

A mutual information based medical image registration methodology based on JVHW entropy estimator

Sihan Li, Tsinghua University

Abstract—Mutual information (MI) based medical image registration has gained much attention for its robustness. However, Jiao found that maximum likelihood estimator (MLE), although widely used to implement MI-based registration, is sub-optimal when the number of observations is comparable to the parameter dimension [1]. They proposed a new entropy estimator, JVHW estimator, which is claimed to outperform MLE estimator. Based on previous works, we propose a mutual information based registration methodology based on JVHW entropy estimator. The contribution of this paper can be divided into three parts: 1) A C++ version of JVHW estimators using only C++ standard library is effectively implemented, 2) a new similarity metric based on JVHW mutual information estimator is created as an elastix component, and 3) the new metric is evaluated and compared with other metrics. Dataset from Retrospective Image Registration Evaluation Project (RIRE) is used to evaluate and compare various implementations.

I. INTRODUCTION

Image registration is a process of aligning two or more images geometrically. These images can be taken at different time, from different viewpoints or from different sensors. It is vital in image analysis when the data collected from different sources need to be combined. To achieve this, a variety of techniques have been developed and researched, aiming for different kinds of applications. [2], [3]

Image registration methodologies consist of three major parts: *Similarity measures*, which measure the degree of similarity between two images. *Transforms*, which transform pixels/voxels from one image domain to another. *Optimizations*, which minimize the similarity measures to achieve the registration.

Similarity measure is a criteria to judge the degree of similarity of two images. It is known as *cost function*, or *registration basis* in Maintz's classification [4]. Commonly used measures include intensity based measures, which analysis the images as a whole, and feature based measures, which detect and match features of two images.

A. Mutual information

Mutual information based registration methodologies use mutual information (MI) or its variant as the similarity measures. It is a kind of area-based metric. The Methodology based on a simple assumption that the gray level distributions of images to be registered are correlated. Mutual information is defined as:

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (1)$$

where $H(A)$ is the entropy of A . It describes the amount of uncertainty about image minus the uncertainty about A when B is known [5]. The most common entropy measure used in paper is Shannon entropy, which is defined as

$$H = - \sum_i p_i \log p_i \quad (2)$$

where p_i represents the probability of a event e_i . In order to register two images, the joint gray value distribution is calculated from them. Other entropy measures include Arimoto entropy [6], Tsallis entropy [7] etc.

A popular variant of MI is Normalized mutual information (NMI), which is overlap invariant, which is defined as

$$NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)} \quad (3)$$

by Studholme in 1999 [8]. To achieve image registration, the MI/NMI between two images are maximized. Mutual information based measures are generally applicable, and does not need complex initialization [5].

B. JVHW estimators

As we mentioned before, MI-based methodologies calculate the mutual information between two gray level distributions. However, in reality the distributions are unknown, so the exact MI can not be calculated and needs to be estimated. Most implementations simply use a straightforward maximum likelihood estimator (MLE) to estimate MI. However, Jiao pointed out that MLE and other previous prevailing approaches can be highly sub-optimal, and proposed a *minimax* entropy/MI estimator, which means the estimator minimizes the maximum L_2 risk [1]. They showed that their estimator outperform MLE and many other estimators.

II. IMPLEMENTATIONS

A. C++ version of JVHW estimator

In order to make a comparison between MLE and JVHW estimator, we are going to re-implement both of them in C++. To reuse as much as code as possible, we have to analyze both estimator and point out what they have in common. Actually, both of the estimator are linear estimators which can be expressed in the following way:

$$H = \sum_{i=1}^n a_j F_j, \quad (4)$$

Where F_j is called “fingerprint”, representing the number of symbols that show up for j times in the sample. The only different between two estimator is a_j . In MLE it follows the following form:

$$a_j = -\frac{j}{n} \log_2 \frac{j}{n}, \quad (5)$$

and in JVHW estimator a_j is calculated during run time. So instead of simply migrating MATLAB or Python code into C++ code, we divide the function into several sub-function. That is:

1) *CalculateFingerprint*: Collect the samples into a certain number of bins to get a histogram, then calculate the fingerprint from it.

2) *est_entro_JVHW*: Estimator entropy from an integer vector using JVHW estimator.

3) *est_entro_MLE*: The same as above, except using MLE.

4) *NormalizeVector*: Normalize the sample from a double vector to an integer vector.

5) *CombineVector*: Combine samples to form a joint-distributed sample.

6) *est_MI_JVHW*: Invoke above functions to calculate the MI.

7) *est_MI_MLE*: The same as above, except using MLE.

To avoid file operations which will take lots of time, we embed `poly_coeff_r` into our source code. That is, we make it a static array, which will be loaded into the memory only once when the program starts. It will take about 314 KB memory. Given that metric function will be invoked continuously during registration, this memory consumption is really acceptable.

B. Elastix component

Elastix is built using CMake, and has been divided into small components. In order to add new metrics, optimizers, transformations or something else into elastix, one has to:

- 1) Implement an `itk` class, which does the actual works.
- 2) Implement an `elastix` class, which takes care of things like parameter-setting.
- 3) Put source files in the correct directory together with a `CMakeLists.txt`, which declares the component.
- 4) Re-compile elastix.

So in order to use JVHW entropy estimator in the metric, we have to implement two classes. Building the classes from nowhere can be a painful and error-prone, so we take most of the code from `AdvancedMeanSquares`, which is a simple metric that iterate over every voxels and compute the mean square error.

C. Optimizers and parameters

Elastix offers a branch of built-in optimizers, including *Adaptive Stochastic Gradient Descent*, *Standard Gradient Descent*, *Quasi Newton LBFGS* etc. However, it should be noticed that most of the optimizers need to know the derivative of the cost function, which is especially hard to acquire when applying JVHW entropy estimator. Therefore, we will pick our optimizer from the ones that do not need to know the

derivative of the cost function. Both *CMA Evolution Strategy* and *Simultaneous Perturbation* has been tested, and the latter one shows more robustness when applied to our problem.

When it comes to parameters, *Maximum Number Of Iterations* has been set to 500.

The source code is available at <https://github.com/ThomasLee969/stanford-project>.

III. RESULTS

To evaluate the two new metrics, we have to make sure that they are valid metric, and can be used to preform image registration. Thus, we perform registration on RIRE dataset using four metrics: *Mattes MI*, *NMI*, *MLE* and *JVHW*. The former two metrics are shipped with elastix. They use *Parzen window method*, also known as *kernel density estimate method*, to estimate the underlying distribution from a series of samples, as depicted in Figure 1

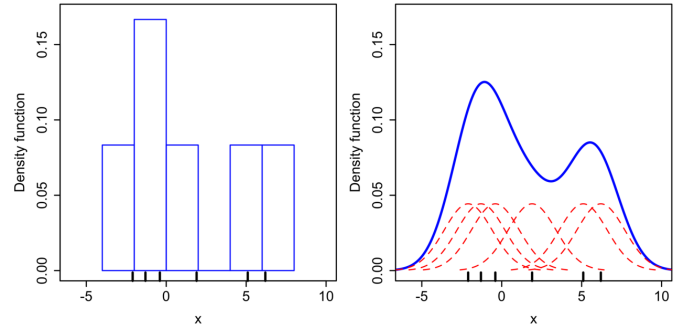


Fig. 1. Comparison of a histogram and a kernel density estimate.

Also, for *Mattes MI*, *NMI* method we also deploy a *Adaptive Stochastic Gradient Descent* optimizer. The result can be seen in Figure 2 and Figure 3

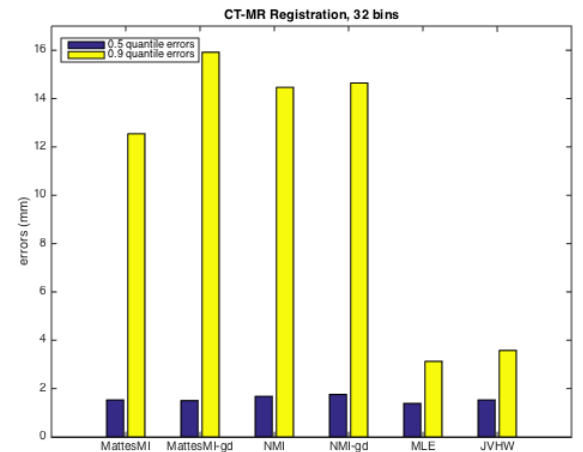


Fig. 2. Quantile errors for CT to MR registration using 32 bins.

We adopt the methodologies using in [9] to evaluate the accuracy of each implementation. The 0.5 and 0.9 quantile errors over all CT-MR and PET-MR pairs of 9 patients have been calculated. 0.9 quantile is used instead of the maximum

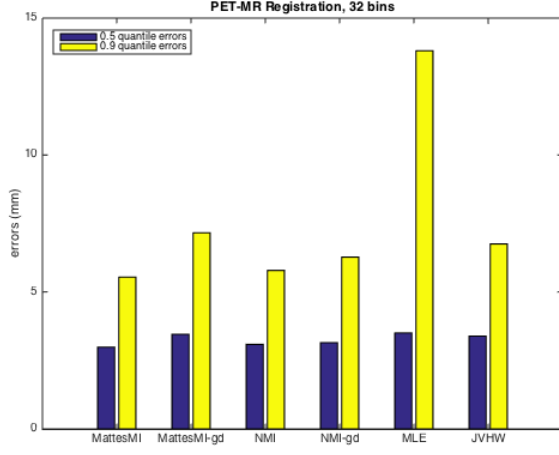


Fig. 3. Quantile errors for PET to MR registration using 32 bins.

value, because the maximum can be affected by a single outliers. It can be seen that we have implemented our metrics correctly. Moreover, our metrics outperform the built-in MI-based metrics.

Having confirmed that we have implemented the metrics correctly, we are going to adjust the number of bins, which is one of the most important parameters in our metrics. The results can be seen in Figure 4 and Figure 5.

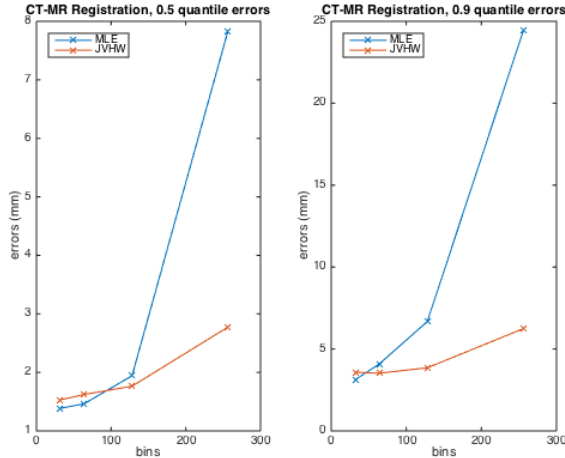


Fig. 4. Quantile errors for CT to MR registration

It can be seen that when there are few bins, MLE performs a little bit better than JVHW. But when the number of bins increase, the errors given by MLE increase faster than the errors given by JVHW. This is because when the number of samples is significantly larger than the support size, the MLE is essentially the optimal scheme. In our situation, the support size is about 10^3 , while the number of samples is about 10^6 . So it can be concluded that our JVHW-based metric outperform traditional MLE metric in general, especially when the number of bins are large.

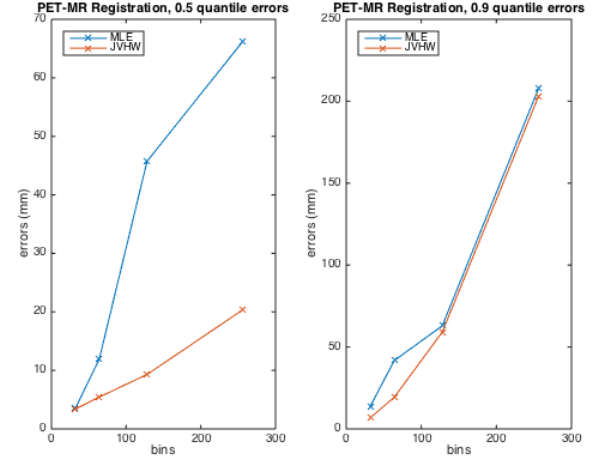


Fig. 5. Quantile errors for PET to MR registration

IV. ACKNOWLEDGMENT

The images and the standard transformation(s) were provided as part of the project, “Retrospective Image Registration Evaluation”, National Institutes of Health, Project Number 8R01EB002124-03, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.

REFERENCES

- [1] J. Jiao, K. Venkat, Y. Han, and T. Weissman, “Minimax estimation of functionals of discrete distributions,” *Information Theory, IEEE Transactions on*, vol. 61, no. 5, pp. 2835–2885, 2015.
- [2] L. G. Brown, “A survey of image registration techniques,” *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.
- [3] B. Zitova and J. Flusser, “Image registration methods: a survey,” *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [4] J. A. Maintz and M. A. Viergever, “A survey of medical image registration,” *Medical image analysis*, vol. 2, no. 1, pp. 1–36, 1998.
- [5] J. P. Pluim, J. A. Maintz, M. Viergever *et al.*, “Mutual-information-based registration of medical images: a survey,” *Medical Imaging, IEEE Transactions on*, vol. 22, no. 8, pp. 986–1004, 2003.
- [6] B. Li, G. Yang, J. L. Coatrieux, B. Li, and H. Shu, “3d nonrigid medical image registration using a new information theoretic measure,” *Physics in medicine and biology*, vol. 60, no. 22, p. 8767, 2015.
- [7] M. Khader and A. B. Hamza, “An information-theoretic method for multimodality medical image registration,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5548–5556, 2012.
- [8] C. Studholme, D. L. Hill, and D. J. Hawkes, “An overlap invariant entropy measure of 3d medical image alignment,” *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.
- [9] J. P. Pluim, J. A. Maintz, M. Viergever *et al.*, “F-information measures in medical image registration,” *Medical Imaging, IEEE Transactions on*, vol. 23, no. 12, pp. 1508–1516, 2004.