Comparing the performance of image registration frameworks (Voxelmorph, SPAM, Airlab) on video of retinal

Anh Quang

January 4, 2024

1 Introduction

The retinal video consists of retinal images with displacement between the frames, which results in an un-smooth video. For this reason, frames in the retinal video need to be register to compensate for the motion between the frames. For image registration purpose, the following three methods are used: Voxelmorph (4), The Software for the Practical Analysis of Materials (SPAM) (7) and Autograd Image Registration Laboratory (Airlab) (6). The results of registration (including the loss and the registered video) are compared and stored. These results will be also used to compare with the algorithm I am working on in the lab of Professor Kybic, based on the paper "Fast registration by boundary sampling and linear programming".

1.1 Data

Several retinal videos (Sada dataset) are provided. The video are obtained from different patients, using different wavelengths as illumination source. Each video contains approximately 300-400 RGB frames, each has size 970x1224 pixels.
Since the retinal images cut from the video are blurred and difficult to be registered (as the performances of the 3 frameworks on these images are not good), I tried to use the MNIST data and another simpler image of vessels, in order to have a better conclusion on the performances.

1.2 Voxelmorph

Voxelmorph is a learning-based framework for image registration, which can be applied for several registration purposes such as medical images, hand written digits... This framework receives pairs of images as input and learns a function that aligns these images by mapping them to a deformation field (4).

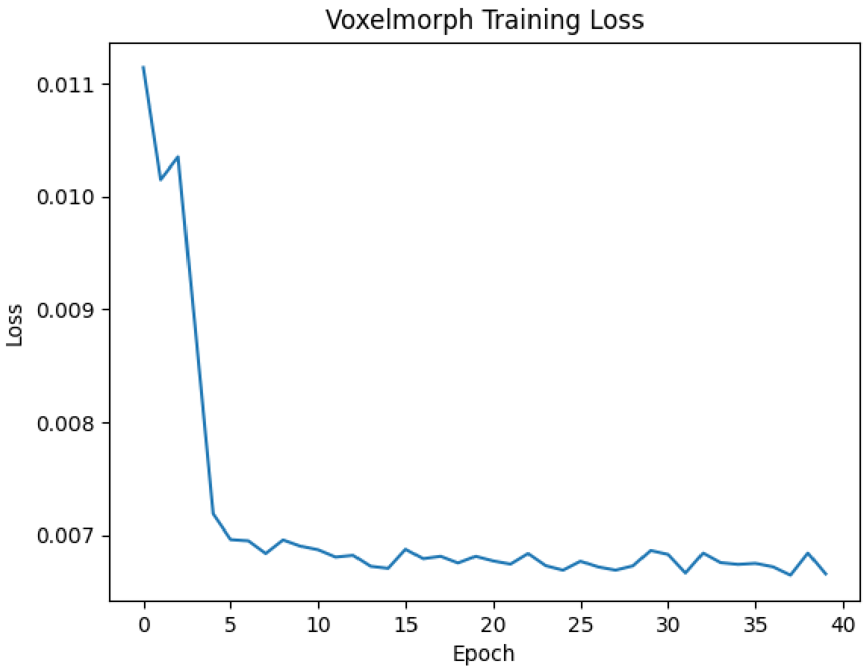

Figure 1: Voxelmorph training loss for retinal images

1.2.1 Training data and validation data

Frames are extracted from the retinal videos. There are 30,000 training images and 300 validation images.
Total training time in hour: 14.88
Number of epoch: 40
Time to train each epoch: 0.372 hour
Resized image: 128x160 (I resized the image because it takes longer to train the model with full size image)

1.2.2 Limitation

One of the limitation of voxelmorph framework is that it can only be used for the same type of images used for training. For example, a voxelmorph framework for brain images cannot be applied for retinal images. Thus, it is required to train a new framework for each type of image.

1.3 SPAM

SPAM is a Python-based enhanced library used mainly for data analysis in material science (spa). It can be used for several purposes such as image (2D/3D) correlation, multimodal registration... The registration is based on the digital image correlation, which uses Newton's method to find the convergence of the loss function. SPAM provides practical and useful functions for manipulating images, for example, function

"spam.deformation.computePhi" returns a homogenous transformation matrix from a given translation and rotation parameter that can be applied to an image, or function spam.DIC.applyPhi() to transform an image with the homogenous transformation matrix. There are many more functions for computing histogram or morpholigical operations, but for the semester work, the two functions to find and apply transformation between a pair of images is sufficient.
In order to apply SPAM on retinal video, there is no need to provide many images as training data like Voxelmorph. For one video, I take the first frame as the fixed image (reference image) and register all the subsequent frames (moving images) to the first frame.

Parameter:
The registration result is returned as a matrix consisting of translation and rotation angle ϕ.
Number of iterations: 300
Condition to stop iteration: $\Delta\phi_{min} = 0.01$
(Registered video for resized image: $output\_spam\_128.avi$, registered video for full size image: $output\_spam\_fullsize\_1k5iteration$)

1.4 Airlab

Airlab is an open Python-based enhanced library for medical image registratiom. It also combines optimization library such as torch Adam and basic image IO library such as SimpleITK (air).
The usage of Airlab is similar to SPAM. While SPAM can work on original images from the video (970x1224), Airlab has to normalize the images before the registration process.

Parameter:
The registration result is returned as a matrix having the same size as the image, whose elements are the displacement of each pixel.
Number of iteration: 2000 in experiment 1, 1000 in experiment 2 (since the loss does not decrease)

2 Experiments and results

The loss used to compare the performance of 3 frameworks is the mean squared intensity difference between the fixed (F) and the transformed version of the moving image (G'). X,Y are the sizes of the images.

$$MSE = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} \left(F(x,y) - G'(x,y)\right)^2 \tag{1}$$

2.1 MNIST dataset number 5

In this section, the performance of 3 methods is compared by using the same sets of fixed image and moving images. There is only one fixed image used, which is the image of number five in the MNIST data set. I tried with MNIST data set because this is a simple data set (bright pixels on black background). The images have size 28x28, since the Voxelmoprh model is trained with image size 28x28.

2.1.1 Synthetic moving image

In the first subsection, the moving images are synthetically generated from the fixed image by applying affine transformation of different values of translation (in *pixel*) and rotation (in *degree*).

$$\begin{aligned} t_x &\in [-5, 5] \\ t_y &\in [-5, 5] \\ \phi &\in [-60, 60] \end{aligned} \tag{2}$$

Homogenous affine transformation matrix T_{trans}:

$$T_{trans} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

I created the first set of moving images by translating the fixed image in x- and y-direction by the same amount. For example, the first moving image is the fixed image translated by -5 pixel in x-direction and y-direction, respectively, and the translation amount is increased equidistantly to maximal 5 pixels in each direction. There are 100 moving images in the first set.
Homogenous affine rotation T_{rot}:

$$T_{rot} = \begin{bmatrix} \cos\phi & -\sin\phi & t_x \\ \sin\phi & \cos\phi & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

I created the second set of moving images by rotating the fixed image by $\phi degree$ around the center point of the image. For example, the first moving image is the fixed image rotated by $-60°$ around the center of the image. There are 100 moving images in the second set.
Each set of moving images is registered to the fixed image by appling the Voxelmorph, SPAM and Airlab. The loss is the squared intensity difference between the fixed image

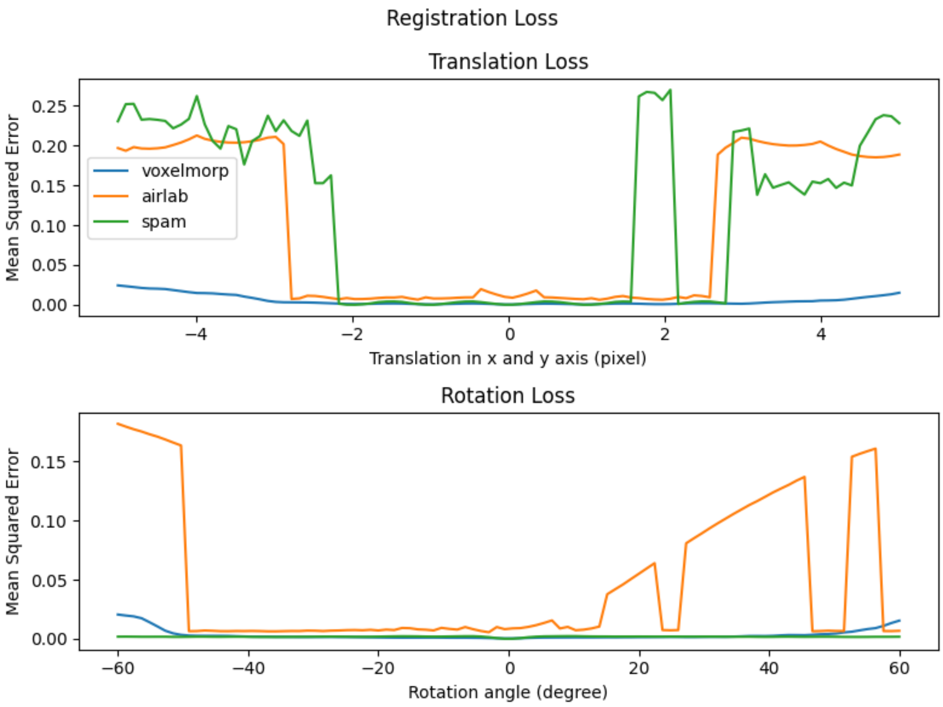

Figure 2: Upper diagram: Moving image is a translated version of fixed image. Lower diagram: Moving image is a rotated version of fixed image (git)

and moving image. The results of registration of the first set (translation) is shown in figure 2: We can see that, by translation only, Airlab can well register 2 images when the translation in each direction is smaller than approximately 2.8 pixels ($[-2.8, 2.8]$) (loss is about 0.02). When the translation in each direction exceeds this range, the loss is about 10 times higher (0.2). On the other hand, by rotation only, Airlab can well register if the rotation angle is in $[-50°, 15°]$. In the range $[15°, 60°]$, the loss fluctuates unexpectedly. I guess this is because the algorithm is stuck in some local minima, thus cannot provide good registration result.

By translation only, SPAM can well register 2 images when the translation in each direction is smaller than approximately 2 pixels ($[-2.2, 1.6]$). When the translation in each direction exceeds this range, the loss is about 20 times higher. By rotation only, SPAM can do the registration very well, as the loss is approximately only 0.002.

For Voxelmorph, the framework can register well in both translation and rotation cases. The loss is not higher than 0.05

From this result, I can conclude that, by rigid affine transformation, Voxelmorph performs better than Airlab and SPAM. SPAM can register well in rotation case but not translation case. Airlab

2.1.2 Original MNIST data set number 5

This experiment is to test the ability to perform non-rigid registration of the 3 frameworks. The Voxelmorph model for MNIST data is given as an online tutorial and can be downloaded. This model uses 5000 training images and 1000 validation images. In this experiment, I used 1000 images of number 5 in the MNIST data set, which are not in the training data of Voxelmorph. The registration loss of each framework is shown in table 2.1.2.

	Airlab	SPAM	Voxelmorph
Loss (mean for 1000 pairs)	0.09	4909 (diverge)	0.002
Speed(frame/second)	0.128	37.43	14.83

We can see that only Voxelmorph can perform well on the MNIST data number 5. The loss (mean squared error) of Airlab is too high, thus, the transformed moving image is almost the same as the moving image (ideally the transformed image should look like the fixed image) 3.

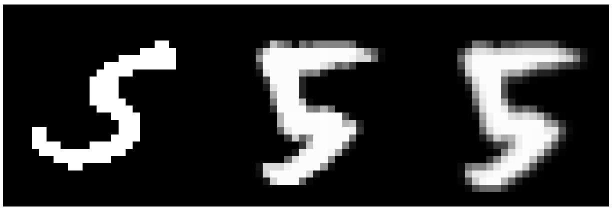

Figure 3: Left: fixed image. Middle: moving image. Right: transformed version of the moving image

2.2 Image of retina/vessel

2.2.1 Retinal images cut from video

In this section, I tried to apply the 3 algorithms to register the image sequence cut from the video. The original image size is 1920x1224. Each image is resized to 128x160. For 3 frameworks, only SPAM can well register the image sequence (see video). However, SPAM cannot perform well on full size images. Then I tried with another image of vessels, which looks quite simpler than the retinal image from the video.

2.2.2 A simpler image of vessels

I only use this image for Airlab, because Voxelmorph requires data and time to train the model.
6. The moving image is generated by rotating the fixed image around its center by angle $1°, 2°$... and translating by 10 pixels in the x-direction, 5 pixels in the y-direction. Airlab can only register the fixed image and synthetic moving image with rotation angle $1°$ only (the overlay can be seen in 5), I tried to draw some pattern to the fixed image and generate the moving image again, in order to assist the registration process. Firstly, I tried to draw some rectangles at the pixels of high gradient on the fixed image, and generate the

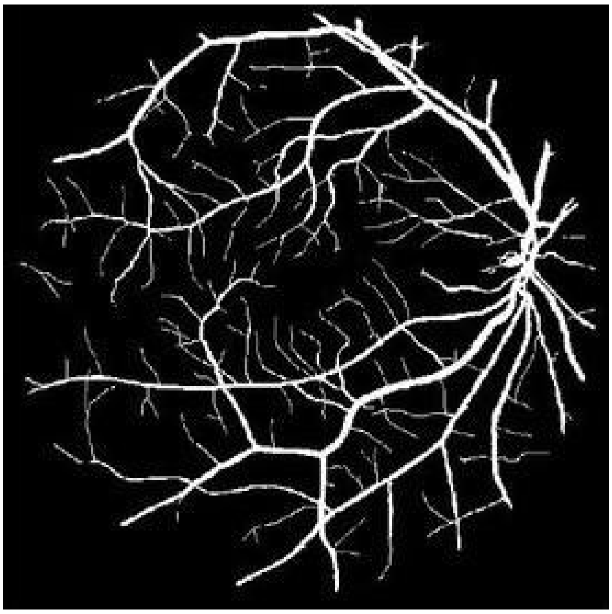

Figure 4: Image of clear retinal vessels (5)

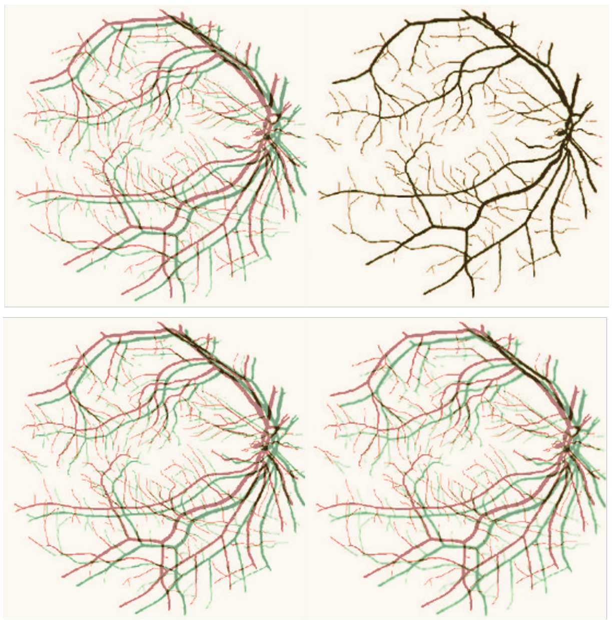

Figure 5: $t_x = 10, t_y = 5$. Upper row: $\phi = 1°$. Lower row: $\phi = 2°$. Left column: overlay before registration. Right column: overlay after registration. Registration works up to $\phi = 1°$, increasing ϕ makes the registration result worse.

moving images again. Airlab can well register pairs of images up to the rotation angle of 4° (same amount of translation), from 5°, it cannot register the fixed and moving image. The overlay can be seen in 6. After that, I tried to draw a more specific pattern on the

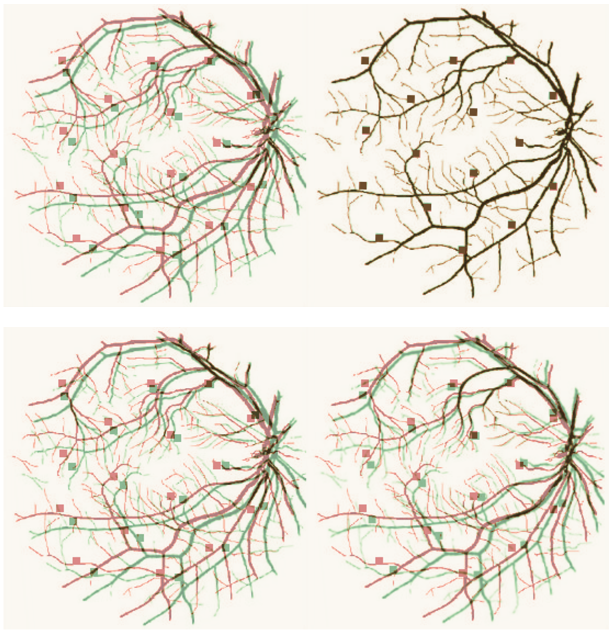

Figure 6: $t_x = 10, t_y = 5$. Upper row: $\phi = 4°$. Lower row: $\phi = 5°$. Left column: overlay before registration. Right column: overlay after registration. Registration works up to $\phi = 4°$, increasing ϕ makes the registration result worse.

fixed image, which is a line at random position, then I re-generate the moving images. Airlab can well register pair of images up to the rotation angle of 7° (same amount of translation), from 8°, it cannot register the fixed and moving image. The overlay can be seen in figure 7. From the result of this experiment, I guess that Airlab cannot register well the retinal image sequence cut from the video because the image is too complex. When using a simpler image of vessels, Airlab can work to some extent, and by including some patterns, it can register better. I guess Airlab can only work well on images with specific features.

3 Results

4 Discussion

4.1 Airlab

Good key point matching could lead to better registration results for Airlab. For the retinal images from the video, there is no distinct feature, therefor Airlab cannot do the job. It seems that Airlab also does not work for non-linear registration (not work with MNIST data set).

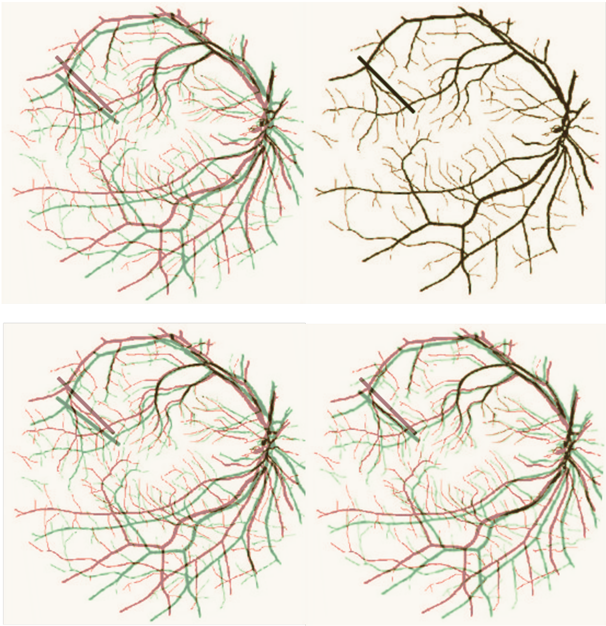

Figure 7: $t_x = 10, t_y = 5$. Upper row: $\phi = 7°$. Lower row: $\phi = 8°$. Left column: overlay before registration. Right column: overlay after registration. Registration works up to $\phi = 7°$, increasing ϕ makes the registration result worse.

4.2 SPAM

SPAM can register the retinal sequence well, but only with the resized image (128x160). SPAM performs worse with MNIST data set. This framework may not work for non-linear registration.

4.3 Voxelmorph

Voxelmorph can register well the fixed and moving images in all the above experiments. However, it does not work for the retinal video. According to the Voxelmorph paper (4) : "We also assume that f and m are affinely aligned as a preprocessing step, so that the only source of misalignment between the volumes is nonlinear. ". In the experiment, in which the number 5 is manually translated and rotated, Voxelmorph can still register the pairs of images, while in some neighboring frames of the video, where the frames are not so much affinely misaligned, Voxelmorph cannot register those frames. This problem requires more experiments to be fully understood.

References

[air] Autograd image registration laboratory.

[git] Chatgpt and github copilot.

[spa] The software for the practical analysis of materials.

[4] Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. (2019). Voxelmorph: A learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38(8):1788–1800.

[5] Pan, L., Zhang, Z., Zheng, S., and Huang, L. (2021). Msc-net: multitask learning network for retinal vessel segmentation and centerline extraction. *Applied Sciences*, 12(1):403.

[6] Sandkühler, R., Jud, C., Andermatt, S., and Cattin, P. C. (2020). Airlab: Autograd image registration laboratory.

[7] Stamati, O., Andò, E., Roubin, E., Cailletaud, R., Wiebicke, M., Pinzon, G., Couture, C., Hurley, R. C., Caulk, R., Caillerie, D., Matsushima, T., Bésuelle, P., Bertoni, F., Arnaud, T., Laborin, A. O., Rorato, R., Sun, Y., Tengattini, A., Okubadejo, O., Colliat, J.-B., Saadatfar, M., Garcia, F. E., Papazoglou, C., Vego, I., Brisard, S., Dijkstra, J., and Birmpilis, G. (2020). 'spam': Software for practical analysis of materials. *Journal of Open Source Software*, 5(51):2286.