An Efficient and Robust Algorithm for Parallel Groupwise Registration of Bone Surfaces

Martijn van de Giessen[1,2,3,4], Frans M. Vos[1,5], Cornelis A. Grimbergen[4], Lucas J. van Vliet[1], and Geert J. Streekstra[4]

[1] Quantitative Imaging Group, Delft University of Technology, The Netherlands
[2] Division of Image Processing, Leiden University Medical Center, The Netherlands
[3] Department of Intelligent Systems, Delft University of Technology, The Netherlands
[4] Dept. of Biomed. Engineering and Physics, AMC Amsterdam, The Netherlands
[5] Dept. of Radiology, AMC Amsterdam, The Netherlands
m.vandegiessen@lumc.nl

Abstract. In this paper a novel groupwise registration algorithm is proposed for the unbiased registration of a *large* number of densely sampled point clouds. The method fits an evolving mean shape to each of the example point clouds thereby minimizing the total deformation. The registration algorithm alternates between a computationally expensive, but parallelizable, deformation step of the mean shape to each example shape and a very inexpensive step updating the mean shape.

The algorithm is evaluated by comparing it to a state of the art registration algorithm [5]. Bone surfaces of wrists, segmented from CT data with a voxel size of $0.3 \times 0.3 \times 0.3\,\mathrm{mm}^3$, serve as an example test set. The negligible bias and registration error of about 0.12 mm for the proposed algorithm are similar to those in [5]. However, current *point cloud* registration algorithms usually have computational and memory costs that increase quadratically with the number of point clouds, whereas the proposed algorithm has linearly increasing costs, allowing the registration of a much larger number of shapes: 48 versus 8, on the hardware used.

1 Introduction

Groupwise registration is a recurring problem in many medical applications. Two prominent applications are atlas building and the construction of statistical shape models (SSM). Such registrations should be unbiased in the sense that the outcome must not depend on the selection of a target or on the order in which the samples are processed. Furthermore, it is often desirable to register a large number of samples, such that the atlas or SSM generalizes well. However, depending on chosen similarity criteria and allowable transformations the groupwise registration problem may become intractable, both in terms of computational expense as well as memory requirements.

Group-wise registration algorithms are available both for voxel-based registrations, e.g. [3] as for point cloud registrations, e.g. [2]. The most important methodological difference between voxel-based and point cloud registrations is in the correspondence measure, e.g. intensity based vs. distance based. In [6]

N. Ayache et al. (Eds.): MICCAI 2012, Part III, LNCS 7512, pp. 164–171, 2012.

it is proposed to approximate the point clouds with Gaussian kernels and to (densely) sample the space with clouds on a grid, effectively transforming the point cloud registration problem into an intensity registration problem. However, using the L_2 divergence and Gaussian kernels to estimate the density function such a sampling can be avoided and the divergence can be computed efficiently in closed form [5]. Unfortunately, the computational and memory costs of the solution proposed in [5] grow quadratically with the number of registered shapes.

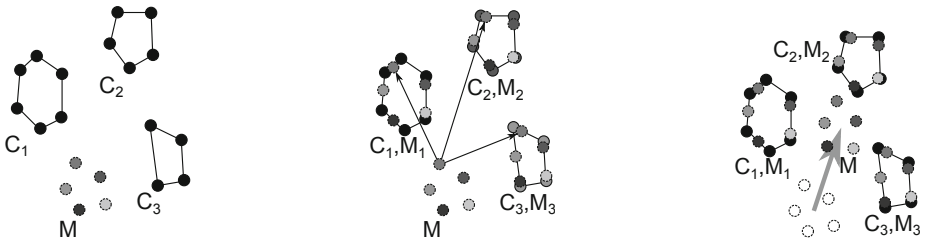
In this work we propose a solution to the groupwise point cloud registration problem that has a computational complexity that increases *linearly* with the number of point clouds to be registered. In this algorithm (I) the computational and memory costs grow linearly with the number of clouds, and (II) the algorithm is trivially parallellizable. This allows the unbiased registration of a very large number of point clouds on regular hardware and permits GRID computing. The registration problem is solved by independently evolving copies of a mean cloud that has minimal deformation with respect to each of the point clouds. This solution differs from common approaches where all example point clouds are deformed to the evolving mean [3,6,5] and has the advantage that an implicit point correspondence is present between all registered shapes, without the need for an image grid as in [3,6] This implicit correspondence allows the computationally inexpensive mean shape update. Both the accuracy and efficiency of the proposed method are compared to a state-of-the-art method [5] by applying the registration to three sets of 48 wrist bones (See Figure 3).

2 Methods

The registration algorithm that is proposed in this paper establishes correspondence between N point clouds $C_i, i = 1, \ldots, N$ and an evolving mean cloud M with n_m points. The numbers of points n_i in all clouds C_i do not need to be the same. For each cloud C_i, a deformed copy of M that approximates C_i is denoted as M_i. The algorithm to evolve clouds M and $M_i, i = 1, \ldots, N$ consists of five steps, outlined below. The first three steps are illustrated in Figure 1. After estimating an initial cloud M (step 1), the non-rigid registration in step 2 is the computationally most expensive step. Due to the splitting up of the procedure in a registration (step 2) and update of the mean shape (step 3), step 2 can be performed separately for each cloud C_i and is therefore trivially parallelizable.

Step 1: Estimate an Initial Mean Cloud M. In this work an initial *coarse* alignment of the point clouds C_i is assumed. Furthermore, the initial mean cloud M is assumed to come from a surface with the same topology as the clouds C_i. In this work the surfaces sampled by C_i are available and the initial estimate of M is obtained by sampling the 0-level of the average signed distance transforms of the surfaces.

Step 2: Register M to Each Cloud C_i. Following many recent registration methods, shapes, initially represented as point clouds, are modeled using a



(a) Step 1 (b) Step 2 (c) Step 3

Fig. 1. Schematic representation of the registration algorithm. Step 1: (a) Three point clouds $C_{1,2,3}$ (solid contours) and initial estimate of mean shape M (dashed contours). Step 2: (b) Copies $M_{1,2,3}$ of M have been registered to $C_{1,2,3}$. For one point correspondences are denoted by arrows. Step 3: (c) New estimate of shape M, with minimal deformation with respect to clouds $M_{1,2,3}$.

mixture of Gaussians. In this work, all Gaussian kernels are isotropic and have the same size, determined by parameter σ. The density D_i of each cloud at coordinates $\mathbf{x}$ is therefore described by

$$D_i(\mathbf{x}) = \frac{1}{n_i (2\pi)^{d/2} \sigma^d} \sum_{j=1}^{n_i} \exp\left(-(\mathbf{x}-\mathbf{p}_{ij})^T (\mathbf{x}-\mathbf{p}_{ij}) \diagup 2\sigma^2\right) \tag{1}$$

where n_i is the number of points in cloud C_i, d is the spatial dimensionality of the cloud points and $\mathbf{p}_{ij}$ are the coordinates of a point indexed by j in cloud C_i.

As in [5] the L_2 divergence is used as a distance measure between two density functions. For two clouds M_i and C_i with density functions D_m and D_c, this measure is defined as

$$f_{L2}(M_i, C_i) = \int_{\mathbb{R}^d} \left(D_m^2 - 2D_m D_c + D_c^2\right) d\mathbf{x} \tag{2}$$

where $\mathbb{R}^d$ is the spatial domain in which the point-clouds reside. The L_2 divergence is a member of the family of Density Power Divergences [5], which also contains the well-known Kullback-Leibler (KL) divergence. The L_2 divergence is symmetric and (2) can be evaluated in closed form for Gaussian density functions, using the identity:

$$\int_{\mathbb{R}^d} G(\mathbf{x}|\mu_1, \Sigma_1)\, G(\mathbf{x}|\mu_2, \Sigma_2)\, d\mathbf{x} = G(0|\mu_1 - \mu_2, \Sigma_1 + \Sigma_2) \tag{3}$$

where $G(\mathbf{x}|\mu_1, \Sigma_1)$ and $G(\mathbf{x}|\mu_2, \Sigma_2)$ are (multivariate) Gaussian density functions with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2, respectively.

In this work the similarity between M_i and C_i is maximized through the minimization of (2). To this end M is transformed (into M_i) with a thin-plate-spline (TPS) transform [1] with n_ϕ control points $\mathbf{p}_l^\phi, l = 1, \ldots, n_\phi$. For cloud C_i

the optimally deformed mean cloud is given by $M_i = \Phi\Theta_i + M$ where Φ is the $n_m \times n_\phi$ matrix that contains the radial basis functions of the TPS transform and Θ_i is the $n_\phi \times d$ matrix that contains the transformation coefficients. The transformation is regularized with the costs for the deformation of M_i

$$f_{stress}\left(\Theta_i, M\right) = \mathrm{tr}\left(\Theta_i^T \Phi^T \Phi \Theta_i\right) \tag{4}$$

where tr stands for the matrix trace. This regularization prevents large deformations from M and thereby preserves shape similarity and meaningful point-point correspondence between the clouds M_i. f_{stress} is a function of M because matrix Φ is a function of M. For all clouds combined, the function to be optimized is defined as the sum

$$F\left(\Theta_1, \ldots, \Theta_N, M\right) = \sum_{i=1}^{N} \left[f_{L2}\left(M_i, C_i\right) + \lambda f_{stress}\left(\Theta_i, M\right)\right] \equiv F_{L2} + \lambda F_{stress} \tag{5}$$

where λ is a regularization weight and $f_{L2}\left(M_i, C_i\right)$ is a function of M and Θ_i, because M_i is a function of M and Θ_i. This function can be minimized directly using iterative methods, such as a quasi-Newton optimization. However, this is a costly optimization, due to the large number of parameters $(n_\phi \cdot d \cdot N)$ and the necessary update of matrix Φ as a function of M. However, by keeping M constant, the cost functions within the sum have no shared optimization variables and can be minimized separately. This the key novelty of our work and of crucial importance to subsequence parallelization.

To account for misalignments, f_{L2} will also include rigid transformations for all clouds C_i. Because the clouds M_i do not deform (only move) with respect to M and f_{stress} can be kept as in (4).

Step 3: Update the Current Estimate of M by Computing the Mean of M_i. Keeping M constant during the minimization of (5) prohibits finding the global minimum of (5). Therefore M needs to be updated separately. With M_i constant, M only affects the term that describes the total deformation costs $F_{stress} = \sum_{i=1}^{N} \mathrm{tr}\left(\Theta_i^T \Phi^T \Phi \Theta_i\right)$. From $\Phi\Theta_i = M_i - M$ follows that

$$F_{stress} = \sum_{i=1}^{N} \mathrm{tr}\left(\left(M_i - M\right)^T \left(M_i - M\right)\right) = \sum_{i=1}^{N} \sum_{j=1}^{n_m} \left\|\mathbf{p}_j^{m_i} - \mathbf{p}_j^{m}\right\|^2$$

where $\mathbf{p}_j^{m_i}$ and $\mathbf{p}_j^{m}$ are the points with index j in cloud M_i and M, respectively. Therefore, F_{stress} is minimal when $\mathbf{p}_j^{m}$ is the mean of $\mathbf{p}_j^{m_i}, i = 1, \ldots, N$, thus the optimal estimate of the mean shape is given by $M \leftarrow \frac{1}{N} \sum_{i=1}^{N} M_i$. This inexpensive step takes care of the coupling of the clouds. Note that this simple mean computation is only possible because of the implicit correspondence between all deformed mean shapes, which is particular for the proposed algorithm.Θ_i is updated using a linear least-squares estimate.

Step 4: Test for Convergence. If converged, continue, otherwise go to step 2.

Step 5: Transform C_i to the Mean Cloud Using the Correspondence between M_i and M. One should note that minimizing $F\left(\tilde{\Theta}_1, \ldots, \tilde{\Theta}_N, M\right)$ as in (5) results in sets of corresponding points M_i. However, the much denser clouds C_i have not deformed and, thus, are not registered to M (See Figure 1c). To do so, all point clouds C_i are rigidly transformed using $\mathbf{r}_i$ and $\mathbf{t}_i$ and deformed towards M by computing the inverse TPS transforms $C_i = \Phi_i^C \Theta_i + \hat{C}_i$ where the matrix with TPS basis functions Φ_i^C and the deformed point cloud $\hat{C}_i$ have to be estimated for a given set of transformation parameters Θ_i. This can be done using an iterative procedure. For an accurate registration, all clouds $\hat{C}_i$ represent a surface of the same shape. These registered surfaces can then be used for a *dense* correspondence estimate between the clouds C_i.

3 Experiments

The experiments in this section evaluate both the accuracy and the precision of the proposed registration algorithm and compare these to the registration method in [5]. The chosen application is the registration of wrist bones to establish a dense correspondence of points on the bone surfaces of different individuals. Specifically, the focus is on the scaphoid, lunate and hamate bones (See Figure 3). The 3×48 bone surfaces are represented as triangulated surfaces with vertices V_i at a sampling density of approximately 0.14 vertices/mm^2 (voxel size $0.3 \times 0.3 \times 0.3$ mm^3), typically yielding 1.8×10^4 (SD 3.4×10^3) points per bone surface. The bones are coarsely aligned by ensuring the same scan orientations and by translating the centers of gravity of each bone to the origin. Optimal parameter settings (See Section 2) were experimentally determined for this data as $\sigma = 0.6$ mm, $n_m = 1000$, $\lambda = 10^{-6}$ and $n_\phi = 600$. Each bone is represented by a point cloud C_i with a subset of n_m vertices of V_i.

In all experiments the registration accuracy E_{acc} and precision E_{prec} are evaluated on the transformed bone surfaces i.e. after the vertices V_i are deformed towards M as in Step 5. This allows comparable results for different sample densities. The resulting clouds are denoted as $\hat{V}_i$. E_{acc} is defined as the average (over shapes) norm of the average (over points) *signed* point-to-plane distance between all pairs of clouds $\hat{V}_i$ and $\hat{V}_j$, evaluated on points that are not in C_i (thus not used for the registration). E_{prec} is the average *absolute* point-to-plane distance between all pairs of clouds $\hat{V}_i$ and $\hat{V}_j$, again using points not in C_i. E_{acc} reflects the presence of a bias, while E_{prec} reflects the remaining matching error.

3.1 Robustness to Initial Mean Cloud Estimate

The initial estimate of M has a strong influence on F_{stress} during the first iterations. The robustness of the algorithm was tested by using four different initial estimates of M: (I) Points drawn randomly from the 0-level set of the average signed distance transform (SDT) as in Section 2, Step 1. (II) Points drawn randomly from all clouds to be registered (rand). (III) Points drawn

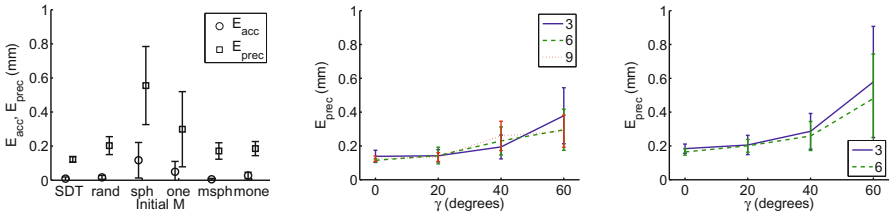


Fig. 2. (a) Registration accuracy and precision for four initial estimates of the mean shape M: (I) SDT (II) rand, (III) sph, (IV) one. For 'sph' and 'one' also the results of a multiscale registration are shown (msph, mone). (b) Registration precision for the proposed algorithm and (c) for the algorithm from [5] for increasing γ and N. (a-c) The errorbars denote standard deviations.

randomly from a sphere with radius equal to the mean standard deviation of the point cloud coordinates (sph). (IV) One of the point clouds (one). For each of these initial point clouds a random selection of $N = 3$ three clouds were registered, repeated with 10 drawings for each of the 3 bone types.

The registration accuracies E_{acc} and precisions E_{prec} in Figure 2a show that initializing M by taking points from the SDT gives the most accurate results, closely followed by a random selection of points from the non-registered point clouds. With a sphere or a single shape the registration algorithm often converged in local minima due to a lack of overlap between the kernels of M and the kernels of C_1, C_2 and C_3 for the current, small, choice of $\sigma = 0.6$ mm. A multiscale approach where σ decreases from $\sigma = 3$ to $\sigma = 0.6$ mm solves this as depicted by the last two results of Figure 2a.

3.2 Robustness to Initial Shape Alignment

When more shapes are present, rotated over a random angle, it is more likely that a shape that has a rotation 'in between' improves the convergence of the algorithm. Therefore N previously aligned shapes were rotated around randomly distributed rotation axes, with angles randomly sampled from $[-\gamma, \gamma]$, for different values of N and γ. The initial M was obtained as in Section 2, step 1. The clouds were registered using both the proposed algorithm and the algorithm in [5]. The experiment was repeated 30 times for each combination of the following settings: $N \in \{3, 6, 9\}$ randomly selected shapes (10 selections for each of the three bone types) and $\gamma \in \{20, 40, 60\}$ degrees.

E_{acc} for the proposed method and the method from [5] were all in the order of 10^{-4} mm, except at $\gamma = 60°$. Here E_{acc} was approximately 0.07 mm for $N = 3$ shapes. In the latter case, both algorithms converged in a local minimum with large shape deformations. E_{prec} is shown in Figure 2b for the proposed method and in Figure 2c for the method in [5]. For $\gamma = 0°, 20°$ both methods perform equally well. For $\gamma = 40°, 60°$ the proposed method is more precise than the

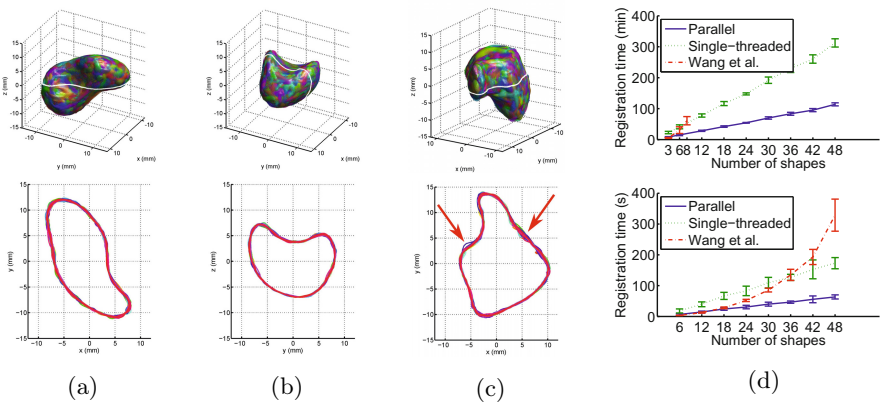


Fig. 3. Bone surfaces and intersections after registration of $N = 48$ (a) scaphoids, (b) lunates and (c) hamates. The white lines on the surfaces show the locations of the contours. (d) Average registration times (and standard deviations) in minutes for the registration of $N \in \{6, 12, \ldots, 48\}$ shapes, for the proposed method parallel on three cores and single-threaded and the method in [5] for $n_m = 1000$ (top) and $n_m = 100$ (bottom).

method of [5]. We hypothesize that the method of [5] is slightly more susceptible to local minima, due to the larger number of concurrently optimized parameters. Furthermore, an increase in N decreases the registration error for large angles γ. This is mainly due to the denser sampling of poses between $-\gamma$ and γ. For [5] experiments with 9 shapes did not succeed due to a lack of computer memory.

3.3 Feasibility of Large Data Set Registration

To investigate the feasibility of registering large datasets, an increasing number of randomly selected sets of bones were registered: $N \in \{6, 12, \ldots, 48\}$. Registrations were performed for each type of bone and for three different shape set selections. Experiments were performed on a computer with an Intel Xeon processor at 2.67GHz with 6.0 GB of RAM memory. The method was implemented in MATLAB R2010b, from The Mathworks, Inc.

Example registration results are shown in Figure 3 for $N = 48$ bones. For the scaphoid and lunate, all contours are aligned. For the hamate, however, small misalignments can be observed on the left and right of the protrusion of the bone (see arrows). This is due to the large shape variations of the protrusion, combined with the TPS interpolation of the shape surfaces. For correspondence estimates between the surfaces, however, these small misalignments do not form a problem. For 12 shapes and more, the registration accuracy E_{acc} and precision E_{prec} do not depend on the number of shapes and are $E_{acc} \approx 0.00$ mm and $E_{prec} \approx 0.12$ mm, with negligible standard deviations.

Figure 3d shows that the registration time increases linearly with an increasing number of shapes, using the proposed method, while it increases approximately quadratically using the method proposed in [5]. For $n_m = 1000$, registering more than 8 shapes, using the method of [5] was not possible, as the data did not fit in the availabe RAM. Therefore also results for $n_m = 100$ are shown. The parallelization of the registration using three cores shortens the registration time with approximately a factor 2.8, an efficient parallelization.

4 Discussion

In this paper, a method was presented for the non-rigid registration of a large number of shapes, whose surfaces are represented by point clouds. Experiments showed that the proposed algorithm indeed can register large numbers of point clouds with high accuracy and precision with modest hardware demands. The registration time increases linearly with the number of shapes N. Furthermore, the registration accuracies and precisions are similar to the method of [5], which itself was compared favorably to other state of the art methods, e.g. [6].

Although, as in [2] both algorithms evolve a mean cloud that is only mildy constrained, the proposed method does not suffer from the instabilities in [2]. This is because the proposed method by definition does not need the assumption that the 'forward' and 'backward' thin-plate spline transform are exactly the same and because of an improved similarity measure. Interesting follow-up research is if current based methods, e.g. [4] also allow group-wise registration with a closed-form mean estimate as in step 3.

As shown in Figure 3, a much denser correspondence than the $n_m = 1000$ points used for registration can easily be obtained from the registered surfaces. Furthermore, taking the linear increase of registration times into account, combined with the parallelization of the non-rigid registration step, one could register almost 600 surfaces in a day. Note that such a dataset is not easily obtained.

References

1. Bookstein, F.L.: Principal warps - thin-plate splines and the decomposition of deformations. IEEE Trans. on Patt. An. and Mach. Intell. 11(6), 567–585 (1989)
2. Chui, H., Rangarajan, A., Zhang, J., Leonard, C.M.: Unsupervised learning of an atlas from unlabeled point-sets. IEEE Trans. on Patt. An. and Mach. Intell. 26(2), 160–172 (2004)
3. Joshi, S., Davis, B., Jomier, M., Gerig, G.: Unbiased diffeomorphic atlas construction for computational anatomy. Neuroimage 23, S151–S160 (2004)
4. Myronenko, A., Song, X.: Point set registration: Coherent point drift. IEEE Trans. on Patt. An. and Mach. Intell. 32(12), 2262–2275 (2010)
5. Wang, F., Vemuri, B., Syeda-Mahmood, T.: Generalized L2-Divergence and Its Application to Shape Alignment. In: Prince, J.L., Pham, D.L., Myers, K.J. (eds.) IPMI 2009. LNCS, vol. 5636, pp. 227–238. Springer, Heidelberg (2009)
6. Wang, F., Vemuri, B.C., Rangarajan, A., Eisenschenk, S.J.: Simultaneous nonrigid registration of multiple point sets and atlas construction. IEEE Trans. on Patt. An. and Mach. Intell. 30(11), 2011–2022 (2008)