A FAST AND ACCURATE PARALLEL ALGORITHM FOR NON-LINEAR IMAGE REGISTRATION USING NORMALIZED GRADIENT FIELDS

Lars König and Jan Rühaak

Fraunhofer MEVIS, Project Group Image Registration, Lübeck, Germany

ABSTRACT

We present a novel parallelized formulation for fast non-linear image registration. By carefully analyzing the mathematical structure of the intensity independent Normalized Gradient Fields distance measure, we obtain a scalable, parallel algorithm that combines fast registration and high accuracy to an attractive package. Based on an initial formulation as an optimization problem, we derive a per pixel parallel formulation that drastically reduces computational overhead.

The method was evaluated on ten publicly available 4DCT lung datasets, achieving an average registration error of only 0.94 mm at a runtime of about 20 s. By omitting the finest level, we obtain a speedup to 6.56 s with a moderate increase of registration error to 1.00 mm. In addition our algorithm shows excellent scalability on a multi-core system.

Index Terms— Image registration, Computational efficiency, Parallel algorithms

1. INTRODUCTION

The problem of image registration and generally correspondence detection between two or more images has been extensively studied [1]. Applications in medical imaging range from motion compensation to intra-operative fusion of different modalities. In particular, non-linear registration methods are able to capture complex deformations with high accuracy, enabling advanced diagnosis and treatment [2]. Many of these methods, however, exhibit long processing times or require special hardware such as GPUs. While both resolutions and the number of imaging modalities are increasing, efficient tools that run on available hardware are needed.

In this paper, we present a novel approach to both efficient and accurate non-linear image registration. We directly target the underlying mathematical structure of the entire algorithm instead of only optimizing selected parts. We perform a deep analysis of the objective function associated with the registration model, by which we join the major building blocks to a closed analytical formulation. This allows parallelization on a per pixel level with close-to-zero memory consumption, directly executable on standard CPUs.

This work was partly funded by the European Regional Development Fund (EFRE).

2. RELATED WORK

The efficiency of registration algorithms has been widely discussed. A general framework for fast registration has been presented in 2004 [3]. Related approaches try to reduce the computational complexity using adaptive discretizations [4].

With the ubiquity of multicore systems, parallel implementations have moved into the focus of the research community, see [5] for an overview. A detailed approach to data-distributed parallel registration was presented in [6], whereas newer work deals with the use of GPUs for accelerating non-linear registration, e.g. [7] and references therein. A different approach for rigid registration has been provided in [8], exploiting the mathematical structure to obtain a fully parallel algorithm. This idea is picked up in our work and extended to non-linear image registration.

3. METHOD

To obtain a custom-tailored, efficient algorithm, we first give a short outline about our registration framework which allows a thorough analysis of the components and their interaction.

3.1. Registration Framework

The goal of image registration is to establish correspondence between a reference and a template image [9]. The images are acquired as discrete arrays $R \in \mathbb{R}^{abc}$ and $\hat{T} \in \mathbb{R}^{\hat{a}\hat{b}\hat{c}}$ in column vectors representing three-dimensional images, R of size $a \times b \times c$ with grid spacings h_x, h_y, h_z, $\hat{T}$ analogously.

Correspondence is established by deforming the template image onto the reference image using a transformation $Y \in \mathbb{R}^{3ABC}$ consisting of ABC three-dimensional deformation coordinates. To be able to evaluate the template at those coordinates, the discrete image is transferred to a continuous model using trilinear interpolation, obtaining the interpolation function $T : \mathbb{R}^{3abc} \to \mathbb{R}^{abc}$, which maps a set of coordinates to a deformed image in the reference image space.

In our model, the size of the deformation Y is independent of any image extent. This allows to adapt the deformation resolution to the size of the structures to be registered, thus decreasing both problem size and registration time. For comparing the deformed template with the reference image, the de-

Preprint submitted to 2014 IEEE International Symposium on Biomedical Imaging.
Received October 10, 2013.

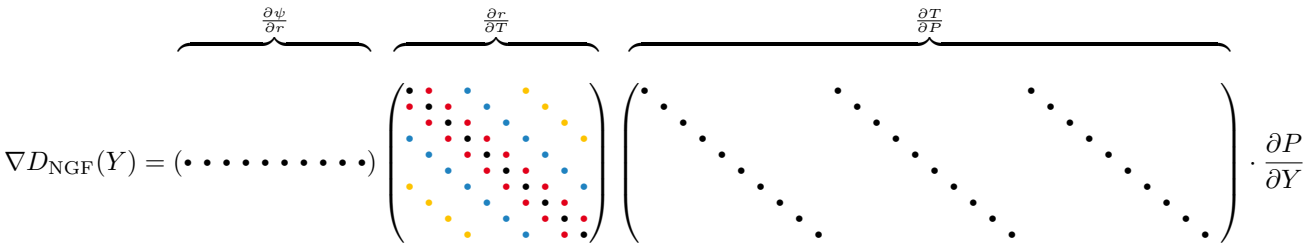


Fig. 1. Schematic view of the sparse matrix structure in the computation of ∇D. Diagonals in $\frac{\partial r}{\partial T}$ resulting from neighboring points in the same direction are shown in the same color.

formation needs to be converted to the reference image extent using a function $P : \mathbb{R}^{3ABC} \to \mathbb{R}^{3abc}$, so that the deformed template can be evaluated as $T(P(Y)) : \mathbb{R}^{3ABC} \to \mathbb{R}^{abc}$.

To quantify correspondence between the two images, we define a distance measure $D(Y) : \mathbb{R}^{3ABC} \to \mathbb{R}$, which measures the similarity of reference and deformed template image, depending on the deformation Y. The minimization of D is an ill-posed problem and needs a regularization term $S(Y)$ to ensure certain deformation properties, such as smoothness or specific physical behavior. Combining these two terms, the registration problem can be written as an optimization problem $J(Y) = D(Y) + \alpha S(Y) \xrightarrow{Y} \min$, where α balances image similarity and deformation regularity. The optimization problem is then solved by Newton-type methods [10].

Since the formulation of each part of this optimization problem is crucial, we will now look precisely at the components and derive specific methods for efficient parallel computation of their function values as well as their derivatives.

3.2. Distance measure

We focus on the Normalized Gradient Fields (NGF) distance measure [9], that has been successfully proven to be both well suited for multimodal registration problems and parallelization [8]. The general assumption in this distance term is that intensity changes, which naturally represent edges, are preserved across different modalities. The NGF evaluates the angles between these image gradients and has a lower value the more parallel the gradients are aligned. The maximum value is obtained by orthogonal gradients.

In [9] NGF has been introduced in a continuous framework. To obtain a discretized formulation, we use the midpoint quadrature rule on the reference image domain. With the product of the image grid spacings $\bar{h} = h_x h_y h_z$ and $\|\cdot\|_\varepsilon = \sqrt{\langle\cdot,\cdot\rangle + \varepsilon^2}$ we can write the NGF as

$$D(Y) = \frac{\bar{h}}{2} \sum_{i=1}^{abc} \left(1 - \left(\frac{\langle \nabla T_i(P(Y)), \nabla R_i \rangle + \tau\varrho}{\|\nabla T_i(P(Y))\|_\tau \|\nabla R_i\|_\varrho} \right)^2 \right), \tag{1}$$

where $\tau, \varrho > 0$ are modality dependent parameters, which achieve robustness against noise.

3.3. Parallel derivative computation

The most time of the registration is typically spent evaluating the distance measure and its derivative. While the function value computation is directly parallelizable using (1), the gradient computation is more involved. It consists of several separate steps, that need to be investigated in detail to derive a joint, parallelizable formulation. The steps can be described as follows: Convert deformation to reference image grid → Compute deformed template → Compute NGF residual → Final reduction step. These steps translate to the function chain

$$\mathbb{R}^{3ABC} \xrightarrow{P} \mathbb{R}^{3abc} \xrightarrow{T} \mathbb{R}^{abc} \xrightarrow{r} \mathbb{R}^{abc} \xrightarrow{\psi} \mathbb{R} \tag{2}$$

with reduction function $\psi : \mathbb{R}^{abc} \to \mathbb{R}$, $(r_1, \ldots, r_{abc})^\top \mapsto \frac{\bar{h}}{2} \sum_{i=1}^{abc} (1 - r_i^2)$. Using (1), the ith component of r can be written as

$$r_i(T) = \frac{\langle g(T_i), g(R_i) \rangle + \tau\varrho}{\|g(T_i)\|_\tau \, \|g(R_i)\|_\varrho},$$

where $g(T_i)$ is the image gradient approximation of T at the point i using forward/backward finite differences as in [8].

Mathematically, the derivative of (1) can directly be computed using the chain-rule, yielding $\nabla D_{\mathrm{NGF}}(Y) = \frac{\partial \psi}{\partial r} \frac{\partial r}{\partial T} \frac{\partial T}{\partial P} \frac{\partial P}{\partial Y}$. Calculating this in a matrix-based fashion, the formulation is difficult to parallelize because of dependencies on intermediate results and unknown matrix structures. Hence, we take a closer look at the structure of the single components, which is visualized in Fig. 1. Exploiting the banded structure of $\frac{\partial r}{\partial T}$, which only contains non-zero elements at neighboring points, we can derive a compact closed formulation of each gradient element. By evaluating the complete matrix chain, point-wise, down to its very basic elements (the images), this formulation can directly be computed fully in parallel from the input data, eliminating intermediate memory write accesses and computational overhead.

With the set of offsets to points adjacent to point i in a 3D-neighborhood defined as $\mathcal{M} = \{-z, -y, -x, 0, +x, +y, +z\}$ with zero Neumann boundary conditions, using the notation as in [8] we can first define

$$(\hat{r}_i)_l = \frac{1}{2h_l} \left(\frac{R_{i-l} - R_i}{\|g_i(R)\|_\varrho \|g_i(T)\|_\tau} - \frac{\left(\langle g_i(T), g_i(R) \rangle + \varrho\tau\right)(T_{i-l} - T_i)}{\|g_i(R)\|_\varrho \|g_i(T)\|_\tau^3} \right)$$

Are other more common metrics parallelizable?

HOw are these modality dependent parameters and how do they achieve robustness agains noise?

What's this Final reduction step?

This seems to be a crucial part but it's not really clear how it's being parallelized.

Preprint submitted to 2014 IEEE International Symposium on Biomedical Imaging.
Received October 10, 2013.

with $T_i := T_i(P(Y))$. Then the $i + l$th element of the row vector $\frac{\partial r_i}{\partial T}$ can be written as

$$\left(\frac{\partial r_i}{\partial T}\right)_{i+l} = \begin{cases} (\hat{r}_i)_l, & \text{if } l \in \mathcal{M} \setminus 0 \\ -\sum_{j \in \mathcal{M} \setminus 0} (\hat{r}_i)_j, & \text{if } l = 0 \\ 0, & \text{otherwise} \end{cases}.$$

The final gradient element at position i is given by

$$(\nabla D)_i = \left(\sum_{j \in \mathcal{M}} -r_j \left(\frac{\partial r_j}{\partial T}\right)_{i-j} \right) \frac{\partial T_i}{\partial p_{i+d \cdot abc}}, \quad (3)$$

with $d = 0, 1, 2$ for derivatives regarding x-,y-,z-coordinates, respectively. This formulation does not contain dependencies between single gradient elements and can be calculated without intermediate steps from the input data. Thus it can be fully parallelized, given a per-element formulation of the interpolation function and grid conversion $T(P(Y))$. This will be discussed in the next section.

3.4. Grid conversion

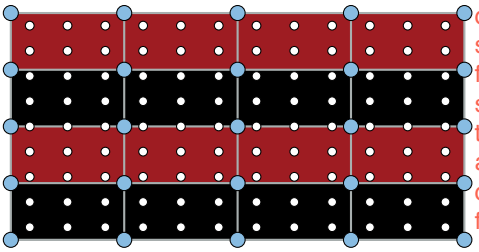
What does the "fixed Jacobian P" have to do with the interpolation?

The conversion between deformation and reference image discretization is performed using trilinear interpolation. As the interpolation weights only depend on the spacing of deformation and reference image, not on the current deformation, the conversion is a linear operation with fixed Jacobian P. For both NGF function value and gradient, the conversion from deformation to image grid is needed. This can easily be implemented in a matrix-free fashion by looping over the image grid, collecting all adjacent deformation grid points with their associated interpolation weights and summing up. Moreover, the computation can directly be parallelized as there are no write conflicts.

Setting $v := \frac{\partial \psi}{\partial r} \frac{\partial r}{\partial T} \frac{\partial T}{\partial P}$, the gradient computation for NGF is equivalent to the matrix-vector product $P^\top v$. We use a red-black scheme for efficient parallel implementation. The iteration is performed over the deformation grid cells, allowing write access to eight grid points at the same time. The algorithm is parallelized on the image slices: In the first loop, only the odd slices are considered, allowing for unconflicted writes as the slices themselves are computed serially. In the second loop, the even slices are calculated, finalizing the result. Fig. 2 illustrates our approach.

3.5. Regularizer

The last term in the objective function is the regularizer. Here, we choose Curvature Regularization [11], which favors a smooth deformation field. It has successfully been used in non-linear registration problems [12]. Since its computation is lightweight and easy parallelizable it is well suited to accompany the parallelized NGF. Discretized on the transformation grid and using the decomposition $Y = X + U$, where



Why is the deformation spacing different from the image spacing? I thought this was a dense displacement field.

Fig. 2. Red-black scheme for transposed grid conversion in 2D, with deformation (blue) and image grid (white). The red rows are processed in parallel, followed by the black rows. Only the adjacent blue nodes are written in each step such that no write conflicts can occur.

X is the identity, the curvature regularizer can be written as

$$S_{\text{Curv}}(Y) = \frac{\bar{h}_Y}{2} \sum_{i=i}^{ABC} \left((\Delta_i U_1)^2 + (\Delta_i U_2)^2 + (\Delta_i U_3)^2 \right),$$

where U_i represents the ith component function of the vector field deformation U. The function $\Delta_i : \mathbb{R}^{ABC} \to \mathbb{R}$ is a finite difference approximation to the Laplace operator at point i

$$\Delta_i U_k = \sum_{j \in \{x,y,z\}} \frac{1}{h_k^Y h_k^Y} \left((U_k)_{i-j} - 2 (U_k)_i + (U_k)_{i+j} \right),$$

where $i \pm x, i \pm y, i \pm z$ represent the neighboring points of i in the respective directions and h_k^Y the grid spacing of Y. Here, we use zero Neumann boundary conditions. The i-th element of the gradient of the regularizer is then given by

$$(\nabla S_{\text{Curv}})_i = \bar{h}_Y \left(\Delta_i U_1 + \Delta_i U_2 + \Delta_i U_3 \right).$$

Note that due to discretization on the deformation grid, no grid conversion is needed for the regularizer. With this formulation we have the complete objective function available as per point parallelizable terms.

3.6. Optimization

To gain additional speedup and avoid being trapped in local minima, the presented objective function is optimized in a multi-level approach. For this, the problem is successively solved on finer representations, using the minimizer from each coarser level as a starting guess for the next finer level. On each level the objective function is iteratively minimized using an L-BFGS approach, which is known for its memory efficiency and fast convergence [10].

4. EVALUATION

We have evaluated the accuracy and computational efficiency of our method on the challenging problem of CT lung registration. Since the air volume inside the lung varies while

Case	LME (a)	LME (b)	Time (a)	Time (b)
1	0.78 ± 0.89	0.76 ± 0.89	18.71 s	4.12 s
2	0.79 ± 0.90	0.80 ± 0.88	19.58 s	5.71 s
3	0.93 ± 1.05	0.96 ± 1.07	18.64 s	4.42 s
4	1.27 ± 1.27	1.33 ± 1.29	22.95 s	4.05 s
5	1.07 ± 1.46	1.18 ± 1.45	18.77 s	5.50 s
6	0.90 ± 0.99	1.03 ± 1.04	19.71 s	7.31 s
7	0.85 ± 0.98	0.92 ± 0.93	27.34 s	10.12 s
8	1.03 ± 1.23	1.13 ± 1.15	24.98 s	9.22 s
9	0.94 ± 0.93	1.00 ± 0.96	20.42 s	6.82 s
10	0.86 ± 0.97	0.91 ± 0.99	17.89 s	8.36 s
Avg.	0.94 ± 1.07	1.00 ± 1.07	20.90 s	6.56 s

Table 1. DIR-Lab datasets: Comparison of runtime and landmark error (LME) with $\alpha = 5, \tau, \varrho = 100$. Configuration (a) uses the full resolution, (b) omits the finest level in the multilevel approach. All values are given in millimeters. The initial landmark error ranged from 3.89 ± 2.79 mm to 14.99 ± 9.01 mm. The registrations were performed on a stock 3.4 GHz Intel i7-2600 quad-core PC running Ubuntu Linux.

Method	Serial	Parallel	Speedup
NGF	55.08 s	4.13 s	13.34
∇ NGF	94.96 s	7.72 s	12.30
Px	8.98 s	0.76 s	11.82
$P^\top x$	9.18 s	0.77 s	11.92

Table 2. Higher resolution datasets (512^3 image resolution, 129^3 deformation grid size): Scaling of NGF gradient and grid conversion on a 12-core dual CPU Intel Xeon E5645

breathing, the intensities in the acquired images are not directly comparable, which makes the datasets appropriate for the intensity independent NGF. For the evaluation we used the publicly available DIR-Lab 4DCT datasets [13, 14]. As we are only interested in the deformation of lung tissue, we segmented the lungs from the images [15]. The inhale and exhale phases come with 300 expert annotated landmark pairs that can be used to assess registration accuracy.

To show the scalability of our algorithm, we performed separate calculations of the NGF and the grid change operators single and multithreaded on a 12-core workstation.

5. RESULTS

On the DIR-Lab data we achieved a mean registration error of 0.94 mm with an average complete runtime of 20.9 seconds. Omitting the finest level, we obtained a speedup to 6.56 seconds with a moderate increase of average registration error to 1.00 mm. The detailed results of all cases are shown in Table 1. The result deformation fields were automatically checked and found to be free of singularities.

For eight of the ten cases the landmark errors were equal to or better than the lowest errors reported in [16]. Additionally the computation time compares very favorably with the competing algorithms. Comparing the single threaded computation time to a multithreaded calculation on a 12-core system, shown in Table 2, speedup factors from 11.82 to 13.34 are obtained, which implies a perfect linear scalability.

Hence, our algorithm combines accuracy and efficiency to a very attractive package. In addition our method does not require any special equipment such as multi-CPU servers or specialized GPUs. It runs on readily available stock hardware that is already used in the clinic.

6. REFERENCES

[1] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: a survey.," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1153–90, 2013.

[2] C. J. Galbán et al., "Computed tomography-based biomarker provides unique signature for diagnosis of COPD phenotypes and disease progression," *Nature Medicine*, 2012.

[3] B. Fischer and J. Modersitzki, "A unified approach to fast image registration and a new curvature based registration technique," *Linear Algebra Appl*, vol. 380, pp. 107–124, 2004.

[4] E. Haber, S. Heldmann, and J. Modersitzki, "Adaptive mesh refinement for nonparametric image registration," *SIAM J Sci Comput*, vol. 30, pp. 3012–3027, 2008.

[5] R. Shams et al., "A survey of medical image registration on multicore and the GPU," *Signal Processing Magazine, IEEE*, vol. 27, no. 2, pp. 50–60, 2010.

[6] F. Ino, K. Ooyama, and K. Hagihara, "A data distributed parallel algorithm for nonrigid image registration," *Parallel Computing*, vol. 31, no. 1, pp. 19–43, 2005.

[7] X. Gu et al., "Implementation and evaluation of various demons deformable image registration algorithms on a GPU," *Phys Med Biol*, vol. 55, no. 1, pp. 207, 2010.

[8] J. Rühaak, L. König, et al., "A fully parallel algorithm for multimodal image registration using normalized gradient fields," in *ISBI, IEEE*, 2013, pp. 572–575.

[9] J. Modersitzki, *FAIR: Flexible Algorithms for Image Registration*, SIAM, 2009.

[10] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, 2006.

[11] B. Fischer and J. Modersitzki, "Curvature based image registration," *J Math Imaging Vis*, vol. 18, no. 1, pp. 81–85, 2003.

[12] J. Rühaak, S. Heldmann, T. Kipshagen, and B. Fischer, "Highly accurate fast lung CT registration," in *SPIE Medical Imaging, Image Processing*, 2013.

[13] R. Castillo et al., "A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets," *Phys Med Biol*, vol. 54, no. 7, pp. 1849, 2009.

[14] E. Castillo et al., "Four-dimensional deformable image registration using trajectory modeling," *Phys Med Biol*, vol. 55, no. 1, pp. 305, 2010.

[15] B. Lassen et al., "Lung and lung lobe segmentation methods at Fraunhofer MEVIS," in *Fourth International Workshop on Pulmonary Image Analysis*, 2011, pp. 185–200.

[16] R. Castillo, "DIR-Lab - Results," `www.dir-lab.com/Results.html`, 2013, [Online; accessed 01-Oct-2013].

Preprint submitted to 2014 IEEE International Symposium on Biomedical Imaging.
Received October 10, 2013.