Gravitational Approach for Point Set Registration

Vladislav Golyanik[1,2]
vladislav.golyanik@dfki.de

Sk Aziz Ali[1]
saali@rhrk.uni-kl.de

Didier Stricker[1,2]
didier.stricker@dfki.de

[1] University of Kaiserslautern, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Germany

Abstract

天体动力学

In this paper a new astrodynamics inspired rigid point set registration algorithm is introduced — the Gravitational Approach (GA). We formulate point set registration as a modified N-body problem with additional constraints and obtain an algorithm with unique properties which is fully scalable with the number of processing cores. In GA, a template point set moves in a viscous medium under gravitational forces induced by a reference point set. Pose updates are completed by numerically solving the differential equations of Newtonian mechanics. We discuss techniques for efficient implementation of the new algorithm and evaluate it on several synthetic and real-world scenarios. GA is compared with the widely used Iterative Closest Point and the state of the art rigid Coherent Point Drift algorithms. Experiments evidence that the new approach is robust against noise and can handle challenging scenarios with structured outliers.

1. Introduction

We enter the era of pervasive 3D technologies. This development is accompanied by a clear tendency: as 3D acquisition devices become ubiquitous, the need for reliable point cloud processing algorithms including those for alignment grows. Point set registration is an actively researched area with applications in different domains of computer science and engineering such as shape recognition, action transfer, 3D reconstruction and animation, computer-aided design, industrial quality control and robotics.

In the point set registration problem, the objective is to find an optimal alignment between two (generally several) point sets and recover transformation parameters. Thereby an optimal transformation of the *template* point set to the *reference* point set as well as point affiliations are sought, so that both point sets coincide in an optimal way. Several optimality criteria are possible. In the case of *rigid* point set registration, a transformation is entirely described by parameters of the rigid body motion, i.e. rotation and translation applied to all points simultaneously. Often scaling is

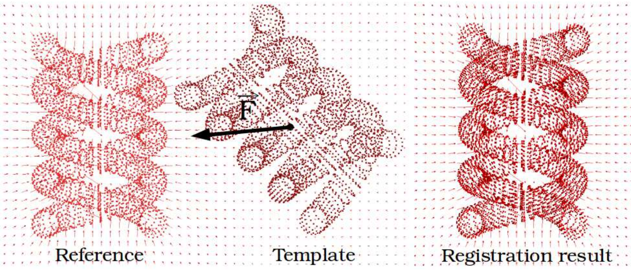
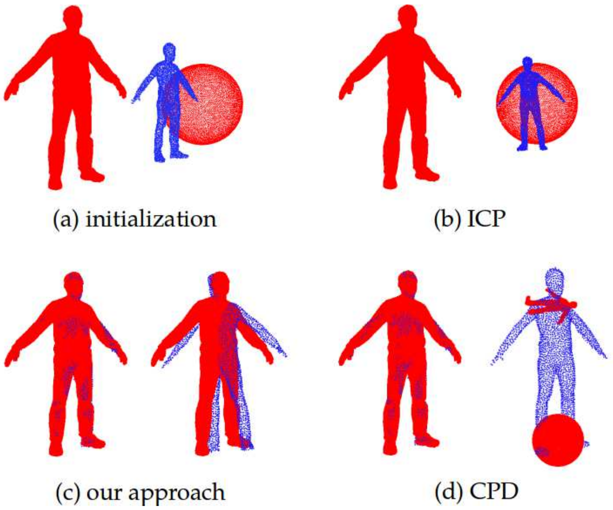


Figure 1: Point set registration with the Gravitational Approach: template moves in the gravitational field induced by the reference. Coordinates of the template points are individually updated by solving equations of particle motion in a viscous medium, whereupon rigidity constraints are applied. Left: initial misalignment of the helix point sets [8] and the induced gravitational field; right: registration result after 150 iterations.



(a) initialization

(b) ICP

(c) our approach

(d) CPD

Figure 2: Registration results of ICP [6], CPD [24] and our approach on real data with introduced clustered outliers. (a) Initialization; template (shown in blue) is located between the reference human scan and outliers arranged as a sphere (shown in red). (b) ICP registration result — the algorithm is trapped into a local minimum; (c) GA registration results (left: an optimal parameter; right: a suboptimal parameter); (d) CPD registration results (left: an optimal parameter; right: a suboptimal parameter).

also added to this parameter set, though strictly speaking it makes registration affine. In the non-rigid case, a transformation is described more generally by a point displacement field, since non-rigid deformations imply individual displacements for every point. Therefore, the problem becomes highly ill-posed and regularization of the displacement field is required to obtain a plausible solution.

Point sets can be noisy, contain clustered outliers and differ significantly; not every template's point may possess a valid correspondence in the reference and vice versa. Thus, 3D acquisition devices often output noisy point clouds. Additional information (normals, manifold structure, point topology) is not available. Note that algorithms incorporating such additional information constitute a separate class of shape registration methods.

The Iterative Closest Point (ICP) [6] is an early iterative point set registration algorithm. On every iteration, it minimizes over all template points the mean squared error of distances to corresponding closest points. Therefore, nonlinear optimization algorithms are used. The algorithm is known to perform weakly in presence of outliers and is strongly dependent on initial alignment of the point sets. Through its simplicity and despite the disadvantages, it is one of the most widely used algorithms. The original ICP engendered a bunch of descendants differing in distance cost functions and optimization techniques [11, 27]. ICP is suitable for arbitrary dimensions.

Point set registration was improved by probabilistic approaches. Robust point matching (RPM) [13] performs better on noisy point sets, since correspondences are assigned softly with probabilities. This can be interpreted as generalization of the binary 0-1 correspondence assignment. Later, probabilistic approaches were formulated in terms of the Gaussian Mixture Models (GMM), where template points determine positions of the GMM centroids and reference points serve as covered data. To find transformation parameters, Expectation-Maximization (EM-) algorithm for the likelihood function optimization is used. Several methods introduce an extra term to cope with outliers explicitly [37, 24] while making an additional assumption on the type of the noise distribution. The idea of probabilistic assignments was further evolved in the Coherent Point Drift (CPD) [24]. The authors provide a closed-form solution to the M-step of the EM-algorithm which makes the algorithm applicable in multidimensional cases. CPD was extended in [35] with an additional term for outlier modelling. A unified framework for probabilistic point set registration with a closed-form expression was shown in [17].

Tsin and Kanade introduced the point set registration technique known as kernel correlation (KC) [34] further extended with GMM in [18]. The method is a *multiply-linked* algorithm, i.e. all reference points influence all template points. Being less sensitive to noise compared to ICP, KC is not widely used due to the high computational complexity. A subclass of the probabilistic algorithms takes advantage of particle filters [21, 28]. Only the latest particle filtering approach to point set registration [28] can be seen as a general-purpose point set registration algorithm, as it makes no assumptions on the point set density and is more accurate by a moderate computational cost. This method was developed to cope with partial rigid registrations. However, it is not able to resolve scaling and includes ICP as an intermediate step. Another class of methods developed to overcome the shortcomings of ICP constitute spectral methods (e.g. [29]). Operating on proximity matrices with distance measures between the points, they are generally computationally expensive. This circumstance narrows their scope.

In this paper we present a novel physics based rigid point set registration algorithm — the Gravitational Approach (GA). Inspired by astrodynamics, it does not possess a direct ancestor. We couple N-body simulation with rigid body dynamics. In an N-body simulation, future trajectories of n bodies with the specified initial coordinates, masses and velocities are estimated. Thus, the main idea of GA consists in dynamics modelling of rigid systems of particles under gravitational forces in a viscous medium (see Fig. 1). Particle movements are expressed by differential equations of Newtonian mechanics. In our model, every point from both a reference and a template point set is treated as a particle with its own position, velocity and acceleration (in text, the terms point and particle are used interchangeably). GA is a multiply-linked algorithm, as all template points are moving in the superimposed force field induced by the reference. To impose rigidity constraints, laws of rotational and translational motion of rigid bodies are employed. To resolve rotation, a formulation based on singular value decomposition (SVD) for finding an optimal rotation matrix is used [19, 23]. This method was extensively applied to recover rotation matrixes in computer vision applications [39, 24, 26, 7]. Similar to ICP and CPD, GA can operate in multiple dimensions. GA also supports a basic form of including prior correspondences into the registration procedure by assigning different masses to regular and matching points. Embedding the prior correspondences is not trivial, especially in the case of probabilistic approaches Thus, for this class of algorithms it was only recently shown in the literature, for the non-rigid case [14].

Motivation for a conceptually new registration algorithm is manifold. Firstly, because of a new formulation, an algorithm with unique properties among point set registration algorithms is obtained. Thus, GA can take an initial velocity of the template as a parameter. There are application scenarios where such velocity can be estimated (e.g. from an optic flow between frame pairs in simultaneous localization and mapping (SLAM) systems). Secondly, registration algorithms are sensitive to noise and improving performance of the rigid point set registration is a fundamental task in computer vision. In fact, GA can perform better than the state of the art CPD in some scenarios with structured outliers and in presence of noise (see Fig. 2 and Sec. 4 for a detailed description). Thirdly, we place parallelizability of operations to the foreground, because many existing methods (e.g CPD and KC) contain a significant portion of serial code. Fourthly, in GA point set registration is formulated as an energy minimization. In this paper, one possible minimization through forward integration is proposed, but many other minimization techniques can be tried out (non-linear optimization of the gravitational potential energy function, simulated annealing for a globally optimal solution). Besides, a new algorithm is, of course, interesting both from the theoretical and practical point of views and can encourage further ideas in the area. To the best of our knowledge, formulating point set registration problem as a modified N-body simulation was not shown in the literature so far. We also did not find any evidence for interpreting the problem as an object moving in a force field, considering early and pre-ICP works in the field [1].

The rest of the paper is organized as follows. In the next section an overview of the classic N-body problem is given. In Sec. 3 we introduce GA and discuss acceleration techniques followed by experiments, discussion and conclusions in Secs. 4 and 5 respectively.

2. N-body problem

In N-body problems, numerical solutions to the motion equations for n particles interacting gravitationally are sought [10]. First N-body problems emerged in astrophysics where movements of celestial bodies under influence of other celestial bodies are studied. A superimposed gravitational field induced by individual particles exerts gravitational force $\vec{F}$ to every particle $i \in \{1, \dots, n\}$ in the system [33]:

$$\vec{F}_i = -Gm_i \sum_{i \neq j} \frac{m_j(\vec{r}_i - \vec{r}_j)}{\|\vec{r}_i - \vec{r}_j\|^3} - \nabla \phi_{ext}(\vec{r}_i) \ , \tag{1}$$

where G is the gravitational constant determining the gravitational strength between two bodies of unit masses separated by a unit distance, m_i, m_j and $\vec{r}_i$, $\vec{r}_j$ are particle masses and position vectors respectively, ϕ_{ext} is an external gravitational potential, ∇ denotes the gradient operator and $\|.\|$ denotes the L^2-norm. Newton's second law of motion relates the force exerted on a particle with its acceleration. Thus, an N-body problem can be described by the second-order ordinary differential equations (ODE):

$$\ddot{\vec{r}}_i = \frac{\vec{F}_i}{m_i}, \tag{2}$$

where $\ddot{\vec{r}}_i$ is a particle's acceleration. There exists a unique solution to the system in Eq. 2 as long as initial conditions are specified, i.e. an initial position and velocity of every particle. A solution is obtained by means of numerical integration, since no analytical solution for $n > 3$ exists. Depending on the assumptions and objectives, N-body simulations can be classified as *collisional* or *collisionless*. While the former class allows the particles (the bodies) to merge, the latter prohibits merging. For more details on classical N-body problems the reader may refer to [2, 3].

3. Gravitational Approach

In a point set registration problem, two D-dimensional point sets $\mathbf{X}_{N \times D} = (X_1, \dots, X_N)^T$ (a reference) and $\mathbf{Y}_{M \times D} = (Y_1, \dots, Y_M)^T$ (a template) are given. We search parameters of the rigid transformation, i.e a tuple $(\mathbf{R}, \mathbf{t}, \mathbf{s})$ which optimally aligns the template point set to the reference point set.

Since we target an efficient point set registration, we adopt the N-body problem while abstracting from a realistic physical model and alter the simulation objective. Specifically, the following assumptions and modifications are made:

i. every point represents a particle with a mass condensed in an infinitely small area of space

ii. a reference $\mathbf{X}$ induces a constant inhomogeneous gravitational field

iii. particles Y_i move in the gravitational field induced by the reference and do not affect each other

iv. $\mathbf{Y}$ moves rigidly, i.e. transformation of the template particle system is described by the tuple $(\mathbf{R}, \mathbf{t}, \mathbf{s})$

v. a collisionless N-body simulation is performed, since the number of particles cannot be changed according to the problem definition

vi. astrophysical constants (e.g. G) and units are considered as algorithm parameters

vii. a portion of kinetic energy is dissipated and drained from the system — the physical system is not isolated.

Modification (ii) reflects that the reference point set remains idle. Physically, it is said to be fixed by an external force. Modification (vii) — introduction of an *energy dissipation* or *viscosity* term — arises through the physical analogy of movement with friction in a viscous medium (gas, fluid), whereby a part of the kinetic energy transforms to heat.

In GA, potential and kinetic energy are continuously redistributed. Second-order ODEs in Eq. (2) without an external stimulus describe endless oscillatory phenomena. If a part of the kinetic energy which has been converted from the potential energy under the influence of the gravitational field is dissipated, the system gradually converges to its most stable state with locally minimal potential energy. This state corresponds to a locally optimal solution to the point set registration problem. Moreover, the viscosity term

[1] as a starting point for pre-ICP works we used [6]

is necessary to assure the algorithm's convergence. Without viscosity, it would be difficult to refine the solution, as $\mathbf{Y}$ may have a high speed close to a local minimum.

We find the force exerted on a particle Y_i by all particles of the reference $\mathbf{X}$:

$$\vec{F}_{Yi} = -Gm^{Yi} \sum_{j=1}^{N} \frac{m^{Xj}}{\|r^{Yi} - r^{Xj}\|^2} \hat{\mathbf{n}}_{ij} \ , \tag{3}$$

where m^{Yi} (m^{Xi}) and r^{Yi} (r^{Xi}) denote mass and absolute coordinates of a particle Y_i (X_i) respectively and $\hat{\mathbf{n}}_{ij} = \frac{(r^{Yi} - r^{Xj})}{\|r^{Yi} - r^{Xj}\|}$ is a unit vector in the direction of force. Note that we depart from the notations used in Sec. 2, as we deal with two non-overlapping particle sets. Besides, instead of position vectors absolute point coordinates are used. The gravitational force in Eq. (3) can lead to a singularity during a collisionless simulation, since two or more particles can be pushed infinitely close to each other. The singularity can be avoided by revising gravitational interaction at small scales. Thus, we introduce the softening length ϵ — a threshold distance, below which gravitational interaction does not increase severely. The force acting on a particle takes the form of a cubic spline [33, 1]:

$$\vec{F}_{Yi} = -Gm^{Yi} \sum_{j=1}^{N} \frac{m^{Xj}}{(\|r^{Yi} - r^{Xj}\|^2 + \epsilon^2)^{3/2}} \hat{\mathbf{n}}_{ij} \ . \tag{4}$$

The dissipation term is expressed by a drag force acting in the opposite to the particle's velocity direction with a magnitude proportional to its speed:

$$\vec{F}^{d}_{Yi} = -\eta \, v^{Yi}, \tag{5}$$

where the dimensionless constant parameter η jointly reflects properties of the particle Y_i and the viscous medium. Thus, the resultant force exerted on a particle Y_i reads

$$\vec{\mathbf{f}}_{Yi} = \vec{F}_{Yi} + \vec{F}^{d}_{Yi} \ . \tag{6}$$

Using Euler's method for second order ODEs we perform forward integration, i.e. solve the system in Eq. (2) and get updates for an *unconstrained* velocity and displacement of every particle Y_i:

$$\vec{v}^{t+1}_{Yi} = \vec{v}^{t}_{Yi} + \Delta t \, \frac{\vec{\mathbf{f}}_{Yi}}{m^{Y_i}} \ \text{(velocity)}, \tag{7}$$

$$\vec{d}^{t+1}_{Yi} = \Delta t \ \vec{v}^{t}_{Yi} \ \text{(displacement)}. \tag{8}$$

Unconstrained velocities and displacements can be combined into the velocity and displacement field matrices $\mathbf{V}_{M \times D}$ and $\mathbf{D}_{M \times D}$:

$$\mathbf{V} = \begin{bmatrix} \vec{v}^{t+1}_{Y1} & \vec{v}^{t+1}_{Y2} & \ldots & \vec{v}^{t+1}_{Ym} \end{bmatrix}^T, \tag{9}$$

$$\mathbf{D} = \begin{bmatrix} \vec{d}^{t+1}_{Y1} & \vec{d}^{t+1}_{Y2} & \vdots & \vec{d}^{t+1}_{Ym} \end{bmatrix}^T. \tag{10}$$

$\mathbf{V}$ and $\mathbf{D}$ are subjects to further regularization which depends on the type of point set registration.

3.1. Rigidity constraints

In the rigid case, rigidity constraints on the displacement field $\mathbf{D}$ must be imposed and rigid body physics takes effect. **Resolving translation.** Since distances between points are preserved, several simplifications can be carried out. First, the resultant force exerted on a rigid body is equal to the sum of the forces exerted on individual particles:

$$\vec{F}^{t+1}_{\mathbf{Y}} = \sum_{i=1}^{M} \vec{\mathbf{f}}^{t}_{Yi}. \tag{11}$$

Second, the resultant velocity changes depending on the action of the resultant force on the total mass of the template $m^{\mathbf{Y}}$ per unit of time as

$$\vec{v}^{t+1}_{\mathbf{Y}} = \vec{v}^{t}_{\mathbf{Y}} + \Delta t \, \frac{\vec{F}^{t+1}_{\mathbf{Y}}}{m^{\mathbf{Y}}}. \tag{12}$$

From the resultant velocity $\vec{v}^{t+1}_{\mathbf{Y}}$, the resultant translation can be computed as

$$\vec{\mathbf{t}}^{t+1} = \Delta t \ \vec{v}^{t+1}_{\mathbf{Y}}. \tag{13}$$

Resolving scale. We find a scale $\mathbf{s}$ in the least-square sense. $\mathbf{s}$ relates the current position of a template $\mathbf{Y}_t$ with the predicted position $\mathbf{Y}_{t+1} = \mathbf{Y}_t + \mathbf{D}$ as

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t \mathbf{s}. \tag{14}$$

Suppose $\hat{\mathbf{Y}}_t$ and $\hat{\mathbf{Y}}_{t+1}$ are column vectors of length DM with vertically stacked entries of $\mathbf{Y}_t$ and $\mathbf{Y}_{t+1}$ respectively. In that case, the following proposition holds.

Proposition 1. *In Eq.* (14) *the optimal scaling factor* $\mathbf{s}$ *in the least-squares sense is equal to the ratio of two vector dot products* $\frac{\hat{\mathbf{Y}}_t^T \cdot \hat{\mathbf{Y}}_{t+1}}{\hat{\mathbf{Y}}_t^T \cdot \hat{\mathbf{Y}}_t}$.

A proof can be found in the supplementary material.
Resolving rotation. Rigorously, rotation can be inferred from a *torque* acting on a rigid body. A torque (a moment of force) is a physical quantity reflecting the tendency of the force to change the angular momentum of the system, i.e. to rotate an object. Resolving rotation rigorously applying physics of rotational motion introduces to GA an additional parameter ω (angular velocity) and generates a bunch of new intermediate quantities (see supplementary material for details on the rigorous rotation resolving).
Thus, computing R using torque requires several non-trivial steps. Instead, a different method is used in this paper. To recap the initial conditions, in every iteration the starting and final position vectors of M points are given and the task is to find a rotation matrix which optimally aligns both vectors. This can be efficiently addressed by solving

a corresponding *Generalized Orthogonal Procrustes Problem*. The disadvantage is the loss of ω, since no angular acceleration from previous iterations is considered. The solution in a closed-form is given in Lemma 1. It resembles the Kabsch algorithm [19] and is provided without a proof.

Lemma 1. *Given are point matrices* $\mathbf{Y}$ *and* $\mathbf{Y}_D = \mathbf{Y} + \mathbf{D}$. *Let* μ_Y *and* μ_{YD} *be the mean vectors of* $\mathbf{Y}$ *and* $\mathbf{Y}_D$ *respectively,* $\hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_Y^T$ *and* $\hat{\mathbf{Y}_D} = \mathbf{Y}_D - \mathbf{1}\mu_{YD}^T$ *point matrices centered at the origin of the coordinate system and* $\mathbf{C} = \hat{\mathbf{Y}_D}^T\hat{\mathbf{Y}}$ *a covariance matrix. Let* $\mathbf{U}\mathbf{S}\hat{\mathbf{U}}^T$ *be SVD of* $\mathbf{C}$. *Then the optimal rotation matrix* $\mathbf{R}$ *reads*

$$\mathbf{R} = \mathbf{U}\Sigma\hat{\mathbf{U}}^T, \tag{15}$$

where $\Sigma = diag(1, \dots, sgn(|\mathbf{U}\hat{\mathbf{U}}^T|))$.

The covariance matrix C has dimensions 3×3. Thus, the sequential code portion dedicated to the (parts of) SVD computation is negligible.

Finally, having resolved the translation, scale and rotation it is possible to update the template's pose as

$$\mathbf{Y}_{t+1} = \mathbf{s}\mathbf{Y}_t\mathbf{R} + \mathbf{t}. \tag{16}$$

Note that the center of mass of $\mathbf{Y}_t$ must coincide with the origin of the coordinate system for the rotational update.

The Gravitational Approach is summarized in Alg. 1. As an optional parameter a non-zero template's velocity $\vec{v}_{\mathbf{Y}}^0$ can be provided. Following the rigorous approach to resolve rotation would require an additional parameter ω as well as a modification of line 7 in Alg. 1. As stated so far, GA has complexity $\mathcal{O}(MN)$, since every particle of the reference perturbs every template's particle. The stopping criterion is formulated in terms of the difference in the gravitational potential energy (GPE) which reads

$$E(\mathbf{R}, \mathbf{t}, \mathbf{s}) = -G \sum_{i,j} \frac{m^{Yi}\, m^{Xj}}{\|\mathbf{R}\, r^{Yi}\, \mathbf{s} + \mathbf{t} - r^{Xj}\| + \epsilon}. \tag{17}$$

See supplementary material for the detailed explanation of the GPE expression.

3.2. Acceleration techniques

Acceleration techniques from both areas of N-body simulations and point set registration can be adopted for GA. They enable a drop in computational complexity to at least $\mathcal{O}(N \log M)$ as well as a speedup in a corresponding complexity class in terms of the number of operations.

Various techniques to accelerate N-body simulations were developed in the past three decades [33]. Ahmad-Cohen (AC) neighbour scheme employs two time scales for each particle [4]. Thereby, force evaluations for neighbouring particles occur more frequently than for distant particles.

Algorithm 1 Gravitational Approach

```
Input: a reference $\mathbf{X}_{N\times D}$ and a template $\mathbf{Y}_{M\times D}$
Output: parameters $(\mathbf{R}, \mathbf{t}, \mathbf{s})$ aligning $\mathbf{Y}$ to $\mathbf{X}$ optimally
Parameters: $G \in (0, 1]$, $\epsilon \in (0; 0.5)$, $\eta \in (0; 1]$, $m^{Xj}$, $m^{Yi}$,
    $j \in \{1, \dots, N\}$, $i \in \{1, \dots, M\}$, $\vec{v}_{\mathbf{Y}}^0$, $\rho_E$, $\Delta t$, $K$
1:  Initialization: $\mathbf{R} = \mathbf{I}$, $\mathbf{t} = \mathbf{0}$, $\mathbf{s} = \mathbf{1}$, $E_{curr} = E(\mathbf{R}, \mathbf{t}, \mathbf{s})$,
    $E_{prev} = 0$
2:  while $|E_{curr} - E_{prev}| < \rho_E$ do
3:      update force $\vec{\mathbf{f}}_{Yi}$ for every particle $Y_i$ (Eqs. (4)–(6))
4:      update velocity and displacement matrices $\mathbf{V}$ and $\mathbf{D}$
        (Eqs. (7)–(10) )
5:      compute translation $\mathbf{t}_k$ according to Eqs. (11)–(13)
6:      compute scale $\mathbf{s}_k$ as stated in Proposition (1)
7:      update rotation $\mathbf{R}_k$ (Eq. (15))
8:      $\mathbf{Y}_{t+1} = \mathbf{s}_k\mathbf{Y}_t\mathbf{R}_k + \mathbf{t}_k$ (Eq. (16))
9:      $\mathbf{R} = \mathbf{R}\mathbf{R}_k$, $\mathbf{t} = \mathbf{t} + \mathbf{t}_k$, $\mathbf{s} = \mathbf{s}\mathbf{s}_k$ (optimal parameters)
10:     $E_{prev} = E_{curr}$, update $E_{curr}$ according to Eq. (17)
11:     if the current iteration number $k$ exceeds $K$ then
12:         break
13:     end if
14: end while
```

For smaller time steps, contributions from the distant points are approximated. Various strategies for neighbourhood selection were proposed [3]. Though the AC scheme allows to achieve a speedup, the complexity class remains $\mathcal{O}(n^2)$. Barnes and Hut [5] introduced a recursive scheme for the force computation based on the space subdivision and particle grouping in an octree. The algorithm achieves $\mathcal{O}(n \log n)$ for an N-body simulation. Adopting it for GA will decrease the complexity to $\mathcal{O}(M \log N)$. Fast multipole methods (FMM) [15] also employ hierarchical space decomposition, but additionally take advantage of multipole expansions. Thus, adjacent particles in the near-field tend to accelerate similarly under forces exerted by particles in the far field. This class of algorithms exploits the idea of rank-deficiency of the $n \times n$ interaction matrix considering the nature of the far-field interactions. This results in an $\mathcal{O}(n \log n)$ algorithm, whereby a multiplicative constant depends on the approximation accuracy. An $\mathcal{O}(n)$ algorithm is also possible, but requires a lot of additional effort accompanied by an increase in the multiplicative constant. When adopting FMM for GA, the complexity can theoretically drop to $\mathcal{O}(M \log N)$ and $\mathcal{O}(M + N)$ respectively. Interestingly, CPD employs the Fast Gauss Transform (FGT) to approximate sums of Gaussian kernels in $\mathcal{O}(M + N)$ time. FGT was developed by Greengard and Strain following the principles of FMM [16].

Common acceleration techniques from the area of point set registration can be applied to GA. Firstly, both the reference and the template can be subsampled. Assume s_X and s_Y are corresponding subsampling factors. The speedup amounts to $\sim s_X s_Y$ is this case. Subsampling has to be used with caution, since it can cause loss of information. Secondly, a coarse-to-fine strategy can be applied — starting on a rough scale, the solution is refined while involving more and more points for the registration. Thirdly, dedicated data structures such as kd-tree for nearest neighbour search can be used. Such data structures, hierarchically reordering the points according to their spatial positions also find their application in tree codes and FMM.

Furthermore, parallel hardware can be used to speedup GA, since the algorithm is inherently data- and task-parallel with the portion of parallel code $> 99\%$. Though, with decrease in computational complexity of a GA variant (e.g. from $\mathcal{O}(MN)$ to $\mathcal{O}(M \log N)$), memory complexity and the effort to parallelize the algorithm may increase. In 2007, Nyland *et al.* reported a fiftyfold speedup of the GPU implementation of an all-pairs N-body simulation compared to a tuned serial implementation [25]. Later, a first GPU implementation of the Barnes-Hut octree was presented. It allows to simulate interactions of $5 \cdot 10^6$ particles with 5.2 seconds per time step [9]. A recent tendency is to unify tree codes and FMM with automatic parameter tuning for heterogeneous platforms [38]. Using an efficient implementation, it should be possible to run GA for point sets with 10^7 points on a single GPU in reasonable time.

Apart from the abovementioned techniques, a further one is conceivable for GA. Since only the template point set is moving and its particles do not affect each other, the force field induced by the reference can be precomputed once in a grid. Thereby, the gravitational force field can be sampled with a higher density in the proximity to the reference points. This can be especially advantageous when many templates are registered with the same reference and there is no memory restriction to achieve the desired accuracy. This technique exhibits resemblance with the particle-mesh class of methods in the area of N-body simulations (see e.g. [3]). In the particle-mesh methods, particles interact with each other through a mean force field changing over time. The methods achieve complexity $\mathcal{O}(n \log n)$. In the case of GA, the technique raises the algorithm's memory complexity and preprocessing time, but reduces computational complexity to $\mathcal{O}(M)$, since the gravitational force field remains constant and does not need to be recomputed.

4. Evaluation

In this section we focus on the qualitative evaluation of the first GA implementation and compare it with ICP and CPD in synthetic and real-world scenarios. The Matlab implementations of the ICP and CPD algorithms are taken from [20] and [22] respectively. GA is implemented in C++ and runs on a system with 3.5 GHz Intel Xeon E5-1620 processor and 32 GB RAM.

Experiments on synthetic data. In the first experiment we compare ICP, CPD and GA in a registration scenario with the Bunny point cloud from the Stanford 3D Scanning Repository [31]. We copy the downsampled version of the Bunny (1889 points), translate it and change its orientation (angle $\phi \in [0; \frac{\pi}{2}]$). The resulting point set serves as a template and the original one as a reference. We also introduce uniformly and Gaussian distributed noise so that for each noise type 5%, 10%, 20%, 40% or 50% of points in the resulting point cloud represent noise. In the case of the uniform distribution, noise is added to the bounding box of the 3D scan. In case of Gaussian noise, the mean value in the center of the bounding box is taken. For every noise com-

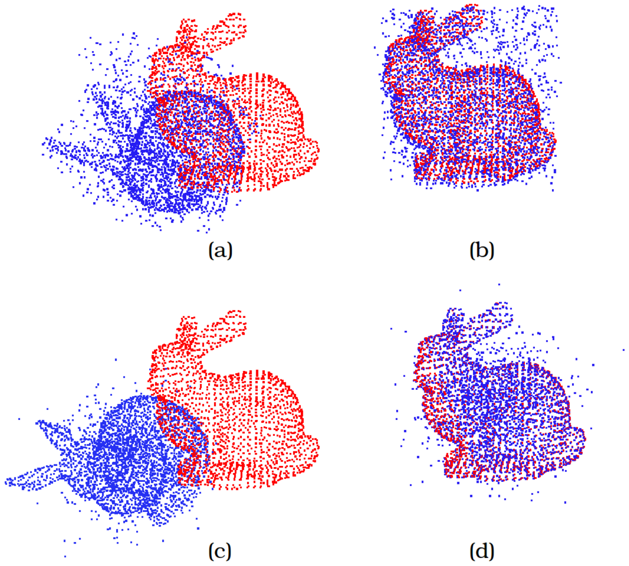


Figure 3: Registration results from the experiment on synthetic data (Stanford Bunny [31]). The reference is shown in red, the noisy template in blue. (a) Initialization, template contains 40% of uniformly distributed noise; (b) result with a uniformly distributed noise; (c) initialization, template contains 40% of a Gaussian noise; (d) result with a Gaussian noise. GA is more robust to Gaussian noise in terms of the mean distance and RMSE, but it resolves rotation more often under uniformly distributed noise.

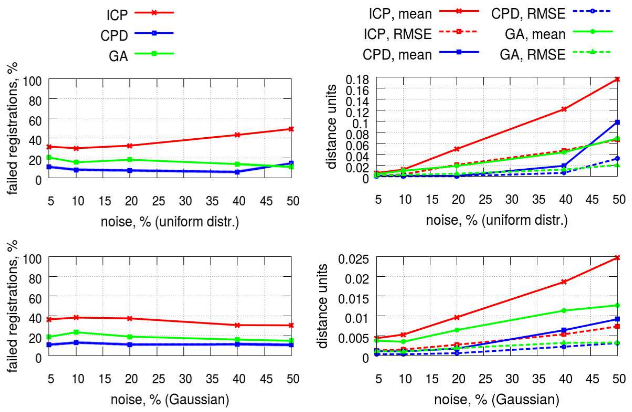


Figure 4: Results of the experiment on synthetic data: reference is the Stanford Bunny [31], template is a randomly transformed copy of it with 5%, 10%, 20%, 40% and 50% added uniformly distributed and Gaussian noise. Metrics are calculated over 500 runs for every noise level and type.

bination out of ten, 500 random transformations are applied resulting in 500 random initial misalignments. For every initial misalignment and noise combination, rigid registrations with ICP, CPD and GA are performed. To assure the highest accuracy, a CPD version without FGT is used. CPD takes one parameter, i.e the estimated amount of noise in a data set which is set to the corresponding noise level in every run. For GA, $G = 6.67 \cdot 10^{-5}$ is set. We measure mean distance and root-mean-square error (RMSE) between the reference and a registered template. The noise is removed while computing the metrics, since we are interested in the quality of data alignment and exact correspondences are known in advance. For all algorithms, failed registrations are not considered in computation of the metrics. Instead, the amount of failures is reported separately. A criterion for registration failure is defined in terms of a threshold on RMSE. We observe that in the experiment with the 3D Bunny, RMSE is either < 0.2 or > 0.4. In the former case point clouds always appear to be registered correctly, at least approximately well, whereas in the latter case they are never registered correctly. Thus, we set the failure threshold to 0.3. In Fig. 3, exemplary results of GA are shown (40% noise level). Running time of GA ranges from 1.5 to 10 minutes per run depending on the noise level. The algorithm converges at most after 100 iterations when possible oscillations around the local minimum attenuate. Results of the experiment are summarized in Fig. 4.

GA shows intermediate performance between ICP and CPD. In average, it fails more rare than ICP and more often than CPD when resolving rotation. The angle of initial misalignment causing GA to fail lies in the range $[\frac{\pi}{4}; \frac{\pi}{2}]$, whereby the higher the angle, the smaller the probability to resolve rotation correctly. CPD starts to fail when the angle of initial misalignment exceeds $65°$. Results of the experiment confirm the tendency — since the set of initial misalignments is equal for all algorithms, direct angle comparison in failure cases is performed. All three algorithms are stable against Gaussian noise while resolving rotation (the number of failures does not correlate with the level of Gaussian noise). CPD and GA are also stable to uniformly distributed noise. In case of 50% of uniformly distributed noise, GA outperforms CPD both in terms of the mean distance, RMSE and amount of correct registrations. Here, the difference between a probabilistic approach and our method comes to light: in the case of GA, more distant points contribute more significantly (hyperbolic expression) than in the case of CPD (Gaussian vicinity) allowing for more robust cumulative compensation.

Experiments on real data. We evaluate GA with several experiments on real data. Fig. 2 depicts the course of the first one. We take a human body scan ($4.2 \cdot 10^4$ points) and a template ($3 \cdot 10^3$ points) reconstructed on a multi-view system with an algorithm described in [12]. We add synthetic clustered outliers (forming a sphere) to the reference. Initially, the template is located exactly between the human scan and the sphere (Fig. 2-a). Both point sets are registered with ICP, CPD and GA. ICP fails to associate the human scan with the human template. Being influenced by direct nearest neighbours, it converges to the sphere (if the template is located closer to the human scan, ICP resolves rotation less accurate than CPD and GA) (Fig. 2-b). GA and CPD, provided appropriate parameters are chosen, resolve registration correctly (Fig. 2-c,-d). If the corresponding parameter is set suboptimally (either G for GA or the weight w for CPD), result of GA is not as accurate, whereas CPD may fail to resolve the example. The experiment demonstrates the gravitational nature of GA — subspaces with a higher total mass win against outliers, even if they are clustered.

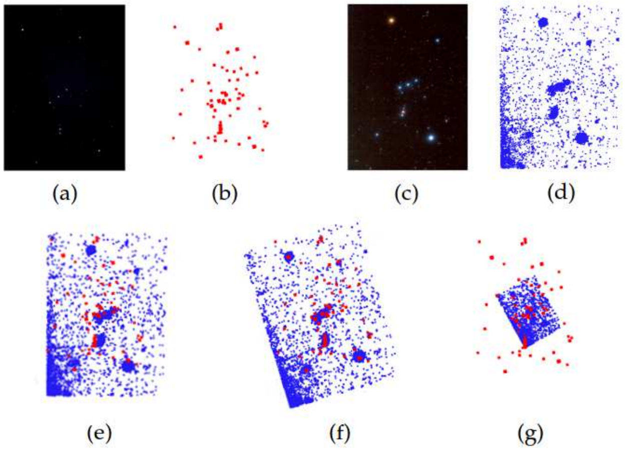


Figure 5: Experiment with prior correspondences as applied to image registration. Pictures are converted into 2D point sets, whereby pixel intensities determine point masses. (a) Reference image [32] and (b) the corresponding point set; (c) template image [30] and (d) the corresponding point set; (e) initial alignment of the point sets; (f) recovered transformation by GA; (g) registration result of CPD.

The second experiment on real data is selected to demonstrate performance of GA in presence of structured outliers and with missing parts. We use two scans of the guardian lions reconstructed with the multi-view algorithm [12]. The reference (Fig. 6-a) represents a processed 3D model. We register it with a rough reconstruction of the lion with 7% of contiguous points removed from the area of the head (Fig. 6-b). Fig. 6-b,-c show initialization and the registration result respectively. During the registration, a 30x subsampling of both point clouds is used and the recovered transformation is applied to the initial template. Results are highly accurate despite of outliers and missing parts.

In the third experiment on real data, the influence of different particle masses is evaluated. The point sets are obtained from two different images of the Orion constellation. For every point, a weight according to the grayscale pixel intensity is set. Fig. 5 depicts the course of the experiment. The

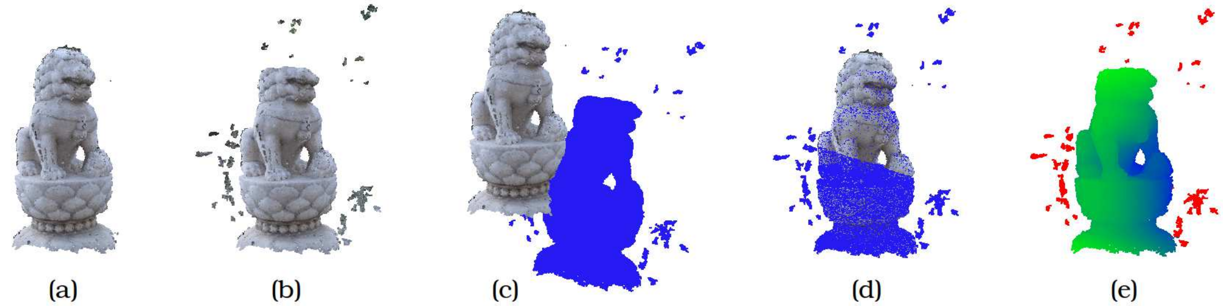


Figure 6: Results of the experiment with structured outliers and missing parts. (a) the processed reference 3D model, $1.45 \cdot 10^5$ points; (b) reconstruction with removed 7% of the points, $1.34 \cdot 10^5$ points. (c) initialization; (d) GA registration result; (e) cloud-to-cloud distance visualized with a Blue<Green<Yellow<Red color scale; the mean distance amounts to 0.109, the RMSE to 0.63; outliers and missing parts explain the high RMSE.

template and the reference contain 324 and $\sim 10^4$ points respectively, whereby the template contains $\sim 95\%$ noise including clustered outliers. GA downweights darker points corresponding to the noise and emphasizes star clusters with the higher weights. It successfully accomplishes the task and the corresponding star clusters are aligned correctly, as can be observed in Fig. 5-f. CPD is not able to incorporate weighting information and fails, although the noise weight w is set to 0.95. This example shows an advantage of GA against CPD — incorporating weights — which can influence the registration procedure in a favourable way. Different masses can be also assigned to particles if prior correspondences between point sets are known in advance.
An additional experiment on the SLAM benchmark datasets — the Stanford 3D Scene Dataset [40] and CoRBS [36] — is placed in the supplementary material. The experiment shows that GA can potentially be used in a SLAM system to register point clouds captured by a depth sensor.
Discussion. The experiments confirm our hypothesis — it is possible to register point sets through modelling a rigid system of particles in a force field. The results evince suitability of the proposed method to cope with real-world scenarios. In the above experiments GA performs robustly in presence of large amounts of noise, especially uniformly distributed noise. We believe that the unique properties such as embedding of prior correspondences through different point masses and outlier suppression need to be further investigated. In the experiments $G \in [6.67 \cdot 10^{-6}; 6.67 \cdot 10^{-5}]$ and $\eta \in [0.2; 0.9]$ were chosen and the step size Δt was fixed to 1. G and η counterbalance each other and should be set depending on the scale and the total mass of the points involved in the registration. Higher values of η might lead to a faster convergence, but might also hinder the algorithm to find a solution. The current limitation of the proposed approach consists in its limited capability to resolve scale which requires a special parameter tuning. If the parameters are set suboptimally, the template may shrink to a single point. Also in the current implementation, GA can handle large point sets through subsampling.

5. Conclusions

In this paper, a novel multiply-linked rigid point set registration algorithm is introduced — the Gravitational Approach — which is based on the concept of the particle movement in a force field. The new approach is well parallelizable and allows to embed prior correspondences through inhomogeneous point weights. Various acceleration techniques can be adopted for GA reducing the computational complexity or providing a speedup in a respective complexity class. Experiments on synthetic and real data show that GA is robust against clustered outliers. The new method outperforms ICP in terms of the ability to resolve rotation and in the mean distance and RMSE metrics. In scenarios with especially large amount of a uniformly distributed noise, GA may also outperform CPD. In future work, we will focus on the stable scale resolving as well as finding an efficient mixture of acceleration techniques for GA. We also plan to generalize it to the non-rigid case.

Acknowledgements

The work was funded by the BMBF project DYNAMICS (01IW15003). The authors thank Dr. Gerd Reis and Dr. Bertram Taetz for the valuable comments. A special thank of the first author goes to Dr. Willi Freeden whose lecture *Inverse Problems* at the University of Kaiserslautern inspired him for the proposed algorithm.

References

[1] S. J. Aarseth. Dynamical evolution of clusters of galaxies, I. *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 126:223, 1963. 4

[2] S. J. Aarseth. *Gravitational N-body Simulations Tools and Algorithms*. Cambridge University Press, 2003. 3

[3] S. J. Aarseth, C. A. Tout, and R. A. Mardling, editors. *The Cambridge N-body lectures*. Springer, 2008. 3, 5, 6

[4] A. Ahmad and L. Cohen. A numerical integration scheme for the n-body gravitational problem. *Journal of Computational Physics*, 12:389–402, 1973. 5

[5] J. Barnes and P. Hut. A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324:446–449, 1986. 5

[6] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992. 1, 2, 3

[7] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D object pose estimation using 3D object coordinates. In *European Conference on Computer Vision (ECCV)*, pages 536–551. Springer, 2014. 2

[8] J. Burkardt. PLY repository of the Florida State University. http://people.sc.fsu.edu/~jburkardt/data/ply/ply.html. [accessed on 30.10.2015]. 1

[9] M. Burtscher and K. Pingali. An efficient CUDA implementation of the tree-based barnes hut n-body algorithm. *GPU Computing Gems Emerald Edition*, pages 75–92, 2011. 6

[10] F. Diacu. The solution of then-body problem. *The Mathematical Intelligencer*, 18(3):66–70, 1996. 3

[11] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *British Machine Vision Conference (BMVC)*, pages 662–670, 2001. 2

[12] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1362–1376, 2010. 7

[13] S. Gold, A. Rangarajan, C. ping Lu, and E. Mjolsness. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognition*, 31:957–964, 1997. 2

[14] V. Golyanik, B. Taetz, G. Reis, and D. Stricker. Extended coherent point drift algorithm with correspondence priors and optimal subsampling. In *Winter Conference on Applications of Computer Vision (WACV)*, 2016. 2

[15] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987. 5

[16] L. Greengard and J. Strain. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991. 5

[17] B. Jian and B. Vemuri. Robust point set registration using gaussian mixture models. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(8):1633–1645, 2011. 2

[18] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *International Conference on Computer Vision (ICCV)*, pages 1246–1251, 2005. 2

[19] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. 2, 5

[20] MathWorks. File exchange: Iterative closest point. http://www.mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point. [accessed on 30.10.2015]. 6

[21] M. Moghari and P. Abolmaesumi. Point-based rigid-body registration using an unscented kalman filter. *IEEE Transactions on Medical Imaging*, 26(12):1708 – 1728, 2007. 2

[22] A. Myronenko. Coherent point drift (cpd) project page. https://sites.google.com/site/myronenko/research/cpd. [accessed on 30.10.2015]. 6

[23] A. Myronenko and X. Song. On the closed-form solution of the rotation matrix arising in computer vision problems. *Computing Research Repository (CoRR)*, 2009. 2

[24] A. Myronenko and X. Song. Point-set registration: Coherent point drift. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(12):2262–2275, 2010. 1, 2

[25] L. Nyland, M. Harris, and J. Prins. Fast n-body simulation with CUDA. In *GPU Gems 3*, pages 677–795, 2007. 6

[26] C. Papazov and D. Burschka. Deformable 3D shape registration based on local similarity transforms. *Computer Graphics Forum*, 30(5):1493–1502, 2011. 2

[27] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3-D Imaging and Modeling (3DIM)*, pages 145–152, 2001. 2

[28] R. Sandhu, S. Dambreville, and A. Tannenbaum. Particle filtering for registration of 2D and 3D point sets with stochastic dynamics. In *Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[29] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London B: Biological Sciences*, 244(1309):21–26, 1991. 2

[30] Sloan Digital Sky Survey. http://sdssorgdev.pha.jhu.edu/dr1/en/proj/kids/constellation/images/orionstars_tiny.jpg. [accessed on 25.10.2015]. 7

[31] The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/. [accessed on 30.10.2015]. 6

[32] The UK Dark Sky Discovery Partnership. http://www.darkskydiscovery.org.uk/the_night_sky/orion_the_huntersmall.jpg. [accessed on 25.10.2015]. 7

[33] M. Trenti and P. Hut. Gravitational N-body simulations. *Scholarpedia*, 3(5), 2008. 3, 4, 5

[34] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision (ECCV)*, pages 558–569. Springer, 2004. 2

[35] P. Wang, P. Wang, Z. Qu, Y. Gao, and Z. Shen. A refined coherent point drift (cpd) algorithm for point set registration. *Science China Information Sciences*, 54(12):2639–2646, 2011. 2

[36] O. Wasenmüller, M. Meyer, and D. Stricker. CoRBS: Comprehensive rgb-d benchmark for slam using kinect v2. In *Winter Conference on Applications of Computer Vision (WACV)*, 2016. 8

[37] W. M. Wells III. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1-2):63–98, 1997. 2

[38] R. Yokota and L. A. Barba. Hierarchical n-body simulations with autotuning for heterogeneous systems. *Computing in Science and Engineering*, 14(3):30–39, 2012. 6

[39] Z. Zhang. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11):1330–1334, 2000. 2

[40] Q.-Y. Zhou and V. Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (TOG)*, 32(4):112:1–112:8, 2013. 8