Fast Gravitational Approach for Rigid Point Set Registration With Ordinary Differential Equations

SK AZIZ ALI[1,2]**, KEREM KAHRAMAN**[2]**, CHRISTIAN THEOBALT**[3]**,
DIDIER STRICKER**[1,2]**, AND VLADISLAV GOLYANIK**[3]

[1]Department of Computer Science, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence GmbH (DFKI), Augmented Vision Group, 67663 Kaiserslautern, Germany
[3]Max Planck Institute for Informatics, Saarland Informatics Campus, 66123 Saarbrücken, Germany

Corresponding authors: Sk Aziz Ali (sk_aziz.Ali@dfki.de) and Vladislav Golyanik (golyanik@mpi-inf.mpg.de)

ABSTRACT This article introduces a new physics-based method for rigid point set alignment called Fast Gravitational Approach (FGA). In FGA, the source and target point sets are interpreted as rigid particle swarms with masses interacting in a globally multiply-linked manner while moving in a simulated gravitational force field. The optimal alignment is obtained by explicit modeling of forces acting on the particles as well as their velocities and displacements with second-order ordinary differential equations of n-body motion. Additional alignment cues can be integrated into FGA through particle masses. We propose a smooth-particle mass function for point mass initialization, which improves robustness to noise and structural discontinuities. To avoid the quadratic complexity of all-to-all point interactions, we adapt a Barnes-Hut tree for accelerated force computation and achieve quasilinear complexity. We show that the new method class has characteristics not found in previous alignment methods such as efficient handling of partial overlaps, inhomogeneous sampling densities, and coping with large point clouds with reduced runtime compared to the state of the art. Experiments show that our method performs on par with or outperforms all compared competing deep-learning-based and general-purpose techniques (which do not take training data) in resolving transformations for LiDAR data and gains state-of-the-art accuracy and speed when coping with different data.

INDEX TERMS Rigid point set alignment, gravitational approach, particle dynamics, smooth-particle masses, Barnes-Hut 2^D-Tree.

I. INTRODUCTION

Rigid point set registration (RPSR) is essential in many computer vision and computer graphics tasks such as camera pose estimation [1], 3D reconstruction [2], CAD modeling, object tracking and simultaneous localization and mapping [3], [4] and autonomous vehicle control [5], to name a few. Suppose we would like to merge partial 3D scans obtained by structured light into a single and complete 3D reconstruction of a scene, identify a pre-defined pattern in the 3D data or estimate the trajectory of the sensor delivering 3D point cloud measurements. All of these tasks can be addressed by a robust RPSR approach which can cope with partially overlapping and noisy data.

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao.

The objective of *pairwise* RPSR is, given a pair of unordered sets of points generally in 2D, 3D, or higher-dimensional space, to find optimal rigid transformation parameters (*e.g.,* 6DOF in 3D, rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$) aligning the *template* point set to the fixed *reference* point set. One of the earliest method classes — iterative closest points (ICP) [6], [7] — is still among the most widely-used techniques nowadays, due to its simplicity and speed. In ICP, the problem of RPSR is converted to transformation estimation between points with known point correspondences. In each iteration, correspondences are selected according to the nearest neighbor rule. Nevertheless, ICP is not the ideal choice in many challenging scenarios with noise, partial overlaps or missing entries in the data, due to its inherent sensitivity to all these disturbing effects and the deterministic correspondence selection rule.
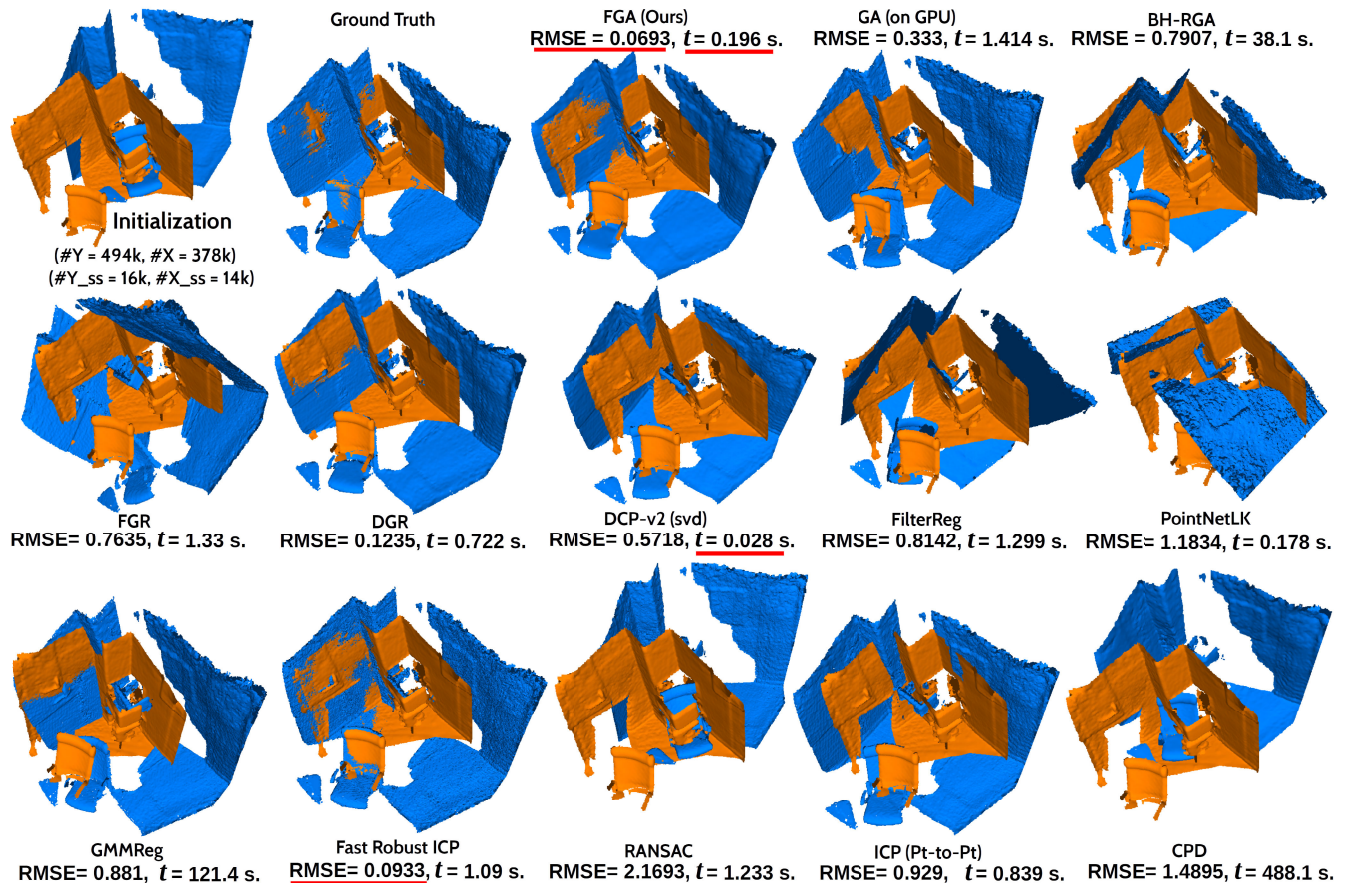
FIGURE 1. A comparison between several standout rigid registration methods (state-of-the-art) on a randomly selected pair of low-overlapping point clouds from popular 3DMatch [10] dataset. #Y and #X are the initial point sizes of the source and target scans. Due to the computational constraint, the corresponding subsampled versions of the inputs (with point sizes $\#Y_{ss}$ and $\#X_{ss}$) are used in the tests. The comparison shows alignment accuracy as root mean squared error (RMSE) of distances between actual and transformed source, and runtime of each tested method. The RMSE metric, which is mesured as $(\frac{1}{\#Y}\sum_{i=1}^{\#Y}\|\mathbf{R}_{gt}\mathbf{Y}+\mathbf{t}_{gt}-\mathbf{R}^*\mathbf{Y}-\mathbf{t}^*\|_2^2)^{1/2}$, defines the alignment accuracy based on the ground-truth transformation ($\mathbf{R}_{gt}$, $\mathbf{t}_{gt}$) and estimated transformation ($\mathbf{R}^*$, $\mathbf{t}^*$). The red underline markers highlight the best case scenarios. Our method FGA is compuationally fast and robust compared to GA [11], BHRGA [12], FGR [13], DGR [14], DCP-v2 [15], FilterReg [16], PointNetLK [17], Fast Robust ICP [18], point-to-point ICP [6], RANSAC [19], GMMReg [20] and CPD [21].

To tackle these difficulties, many other techniques for RPSR were subsequently proposed over the last decades [8], [9].

In recent times, RPSR methods relying on physical analogies [11], [12], [22], [23] are emerging. They offer an alternative perspective to the problem and can often successfully handle cases which are difficult for other algorithmic classes. Physics-based methods have been successful in many domains of computer vision [24]–[26] and have multiple advantages over other method classes. Moreover, by using physical principles, we have access to a large volume of research on computational physics. For instance, we can borrow data structures and acceleration techniques which were successfully applied in numerical simulations [27], [28].

Although RPSR is a well-studied research area, we believe that further advances are possible here with physics-based techniques. With new sensors (*e.g.*, LiDAR), the spatial properties like sampling accuracy and density or physical properties like light-reflectance and color-consistency of the point clouds are considerable factors for the current alignment

techniques. Thus, automotive applications require real-time methods for aligning large, partially overlapping data with outliers and inhomogeneous point densities.

A. CONTRIBUTIONS

In this article, we propose a new physics-inspired approach to rigid point set alignment — Fast Gravitational Approach (FGA, Sec. IV) — which is our core contribution. In FGA, point sets are interpreted are particle swarms with masses moving under the simulated gravitational force field induced by the reference as depicted in Fig. 1. The consecutive states of the template are obtained by explicit modeling of Newtonian particles dynamics and solving for displacements and velocities of the particles with second-order ordinary differential equations (ODEs). In contrast to methods based on correspondences selection and filtering [13], [19], [29], FGA is a correspondence-free approach, *i.e.*, its energy function is defined in terms of interactions between all template and reference points. Thus, our method is globally-multiply linked and has properties not found in other related algorith-

mic classes (*e.g.,* high robustness to noise, versatile applicability of point masses and the fact that the locally-optimal alignment is reached when the gravitational potential energy of the system is locally-minimal).

This article is partially based on our published conference paper [11], while featuring further contributions towards the accuracy gain, computational complexity reduction, handling inhomogeneity in point clouds and feature-based boundary conditions for tackling partially overlapping data. To summarize:

- Besides the general settings of particle interactions and dynamics in [11] (Secs. IV-A–IV-D), we propose an acceleration technique with point clustering, *i.e.,* Barnes-Hut (BH) tree [30] (Secs. IV-E–IV-F) which reduces the algorithm's quadratic computational complexity to quasilinear. Especially for large point sets as those arising in automotive and augmented reality applications, FGA without acceleration can become prohibitive. Subsampling can alleviate the problem but leads to data loss (especially of high-frequency details) as, ideally, one would like to use all available data. In contrast, our acceleration technique preserves the explicit influence of all available data points as well as the globally multiply-linked point interactions.
- Next, we show that particle masses in FGA can be initialized using different types of boundary conditions such as prior correspondences and feature-based alignment cues (Sec. IV-G). Thus, we propose a *normalized intrinsic volume* (NIV) measure per point to be assigned as their mass. This is an effective weighting scheme which smoothly balances the inhomogeneous point sampling density. Similarly, if some matches are given, radially symmetric weights can be assigned to the masses via a radial basis function (RBF) [31]. In the case of partially overlapping data with some known prior correspondences, the Hadamard product of RBF and NIV values as the assigned masses makes the method robust and addresses the local minima issue. Other point features like Fast Point Feature Histogram (FPFH) [19] are also adaptable for the point mass initialization.
- As it has been recently shown, resolving the uniform scale difference, *i.e.,* the seventh DOF is not possible while using the globally multiply-linked interaction policy [32]. Thus, in 3D 6DOF pose estimation, we perform an extensive evaluation of FGA against multiple widely-used and state-of-the-art rigid point set alignment methods and show applications of FGA to LiDAR-based odometry [33] and completions of real RGB-D scans [34] (Sec. V). FGA outperforms general-purpose RPSR methods in scenarios with large amounts of noise and partial overlaps. We can cope with large samples originating from the modern LiDAR sensors — containing ∼400*k* points each — and can support point cloud based odometry at ∼1.5 pairs per second with our GPU version of FGA. In contrast, other

competing methods, to the best of our knowledge, cannot cope with such large data in such short time.

B. STRUCTURE OF THE ARTICLE

The rest of the article is organized as follows. In Sec. III, we first review the classical n-body problem and show how it can be adapted for RPSR. FGA is introduced in Sec. IV. We define our gravitational model with second-order ODEs and simulation rules in Secs. IV-A–IV-E. Details on BH tree building and integration into FGA for accelerated alignment can be found in Sec. IV-F. Embedding boundary conditions through masses is elaborated in Sec. IV-G. FGA is then evaluated against multiple RPSR methods in Sec. V, followed by discussion and conclusive remarks.

II. RELATED WORK

Related work on point set alignment is vast and can be reviewed from different perspectives. We classify the methods according to whether they alternate between the correspondence search and transformation estimation (Sec. II-A), rely on probabilistic modeling (Sec. II-B) of input data, are deep-learning-based (Sec. II-C) or are physics-based (Sec. II-D).

A. FROM TRANSFORMATION ESTIMATION TO ICP

Some approaches [13], [19], [29], [35] first extract a sparse set of descriptive key points from point sets [36], [37] and then find optimal alignment parameters with a transformation estimation approach [38]–[42]. This policy does not use all available points and often leads to coarse alignments but, on the other hand, can result in a significantly improved initialization for other RPSR approaches [36]. Iterative Closest Point (ICP), pioneered by Besl and McKay [6] as well as Chen and Medioni [7], is an approach alternating between correspondence search and transformation estimation.

Various modifications of ICP have been introduced over the years [43]–[46] to tackle its local minima trapping issue. Segal and coworkers [47] extended the classical ICP with probabilistic transformation estimation. In [9], a comprehensive overview of ICP variants is available.

B. PROBABILISTIC METHODS

Probabilistic approaches assign a probability of being a valid correspondence to point pairs [20], [21], [48]. Chui and Rangarajan [48] interpret point set alignment as a mixture density estimation problem. Their Mixture Point Matching (MPM) approach iteratively updates probabilistic correspondences and the transformation with annealing and Expectation-Maximization (EM) schemes, respectively. Coherent Point Drift (CPD) [21] models the template as a Gaussian Mixture Model (GMM) which is fit to the reference interpreted as data points. FilterReg [16] pursues an alternative approach, *i.e.,* the reference induces a GMM. This results in a simpler, faster and — in some scenarios — a more accurate algorithm compared to CPD. In contrast to [16], [21], GMM Registration (GMMReg) [20] interprets both point

sets as GMM, and point set alignment is posed as mixture density alignment. The approach of Tsin and Kanade [49] finds a configuration with the highest correlation between point sets leading to the optimal alignment. Eckart *et al.* align inhomogeneous point clouds with hierarchical GMM [50]. Several approaches additionally use alignment cues as prior correspondences or colors of point clouds [51]–[54]. Our formulation integrates prior correspondences and point features by mapping them to point masses.

C. DEEP LEARNING APPROACHES

Multiple techniques with deep neural networks (DNN) for point cloud processing tasks (*e.g.,* classification and segmentation [55], [56] or shape matching [57]–[59]) have been recently proposed. RPSR approaches using DNNs have also appeared recently in numbers [15], [17], [60]–[64]. Most of them [15], [17], [61], [62] utilize PointNet [55] as a deep feature extractor and feature matching layers for estimating rigid transformations. In contrast, Deep Global Registration (DGR) [14] — which is a data-driven version of Fast Global Registration (FGR) [13] — uses 3D U-Net type feature extractors and a differentiable weighted Procrustes approach.

For all these DNN-based methods, the question of the generalizability to point sets with arbitrary and different point cloud characteristics — such as point set density and volumetric sampling (*e.g.,* in the case of a volumetric 3D scan of a human brain, in contrast to point sets representing surfaces) — remains open. We assume that no training data is available and primarily (except for Sec. V-A) compare our technique to the methods making the same assumptions.

D. PHYSICS-BASED APPROACHES

Physics-based methods are also popular for many computer vision and graphics tasks. These methods interpret inputs as physical quantities and transform the data according to physical laws [65]–[67]. Several algorithms from different domains of computational science use the law of universal gravitation [25], [26], [65], [66], [68]–[70].

A modification of Wright's gravitational clustering [65] algorithm was used for image segmentation [24]. The method of Sun *et al.* for edge detection [66] — later improved by Lopez-Molina *et al.* [69] — was shown to be more robust in scenarios with noise, compared to several competing methods. A gravitational analogy was also applied in image smoothing [70]. Recently, [25] proposed a weighting scheme with a gravitational model for stereo matching.

Jauer *et al.* [23] developed a framework for RPSR based on the laws of mechanics and thermodynamics. Similarly to our FGA, they model point clouds as rigid bodies of particles and additionally support arbitrary driving forces such as gravitational or electromagnetic, as well as repulsive forces. To reduce the runtime, they apply simulated annealing along with the Monte-Carlo re-sampling and calculate particle interactions in parallel on a GPU. This policy does not improve upon the quadratic complexity. In contrast, we reduce the computational complexity of particle

interactions to quasilinear, and our parallelization on a GPU further lowers the runtime. At the same time, we obtain positional updates for particles by solving second-order ODEs.

BHRGA [12] is another gravitational approach for RPSR that modifies laws of gravitational attraction and converts it to non-linear least squares (NLLS) optimization problem with globally multiply-linked point interactions [12]. Similar to our method, they also adapt a BH tree [30]. Relying on a Gauss-Newton solver for NLLS has both advantages and downsides. Even though [12] requires a smaller number of iterations until convergence on average compared to FGA, our method enables fine-grained control over the parallelization and is tailored for a single GPU. We require ~1 second for LiDAR data alignment, whereas BHRGA needs ~1.5 minutes [12] for the inputs of the same size, *i.e.,* an improvement of two orders of magnitude.

In another physical interpretation, [71] solves small correspondence problems on point sets using quantum annealing [72].

III. N-BODY SIMULATIONS

The n-body problem is defined for a system of *n* astrophysical particles in a state of dynamic equilibrium following Newton's law of gravitational interactions [73]. For a two-body system, *the total work done by the gravitational force of attraction* $\mathbf{F}_i$ *by a stationary particle (at position* $\mathbf{r}_j$ *with mass* m_j*) in bringing the other particle (at position* $\mathbf{r}_i$ *with mass* m_i*) towards it by displacing a distance of r units is defined as the Gravitational Potential Energy (GPE)* $\mathbf{E}_i = -\int_{\mathbf{r}_i}^{\mathbf{r}_j} \mathbf{F}_i dr = -\int_{\mathbf{r}_i}^{\mathbf{r}_j} \frac{Gm_im_j}{r^2} dr$. Analogously, for the case of *n* particles the total GPE of an n-body system is defined as:

$$\mathbf{E} = -\sum_{\substack{i,j=1 \\ i \neq j}}^{n} \int_{\mathbf{r}_i}^{\mathbf{r}_j} \frac{Gm_im_j}{r^2} dr. \tag{1}$$

The total gravitational force $\mathbf{F}_i$ exerted on the particle *i* by the remaining $n-1$ particles can be expressed as the sum of the negative gradients of the total GPE $\phi(\mathbf{r}_i, t)$ and an external potential ϕ_{ext} [74]:

$$\mathbf{F}_i = \underbrace{-\sum_{j=1, j \neq i}^{n} \frac{Gm_im_j(\mathbf{r}_i - \mathbf{r}_j)}{\left(\|\mathbf{r}_i - \mathbf{r}_j\|^2 + \epsilon^2\right)^{3/2}}}_{\nabla\phi(\mathbf{r}_i, t)} - \nabla\phi_{ext}(\mathbf{r}_i), \tag{2}$$

where ∇ denotes the gradient operator and $\|\cdot\|$ denotes ℓ_2-norm. The instantaneous system's state is defined by *n* position ($\mathbf{r}_i$) and velocity ($\dot{\mathbf{r}}_i$) vectors at time *t*. The force softening parameter ϵ helps to avoid degeneracy inside the interaction region, *i.e.,* when $\|\mathbf{r}_i - \mathbf{r}_j\| \leq \epsilon$. Absence of ϵ also indicates collisional particle interactions. $-\nabla\phi_{ext}$ accounts for any probable external forces due to the friction or other annealing factors in the system. This friction dissipates the fraction η of the particle momentum. After solving the second-order ODEs of motion, the acceleration

$$\ddot{\mathbf{r}}_i = \frac{\mathbf{F}_i}{m_i} \tag{3}$$

provides the updated velocity and displacement as single and double integrals ($\int \ddot{\mathbf{r}}_i(t)dt$ and $\int\int \ddot{\mathbf{r}}_i(t)dt$) over time, and the trajectories of *n* particles are obtained in this phase space.[1] The interactions between particles can either be *collisional* or *collisionless*. The total energy, kinetic and potential, of every particle is conserved for the collisionless interactions or redistributed when collisions occur. The rules of the energy exchange and altered kinematics in collisional interactions [75] will force the particles to collapse and cause topological degeneracy. For this reason, collisional dynamics is beyond the scope of the RPSR problem domain.

IV. THE PROPOSED APPROACH

This section describes our FGA with algorithmic steps on (i) how a constrained n-body simulation is fitted to define the dynamics of the source object in Sec. IV-A, (ii) the use of ODEs of dynamics to obtain minimum GPE for locally-optimal alignment between the source and the target in Secs. IV-B–IV-D, (iii) acceleration scheme for the whole process of n-body simulation in Secs. IV-E–IV-F and, finally, (iv) defining boundary conditions for our energy function using mass initialization policies based on shape descriptors in Sec. IV-G.

A. NOTATIONS AND ASSUMPTIONS

In FGA, $\mathbf{Y}_{M\times D} = (\mathbf{Y}_1 \ldots \mathbf{Y}_M)^\mathsf{T}$ and $\mathbf{X}_{N\times D} = (\mathbf{X}_1 \ldots \mathbf{X}_N)^\mathsf{T}$ are unordered sets of *D*-dimensional *M template* points and *N reference* points, where $\mathbf{Y}_i$ and $\mathbf{X}_j$ denote elements with indices *i* and *j* from respective point sets. The following parameters need to be set for n-body simulation which are inherited in our FGA and explained in Sec. III:

- G — **gravitational constant**,
- ϵ — near field **force softening length**,
- η — constant system **energy dissipation rate**,
- Δt — **time integration step** for the ODEs of motion,
- $m_{\mathbf{Y}_i}$ and $m_{\mathbf{X}_j}$ — **point masses** of $\mathbf{Y}_i$ and $\mathbf{X}_j$.

The proposed method considers the general settings for particle interactions from [11]. It takes $\mathbf{X}$ and $\mathbf{Y}$ as input sets of points and interprets them as $(M + N)$-body system, where $\mathbf{Y}$ is optimally registered to $\mathbf{X}$ by estimating the rigid transformation tuple $(\mathbf{R}, \mathbf{t})$. For the registration purpose, several assumptions and modifications are made for our $(M + N)$-body system:

i) Every point represents a particle with a mass condensed in an infinitely small volume;

ii) A static reference $\mathbf{X}$ induces a constant inhomogeneous gravitational force field and its points do not interact with each other;

iii) Particles $\mathbf{Y}_i$ move in the gravitational force field induced by all $\mathbf{X_j}$ and do not affect each other;

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

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*) are considered as algorithm parameters;

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

B. GRAVITATIONAL POTENTIAL ENERGY (GPE) FUNCTION OF FGA

The locally-optimal rigid alignment is achieved at the system's state with locally minimal GPE as shown in Fig. 2. If the system of particles $\mathbf{Y}$ moves as a rigid body, the optimal alignment is achieved when the GPE between $\mathbf{Y}$ and $\mathbf{X}$ is minimum. We solve the rigid transformation estimation problem, *i.e.,* estimating rigid rotation $\mathbf{R}$ and translation $\mathbf{t}$, applying rigid body dynamics [76] on the displacement fields of the points in $\mathbf{Y}$. The GPE to rigidly move $\mathbf{Y}$ from its starting position towards $\mathbf{X}$ is expressed as a weighted sum of *inverse multiquadric* functions on the distance fields:

$$\mathbf{E}(\mathbf{R}, \mathbf{t}) = -G \sum_{i,j} \frac{m_{\mathbf{Y}_i} m_{\mathbf{X}_j}}{(\|\mathbf{R}\mathbf{r}_{\mathbf{Y}_i} + \mathbf{t} - \mathbf{r}_{\mathbf{X}_j}\| + \epsilon)}. \quad (4)$$
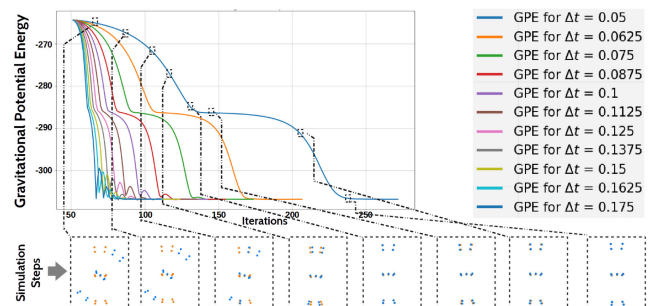

FIGURE 2. GPE function plot with different optimization momenta handled by the increasing time differential (time step Δt). *Simulation steps* depict the reference-template configuration at the current iteration.

Since the energy function in Eq. (4) is inverse multiquadric, non-linear least-squares optimization methods do not fit. Also note, there exists a singularity (without ϵ) at the optimal state of $\mathbf{Y}$ as the distance field in the denominator approaches zero (whereas $\mathbf{E} \to -\infty$). We minimize (4) using second-order ODEs of motion for particle dynamics of $\mathbf{Y}$, and iteratively recover rigid transformations on its successive states.

C. ODEs OF PARTICLE MOTION (NEWTONIAN DYNAMICS)

The assumption (ii) constrains $\mathbf{X}$ to remain idle at a fixed position like a single body, formed by a set of non-interactive points, which attracts $\mathbf{Y}$. On the other hand, the assumption (iii) allows every $\mathbf{Y}_i$ to be attracted by $\mathbf{X}$ only. To this point, the dynamics of $\mathbf{Y}$ remains unconstrained, whereas relative positions of $\mathbf{Y}_i$ will change at every time step. The momentum of $\mathbf{Y}_i$ (from the intermediate previous state) will be preserved, and motion is tractable as the assumption (v) restricts any form of merging or splitting of the points in $\mathbf{Y}$.

---

[1] the space spanned by all possible particle states

Similar to Eq. (2), the gravitational force of attraction exerted on $\mathbf{Y}_i$ by all the static particles from $\mathbf{X}$ reads as:

$$\mathbf{F}_{\mathbf{Y}_i} = -G \sum_{j=1}^{N} \frac{m_{\mathbf{X}_j} m_{\mathbf{Y}_i} (\mathbf{r}_{\mathbf{Y}_i} - \mathbf{r}_{\mathbf{X}_j})}{(\|\mathbf{r}_{\mathbf{Y}_i} - \mathbf{r}_{\mathbf{X}_j}\|^2 + \epsilon^2)^{3/2}}. \quad (5)$$

Since gravitational force depends on the initial and final positions of a particle pair, it conserves the total mechanical energy of the system. The total energy conservation in our case is analogous to preserve the momentum of $\mathbf{Y}$ only as $\mathbf{X}$ is static. The force from $\mathbf{X}$, in Eq. (5), will bring the template $\mathbf{Y}$ closer to its centroid and then $\mathbf{Y}$ will endlessly oscillate around that centroid. Modification (vii) allows for draining some energy in the form of heat as $\mathbf{Y}$ moves in a viscous medium. We introduce a dissipative force $\mathbf{F}_{\mathbf{Y}_i}^d$:

$$\mathbf{F}_{\mathbf{Y}_i}^d = -\eta \, \dot{\mathbf{r}}_{\mathbf{Y}_i}, \quad (6)$$

which is proportional to the particle velocity with the factor η. It allows the otherwise endless periodic motion of the template – due to second-order ODEs of motion – to settle. Hence, the resultant force acting on every $\mathbf{Y}_i$ is

$$\mathbf{f}_{\mathbf{Y}_i} = \mathbf{F}_{\mathbf{Y}_i} + \mathbf{F}_{\mathbf{Y}_i}^d. \quad (7)$$

The next simulated state of $\mathbf{Y}$ at the time $t + \Delta t$ in the phase space is obtained by estimating the *unconstrained* velocity

$$\dot{\mathbf{r}}_{\mathbf{Y}_i}^{t+\Delta t} = \dot{\mathbf{r}}_{\mathbf{Y}_i}^{t} + \Delta t \frac{\mathbf{f}_{\mathbf{Y}_i}}{m_{\mathbf{Y}_i}}, \quad (8)$$

and updating the previous $\mathbf{r}_{\mathbf{Y}_i}^t$ with the displacement

$$\mathbf{d}_{\mathbf{Y}_i}^{t+\Delta t} = \Delta t \, \dot{\mathbf{r}}_{\mathbf{Y}_i}^{t+\Delta t} \quad (9)$$

for all template points. The updated velocity and displacement fields of the template points are stacked into matrices:

$$\mathbf{V} = \begin{bmatrix} \dot{\mathbf{r}}_{\mathbf{Y}_1}^{t+\Delta t} & \dot{\mathbf{r}}_{\mathbf{Y}_2}^{t+\Delta t} & \cdots & \dot{\mathbf{r}}_{\mathbf{Y}_M}^{t+\Delta t} \end{bmatrix}^{\mathsf{T}} \quad \text{(velocities)}, \quad (10)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_{\mathbf{Y}_1}^{t+\Delta t} & \mathbf{d}_{\mathbf{Y}_2}^{t+\Delta t} & \cdots & \mathbf{d}_{\mathbf{Y}_M}^{t+\Delta t} \end{bmatrix}^{\mathsf{T}} \quad \text{(displacements)} \quad (11)$$

and, similarly, the force residual and particle mass matrices:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_{\mathbf{Y}_1} & \mathbf{f}_{\mathbf{Y}_2} & \cdots & \mathbf{f}_{\mathbf{Y}_M} \end{bmatrix}^{\mathsf{T}} \quad \text{(forces)}, \quad (12)$$

$$\mathbf{m}_{\mathbf{X}} = \begin{bmatrix} m_{\mathbf{X}_1} & m_{\mathbf{X}_2} & \cdots & m_{\mathbf{X}_N} \end{bmatrix}^{\mathsf{T}} \quad \text{(masses of } \mathbf{X}\text{)}, \quad (13)$$

$$\mathbf{m}_{\mathbf{Y}} = \begin{bmatrix} m_{\mathbf{Y}_1} & m_{\mathbf{Y}_2} & \cdots & m_{\mathbf{Y}_M} \end{bmatrix}^{\mathsf{T}} \quad \text{(masses of } \mathbf{Y}\text{)}. \quad (14)$$

D. RIGID BODY DYNAMICS USING ODEs OF MOTION

Newton-Euler equations of motion in mechanics (first, second and third law of motion with Euler time integration) relate any external force $\mathbf{f}$ with the inertial state of the body. Newtonian mechanics assumes an inertial frame of reference which is fixed and excluded from any external force. According to assumption (ii), the template and the reference are attached to the moving *body-fixed* and the inertial frames of reference, respectively. After one step of our $(M + N)$-body simulation, two states of the template are available — the previous state $\mathbf{Y}$ and the current state $\mathbf{Y} + \mathbf{D}$. To recover a single consensus

rigid transformation between these two states, we solve the *generalized orthogonal Procrustes alignment* problem in the closed form.

Rotation Estimation: Given are point matrices $\mathbf{Y}$ and $\mathbf{Y_D} = \mathbf{Y} + \mathbf{D}$. Let $\mu_{\mathbf{Y}}$ and $\mu_{\mathbf{YD}}$ be the mean vectors of $\mathbf{Y}$ and $\mathbf{Y_D}$ respectively, let $\hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_{\mathbf{Y}}^T$ and $\hat{\mathbf{Y}}_{\mathbf{D}} = \mathbf{Y_D} - \mathbf{1}\mu_{\mathbf{YD}}^T$ be point matrices centered at the origin of the coordinate system and let $\mathbf{C} = \hat{\mathbf{Y}}_{\mathbf{D}}^T \hat{\mathbf{Y}}$ be a covariance matrix. Let $\mathbf{U}\mathbf{S}\hat{\mathbf{U}}^T$ be singular value decomposition of $\mathbf{C}$. Then the optimal rotation matrix $\mathbf{R}$ is given by [38], [41]:

$$\mathbf{R} = \mathbf{U}\Sigma\hat{\mathbf{U}}^T, \text{ where } \Sigma = \text{diag}(1, \ldots, \text{sgn}(|\mathbf{U}\hat{\mathbf{U}}^T|)). \quad (15)$$

Translation Estimation: Once the rotation $\mathbf{R}$ is resolved, such that $\mathbf{R}\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_{\mathbf{D}}$, the current state of template point set can be updated as $\mathbf{Y}_D = \mathbf{R}\mathbf{Y} + (\mathbf{1}\mu_{\mathbf{YD}}^T - \mathbf{R}\mathbf{1}\mu_{\mathbf{Y}}^T)$ where the translation component is:

$$\mathbf{t} = \mathbf{1}\mu_{\mathbf{YD}}^T - \mathbf{R}\mathbf{1}\mu_{\mathbf{Y}}^T. \quad (16)$$

E. ACCELERATION POLICIES

Many acceleration policies can be used for n-body problems. Some studies show specialized hardware configurations for massively parallel n-body simulations. GRAPE-4 (GRAvity PipE) and GRAPE-6 [77] use pipelining of instructions in force computation and position update of particle set sizes up to 10^4 and 10^6, respectively. Logical parallelism of n-body simulation is also achieved on FPGA [78]. Few seminal works have reduced the algorithmic complexity of the n-body problem (*e.g.,* Fast Multi-pole Method (FMM) [27]). This runs the n-body algorithm in $\mathcal{O}(N \log N)$ time, but can also achieve $\mathcal{O}(N)$ at the expense of higher force approximation tolerance. It also has a relatively lower force approximation accuracy than BH method [30]. Although the runtime of FMM is similar to the BH method, we adapt the BH method because its underlying concept is simple and many astrophysical particle simulations using BH method are also successfully ported on FPGA and GPU [28], [79]. The algorithmic steps of our RPSR method are not exactly the same as n-body algorithm. Hence, in this section, we show how BH tree can be applied to FGA. We are the first, to the best of our knowledge, to decompose the quadratic computational complexity of $\mathcal{O}(MN)$ for all-to-all force computations to the quasilinear complexity of $\mathcal{O}(M \log N)$ using the BH force approximation in a gravitational point set alignment method with second-order ODEs.

F. BARNES-HUT FORCE APPROXIMATION

BH-tree-based force computation in FGA requires two steps — BH tree construction and BH force calculation.
BH Tree τ^θ. The construction of τ^θ is a method of hierarchical space partitioning and grouping of particles as described in Alg. 1 [30]. Given a system of particles with their known position vectors, masses and dimensionality (D) as input, this method at first constructs a tree structure of virtual 2^D dimensional lattices, *i.e.,* either *cubical* cells as nodes of an octree for 3D data or *square* cells as nodes of a quad-tree for

2D data. To construct τ^θ, the following steps are performed iteratively until each particle is assigned to its leaf nodes:

i) BH tree partition starts from the center of the particle system as root node, which encapsulates all the points. With this encapsulation, we split the cell in 2^3 cubical (3D) or 2^2 quad (2D) sub-cells. This opens up *three* options on the occupancy status — either the newly-created sub-cells are *empty*, *non-empty* (with multiple particles) or have exactly *one particle*.

ii) For *non-empty* cells, calculate the aggregated masses of the particles as the mass of the cell/node and the *center of mass* (CoM) which is a weighted average of the particle positions and their respective masses. These two attributes and the cell's length are associated with every cell. For a singleton cell, no further action is taken.

iii) For non-empty cells, we progress with further sub-divisions by repeating step i). If duplicate points exist, the partitioning can continue up to infinite depth. To avoid this, we normalize inputs within a numerical range of $[a, b]$ before building the tree as per Alg. 1, and recurse until tree depth of $d = 20$ during construction. This sets a floating-point precision range from zero to $\frac{(b-a)}{2^d}$ on the minimum distance between two particles. The interpretation of this limit on the floating-point precision is that if the distance between two particles in a cell is $< \frac{(b-a)}{2^d}$, we do not branch further into higher depth. If we do not scale the particle system in the range $[a, b]$, an arbitrarily high depth limit (*e.g*, $d = 128$ or 256) can be required to stop further branching. After scaling the input in the range $[a, b]$, we must obtain the values of the estimated translation parameters at the original scale (see our appendix).

Consider the extreme case where the positions of points in the *reference* are uniformly spaced along the axes forming a regular grid type pattern. Then, we can claim that all the nodes at any level of tree depth will be non-empty and contain the maximum possible number of particles inside.

Lemma 1: The maximum number of child nodes up to the depth level d, except the root and the leaves, of the BH tree τ_θ built on $\mathbf{X}_{N \times D}$ with maximum depth $d_\circ$ is equal to $\sum_{d=1}^{d_\circ - 1} (2^D)^d$. Hence, the maximum number of possible nodes $N_\circ$ of a BH tree built on an unstructured point cloud is $N_\circ = (N + \sum_{d=1}^{d_\circ - 1}(2^D)^d + 1)$.

BH Force $\mathbf{F}^\theta$: Approximation of gravitational force between distant particles in a one-to-many fashion, Eq. (5), is computed with tree-based near-field and far-field approximation. For a given particle, we always start from the root node of the BH tree τ^θ. Forces will now be compounded over the child nodes recursively. During the recursion, a child node will be searched in higher depths or not depend upon the node opening criteria or *multi-pole acceptance criteria* θ. In [30], θ is set as a lower bound of the ratio between the length (l) of a cell and the distance (r) between query point to the CoM of that cell. It indicates that particles from a

---

Algorithm 1 Build BH Tree

Input: reference $\mathbf{X}_{N \times D}$, masses $\mathbf{M_X}$, maximum tree depth d

Output: BH Tree τ^θ

$[\mathbf{X}_{min}, \mathbf{X}_{max}] \leftarrow$ compute bounding box (BB) of $\mathbf{X}$

$\tau^\theta \leftarrow \texttt{CreateChild}(\text{root}(\tau^\theta), N, \mathbf{X}_{min}, \mathbf{X}_{max})$

Function $\texttt{CreateChild}(node(\tau^\theta), N, \mathbf{X}_{min}, \mathbf{X}_{max})$:

  if $N > 0$ *and* $depth < 20$ **then**

    if $N > 1$ **then**

      node partition center $\mathbf{o} = \mathbf{X}_{min} + \frac{\mathbf{X}_{max} - \mathbf{X}_{min}}{2}$

      for $i = 1$ **to** 2^D **do**

        $[\mathbf{X}_{min}^i, \mathbf{X}_{max}^i] \leftarrow$ BB using $\mathbf{X}_{min}, \mathbf{X}_{max}, \mathbf{o}$

        $N \leftarrow$ number of points inside $[\mathbf{X}_{min}^i, \mathbf{X}_{max}^i]$

        $\texttt{CreateChild}(node(\tau^\theta).child_i, N, \mathbf{X}_{min}^i, \mathbf{X}_{max}^i)$

    $node(\tau^\theta).l \leftarrow \|\mathbf{X}_{max} - \mathbf{X}_{min}\|_2$

    $node(\tau^\theta).\text{mass} \leftarrow \sum_j m_{\mathbf{X}_j}, \mathbf{X}_j \in [\mathbf{X}_{min}, \mathbf{X}_{max}]$

    $node(\tau^\theta).\text{CoM} \leftarrow \sum_j \frac{\mathbf{X}_j m_{\mathbf{X}_j}}{node(\tau^\theta).\text{mass}}, \mathbf{X}_j \in [\mathbf{X}_{min}, \mathbf{X}_{max}]$

  return τ^θ;

---

node will be merged into one particle (see Fig. 3-(A)-(*right*)) *without* further recursions if

$$\theta > \frac{l}{r}. \qquad (17)$$

Thus, a non-empty and non-singleton node which satisfies this inequality, merges the encapsulated particles inside it into one with a heavier mass located at their CoM. Hence, the sum of the *gravitational force residuals* from the encapsulated particles can now be approximated by the force from the merged one (Fig. 3-(B), red cells). This results in a dropout of the numerical accuracy of forces, but also in a runtime speed-up. According to the problem statement, the system is not self-gravitating. This implies that $\mathbf{X}$ remains static, and the state change of $\mathbf{Y}$ has no effect on $\mathbf{X}$. Thus in FGA, we only build τ^θ once on $\mathbf{X}$ and use it to approximate $\mathbf{F}_{Y_i}$ in Eq. (5) as $\mathbf{F}_{Y_i}^\theta$, for all $\mathbf{Y}_i$ in every iteration. It reduces the memory complexity and the overall runtime as we do not need to build the BH tree multiple times.

G. DEFINING BOUNDARY CONDITIONS

If available, additional alignment cues (*e.g.,* point colors [52] and prior correspondences [51]) can be embedded in several RPSR methods [6], [21] as boundary conditions to guide the alignment. A variant of rigid ICP [80] generalizes the Euclidean distances for the color space, whereas a variant of CPD defines the GMM in the color space of a point cloud [52]. Extrinsic features (*e.g.,* tracked marker locations)
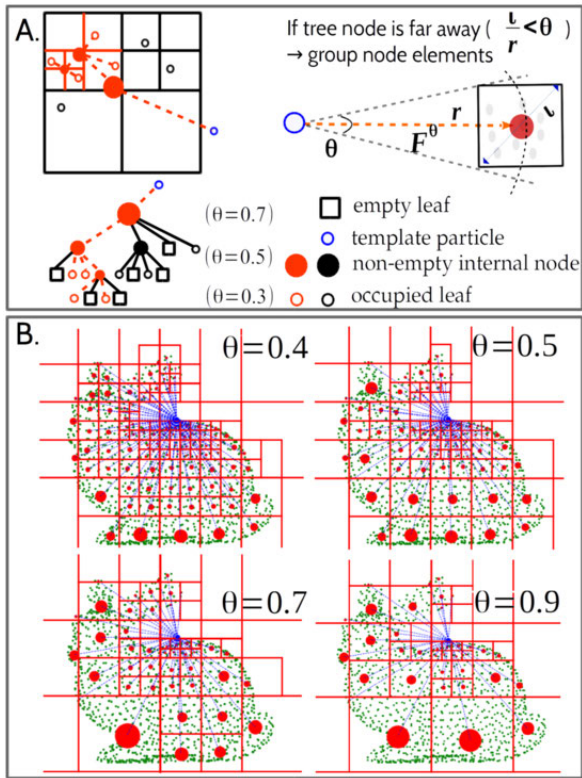
FIGURE 3. (A): BH tree structure with hierarchical cell grouping and pruning depends on the node opening criteria θ. (B): BH force approximation $\mathbf{F}^\theta$ [30] under different values of the node opening parameter θ for a *template* point set (in green color). The red dots represent the CoM of the cells with sizes proportional to the number of points inside the nodes.

over multiple frames [81] can also increase the correspondence reliability during the registration. In contrast, point-based [19] or geometric features [82] extracted from the input can help to boost the correspondence search. Whereas it is seemingly useful for [36] to add feature descriptors from [19] for robust registration, Golyanik *et al.* [51] suggest to define a separate GMM probability density function for a prior set of landmarks. Reference [83] can extract the sensor's viewpoint signatures as well as the underlying geometry descriptors from considerably noisy point clouds, which helps to estimate 6 DOF camera pose in [84].

We define the boundary conditions in two ways, and show how they guide $\mathbf{Y}$ for robust alignment, especially *in the case of partially overlapping and noisy data*. We use (1) a set of known one-to-one correspondences as landmarks and (2) point cloud feature-based weights as point masses. We finally derive a (3) Smooth-Particle Mass (SPM) map using RBFs and NIV measures for assigning weights to each point as its input masses.

1) RBF MASS INTERPOLATION

An RBF [31] is a radially symmetric function around a central point. For any given input vector $\mathbf{Y}_i$, an RBF interpolation using a radial kernel $\Phi : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ with m radial

centroids (suppose these are originating from prior matches $- \mathbf{Y}_{\widehat{c}^1} \dots \mathbf{Y}_{\widehat{c}^m}$, where $C = \{(\widehat{c}^1, \widetilde{c}^1), \dots, (\widehat{c}^m, \widetilde{c}^m)\}$ is a set of correspondence indices with entries $\widehat{c}_Y \in \{1, \dots, M\}$ and $\widetilde{c}_X \in \{1, \dots, N\}$) is defined as:

$$\mathbf{B}(\mathbf{Y}_i) = \begin{cases} \sum_{\widehat{c} \in C} \lambda_{i\widehat{c}} \Phi(\|\mathbf{Y}_i - \mathbf{Y}_{\widehat{c}}\|) & \text{if } C \neq \emptyset \\ 1.0, & \text{if } C = \emptyset. \end{cases} \quad (18)$$

RBFs are invariant under Euclidean transformations, which is suitable for embedding in iterative alignment methods, especially for adaptive mass computation. In FGA, we choose $\Phi(s) = \exp\left(-\frac{s^2}{\sigma^2}\right)$. The computational complexity of evaluating $\mathbf{B}$, which is to determine $\lambda_{i\widehat{c}}$ by collocation on all M *template* points from a sparse set of m anchor points, is $\mathcal{O}(Mm^2)$ ($\mathcal{O}(Nm^2)$ for $\mathbf{X}$). Note that m is usually small. The interpolated masses with three centroids are highlighted in Fig. 4-A-(*middle*).

2) NORMALIZED INTRINSIC VOLUME MEASURE

The first step is to discretize each D-dimensional input point cloud into ϱ^D equispaced lattices $\mathbb{L}$, where $\varrho = 16$ in our method. These lattices contain scattered density information across the domains $\Omega_\mathbf{X}$ and $\Omega_\mathbf{Y}$ of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Each of the non-empty lattice cells in $\mathbb{L}$ that cover only the domain of input point cloud, (*e.g.*, $\Omega_\mathbf{Y}$ of $\mathbf{Y}$) has equal intrinsic volume (or total quermassintegrals V [85]), of either $\mathbb{L}^x \times \mathbb{L}^y$ in 2D or $\mathbb{L}^x \times \mathbb{L}^y \times \mathbb{L}^z$ in 3D. These lattices are $\mathbb{L} \sqcap \mathbf{Y}$,[2] and their combined intrinsic volume is $\int_{\Omega_{\mathbb{L} \sqcap \mathbf{Y}}} V(\mathbb{L})$. Recall that the lattices are either sparsely or densely packed by the convex bodies $\mathcal{B}(\mathbf{Y}_i)$ centered at the locations of all $\mathbf{Y}_i$. Each of these bodies has constant intrinsic volume $V(\mathcal{B}(\mathbf{Y}_i)) = \pi(\frac{b-a}{2d\varrho})^2$ in 2D or $V(\mathcal{B}(\mathbf{Y}_i)) = \frac{4}{3}\pi(\frac{b-a}{2d\varrho})^3$ in 3D. The NIV value of $\mathbf{Y}_i$, contained inside the lattice cell $\mathbb{L}_{\mathbf{Y}_i}$, is expressed as the inverse of the ratio between the $V(\cup_{i \in \mathbb{L}_{\mathbf{Y}_i}} \mathcal{B}(\mathbf{Y}_i))$ and $V(\mathbb{L}_{\mathbf{Y}_i})$, and normalized by $\int_{\Omega_{\mathbb{L} \sqcap \mathbf{Y}}} V(\mathbb{L})$:

$$\mathbf{N}(\mathbf{Y}_i) = \left(\frac{1}{\int_{\Omega_{\mathbb{L} \sqcap \mathbf{Y}}} V(\mathbb{L})} \frac{V(\cup_{i \in \mathbb{L}_{\mathbf{Y}_i}} \mathcal{B}(\mathbf{Y}_i))}{V(\mathbb{L}_{\mathbf{Y}_i})} \right)^{-1}. \quad (19)$$

The area (for 2D) or volume (for 3D) fraction of lattices covered either by the circles (convex bodies for 2D) or spheres (convex bodies for 3D) centered around the enclosed points as the structure descriptors[3] is termed as *normalized intrinsic volume* (NIV) measure. Fig. 4-A-(*right*) shows color-coded NIV values of LiDAR scan points and the NIV measure details are depicted in Fig. 4-B.

---

[2] "$\sqcap$" denotes set intersection on a discretized domain

[3] The motivation of this descriptor comes from the Monte-Carlo Importance Sampling technique — suppose m samples are drawn using some function f from an alternative population (presented by probability density function $q(x)$) of the actual population (presented by probability density function $p(x)$), then the expected value of the samples $\int f(x)p(x)dx$ can also be expressed as $\int \frac{f(x)p(x)}{q(x)} q(x)dx = \mathbb{E}\left[\frac{f(x)p(x)}{q(x)}\right]$. The probability fraction $w(x) = \frac{p(x)}{q(x)}$ are the importance weights, such that the normalized importance is $\sum_{i=1}^m w(x_i) = 1$.
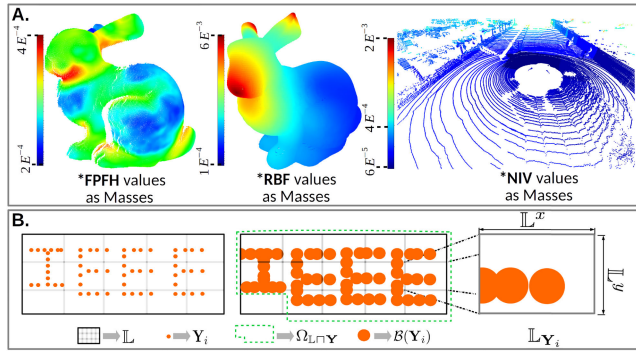
FIGURE 4. (A): The SPM function **S** maps different point-wise feature values to the masses. Shown are the FPFH [19] and the estimated RBF [31] values using landmarks, as well as NIV measures for the mass initialization of *bunny* and KITTI [86] data. (B): A lattice representation of a 2D point set and the visualization of the intrinsic volume defined as the fraction of the cell area occupied by all $\mathbf{Y}_i$-centric balls $\mathcal{B}(\mathbf{Y}_i)$ inside it.

3) SPM: SMOOTH-PARTICLE MASS FUNCTION

FGA assigns point-wise features as their masses. If landmarks are available, we choose strong weights on them and radially interpolated weights on the rest of the points using the RBF map [31] $\mathbf{B}(\mathbf{Y}, C) : \mathbb{R}^{M \times D} \to \mathbb{R}^{M \times 1}$. If no landmarks are available, we use NIV measure $\mathbf{N}(\mathbf{Y}) : \mathbb{R}^{M \times D} \to \mathbb{R}^{M \times 1}$. The values of the feature maps are spatially smooth. In some scenarios when point densities are non-uniform and prior matches are available, then the Hadamard product, symbolized by $\circ$, of RBF and NIV

$$\mathbf{S} = \mathbf{N} \circ \mathbf{B} \qquad (20)$$

assigns balanced weights on $\mathbf{X}$ and $\mathbf{Y}$. Alternatively, we can also map other point-based or geometry-based feature values (*e.g.,* FPFH [19] or CGF [82]) as SPMs to the masses, which updates the matrices $\mathbf{M_X} = \mathbf{S}(X)$ and $\mathbf{M_Y} = \mathbf{S}(Y)$. FGA is summarized in Alg. 2.

H. IMPLEMENTATION DETAILS

Many-body simulation is computationally expensive and demands scalability of its underlying interaction algorithm. Fitting BH algorithm on parallel hardware [79] is not straightforward as irregular tree data structure cannot be mapped easily to modern *uniform memory access* architecture. We modify the C++/CUDA implementation blueprint for BH-tree-based `n-body` simulation [87] method to use in our FGA. We describe only the *force approximation kernel* of [87] which is changed for the CUDA/C++ version of FGA. The rest of the kernels — *bounding box kernel, tree building kernel, node-summarizing kernel, node-sorting kernel* and *rigid alignment kernel* — for FGA are straightforward operations (*i.e.,* either matrix multiplications or sorting).

Following Lemma 1, we allocate global memory blocks for necessary book-keeping tasks on $2^D N_o$ children indices, N_o binary flags for empty or non-empty nodes, N_o starting positions and N_o ending positions of nodes of `integer` type variables. In total, M threads are launched to compute the

---

Algorithm 2 Fast Gravitational Approach

Input: reference $\mathbf{X}_{N \times D}$, template $\mathbf{Y}_{M \times D}$, landmarks
  $\quad C = (\widehat{c}, \tilde{c})$
Output: optimal rigid transformation $\mathbf{T}^*$ registering $\mathbf{Y}$
  $\quad$ to $\mathbf{X}$
Parameters: $\epsilon, \eta, G, \varrho, \Delta t, \theta, \sigma$
Initialization: $\mathbf{T} = [\mathbf{R} = \mathbf{I}_{3\times3} | \mathbf{t} = \mathbf{0}_{3\times1}]$
normalize $\mathbf{X}, \mathbf{Y}$ in the range $[a, b]$
compute $\mathbf{B}(\mathbf{Y}), \mathbf{B}(\mathbf{X})$ using Eq. (18)
compute $\mathbf{N}(\mathbf{Y}), \mathbf{N}(\mathbf{X})$ using Eq. (19)
$\tau^\theta \leftarrow$ build BH tree on $\mathbf{X}$ with SPM $\mathbf{S}(\mathbf{X})$ using Alg. 1
while $\|\mathbf{T}^{t+\Delta t} - \mathbf{T}^{t-\Delta t}\|_F^2 \leq 10^{-4}$ **do**
  $\quad$ compute $\mathbf{F}^t$ using Eqs. (5), (7), and (12)
  $\quad \triangleright$ **for** $k = 1$ **to** M **do**
    $\quad\quad \mathbf{F}_{\mathbf{Y}_k}^\theta \leftarrow \text{BHForce}(\text{root}(\tau^\theta), \mathbf{Y}_k, f = 0)$
  $\quad$ compute $\mathbf{V}^{t+\Delta t}$ and $\mathbf{D}^{t+\Delta t}$ using Eqs. (8)–(11)
  $\quad$ compute transformation $\mathbf{T}^{t+\Delta t} \leftarrow [\mathbf{R}^{t+\Delta t} | \mathbf{t}^{t+\Delta t}]$
  $\quad \triangleright$ solve for $\mathbf{R}^{t+\Delta t}$ using [11], [38], [41], and Eq. (15)
  $\quad \triangleright$ solve for $\mathbf{t}^{t+\Delta t}$ using Eq. (16)
  $\quad \triangleright$ update $\mathbf{Y}^{t+\Delta t} \leftarrow (\mathbf{R}^{t+\Delta t})\mathbf{Y}^t + \mathbf{t}^{t+\Delta t}$
  $\quad \triangleright$ $\mathbf{R} \leftarrow (\mathbf{R}^{t+\Delta t})\mathbf{R}; \mathbf{t} \leftarrow \mathbf{t}^{t+\Delta t} + (\mathbf{R}^{t+\Delta t})\mathbf{t};$
$\mathbf{T}^* = [\mathbf{R} | \mathbf{t}]$
Function $\text{BHForce}(node(\tau^\theta), \mathbf{Y}_k, f)$**:**
  $\quad$ **if** $\frac{node(\tau^\theta).l}{\|\mathbf{Y}_k - node(\tau^\theta).CoM\|} < \theta$ *and* $node(\tau^\theta)$ *not empty*
  $\quad$ **then**
    $\quad\quad f \leftarrow f +$ Force between $\mathbf{Y}_k$ and $node(\tau^\theta)$ using
    $\quad\quad$ Eq. (5)
  $\quad$ **else**
    $\quad\quad$ **for** $i = 1$ **to** 2^D **do**
      $\quad\quad\quad f \leftarrow f + \text{BHForce}(node(\tau^\theta).\text{child}_i, \mathbf{Y}_k, f)$
  $\quad$ **return** f;

---

forces on every template point. In this process, no synchronization barriers are required to summarize forces.

V. EXPERIMENTAL RESULTS

We evaluate FGA on a wide range of real and synthetic datasets with multiple types of data disturbances (*e.g.,* noise and partial overlaps). The experiments cover indoor as well as outdoor scenarios, and we compare FGA to several state-of-the-art approaches from different method classes.

Experimental Datasets: We use the Synthetic *bunny* dataset[4] from the Stanford scans repository in the runtime experiments (the version with ~35k points, see Sec. V-B), as well as for emulating effects of added synthetic noises and other data disturbances (the version with ~1.8k points, see Sec. V-E). ModelNet40 [88] is another synthetic dataset comprised of multiple instances of 40 different object categories (*e.g., plant, vase, toilet, and table*). It has ~9.8k training samples and ~2.4k testing samples to be used by deep

[4]www.graphics.stanford.edu/data/3Dscanrep/

learning methods. In Sec. V-A, we first compare deep learning methods and our FGA (a non-neural method) for the generalizability across different input data. Stanford *lounge*[5] [89] and *Freiburg*[6] [34] datasets contain partial scans of indoor scenes generated using RGB-D sensors. They are widely-used in evaluations of methods for simultaneous localization and mapping (SLAM) where the ratio between the area of intersection and the whole inputs — further referred to as the ratio of overlaps — for the consecutive frames is moderate (*i.e.,* $> 50\%$, and often $> 80\%$). We apply FGA on these two datasets and another more challenging dataset of similar type with indoor scenes 3DMatch [10] (in Sec. V-D), where the ratio of overlaps between the source and target frames is lower (*i.e.,* $< 50\%$, and often $< 30\%$) which makes it highly challenging for point set alignment methods. Next, we use several driving sequences from the *KITTI*[7] [86] and *Ford*[8] [90] datasets which contain non-uniformly sampled point clouds generated using Velodyne LiDAR sensors.

Evaluation Criteria: Bunny dataset provides ground-truth correspondences, and the *lounge* dataset [89] provides ground-truth transformations ($\mathbf{R}_{gt}, \mathbf{t}_{gt}$). We calculate the root-mean-squared error (RMSE) on the distances between registered source and target point clouds with known correspondences and angular deviation φ for the *lounge* dataset, which can be measured as the chordal distance between the estimated ($\mathbf{R}^*$) and ground-truth ($\mathbf{R}_{gt}^\mathsf{T}$) rotations or as an Euler angular deviation [91]:

$$\varphi = \frac{180°}{\pi} \left(\cos^{-1} \left(0.5(\text{trace}(\mathbf{R}_{gt}^\mathsf{T}\mathbf{R}^*) - 1) \right) \right). \quad (21)$$

For KITTI [86] and 3DMatch [10] datasets, we measure the angular deviation φ and Euclidean distance $\Delta\mathbf{t}$ between the translation components $\mathbf{t}^*$ (estimated) and $\mathbf{t}_{gt}$ (ground truth). We also report the total transformation error:

$$\Delta\mathbf{T} = \varphi + \underbrace{\|\mathbf{t}_{gt} - \mathbf{t}^*\|}_{\Delta\mathbf{t}} \quad (22)$$

in the experiments for parameter selection in Sec. V-G, where the angular error φ is a small residual part of $\Delta\mathbf{T}$.

Baseline Methods and Parameter Settings: FGA is a *general-purpose* registration method which does not require training data and which performs equally well on volumetric point clouds and also on data with well-defined surface geometry. We hence focus on comparisons to methods making the same assumptions and report alignment results of CPD [21], GMMReg [20], FilterReg [16], FGR [13], RANSAC [19], point-to-point ICP [6] and GA [11] as well as our FGA. All these methods, like our FGA, do not require information on prior correspondences and any special geometric knowledge about the inputs. However, if available, prior correspondences can be used by FGA as boundary conditions. We also include

[5]www.qianyi.info/scenedata.html
[6]https://vision.in.tum.de/data/datasets/rgbd-dataset
[7]www.cvlibs.net/datasets/kitti/raw_data.php
[8]http://robots.engin.umich.edu/SoftwareData/Ford

a few comparisons on ModelNet40 [88] dataset where deeplearning-based methods such as PointNetLK [17] and Deep Closest Point (DCP) [15] are exposed to generalization gaps, *i.e.,* they perform poorly when tested on data with point disturbances not observed in the training data. We show that such neural methods make strong assumptions about the types of data they can deal with. Hence, they are not in the focus of this article.

FGA runs on a heterogeneous platform with CPU (C++ code) and GPU (in CUDA/C++ code). All experiments are performed on a system with the Intel Xeon E3-1200 CPU with 16GB RAM and NVIDIA 1080 Ti graphics card. We scale our input data in the range $[-5, 5]$ to build the BH tree on the reference $\mathbf{X}$ up to a fixed depth of $d = 20$, as mentioned in Sec. IV-F. In all experiments, RBF kernel width (σ), NIV lattice resolution (ϱ), gravitational constant (G), gravitational force softening length (ϵ), time integration step (Δt), force damping constant (η) and BH-cell opening criteria (θ) are set as — $\sigma = 0.03$, $\varrho = 16$, $G = 66.7$, $\epsilon = 0.2$, $\Delta t = 0.1$, $\eta = 0.2$ and $\theta = 0.6$. Finally, we show through experiments with different datasets that these settings are optimal. FGA converges in $80 - 100$ iterations.

A. FGA AGAINST DEEP LEARNING METHODS

This section provides a detailed analysis of the registration results using state-of-the-art deep learning methods — PointNetLK [17] and DCP [15] on the ModelNet40 [88] dataset. Both methods are trained from scratch (to account for noisy samples) for 250 epochs with a learning rate of 10^{-3} using ADAM optimizer [92]. While a batch size 32 is used for PointNetLK with ten internal iterations, DCP is trained with batch size 10 (as recommended in [15], [17]). Scalability of processing large point clouds is a common problem for both these methods (also, both networks have to use a fixed number of multi-layer perceptrons (MLPs) to match the embedding dimensions). Hence, we subsample all CAD shapes to 2048 points. To enhance the accuracy of both the networks in handling data with disturbing effects, we perform data augmentation. We randomly select 950 CAD objects from the training set of $\sim 9.8k$ samples and merge them into a single extended training set after applying four different types of data disturbances — (i) adding 10% Gaussian noise with zero mean and the standard deviation of 0.02, (ii) adding 10% (out of 2048 points) uniformly distributed noise in the range $[-0.5, 0.5]$, (iii) adding perturbations to the actual point positions with maximum displacement tolerance of 0.01, and finally (iv) removing 20% of the points in a chunk at random. The choice of applying the above four disturbances on any given sample is random (in all of the above cases, the total amount of points remains 2048 for each CAD sample). For comparing the errors, we prepare five different validation sets — originating from the same test set — with increasing levels of the aforementioned data disturbance types — *i.e.,* by adding 1%, 5%, 10%, 20%, and 40% noisy points or cropping the same amount of points.

In spite of training with the additional 950 samples, both DCP and PointNetLK show a common generalizability issue, see the transformation error plots in Fig. 5. The evaluation shows that error metrics of PointNetLK becomes significantly higher when the noise level increases just from 1% to 5% (for Gaussian noise, the rotational error increases from 4.37° to 79.3°, and the translational error increases from 0.0252 to 6.05, whereas for uniform noise, the rotational error increases from 4.29° to 78.42°, and the translational error increases from 0.0249 to 3.43). While DCP approach is more robust than PointNetLK, the noise intolerance issue is still pertinent for it. Our FGA is far more robust compared to both neural approaches, *i.e.,* its transformation estimation errors, especially rotational errors, are consistently smaller by more than 15 times compared to PointNetLK and $\approx$1.2 times compared to DCP. Only the rotational error of FGA is close — but still lower at several increasing noise levels — compared to the DCP's error for cropped inputs.

We conclude that there exist generalization gaps and robustness issues in these neural approaches when tackling noisy data which is often encountered in practical applications. Hence, further in the experimental section, we keep our focus on unsupervised and general-purpose RPSR methods which do not require training data and can generalize across various alignment scenarios.

B. FGA RUNTIME AND ACCURACY ANALYSIS

To evaluate the runtime versus accuracy of FGA, we take a clean *bunny* with $\approx$35k points and subsample it with ten increasing subsampling factors. Ten $(\mathbf{X}, \mathbf{Y})$ pairs are obtained applying random rigid transformations on each $\mathbf{Y}$. Fig. 6 illustrates the computational throughput on a CPU in frames per second and the accuracy as RMSE of FGA (for $\theta = 0.5, 0.6, 0.7$) against other methods, *i.e.,* CPD [21], GMMReg [20], FilterReg [16], FGR [13], RANSAC [19], point-to-point ICP [6] and GA [11]. FGA ranks top in terms of its computational throughput and accuracy for large point set sizes. Only FilterReg [16] and FGR [13] rival our method on a small subset of cases, see Fig. 6-(B) for a bivariate correlation plot for RMSE and the registration speed. Thanks to the full data parallelization of FGA, its GPU version runs $\sim$100 times faster than its CPU version and also outperforms in speed all other tested CPU versions. Note the GPU version of FGA corresponds entirely to the CPU version, and the negligible discrepancy in the RMSE is due to differences in floating-point calculations between CPU and GPU and the possible different number of iterations.

The multi-pole acceptance criteria θ reduces the number of BH tree traversals with its increasing value from 0 to 1. The effect of increasing θ to reduce the computational time of FGA is reflected in Fig. 7. The reason for the speed-up is that the information about the nodes (positions and masses) at higher depths are being summarized by the nodes of the same type at a lower depth. In the range $\theta \in (0, 0.6]$, the loss of accuracy in the gravitational force approximation is negligible. Note that for some datasets with a high percentage
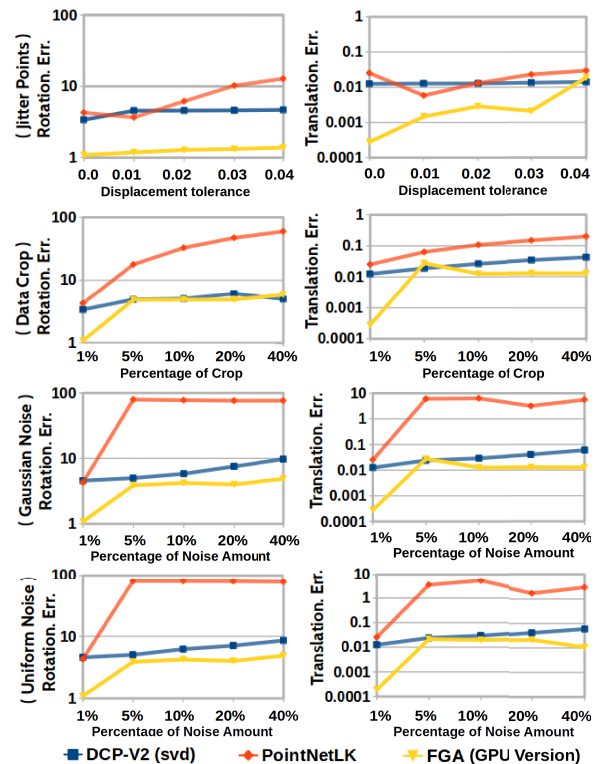

FIGURE 5. The accuracy of our FGA and two deep learning methods PointNetLK [17] and DCP [15] trained on ModelNet40 [88] dataset, with additional 10% of the samples included in the training set after applying *four* different types of data disturbances. Our FGA is compared on *five* different validation sets with increasing levels of data disturbances. The error plots show that FGA outperforms the other two methods and highlight the robustness issues of the learning-based approaches.

of noise and data discontinuities, choosing a higher range of $\theta \in [0.7, 1]$ is not suitable to trade *speed* for *accuracy gain*. Note that a spiking effect is observed on the runtime curve for $\approx$35k points in Fig. 7. The reason is that for larger point sets, the number of opened nodes does not continuously change (increase or decrease) for a continuous change on the θ value. Moreover, due to the same reason, the alignment process can require a different number of iterations to converge.

C. PARTIALLY OVERLAPPING DEPTH DATA

In RGB-D based SLAM, globally-optimal rigid alignment provides camera trajectories by mapping partial scenes. For our quantitative evaluation, we choose the first 400 frames of depth data from Stanford *lounge* [89] dataset. We next perform registration on every fifth frame (see Fig. 8) after down-sampling those to $\approx$5000 points each and report the final Euler angular deviation φ (no prior matches are used in this experiment). Three different success rates of FGA are measured as the percentages of total experimental outcomes when φ is below three different cut-off levels –4°, 3°,, and 2°, respectively. Moreover, we report the average, minimum and maximum angular errors denoted by $\varphi_{avg.}, \varphi_{min}$ and φ_{max}, respectively, see Table 1. On CPU, FGA takes around 1.2 seconds on average until convergence (in $30 - 60$ iterations) and performs as the second most
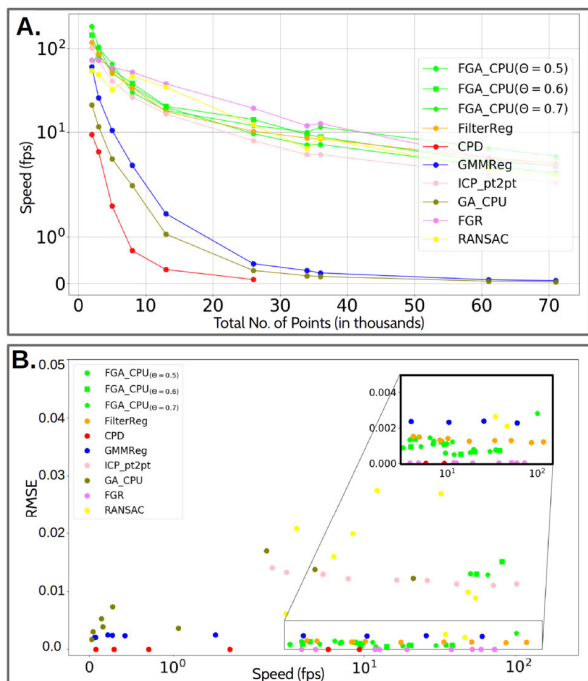
FIGURE 6. **(A:)** Throughput of different methods in frames per second and **(B:)** Bivariate correlation plot for the alignment accuracy and speed, for ten data sizes and random initial misalignments. The bottom right area of the plot, dominated by FGA, reflects the most efficient marks.
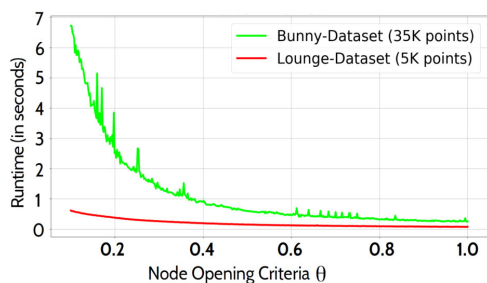

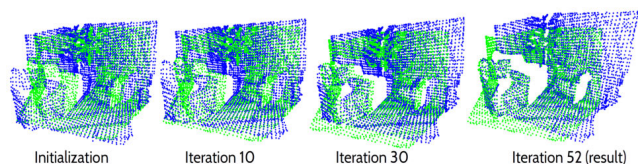**FIGURE 7.** Runtime (in seconds) of FGA for increasing values of the node opening criteria θ in FGA.

Initialization   Iteration 10   Iteration 30   Iteration 52 (result)

FIGURE 8. A sample result of FGA tested with two frames (separated by four consecutive frames) with moderate partial overlaps from the Stanford *lounge* dataset [89].

accurate method after GMMReg [20]. Note that FGA achieves the lowest φ_{min} among all compared methods.

D. PAIRWISE REGISTRATION OF INDOOR SCENES

The 3DMatch [10] benchmark dataset contains eight sequences of minimally-overlapping partial scan pairs of indoor scenes. It is thus highly challenging for unsupervised point set alignment methods. CPD [21], ICP [6],

TABLE 1. The success rate of compared methods for three upper bounds (4°, 3°, 2°) on angular deviation from ground truth after registration.

	Angular Err(φ) $\varphi_{avg.}, \varphi_{min}, \varphi_{max}$	success % ($\varphi < 4°$)	success % ($\varphi < 3°$)	success % ($\varphi < 2°$)
FGA (ours)	2.74, **0.071**, 19.61	**85.8%**	**75.8%**	**65%**
GA [11]	2.85, 0.148, 15.73	77.2%	66%	58.7%
CPD [21]	3.21, 0.075, 32.01	72%	60.8%	51.7%
ICP [6]	3.32, 0.284, 14.92	72.8%	57.5%	46.5%
GMMReg [20]	**1.15**, 0.072, 16.61	**91.2%**	**90.1%**	**90%**
FilterReg [16]	2.67, 0.260, **14.91**	85%	74%	59%
FGR [13]	3.267, 0.214, 16.09	78.8%	63.5%	26.7%
RANSAC [19]	3.498, 0.189, 31.80	71.33%	68.77%	39.1%

GMMReg [20] and FilterReg [16] all perform poorly on partial data and particularly on 3DMatch. Furthermore, subsampling of 3DMatch scans with a voxel size of 3 cm results in point clouds with ~15*k* points which is still computationally expensive for these methods. The evaluation includes 506, 156, 207, 226, 104, 54, 292 and 77 challenging pairs from the 3DMatch sequences. Fig. 9 depicts sample registration results of FGA on pairs of scans from all eight sequences. Fig. 10 shows the total number of successful registrations using FGA and its runtime compared to other benchmark methods on this dataset. We have also chosen a strict upper bound on rotational error ($<5°$) and translational error ($<20cm$) to define the success parameter. Our FGA outperforms all compared methods in the overall number of successful registrations over all eight sequences. In the runtime, FGA outperforms other methods by the factor ranging from $\approx$2.5 to $\approx$4. FGR [13] and RANSAC [19], [93] are ranked second and third in the success rate and outperform other compared methods except for FGA.

Using Prior Correspondences (**1**)*:* A qualitative evaluation is also performed by testing FGA on RGB-D *Freiburg* [34] dataset picking frames 1 and 10 as **Y** ($\approx$220*k* points) and **X** ($\approx$240*k* points). *Four* landmark points — (i) *'red button'* of the laptop, (ii) tip of a *'red pen'*, (iii) corner of a *'document folder'* and (iv) a point near *'red apple'* — are manually selected. They can perhaps be non-exact in their positional accuracy. The scenes are moderately overlapping. The RBF values are higher around the landmarks. The results and the runtime of FGA are shown in Fig. 11.

Using Feature-Based Particle Masses (**2**)*:* The next experiment emphasizes that FGA can use different types of point-based and geometry-based features as input weights (point masses). We set products of the masses of the template and reference points as directly proportional to their gravitational forces of attraction. We choose (a) values of FPFH [19] and (b) the values of our SPM function (this is equivalent to the NIV values with no landmarks) as the masses of structured-light scan point clouds with 159*k* and 145*k* points, respectively. In both cases, FGA registers all input scans accurately. There is a marginally higher accuracy when using NIV values as point features. We also compare FGA and FGR on Apollo scans with partial overlaps, see Fig. 12. The Hausdorff distances sampled on **X** show a higher proportion of blue color on the template's surface.
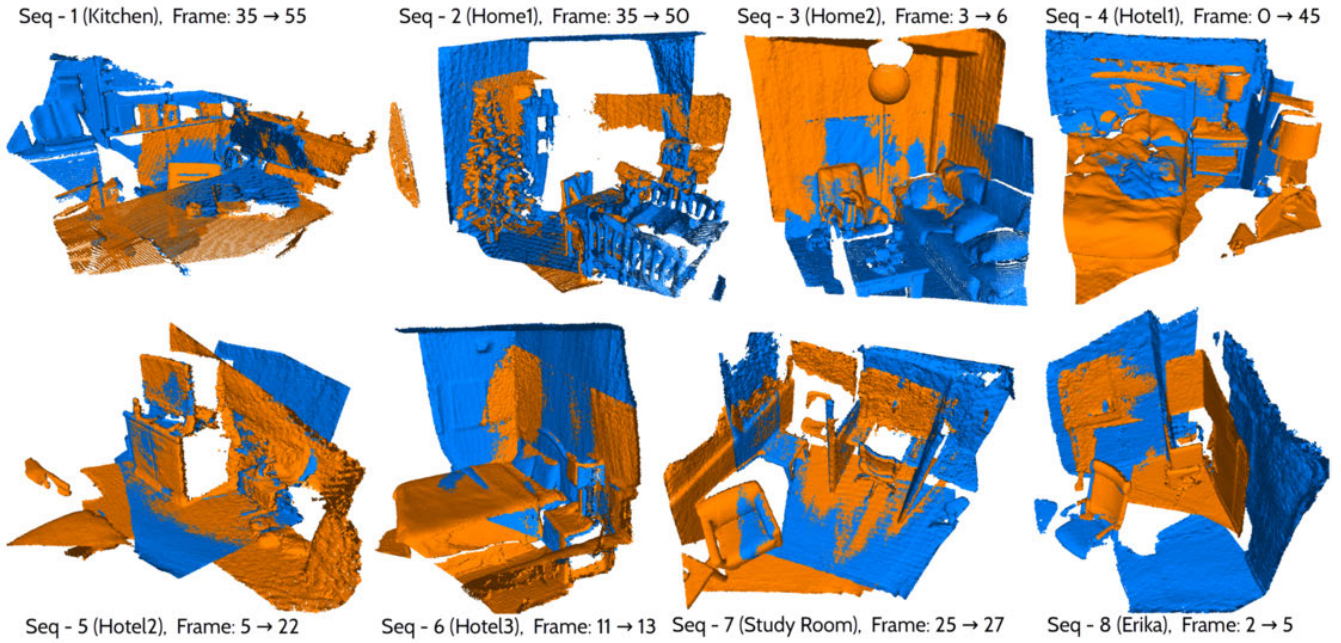
FIGURE 9. Sample pairwise registration results of our method on all eight different test scenes in 3DMatch [10] dataset. The input point set pairs exhibit minimal overlaps and irregular cropped areas. The template Y is colored blue and the reference is colored orange. Results of FGA named by the source-to-target frames ($i \rightarrow j$) are best viewed in color.
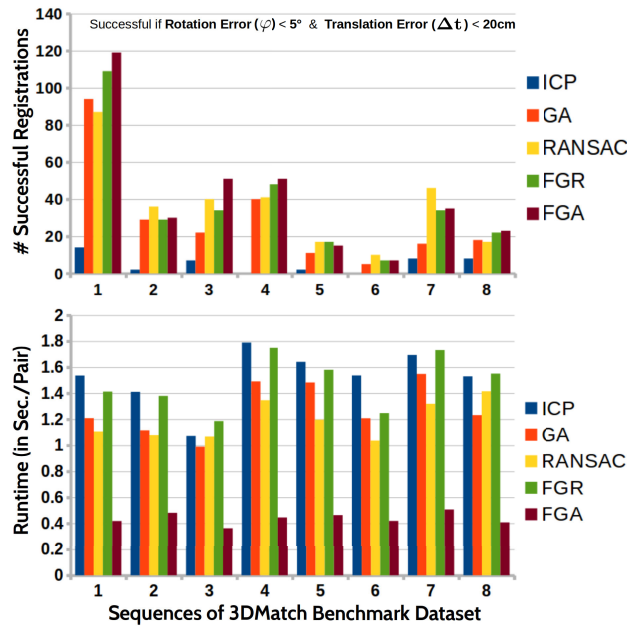

FIGURE 10. Our FGA is tested on eight evaluation sets of 3DMatch [10] dataset. We define a strict measurement (*i.e.,* when the angular error is less than 5° and translational error is less than 20 centimeters) to count successful registrations. *Top plot:* The success rate of FGA is compared against the previously well-performing methods — FGR [13], RANSAC [19] and two other baseline methods — GA [11] and ICP [6]. *Bottom plot:* FGA records the fastest runtime of ∼0.423 seconds, including mass computation via SPM function, on average per a scan pair.

FIGURE 11. FGA applied on *template* and *reference* point clouds (frames 1 and 10 from *Freiburg* [34] dataset). *Top row:* input images with the prior correspondenes and the initialization on the right. *Bottom row:* Our SPM (**S**) function radially distributes weights around *four* different prior landmarks (highlighted by blue lines), and the alignment on the right.

E. ROBUSTNESS AGAINST DATA DISTURBANCES

To evaluate the robustness of FGA against different disturbing effects, we take a clean *bunny* with 1889 points and add
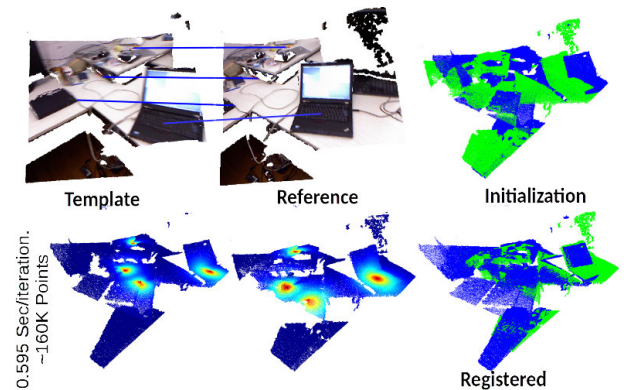
to it **(a)** 40% (of M) random Gaussian, **(b)** 40% (of M) random uniformly distributed noise. We next **(c)** transform the samples randomly with angular deviations φ_x, φ_y, and φ_z where all are $\in \mathcal{U}(0, \frac{3\pi}{4})$. We prepare 100 such independent test samples for each of the three cases and report success rate when RMSE value is <0.01, with average speed (fps) comparison in Table 2 and visualizations in Fig. 13. FGA performs as the second-best method after CPD [21] in the presence of noise. In extreme noisy input scenarios (when the amount of noise is $>60\%$ of input data), the NIV (**N**) measure can be constant (*e.g.*, **N(X)** and **N(Y)** = 1). FGA is more efficient and accurate in registering substantially misaligned data
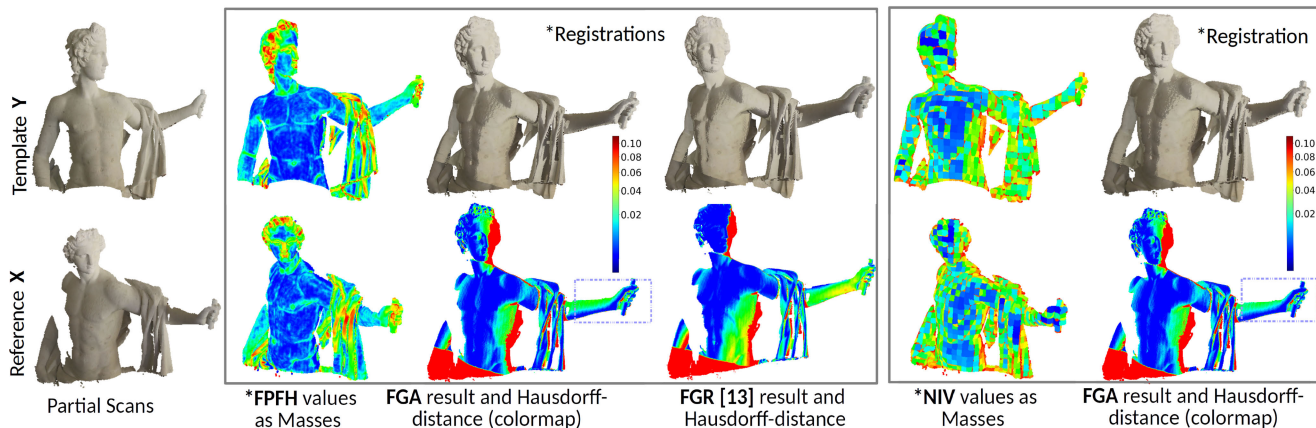
FIGURE 12. Partial scans of the Apollo statue aligned using FGA, with the point masses distributed using feature values. We use our smooth-particle mass mapping as a per-point feature and FPFH [19] as an alternative option to demonstrate the performance of FGA compared to FGR [13] (the second-best performing method). The blue<green<red color scheme shows the Hausdorff distance on the template's surface after the alignment.
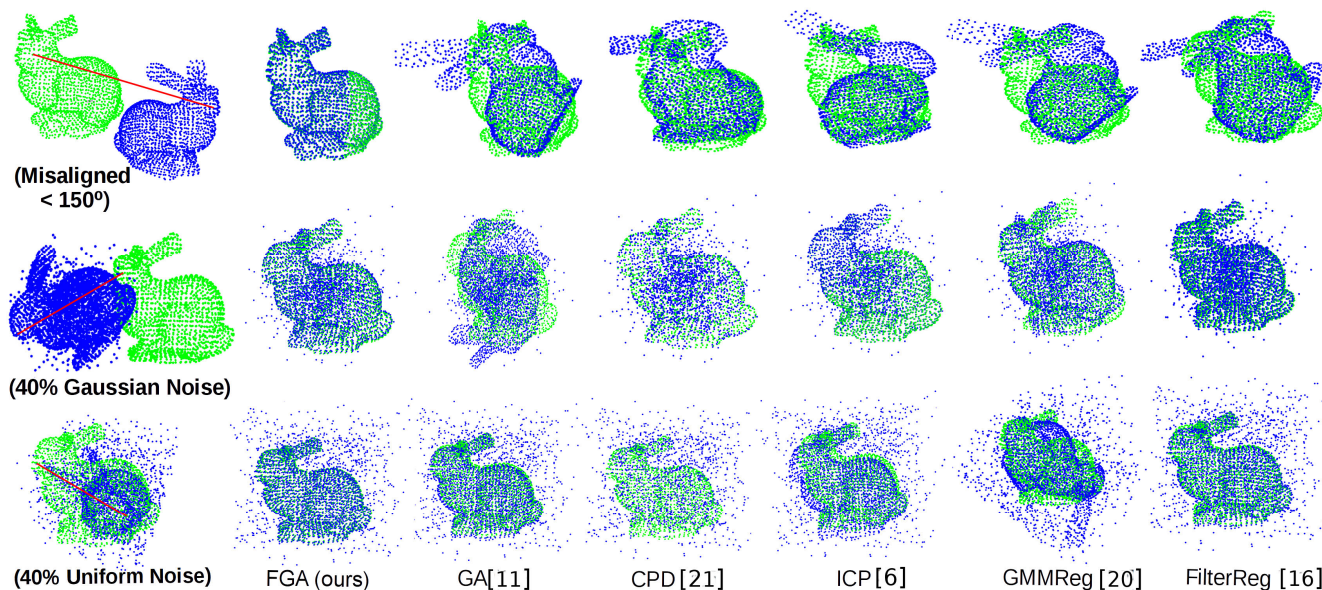

FIGURE 13. Exemplary registration results in the experiment with the data disturbances (applying large misalignment, 40% of Gaussian noise, and 40% of uniformly distributed noise on the template Y). Our FGA uses one prior match in this visualization and outperforms competing methods also without prior correspondences. See Table 2 for complete quantitative results (also without prior correspondences).

TABLE 2. Evaluation on *bunny* under Gaussian and uniform noise and large misalignment. "†" indicates the number of prior matches for FGA.

	misalign($< 150°$) (success%, fps)	$40\% \mathcal{U}$ noise (success%, fps)	$40\% \mathcal{G}$ noise (success%, fps)
FGA (ours)	**62%, 130.1**	69%, 121.3	78%, 89.1
FGA (ours)/1†	**66%, 126.1**	63%, 73	72%, 78
FGA (ours)/3†	**82%, 125**	79%, 73	**81%**, 78
GA [11]	44%, 8	66%, 6.5	75%, 8.1
CPD [21]	59%, 4.9	**95%**, 3.5	**93%**, 4.98
ICP [6]	40%, 113	6%, 39.1	62, %, 25
GMMReg [20]	32%, 18.1	55%, 29	49%, 39
FilterReg [16]	**28%**, 72.2	62%, 46.8	51%, 58.2

when only a few landmark correspondences are available. In this case, even without any landmark correspondences,

FGA has the highest success rate of 62%, whereas FilterReg's performance is the worst with 28%.

F. LIDAR ODOMETRY

LiDAR data exhibits varying point sampling densities at the near and far fields. Already in the early research, *Occupancy Grid* [94] mentioned cases where robotic navigation has to deal with differently tessellated spatial information. Our NIV measure is especially helpful in such scenarios as it plays the role of a density map to spread out weights across the point clouds. We choose the LiDAR dataset available from the *KITTI* [86] and the *Ford Campus Vision* [90] benchmark. No prior point correspondences are used in these experiments. Hence, our SPM function is equal to the NIV values on the point clouds.
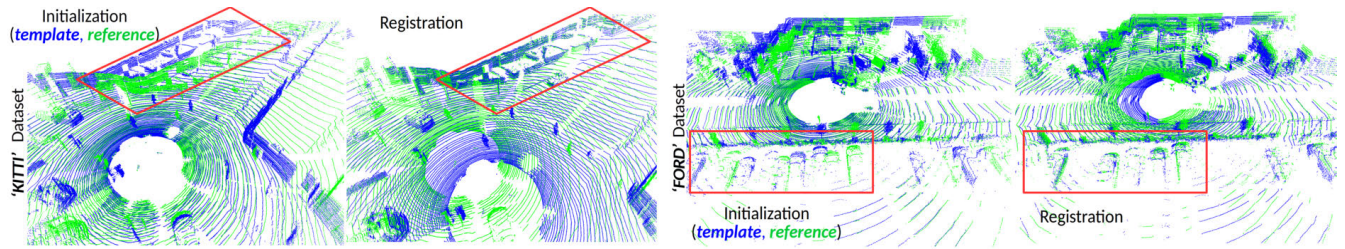
FIGURE 14. Registration results of FGA on KITTI [86] and Ford Campus Vision [90] datasets without subsampling. *Left:* frames 1 and 20 as Y and X, respectively ($M + N \approx 245k$ points) from 2011_09_26_*drive*_0005_*sync* driving sequence. *Right:* frames 1000 and 1010 as Y and X ($M + N \approx 150K$ points). In both cases, no landmarks are available.

Fig. 14 demonstrates results of FGA for two navigation scenarios where a car is either — (i) taking some turn (in KITTI dataset) as orientation change is dominant, or (ii) moving forward (in *Ford* dataset) as translation part is dominant. FGA correctly aligns the predominant scene components such as cars (marked by *red boxes*) and buildings. Note how FGA ignores the points concentrated at the central part and rotates **Y** towards **X** in the KITTI test or translates the template point cloud in the forward direction to align it with **X** on the *Ford* dataset. Our method without the NIV map registers incorrectly. Other methods, except Filter-Reg, FGR, and RANSAC, cannot process such large point clouds. Subsampling here can result in removing a few but necessary data (*e.g.,* the number of points representing a pedestrian is only ~100).

The CPU and GPU versions of FGA run on *six* different *KITTI* [86] sequences resulting in a total of 1327 experiments to obtain the sensor trajectories. We apply consecutive frame-to-frame alignments, where each alignment involves approximately $220k$ points. Fig. 15 illustrates the ground-truth path in red and our estimated path in blue for three of the sequences with minimum sensor drifts. FGA requires 70 seconds on CPU and 0.7 seconds on GPU on average to align a pair of frames (the speedup of two orders of magnitude). If only the sensor's forward view is considered or points on the ground are removed, the input data size is reduced by ≈40%. FGA can process 8−10 frames per second for such input data size. Compared to FGA, other methods — GA [11], CPD [21], point-to-point ICP [6] and GMMReg [20] cannot process input point clouds of large size, except FilterReg [16], FGR [13] and RANSAC [19].

Thus, to evaluate the competing methods (CPD, ICP, RMMReg and RANSAC), we need inputs of substantially small sizes. This can be achieved by either extracting feature descriptors from large inputs or subsampling those with large subsampling factors. Hence, we use density-based sampling of the input point clouds as shown in Fig. 16, *i.e.,* we remove parts of the point clouds in which point density exceeds a threshold. For the sake of speed, this sampling is done naively using regular volumetric non-overlapping bins. In this evaluation, the transformation errors resulting from the pairwise registrations of the LiDAR frames using any method will be lower than that would result by using a simple
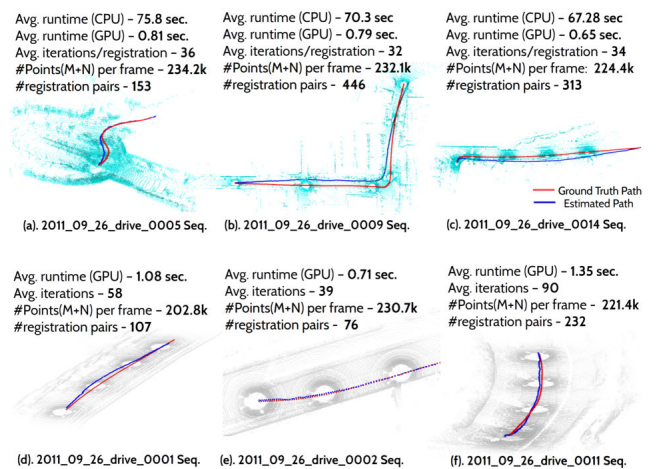

FIGURE 15. Consecutive frame-wise registrations of LiDAR point clouds from *six* different driving sequences of the KITTI-RAW [86] dataset by our FGA. The ground-truth paths are compared with the estimated paths, which reflects minimum drift and transformation error. In the *first* row, the driving sequences have more turns in the trajectories of sensor-carrying vehicle. In contrast, the vehicle trajectories in the *second* row have relatively more pronounced rectilinear motion than in the *first* row.
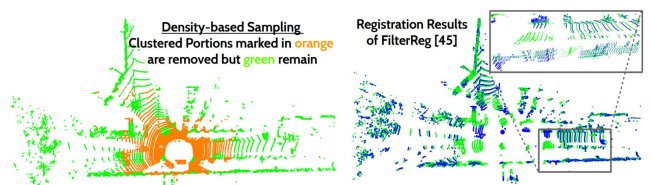

FIGURE 16. *On the right*: Density-based subsampling of input LiDAR point clouds which keeps the salient points (*e.g.,* the points representing the cars, facades and pedestrians) and removes mainly the points in the central area with a large amount of clutters. *On the left*: This careful sampling allows other robust registration methods, *e.g.,* FilterReg [16] to find globally-optimal correspondences.

uniform sampling. The plots in Fig. 17 quantify the *cumulative sum* of the angular errors (φ) and translation errors ($\Delta \mathbf{t}$) between estimated and ground-truth poses of the sensor for three different driving sequences from the KITTI-RAW data. It shows that the compared methods can estimate the sensor orientations (in degree) reasonably close to the ground truth but the translation errors (in centimeters) reflect that FGA, FGR and FilterReg are the standouts among them.
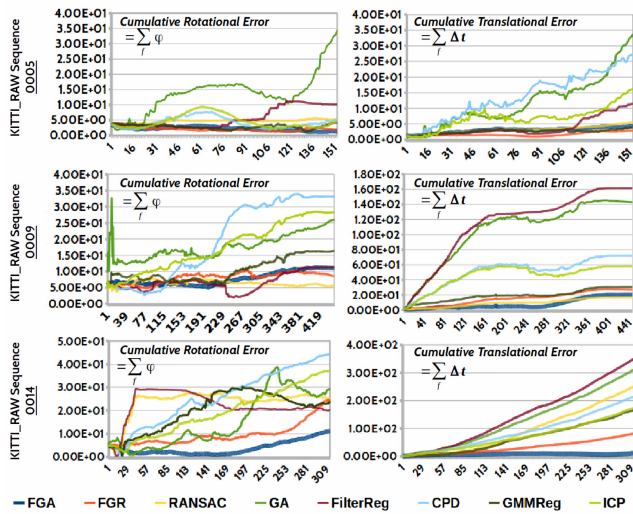
FIGURE 17. Quantitative evaluation of several methods on the KITTI-RAW dataset. Plots show cumulative rotational and translational errors accumulated over pairwise registrations of all consecutive frames. Cumulative translational errors remain lowest across the frames for a given sequence in the case of our FGA. Only FGR [13] and RANSAC [19] show either comparable or higher accuracy over a few short subsequences.

The cumulative translation errors across all the three driving sequences are minimum when using FGA. We run the frame-to-frame registration using FGA on a GPU for all sequences of KITTI RAW dataset. Estimated vehicle trajectories for all other sequences are compared against the ground-truth path in Fig. 15. The estimated paths are close to the ground-truth paths without applying any refinement or loop closure methods.

G. PARAMETER SELECTION

Three main FGA parameters, *i.e.*, G, ϵ, and η, account for the convergence basin of the objective function (4). To determine their best values, we prepare a regular 3D grid covering $G \in [1, 200]$, $\epsilon \in [0.1, 0.5]$, and $\eta \in [0.1, 0.5]$, with the step sizes of 1, 0.1 and 0.1, respectively. FGA is then executed for 10^4 grid cells for each of the three datasets — *bunny, lounge* and *KITTI*. In Fig. 18, the color indicates RMSE between registered pairs of the *bunny*. For *lounge* and *KITTI* data, the color denotes the final Euler angular error in Eq. (21) and the total transformation error in Eq. (22), respectively. The plots in Fig. 18 show that the parameters can be set in a wide interval for all tested datasets. The polygon area in the plot marks a wide and common range of parameters which can be used to obtain minimal alignment errors.

VI. DISCUSSION

All in all, our carefully-designed and extensive experiments confirm the effectiveness of the gravitational point set registration paradigm, also in the context of the recent advances in point set alignment as of 2020. Especially with automatically-extracted features which are mapped to the point masses, the proposed method either performs on par with the respectively best method in a given scenario or surpasses the alignment accuracy of all compared methods, as shown in Secs. V-C–V-D. From the experiments in Secs. V-B and V-F, we also see that FGA is well suitable for parallelization on a single GPU. It thus can be used in applications requiring interactive system response (*e.g.*, autonomous driving). From the correlation plot in Fig. 6-(B), we see that FGA does not trade the computational speed for
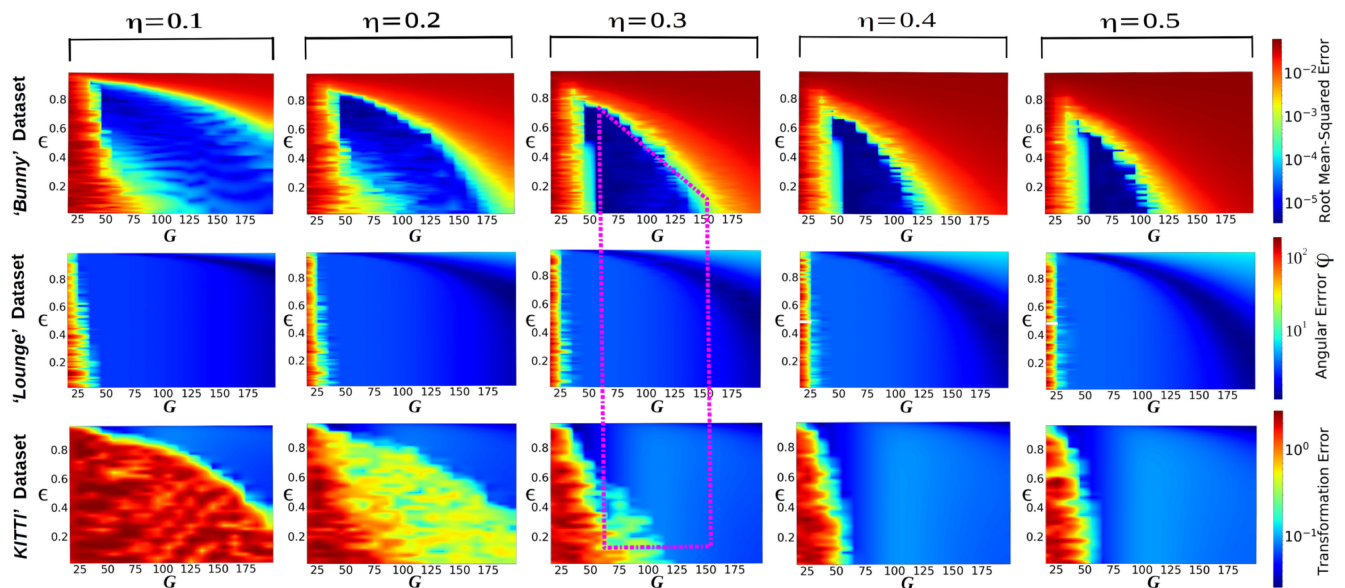

FIGURE 18. Selecting best FGA parameters with a grid search method on G, ϵ and η (slices for $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5). These parameters can be set from the common and wide area (value ranges) marked by the polygon in the middle. The parameter settings for our experiments are drawn from this region. Another figure like this would also draw the same conclusion on datasets scaled within other ranges (*e.g.,* $[-1, 1]$).

the accuracy as several competing methods do (*e.g.,* CPD or GMMReg). Experiments in Sec. V-E also confirm high and steady robustness of FGA to noise which secures the second place for FGA in this category among all tested methods. FGA resolves larger initial misalignments compared to other tested approaches due to the multiply-linked character of point interactions. Thus, there are much fewer alignment scenarios which result in local minima compared to the competing techniques.

The experiment for the parameter choice in Sec. V-G confirms that the parameters can be conveniently fixed across various datasets and scenarios. Thus, FGA addresses several limitations of the original GA [11] (*i.e.,* long runtime, scenario-specific parameters) and establishes a new state of the art in general-purpose point set alignment. Along with that, FGA does not require correspondence extraction in a pre-processing step or local correspondence search in every iteration — a locally-optimal alignment is achieved when the multiply-linked GPE is minimized. This GPE is known upfront and kept unchanged during the entire optimization.

Interestingly, FGA consistently outperforms GA [11] in the accuracy and speed in all performed experiments. The automatically-defined boundary conditions on masses (prior matches and SPM function) guide the alignment procedure away from many local minima. We thus believe that FGA is a significant step forward in general-purpose point set registration techniques.

Limitations: Although the cues such as known landmark positions, intelligent use of the feature values as masses, and smooth-particle mass functions make our method robust, the recovered transformation is still locally-optimal. Since our method uses values of SPM function or other feature values as the point masses, registration accuracy depends upon the feature accuracy (matching precision and recall).

VII. CONCLUSION AND FUTURE WORK

We introduce FGA — a fast physics-based rigid point set registration method which is applicable for various types of data with noise and clustered outliers (such as LiDAR scans). While most competing methods get trapped in the local minima caused by non-uniform point sampling, our FGA recovers transformations which are closer to the optimal ones, thanks to the boundary conditions defined on masses via the SPM function. The acceleration policy of FGA hierarchically divides the input point clouds in a BH tree with local space properties, and solves the gravitational force approximation problem with second-order ODEs in quasilinear time. From the experimental results, we draw the conclusion that our method is fast (*i.e.,* it can support interactive and real-time frame rates), accurate and robust on most of the general and more challenging datasets, especially when dealing with noise and non-uniform point sampling density.

FUTURE WORK

Our BH tree representation can be an efficient alternative to computationally expensive grid representations of point clouds for deep-learning-based alignment methods, which we are planning to investigate next. Another direction is extending FGA for estimating non-rigid motion fields and real-time RGB-D scene flow on a single GPU, with applications to autonomous driving systems.

APPENDIX
ROTATION AND TRANSLATION FOR DATA SCALED IN [a, b]

For a given input pair of *template* $\mathbf{Y}$, *reference* $\mathbf{X}$ and a range $[a, b]$, the normalization of the pair to $(\mathbf{X}, \mathbf{Y}) \mapsto (\mathbf{X}_{\widehat{n}}, \mathbf{Y}_{\widehat{n}})$, is formalized in three steps. First, the mean positions of $\mathbf{X}$ and $\mathbf{Y}$, which read as

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{X}_j; \quad \bar{\mathbf{Y}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{Y}_i; \tag{A.1}$$

are subtracted from the position vectors of the respective point clouds. This step shifts the center of the point clouds at the origin. Thereafter, the minimum and the maximum of the position vectors from the shifted point clouds

$$l = \min \left\{ \min \left\{ \cup_{i=1}^{N} (\mathbf{X}_i - \bar{\mathbf{X}}) \right\}, \min \left\{ \cup_{j=1}^{M} (\mathbf{Y}_j - \bar{\mathbf{Y}}) \right\} \right\},$$
$$r = \max \left\{ \max \left\{ \cup_{i=1}^{N} (\mathbf{X}_i - \bar{\mathbf{X}}) \right\}, \max \left\{ \cup_{j=1}^{M} (\mathbf{Y}_j - \bar{\mathbf{Y}}) \right\} \right\} \tag{A.2}$$

are used to scale them in an arbitrary range $[a, b]$ as

$$\mathbf{X}_{\widehat{n}} = \begin{bmatrix} \frac{(\mathbf{X}_1 - \bar{\mathbf{X}} - \mathbf{l})(b-a)}{r-l} + \mathbf{a} \\ \vdots \\ \frac{(\mathbf{X}_N - \bar{\mathbf{X}} - \mathbf{l})(b-a)}{r-l} + \mathbf{a} \end{bmatrix}^T,$$

$$\mathbf{Y}_{\widehat{n}} = \begin{bmatrix} \frac{(\mathbf{Y}_1 - \bar{\mathbf{Y}} - \mathbf{l})(b-a)}{r-l} + \mathbf{a} \\ \vdots \\ \frac{(\mathbf{Y}_M - \bar{\mathbf{Y}} - \mathbf{l})(b-a)}{r-l} + \mathbf{a} \end{bmatrix}^T, \tag{A.3}$$

where $\mathbf{l} = (l, l, l)^T$ and $\mathbf{a} = (a, a, a)^T$. FGA is applied on the normalized input data pair $(\mathbf{X}_{\widehat{n}}, \mathbf{Y}_{\widehat{n}})$. The optimal rigid transformation parameters $\mathbf{T}^* = [\mathbf{R}|\mathbf{t}]$, as shown in Alg. 2, register $\mathbf{Y}_{\widehat{n}}$ to $\mathbf{X}_{\widehat{n}}$ as $\mathbf{Y}_{\widehat{n}} \longmapsto [\mathbf{R}|\mathbf{t}] \times \mathbf{Y}_{\widehat{n}}$. In the denormalized domain of $\mathbf{Y}_{\widehat{n}}$, the same mapping requires a denormalized translation component 'dnorm(**t**)'of transformation matrix $\mathbf{T}^*$ for mapping $\mathbf{Y} \longmapsto [\mathbf{R}|\text{dnorm}(\mathbf{t})] \times \mathbf{Y}$. The derivation steps for 'dnorm(**t**)'are:

$$\Rightarrow [\mathbf{R}|\text{dnorm}(\mathbf{t})] \times \mathbf{Y} \equiv \underbrace{\text{dnorm}([\mathbf{R}|\mathbf{t}] \times \mathbf{Y}_{\widehat{n}})}_{\text{reverse of Eq. A.3}}$$

$$\Leftrightarrow [\mathbf{R}|\text{dnorm}(\mathbf{t})] \times \mathbf{Y} = \frac{(([\mathbf{R}|\mathbf{t}] \times \mathbf{Y}_{\widehat{n}}) - \mathbf{a})(r-l)}{(b-a)} + \bar{\mathbf{Y}} + l$$

$$\Leftrightarrow \mathbf{R}\mathbf{Y} + \text{dnorm}(t) = \frac{(([\mathbf{R}|\mathbf{t}] \times \mathbf{Y}_{\widehat{n}}) - \mathbf{a})(r-l)}{(b-a)} + \bar{\mathbf{Y}} + l,$$

and after substituting $\mathbf{Y}_{\widehat{n}}$ by the right hand side of Eq.(A.3)

$$\Leftrightarrow \ldots = \frac{\left(\mathbf{R} \left(\frac{(\mathbf{Y} - \bar{\mathbf{Y}} - \mathbf{a})(b-a)}{(r-l)} + \mathbf{a} \right) + \mathbf{t} - \mathbf{a} \right)(r-l)}{(b-a)} + \bar{\mathbf{Y}} + \mathbf{l}$$

$$\Leftrightarrow \ldots = \mathbf{RY} \underbrace{- \mathbf{R\bar{Y}} - \mathbf{Rl} + \frac{(r-l)}{(b-a)}(\mathbf{Ra} + t - \mathbf{a})} + \mathbf{\bar{Y}} + \mathbf{l}.$$
$$\text{dnorm}(\mathbf{t})$$

As it stands, the difference between the centroids $\mathbf{\bar{Y}} - \mathbf{\bar{X}}$ must be added to dnorm($\mathbf{t}$), as the translation was computed for the normalized point sets which are centered on the origin.

REFERENCES

[1] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 12, pp. 2633–2651, Dec. 2016.

[2] 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, May 2007.

[3] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2011, pp. 127–136.

[4] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 598–626, Apr. 2015.

[5] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auto. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[6] 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.

[7] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, Apr. 1992.

[8] B. Bellekens, V. Spruyt, R. Berkvens, and M. Weyn, "A survey of rigid 3D point cloud registration algorithms," in *Proc. Int. Conf. Ambient Comput., Appl., Services Technol. (IARA)*, 2014, pp. 8–13.

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

[10] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1802–1811.

[11] V. Golyanik, S. A. Ali, and D. Stricker, "Gravitational approach for point set registration," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 5802–5810.

[12] V. Golyanik, C. Theobalt, and D. Stricker, "Accelerated gravitational point set alignment with altered physical laws," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2080–2089.

[13] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 766–782.

[14] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2514–2523.

[15] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3523–3532.

[16] W. Gao and R. Tedrake, "FilterReg: Robust and efficient probabilistic point-set registration using Gaussian filter and twist parameterization," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11095–11104.

[17] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. CVPR*, Jun. 2019, pp. 7163–7172.

[18] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 26, 2021, doi: 10.1109/TPAMI.2021.3054619.

[19] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2009, pp. 3212–3217.

[20] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2005, pp. 1246–1251.

[21] 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.

[22] Y. Deng, A. Rangarajan, S. Eisenschenk, and B. C. Vemuri, "A Riemannian framework for matching point clouds represented by the schrödinger distance transform," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 3756–3761.

[23] P. Jauer, I. Kuhlemann, R. Bruder, A. Schweikard, and F. Ernst, "Efficient registration of high-resolution feature enhanced point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1102–1115, May 2019.

[24] A. H. S. Lai and H. Yung, "Segmentation of color images based on the gravitational clustering concept," *Proc. SPIE*, vol. 37, no. 3, p. 989, Mar. 1998.

[25] Z. Zhang, Y. Wang, T. Huang, and L. Zhan, "A weighting algorithm based on the gravitational model for local stereo matching," *Signal, Image Video Process.*, vol. 14, no. 2, pp. 315–323, Mar. 2020.

[26] M. S. Nixon, X. U. Liu, C. Direkoglu, and D. J. Hurley, "On using physical analogies for feature and shape extraction in computer vision," *Comput. J.*, vol. 54, no. 1, pp. 11–25, Jan. 2011.

[27] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, no. 2, pp. 325–348, Dec. 1987.

[28] R. Spurzem, P. Berczik, G. Marcus, A. Kugel, G. Lienhart, I. Berentzen, R. Männer, R. Klessen, and R. Banerjee, "Accelerating astrophysical particle simulations with programmable hardware (FPGA and GPU)," *Comput. Sci., Res. Develop.*, vol. 23, nos. 3–4, pp. 231–239, Jun. 2009.

[29] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[30] J. Barnes and P. Hut, "A hierarchical $O(N\log N)$ force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, Dec. 1986.

[31] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *J. Geophys. Res.*, vol. 76, no. 8, pp. 1905–1915, Mar. 1971.

[32] V. Golyanik and C. Theobalt, "Optimising for scale in globally multiply-linked gravitational point set registration leads to singularities," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 164–172.

[33] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "LO-Net: Deep real-time LiDAR odometry," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8473–8482.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 573–580.

[35] Z. J. Yew and G. H. Lee, "3DFeat-Net: Weakly supervised local 3D features for point cloud registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 607–623.

[36] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2008, pp. 3384–3391.

[37] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. Int. Conf. Comput. Vis. Workshops (ICCV)*, Sep. 2009, pp. 689–696.

[38] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallograph. A, Cryst. Phys., Diffraction, Theor. Gen. Crystallogr.*, vol. 32, no. 5, pp. 922–923, Sep. 1976.

[39] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.

[40] 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, Sep. 1987.

[41] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 5, no. 7, pp. 1127–1135, Jul. 1988.

[42] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.

[43] M. Greenspan and M. Yurick, "Approximate K-D tree search for efficient ICP," in *Proc. Int. Conf. Recent Adv. 3D Dig. Imag. Modeling (3DIM)*, 2003, pp. 442–448.

[44] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter, "Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration," *J. Softw. Eng. Robot.*, vol. 3, no. 1, pp. 2–12, Feb. 2012.

[45] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2001, pp. 1145–1153.

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

[47] A. Segal, D. Hähnel, and S. Thrun, "Generalized-ICP," *Robot., Sci. Syst.*, vol. 2, no. 4, p. 435, 2009.

[48] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Math. Methods Biomed. Image Anal. (MMBIA)*, Jun. 2000, pp. 190–197.

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

[50] B. Eckart, K. Kim, and J. Kautz, "HGMR: Hierarchical Gaussian mixtures for adaptive 3D registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 705–721.

[51] V. Golyanik, B. Taetz, and D. Stricker, "Joint pre-alignment and robust rigid point set registration," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 4503–4507.

[52] M. Danelljan, G. Meneghetti, F. S. Khan, and M. Felsberg, "A probabilistic framework for color-based point set registration," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1818–1826.

[53] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 143–152.

[54] M. Saval-Calvo, J. Azorin-Lopez, A. Fuster-Guillo, V. Villena-Martinez, and R. B. Fisher, "3D non-rigid registration using color: Color coherent point drift," *Comput. Vis. Image Understand.*, vol. 169, pp. 119–135, Apr. 2018.

[55] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 652–660.

[56] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[57] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3D point cloud matching with smoothed densities," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5545–5554.

[58] S. Shimada, V. Golyanik, E. Tretschk, D. Stricker, and C. Theobalt, "DispVoxNets: Non-rigid point set alignment with supervised learning proxies," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 27–36.

[59] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8592–8601.

[60] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2472–2481.

[61] Y. Wang and J. Solomon, "PRNet: Self-supervised learning for partial-to-partial registration," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 8812–8824.

[62] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 12–21.

[63] J. Groß, A. Ošep, and B. Leibe, "AlignNet-3D: Fast point cloud registration of partially observed objects," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 623–632.

[64] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "DeepGMR: Learning latent Gaussian mixture models for registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 733–750.

[65] W. E. Wright, "Gravitational clustering," *Pattern Recognit.*, vol. 9, no. 3, pp. 151–166, Oct. 1977.

[66] G. Sun, Q. Liu, Q. Liu, C. Ji, and X. Li, "A novel approach for edge detection based on the theory of universal gravity," *Pattern Recognit.*, vol. 40, no. 10, pp. 2766–2775, Oct. 2007.

[67] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV)*, Nov. 2011, pp. 1626–1633.

[68] S. Kundu, "Gravitational clustering: A new approach based on the spatial distribution of the points," *Pattern Recognit.*, vol. 32, no. 7, pp. 1149–1160, Jul. 1999.

[69] C. Lopez-Molina, H. Bustince, J. Fernandez, P. Couto, and B. De Baets, "A gravitational approach to edge detection based on triangular norms," *Pattern Recognit.*, vol. 43, no. 11, pp. 3730–3741, Nov. 2010.

[70] C. Marco-Detchart, C. Lopez-Molina, J. Fernandez, and H. Bustince, "A gravitational approach to image smoothing," in *Advances in Fuzzy Logic and Technology*. Springer, 2018, pp. 468–479.

[71] V. Golyanik and C. Theobalt, "A quantum computational approach to correspondence problems on point sets," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9182–9191.

[72] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem," *Science*, vol. 292, no. 5516, pp. 472–475, Apr. 2001.

[73] F. Diacu, "The solution of the *N*-body problem," *Math. Intell.*, vol. 18, no. 3, pp. 66–70, 1996.

[74] M. Trenti and P. Hut, "N-body simulations (gravitational)," *Scholarpedia*, vol. 3, no. 5, p. 3930, 2008.

[75] S. J. Aarseth, *Gravitational N-Body Simulations: Tools and Algorithms* (Cambridge Monographs on Mathematical Physics). Cambridge, U.K.: Cambridge Univ. Press, 2003, doi: 10.1017/CBO9780511535246.

[76] W. W. Armstrong and M. W. Green, "The dynamics of articulated rigid bodies for purposes of animation," *Vis. Comput.*, vol. 1, no. 4, pp. 231–240, Dec. 1985.

[77] J. Makino and M. Taiji, "Scientific simulations with special-purpose computers: The GRAPE systems," *Vis. Comput.*, vol. 1, no. 4, pp. 231–240, 1985, doi: 10.1007/BF02021812.

[78] E. D. Sozzo, L. Di Tucci, and M. D. Santambrogio, "A highly scalable and efficient parallel design of N-body simulation on FPGA," in *Proc. Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2017, pp. 241–246.

[79] M. Winkel, R. Speck, H. Hübner, L. Arnold, R. Krause, and P. Gibbon, "A massively parallel, multi-disciplinary Barnes–Hut tree code for extreme-scale N-body simulations," *Comput. Phys. Commun.*, vol. 183, no. 4, pp. 880–889, Apr. 2012.

[80] M. Korn, M. Holzkothen, and J. Pauli, "Color supported generalized-ICP," in *Proc. Comput. Vis. Theory Appl.*, vol. 3, 2014, pp. 592–599.

[81] P. Bagchi, D. Bhattacharjee, and M. Nasipuri, "Reg3DFacePtCd: Registration of 3D point clouds using a common set of landmarks for alignment of human face images," *Künstliche Intelligenz*, vol. 33, no. 4, pp. 369–387, Dec. 2019.

[82] M. Khoury, Q.-Y. Zhou, and V. Koltun, "Learning compact geometric features," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 153–161.

[83] J. Serafin, E. Olson, and G. Grisetti, "Fast and robust 3D feature extraction from sparse point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4105–4112.

[84] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "CAD-model recognition and 6DOF pose estimation using 3D cues," in *Proc. Int. Conf. Comput. Vis. Workshops (ICCV)*, Nov. 2011, pp. 585–592.

[85] P. Mcmullen, "Inequalities between intrinsic volumes," *Monatshefte Für Math.*, vol. 111, no. 1, pp. 47–53, Mar. 1991.

[86] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[87] M. Burtscher and K. Pingali, "An efficient CUDA implementation of the tree-based Barnes Hut *n*-body algorithm," in *GPU Computing Gems Emerald Edition* (Applications of GPU Computing Series), W.-M. W. Hwu, Ed. Boston, MA, USA: Morgan Kaufmann, 2011, pp. 75–92, doi: 10.1016/B978-0-12-384988-5.00006-1.

[88] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[89] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.

[90] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and LiDAR data set," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, Nov. 2011.

[91] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *Int. J. Comput. Vis.*, vol. 103, no. 3, pp. 267–305, Jul. 2013.

[92] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[93] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS fast global pointcloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, Aug. 2014.

[94] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

SK AZIZ ALI received the B.Tech. degree from the West Bengal University of Technology, in 2011, and the M.Sc. degree in computer science from the University of Kaiserslautern, Germany, in 2017. He is currently a Researcher with the German Research Center for Artificial Intelligence, Augmented Vision Group, Kaiserslautern. He is also doing the doctoral research with the University of Kaiserslautern in the areas of learning representations, physics-based optimization methods, and 3D reconstruction.

KEREM KAHRAMAN received the B.Sc. degree in computer science from the Technische Universität Kaiserslautern, in 2017, where he is currently pursuing the M.Sc. degree. Since 2018, he has been working as an Assistant Researcher with the Augmented Vision Department, DFKI, Kaiserslautern. His research interests include efficient data representations for 3D and 2D convolutional neural networks, rigid and non-rigid registration methods, and the application of these methods on autonomous driver-assistant systems.

DIDIER STRICKER is currently a Professor with the University of Kaiserslautern and the Scientific Director with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, where he leads the research with the Department Of Augmented Vision. From 2002 to June 2008, he lead the Department of Virtual and Augmented Reality, Fraunhofer Institute for Computer Graphics (Fraunhofer IGD), Darmstadt, Germany. In this function, he initiated and participated in many national and international projects in the areas of computer vision and virtual and augmented reality. In 2006, he received the Innovation Prize from the German Society of Computer Science. He also serves as a reviewer for different European or national research organizations. He is also a regular reviewer for the most important journals and conferences in the areas of VR/AR and computer vision.

CHRISTIAN THEOBALT is currently a Professor of computer science and the Director of the Visual Computing and Artificial Intelligence Department, Max Planck Institute for Informatics, Saarbrücken, Germany. He is also a Professor with Saarland University. His research interests include intersection of computer graphics, computer vision, and machine learning. For instance, he works on virtual humans, 3D and 4D scene reconstruction, neural rendering and neural scene representations, marker-less motion and performance capture, machine learning for graphics and vision, and new sensors for 3D acquisition. He received several awards, including the Otto Hahn Medal of the Max-Planck Society, in 2007; the EUROGRAPHICS Young Researcher Award, in 2009; the German Pattern Recognition Award, in 2012; an ERC Starting Grant, in 2013; the ERC Consolidator Grant, in 2017; and the EURIGRAPHICS Outstanding Technical Contributions Award, in 2020. He is also the Co-Founder of theCaptury.

VLADISLAV GOLYANIK received the M.Sc. degree in computer science from Ulm University, in 2013, and the Ph.D. degree from the University of Kaiserslautern, in 2019, for his work titled Robust Methods for Dense Monocular Non-Rigid 3D Reconstruction and Alignment of Point Clouds. He is currently leading the 4D and Quantum Vision Research Group, Visual Computing and Artificial Intelligence Department, Max Planck Institute for Informatics, Saarbrücken, Germany. The primary research interests of his team include 3D reconstruction and analysis of deformable scenes, matching problems on point sets and graphs, neural rendering, physics-based methods, and emerging hardware and sensors, such as quantum computers and event cameras. He was a recipient of the WACV'16 Best Paper Award and the 2020's Yearly Ph.D. Dissertation Award from the German Association for Pattern Recognition (DAGM).

• • •