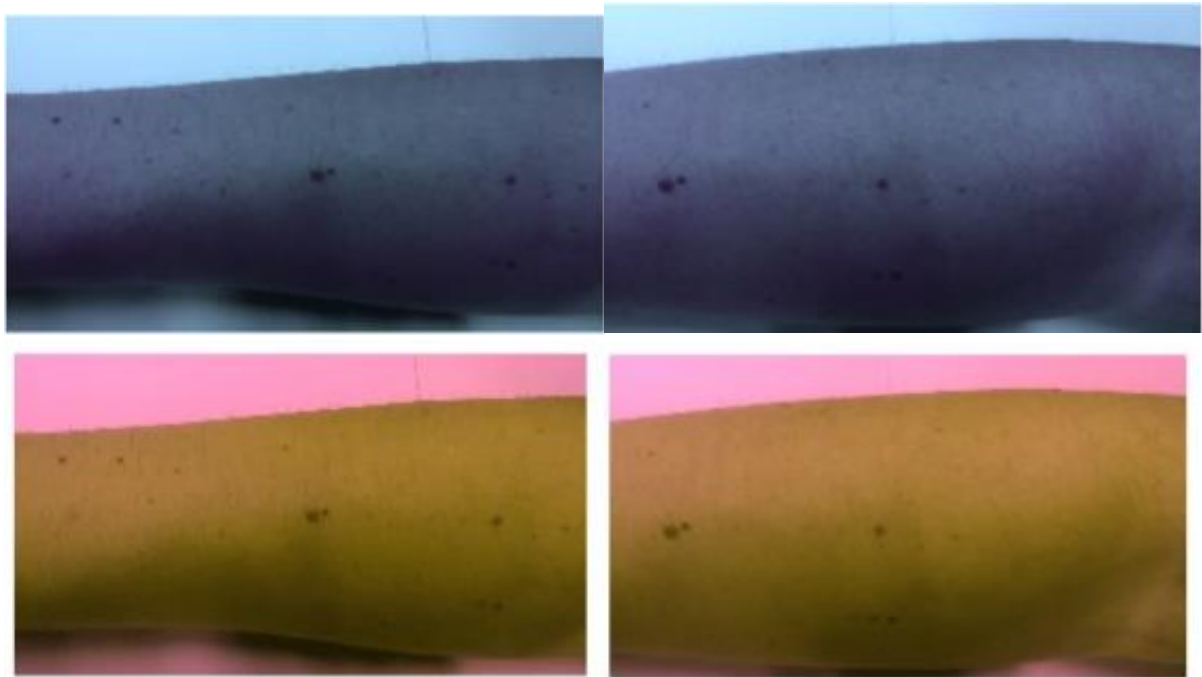
As the first approach, the space was defined as following:

X axis -> Red channel/Hue channel

Y axis -> Blue channel/Saturation channel

Z axis -> Green channel/Luminance channel

Keypoints were taken and created a plane that best fitted those points, then projected every point in the image to this plane.



As it is seen, both image pairs turn to the same colour space, but this colour space is distorting all the colours in the image, so the initial approach must be reconsidered.

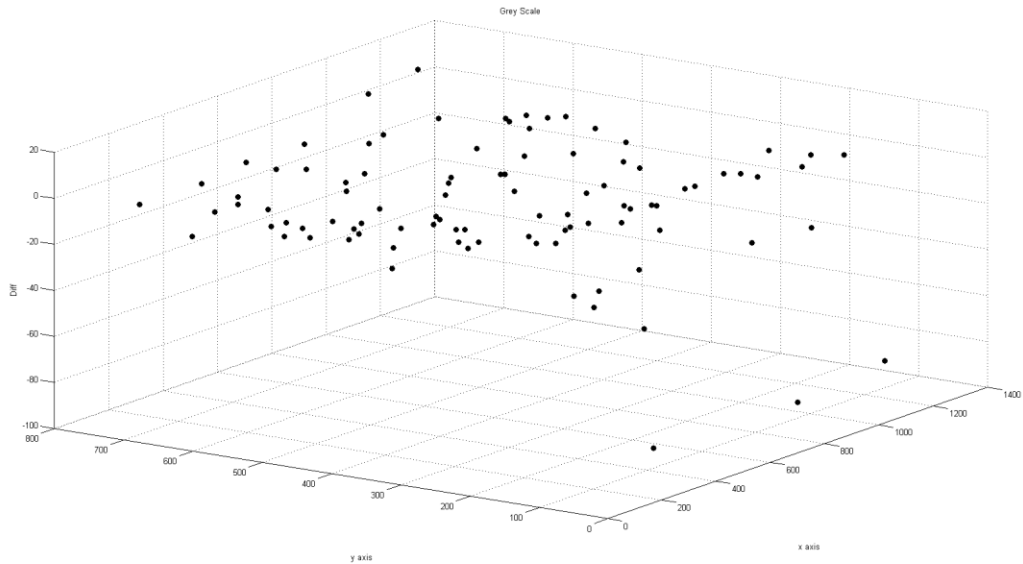
As the second approach, the 3D space is defined this way:

X axis -> x coordinate of pixel

Y axis -> y coordinate of pixel

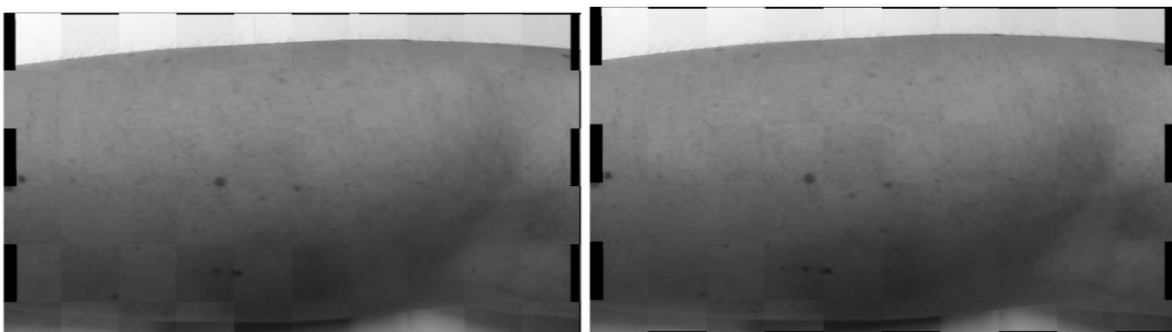
Z axis -> Intensity value of the pixel

For colour images, this process must be applied to all colour channels.



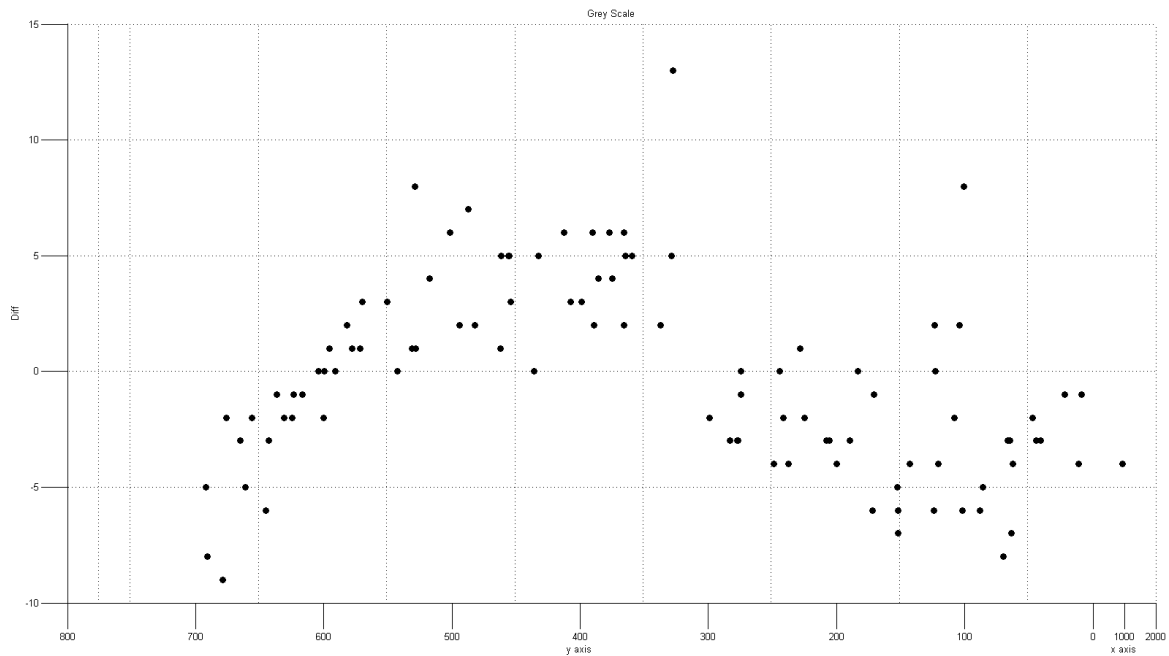
This graph shows the difference of points from base image to target image, in this graph noise points are visible on the lowest right part of the plot. If these points are included, the plane will slightly tilt downwards. Hence, these few points must be removed.

After removing these points, the projection to the plane and checkerboard visualization is done.

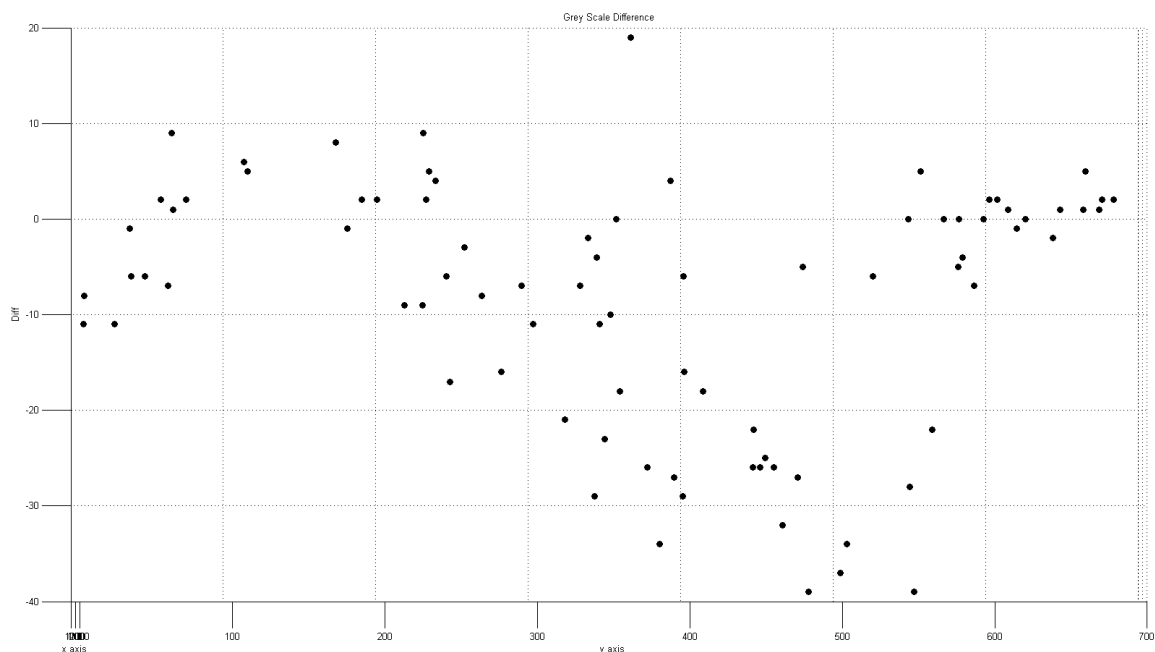


On the left the plane is computed only with the keypoints, so less points in the plane should mean that its precision is less than if hundreds of points are taken. So, in the right image 100 random points were taken and transformed through the homography so they match.

Changes between both images are limited, and the checkerboard shows clearly every tile boundaries. To understand what is happening, the difference between colour projected image and the image before the projection are compared and they only show a variance of 1 to 2 intensity levels, in the RGB scale of 0-255. To understand these changes, it is interesting to see the graph to understand the plane.



In this image, it is clearly seen that the plane approach is invalid, and it should rather be a sinusoidal surface. Therefore, the variance of the image was nearly the same for the entire image, this creates a plane almost parallel to the ground.

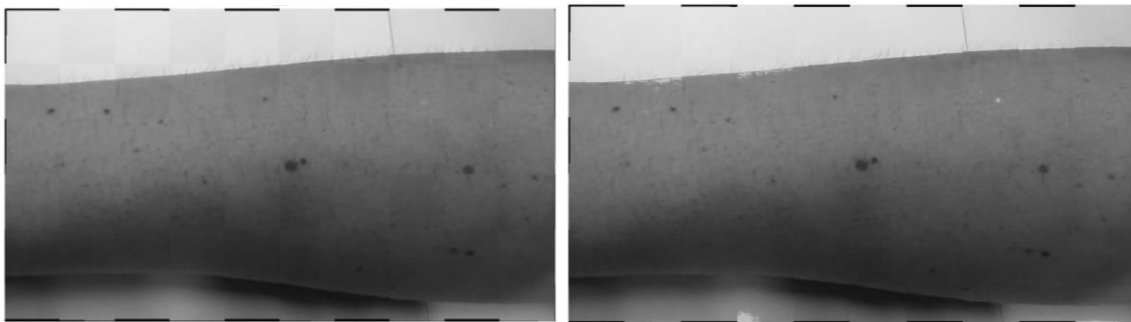


This graph is from a dataset where the colour difference between frames is subtler than the previous one, and as it is not as clear that it is a sinusoidal surface, it is appreciated that it still is not an even plane.

If these graphs are seen as waves, the time period is so large that it is a bit over a wavelength through the entire image. And the amplitude is big enough to not make it a plane.

5.8.b Histogram matching

In the left image, we see the image without any colour correction. In the right one the colour correction was applied to the target image.



The image on the right was the result of doing it with Matlab's integrated functions, we see the artefacts along the left upper edge of the arm, one white dot near the top right edge, and outside the arm, in the ground there are artefacts which were not there before.



In this image, we still see some artefacts along the arm, which can cause confusion to the algorithms that will look for moles in the skin. This image was treated as HSV and only modifying the luminance channel. At first it might look like it is a problem

with the saturation, but in fact it is a problem in which several similar values map to the same value, therefore creating the artefacts if there's a cluster of these values.

5.9 Super-resolution qualitative evaluation



The results illustrated below have been merged at 50%-50%.

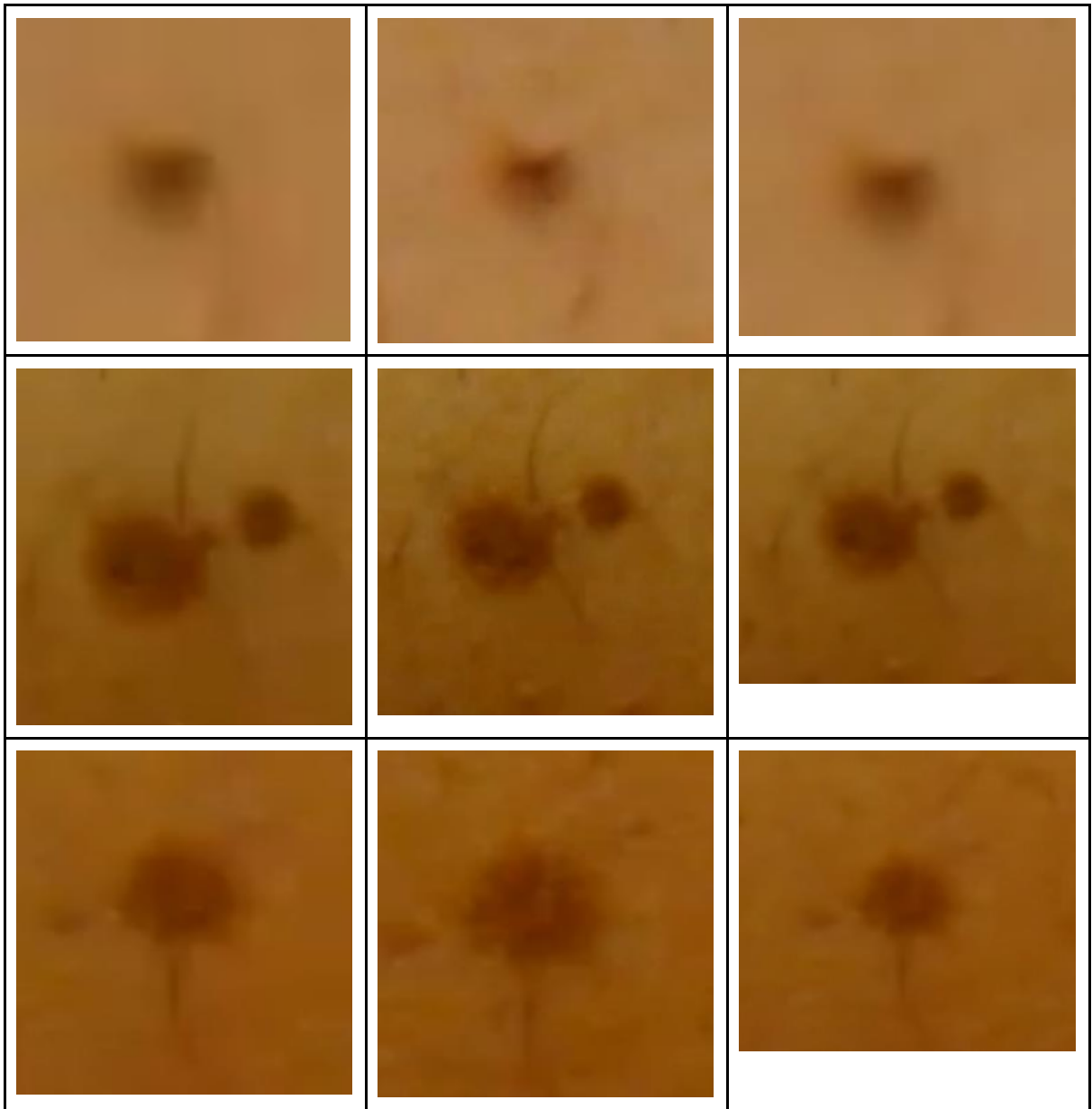








But what it is interesting in this phase are the moles close ups.

Top left		
Base	Target	Mosaic



Top right		
Base	Target	Mosaic
		



Bottom left		
Base	Target	Mosaic
		
		

From these results, it can be understood that maybe a 50-50 weighted blending might be too much, results look smoothed. So, priority to the base image must be given.

6. Conclusions and future work

When doing the stitching not every case will stitch perfectly without distortions when using a manual input of mole locations. This is because locations of the keypoints to build the homography have a massive impact, the more distant and well distributed will give the best results. Also, when no moles, or there are less than 4, the images will not stitch with this method, and SIFT will be required. As seen in the results, manual keypoint annotation stitching works better with well distributed keypoints and low movement between frames. SIFT for getting keypoints works well and it can stitch images that are farther away, the main downside to this method is time, and it is slow in low textured images such as these, if the background is richer than the datasets presented here, it will turn out slow to stitch 2 images, the full sequence can take a lot of time to process. Harris method does not work well and fall behind SIFT.

The time taken by a manual or any distinct method to find moles keypoints must be calculated and compared to SIFT's time, if SIFT is slower a metric must be found to check if the homography from the manual method is better or the same as SIFT's. Because if the given keypoints are too close one to another, the distortion can be huge towards the edges. This gives us an idea, in which when RANSAC selects keypoints to compute the homography, when one point is selected, points in a certain radius are not considered, this could also make RANSAC go through fewer iterations. But also, it must be stated, how big or small this area should be.

For colour correcting the image, histogram matching works well besides being fast and can be used before the homography is computed, so this can be used before matching the images and the resulting mosaic will have a better colour matching.

The plane approach did not work, and that is because after many tries and different datasets involved the conclusion is that it forms an uneven surface and the normal of this is different at each point, this could be costly, but the result would be interesting to see if the differences between intensities converge to 0. In this method, if the plane is to be computed with many random points, the homography must be calculated prior to any of this process, thus making it slower to test.

Super-resolution is a wide field, and in this project the aim was to reduce noise in the final mosaic, using the information from the base and target images. This requires many tests and test which opacity is given to each image. Also, if one of these images is out of focus, the resulting will be out of focus too, therefore making the mosaic useless or not good enough to check the mole's shape.

To solve this problem, a metric to compute the blurriness of the images would be necessary, this could also help remove data and save iterations from happening, making the full sequence compute faster.

Annex

A.1 VLFeat compilation

To make use of this library we must download the binary distribution, unpack and compile it. For the compilation process a full version of Microsoft Visual Studio is required because the Visual Studio Command Prompt and some other utilities from it are needed. In the Makefile we must specify which version of Visual Studio we have installed in our machine.

To start the compilation, we must enter the nmake line VLFeat specifies on its website. Those errors appeared because Matlab was 64 bit whereas Visual Studio was 32bits, the Makefile could find both folders properly without adjusting any extra parameters but it gave the following error: NMAKE: fatal error U1077 the path to a Visual Studio's link.exe file and return code: '0x460'. To overcome this message and continue the compilation process the following line must be run from the same Command Prompt: `%comspec% /k "C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\vcvarsall.bat" amd64`.

Once that instruction is run we must run again the nmake line, and the process will end smoothly. This means that the library is compiled but now we must link it to Matlab for us to take advantage of it. So, under Matlab's toolbox directory with Administrator's privileges we must create an autorun.m file, in this file we just must write `run('path')` where path is where vl_setup.m file is in the binary we downloaded. Now it's time to make sure everything is setup right so when we open a Matlab instance, at the command window we shall see a greeting message or nothing at all (unless there aren't any other plug-in installed). If we happen to get an error saying that there's a problem with the path on the autorun file we wrote, the safest way of fixing it is typing all the path instead of copying and pasting it from the explorer or the security properties of the file.

References

- [1] Skin Cancer. (2017). Skin Cancer Foundation. [online] Available at: <https://www.skincancer.org> [Accessed 1 May 2017].
- [2] Skincancer.org. (2017). The Stages of Melanoma - SkinCancer.org. [online] Available at: <http://www.skincancer.org/skin-cancer-information/melanoma/the-stages-of-melanoma> [Accessed 18 Jun. 2017].
- [3] PDQ Cancer Genetics Editorial Board. Genetics of Skin Cancer (PDQ®): Health Professional Version. 2017 May 11. In: PDQ Cancer Information Summaries [Internet]. Bethesda (MD): National Cancer Institute (US); 2002-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK65895/>
- [4] Litchfield, S. (2017). Camera gimmicks help - but for best quality you need a (much) bigger sensor. [online] All About Windows Phone. Available at: http://allaboutwindowsphone.com/features/item/19406_Camera_gimmicks_help-but_for_b.php [Accessed 18 Jun. 2017].
- [5] Crisp, S. (2017). Camera sensor size: Why does it matter and exactly how big are they?. [online] Newatlas.com. Available at: <http://newatlas.com/camera-sensor-size-guide/26684/> [Accessed 18 Jun. 2017].
- [6] N. Clark, R. (2016). Does Pixel Size Matter Clarkvision.com. [online] Clarkvision.com. Available at: <http://www.clarkvision.com/articles/does.pixel.size.matter/> [Accessed 18 Jun. 2017].
- [7] Openmvg.readthedocs.io. (2017). multiview — openMVG library. [online] Available at: <http://openmvg.readthedocs.io/en/latest/openMVG/multiview/multiview/> [Accessed 18 Jun. 2017].
- [8] Harris, C., and M. Stephens, "A Combined Corner and Edge Detector," Proceedings of the 4th Alvey Vision Conference, August 1988, pp. 147-151.
- [9] Bennett, S. and Lasenby, J. (2014). ChESS – Quick and robust detection of chess-board features. Computer Vision and Image Understanding, 118, pp.197-210.
- [10] Lowe, D. "Distinctive image features from scale invariant keypoints" International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
- [11] VLFeat.org. (2017). VLFeat - Tutorials > SIFT detector and descriptor. [online] Available at: <http://www.VLFeat.org/overview/SIFT.html> [Accessed 18 Jun. 2017].
- [12] Martin A. Fischler & Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" (PDF). Comm. of the ACM. 24 (6): 381–395.
- [13] Marco Ziulani (2014). RANSAC for dummies.
- [14] Ptgui.com. (2017). Image Stitching. [online] Available at: https://www.ptgui.com/info/image_stitching.html [Accessed 18 Jun. 2017].
- [15] Singleton, M. (2017). Sony's latest smartphone camera sensor can shoot at 1,000fps. [online] The Verge. Available at: <https://www.theverge.com/circuitbreaker/2017/2/7/14532610/sony-smartphone-camera-sensor-1000-fps> [Accessed 18 Jun. 2017].

[16] Levin, A.; Zomet, A.; Peleg, S.; Weiss, Y. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2004, Vol.3024, pp.377-389 [Revista Peer Reviewed]