Project 2: Ophthalmology Image Registration

Leonard Karsunky, Florian Aubert, Zoé Jungi
CS-433: Machine Learning, EPFL, Switzerland

***Abstract*—We present here a novel pipeline to register clinical ophthalmic images using various unsupervised learning algorithms. This includes traditional keypoint detection methods such as SIFT, AKAZE and ORB as well as a deep neural network-based model VoxelMorph. We are provided with patient retinal fundus images under the SPHN network, with the aim to align all images from a patient into the same coordinate system. After extensive preprocessing and alignment, we calculate the mutual information between the aligned and the fixed image to compare best results: in our case SIFT works the best followed by AKAZE. However, further research needs to be done to improve the generalization of our method to different imaging modalities.**

I. Introduction

Image registration, also known as image fusion or image matching, is the process of aligning two or more images into a single coordinate system based on their appearances. Medical image registration seeks to find an optimal spatial transformation that best matches the underlying anatomical structures. It has been a topic of active research for decades. In our project, we want to register clinical ophthalmic images from one and later from different devices. The final goal is to use this algorithm in further research studies involving such images from different devices to identify patients and replacing the numeric IDs.

II. Model and Methods

It is common in eye fundus image registration to use either feature-based registration (FBR) or intensity-based registration (IBR). FBR methods use interest points such as landmarks along with local shape features of their neighbourhood to find point correspondence and estimate the spatial transformation between two images. It might not be a robust model for multi-model registration, but since we are only using a single image type, we use a feature-based approach for the registration. Due to a disease, image features can change shape and size in different image modalities and across time. Therefore, important and generally safe features of retinal images are the intersection of blood vessel segments, as well as their shape and locations and the optical nerve head (ONH), whereas light and dark spots appearing on the eye fundus are susceptible to the changes.

A. Loading dataset

First of all, we load the dataset and sort it according to the patient numbers, macula- or OHN-centered images and if it's an image of the right or the left retina of the patient. In the same step, we improve the contrast in each image and then transform it into grayscale to prepare it for the image processing part. As we use the VoxelMorph algorithm in the process of alignment, we transform the images into squares of 1920x1920 pixels. Thereby, we are not losing important information from the retina, only a part of the black background will be removed.

B. Image Processing

Before we are able to extract the relevant retinal features such as the blood vessels or the optic nerve, it's necessary to perform good image processing including contrast improvement, filtering, thresholding, and denoising.

1) Improving contrast: We increase the contrast using enhanced local contrast method. Transforming the initial image into the LAB colour space. This means that for every pixel we get a value for brightness 'L', a value for colour and intensity on a range from green to red 'a' and a value for colour and intensity on a range from blue to yellow 'b'. These values are stored in three matrices (L, a, b). Next, we apply contrast limited adaptive histogram equalization (CLAHE) to the brightness matrix L, and merge it back with the matrices a and b. Histogram Equalization can be used to improve the quality and contrast of an image. If an image is bright or dark, the pixels are restricted to a certain range, illustrated as a confined area of filled bins on a histogram (Figure 1).

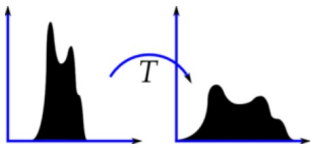

Fig. 1: Stretching the histogram on both ends in order to improve the contrast of the image means that the distribution of the pixel brightness is getting wider.

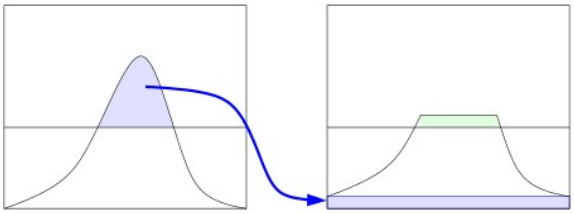

Fig. 2: Histogram bins that are above the clipping limit are redistributed uniformly among the other bins.

The so-called stretching over the image only gives good result though, when the brightness is the same on the entire image. For the retinal images this is not the case: they contain reflecting areas that are very bright and absorbing areas that are dark. To address this problem, we use CLAHE,

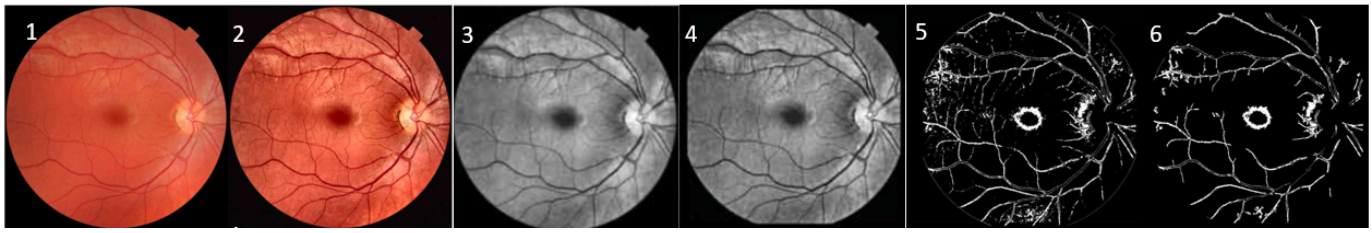

Fig. 3: Showing every important step of the preprocessing part using an example image of the retina: image 1 to 2 represents the contrast improvement, 2 to 3 the grayscaling, 3 to 4 the effect of the Naka-Rushton filter, 4 to 5 the thresholding and the final image 6 is obtained using the denoising function.

which is splitting the image into small blocks of 8x8 pixels and histogram-equalizes them. To avoid noise amplification, contrast limiting is applied. If any histogram bin is above a defined limit, those pixels are uniformly distributed among the other bins (Figure 2) before applying histogram equalization. Figure 3 illustrates the effect of contrast improvement.

2) Naka-Rushton Filter: To get a more compressed image on grayscale levels, we apply the Naka-Rushton Filter, which is based on the following equation:

$$O(i,j) = \frac{I(i,j)}{I(i,j) + \mu_{window}}$$

where $O(i,j)$ is the output matrix that is the transformation result, $I(i,j)$ the original image matrix and μ_{window} is the average of pixels in the chosen exploration window (here we chose the window size to be 44 pixels).

Compressing grey level of an image is another way of contrast improvement between background and components (Figure 4). The range in which the grey levels are distributed has gotten smaller.

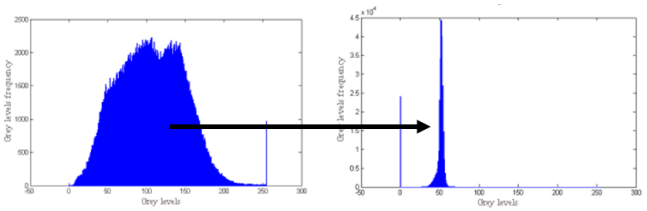

Fig. 4: The effect of the Naka-Rushton filter on the original image histogram (left) transforming it into a grayscale compressed image histogram (right)

3) Thresholding: Now that we have augmented the contrast and compressed the grayscale levels, the vessels need to be separated from the background. The threshold function is applying a threshold to each pixel of the image, below which they are colored white and above which they are colored black. The threshold value is found adaptively through a gaussian-weighted sum, since some images differ in their lighting conditions. The output image shows all vessels in white and the rest of the retina and the whole background in black (Figure 3).

4) Denoising: As it is visible in Figure 3, through thresholding, not only the vessels are colored in white, but also some little points, either belonging to artefact or to the background. With help of the two functions apply_threshold_with_denoising and delete_small_components, based on the non-local means denoising algorithm, these small noise components are removed, rendering the image cleaner and thus avoiding that the noise will be selected as features.

C. Feature Extraction and Alignment

Our retina images being filtered and denoised, the vessels are clearly visible. Using these processed images, we extract relevant features, describe and match them, applying traditional keypoint detection algorithms such as SIFT, ORB and AKAZE, as well as a deep neural network through the VoxelMorph library and are then capable of aligning the two different images.

1) SIFT: The Scale Independent Feature Transform (SIFT) - algorithm is invariant to image scale and rotation. It is designed for images with linear intensity changes and mainly based on following four steps: Scale-space peak selection, keypoint localization and their orientation assignment, local keypoint descriptor and keypoint matching.

The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel, known as Gaussian blur operator $G(x,y,\sigma)$, with the input image. Generating several octaves of the original image, each octave's image size being half the previous one, every pixel within an octave is being blurred and what results is the blurred image. The next step is to use those blurred images to generate another set of images, the Difference of Gaussians (DoG - using two different σ). These DoG images are used to calculate the Laplacian of Gaussian approximations and thereby to find potential keypoints (a pixel best represented in a specific scale).

Having keypoints that are sufficiently stable and knowing their scale, the next step is to assign an orientation to each keypoint to make it not only scale but also rotation invariant. Depending on the scale, a neighborhood is taken around the keypoint location and an orientation histogram covering 360 degrees is created by calculating the gradient magnitude and direction of each pixel in the neighborhood. Any peak above 80% of the highest value is considered to calculate the orientation.

Each keypoint therefore has a location, scale and orientation. To compute its descriptor, a 16x16 window around the keypoint is taken. This window is represented as a feature vector, which is used to form the descriptor.

Finally, keypoints between two images are matched by identifying their nearest neighbors.

2) ORB: The 'Oriented FAST and Rotated BRIEF' (ORB) algorithm uses local binary descriptors, being an efficient and viable alternative to SIFT with good performance and low cost.

FAST standing for 'Feature from Accelerated Segment Test' is comparing, for a given pixel p, its brightness with the one of the 16 surrounding pixels arranged in a circle. If ≥ 8 pixels are classed different, meaning brighter or darker, p will be selected as a keypoint. To add an orientation component to this feature selection, ORB uses a multiscale image pyramid. Each level in the pyramid contains different versions of the image at different resolutions. By detecting keypoints at each level, ORB is effectively locating keypoints at different scales, therefore being partial scale invariant. ORB then detects the levels of intensity change in the neighborhood of the keypoint, using the intensity centroid.

Together with rotated 'Binary Robust Independent Elementary Feature' (BRIEF), the oriented FAST algorithm and therefore ORB can represent a specific object. Each keypoint found by FAST will be described by a binary feature vector or descriptor containing only zeros and ones. BRIEF thereby uses a Gaussian kernel to smooth the image, preventing the descriptor from being sensitive to high-frequency noise.

Even though ORB is much more efficient and faster than SIFT or AKAZE, it presents performance issues that can be seen in the results, mostly due to finding a very limited number of keypoints.

3) AKAZE: The Accelerated KAZE-algorithm uses local binary descriptors as does the ORB-algorithm. It is designed for nonlinear scale spaces, faster, but weaker in accuracy performance than KAZE.

First part consists in computing the contrast factor to build the non-linear scale-space. After smoothing the image by a Gaussian filter, the maximum absolute gradient value of the image (h_{max}) is calculated by using each pixel individually. Using a histogram of 300 bins and finding the histogram index i at which the 70% of the gradient histogram is achieved, we can compute the contrast factor. Second part consists in building the non-linear scale-space with help of which we can detect and match the features. Because the scale-space levels are built by numerically solving a partial differential equation iteratively using the Fast Explicit Diffusion scheme (FED) with varying time steps and additionally for each pixel, the conductivity function is computed, AKAZE is computationally intensive and time-consuming. Like other scale-spaces, the one in AKAZE is a pyramidal framework consisting of octaves and sub-levels. Each octave is quarter the size of the previous.

The keypoints are then selected by comparing the pixels in previously computed DoH images using the determinant of the Hessian (DoH) blob-detector with surrounding window of size 3x3. If the value of this pixel is greater than the surrounding ones and a predefined threshold, and if compared spatially with keypoints in the same sub-level to exclude repeated keypoints, it is extracted as a keypoint.

4) VoxelMorph: In addition to transforming the images in squares, we need to reduce the image dimension to only 30% of original pixels for the VoxelMorph algorithm to work.

The VoxelMorph model is unsupervised and learns a parametrized registration function from a collection of volumes enabling the alignment of a new pair of volumes from the same distribution by simply applying the learned function on the given volumes. Since we use a convolutional neural network (CNN), which takes two input volumes and outputs all voxels of one volume to the other, we need to optimize the parameters of it, i.e. the convolutional kernel weights. This is done by using a training set of volumes from the dataset of interest. The optimization problem can be written as

$$\begin{aligned}\hat{\Phi} &= arg\min_{\Phi} \mathcal{L}(f, m, \Phi) \\ &= arg\min_{\Phi} \mathcal{L}_{sim}(f, m \circ \Phi) + \lambda \mathcal{L}_{smooth}(\Phi)\end{aligned}$$

where f and m denote the fixed and moving images defined over n-D spatial domain $\Omega \subset R^n$ containing single channeled, grayscaled data, respectively, Φ is the registration field mapping coordinates of f to coordinates of m, $m \circ \Phi$ represents m warped by Φ, the function $\mathcal{L}_{smooth}(\cdot)$ imposes regularization, λ is the regularization trade-off parameter and the function $\mathcal{L}_{sim}(\cdot, \cdot)$ measures image similarity between its two inputs depending on the intensities and the spatial regularity of the deformation. We use stochastic gradient descent to minimize the differences between $m \circ \Phi$ and f and find optimal parameters $\hat{\Phi}$ using the training dataset.

We used 10 epochs with 100 steps per epoch to train our model with an average time of about 30 seconds per epoch.

D. Mutual information

In order to test, if our different alignment methods worked, respectively which one worked the best, we compare the aligned image with the fixed image, using the method of mutual information, which is closely linked to the concept of entropy. It is more general than the correlation coefficient as it works for non-linear relationships too, determining how different the joint distribution of (X, Y) is from the product of the marginal distributions of X and Y. MI is the expected value of the mutual information, equaling zero only in case of independence of X and Y. As it only has a lower boundary it is difficult to interpret the result obtained with mutual information (MI is always non-negative). Therefore, we use the normalized mutual information which returns a value of 1 for identical, and 0 for independent sources. For the project we were asked to give a measure that returns 0 for identical, and 1 for independent images. Therefore we use (1 - MI) as a measure of comparison and for simplicity we call that value MI.

III. Results and Discussion

Using SIFT or AKAZE we get well aligned images for the first two patients, on which we also trained the model.

Unfortunately, for other patients we get mixed results. Some images are well aligned and some others are transformed in an odd way, resulting in ellipsoid or other distorted shapes. The MI for those methods are around 0.68 aligning the images of the first two patients (Table I).

Voxelmorph though, generates images that are blurred and misaligned. The vessels are contorted. The transformation does not align the images in the asked way. However, the MI shows a very low value of 0.47. The Method ORB generates only distorted images and is not appropriate for the alignment of patients images. Consistently the MI is very high at 0.86.

Method	MI	SD
SIFT	0.679	± 0.117
ORB	0.862	± 0.105
AKAZE	0.686	± 0.122
VoxelMorph	0.469	± 0.024

TABLE I: Comparison of the mutual information MI and its standard deviation SD of the different feature extraction and alignment methods run over two patients, showing lowest MI (= best aligned images after the whole process) in grey.

Existing keypoint detection algorithms such as SIFT or AKAZE have shown to be robust for different images in our dataset and are very useful for aligning few images (we only need two!). However, they are not robust for retinal fundus images, which present drastically different intensity distributions over the image and/or low blood vessel resolution. On the other hand, neural networks such as VoxelMorph require vast amounts of training data to give accurate results, with 50 images in our training set not being enough to accurately register images.

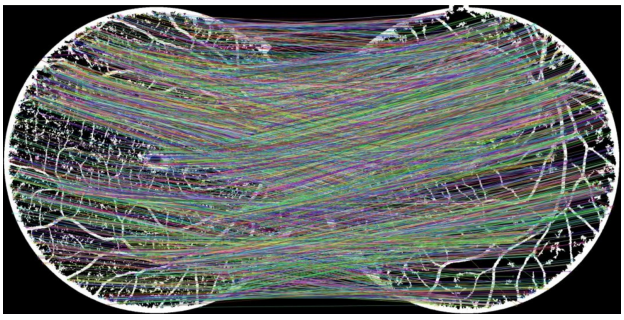

Fig. 5: Visualization of feature matching mechanism of an original and a misaligned image (using AKAZE).

The two methods SIFT and AKAZE perform on a similar level (Table I). The MI of SIFT is slightly smaller, indicating slightly better results.

The method of ORB did not achieve a proper alignment, which is supported by the very high value of MI. We suppose that the algorithm was not capable of finding enough feature points to generate a satisfying homography matrix. The low MI value for the voxelmorph function was confusing at first, indicating a high similarity, the result still being unsatisfactory. Our hypothesis is the following: A low MI indicates that by knowing the pixel of one image, one can predict with a high probability a pixel of the other image. This is the case if the images are similar, as well as when the pixels are transformed following a clear pattern. Voxelmorph transforms pixel per pixel, whereas the other methods multiply the image by a homography matrix, and maybe the MI algorithm detected the pattern of Voxelmorph and could therefore better predict the outcome, even though it was not the outcome we wished for.

Further steps would be to better elaborate a dynamic thresholding. It would better take into account the brightness of an image. Some images have a high reflection on the periphery, it could increase the performance to write functions that remove those parts. We measured the MI for well aligned images and got values around 0.63 (impossible to differentiate with the naked eye!). As we got bad alignment for the patients where the threshold() function is not capable of extracting the vessels, the misalignment causes high MI of up to 0.8. We could implement a function to restrict, if needed, the output to patients, where the MI of the aligned images is below a specific cut-off, i.e. 0.64.

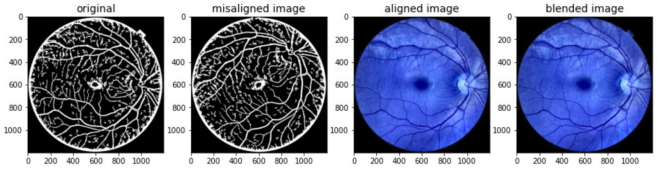

Fig. 6: Example of a misaligned image being processed and aligned. In the last picture the original and moving images are superposed to visualize that there aren't any differences after the alignment.

The similarity measure of mutual information is a convenient way of comparing two methods, both showing a result that can not be distinguished by the bare eye. Though, it can cause problems, when comparing methods that use completely different approaches. That was the case with the comparison between VoxelMorph and the homography methods. Nevertheless, it is far more elaborate and consistent than primitive measures like the mean squared error.

IV. Conclusion

Using different alignment methods after sufficient data processing, it is possible to register clinical ophthalmic images stemming from one device. With the comparison of the mutual information of the results of the different algorithms, we can conclude that SIFT and AKAZE (MI ≈ 0.68) perform the best. Although our algorithms in registering the images gives acceptable results, the aligned image cannot be distinguished from the fixed one, we propose to investigate in a more general algorithm to adapt to more imaging modalities, i.e. images from different devices, and to include diseased retinal images, reaching our final goal to replace the numeric IDs of patients in hospitals.