Efficient Registration of High-Resolution Feature Enhanced Point Clouds

Philipp Jauer, Ivo Kuhlemann, Ralf Bruder, Achim Schweikard, and Floris Ernst, *Member, IEEE*

Abstract—We present a novel framework for rigid point cloud registration. Our approach is based on the principles of mechanics and thermodynamics. We solve the registration problem by assuming point clouds as rigid bodies consisting of particles. Forces can be applied between both particle systems so that they attract or repel each other. These forces are used to cause rigid-body motion of one particle system toward the other, until both are aligned. The framework supports physics-based registration processes with arbitrary driving forces, depending on the desired behaviour. Additionally, the approach handles feature-enhanced point clouds, e.g., by colours or intensity values. Our framework is freely accessible for download. In contrast to already existing algorithms, our contribution is to precisely register high-resolution point clouds with nearly constant computational effort and without the need for pre-processing, sub-sampling or pre-alignment. At the same time, the quality is up to 28 percent higher than for state-of-the-art algorithms and up to 49 percent higher when considering feature-enhanced point clouds. Even in the presence of noise, our registration approach is one of the most robust, on par with state-of-the-art implementations.

Index Terms—Point cloud, registration, rigid, efficiency, high-resolution, features, graphics processors, CUDA, Monte Carlo, simulated annealing, rigid-body dynamics, many-particle systems, Newton's law, Coulomb's law

1 Introduction

Point cloud registration is a challenging problem inherent to a multitude of applications, such as computer vision, 3D laser scan reconstruction, object recognition or shape retrieval. It also plays an important role in medical applications where registration can be used simultaneously for patient localisation, motion tracking, and monitoring purposes. The same applies to most Localisation and Mapping (SLAM) algorithms, used in mobile robotics applications.

A 3D point cloud describes the discretely sampled surface and spatial structures of an object, e.g., the skin of a patient or a landscape. A point cloud is often recorded by depth sensors, such as laser range scanners and time-of-flight cameras [1], [2], or it represents the surface structure segmented from medical images (e.g., CT, MRI, Ultrasound) [3], [4]. In real world applications, two point clouds of the same object recorded over time always differ. One of the main reasons are spatial movements of the camera, of the object to be recorded, or both. But even if both systems remain absolutely still, the sensor noise causes deviations. Considering a misalignment of at least two point clouds, the objective of the registration problem is to find the optimal spatial fit and to retrieve the possible motion between them, whilst the point clouds have a full or at least a partial overlap.

• *The authors are with the Institute for Robotics and Cognitive Systems, University of Lübeck, Lübeck D-23562, Germany.*
E-mail: {jauer, kuhlemann, bruder, schweikard, ernst}@rob.uni-luebeck.de.

Manuscript received 6 Dec. 2016; revised 18 Dec. 2017; accepted 20 Apr. 2018. Date of publication 29 Apr. 2018; date of current version 10 Apr. 2019. (Corresponding author: Philipp Jauer.)
Recommended for acceptance by J. Frahm.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2831670

Explicit requirements on registration algorithms depend on the application. Generally, they have four major requirements in common: First and most important is the *accuracy*. A registration algorithm should always find the best alignment of fully or partially overlapping point clouds. Second, an algorithm must have a high *repeatability*. Since registration algorithms solve optimisation problems, local sub-optimal solutions must be avoided. Which means that the variance of registration results has to be small. Third, a low computational *complexity* is required. With the resolution of point clouds (number of points per cloud) the complexity usually rises, causing higher computational costs. But especially in robotics and medical applications, real-time registration is frequently required. Finally, algorithms must be *robust* against noise and outliers. These interference factors can be caused by the sensor noise of the camera, damaged detectors or environmental influences (e.g., light, reflection or artefacts).

The development of an optimal registration algorithm to meet these requirements is an ongoing problem to be solved. On the one hand, the development of camera systems is extremely fast. Along with new technologies, the resolution of the cameras constantly increases, which requires new strategies to handle the rising complexity of point cloud registration processes, regarding high accuracy and repeatability of the results. On the other hand, the number of applications which rely on point cloud registration increases while the quality requirements are getting more strict.

To improve the registration results, not only the spatial distribution of the points can be used [5]. The points of a cloud can be further enhanced by integrating features. There are depth sensors which provide additional colour information or intensity values (e.g., produced by heat or infrared sensors) per point. Especially in applications where

point clouds have a lack of structural information, these features are essential for high quality registration results.

The high number of emerging applications, the fast progression in depth sensor technologies and the increasing demands on accuracy and processing time are reasons why point cloud registration is an important, challenging and well-studied research field. Hence, an immense variation of approaches and their modifications exists which is subject to ongoing development.

In this paper we propose a novel framework for highly efficient 3D rigid point cloud registration, even if the point clouds are enhanced by features. Hence, our approach is able to match high-resolution point clouds with far more than 100,000 points each, while the runtime of the registration process remains constant and the accuracy is usually better than common techniques. In addition, our framework can easily be extended by various registration metrics measuring point cloud correspondences for specific applications and diverse sensor types which provide further features such as colours.

1.1 Related Work

Over the last decades, numerous rigid registration algorithms, incremental modifications and extensions have been developed and published. They can be divided into two categories: coarse and fine registration. Many registration frameworks use a combination of both. Coarse registration techniques require known point correspondences and are often used for a rough initial pre-alignment, followed by a more precise fine registration step. This is to avoid convergences into local optima and for speed-up purposes. Our approach can be classified as a fine registration algorithm and is designed to work robustly without coarse pre-alignment steps.

Due to the large number of approaches, the overview given in this paper is limited to the most relevant fine rigid registration concepts and does not claim completeness. We further refer to surveys reviewing algorithms of different registration techniques [3], [6], [7], [8], [9], [10], [11], [12].

The best-known and most widely used representative of registration methods is the iterative closest point (ICP) algorithm. The ICP was introduced by Chen and Medioni [13] and by Besl and McKay [14]. Over the last three decades, several variants and extensions have been proposed such as the point-to-point [14], [15] and point-to-plane [13], [16], [17] approaches. Point-to-point directly computes the error metric between corresponding points while point-to-plane finds the closest distance between each point in the one point cloud and the corresponding tangent plane in the other point cloud. Further variants and extensions are reviewed and evaluated in [18], [19], [20], [21], [22]. It could be shown that the ICP performs with high accuracy and high speed when aligning small point clouds with less than 100,000 points each. To increase the performance of the ICP, it is often recommended to down-sample the point clouds before the registration process is started. However, most ICP variants are sensitive to outliers or noise and strongly depend on good initial pre-alignment. Due to the binary nearest-neighbour correspondences, the registration process can converge to a local optimum. To overcome this problem, the Robust Point Matching (RPM) algorithm uses soft assignments of point correspondences, which are continuously weighted and optimised using deterministic annealing [23], [24]. This makes the RPM more robust against noise and outliers than the ICP.

Another approach using soft assignments for registration is the Kernel Correlation (KC) [25]. It models each point cloud using a kernel density function so that entropy measurements can be used as matching criteria. Applying Gaussian kernels, this approach also shows robustness with respect to noisy point clouds as well as outliers and belongs to the probabilistic registration methods. Further probabilistic techniques are methods using Gaussian mixture models [26].

Also, the Coherent Point Drift (CPD) algorithm represents the points of one cloud as centroids of a Gaussian mixture model [27], [28]. These centroids are fitted to the points of the other cloud by an expectation-maximisation algorithm, which finds the maximum likelihood between both cloud representatives. In addition to its robustness with respect to noise and outliers, the CPD is generally very accurate, but has a high computational runtime.

Another probabilistic approach to be mentioned is the Normal Distribution Transform (NDT) [29], [30]. Here, the spatial point space of the cloud to be aligned to is subdivided into cells. All points within a cell are transformed into a normal distribution. Points of the corresponding cloud are matched with respect to these distributions, which provides a locally smooth representation of the surface. The NDT shows less sensitivity with respect to pre-alignments and shows a good performance considering the registration speed. But the NDT also shows a larger variance in the registration results than algorithms such as ICP and CPD.

While many approaches are limited to spatial point correspondences, only a few methods exist which directly extend the fine registration problem by enhancing the point clouds by features such as intensity, colour or texture information. Most fine registration algorithms dealing with feature enhanced point clouds are based on the ICP. There are methods which extend the ICP correspondence search space by adding weighted colour information in different colour spaces as additional dimensions [31], [32], [33], [34], [35]. In [36], a constrained ICP based on hue value classification is proposed. The hue values of each point are classified and scored according to six basic colours. The ICP registration is performed based on that scoring. In [37], the colour ICP described in [31] was modified by quantising the I and Q channel of the YIQ colour model. The ICP in [31] simply finds the closest point according to colour similarities.

Beside ICP extensions, there are two approaches which extend the NDT by colour information. One method applies colour based weights to the score function [38]. The other approach further subdivides one cloud into an octree structure, where the points are classified by the hue values of the HSV colour space [39].

It has been shown that additional feature information can improve the registration results, especially if point clouds are noisy or have outliers. But on the other hand, for most feature enhanced algorithms the computational complexity increases.

It is worth mentioning that there is a large number of improvements to accelerate fine registration algorithms with the help of parallelisation, especially for the ICP. But, to our knowledge, there is no review or survey which

compares the efficiencies of parallelised point cloud fine registration methods. It can be said that these methods will significantly reduce the runtime, but will not increase the accuracy or robustness of the registration process.

Recently, a gravitational approach for point cloud registration was published [40]. This registration technique shows similarities to our approach. Each point of a cloud is represented as a single body or particle. Each particle of the point cloud to be aligned is influenced by gravitational fields evoked by all particles of the other cloud. This gravitational force moves one cloud towards the other, until both are registered. The translational part of the motion is solved by applying rigid-body dynamics, whereas the rotation is based on singular value decomposition (SVD). This rigid registration method also enables scaling to match both clouds. An evaluation of the algorithm shows good results, which are comparable to ICP and CPD implementations. Especially in the presence of a noise and outliers, the gravitational approach outperforms both competitors. Nevertheless, the algorithm has a high computational complexity since the gravitational force field is computed between all points of both clouds. The authors discuss several possibilities to reduce the runtime. However, their tests have shown that the registration process of about 2,000 points per cloud requires 1.5 to 10 minutes, whereby the variation of this processing time mainly depends on the noise level. Since the algorithm focuses on the gravitational approach, it is hardly possible to include feature enhanced point clouds.

1.2 Our Contribution

Often, it is a hard task to adapt the metrics of existing methods to particular requirements of specific applications. The same applies to modifications which should accelerate these approaches regarding high-resolution point clouds. Therefore, we designed a novel paradigm for this registration problem, which provides two main contributions:

1) *Simplicity:* The underlying methodology is motivated by real physical phenomena and is therefore intuitive and easy-to-use. Computations are kept simple and lean, so that the algorithm can also be fitted to multi-core processor architectures per se, e.g., graphics processor units.
2) *Flexibility:* The metric, which describes the correspondences between two point clouds can easily be extended or replaced. The same applies to regularisation, which affects the step size modelling of the optimisation process. This makes the framework adaptable to particular requirements of different applications, without influencing the parallelism of the algorithm. This also includes feature enhancements.

Our approach is based on the principles of classical mechanics and thermodynamics. It is inspired by astrophysics and electrostatic, where the trajectory of objects is either influenced by the gravitational field of many other objects or by the electrostatic field generated by many charges. On this basis, point clouds can be represented as rigid bodies and their points are particles. To align two point clouds, one of them moves towards the other, while the other remains static. This motion is caused by forces acting on the moving point cloud and can be described with rigid-body dynamics, solving both translational and rotational movements. Forces are evoked by the presence of force fields acting on each particle of the moving cloud. The principle of force-induced point cloud motion is detailed in Section 2.

With respect to registration algorithm terminology, the force field is the matching metric which describes the point cloud correspondences. The underlying force model, which generates force fields, can be designed arbitrarily as long as the magnitude of the forces rises according to the quality of the registration or remains constant. As an example and proof of concept, in Section 3 we propose three different force models. One is inspired by Newton's gravitational law and uses spatial information only. The other two are motivated by Coulomb's law, which further includes additional feature information.

Calculating all force fields for each particle of high-resolution point clouds is computationally very complex. However, it can be split into many parallel processes, where each single process has a smaller complexity. That makes this approach per se suitable for massively parallel processor architectures like graphics processing units. The details of the implementation are described in Section 4.

In the last two sections we present and discuss the results of a comprehensive evaluation. We have validated our registration framework in several tests concerning accuracy, repeatability, efficiency and robustness. Furthermore, we compare these results with several common state-of-the-art algorithms and also show that the new paradigm is suitable for point cloud registration problems.

2 Point Cloud Motion

Given two point clouds as finite sets of points $\mathcal{P} = \left(\boldsymbol{p}_j \in \mathbb{R}^3, 1 \leq j \leq M\right)$ and $\mathcal{Q} = \left(\boldsymbol{q}_i \in \mathbb{R}^3, 1 \leq i \leq N\right)$, we call $\mathcal{P}$ the *model* and $\mathcal{Q}$ the *template*. Here, $\boldsymbol{q}_i$ and $\boldsymbol{p}_j$ are spatial positions of the jth point of the model and the ith point of the template, respectively, both located in the same coordinate frame. In addition to the spatial point information, a cloud may be enhanced with features. We define these features as D-dimensional sets of points, with $\mathcal{F}_\mathcal{P} = (\boldsymbol{f}_j^p \in \mathbb{R}^D, 1 \leq j \leq M)$ the features of the model and $\mathcal{F}_\mathcal{Q} = (\boldsymbol{f}_i^q \in \mathbb{R}^D, 1 \leq i \leq N)$ the features of the template. Each feature $\boldsymbol{f}_j^p$ corresponds to $\boldsymbol{p}_j$ and each feature $\boldsymbol{f}_i^q$ corresponds to $\boldsymbol{q}_i$. Hence, $\mathcal{F}_\mathcal{P}$ and $\mathcal{F}_\mathcal{Q}$ must be located in the same feature space. For example, these features can be colour or intensity values as well as local surface structure information, such as normals.

The point cloud registration problem can be formulated as determining a rigid spatial transformation $\mathcal{T}$ which moves the template in a way that both point clouds are optimally aligned, considering both, spatial and feature information. The model cloud remains static. The template cloud as given in $\mathbb{R}^3$ can be transformed using a homogeneous 4×4 matrix

$$\mathcal{T} = \begin{bmatrix} \mathcal{R}_{(\boldsymbol{k},\theta)} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{1}$$

where $\mathcal{R}_{(\boldsymbol{k},\theta)}$ is a 3×3 rotational matrix, rotating an object about the axis $\boldsymbol{k}$ and the angle θ. $\boldsymbol{t}$ is the translation vector. Considering a rigid transformation, $\mathcal{T}$ applies equally to each point $\boldsymbol{q}_i$ of cloud $\mathcal{Q}$.

Since $\mathcal{T}$ is a rigid transformation which is applied to the entire cloud, the registration problem can be regarded as an

analogy to the principles of classical mechanics. By describing the motion of the template with *rigid-body dynamics*, the point cloud can be moved by applying forces that represent the correspondence metric from model to template. These forces act on each point of the cloud and thereby affect the entire point cloud, similar to the theory of *many-particle systems*. Since our methods are related to physical principles but do not describe real-world occurrences, and due to simplicity, the following equations and assumptions are all unit-less.

无单位的

2.1 Many-Particle Systems

In physics (see e.g., [41]), a many-particle system consists of a discrete number of N particles (e.g., the atoms of an object or the planetary system). The ith particle is characterised by its mass m_i and its spatial position $\boldsymbol{r}_i \in \mathbb{R}^3$. Furthermore, a force $\boldsymbol{F}_i \in \mathbb{R}^3$ influences the ith particle. In many-particle systems, this force is composed of an external $\boldsymbol{F}_i^{(ex)}$ and an internal force $\boldsymbol{F}_{ij}^{(in)}$

$$\boldsymbol{F}_i = \boldsymbol{F}_i^{(ex)} + \sum_{j=1}^{N} \boldsymbol{F}_{ij}^{(in)}. \tag{2}$$

还有内力么?

$\boldsymbol{F}_{ij}^{(in)}$ describes the interaction of particle j with particle i, within the same particle system. The external force $\boldsymbol{F}_i^{(ex)}$ is applied to the ith particle from outside the system.

Assuming that Newton's third axiom, $\boldsymbol{F}_{ij}^{(in)} = -\boldsymbol{F}_{ji}^{(in)}$ and $\boldsymbol{F}_{ii}^{(in)} = 0$, applies to the entire system of N particles, the law of conservation of momentum states that

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{F}_{ij}^{(in)} = \frac{1}{2} \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} \left(\boldsymbol{F}_{ij}^{(in)} + \boldsymbol{F}_{ji}^{(in)} \right) = 0. \tag{3}$$

Hence, the internal force has no influence on the ith particle and the total force acting on the ith particle only depends on the external force, i.e.,

$$\boldsymbol{F}_i = \boldsymbol{F}_i^{(ex)}. \tag{4}$$

Similar to real world many-particles systems, the number of points in a cloud rapidly exceeds the computational feasibility to describe the kinematics of each single particle. Therefore, the model of rigid-body-mechanics abstracts a microscopic many-particle system as one macroscopic physical system. Consequently, the system is treated as a centred mass point

$$\boldsymbol{c} = \frac{1}{G} \sum_{i=1}^{N} m_i \cdot \boldsymbol{r}_i, \tag{5}$$

with $G = \sum_{i=1}^{N} m_i$ as the total mass of the system. Furthermore, using the superposition principle, the total force of the system is the composition of each external force per particle

$$\boldsymbol{F} = \sum_{i=1}^{N} \boldsymbol{F}_i^{(ex)}. \tag{6}$$

In classical mechanics, forces that interact with an object affect its motion. This motion of rigid many-particle systems can be abstracted by using the model of rigid-body dynamics.

2.2 Rigid-Body Dynamics

A rigid body is a solid object formed by many particles, whereby particles do not move inside the body. So, neither deformation nor scaling are considered. If a force is applied to the rigid body, the whole system moves as one. Hence, the motion of that rigid body only consists of rotations and translations.

A force $\boldsymbol{F}$ obtained by the many-particle system in Equation (6) acts directly on the centre of mass $\boldsymbol{c}$. A linear motion of the rigid body can be derived from Newton's second law

$$\boldsymbol{F} = \boldsymbol{a} \cdot G, \tag{7}$$

where $\boldsymbol{a}$ is the acceleration and G is the total mass. Using the kinematic equation, the direction and magnitude of a uniform linear motion can be expressed by

$$\boldsymbol{x}(t) = \frac{1}{2}\boldsymbol{a} \cdot t^2 + \boldsymbol{v}_0 \cdot t + \boldsymbol{x}_0, \tag{8}$$

with $\boldsymbol{x}(t)$ as the displacement depending on the time t, $\boldsymbol{v}_0$ as the initial velocity and $\boldsymbol{x}_0$ as the initial displacement.

The same holds for rotational motion. A solid rotates about the centre of mass $\boldsymbol{c}$ which can be described by the kinematic equation of rotation

$$\boldsymbol{\sigma}(t) = \frac{1}{2}\boldsymbol{\alpha} \cdot t^2 + \boldsymbol{\omega}_0 \cdot t + \boldsymbol{\sigma}_0, \tag{9}$$

where $\boldsymbol{\sigma}(t)$ is the rotational displacement depending on the time t. $\boldsymbol{\alpha}$ is the angular acceleration, $\boldsymbol{\omega}_0$ the initial angular velocity and $\boldsymbol{\sigma}_0$ the initial angular displacement.

Considering the algorithmic complexity and runtime aspects, we only assume stepwise motion. Therefore, two assumptions are made:

1. The time always has a step size of $\Delta t = 1$.
2. At the beginning of each motion step, the rigid body is at rest. This means that $\{\boldsymbol{v}_0, \boldsymbol{x}_0, \boldsymbol{\omega}_0, \boldsymbol{\sigma}_0\} = \boldsymbol{0}$.

With these assumptions, the system of particles can be moved iteratively and both kinematic Equations (8) and (9) can be simplified, with

$$\boldsymbol{x} = \frac{1}{2}\boldsymbol{a}, \tag{10}$$

and ?

$$\boldsymbol{\sigma} = \frac{1}{2}\boldsymbol{\alpha}. \tag{11}$$

Please note that, for simplicity and unlike real-world physics systems, we raise no claim of unit preservation. Equations (7) and (10) show that the linear displacement $\boldsymbol{x}$ and the force $\boldsymbol{F}$ are directly proportional while the rotational displacement $\boldsymbol{\sigma}$ has to be established considering the external forces acting on each particle of the rigid body.

The external torque influences the rotation of the system and is defined as

$$\boldsymbol{\tau}_i^{(ex)} = \boldsymbol{l}_i \times \boldsymbol{F}_i^{(ex)}, \tag{12}$$

the cross product of the external force and the lever $\boldsymbol{l}_i = \boldsymbol{q}_i - \boldsymbol{c}$ between the centre of mass and the ith particle. Moreover, the first derivative of the total angular momentum L with respect to the time t is defined as

$$\frac{d}{dt}\boldsymbol{L} = \sum_{i=1}^{N} \boldsymbol{\tau}_i^{(ex)}. \tag{13}$$

Considering the two assumptions of stepwise motion we made before, the angular momentum equals its derivative. Now, the unit vector $\hat{\boldsymbol{L}}$ of the angular momentum is the rotational axis of the particle system about its centre of mass.

The moment of inertia

$$J_i = m_i \|\boldsymbol{q}_i - \boldsymbol{c}\|^2, \tag{14}$$

of the ith particle only depends on its mass m_i and the euclidean distance between the particle and the centre of mass of the entire system. Again, using the superposition principle, the total moment of inertia is

$$J = \sum_{i=1}^{N} J_i. \tag{15}$$

Regarding Newton's second law, the angular acceleration $\boldsymbol{\alpha}$ can be formulated element wise as

$$\alpha_i = J \cdot L_i^{-1} \quad , i = 1, 2, 3. \tag{16}$$

Using Equation (11), the euclidean norm of the angular displacement corresponds to the angle $\theta = \|\boldsymbol{\sigma}\|$ by which the solid rotates about the axis $\hat{\boldsymbol{L}}$.

Finally, a stepwise iterative motion of the particle system is given by the linear displacement $\boldsymbol{x}$ as well as the rotational axis $\hat{\boldsymbol{L}}$ and the angle θ. These three motion parameters can be converted into the matrix representation shown in Equation (1) where the translational part $\boldsymbol{t} = \boldsymbol{x}$ contains the linear motion. The axis of rotation, given by the unit vector $\hat{\boldsymbol{L}} = \left(L_x, L_y, L_z\right)^{\mathrm{T}}$, and the angle θ can be converted into a 3×3 rotation matrix $\mathcal{R}_{(\hat{\boldsymbol{L}},\theta)}$ using the convention presented in [42]. Then the transformation for each iteration step k is

$${}^{k-1}\mathcal{T}_k = \begin{bmatrix} \mathcal{R}_{(\hat{\boldsymbol{L}},\theta)} & \boldsymbol{x} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{17}$$

Since the motion of the particle system is iterative, usually more than one iteration step has to be considered to solve the registration problem. Consequently, the entire body movement is the successive combination of all iteration steps

$${}^{0}\mathcal{T}_K = {}^{0}\mathcal{T}_1 \cdot {}^{1}\mathcal{T}_2 \cdot \ldots \cdot {}^{K-1}\mathcal{T}_K, \tag{18}$$

where K is the number of iterations.

Substituting the particle system by the template point cloud $\mathcal{Q}$ and its points, the motion of the template can be described by using forces and the motion Equation (18).

3 Force Modelling

In the previous section we showed that the mechanical principles of many-particle systems and rigid-body dynamics hold for point cloud motion. We found that the motion of a point cloud only depends on the external forces $\boldsymbol{F}_i^{(ex)}$ acting on each point $\boldsymbol{q}_i$. In this section we present a method to model forces with respect to the registration problem. In other words, given the static model point cloud $\mathcal{P}$ and the moving template point cloud $\mathcal{Q}$, we have to describe forces which move the template towards the model until they are optimally aligned.

One of the main contribution of this work is the flexibility to easily extend, modify or substitute the metrics according to specific applications. Thus, the underlying method which computes these forces can be modelled arbitrarily, as long as two conditions are kept: 力学模型的要求：1. 力的方向；2. 力的大小

1. An attractive force $\boldsymbol{F}_i^{(ex)}$, which acts on $\boldsymbol{q}_i$ has to point towards the most probable fits in $\mathcal{P}$. Repulsive forces are allowed which point in the opposite direction of probably bad fits.
2. The magnitude of $\boldsymbol{F}_i^{(ex)}$ has to rise with respect to the quality of fits or has to be constant.

Otherwise an optimum would never be reached.

As an example and proof of concept, we propose three different force model designs that are motivated by physical laws. However, it is also possible to apply non-physics motivated models as well as hybrids.

3.1 Newton's Law of Gravitation

The first method is motivated by Newton's law of gravitation. It describes forces which direct the planetary motion [43]. Just like planetary objects in a solar system, the points of the template should be attracted by the gravitational field of the model. The gravitational field is generated by all points of the model cloud, depending on their spatial position.

We assume that all points in $\mathcal{P}$ and $\mathcal{Q}$ are mass points in the same coordinate frame and the gravitational forces between points in the same cloud equal zero. Subsequently, each point $\boldsymbol{q}_i \in \mathcal{Q}$ is influenced by the gravitational force field of $\mathcal{P}$. The external force $\boldsymbol{F}_i^{(ex)}$ acting on $\boldsymbol{q}_i$ is given by the gravitational force field

$$\boldsymbol{F}_i^{(ex)} = -\gamma \sum_{j=1}^{M} \frac{m_i m_j}{\left\|\boldsymbol{q}_i - \boldsymbol{p}_j\right\|^3} \left(\boldsymbol{q}_i - \boldsymbol{p}_j\right), \tag{19}$$

with γ being the gravitational constant and m_i and m_j the point masses. We assume that each point has a mass of $m_i = m_j = 1$. Because of relatively small scaled point clouds and consistently small masses (in contrast to planetary gravitational fields), we set $\gamma = 1$.

This force model considers the spatial information of the point clouds. Instead of only using point to point correspondences, gravitational fields may also consider the structural distribution of points within the cloud.

3.2 Coulomb's Law 库仑力

Newton's gravitational law only relies on spatial coherences but does not take feature information into account. For feature enhanced point clouds, we present a further physically motivated approach with two different variants: First, a method based on Coulomb's law only considering attractive forces. Second, a method based on Coulomb's law simulating attractive and repulsive forces.

Similar to the gravitational force field, Coulomb's law relies on the points and surface coherence of two point clouds, but replaces the point masses by electrical charges

[43]. In a static electrical field, the force acting on a charged point can be written as

$$F_i^{(ex)} = \kappa \sum_{j=1}^{M} \Delta_i \frac{(\boldsymbol{q}_i - \boldsymbol{p}_j)}{\|\boldsymbol{q}_i - \boldsymbol{p}_j\|^3}, \tag{20}$$

where $\kappa = \frac{1}{4\pi\epsilon_0}$ and ϵ_0 is the electrostatic constant. Because κ is constant and the point clouds $\mathcal{P}$ and $\mathcal{Q}$ are relatively large scaled (in contrast to real-world electrical charges), we assume $\kappa = 1$. Furthermore, $\Delta_i = \zeta_i \cdot \zeta_j$ describes the influence of the charges on each point $\boldsymbol{q}_i$ and $\boldsymbol{p}_j$, respectively. Per definition of Coulomb's law, the charges are either positive or negative. Consequently, two charges of the same sign produce repulsive forces while charges of opposite signs act attractive.

To consider the features $\mathcal{F}_\mathcal{P}$ and $\mathcal{F}_\mathcal{Q}$ as charges, the values of both feature sets have to be within the same ranges per dimension. Henceforth, we assume a feature range of $[0, 1]$. Moreover, we assume all features are given in the same feature space (were produced by the same modality, e.g., infrared camera or RGB-sensor). And in contrast to real-world electrical charges, feature points which are equal should attract each other.

For the point cloud registration problem, we interpret Coulomb's law in two different ways, with and without repulsive forces. The basic construct of the simplified Coulomb's law remains equal in both variants, except for the factor Δ_i.

The first variant is

$$\Delta_i = 1 - \frac{\left\|\boldsymbol{f}_i^q - \boldsymbol{f}_j^p\right\|}{\sqrt{D}}, \tag{21}$$

with $\|\boldsymbol{f}_i^q - \boldsymbol{f}_j^p\|$ as the euclidean distance between $\boldsymbol{f}_i^q \in [0, 1]^D$ and $\boldsymbol{f}_j^p \in [0, 1]^D$. Likewise, Δ_j is in the range $[0, 1]$ and scales the spatial correspondences. If the features of two points have a proper fit, the force more likely depends on the spatial coherence. Otherwise, if the features of two points do not fit, the distance between both points does have less or no effect on the force.

As a second approach

$$\Delta_i = 2\left(0.5 - \frac{\left\|\boldsymbol{f}_i^q - \boldsymbol{f}_j^p\right\|}{\sqrt{D}}\right), \tag{22}$$

is a variant of Equation (3.2) which additionally considers repulsive forces. Again, $\|\boldsymbol{f}_i^q - \boldsymbol{f}_j^p\|$ is the euclidean distance between $\boldsymbol{f}_i^q \in [0, 1]^D$ and $\boldsymbol{f}_j^p \in [0, 1]^D$. However, in this case $\Delta_j \in [-1, 1]$ does not only scale the spatial coherence concerning fits of features but also produces repulsive forces. These repulsive forces occur whenever it is most likely that the features are different.

4 Algorithmic Implementation

In the previous sections, we described an approach to represent point clouds as an analogy to many-particle systems and rigid-body dynamics, both known from mechanics. Using this representation, a point cloud can be moved towards the other to solve the registration problem. The motion is caused by forces acting on each point of the template point cloud. Forces can be seen as a metric for optimisation processes and can be modelled arbitrarily. In Section 3 we provided three possible approaches. Below, we propose the algorithmic implementation of the point cloud registration framework.

From an algorithmic point of view, we have to consider three major problems. First, the computational complexity of many-particle systems rapidly increases with the number of points per cloud. Second, the forces increase proportionally to the registration quality. Due to the fact that points do not collide (since they have no matter) and due to increasing forces, the system tends to overshoot and the forces can become infinitely high for perfect alignments. Third, since our approach is an iterative optimisation algorithm, it requires a termination criterion.

To solve these problems, we derived an algorithmic approach based on the idea of *simulated annealing* and *Monte-Carlo Simulation*. We will show how these principles can be adapted for motion controlling and complexity reduction. The proposed framework can be easily adapted to new force model designs and can be further evaluated against other point cloud registration approaches.

4.1 Motion Controlling

In contrast to other state-of-the-art registration methods, the point cloud motion is modelled by attracting and repelling forces. That means, a superposed force points to the direction of local and global optima. But without direct correspondences between both clouds (e.g., point-to-point, point-to-plane or centroids of Gaussian mixture model) the magnitude of the forces does not correspond to the displacement of both clouds. Furthermore, the force field produces rising forces while achieving better alignments, which can even become infinite for perfect fits. As a consequence, large overshoots or oscillations in the registration process can occur. To solve this problem, we implemented an approach similar to the simulated annealing algorithm, which regulates the step sizes of the motion.

Simulated annealing is a stochastic algorithm which originally was introduced to solve combinatorial optimisation problems [44], [45]. The algorithm was derived from the analogy between physical annealing processes and iterative optimisation. In statistical thermodynamics, a system which is heated has a high energy and is therefore very unstable and behaves virtually random. When the material slowly cools down, the system relaxes and becomes stable. This cooling process is called annealing. Simulated annealing can be characterised as a gradient descent search method that takes uncertainties in the state transition process into account. Each state of the system has a specific energy, which is equal to the cost function of each optimisation step. The transition probability is characterised by the Boltzmann-Gibbs distribution and therefore depends on the current system temperature and energy. In contrast to classic gradient descent optimisation, simulated annealing allows for unexpected state transitions, making it less likely to get stuck in local optima. Since simulated annealing uses the system temperature as a termination criterion and energies as cost functions, it is possible to adopt it for our registration approach.

Our registration process starts with an initial arbitrary temperature $T_0 > 0$. At each iteration, the system anneals by a constant cool-down-factor $c \in]0, 1[$, with

$$T_k = T_{k-1} \cdot c, \tag{23}$$

where k is the current iteration step. Furthermore, we characterise the registration quality of each iteration by its *kinetic energy* E_k. For a rigid body, the kinetic energy can be written as

$$E = \frac{1}{2} M v^2 + \frac{1}{2} J \omega^2. \tag{24}$$

Here, M is the mass of the system, which we assume to be $M = 1$. Considering stepwise rigid body motion, the velocity v equals the magnitude of the linear displacement $\boldsymbol{x}$, which was derived in Equation (10). The angular velocity ω, which equals the angle of rotation θ, and the moment of inertia J are already known from Equation (15). Again, please note that this approach does not claim to preserve real-world physical units.

Since the forces and the registration quality are directly proportional, the algorithm has to solve a maximisation problem, where the kinetic energy continuously rises. However, this would imply that for higher energies, the point cloud motion gets larger and thus an optimum cannot be reached. To solve this, we refer to a real-world physical behaviour, where atomic movement slows down with decreasing temperatures, stopping completely at zero Kelvin. In analogy, the rigid-body-motion depends on the current system temperature T_k at iteration step k. Assuming a perfect alignment of both point clouds, the forces produced by the force field are infinitely high. Hence, the algorithm has to avoid unrealistically high energy gradients between two iterations. Especially at the end of the registration, forces are getting very high and the motion has to be very fine, large translational and angular displacements must be avoided.

Algorithm 1 shows an overview of our adapted simulated annealing approach. For each iteration step, two stages have to be regarded. The first stage considers the transition probability, with ΔE being the difference between the kinetic energy E_k of the current state and E_{k-1} the predecessor's energy. The transition probability is defined by the function $exp(\Delta E, T_k)$, which is defined as

$$\rho_T = e^{\left(\frac{\Delta E}{T_k}\right)}, \tag{25}$$

the probability of ΔE depending on the current temperature T_k, regarding the Boltzmann-Gibbs distribution [45]. The current linear displacement as well as the current axis and angle of rotation will be used, if $\Delta E < 0$. This extends to the case if ρ_T is smaller than a uniformly distributed random number $p \in [0, 1]$. Otherwise, the current translation $\boldsymbol{x}_k$ and the current rotation axis $\hat{\mathbf{L}}_k$ direct to an optimum, but the amount of motion could be too high. So it is likely that this motion produces an overshoot. Hence, the direction of $\boldsymbol{x}_k$ and $\hat{\mathbf{L}}_k$ remain as they are. But the magnitude of $\boldsymbol{x}_k$ is replaced by $\|\boldsymbol{x}_{k-1}\|$, the magnitude of the previous iteration. The same holds for the angle θ_k and θ_{k-1}.

Hence, the first stage accepts changes in both ways, no matter if the kinetic energy falls or rises, except for extremely large increases. Since the acceptance probability also depends on the current system temperature, larger increases of the step size will be accepted more likely at the beginning of the registration, where the registration is most coarse. For lower temperatures, the acceptance probability rapidly falls and large motions will be avoided.

The second stage scales the point cloud motion regarding the temperature. With lower temperatures, the motion of the template cloud decreases accordingly, holding for the linear displacement $\boldsymbol{x}_k$ as well as for the rotational displacement θ_k. This avoids large overshoots and oscillation about the optimum, especially at the end of the fine registration process.

Algorithm 1. Pseudo Code of the Point Cloud Motion Controlling Algorithm, Based on the Simulated Annealing Approach

• Stage 1: Transition probability
$\Delta E \leftarrow E_k - E_{k-1}$
if $\Delta E \geq 0$ **then**
  $\rho_T \leftarrow exp(\Delta E, T_k)$
  $p \leftarrow rand(0, 1)$
  if $(\rho_T < p)$ **then**
    $\boldsymbol{x}_k \leftarrow \frac{\boldsymbol{x}_k}{\|\boldsymbol{x}_k\|} \cdot \|\boldsymbol{x}_{k-1}\|$
    $\theta_k \leftarrow \theta_{k-1}$
    $E_k \leftarrow E_{k-1}$
  end if
end if
• Stage 2: Motion regulation
$T \leftarrow T \cdot c$
$\boldsymbol{x}_k \leftarrow \boldsymbol{x}_k \cdot T$
$\theta_k \leftarrow \theta_k \cdot T$

4.2 Complexity Reduction

The presented registration approach relies on forces which define the direction of the template cloud's motion. For each point of the template cloud and for each iteration step, the force field with respect to the model cloud has to be computed. Thus, the complexity of the algorithm is $\mathcal{O}(M \cdot N \cdot K)$, with M the number of points in the model, N the number of template points and K the number of registration iterations. Considering modern depth sensor technologies or medical imaging applications, with usually more than one hundred thousand points per cloud, the complexity is too high.

However, a point cloud registration approach based on classical mechanics allows us to substantially reduce the complexity and run the algorithm in nearly constant duration. On the one hand, we exploit the parallelisation of the force field computation. On the other hand, we further reduce the complexity of the many-particle system using a Monte-Carlo simulation approach.

For each iteration step, the single force fields of a template cloud have to be summed up according to the superposition principle in Equation (6) and the resulting point cloud motion has to be computed. This is an iterative process and has to be done sequentially. However, the computational complexity of both processes is constant and the computational costs are relatively low. Aside from that, the force field of each single point in the template cloud is completely independent and can therefore be determined in parallel. The complexity reduces to an approximation of $\mathcal{O}(M \cdot K)$ per template point.

This algorithmic design allows the use of *graphics processor units (GPUs)*, which are made to run a high number of

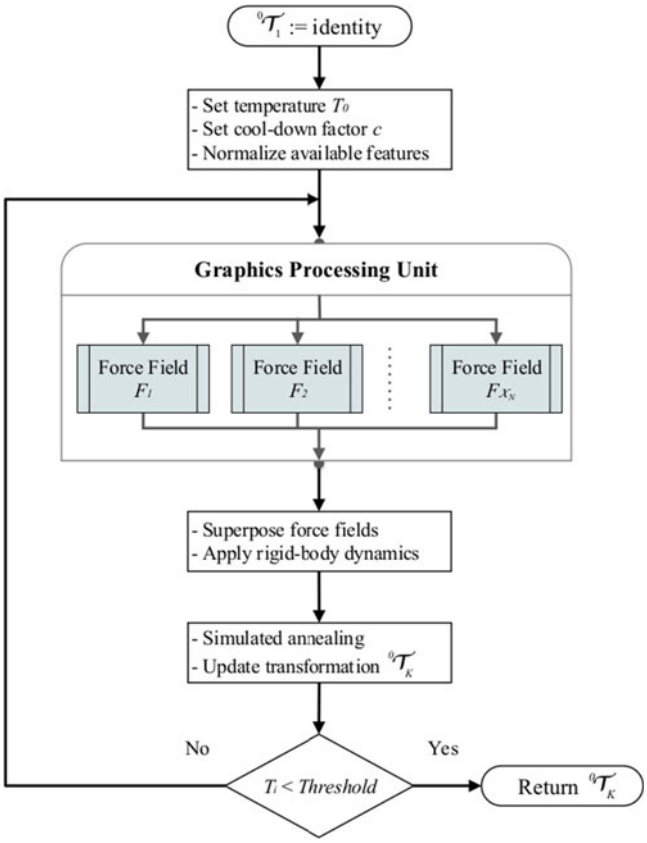


Fig. 1. Flowchart of the registration framework.

processes (threads) in parallel. In contrast to central processing units (CPUs), each process should be lean concerning the number of operations. The computation of N forces is too complex for a single GPU thread and the number of M parallel processes is still too high to handle each in a single thread, even for the newest generation of graphics hardware.

Instead of computing all force fields for each single point of the template cloud, we found that the complexity can be reduced by a Monte-Carlo approach. This, however, is not done by down-sampling both point clouds at the beginning of the registration process, as common approaches do. For each iteration a new random sample is selected. This means that, for each iteration step, the algorithm selects a new uniformly distributed sample of $\mathcal{X}_N$ points from the template cloud. For each of these sample points the force field will be computed, but only considering $\mathcal{X}_M$ uniformly distributed samples of the model cloud. This avoids possible poor registration results as a consequence of a single bad sample selection.

However, the complexity depends on $\mathcal{X}_N$ and $\mathcal{X}_M$ which can be chosen arbitrarily, allowing for $1 \leq \mathcal{X}_N \leq N$ and $1 \leq \mathcal{X}_M \leq M$. For larger sample sizes, the complexity rises while for smaller sample sizes, worse registration results have to be expected. When using a GPU, we found that an optimal ratio between speed and accuracy can be achieved when utilising the full capacity of GPU threads per iteration. For the evaluation of our registration framework, we used a sample size of $\mathcal{X}_N = \mathcal{X}_M = 1024$, equal to the number of threads of the GPU being used (NVIDA GeForce GTX 970). So, for each iteration step the force field of all template sample points is computed in parallel. This guarantees a constant complexity of $\mathcal{O}(\mathcal{X}_M)$ per iteration step and these sample sizes proved to be sufficient for highly accurate registration results. Nevertheless, lower GPU capacities can be compensated by increasing the cool-down-factor c, which leads to a larger number of iterations but also a higher computational time.

4.3 Workflow

Fig. 1 shows the general flowchart of our registration framework. The registration starts with the transformation matrix $^0\mathcal{T}_1$, which can either be the identity or an initial pre-alignment transformation. The initial temperature T_0 as well as the cool-down-factor c need to be set to an initial value greater than zero. If the point clouds are enhanced by feature information, these features need to be normalised. After the initialisation step, the iteration process starts. First, $\mathcal{X}_N$ threads are started on the GPU. Each thread computes the force field of a randomly chosen template point. For each force field calculation $\mathcal{X}_M$ randomly chosen points of the model cloud are considered. The selected sample of model points per iteration is identical for each thread. When the force field computations are completed, the resulting forces are superposed to be used for rigid-body dynamics. The resulting motion parameters and the actual temperature are then applied to the simulated annealing method. It computes the final motion parameters to update the current transformation matrix $^0\mathcal{T}_k$ and updates the current temperature T_k, at iteration step k. If T_k is below a predefined threshold, the algorithm terminates and returns the transformation $^0\mathcal{T}_K$, which aligns both clouds. Otherwise, the next iteration step will be started. For the evaluation of our framework, we used an initial temperature of $T_0 = 1$, a cool-down-factor of $c = 0.98$ and a termination threshold of 10^{-4}.

5 Evaluation

We performed a comprehensive evaluation of our algorithm using different experiments. These are designed to analyse the performance of the algorithm looking at: A) accuracy and repeatability of the registration; B) robustness of the algorithm in the presence of noise; C) efficiency concerning runtime and registration quality; D) accuracy and repeatability of the registration including enhanced feature information and less spatial structures; E) influences of the random step size sampling of the simulated annealing process.

The registration errors are measured by the *root mean square error* (RMSE). For each dataset used in our experiments, $\mathcal{T}_S$ is the known transformation, which perfectly aligns both point clouds $\mathcal{Q}$ and $\mathcal{M}$. $\mathcal{T}_R$ is the transformation produced by a registration algorithm. Then, the RMSE of the algorithm is measured by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \| \mathcal{T}_S \cdot \mathbf{q}_i - \mathcal{T}_R \cdot \mathbf{q}_i \|^2}{N}}. \tag{26}$$

Because the ICP is the most popular robust rigid registration algorithm, with demonstrated robustness and accuracy in numerous benchmarks, we used three different implementation of the ICP as a ground truth.

First, we used the ICP implementation of MATLAB R2016a integrated in the Computer Vision System Toolbox [46]. We considered the point-to-point (*ICP_MP*) and the

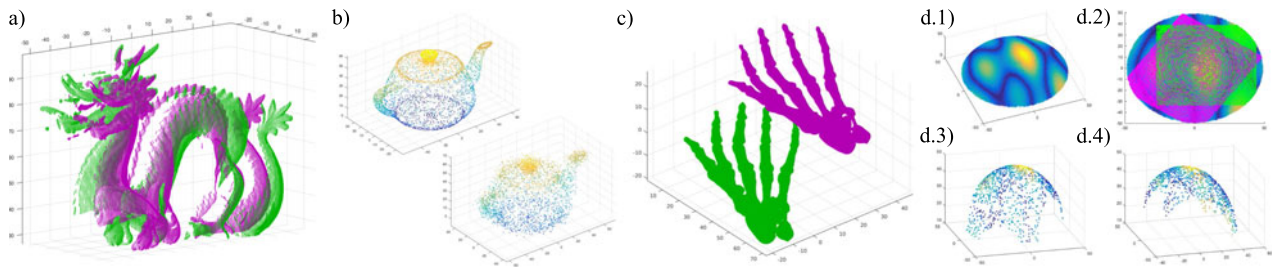


Fig. 2. Evaluation datasets: a) Two point clouds of the Dragon dataset scanned from different angles, before the reconstruction. b) Teapot without (left) and with (right) Gaussian distributed noise, using a variance of $\sigma^2 = 3$ per point. c) High-resolution point clouds with 327,323 points each, used for efficiency analysis. The point clouds are displaced by a randomly generated transformation. d.1) Generated coloured hemisphere, from which d.2) two different but partially overlapping patches were selected. Afterwards, each selected region d.3) and d.4) was sampled separately.

point-to-plane metric (*ICP_MN*). As a second implementation of the ICP algorithm we used the ICP toolbox by Kjer and Wilm [47], [48], which is another common Matlab implementation. Again, we considered both metrics, point-to-point (*ICP_KP*) as well as point-to-plane (*ICP_KN*).

To consider point clouds enhanced by feature information such as intensity values, we provide a third implementation, in which we extended the ICP toolbox by Kjer and Wilm. As already proposed in [31], [33] the ICP's nearest neighbour search was extended by the feature dimensions. But instead of considering a 3D colour space only, the algorithm dynamically extends the search space by the dimension of features for each point. As matching metric we again used both, point-to-point (*ICP_FP*) as well as point-to-plane (*ICP_FN*). Everything else remains identical to the implementation of Kjer and Wilm. Consequently, when no features are available, the ICP_FP behaves equally to ICP_KP and ICP_FN equally to ICP_KN. Hence, for the sake of clarity and comprehensibility, in experiments without feature enhancements, we do not consider the results of ICP_FP and ICP_FN.

Finally, to ensure objectivity, we further compared our algorithm to the coherent point drift (*CPD*) algorithm, which is an alternative approach and represents the class of probabilistic registration algorithms. We used the Matlab implementation provided by Myronenko [27], [28], [49].

For our algorithm, we considered the three force models previously presented, the gravitational force field point registration model (*FPR_G*) and Coulomb's force field model with (*FPR_CR*) and without (*FPR_CS*) repulsive forces. If no features have to be considered, FPR_CR and FPR_CS are equal to the gravitational force field model FPR_G and do not have to be considered in the results. The parameter for the simulated annealing process we set to $T = 1$, $c = 0.98$ and the threshold to 10^{-4}.

All experiments were performed using the same hardware, an INTEL Core i7-6700 CPU, with 16 GB RAM and a NVIDIA GeForce GTX 970 GPU. The system runs Windows 10 and MATLAB R2016a.

A) Accuracy and Repeatability. This experiment is designed to analyse the accuracy and repeatability of the algorithms without feature enhancements. The dataset was taken from the *Stanford 3D Scanning Repository* [50]. These are point clouds of a dragon statue originally generated using a Cyberware 3030 MS laser scanning system. We used the two point clouds from the *Dragon* dataset, shown in Fig. 2a. The object was scanned from two angles differing by about 24 degrees. Hence, both, the model and template cloud are disparate and only have a partial overlap. The model point cloud consists of 41,841 points and has a size of $102 \times 73 \times 36$ units per spatial dimension. The template consists of 34,836 points and has a size of $98 \times 72 \times 50$ units.

The point clouds are originally misaligned, but the transformation $\mathcal{T}_S$ to align them is given by the authors. Thus, $\mathcal{T}_S$ can be considered as ground truth. Hence, at the beginning of each registration trial, the template point cloud will be misaligned by the inverse transformation $\mathcal{T}_S^{-1}$. Subsequently, each algorithm computes the transformation $\mathcal{T}_R$. Given $\mathcal{T}_S$ and $\mathcal{T}_R$, the RMSE can be determined. Additionally, the runtime of each registration procedure was measured.

To be statistically significant, the experiment consists of 1,000 trials, where each algorithm computes one registration. The results are given in Table 1. Since the registration

TABLE 1
Results of a Reconstruction Experiment to Evaluate the Accuracy and Repeatability of the Registration Algorithms

Method	Avg. time in s	# of fails	RMSE Median	RMSE IQR	RMSE Range	RMSE Boxplot
ICP_MP	0.1041	0	0.2649	0.0000	0.0000	
ICP_MN	0.0496	0	0.3002	0.0000	0.0000	
ICP_KP	1.0804	0	0.2625	0.0000	0.0000	
ICP_KN	1.4150	0	0.2091	0.0000	0.0000	
CPD	18.1023	0	0.1012	0.0000	0.0000	
FPR_G	0.6328	0	0.0728	0.0257	44.2111	

The methods are compared using the average runtime, the number of failed trials without a match, the RMSE median, the RMSE interquartile range, the entire RMSE range and the logarithmic scaled Boxplot concerning 1,000 trials.

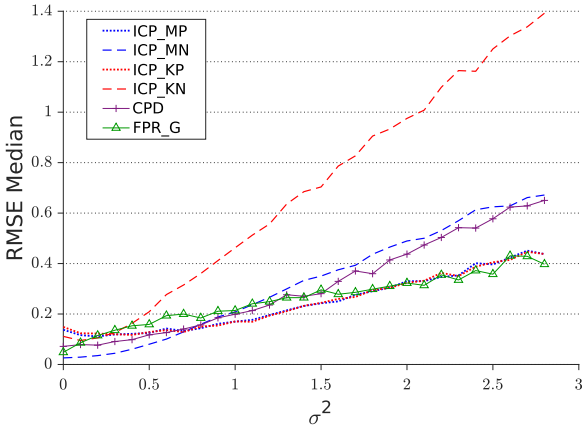


Fig. 3. Registration errors with respect to the Gaussian distributed noise level, characterised by variance σ^2.

accuracy and repeatability was measured by using the RMSE, relevant aspects of the RMSE are listed as well as failed trials without a match.

The results reveal that all algorithms show high accuracy within the sub-unit range. For point clouds with a spatial expansion of 100 units, the RMSE median is below 0.4 units. Since the misalignment was identical for each trial, the RMSE of the ICP and CPD implementation was always the same and shows no variances within the registration results. This is a consequence of the deterministic behaviour of these methods. In contrast, the Monte-Carlo approach leads to a larger variance in the registration results. However, except for two outliers, the full range of the RMSE is comparable to the other algorithms. Regarding the RMSE median, our FPR_G algorithm is 28 percent more accurate than the ICP variants and the CPD. Even the interquartile range (IQR) is below the RMSE median of the other algorithms. The results further show that the Matlab implementations of the ICP are the fastest, with a runtime far below $1\,s$, followed by the proposed algorithm, which shows a runtime below $1\,s$. ICP_KP and the ICP_KN also perform well, but need $1.08\,s$ and $1.415\,s$. The average runtime of the CPD implementation is above $18\,s$, which is very high in comparison to its competitors.

B) Robustness. In this experiment we evaluate the robustness of the registration algorithms with respect to spatial noise. We used the *Utah Teapot* point cloud dataset, provided by Matlab. The point cloud has a size of $129 \times 80 \times 63$ units per spatial dimension. We down-sampled the model point cloud to 5,391 points and the template to 3,318 points. Both clouds were sampled randomly, so that point to point correspondences were unlikely. Additionally, we applied 30 different levels of Gaussian distributed noise to each point of the template point cloud, considering all three axes. The noise level ranges from a variance of $\sigma^2 = 0$ (Fig. 2b), top left) to $\sigma^2 = 2.9$ (Fig. 2b), bottom right), with an incrementation of 0.1. For each noise level, 100 registration trials were performed, each using a new noise distribution. At the beginning of each trial, a randomly generated transformation was applied to the template cloud, with angles in the range of $[-20, +20]$ degrees about each axis and random translations within $[-50, +50]$ units per axis.

Fig. 3 shows the RMSE median with respect to the noise level characterised by σ^2. The expected RMSE between the untransformed templates with and without noise is equal to the current variance σ^2. All algorithms are below the expected RMSE and therefore are robust to noise. Both ICP variants using the point-to-plane metric (ICP_MN and ICP_KN) and the CPD show high accuracy while registering point clouds without or with little noise. However, the registration error increases faster with respect to the noise than for other algorithms. The point-to-point variants of the ICP and the FPR_G are less sensitive to noise and are equally robust.

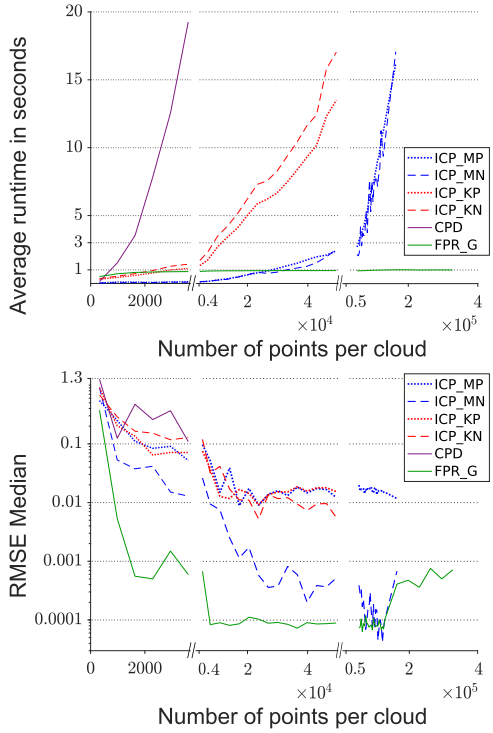


Fig. 4. Results of the efficiency experiments. The top graph shows the dependency of the algorithms' runtime with respect to the number of points. The bottom graph visualises the registration error with respect to the number of points per cloud.

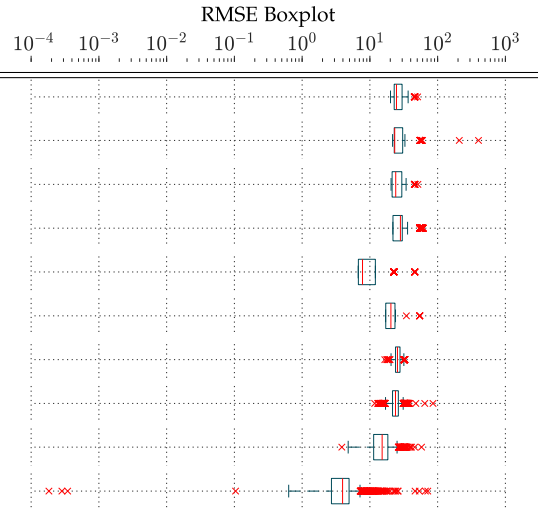
C) Efficiency. This experiment measures the ratio between runtime and registration quality regarding the number of points per cloud. Therefore, we used the *Skeleton Hand* point cloud model provided by the Large Geometric Model Archive at Georgia Institute of Technology [51]. The point cloud is a medical example from the Stereolithography Archive at Clemson University, consists of 327,323 points in total and is shown in Fig. 2c. This model has a size of $66 \times 45 \times 23$ units.

In this experiment the template and the model point clouds represent the same structure, but are differently and randomly sampled. Thus, point to point correspondences are unlikely. Additionally, we down-sampled the point cloud in 51 different increments, from 327 to 327,3323 points per cloud, ensuring identical resolutions for both clouds. For each increment, 100 registrations trials were performed and the RMSE median and average runtime were measured. For this experiment one random transformation was generated, which is a translation of $(-2, -19, -14)$ units and a rotation of 22.44 degree about the rotational axis $(-0.50, 0.26, -0.82)$. The transformation can be seen in Fig. 2c.

The results are shown in Fig. 4. Whenever the runtime of an algorithm exceeded $15\,s$ it was no longer considered for further increments. The top graph shows the average runtime of each algorithm with respect to the number of points

TABLE 2
Result of the Coloured Hemisphere Patches Matching to Evaluate the Influence of Features Enhancements on the Registrations Quality

Method	Avg. time in s	# of fails	RMSE Median	RMSE IQR	RMSE Range	RMSE Boxplot
ICP_MP	0.1156	0	24.6922	7.0316	30.5273	
ICP_MN	0.832	53	23.2517	7.6001	379.4574	
ICP_KP	0.4425	0	24.0695	8.3177	30.6067	
ICP_KN	0.5398	0	28.2278	8.1558	39.0063	
ICP_FP	0.6191	0	7.7855	5.2837	39.2821	
ICP_FN	0.7502	0	20.4288	6.3175	37.4218	
CPD	4.2819	0	25.5296	3.9242	16.2561	
FPR_G	0.8028	0	23.8756	4.7242	73.6676	
FPR_CS	1.0053	0	15.1669	7.0481	54.0324	
FPR_CR	0.7855	0	3.9543	2.2213	71.5187	



The methods are compared using the average runtime, the number of failed trials without a match, the RMSE median, the RMSE interquartile range, the entire RMSE range and the logarithmic scaled Boxplot concerning 1,000 trials.

per cloud. All registration algorithms considered show an exponential increase, except for our FPR_G with an almost constant runtime of about $1\,s$. The CPD's runtime increases most rapidly and exceeds $15\,s$ at about 3,000 points. Matlab's ICP implementations have a runtime below $1\,s$ until a resolution of 27,000 points is reached. Afterwards, the runtime also increases rapidly.

The bottom graph of Fig. 4 shows the logarithmically scaled RMSE median of each registration algorithm with respect to the number of points per cloud. It can be seen that all algorithms reach a better registration result whenever the resolution of the point clouds gets higher. This is due to the smaller amount of spatial information provided at low resolutions. Again, FPR_G achieves the highest accuracy concerning the RMSE median. Even if the point cloud resolution exceeds 300,000 points, our algorithm has an RMSE median of below 0.001 units, while the runtime remains at about $1\,s$.

D) Feature Enhancements. In this experiment we evaluate the accuracy and repeatability of all algorithms concerning point clouds with enhanced features and little spatial structure. Therefor, we generated a very challenging dataset. We created a coloured hemisphere, shown in Fig. 2d.1. Each point is enhanced by intensity information, in a range of $[0, 1]$, i.e., a feature space with one dimension. The radius of the hemisphere is 50 units. Based on that hemisphere, two different but partially overlapping regions were selected, as can be seen in Fig. 2d.2. Afterwards, each region was sampled separately, with the result that it is unlikely that point to point correspondences exist. Both point clouds are shown in Figs. 2d.3 and 2d.4. The model point cloud (Fig. 2d.3) consists of 1,095 points and the template consists of 971 points.

At the beginning of each trial, the template was randomly transformed by a rotation about all three axes with each angle in the range of $[-20, +20]$ degrees and a translation in all three directions about a random value within $[-50, +50]$ units each, using a uniform probability distribution. Similar to experiment A), the experiment included 1,000 trials. The registration RMSE and the algorithm runtime are summarised in Table 2.

The results clearly show that an enhancement by feature information is highly beneficial for point clouds with less spatial structures to improve the registration results. All algorithms that consider only spatial information showed a RMSE median of > 20 units. Our FPR_CR approach using Coulomb's law with repulsive forces shows the best performance concerning the RMSE median (by 49 percent) and the RMSE IQR. The ICP_FP also shows good performances and has a smaller RMSE range. Beside the good result, the FPR_CR shows few outliers resulting in a larger RMSE range. These outliers are caused by the Monte-Carlo approach. Concerning the average runtime, the ICP_FP and the FPR_CR are comparable.

E) Motion Controlling. Finally, a last experiment investigates the influence of the random step size sampling during the simulated annealing process. Again, we used the Dragon reconstruction dataset as for experiment A). We turned off the random motion sampling (Stage 1, Algorithm 1) and start each trial with an initial step size of 1 unit. Then, for each iteration the step size is reduced by the constantly decreasing temperature only (Stage 2, Algorithm 1). Considering the gravitational force model, we ran the reconstruction for 1,000 trials.

The results of this experiment reveal that the optimisation process using superposed force fields is valid, even without random step size sampling. The RMSE median of this experiment is 0.3257. Considering the results of Table 1, this is much more inferior to the FPR_G and CPD, but comparable to the RMSE median of the ICP implementations. Also, the RMSE IQR of 0.6999 is in an acceptable range. But with a total RMSE range of 160.9656 the number of significant outliers extremely rises, which indicates a higher number of false local optimisations.

6 Discussion and Conclusion

We introduced a framework for rigid point cloud registration. The registration process is motivated by the physical

principles of classical mechanics and thermodynamics. We consider point clouds as rigid bodies and their points as particles forming these bodies. Force fields are the registration metric, whereby the underlying force model can be designed arbitrarily, either motivated by physical laws or not. As an example and as proof of concept we provided three models. Moreover, we regulated the cloud motion by an approach similar to the simulated annealing algorithm, where a constantly decreasing temperature is applied. Finally, we proposed a Monte-Carlo approach implemented on the GPU to extremely reduce the complexity of the algorithm.

We performed several challenging experiments to evaluate the performance of the framework. The performance was then compared to state-of-the-art registration techniques, using different variants of the ICP and the CPD.

One experiment was set up to evaluate the registration accuracy and repeatability of the registration algorithms. Since the misalignment of the point clouds was identical for each trial of the experiments, the ICP and CPD implementations showed no variance in the registration results. This is an consequence of their deterministic functionality when using on-to-one correspondences between all points in each registration step. In contrast, our algorithm is based on superposed forces and the Monte-Carlo approach, both leading to many-to-many correspondences per iteration step. Consequently, it shows a much higher variance. However, in 850 out of 1,000 trials, our approach shows the highest accuracy. For 992 trials our algorithm outperforms Matlab's point-to-plane ICP implementation. Considering the RMSE median, our algorithm performs best, 28 percent more accurate than the CPD, the second best algorithm. These results demonstrate that our approach is a valid registration algorithm, which is indeed comparable to other state-of-the-art methods.

A second experiment analysed the robustness of our method in the presence of noise compared to the ICP and CPD implementations. All tested algorithms show robust registration results while the point-to-plane ICP variants and the CPD are much more sensitive to Gaussian distributed noise than the point-to-point ICP variants and our registration approach. This can be explained by the superposed force fields which cause a many-to-many point correspondences. This leads to an averaging of the point misalignments produced by the noise. The same holds for the point-to-point correspondences of the respective ICP variants. The normal vectors of planes and the centroids of the Gaussian mixture models are more susceptible to noise.

One main contribution of the proposed framework becomes clear when reviewing the results of the third efficiency test. In contrast to the other registration techniques, our approach is the most efficient. It is possible to register point clouds with nearly constant runtime, no matter the number of points per cloud. Even point clouds with more than 300,000 points each can be registered accurately within about one second. There is a slight increase of the run time, which is a consequence of data pre-processing to be made and the rising data transfer between CPU- and GPU-memory. Overall, the run time of our approach is constant, even for low-resolution point clouds. In this proof of concept the sampling size of the force field method is equal to the thread capacity of the GPU. In future work, this sampling size could be dynamically lowered for low-resolution point clouds.

The most competitive methods in this experiment are ICP implementations provided by Matlab, which have a runtime below one second up to a resolution of 27,000 points. At 164,000 points per cloud, the runtime exceeds $15\,s$. That is the reason why, in contrast to our approach, current state-of-art algorithms have to be sampled down before registration to work efficiently. Accelerating data structures, like kD-trees used by ICP, inhibit the use of re-sampling per iteration step as our force field method does. A Monte-Carlo brute-force approach using very small samples per iteration (e.g., 1,024 points per cloud) would not work for ICP, because the ICP uses one-to-one correspondences to match point cloud instead of superposed force fields.

A fourth experiment compared the ability of the registration algorithms to accurately match point clouds which have little to no spatial information but are enhanced by additional features, such as intensity values. We generated a challenging point cloud of a hemisphere enhanced by a unique colour pattern. As expected, all algorithms which only take spatial information into account showed bad registration results. Our proposed force model inspired by Coulomb's law using both, attractive and repulsive forces, performed best, followed by the colour enhanced point-to-point ICP implementation. More precisely, our algorithm is 49 percent more accurate, with respect to the RMSE median. The second proposed force model based on Coulomb's law, which only considers attractive forces, shows poor results. This is due to the fact that the colour information only scales spatial information, which was virtually not available in this scenario. This is similar to the colour enhanced ICP, which only weights the point-to-point or point-to-plane correspondences by the feature information.

Finally, we tested the influence of the random step size sampling caused by the simulated annealing process. We see that the optimisation process driven by force fields is a valid gradient ascent approach. Using a constantly decreasing step size, our approach iterates toward the best local optima. The RMSE median is comparable to the registration results of the ICP implementations, but significantly worse than CPD and FPR_G. This clearly shows, that the random step size sampling at high energy transitions and low temperature not only suppresses overshoots, but also provides the possibility to jump out of local optima. Hence, finding the global optimum instead of a false local one is much more likely, which leads to better optimisation quality and lower variances in the registration results.

However, the low complexity offers the possibility to run more than one registration for the same matching problem. This and an optimisation of the initial temperature T_0 and the cool-down-factor c could further reduce the variance in the registration results and will be investigated in future work. Moreover, additional force field models will be investigated and tested for a set of diverse applications, especially with respect to feature enhanced point clouds. These include point clouds recorded by the same sensor as well as multimodal feature types.

Our proposed framework has proven to be a competitive registration approach to current state-of-the-art techniques. This applies for the accuracy as well as for the robustness in the presence of noise. In contrast to existing methods, our approach is designed to work in parallel on multi-core

processor architectures, like GPUs. Efficiency proves whenever point clouds have a very high resolution. No initial pre-processing, coarse pre-alignment or sub-sampling of the point clouds is needed. Moreover, the registration metric and regularisation approach can easily be altered or extended, e.g., when dealing with specific application.

7 Toolbox

The convincing results make our method interesting for numerous applications. Therefor and to further research, we provide a freely accessible Matlab implementation of the framework, which can be found at *www.rob.uni-luebeck.de*.

References

[1] N. Pears, Y. Liu, and P. Bunting, *3D Imaging, Analysis and Applications*. London, U.K.: Springer, 2012.

[2] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, *3D Scene Reconstruction from Depth Camera Data*. Cham, Switzerland: Springer, 2016.

[3] M. A. Audette, F. P. Ferrie, and T. M. Peters, "An algorithmic overview of surface registration techniques for medical imaging," *Med. Image Anal.*, vol. 4, no. 3, pp. 201–217, 2000.

[4] T. Wissel, P. Stüber, B. Wagner, R. Bruder, C. Erdmann, C.-S. Deutz, B. Sack, J. Manit, A. Schweikard, and F. Ernst, "Enhanced optical head tracking for cranial radiotherapy: Supporting surface registration by cutaneous structures," *Int. J. Radiation Oncology Biol. Physics*, vol. 95, no. 2, pp. 810–817, 2016.

[5] H. Kim and A. Hilton, "Influence of colour and feature geometry on multi-modal 3D point clouds data registration," in *Proc. 2nd Int. Conf. 3D Vis.*, Dec. 2014, pp. 202–209.

[6] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu, "Geometry and convergence analysis of algorithms for registration of 3D shapes," *Int. J. Comput. Vis.*, vol. 67, no. 3, pp. 277–296, 2006.

[7] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, 2007.

[8] J. Santamaría, O. Cordón, and S. Damas, "A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling," *Comput. Vis. Image Understanding*, vol. 115, no. 9, pp. 1340–1354, 2011.

[9] G. K. L. Tam, Z. Q. Cheng, Y. K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.

[10] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points: Evaluating recent 3D scan-matching algorithms," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 3631–3637.

[11] Y. Díez, F. Roure, X. Lladó, and J. Salvi, "A qualitative review on 3D coarse registration methods," *ACM Comput. Surv.*, vol. 47, pp. 45:1–45:36, Feb. 2015.

[12] M. Attia, Y. Slama, and M. A. Kamoun, "On performance evaluation of registration algorithms for 3D point clouds," in *Proc. 13th Int. Conf. Comput. Graph. Imag. Vis.*, Mar. 2016, pp. 45–50.

[13] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, pp. 145–155, Apr. 1992.

[14] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[15] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, May 1987.

[16] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a general multi-view registration technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 5, pp. 540–547, May 1996.

[17] S.-Y. Park and M. Subbarao, "An accurate and fast point-to-plane registration technique," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 2967–2976, 2003.

[18] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[19] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[20] B. Bellekens, V. Spruyt, and R. B. Maarten Weyn, "A survey of rigid 3D pointcloud registration algorithms," in *Proc. 4th Int. Conf. Ambient Comput. Appl. Serv. Technol.*, 2014, pp. 8–13.

[21] B. Bellekens, V. Spruyt, R. Berkvens, R. Penne, and M. Weyn, "A benchmark survey of rigid 3D point cloud registration algorithms," *Int. J. Advances Intell. Syst.*, vol. 8, no. 1/2, pp. 118–127, 2015.

[22] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot*, vol. 4, pp. 1–104, May 2015.

[23] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan, "A robust point-matching algorithm for autoradiograph alignment," *Med. Image Anal.*, vol. 1, no. 4, pp. 379–398, 1997.

[24] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[25] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, pp. 558–569.

[26] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Math. Methods Biomed. Image Anal.*, 2000, Art. no. 190.

[27] A. Myronenko, X. Song, and M. Á. Carreira-Perpiñán, "Non-rigid point set registration: Coherent point drift," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1009–1016.

[28] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[29] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2003, pp. 2743–2748.

[30] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[31] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3D data," *Image Vis. Comput.*, vol. 17, no. 2, pp. 135–147, 1999.

[32] S. Druon, M.-J. Aldon, and A. Crosnier, "Pair-wise registration of 3D/color data sets with ICP," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 663–668.

[33] M. Korn, M. Holzkothen, and J. Pauli, "Color supported generalized-ICP," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, Jan. 2014, pp. 592–599.

[34] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1511–1516.

[35] H. Men and K. Pochiraju, "Hue-assisted automatic registration of color point clouds," *J. Comput. Des. Eng.*, vol. 1, no. 4, pp. 223–232, 2014.

[36] S. Druon, M. J. Aldon, and A. Crosnier, "Color constrained ICP for registration of large unstructured 3D color data sets," in *Proc. IEEE Int. Conf. Inf. Acquisition*, Aug. 2006, pp. 249–255.

[37] J. H. Joung, K. H. An, J. W. Kang, M. J. Chung, and W. Yu, "3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 3082–3088.

[38] B. Huhle, M. Magnusson, W. Strasser, and A. J. Lilienthal, "Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 4025–4030.

[39] H. Hong and B. Lee, "Colored point cloud registration with improved hue-assisted normal distributions transform," *Int. J. Comput. Theory Eng.*, vol. 8, no. 1, 2016, Art. no. 63.

[40] V. Golyanik, S. A. Ali, and D. Stricker, "Gravitational approach for point set registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5802–5810.

[41] W. Nolting, *Grundkurs Theoretische Physik 1*, 10 ed. Berlin, Germany: Springer-Verlag, 2013.

[42] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3. New York, NY, USA: Wiley, 2006.

[43] R. M. Dreizler and C. S. Lüdde, *Theoretical Mechanics - Theoretical Physics 1*, 1st ed. Berlin, Germany: Springer-Verlag, 2010.

[44] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *J. Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[45] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optimization Theory Appl.*, vol. 45, no. 1, pp. 41–51, 1985.
[46] The MathWorks, Inc., Natick, Massachusetts, USA, *Computer Vision System Toolbox 2016a*, Revised for Version 7.1,2016, https://de.mathworks.com/help/pdf_doc/vision/vision_ref.pdf
[47] M. Kjer and J. Wilm, *Iterative Closest Point*. Technical University of Denmark, 2012. [Online]. Available: https://mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point
[48] H. M. Kjer and J. Wilm, "Evaluation of surface registration algorithms for pet motion correction," Bachelor's Thesis, Dept. Informat. Math. Modelling, Tech. Univ. Denmark, Kongens Lyngby, Denmark, 2010.
[49] A. Myronenko, *Coherent Point Drift (CPD) Matlab Package (Version 2.1)*. Oregon Health and Science University, 2009. [Online]. Available: https://sites.google.com/site/myronenko/research/cpd
[50] M. Levoy, J. Gerth, B. Curless, and K. Pull, *The Stanford 3D Scanning Repository*, 2005. [Online]. Available: http://www-graphics.stanford.edu/data/3dscanrep
[51] G. Turk and B. Mullins, *Large Geometric Models Archive*. Georgia Inst. Technol., 2007. [Online]. Available: http://www.cc.gatech.edu/projects/large_models/

Philipp Jauer received the bachelor's degree, in 2010, and the MSc degree in computer science (with specialisation on robotics and automation), in 2012. He is working toward the doctoral degree focusing on real-time computations in robotics and imaging at the University of Lübeck. In 2012, he joined the Institute for Robotics and Cognitive Systems, University of Lübeck, where he is working as a research assistant.

Ivo Kuhlemann received the BEng degree in production engineering, in 2011, and the MSc degree in medical engineering science from the University of Lübeck, in 2013. He is working toward the PhD degree in robotics in the Institute for Robotics and Cognitive Systems, University of Lübeck. His research focuses on high redundant robot kinematics.

Ralf Bruder received the degree in mathematics and computer science from the Westfälische Wilhelmsuniversität Münster and Technische Universität Clausthal. In 2006, he joined the Institute for Robotics and Cognitive Systems, where he develops medical sensor systems for automated interventions.

Achim Schweikard received the degree in mathematics and computer science from Hamburg University, Freiburg University and University of Paris XI, and the PhD and state doctorate degrees from the Technical University of Berlin, in 1989. After that, he worked as a research associate with the Department of Computer Science, Stanford University, within a joint appointment between Stanford's Departments of Neurosurgery and Computer Science. In 1994, he joined the Technical University of Munich as an associate professor. In 2002, he joined the University of Lübeck as a full professor.

Floris Ernst received the degree in mathematics from Friedrich-Alexander-University Erlangen/Nuremberg and the University of Otago (Dunedin/New Zealand), the PhD degree from the Institute for Robotics and Cognitive Systems, University of Lübeck, in 2011, and the habilitation degree from the Institute for Robotics and Cognitive Systems, in 2016. From 2011 to 2012, he worked as a software engineer at an engineering consultancy, specialising on applications in medicine and life sciences. In 2013, he returned to the Institute for Robotics and Cognitive Systems as a senior research associate and now is a full professor. He is a member of the IEEE.