Lung registration using the NiftyReg package

Marc Modat, Jamie McClelland and Sébastien Ourselin

Centre for Medical Image Computing, University College London, UK

Abstract. The EMPIRE 2010 grand challenge is a contest on lung image registration. This paper describes the implementation that has been used by the *Niftyreggers* team, as well as the results. The registrations were performed using a block-matching approach and the free-form deformation algorithm for global and local registration respectively.

1 Introduction

This paper presents the method and results for the *Niftyreggers* team's entry to the EMPIRE 2010 (Evaluation of Methods for Pulmonary Image Registrations 2010) grand challenge. The EMPIRE 2010 challenge, its goals, and the datasets used are explained elsewhere [1]. This paper details the methods used, including both the generic algorithms that can be used for many different registration tasks, and the specific protocols and parameters used for registering the EMPIRE 2010 datasets. It then discusses the results obtained and where and how these may be further improved in the future.

The NiftyReg package contains a global as well as a local registration algorithm. The global registration is based on a block-matching technique, as proposed by Ourselin *et al.* [2]. The local registration is based on the Free-Form Deformation (FFD) algorithm presented by Rueckert *et al.* [3]. The FFD technique has been re-factored by Modat *et al.* [4] in order to decrease the computation time, make the algorithm suitable for a Graphics Processing Units (GPU) implementation and improve the convergence trough an analytical formulation.

2 Method

2.1 Global registration

The block-matching algorithm for global (rigid or affine) registration can be thought of as a two-step process. The first consists in finding correspondences between sub-volumes of the two images to register. The second step deals with extracting from these correspondences some parameters for global deformation. These parameters correspond to rotation and translation when a rigid registration is performed and rotation, translation, scaling and shearing when an affine transformation is required.

The reference image R and the transformed floating image $F(\mathbf{T})$, where $\mathbf{T}$ represents the transformation, are divided into small sub-volumes, or blocks.

The similarity between each block in R and its neighbourhood in $F(\mathbf{T})$ is assessed using any intensity-based similarity measure. The block in $F(\mathbf{T})$ with the highest similarity is chosen as the corresponding block to the block in R. The global parameters are then extracted from these correspondences through a least trimmed square (LTS) regression method. The LTS method is robust to outliers [2]. The trimming constant is set to obtain 50% of the corresponding points as inliers. An iterative least square estimation is used to minimise the LTS criterion. This optimisation technique is computation efficient as each least square iteration has an analytical solution.

Once the correspondences and the transformation parameters have been evaluated, the global transformation is updated and the floating image is warped. This is done either until convergence of the transformation parameters or until a user-defined maximal number of iterations is reached. We also used a multi-scale approach where the first levels of registration are performed on down-sampled images. Thus decreases the computation time and increases the capture range of the method.

The final global registration parameters can then be used to initialise the local registration scheme.

2.2 Local Registration

The FFD algorithm is composed of three main parts: a deformation model, an objective function and an optimisation scheme.

Deformation model The deformation model is based on a lattice of control points $\boldsymbol{\mu}$ overlaid on the reference image R. The deformation field $\mathbf{T}(\boldsymbol{x})$ at each voxel position $\boldsymbol{x}$ can be computed from the control point positions through a cubic B-Spline interpolation scheme.

The cubic B-Spline interpolation scheme has the advantage of being C^2 continuous, which means that analytical first and second derivatives of the deformation model exist and can be computed, which helps the registration optimisation problem.

Objective function The positions of the control points $\boldsymbol{\mu}$ are displaced until an objective function is optimised. This function is, in the *classical* FFD scheme, composed from the normalised mutual information (NMI) as a metric and the bending energy (BE) as a penalty term. The NMI is a similarity measure based on entropy, whose computation is performed through a joint-histogram. Its optimisation leads to maximise the amount of information that one image has about another. As such, it does not require a linear relationship between the reference and floating image voxel intensities and is thus suitable for multi-modal image registration. The BE is computed from the cubic B-Spline deformation model and favours smooth warping between both images. For this application, we also add a second penalty term based on the squared Jacobian determinant in order to ensure a one-to-one mapping. The objective function $\mathcal{O}(R, F(\mathbf{T}); \mu)$ we used was then defined as:

$$\begin{aligned}\mathcal{O}(R, F(\mathbf{T}); \mu) = (1\alpha - \beta) \times & \frac{(H(R) + H(F(\mathbf{T}))}{H(R, F(\mathbf{T}))} \\ & - \alpha \times \sum_{\boldsymbol{x} \in R} \left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial x^2}\right)^2 + \left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial y^2}\right)^2 + \left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial z^2}\right)^2 \\ & +2 \times \left(\left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial xy}\right)^2 + \left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial yz}\right)^2 + \left(\frac{\partial^2 \mathbf{T}(\boldsymbol{x})}{\partial xz}\right)^2\right) \\ & - \beta \times \sum_{\boldsymbol{x} \in R} \log\left(|\mathrm{Jac}(\mathbf{T}(\boldsymbol{x}))|\right)^2\end{aligned} \tag{1}$$

where α and β are user-defined weights, H and $H(F(\mathbf{T}))$ are the marginal entropies and $H(R, F(\mathbf{T}))$ the joint entropy.

Optimization scheme In order to optimise $\mathcal{O}(R, F(\mathbf{T}); \mu)$, we used a conjugate gradient descent scheme. The gradient is computed analytically for each component of the function. The gradient of the NMI is assessed as described in [4]. Briefly, a Parzen window scheme is used to fill a joint-histogram. This allows us to calculate the derivative of the joint histogram and then the derivatives of the joint and marginal intensity probabilities. These derived probabilities are used to compute the derivatives of the joint and marginal entropies and subsequently the NMI gradient. All these computations are performed in a voxel-wise referential and then the node-wise gradient is extracted through convolution by a cubic spline kernel.

The BE is defined as the sum of the squared second derivatives of the deformation model. Its gradient can be easily computed at each voxel position using the chain rule, as described in [4]. For greater efficiency, we only approximate the BE itself and its gradient at the control position as suggested by Rolhfing *et al.* [5]. It allows reducing the amount of computation since there are less control points than voxels and the spline basis values can be pre-computed. The Jacobian-based penalty term is also computed only at the control point positions to improve the algorithm efficiency. In contrast to Rolhfing *et al.* [6], who used an approximated gradient of their objective function, we compute the gradient of the Jacobian-based penalty term analytically. It can be done as:

$$\begin{aligned}\frac{\partial\left(\sum_{\boldsymbol{x} \in R} \log\left(|\mathrm{Jac}(\mathbf{T}(\boldsymbol{x}))|\right)^2\right)}{\partial \mu_{i,j,k}} = & \; 2 \times \log\left(\det(\mathrm{Jac}(\mathbf{T}))\right) \\ & \times \mathrm{Tr}\left(\mathrm{Jac}^{-1}(\mathbf{T}) \frac{\partial \mathrm{Jac}(\mathbf{T})}{\partial \mu_{i,j,k}}\right),\end{aligned} \tag{2}$$

where i, j and k correspond to the control point indices respectively along the x-, y- and z-axis and Tr(.) the trace operator.

Since the Jacobian-based penalty term is, in our case, approximated and its gradient weighted against the NMI and BE derivatives, it does not strictly ensure non-folding and thus a one-to-one mapping. Previous work by Karacali *et al.* [7], aimed at correcting a deformation field for folding. We introduce a folding correction scheme performed concurrently with the registration process. For each voxel, which corresponds to a negative Jacobian determinant, we compute its influence on its neighbourhood control points and change the control point positions in order to increase the determinant value. This is done using the Jacobian determinant gradient itself. This scheme is used, using an approximation, each time the Jacobian-based penalty term reports some folded voxel and without approximation at the end of each level of a pyramidal approach.

2.3 Implementation

The code is available on http://sourceforge.net/projects/niftyreg and can be freely downloaded and compiled using Linux, MacOs or Windows. The code is based on the NIfTI-1.1 library [8], it thus deals with nifti and analyze formats.

Figure 1 presents the organisation in the framework as a diagram of the main functions. The code has been implemented using C/C++ and CUDA from NVidia. There are two versions of the code: a CPU- and a GPU-based implementation. We refer readers to [4, 9] for more details about the GPU-based implementation.

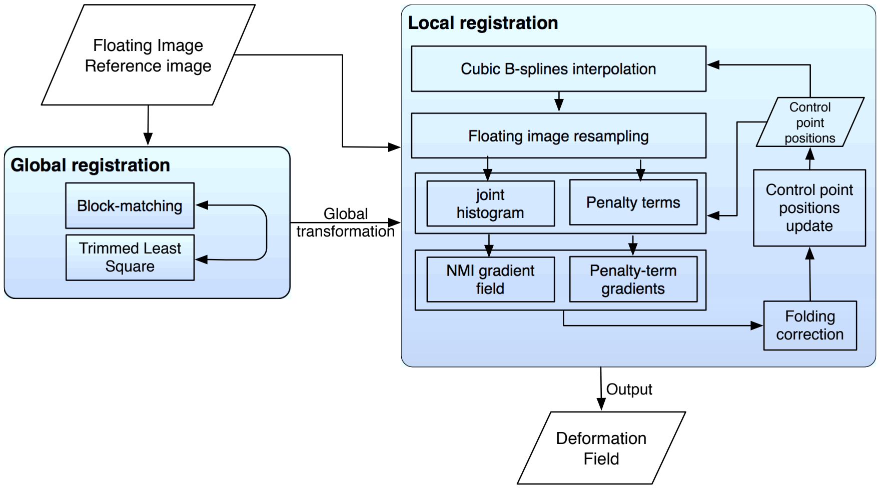

Fig. 1. Diagram of the NiftyReg framework main functions

3 Application

In order to perform the lung registrations between the different stages of the respiratory motion we used the previously described package using identical parameters for all the cases.

It consists of 1 global registration stage and 3 consecutive local registration stages. Figure 2 shows the result of the different stages applied to the first pair of scans. For each stage, the result of the previous stage was used to initialise the transformation. we have ensured that the total registration time for all stages is reasonable as there is a time constraint of 3 hours in which to perform another 10 registrations at the workshop. Furthermore, we always ensure that all stages of the registration can fit onto the GPU memory. Parameters were first determined using trial and error and previous experience of lung CT registration [10].

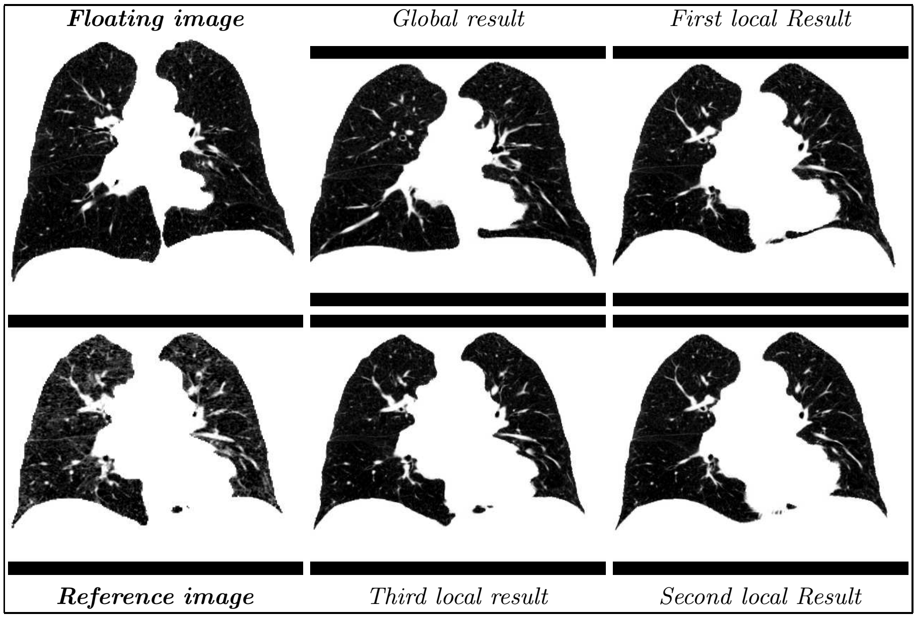

Fig. 2. Illustration of the different registration steps for case 01.

3.1 Global registration stage

A block matching approach was preferred since it allows feature alignment in the lung rather than global alignment of external lung volume boundaries. We obtained a better initialisation for the local registration than using a global scheme.

The size of the block was set to 4^3 voxels and the neighbourhood area of a block has been defined such that the block in the reference image and the block in the floating image always overlap by at least one voxel. As a measure of similarity we used cross-correlation. This has the advantage of being suitable for local intensity difference when used locally. In order to decrease the computation time and increase the robustness, only the blocks in the reference image with the highest variability were considered. They correspond to contrast areas. Only half of the possible blocks were then considered. Moreover, we used the provided reference lung mask to ignore any correspondences outside of the region of interest. Due to the large number of voxels, we only performed the two first levels of a three-level pyramidal approach. The TLS optimisation was considering 50% of the correspondence as inliers and thus 50% as outliers.

3.2 First local registration stage

The previously obtained global registration parameters were used to initialise the control point position. The aim of the initial local registration was to quickly register the main structures in the lung.

This has been done performing only 2 registration levels of a 4-level pyramidal approach. In other words, warping was performed on 3-time and twice down-sampled images. The control point spacing has been set to 6 voxel-width. The maximal number of iterations per level was set to 500 and the weight of the bending-energy penalty term to 0.01%. After computation of the NMI gradient, it is smoothed using a Gaussian kernel with standard deviation set to half the control point spacing size (3 voxels width). Moreover the reference image mask was used to ignore every voxel outside of the mask from the similarity measure computation, thus allowing the algorithm to align the lung features rather than the lung boundaries. No penalty term based on the Jacobian determinant has been used for this stage.

3.3 Second local registration stage

The second local registration stage was performed on twice down-sampled input images. In contrast to the previous stage, no mask is used, which ensure the algorithm aims at aligning the border of the lungs. Structures outside the lungs can move differently than the lungs during respiration. So that they do not affect the registration, all voxels outside the lung mask are set to 0 Hu (approximately soft tissue). The NMI gradient is still smoothed with the same kernel size, the bending-energy weight is set the 0.1% and the Jacobian-based penalty term is introduced with the same weight. The maximal number of iteration is set to 500. The aim of this stage was to quickly align the border of the lung.

3.4 Third local registration stage

Whereas the previous two stages aimed to coarsely align the internal structures and the lung borders, the goal of the final stage was to established a detailed alignment of the entire lung. This last stage was performed using three levels in the pyramidal approach: 3-time, twice and once down-sampled images. No mask was used, the penalty term weights were both set to 0.1% and no smoothing was performed on the NMI gradient field. The maximal number of iterations was set to 300 for each of the three levels. Since the registration was never performed at full image resolution due to the contest time constraint; we ensured that the produced transformation did not generate any folding. This was done by running our folding correction scheme at full resolution as a post-registration step.

4 Results

Table 1 shows the registration accuracy evaluation for the 20 pairs of images provided for the EMPIRE challenge. Evaluation has been performed by the EMPIRE 2010 (Evaluation of Methods for Pulmonary Image Registration 2010) board. Different criteria have been assessed such as lung boundary, fissure and landmark alignment. The singularities of the deformation field have also been evaluated using the Jacobian maps of the deformation.

Each registration was performed using our CPU-based single-threaded implementation. A computer cluster was used to assess the deformations, we are thus unable to report a specific architecture, but the range of processor clocks was 1.8 to 3 GHz. For comparison, the registrations were also performed using our GPU-based implementation. Moreover, it should be noted that we are planning to use the same graphical card (NVidia Quadro FX 2800m) on the same laptop during the workshop. Note that the global registration and the folding correction step require to much memory to be performed on the GPU. Table 2 reports the mean computation times for each architecture.

5 Discussion - Conclusion

The results in table 1 indicate that our registrations performed well in all the datasets. The method ensured that there were no singularities in any of the results. Our method was successful at aligning the lung boundaries, with less than 0.01 % error for any dataset. Although the proposed pipeline also performed reasonably well on the fissure and landmark evaluation, there were still some problematic cases. From visual inspection of the datasets before and after registration, we were able to draw some conclusions about the misalignment. All registrations appeared to align most of the lungs with no errors larger than the voxel size. However, some results contained particular regions, often near the back or base of the lung or near the fissures between different lobes that were misaligned, and leading to worse scores.

The amount of deformation changed greatly between datasets. Examination of the global results showed that for some cases the deformation could be reasonably well approximated by an affine transformation, whereas for others the deformation varied locally from one region to another, and could not be well approximated by an affine transformation. Datasets with the largest local variation tend to have the poorer results; in particular datasets 7 and 14, but also

	Lung Boundaries		**Fissures**		**Landmarks**		**Singularities**	
Scan Pair	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**
01	0.00	2.50	0.00	4.00	1.23	3.00	0.00	11.50
02	0.00	11.00	0.00	15.00	0.45	14.00	0.00	12.50
03	0.00	5.50	0.00	12.50	0.45	16.00	0.00	12.00
04	0.00	9.00	0.00	16.50	0.94	9.00	0.00	14.00
05	0.00	13.00	0.00	16.00	0.03	17.00	0.00	13.50
06	0.00	16.00	0.00	7.00	0.26	4.00	0.00	14.00
07	0.00	1.50	1.69	20.00	1.38	3.00	0.00	10.00
08	0.00	4.00	0.00	9.00	0.71	5.00	0.00	12.50
09	0.00	9.00	0.00	6.50	0.56	10.00	0.00	13.00
10	0.00	6.00	0.00	15.00	1.11	6.00	0.00	13.00
11	0.00	4.00	0.01	9.00	0.84	10.00	0.00	11.50
12	0.00	20.00	0.00	13.50	0.05	10.00	0.00	14.50
13	0.00	3.00	0.09	16.00	0.74	1.00	0.00	13.00
14	0.00	1.00	2.86	10.00	1.23	3.00	0.00	9.50
15	0.00	8.00	0.00	23.00	0.63	7.00	0.00	12.50
16	0.00	3.50	0.00	5.00	0.93	6.00	0.00	13.50
17	0.00	14.00	0.04	12.00	0.59	1.00	0.00	14.00
18	0.00	3.00	0.04	2.00	1.22	2.00	0.00	10.50
19	0.00	14.00	0.00	30.00	0.54	16.00	0.00	14.50
20	0.00	3.50	0.74	4.00	1.06	2.00	0.00	10.50
Avg	0.00	7.57	0.27	12.30	0.75	7.25	0.00	12.50
Average Ranking Overall								9.90
Final Placement								2

Table 1. Results for each scan pair, per category and overall. Rankings and final placement are from a total of 34 competing algorithms.

to a smaller degree datasets 1, 18, and 20. In these datasets some regions of the lungs expanded/contracted considerably (Fig. 3). Theses deformations induced changes in the density of the lung tissue and hence changes in intensity in the CT images. Visual examination of the results indicate that when there were large intensity changes in one region but not in another region close by (e.g. adjacent lobes) the registrations failed to recover the correct deformation (Fig. 3). This appeared to be the main cause of most of the errors that were observed. Other factors that appeared to contribute to the registration errors in some datasets were some relatively homogenous regions of the images which had few features to guide the registrations and sliding between adjacent lobes (Fig. 3).

As already mentioned the submitted results were performed using the CPU-based implementation. Table 2 shows that the CPU registration times are acceptable for the on-site registrations, with all 20 registrations taking just over 6 hours. Table 2 shows that the GPU-based implementation achieves a speed up for all the stages of the registration where it was used, but especially for stage

Computation time	CPU-based implementation	GPU-based implementation
Global registration	1.40 (0.57)	NA
Local registration 1	1.28 (0.47)	1.06 (0.41)
Local registration 2	1.32 (0.84)	0.83 (0.36)
Local registration 3	12.25 (5.13)	1.56 (0.27)
Folding correction	1.16 (0.57)	NA

Table 2. Mean computation time in minutes (and standard deviation) of the different registration stages.

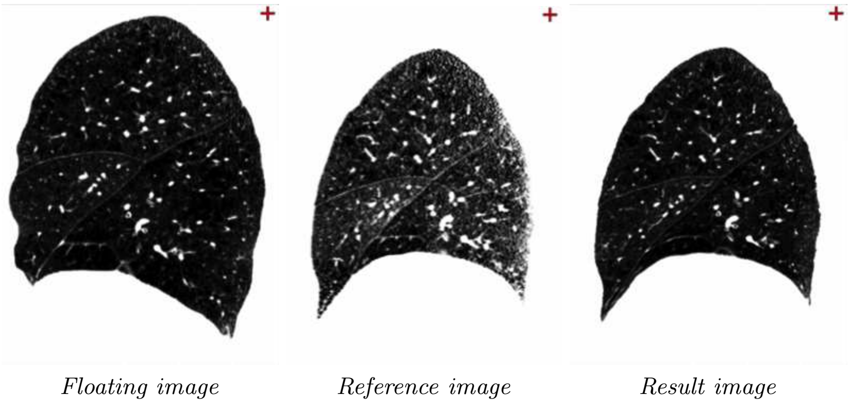

Fig. 3. Comparing the floating and reference images it can be seen that the lower lobe is deforming much more than the upper lobe, some regions contract more than others (such as near the back of the lung and just above the fissure) causing a larger increase in CT intensity in these regions, and that the two lobes are sliding past each other along the fissure.

3, which is the most computationally demanding stage, and where a speed up of greater than 7.8 times was achieved. The average total registration time is just under 6 minutes for the GPU-based implementation, so it should be possible to register the 10 datasets on-site in approximately 1 hour, well within the 3 hour time limit. It should be noted that one of the datasets (dataset 2) was too large for final local registration stage to be run on the GPU. This can be detected prior to running the registration, and when the memory requirements are too great for the GPU-based implementation, the CPU-based implementation can be used instead.

Better registration accuracy might be possible by modulation of the image intensities according the volume change, as suggested by Yin *et al.* [11]. The use of a locally weighted metric [12] or a local similarity measure [13] might also improve the registration result. To account for sliding between different lobes the lobes could be segmented and registered separately [14]. We intend to investigate each of these approaches to improve the registrations in the future.

Acknowledgments

Authors would like the thank Pr. D.J. Hawkes, P. Daga, M.J. Cardoso, Dr. G.R. Ridgway and J. Cleary for their help and support with the NiftyReg package and T. Clark from the UCL CS Technical Support Group for assistance using the cluster.

References

1. Murphy, K., van Ginneken, B., Reinhardt, J., Kabus, S., Ding, K., Deng, X., Pluim, J.: Evaluation of methods for pulmonary image registration: The EMPIRE10 study. In: Grand Challenges in Medical Image Analysis. (2010)
2. Ourselin, S., Roche, A., Subsol, G., Pennec, X., Ayache, N.: Reconstructing a 3D structure from serial histological sections. **19**(1-2) (2001) 25–31
3. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: Application to breast MR images. IEEE Trans. Med. Imag. **18** (1999) 712–721
4. Modat, M., Ridgway, G.R., Taylor, Z., Lehmann, M., Barnes, J., Hawkes, D., Fox, N.C., Ourselin, S.: Fast free-form deformation using graphics processing units. Comput. Methods Programs Biomed. **98**(3) (2010) 278–84
5. Rohlfing, T., Maurer, C.: Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. IEEE Trans. Inf. Technol. Biomed. **7**(1) (2003) 16–25
6. Rohlfing, T., Maurer, C., Bluemke, D., Jacobs, M.: Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint. IEEE Trans. Med. Imag. **22**(6) (2003) 730–741
7. Karaçali, B., Davatzikos, C.: Estimating topology preserving and smooth displacement fields. **23**(7) (2004) 868–80
8. Cox, R., Ashburner, J., Breman, H., Fissell, K., Haselgrove, C., Holmes, C., Lancaster, J., Rex, D., Smith, S., Woodward, J., et al.: A (sort of) new image data format standard: Nifti-1. Human Brain Mapping **25** (2004)
9. Daga, P., Modat, M., Micallef, C., Mancini, L., White, M., Cardoso, M.J., Kitchen, N., McEvoy, A., Thornton, J., Yousry, T., Hawkes, D., Ourselin, S.: Near real time brain shift estimation for interventional MRI suite. In: High-Performance MICCAI. (2010)
10. McClelland, J.R., Blackall, J.M., Tarte, S., Chandler, A.C., Hughes, S., Ahmad, S., Landau, D.B., Hawkes, D.J.: A continuous 4d motion model from multiple respiratory cycles for use in lung radiotherapy. Med. Phys. **33**(9) (2006) 3348–58
11. Yin, Y., Hoffman, E., Lin, C.: Mass preserving nonrigid registration of ct lung images using cubic b-spline. Med. Phys. **36** (2009) 4213–4222
12. Loeckx, D., Slagmolen, P., Maes, F., Vandermeulen, D., Suetens, P.: Nonrigid image registration using conditional mutual information. IEEE Trans. Med. Imag. **29**(1) (2010) 19–29
13. Avants, B., Epstein, C., Grossman, M., Gee, J.: Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Med. Image Anal. **12**(1) (2008) 26–41
14. Ding, K., Youbing, Y., Cao, K., Christensen, G., Lin, C., Hoffman, E.A.and Reinhardt, J.: Evaluation of lobar biomechanics during respiration using image registration. In: Proc. MICCAI'09. Volume 5761. (2009) 739–746