Supervised learning in medical image registration

Sokooti, H.

4

Quantitative Error Prediction of Medical Image Registration using Regression Forests

This chapter was adapted from:

H Sokooti, G Saygili, B Glocker, BP Lelieveldt, and M Staring. **Quantitative Error Prediction of Medical Image Registration using Regression Forests**, *Medical Image Analysis*, 2019.

Abstract

Predicting registration error can be useful for evaluation of registration procedures, which is important for the adoption of registration techniques in the clinic. In addition, quantitative error prediction can be helpful in improving the registration quality. The task of predicting registration error is demanding due to the lack of a ground truth in medical images. This paper proposes a new automatic method to predict the registration error in a quantitative manner, and is applied to chest CT scans. A random regression forest is utilized to predict the registration error locally. The forest is built with features related to the transformation model and features related to the dissimilarity after registration. The forest is trained and tested using manually annotated corresponding points between pairs of chest CT scans in two experiments: SPREAD (trained and tested on SPREAD) and inter-database (including three databases SPREAD, DIR-Lab-4DCT and DIR-Lab-COPDgene). The results show that the mean absolute errors of regression are 1.07 ± 1.86 and 1.76 ± 2.59 mm for the SPREAD and inter-database experiment, respectively. The overall accuracy of classification in three classes (correct, poor and wrong registration) is 90.7% and 75.4%, for SPREAD and inter-database respectively. The good performance of the proposed method enables important applications such as automatic quality control in large-scale image analysis.

4.1 Introduction

Image registration is the task of finding the optimal spatial transformation between two or more images. In most registration methods, no assessment of the registration quality is provided, and simply the result is returned. Evaluation of the registration is devolved to human experts, which is very time-consuming and prone to inter-observer errors as well as human fatigue [80]. Automatic quantitative error prediction of registration would decrease quality assessment time and can provide information about the registration uncertainty. Many medical pipelines are based on registered images and it is important to know the uncertainty of registration before continuing to a next phase in order to prevent accumulation of errors. For example, in online adaptive radiotherapy daily contouring of the tumor and organs-at-risk can be performed with the help of image registration [30]. In this task, quality assessment (QA) is mandatory to ensure patient safety. In addition, the accumulation of delivered dose over several treatment fractions is also impacted by the quality of registration [81, 82, 83]. Registration quality therefore has to be checked before the treatment starts. Visualizing the error of registration can also be directly helpful in medical applications before making a clinical decision. Smit et al. [31] localized autonomic pelvic nerves by registering a pre-operative MRI scan to an atlas model that includes nerve information. These nerves are not visible in the MRI scans and are prone to be damaged during a surgical procedure. Utilizing registration uncertainty yield better visualization of the autonomic nerves.

Refinement of registration is another important application of automatic error prediction. Muenzing et al. [32] improved registration by focusing only on regions with high registration error and discarding pixels which are aligned correctly. Registration refinement can also be done with the feedback of human experts by manually adding several corresponding landmarks [84].

Schlachter et al. [85] did a comprehensive study on visualization of registration quality with the help of three radiation oncologists on the DIR-Lab-COPDgene data, which has a slice thickness of 2.5 mm. The [average, maximum] TRE of the landmarks that were rated to be of acceptable registration quality with the conventional visualization method (checkerboard visualization and color blended) was [2.3, 6.9] mm, while with the best visualization method (histogram intersection) [1.8, 3.3] mm was achieved.

A few methods have been proposed to detect the misalignment of a pair of images with the purpose to refine the registration result. Rohde et al. [38] proposed to use the gradient of the cost function to detect which region in the image pair is poorly registered and potentially can be improved. Schnabel et al. [86] suggested to refine the registration by increasing the number of registration parameters in regions with

high local entropy, or with high local variation in the intensity or with relatively steep cost function. In another work, analyzing the shape of the cost function around each voxel was used to estimate the confidence of registration [87]. Park et al. [37] used normalized local mutual information to find poorly aligned regions in order to increase the number of registration parameters. Forsberg et al. [88] utilized the outer product of the intensity gradient as an uncertainty measure in multi-channel diffeomorphic Demons registration. Although the mentioned metrics can be used to improve the image registration, it has not been shown how these metrics are correlated with the image registration error.

Several methods exploit continuous probabilistic image registration by utilizing Bayesian inference to achieve an intrinsic transformation uncertainty measure [89, 90]. However, it has been shown that there is no clear statistical correlation between transformation uncertainty and registration uncertainty [91]. The transformation and corresponding label (of a pair of images) are two random variables and it is not possible to quantify the uncertainty of the corresponding label by the summary statistics of the transformation. Another downside of these methods is that they can only be used for the specific paradigm of Bayesian registration.

Some methods are based on the consistency of multiple registrations between a group of images [92, 93], but these methods cannot be used in pairwise registrations.

In the stochastic approaches, Kybic [94] suggested to perform multiple registrations with random sampling of pixels with replacement. He found a correlation between the true registration error and the variation of the 2D translational parameters. The method was not extended to 3D and to nonrigid registration. Hub et al. [35] calculated the local mean square intensity difference multiple times by perturbing the B-spline grid. They showed that the maximum change of the dissimilarity metric in a local region is correlated with the registration error in that region. The drawback of this method is that it is not efficient in homogeneous areas [95]. In a related work they showed that the variance of the final deformation vector field (DVF) is related to the registration error [95], using the Demons algorithm. However, to find large misalignment a large search region is needed.

In this paper, we turn our attention to methods capable of *learning* the registration error allowing to take advantage of multiple features related to registration uncertainty within a single framework. Muenzing et al. [39] casted the registration assessment task to a classification problem with three categories (wrong, poor and correct registrations). In their method, they mostly utilize intensity-based features, except for the determinant of the Jacobian of the transformation. Although their training samples consist of manually selected landmarks, later they showed that assessing registration in all regions is possible by interpolation [32].

In our paper, instead of casting the uncertainty estimation task to a classification

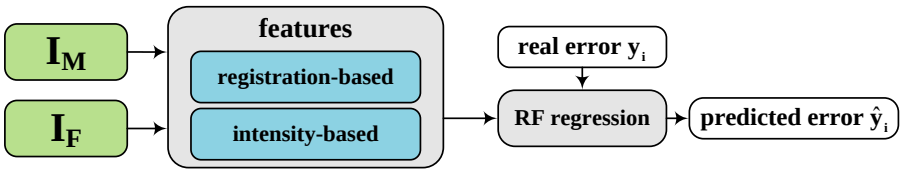


Figure 4.1: A block diagram of the proposed algorithm.

problem, we formulate it as a regression problem. To the best of our knowledge, in the field of continuous prediction of 3D registration error, Lotfi et al. [96] only tested their method on artificially deformed images. Recently Eppenhof et al. [44] estimated the registration error by utilizing convolutional neural networks. Only preliminary results were available for synthetic 3D data.

We explore several features related to the uncertainty of the registration transformation as well as related to intensity. All features are calculated in physical units, i.e. mm, which makes the system independent of voxel size. Finally, features are combined by using regression forests. The proposed method is applied and evaluated on chest CT scans. This work is an extension of [55] with updated methodology and substantially extended evaluation.

4.2 Methods

4.2.1 System overview

A block diagram of the proposed algorithm is shown in Fig. 4.1. The system has two inputs: a fixed image I_F and a moving image I_M. Several registration-based and intensity-based features are generated. A regression forests (RF) is then trained from all features to estimate the registration error.

The proposed system is trained to predict residual distances y (registration errors) obtained from a set of semi-automatically established corresponding landmarks. During evaluation, the prediction result $\hat{y}$ is compared with errors obtained from an independent set of ground truth landmarks, using cross-validation. The proposed system therefore estimates registration errors in physical units, i.e. mm. More information about the ground truth is available in Section 4.3.1. Details of the features are elaborated in Section 4.2.3.

4.2.2 Registration

Registration can be formulated as an optimization problem in which the cost function $\mathscr{C}$ is minimized with respect to $\boldsymbol{T}$:

$$\widehat{\boldsymbol{T}} = \arg\min_{\boldsymbol{T}} \mathscr{C}\left(\boldsymbol{T}; I_F, I_M\right), \tag{4.1}$$

where $\boldsymbol{T}$ denotes the transformation. The optimization is usually solved by an iterative method embedded in a multi-resolution setting. A registration can be initialized by an initial transform $\boldsymbol{T}^{\text{ini}}$.

4.2.3 Features and pooling

The features we used in our system, consist of several registration-based as well as intensity-based features. Some features are intrinsically capable to be calculated over differently sized local boxes, for others, a pool of features is created by computing local averages and maxima afterwards. The features used in this paper are listed in Table 4.1. We propose the following features:

4.2.3.1 Registration-based features

Variation of deformation vector field ($\operatorname{std}\boldsymbol{T}$**):** The final solution of an iterative optimization problem can be influenced by the initial parameters. If in a region the cost function has multiple local minima or is semi-flat, a slight change in the initial parameters can lead to a different solution. In contrast, in areas where the cost function is well-defined, variations in the initial state are expected to have much less effect on the final solution. A flow chart of the described feature is available in Fig. 4.2. Given P random initial transformations $\boldsymbol{T}_i^{\text{ini}}$, $i \in \{1,\ldots,P\}$, that are used as initializations of the registration algorithm from Eq. (4.1), the variation in the final transformation results $\widehat{\boldsymbol{T}}_i$ is a surrogate for the precision of the registration. We propose to use the standard deviation $\operatorname{std}\boldsymbol{T}$ of those final transformations as a feature:

$$\overline{\boldsymbol{T}} = \tfrac{1}{P}\sum \widehat{\boldsymbol{T}}_i, \tag{4.2}$$

$$\operatorname{std}\boldsymbol{T} = \sqrt{\tfrac{1}{P-1}\sum \left\|\widehat{\boldsymbol{T}}_i - \overline{\boldsymbol{T}}\right\|^2}. \tag{4.3}$$

In this work, the initial transformations $\boldsymbol{T}_i^{\text{ini}}$ are created by uniformly distributed offsets in the range $[-2,2]$ mm to all B-spline coefficients. The offset range is chosen to be relatively small in comparison to the B-spline grid spacing in order to avoid unrealistic deformation. An example of $\operatorname{std}\boldsymbol{T}$ in a synthetically deformed image is given in Fig. 4.3a.

Instead of perturbing the initial state of the registration, it is also possible to first perform the registration without any manipulated initial state, resulting in a transformation $\boldsymbol{T}^{\text{b}}$ [97]. Then, random offsets $\boldsymbol{T}_i^{\text{offset}}$ are added to $\boldsymbol{T}^{\text{b}}$ after which

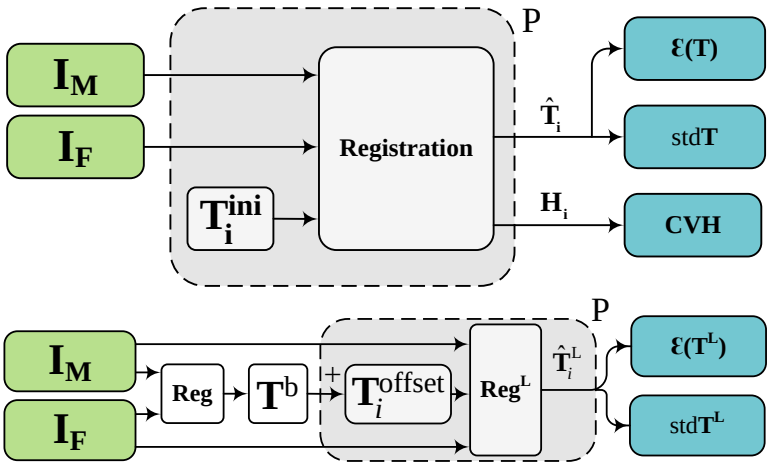
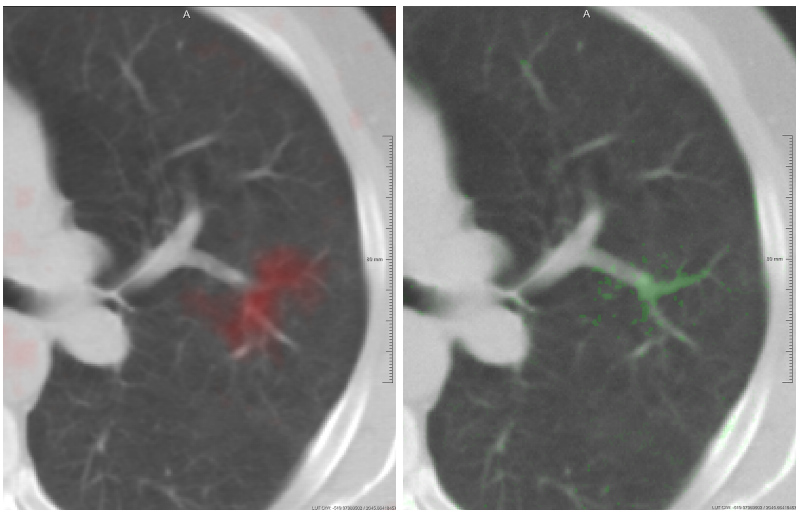


Figure 4.2: Multiple registrations are performed to create registration-based features. Either the initial transformation is varied, or the transformation after the base registration.



(a) std $\boldsymbol{T}$ (b) CVH

Figure 4.3: Visualization of std $\boldsymbol{T}$ and CVH in a synthetically deformed image. The deformed image is created by a random deformation vector field which is smoothed by a Gaussian kernel similar to [17].

another registration is performed, resulting in $\widehat{\boldsymbol{T}_i^L}$. This is close to the work of Hub et al. [95], and approximately measures the concavity of the cost function. The feature

$\text{std}\,\boldsymbol{T}^{\text{L}}$ is then derived akin to Eq. (4.3):

$$\overline{\boldsymbol{T}^{\text{L}}} = \tfrac{1}{P}\sum \widehat{\boldsymbol{T}_i^{\text{L}}}, \tag{4.4}$$

$$\text{std}\,\boldsymbol{T}^{\text{L}} = \sqrt{\tfrac{1}{P-1}\sum \| \widehat{\boldsymbol{T}_i^{\text{L}}} - \overline{\boldsymbol{T}^{\text{L}}} \|^2}. \tag{4.5}$$

It is expected that $\text{std}\,\boldsymbol{T}^{\text{L}}$ is small in regions where the cost function is concave, as by adding small offsets $\boldsymbol{T}_i^{\text{offset}}$ to the parameters, it can still move back to the previous optimal point. A flow chart of $\text{std}\,\boldsymbol{T}^{\text{L}}$ is shown in Fig 4.2. $\text{std}\,\boldsymbol{T}^{\text{L}}$ is calculated using the same setting as $\text{std}\,\boldsymbol{T}$, except that only one resolution is used.

If the difference between $\overline{\boldsymbol{T}}$ and $\boldsymbol{T}^{\text{b}}$ is relatively large, regions indicating a small $\text{std}\,\boldsymbol{T}$ are still potentially regions of low registration quality. We then consider the bias $\mathscr{E}(\boldsymbol{T})$ and $\mathscr{E}(\boldsymbol{T}^{\text{L}})$ as complementary features to $\text{std}\,\boldsymbol{T}$ and $\text{std}\,\boldsymbol{T}^{\text{L}}$ computed by:

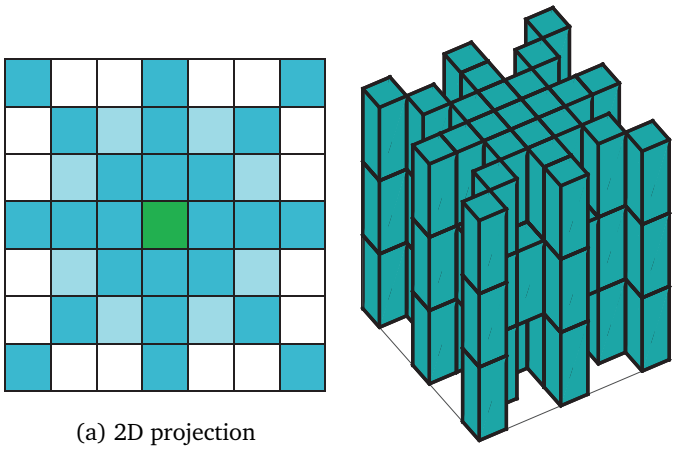
$$\begin{aligned} \mathscr{E}(\boldsymbol{T}) &= \| \boldsymbol{T}^{\text{b}} - \overline{\boldsymbol{T}} \|, \\ \mathscr{E}(\boldsymbol{T}^{\text{L}}) &= \| \boldsymbol{T}^{\text{b}} - \overline{\boldsymbol{T}^{\text{L}}} \|. \end{aligned} \tag{4.6}$$

Coefficient of variation of joint histograms (CVH): Multiple registration results can be used to extract additional information from the matched intensity patterns of the images. Given a fixed image I_F and a registration sub-result $I_M(\boldsymbol{T}_i)$, we calculate their joint histogram $\text{H}_i, \forall i$. For identical sub-registrations, all resulting joint histograms are equal. Variation in the joint histograms implies registration uncertainty as a surrogate for registration error. The coefficient of variation of the joint histograms is calculated by dividing the standard deviation of all joint histograms over the average, $\overline{\text{H}}$, of them. This normalization is done to compensate for large differences between the elements of $\overline{\text{H}}$. We obtain the CVH in histogram space as follows:

$$\text{CVH}^{\text{B}\times\text{B}} = \frac{\text{std}\,\text{H}}{\overline{\text{H}} + \epsilon}, \tag{4.7}$$

where B is the number of histogram bins, and ϵ a constant to avoid division by zero. In the experiments we set ϵ to 5. The $\text{CVH}^{\text{B}\times\text{B}}$ in histogram space is subsequently transferred to the spatial domain, by assigning voxels x with a particular intensity combination $\left(I_F(\boldsymbol{x}), I_M(\boldsymbol{T}^{\text{b}}(\boldsymbol{x}))\right)$ the corresponding value from $\text{CVH}^{\text{B}\times\text{B}}$, resulting in the final CVH feature with size equal to the fixed image. Note that the CVH can be used in a multi-modality setting, like the previous features. An example of the CVH on a synthetically deformed image is given in Fig. 4.3b.

Determinant of the Jacobian (Jac): Jac measures the relative local volume change. This can point to poor registration quality in case of very large ($\text{Jac} \gg 1$) or very small ($\text{Jac} \ll 1$) values, or discontinuous transformations in case of a negative value ($\text{Jac} < 0$). In the experiments, the determinant of the Jacobian of $\boldsymbol{T}^{\text{b}}$ is used.



(a) 2D projection

(b) 3D view

Figure 4.4: MIND search region. (a) The green cell indicates the center and darker blue cells indicate more accumulated cells in the projection view.

4.2.3.2 Intensity-based features

MIND: The Modality Independent Neighborhood Descriptor (MIND) was introduced by Heinrich et al. [14] in order to register multi-modal images. In this local self-similarity metric, a patch is considered to compare intensities between fixed and moving images. Finally, the sum of absolute differences between the MIND vector of I_F and that of $I_M(\boldsymbol{T}^{\mathrm{b}})$ is computed. We calculate MIND with a sparse patch including 82 voxels inside a $[7 \times 7 \times 3]$ box, which is approximately physically isotropic for the data used in the experiments (see Fig. 4.4).

Local normalized mutual information: Mutual information is used as an entropy-based similarity measure of two images. Similar to [39] we use the following definitions for local normalized mutual information:

$$\begin{aligned} \mathrm{NMI} &= \frac{H(I_F) + H(I_M(\boldsymbol{T}^{\mathrm{b}}))}{H\Big(I_F, (I_M(\boldsymbol{T}^{\mathrm{b}})\Big)}, \\ \mathrm{PMI} &= \frac{MI\Big(I_F, I_M(\boldsymbol{T}^{\mathrm{b}})\Big)}{min\Big\{H(I_F), H(I_M(\boldsymbol{T}^{\mathrm{b}}))\Big\}}. \end{aligned} \tag{4.8}$$

Both metrics are calculated over 8 differently sized boxes: [5, 10, 15, 20, 25, 30, 35, 40] mm. Two strategies for the selection of the number of bins are used, one uses a constant value $\mathrm{B_C}$, the other strategy depends on the number of samples $|\mathrm{B}| = log_2(n) + 1$, in which n is the number of samples in each box. The notations NMIS and PMIS indicate mutual information calculated with the latter strategy.

Table 4.1: An overview of the proposed features. Averages and maxima are taken over boxes of diameter [2, 5, 10, 15, 20, 25, 30, 35, 40] mm for the features: MIND, std $\boldsymbol{T}$, std $\boldsymbol{T}^{\mathrm{L}}$, CVH, $\mathscr{E}(\boldsymbol{T})$, $\mathscr{E}(\boldsymbol{T}^{\mathrm{L}})$ and Jac. Mutual information measures are calculated in boxes of [5, 10, 15, 20, 25, 30, 35, 40] mm. SID and GID are computed using Gaussian derivatives with standard deviations in the range [0.5, 1, 2, 4, 8, 16] mm.

Feature	N_f	
MIND	18	9 average boxes + 9 maxima boxes
MI	32	NMI, NMIS, PMI, PMIS calculated over 8 boxes
std $\boldsymbol{T}$	18	9 average boxes + 9 maxima boxes
std $\boldsymbol{T}^{\mathrm{L}}$	18	9 average boxes + 9 maxima boxes
CVH	18	9 average boxes + 9 maxima boxes
$\mathscr{E}(\boldsymbol{T})$	18	9 average boxes + 9 maxima boxes
$\mathscr{E}(\boldsymbol{T}^{\mathrm{L}})$	18	9 average boxes + 9 maxima boxes
Jac	18	9 average boxes + 9 maxima boxes
NC	8	calculated over 8 boxes
SID&GID	12	calculated over 6 sigma's

Modality-dependent features: In addition to the modality-independent features from above, we consider the use of several modality-dependent features. In the experiments we assess their contributed value. Similar to [39] the squared intensity difference (SID) and the gradient of intensity difference (GID) are computed using Gaussian (derivative) operators with standard deviations of [0.5, 1, 2, 4, 8, 16] mm. Normalized correlation (NC) is calculated within boxes of size [5, 10, 15, 20, 25, 30, 35, 40] mm akin to [39].

4.2.3.3 Pooling

In order to reduce discontinuities and improve interaction with other features, the total set of features is increased by generating a pool from those mother features by calculating averages and maxima over them using differently sized boxes. The features MI, SID, GID and NC are inherently computed over differently sized local regions. The features MIND, std $\boldsymbol{T}$, std $\boldsymbol{T}^{\mathrm{L}}$, CVH, $\mathscr{E}(\boldsymbol{T})$, $\mathscr{E}(\boldsymbol{T}^{\mathrm{L}})$ and Jac are calculated in a voxel-based fashion, and then pooled afterwards. Average and maximum pooling is performed with box sizes of $[2,5,10,15,20,25,30,35,40]$ mm. As a result, for each feature we obtain a pool of 18 features: 9 from box averages and 9 from box maxima. The average-pooling is done efficiently by the help of integral images introduced by Viola et al. [98]. A list of the proposed mother features together with the number of derived features N_f are given in Table 4.1.

4.2.4 Regression forests

Random forests were introduced by Breiman [99] by extending the idea of bagging. The forests consist of several weak learners (trees) which are combined in an efficient

fashion. Each tree is started from a node and continues splitting until reaching certain criteria. In contrast to bagging, splitting is performed with a random subset of features which makes the training phase faster and reduces correlation between trees, consequently decreasing the forest error rate. The reason that we chose the random forest is that it can handle data without preprocessing. For instance rescaling of data, outlier removal and selection of features are not necessary in random forests. In addition, random forest are efficient to train and fast at runtime.

Random forests have the capability to calculate the importance of each feature with a little additional computation, which shows the contribution of each feature to the forest. Training of each tree is based on a bootstrap of all samples, and the so-called out-of-bootstrap samples Ω are used to compute the importance of a feature x_i. Importance is then defined as the difference between the mean square error (MSE) before and after a permutation of this feature:

$$\mathrm{Imp}(x_i) = \frac{1}{N_t}\sum_{t=1}^{N_t}\Big(\underset{j\in\Omega}{\mathrm{MSE}}\big(\hat{y}_{\pi_i j}, y_j\big) - \underset{j\in\Omega}{\mathrm{MSE}}\big(\hat{y}_j, y_j\big)\Big), \tag{4.9}$$

where y_j is the real value, $\hat{y}_j$ the predicted value from the regression, $\hat{y}_{\pi_i j}$ the predicted value when permuting feature i, and N_t the number of trees.

In this work, random forests are trained with different combinations of the proposed features (see Table 4.1). The dependent variable y is the registration error in mm, which is described in Section 4.3.1.

4.3 Experiments and results

4.3.1 Materials and ground truth

The SPREAD [47] DIR-Lab-4DCT [74] and DIR-Lab-COPDgene [75] databases have been used in this study. In the SPREAD study, there are 21 pairs of 3D follow-up lung CT images. Each patient in this database has a baseline and a follow-up image (which is taken after 30 months) both in inhale phase. The age of the patients ranges from 49 to 78 years old. The average size of the images is $446 \times 315 \times 129$ with an average voxel size of $0.78 \times 0.78 \times 2.50$ mm. In each pair of images, about 100 well-distributed corresponding landmarks were previously selected [73] semi-automatically on distinctive locations [48].

From the DIR-Lab-4DCT data, five cases (4DCT1 to 4DCT5) are selected with each five phases between maximum inhalation and exhalation. The average image size is $256 \times 256 \times 103$ with an average voxel size of $1.10 \times 1.10 \times 2.50$ mm. Each scan has 75 corresponding landmarks annotated. Ten cases with severe breathing disorders are available via the DIR-Lab-COPDgene database. The images are taken in inhale and exhale phases. In total, 300 landmarks are annotated. The average image size and the average voxel size are $512 \times 512 \times 120$ and $0.64 \times 0.64 \times 2.50$ mm, respectively.

Accuracy of the registration can be defined as the residual Euclidean distance after registration between the corresponding landmarks:

$$y = \| \boldsymbol{T}^{\mathrm{b}}(\boldsymbol{x}_F) - \boldsymbol{x}_M \|_2, \tag{4.10}$$

with $\boldsymbol{x}_F$ and $\boldsymbol{x}_M$ the corresponding landmark locations. Based on the idea that the registration error is smooth, we include voxels from a small local neighborhood around the landmarks to increase the total set of available landmarks. In this small neighborhood we assume that the registration error is equal to the error at the center of the neighborhood. This assumption seems reasonable for smooth transformations and within a small region. The neighborhood size is chosen as $10 \times 10 \times 7.5$ mm, which is approximately equivalent to the final grid spacing of the B-spline registration (see Fig. 4.5).

The core software is written in Python. The feature pooling is performed with a C++ program [100] and the regression forest is calculated with the help of the Scikit-learn package [101]. All registrations are performed by `elastix` [52]. Detailed registration setting can be found in the `elastix` parameter file database (elastix.isi.uu.nl, entry par0049). The code is publicly available via github.com/hsokooti/regun.

4.3.2 Evaluation measures

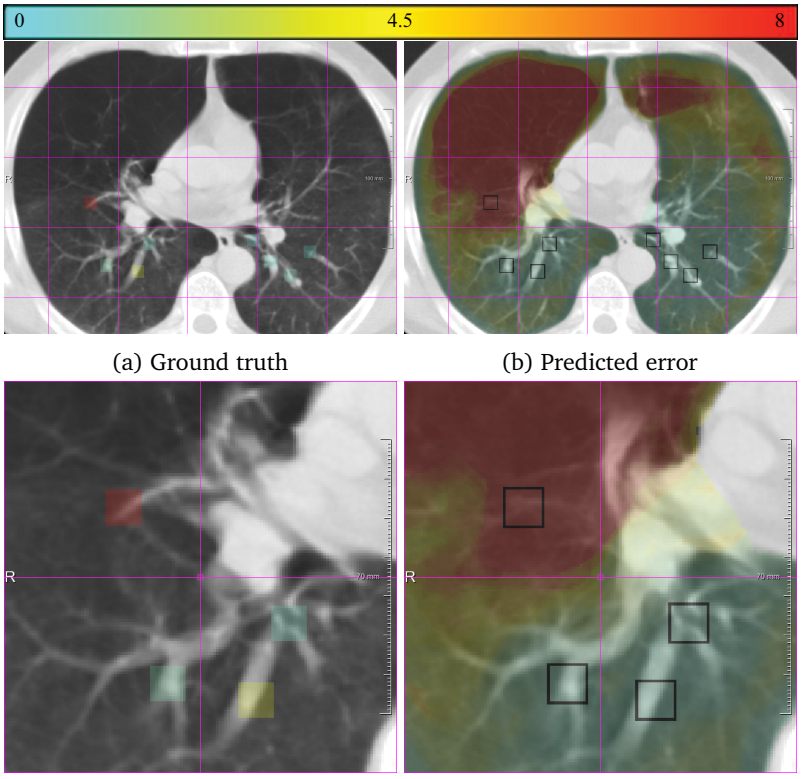
In the SPREAD database, we employ 10 cross-validations by randomly splitting the data in 15 image pairs for training and the remaining 6 pairs for testing. To evaluate the regression performance, the mean absolute error (MAE) of the real registration error y_i and the estimated one $\hat{y}_i$ is calculated over the neighborhood of the landmarks by:

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|. \tag{4.11}$$

To further detail the regression performance, the MAE is subdivided into three categories: $\mathrm{MAE_c}$, $\mathrm{MAE_p}$ and $\mathrm{MAE_w}$ with y in $[0,3)$, $[3,6)$ and $[6,\infty)$ mm, corresponding to correct, poor and wrong registration, similar to Muenzing et al. [39]. We then do the same for $\hat{y}_i$, and report the accuracy and F1 score for classifying the registration error in these three categories.

4.3.3 Parameter selection

The RF is trained using 100 trees with a maximum tree depth of 9, while at least 5 samples remain in the leaf nodes. At each splitting node, m features are randomly selected. We set m to the square root of the total number of features in that experiment, which performed slightly better than $m =$ (number of features)$/3$ [102]. The total number of registrations P is chosen as 20 to ensure that the estimation of std $\boldsymbol{T}$ does not change considerably when increasing the number of registrations [55].



(a) Ground truth (b) Predicted error

(c) Magnification of (a) (d) Magnification of (b)

Figure 4.5: Example data from the SPREAD dataset. The left column (a,c) shows the fixed image with the ground truth registration error overlaid in color. The square boxes around each landmark are given the same error as the error at the landmark. The right column (b,d) shows the moving image after registration with the registration error predicted by the proposed method overlaid in color. (c) and (d) are zoomed in versions of (a) and (b).

4.3.4 Reference registration error set

For the SPREAD and the DIR-Lab-4DCT study, registrations are based on free-form deformations by B-splines [6]. The cost function is mutual information, which is optimized by adaptive stochastic gradient descent. We used three resolutions with a final B-spline grid spacing of $[10, 10, 10]$ mm. We collect samples by performing four different registrations using 20, 100, 500 and 2000 iterations, respectively. All other registration settings remain the same in these registrations. By varying the number of iterations we increase the variation in the samples, as well as the training size. Table 4.2 gives the distribution of reference registration errors in each database. As expected, increasing the number of iterations shifts the distribution towards the

Table 4.2: Distribution of the reference registration errors in each database, used during testing.

Database-iters	correct		poor		wrong		total
SPREAD 20	848789	(84.1%)	102837	(10.2%)	58059	(5.8%)	1009685
SPREAD 100	904796	(89.6%)	66467	(6.6%)	38422	(3.8%)	1009685
SPREAD 500	925840	(91.7%)	51910	(5.1%)	31935	(3.2%)	1009685
SPREAD 2000	935676	(92.7%)	46170	(4.6%)	27839	(2.8%)	1009685
SPREAD together	3615101	(89.5%)	267384	(6.6%)	156255	(3.9%)	4038740
DIR-Lab-4DCT 20	521481	(84.5%)	71282	(11.5%)	24543	(4.0%)	617306
DIR-Lab-4DCT 100	540989	(87.6%)	61131	(9.9%)	15186	(2.5%)	617306
DIR-Lab-4DCT 500	553757	(89.7%)	53067	(8.6%)	10482	(1.7%)	617306
DIR-Lab-4DCT 2000	561909	(91.0%)	46679	(7.6%)	8718	(1.4%)	617306
DIR-Lab-4DCT together	2178136	(88.2%)	232159	(9.4%)	58929	(2.4%)	2469224
DIR-Lab-COPD ANTsBSplineSyN	2643	(88.1%)	184	(6.1%)	173	(5.8%)	3000
DIR-Lab-COPD `elastix-advanced`	2420	(80.7%)	259	(8.6%)	321	(10.7%)	3000

Table 4.3: Distribution of the reference registration errors, used during training.

Database	correct		poor		wrong		total
SPREAD together	589854	(58.0%)	270523	(26.6%)	156881	(15.4%)	1017258
DIR-Lab-4DCT together	328055	(53.0%)	232499	(37.5%)	58929	(9.5%)	619483

“correct" registration category. The maximum registration error is 81.8 mm in the SPREAD database, 17.6 mm in the DIR-Lab-4DCT database.

Since the a priori distribution of registration errors is imbalanced, with much more samples in the “correct" category, we perform the following balancing step during training. For landmarks that fall in the category “correct", we only add samples from a smaller neighborhood of $5 \times 5 \times 2.5$ mm instead of the $10 \times 10 \times 7.5$ mm neighborhoods used for landmarks in the categories “poor" and “wrong". The distribution of reference registration errors of the training samples is shown in Table 4.3.

For the DIR-Lab-COPDgene study, more advanced settings of the registration are used. In this experiment, samples are taken only on the landmark locations. More details are given in Section 4.3.5.8. The maximum registration error in this data is 31.5 mm.

4.3.5 Experiments

4.3.5.1 Single feature performance in SPREAD

The proposed features are described in Section 4.2.3 and summarized in Table 4.1. To investigate the strength of the individual features, we trained the random forest with only a single feature with pooling. By comparing the MAE results in Table 4.4, it can be seen that MIND, std $\boldsymbol{T}^{\mathrm{L}}$ and SID&GID are the best single features in the categories

Table 4.4: Regression results for single features on the SPREAD database. The columns indicate the number of features (N_f), the mean absolute error (MAE), the accuracy (Acc) and the F1 score. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes, respectively.

	N_f	MAE	MAE_c	MAE_p	MAE_w	Acc	F1_c	F1_p	F1_w
MIND	18	1.10 ± 1.97	0.76 ± 0.72	1.59 ± 1.39	6.50 ± 5.88	89.8	94.9	34.1	83.0
MI	32	1.20 ± 1.88	0.89 ± 0.71	1.53 ± 1.14	6.30 ± 5.58	87.9	93.9	30.1	79.9
std $\boldsymbol{T}$	18	1.59 ± 2.79	1.15 ± 1.78	2.98 ± 4.06	7.60 ± 6.12	85.5	92.7	22.4	64.4
std $\boldsymbol{T}^\text{L}$	18	1.51 ± 2.40	1.11 ± 1.34	2.49 ± 3.05	7.32 ± 5.79	86.7	93.4	18.3	70.7
CVH	18	1.93 ± 3.29	1.49 ± 2.22	1.82 ± 2.00	9.80 ± 7.19	75.2	87.2	16.9	37.0
$\mathscr{E}(\boldsymbol{T})$	18	2.00 ± 2.80	1.61 ± 1.76	2.18 ± 3.12	8.52 ± 6.48	69.8	82.8	17.0	43.5
$\mathscr{E}(\boldsymbol{T}^\text{L})$	18	1.68 ± 2.85	1.19 ± 1.71	3.19 ± 3.28	8.34 ± 6.74	84.4	92.6	11.7	54.8
Jac	18	2.15 ± 3.15	1.72 ± 1.90	1.91 ± 2.27	10.03 ± 6.97	68.2	83.7	13.0	31.4
NC	8	1.38 ± 2.89	0.90 ± 0.71	1.70 ± 1.68	9.41 ± 9.15	88.2	94.3	28.5	77.0
SID&GID	12	1.30 ± 2.12	0.94 ± 0.90	1.82 ± 1.63	6.95 ± 6.02	89.9	95.1	24.9	74.3

Intensity, Registration and Modality-dependent, respectively.

4.3.5.2 Combined features performance

Instead of using only a single feature, several combinations of features are used to build the RFs:

- **Intensity:** Combination of all modality-independent intensity features: MIND and MI (50 features).
- **Registration:** Combination of all registration features: std $\boldsymbol{T}$, std $\boldsymbol{T}^\text{L}$, CVH, $\mathscr{E}(\boldsymbol{T})$, $\mathscr{E}(\boldsymbol{T}^\text{L})$ and Jac (108 features).
- **Combined:** Combination of both intensity and registration features (158 features).

All results are available in Table 4.5. By combining features from both the registration and modality-independent intensity category, improvements were obtained in all evaluation measures.

The result of the regression with combined features is detailed in Fig. 4.6(a), which shows the real error (solid blue line) against the predicted error, sorted from small to large. In Fig. 4.6(b) we grouped the real errors in the three categories, each category showing a box-plot of the predicted errors. Intuitively, a smaller overlap between the boxes represents a better regression.

4.3.5.3 Including modality-dependent features

We consider adding the combination of three modality-dependent features to the combined feature set (Combined+MD): NC, SID and GID. In both databases, if we add the modality-dependent features (see Table 4.5), negligible differences are observed. Therefore, to keep the feature set small and modality-independent, we select the

Table 4.5: Regression results for groups of features on the SPREAD database. The columns indicate the number of features (N_f), the mean absolute error (MAE), the accuracy (Acc) and the F1 score. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes, respectively. MD, NN and LR stands for modality dependent, neural networks and linear regression, respectively.

	N_f	MAE	MAE_c	MAE_p	MAE_w	Acc	F1_c	F1_p	F1_w
Intensity	50	1.09 ± 1.88	0.77 ± 0.68	1.49 ± 1.26	6.20 ± 5.68	90.3	95.1	35.7	83.6
Registration	108	1.32 ± 2.35	0.90 ± 1.04	2.10 ± 2.71	7.76 ± 6.01	90.0	95.1	31.5	78.4
Combined	158	1.07 ± 1.86	0.76 ± 0.65	1.47 ± 1.22	6.12 ± 5.64	90.7	95.4	38.1	84.4
Combined-no pooling	8	1.24 ± 2.22	0.85 ± 0.73	1.72 ± 1.64	7.39 ± 6.62	89.4	94.8	32.6	79.1
Combined+MD	178	1.07 ± 1.83	0.76 ± 0.65	1.46 ± 1.20	5.95 ± 5.59	90.7	95.4	38.3	84.5
Combined (LR)	158	1.86 ± 2.03	1.58 ± 1.34	2.47 ± 2.21	6.12 ± 4.97	77.3	87.3	17.0	67.6
Combined (NN)	158	1.13 ± 2.07	0.74 ± 0.70	1.81 ± 1.67	7.08 ± 5.88	89.8	95.0	31.2	79.6

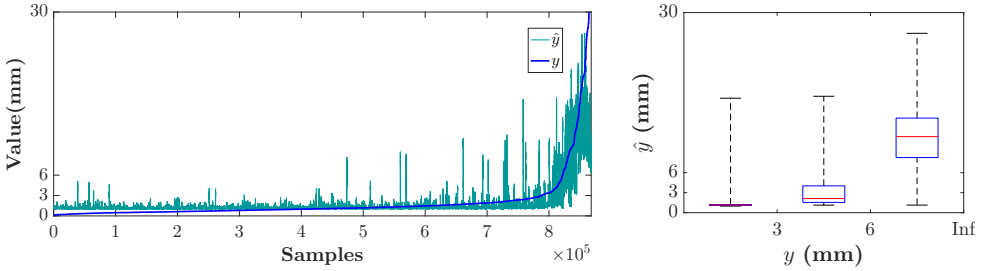


Figure 4.6: Real (y) vs predicted registration error ($\hat{y}$) for Combined features in the SPREAD database. (a) The real error (solid blue line y) against the predicted error ($\hat{y}$), sorted from small to large. In (b) we grouped the real errors in the three categories, each category showing a box-plot of the predicted errors.

"combined features" class without the modality-dependent features as the final system in the remainder of this paper.

4.3.5.4 The effect of pooling

To examine the effect of pooling, we perform an experiment without pooling on the combined feature set. We only calculate PMIS within a box size of 15 mm in this experiment. From Table 4.5 the benefit of pooling can be observed.

4.3.5.5 Alternative regression methods

In this section, we compare RF regression with linear regression (LR) and neural networks (NN). Feature normalization is done for both regressors. We utilized neural networks with three hidden layers of 1024, 512 and 256 units each. ReLU is used as an activation function and Huber is utilized as a loss function. Table 4.5 gives the results of these experiments. The performance of neural networks is on par with random forests. However, the results of linear regression are not comparable to that of random forests, both in MAE and accuracy.

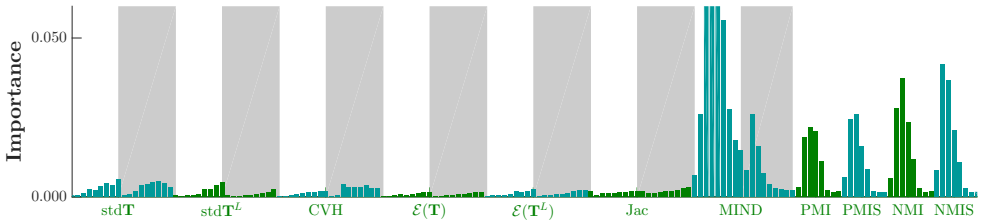


Figure 4.7: Feature importance of the SPREAD combined experiment. White areas correspond to box averages, while shaded areas correspond to box maxima.

Table 4.6: Leave one feature out results of SPREAD data. The columns indicate the number of features (N_f), the mean absolute error (MAE), the accuracy (Acc) and the F1 score. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes, respectively.

	N_f	MAE	MAE_c	MAE_p	MAE_w	Acc	F1_c	F1_p	F1_w
Combined	158	1.07 ± 1.86	0.76 ± 0.65	1.47 ± 1.22	6.12 ± 5.64	90.7	95.4	38.1	84.4
−MIND	140	1.18 ± 1.96	0.83 ± 0.66	1.56 ± 1.50	6.70 ± 5.69	90.2	95.1	36.2	83.0
−MI	126	1.10 ± 1.98	0.75 ± 0.67	1.54 ± 1.30	6.66 ± 5.84	90.6	95.3	37.0	84.2
− std $\boldsymbol{T}$	140	1.08 ± 1.86	0.76 ± 0.65	1.46 ± 1.18	6.14 ± 5.65	90.7	95.3	38.1	84.3
− std $\boldsymbol{T}^\text{L}$	140	1.08 ± 1.89	0.76 ± 0.65	1.46 ± 1.22	6.21 ± 5.73	90.6	95.3	38.3	83.7
−CVH	140	1.07 ± 1.81	0.75 ± 0.65	1.46 ± 1.21	6.06 ± 5.98	90.7	95.4	38.4	84.3
$-\mathscr{E}(\boldsymbol{T})$	140	1.07 ± 1.86	0.76 ± 0.65	1.46 ± 1.21	6.13 ± 5.64	90.7	95.4	38.2	84.5
$-\mathscr{E}(\boldsymbol{T}^\text{L})$	140	1.08 ± 1.85	0.76 ± 0.65	1.47 ± 1.22	6.12 ± 5.61	90.6	95.3	37.5	84.3
−Jac	140	1.08 ± 1.87	0.76 ± 0.65	1.49 ± 1.31	6.06 ± 5.72	90.7	95.4	37.9	84.8

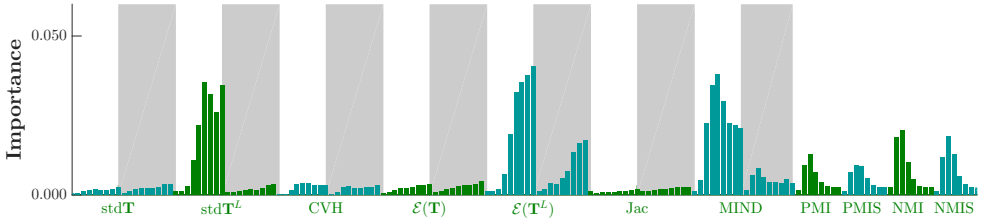
4.3.5.6 Feature importance

The feature importance, see Eq. (4.9), is displayed in Fig. 4.7. It shows that MIND and MI are the features contributing most to the RF performance, followed by std $\boldsymbol{T}$, std $\boldsymbol{T}^\text{L}$ and CVH.

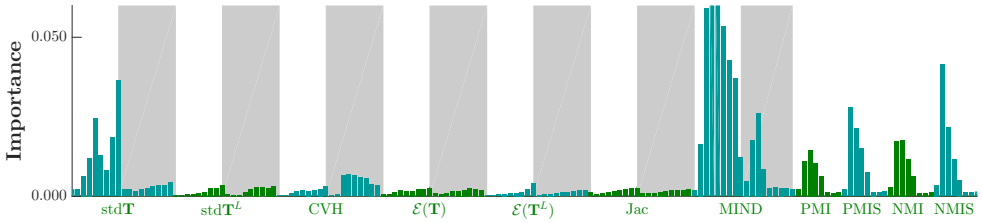
The feature importance using a different number of iterations is shown in Fig. 4.8. The contribution of all intensity features stay the same in all experiments, while some of the registration features contribute differently with respect to the number of iterations. For instance, the importance of std $\boldsymbol{T}$ and CVH increase with increasing the number of iterations. The features std $\boldsymbol{T}^\text{L}$ and $\mathscr{E}(\boldsymbol{T}^\text{L})$ play important roles when the number of iterations is not enough for registration convergence.

4.3.5.7 Excluding a single feature

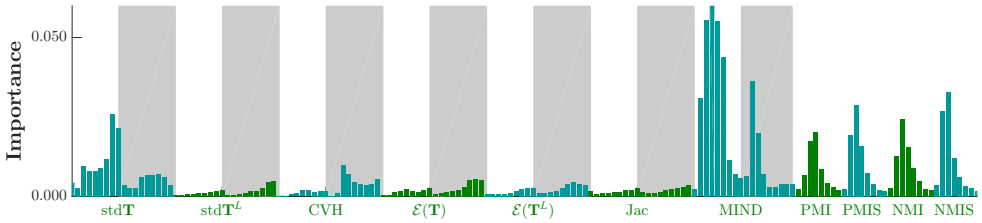
To further investigate the importance of the several features, we additionally perform an experiment where we leave one feature out of the combined feature set. The results are reported in Table 4.6. In these experiments, feature redundancy can be found. For instance, MI has a large importance values in random forests, but if we leave that feature out, other features can compensate for that.



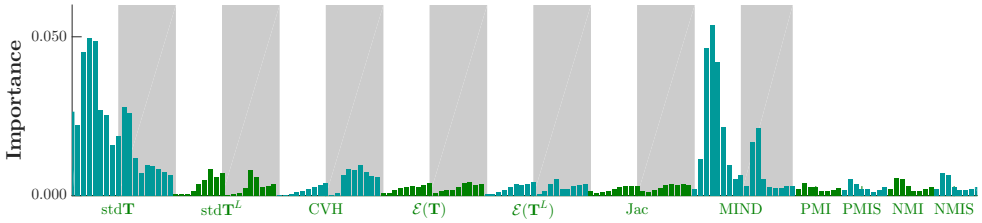
(a) SPREAD 20 iterations



(b) SPREAD 100 iterations



(c) SPREAD 500 iterations



(d) SPREAD 2000 iterations

Figure 4.8: Feature importance of the SPREAD combined experiment with different iterations. The contribution of all intensity features stay the same in all experiments, while some of the registration features contribute differently with respect to the number of iterations. White areas correspond to box averages, while shaded areas correspond to box maxima.

4.3.5.8 Inter-database validation

To study the generalizability of the proposed system, instead of cross-validation on a single database, we perform training on the DIR-Lab-4DCT database and test it on the

SPREAD database. As mentioned before, the SPREAD database consists of only inhale images but the DIR-Lab-4DCT database has images from inhale to exhale phases. Therefore, this makes the DIR-Lab-4DCT more suitable for training. The result of this experiment is available in Table 4.7. Once more, we can draw the conclusion that by combining both intensity and registration-based features, the regression performance can be improved. In contrast to the SPREAD experiment, this time it is observed that the registration features perform better than the intensity features.

To further evaluate the generalizability of the proposed method, we test it for different registration methods on a third independent test set, the DIR-Lab-COPDgene dataset. The regression forest is trained on a combination of the SPREAD and DIR-Lab-4DCT data. We evaluate two registration algorithms that achieved excellent performance in the EMPIRE10 challenge [80], i.e. the ANTs registration package [103, 104] and `elastix` with advanced settings [105].

Prior to deformable registration we perform an affine registration using 5 resolutions and utilizing torso masks. For the deformable registration we use settings similar to the ones used in the EMPIRE10 challenge, specifically:

ANTs-BSplineSyN: With respect to the EMPIRE10 challenge we increased the number of iterations to 1000 for each of the 4 resolutions, using a 10% sampling rate. This improved the performance on our data and considerably reduced the calculation time. As suggested in [104], several preprocessing steps are used, including masking out the lungs, and inverting the image intensities and rescaling them between 0 and 1. Further settings include: registration model: symmetric diffeomorphic; dissimilarity metric: local cross correlation; number of resolutions: 4; maximum number of iterations: 1000; sampling: 10% random samples; convergence threshold: 1e-6. The average TRE on DIR-Lab-COPDgene is 1.90 ± 2.86 mm.

`elastix-advanced`: Settings are adopted from [105]. The most important ones are: registration model: B-spline; dissimilarity metric: normalized correlation; number of resolutions: 6; number of iterations: 1000; sampling: 2000 random samples; B-spline grid spacing: [5, 5, 5] mm. The average TRE with this setting is 3.39 ± 4.30 mm on the DIR-Lab-COPDgene dataset.

Detailed parameter files for both registration methods are available via elastix.isi.uu.nl (entry par0049) and github.com/hsokooti/regun. The calculation time of ANTs was about 60 hours per registration, comparing to 12 minutes for `elastix`.

In this experiment, the evaluation is performed only on the landmarks locations, where Table 4.2 displays the distribution of reference registration errors during testing. The results of the experiments are given in Table 4.8. A scatter plot is also depicted in Fig. 4.9. Similar to the previous inter-database experiment (Table 4.7), the MAE and accuracy of the registration features are slightly better than the MAE and accuracy of the intensity-based features. However, intensity features obtained better classification

Table 4.7: Regression results for the SPREAD data trained on the DIR-Lab-4DCT data with `elastix` using 20, 100, 500 and 2000 iterations. The columns indicate the number of features (N_f), the mean absolute error (MAE), the accuracy (Acc) and the F1 score. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes, respectively.

	N_f	MAE	MAE_c	MAE_p	MAE_w	Acc	$F1_c$	$F1_p$	$F1_w$
Intensity	50	1.90 ± 3.63	1.56 ± 1.49	1.26 ± 1.01	10.83 ± 14.32	71.0	82.8	21.7	48.0
Registration	108	1.62 ± 3.59	1.23 ± 0.88	1.13 ± 0.81	11.53 ± 14.60	77.1	87.4	27.7	53.9
Combined	158	1.73 ± 3.56	1.36 ± 0.97	1.14 ± 0.83	11.30 ± 14.49	77.2	87.2	26.0	59.9

Table 4.8: Regression results for the DIR-Lab-COPDgene data with `elastix-advanced` and ANTs-BSplineSyN registrations trained on the SPREAD and DIR-Lab-4DCT data. The columns indicate the number of features (N_f), the mean absolute error (MAE), the accuracy (Acc) and the F1 score. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes, respectively.

	N_f	MAE	MAE_c	MAE_p	MAE_w	Acc	$F1_c$	$F1_p$	$F1_w$
`elastix-advanced`									
Intensity	50	2.17 ± 2.34	1.69 ± 1.35	2.81 ± 2.66	5.15 ± 4.44	64.2	77.9	20.8	64.6
Registration	108	1.84 ± 2.50	1.31 ± 1.66	2.22 ± 2.12	5.36 ± 4.29	76.0	87.6	29.9	57.6
Combined	158	1.86 ± 2.05	1.50 ± 1.16	1.92 ± 1.80	4.48 ± 4.21	75.3	86.9	29.5	66.1
ANTs-BSplineSyN									
Intensity	50	2.03 ± 2.01	1.80 ± 1.25	2.20 ± 2.27	5.30 ± 5.38	57.3	71.6	14.2	62.7
Registration	108	1.71 ± 2.39	1.43 ± 1.98	2.56 ± 2.01	5.06 ± 4.67	72.8	85.5	17.3	38.5
Combined	158	1.73 ± 1.80	1.52 ± 1.23	2.22 ± 2.27	4.45 ± 4.40	76.5	87.3	20.4	59.7

score in the wrong category. We conclude that the proposed method indeed generalizes to different settings of the same method (`elastix-advanced`), as well as registration methods with quite a different underlying transformation model (ANTs-BSplineSyN, which uses a symmetric diffeomorphic model).

4.4 Discussion

A system for quantitative error prediction of medical image registration is proposed and it is quantitatively evaluated on multiple chest CT datasets.

4.4.1 Features

In the intra-database (SPREAD) validation, it is observed that the single MIND feature can perform almost as good as the overall combined system. By adding MI and registration features, the results slightly improved. Muenzing et al. [39] did not consider MIND in their feature set and found that the most important single features in their classification experiments are mutual information and Gaussian intensity, whereas, based on Table 4.4 these features are less important than MIND in our experiments. Furthermore, the calculation time of MI for the whole image is about 3 h, as opposed to the calculation time of MIND, which is about 8 min (~3 min MIND + ~5 min pooling). Although less accurate, it is possible to reduce the calculation

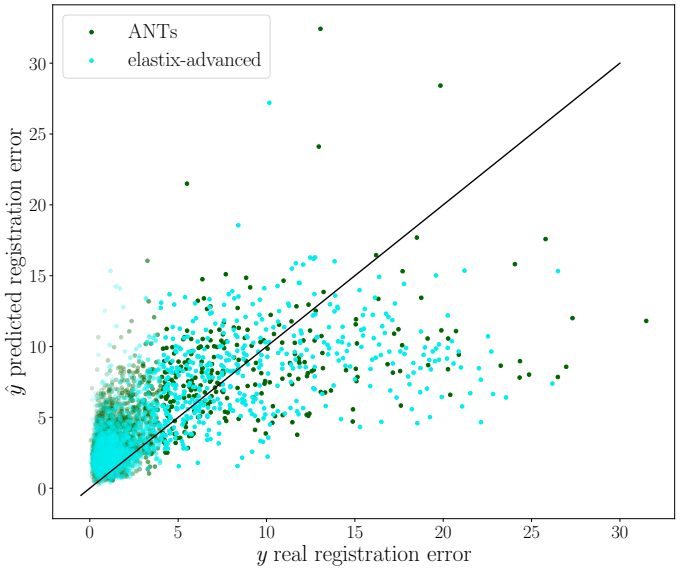


Figure 4.9: Scatter plot of real and predicted registration errors in the DIR-Lab-COPDgene database using `elastix-advanced` and ANTs-BSplineSyN registration. In total, 3000 landmarks are shown for each registration.

time of the MI feature by calculating it over a single window and then aggregate by pooling.

The modality-dependent intensity features do not increase regression accuracy on the data used in our paper. Consequently more generally applicable modality-independent features can be used, even for mono-modal problems.

Table 4.5, 4.7 and 4.8 together suggests that features in the intensity and registration categories provide complementary information, and that a better system can be obtained in terms of MAE and accuracy by considering both intensity and registration-derived features.

The intensity features were better predictors than the registration features in the intra-database experiment. However, in the inter-database experiment, the registration features outperform intensity features in terms of total accuracy and MAE. The same observation can be made for the average F1 score in the inter-database experiments using `elastix` (See Table 4.7, 4.8). For ANTs (Table 4.8), the average F1 score of the intensity-based features was slightly higher than that of the registration-based features.

The registration features contribute differently with respect to the number of iterations (See Fig. 4.8). The features std $\boldsymbol{T}^{\mathrm{L}}$ and $\mathscr{E}(\boldsymbol{T}^{\mathrm{L}})$ play important roles when the number of iterations is not enough for convergence. When the number of iterations

increases, the contribution of std $\boldsymbol{T}$ and CVH go up. In the work of Muenzing et al. [39], only one registration feature, Jac, was used and they reported that the impact of this feature is relatively low in comparison with intensity-based features. We observed the same result for Jac, but it should be pointed out that the range of Jac in our database was [0.3, 3.9] so voxels with negative or very large values were not encountered.

Feature pooling improves the regression results in all evaluation measures, due to the addition of contextual information. In some features like std $\boldsymbol{T}$, average pooling contributed more to the regression performance, while in features like CVH, maximum pooling had a higher importance value (See Fig. 4.8d).

As can be seen in Table 4.6, the proposed combined system has redundant features. Hence, by removing a single feature, the system is still able to predict the registration error with almost equal MAE as the total system. However, removing these features may decrease the generalizability of the system. For example, looking at the feature $\mathscr{E}(\boldsymbol{T}^{\mathrm{L}})$ in Fig. 4.7 we see that its contribution is relatively small overall. However, in Fig. 4.8 it can be seen that while it is less important for better registration results (100, 500 and 2000 iterations), it is still important for poor registration results (20 iterations).

Considering the results in both intra and inter-database experiments (Table 4.5, 4.7 and 4.8), the conclusion to be drawn is that the proposed combined feature sets is general and robust.

4.4.2 Quantitative validation

Commonly, in image registration tasks, the distribution of registration errors is not balanced as can be seen in Table 4.3.

In the SPREAD experiment, Table 4.5 reports that in the combined experiment, the MAE of the correct and poor classes are 0.76 ± 0.65 and 1.47 ± 1.22, respectively. On the contrary, the MAE of the wrong class is 6.12 ± 5.64. It is expected that the regression error of values of the wrong class is relatively larger than that of the other classes. However, it should be emphasized that only 3.9% of samples are available to make a regression model between 6 and 81.8 mm. We tried to add more samples and make the distribution more balanced by performing registrations with different number of iterations, but there is still room for improvement for the wrong class by adding more samples and data.

In terms of classification, we obtained F1 scores of 95.4%, 38.1% and 84.4% in the classes correct, poor and wrong, respectively (Table 4.5). For the wrong class, which is arguably most important for clinical application, the precision and recall are 84.6% and 84.3%, respectively. This means that 84.6% of all samples predicted to be over 6 mm are correct and the proposed method caught 84.3% of larger registration errors. Muenzing et al. [39] obtained F1 scores of 95.3%, 73.8% and 86.6% in the classes

[0,2), [2,5) and [5, ∞) mm. They achieved a better F1 score in the poor class and they also reported zero overlap between the correct and wrong classes. However, the comparison between the two methods is not easy because of the differences in the data. For example, the slice thickness in SPREAD is 2.5 mm, while it is 1 mm for Muenzing's data, which may affect the performance especially in the poor class. Moreover, we generated the classes by thresholding the regression values. Thus, the forests are optimized for regression not for classification.

4.4.3 Qualitative validation

Muenzing et al. [32] generated an uncertainty map by spatial interpolation of landmark-based quality estimates. On the contrary, our proposed system, which is trained on landmark locations, can be applied in all regions of the image. We showed this for two example images, see Fig. 4.5. It can be easily visualized that in the blue region, images are matched correctly. On the other hand, by tracking the vessels in the red region misalignment can be seen. Another note about the prediction is that there are no abrupt changes, and error varies smoothly from blue to yellow and then red, even though the error is predicted for each voxel independently.

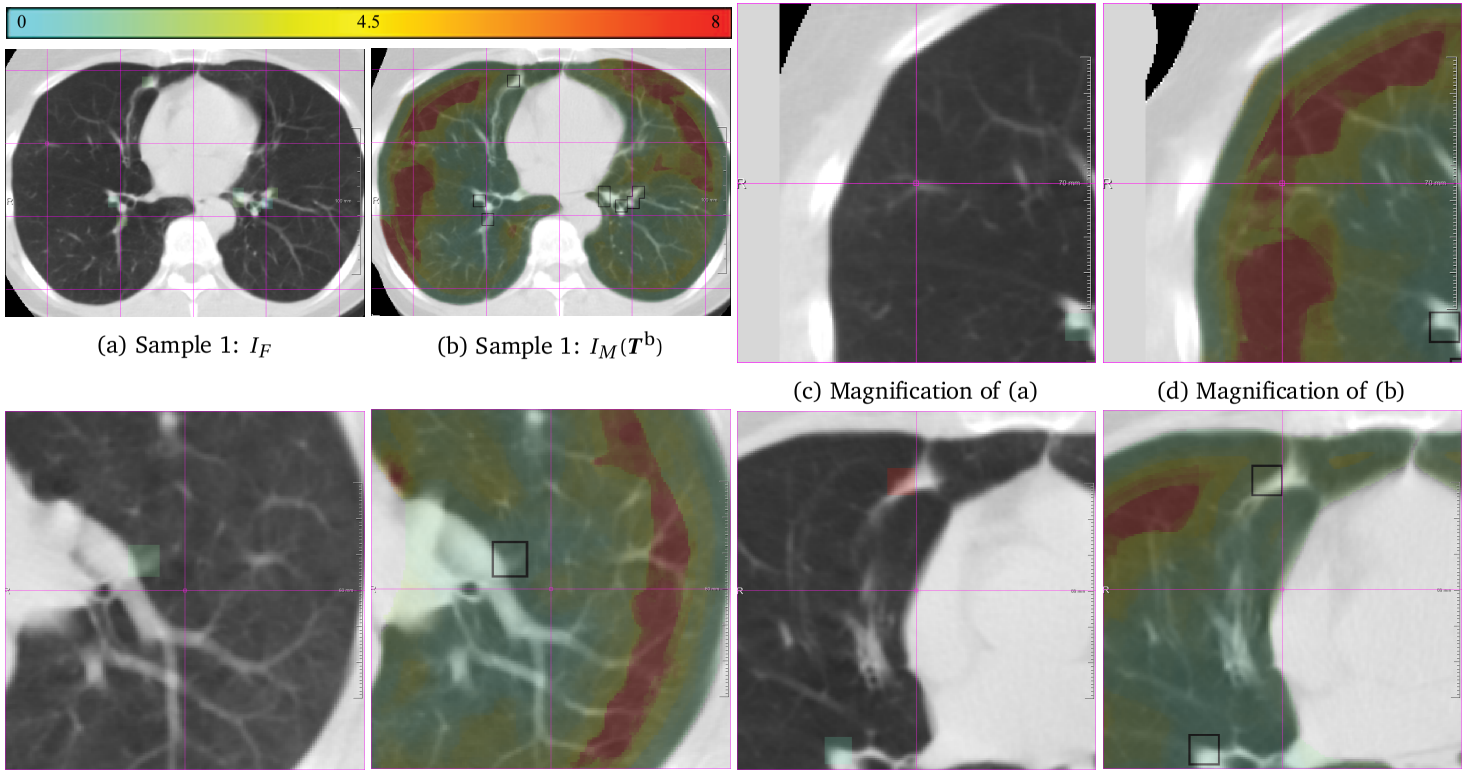
Another example is given in Fig. 4.10(a-d). Although all landmarks indicate that the registration error is small in this slice, the quantitative results found several misregistered regions, which implies that few landmarks may not be sufficient to assess the registration quality of the whole image. In Fig. 4.10(e, f), it can be observed that the performance in the homogeneous area (left side of the images) is as good as the performance in the area with structure. The main reason of acceptable performance in the homogeneous area is that the training samples consist of landmarks as well as their neighborhood region, which can be homogeneous. Thus, the system is trained both for homogeneous regions and regions with structure.

Another example is given in Fig 4.10(g, h), where the proposed system is not able to predict the registration error correctly because of a shift in the slice direction.

4.4.4 Limitations

Discrete optimization: If the optimization method is less or not dependent on the initial state, for instance for discrete optimization methods [34, 106], many of the proposed registration features, which are generated by varying the initial transformation of the registration, are not informative anymore. In such cases, instead of std $\boldsymbol{T}$ or std $\boldsymbol{T}^{\mathrm{L}}$, other measures can be used. For example, by utilizing the adaptive mean-shift algorithm, the local standard deviation of the displacement distribution can be calculated [106].

Anatomical changes: The proposed method is trained in such a way that any dissimilarity between the fixed and moving images is counted as misalignment in registration. In case of anatomical changes this assumption may be invalid, but



(a) Sample 1: I_F (b) Sample 1: $I_M(\boldsymbol{T}^{\mathrm{b}})$ (c) Magnification of (a) (d) Magnification of (b)

(e) Sample 2: I_F (f) Sample 2: $I_M(\boldsymbol{T}^{\mathrm{b}})$ (g) Sample 3: I_F (h) Sample 3: $I_M(\boldsymbol{T}^{\mathrm{b}})$

Figure 4.10: Several samples from the SPREAD dataset. The left column shows the fixed image with the ground truth registration error overlaid in color. The right column shows the moving image after registration with the registration error predicted by the proposed method overlaid in color.

typically prior knowledge of the underlying anatomy is required to determine which regions are allowed to be "misaligned" because of anatomical changes and which are not [107]. The proposed method highlights all changes, coming from misalignment or from anatomical change.

4.4.5 Future work

In the proposed method we predict the misalignment as an Euclidean distance in millimeters, rather than a 3D vector representing residual displacement. This is mostly because the features used in the system are not direction-wise, especially the local intensity features. The use of features that include directional information may help the system to be used in predicting the registration error in each direction, which is then effectively a new registration method.

The proposed method was tested on chest CT scans. Since the proposed features are generic and modality-independent, the overall method can in principle be applied to other modality data from other anatomical regions. The performance in such cases however remains to be investigated.

The uncertainty of affine registration is not measured in this work. Defining a gold standard for this mid-phase result is a complex task. However, extending the experiments to other databases where only affine transformations are applicable can be done in the future.

Instead of manually defined features, it is possible to use convolutional neural networks, which can learn features automatically. Eppenhof et al. [44] predicted the Euclidean distance of registration error. Our own work on CNNs for registration [17] can also be modified to predict registration uncertainty in a direction-wise manner. Both methods are trained only based on intensity, where the current paper shows that registration-derived information still contributes to a better regression. Thus, adding registration information to the neural networks should probably be considered as well.

A larger set of corresponding points annotated more densely throughout the scan could potentially also benefit training of the regression forest. In addition, experimenting on multi-modality data and investigating the contribution of all introduced features on them are future plans of this work.

Finally, the uncertainty map produced by the proposed method may be exploited to improve local registration results.

4.5 Conclusion

In this paper we proposed a method based on random regression forests to predict registration accuracy on chest CT scans from registration-based as well as intensity-based features. We introduced the variation in registration result from differences in initialization (std $\boldsymbol{T}$) and CVH, which showed high feature importance in several exper-

iments. Registration-based features provided additional information on registration error with respect to intensity-based features.

The regression method was evaluated on data from the SPREAD study and predicted the registration error with a mean absolute error of 1.07 ± 1.86 mm. The proposed method gained promising results on inter-database validation with a regression error of 1.76 ± 2.59 mm.

Acknowledgments

This work is financed by the Netherlands Organization for Scientific Research (NWO), project 13351. Ben Glocker received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 757173, project MIRA, ERC-2017-STG). Dr. M.E. Bakker and J. Stolk are acknowledged for providing a ground truth for the SPREAD study data used in this paper. We would like to thank Dr. R. Castillo and T. Guerrero for providing the DIR-Lab database.