

Adaptive Polydisperse Sphere Packings for High Accuracy Computations of the Gravitational Field

Hermann Meißenhelger
Institute for Computer Graphics and Virtual Reality
University of Bremen
Bibliothekstraße 5, 28359 Bremen, Germany
meissenhelger@cs.uni-bremen.de

Matthias Noeker
Royal Observatory of Belgium (ROB)
& Université catholique de Louvain (UCLouvain)
Avenue Circulaire 3, 1180 Uccle, Belgium
matthias.noeker@observatory.be

René Weller
Institute for Computer Graphics and Virtual Reality
University of Bremen
Bibliothekstraße 5, 28359 Bremen, Germany
weller@cs.uni-bremen.de

Tom Andert
Institute of Space Technology and Space Applications (ISTA)
Bundeswehr University Munich
Werner-Heisenberg-Weg 39,85577 Munich, Germany
tom.andert@unibw.de

Gabriel Zachmann
Institute for Computer Graphics and Virtual Reality
University of Bremen
Bibliothekstraße 5, 28359 Bremen, Germany
zach@cs.uni-bremen.de

Abstract—Autonomous missions to small bodies are an essential part of recent space exploration. This increasing interest in asteroids and moons leads to an increased need for high-fidelity trajectory simulations to guarantee mission success. Although their shape might vary greatly, most of those small bodies we know have an irregular shape, thus, a gravity field that is challenging to compute.

We present a new method to model the mass of celestial bodies based on adaptive polydisperse sphere packings. Using polydisperse spheres in the mascon model has shown to deliver a very good approximation of the mass distribution of celestial bodies while allowing fast computations of the gravitational field. However, small voids between the spheres reduce the accuracy especially close to the surface. Hence, the idea of our adaptive sphere packing is to place more spheres close to the surface instead of filling negligible small gaps deeper inside the body. Although this reduces the packing density, we achieve greater accuracy close to the surface. For the adaptive sphere packing, we propose a mass assignment algorithm that uniformly samples the volume of the body. Additionally, we present a method to further optimize the mass distribution of the spheres based on least squares optimization. The sphere packing and the gravitational acceleration remain computable entirely on the GPU (Graphics Processing Unit).

We have evaluated our method by comparing it to analytical solutions (such as a cube) and detailed shape models of real asteroids, for which we have used the traditional polyhedral method as a reference. We compute the gravitational acceleration on the surface since the most significant relative error lies there, which is also crucial for landing maneuvers. Our adaptive sphere packing approach produces reliable gravitation data for physically-based simulation of space missions and shows more accurate results than previous similar methods.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. PREVIOUS WORK	2
3. METHODS	2
4. RESULTS	5
5. CONCLUSION AND FUTURE WORK	8

ACKNOWLEDGMENTS	8
REFERENCES	9
BIOGRAPHY	10

1. INTRODUCTION

The exploration of small solar system bodies such as asteroids and comets is still of very high interest. For example, the European HERA mission [1] is expected to visit the binary asteroid Didymos and its moon Dimorphos in 2026. In addition, NASA's Lucy mission [2], already launched in October 2021, will rendezvous with seven Trojans and one main-belt asteroid. Also, other missions to small celestial bodies are planned. For instance, we are part of KaNaRiA (cognition-based autonomous navigation using the example of resource mining in space) [3], for which the reference scenario is an "asteroid mining mission" in the main asteroid belt between Mars and Jupiter. One key scientific objective of these missions is the detailed characterization of the physical properties of the mission targets. The mass and gravity field of the scientific target is determined by the gravitational acceleration acting on the spacecraft. The mass combined with the volume estimate is inferred to give the total mass. A measure of the density distribution is the principal moments of inertia, which are related to the gravity field. An improved method with higher accuracy for the acceleration computation, therefore, helps to constrain the internal structure better, which can give clues to the origin and dynamic evolution of the target body.

This study compares the polyhedral and the mascon models' accuracy. We contrast our mascon models' computational effectiveness and do an error analysis. A metric for measuring accuracy is the gravitational acceleration error. Triangular surface meshes are used to calculate a solution with the polyhedral method. Mascon distributions are generated from meshes using a polydisperse sphere packing. A cube with homogenous density is used as a first test case to assess the error of the mascon models to a known analytical result [4]. Then we apply our approach to the inner mars-moon Phobos and the asteroid Bennu to evaluate their abilities to model practical geometry.

2. PREVIOUS WORK

The polyhedral method (PM) developed by [6] is a well-established and globally used closed-form analytical method to compute the gravitational field of any polyhedral shape file. The evaluation of the gravitation can even be performed on the surface of the polyhedron, yet not on the vertices and edges of the shape [7]. One drawback of the PM in its original form is that a constant density has to be assumed for the gravitational evaluation throughout the assessed body. This can be circumvented in different ways, e.g. by the sliver approach developed by [8], where the difference of two different-sized tetrahedra (spanned by three co-initial and collinear vectors) describes the gravitational field of the sliver, i.e. the non-mutual volume of the two tetrahedra. Another solution to vary the density inside a body using the PM is to assign differential densities to inhomogeneities (described with another polyhedron shape) inside a celestial body and to sum the gravitational fields [9]. On a local scale, the PM has been implemented in the Wedge-Pentahedra Method (WPM), where the local terrain is described by wedge-pentahedra, which are individually evaluated using the PM, allowing to vary the density per-wedge [10]. Regarding the computation time, it was shown by [7] that the PM is computationally more expensive compared to mascon approaches, while parallelization of the PM for a large number of evaluation points might improve the computation time.

Like the polyhedral model, spherical harmonics are helpful in the representation of the gravity field of a small body. The spherical harmonic coefficients can be estimated from measurements or computed from a constant-density polyhedron [11]. However, the classical application of this method has significant weaknesses when applied to small celestial bodies. The potential field diverges inside the body’s Brillouin sphere, and for highly non-spherical bodies, the spherical harmonic series representation demands an excessive number of terms. In order to overcome the low accuracy for close-proximity operations, some methods extend the spherical harmonic methods. An example is the spherical harmonic expansion method [12][13], which works with bodies of varied density and converges the model inside the Brillouin sphere. Nevertheless, this model’s implementation is complex and takes a long time to generate a configuration.

Mascon models are particularly well adapted to small bodies since they can simulate irregular forms and density distributions at any resolution. A summary of different mascon approaches is given in [14]. The discretized mass distribution causes potential evaluations close to the mascon elements to degenerate even though the mascon approach better represents the distant environment. Regardless, its usual advantage is the simple expression for gravitational potential. Voxels or spheres are typical shapes for the masses. The drawbacks of a constant density mascon model for a small body were underlined in [15], and the accuracy of many different mascon packing strategies was investigated. A similar packing strategy of non-uniform-sized spheres was given in [5] [7]. It utilized the Protosphere algorithm [16] to generate a sphere packing for triangular meshes. Better sphere filling efficiency and mass distribution methods in terms of estimated errors were shown compared to the similar method in [15]. Tetrahedra were used by [17], formed by the polyhedron facets and the body’s center of mass. Mascons based on an octree approach were applied [18], building mascons of uniform and non-uniform voxel for a spatial data structure.

Internal density from shape and gravity information was estimated in [19] by a least squares optimization with tetrahedral

shapes. Optimizing the mascons through non-linear regression was done by [20], noting that optimizing the location gets insignificant for many mascons. Yet, a difficulty of optimization is the large memory footprint, which was noted as a limitation by [21] and suggested solving it by distributed memory. To analyze our mascon model, we will compare it to an optimized solution.

3. METHODS

This section presents a new sphere packing with a novel mass assignment algorithm. Additionally, we describe how we compute optimal masses for a deeper evaluation.

Adaptive Sphere Packing

Our approach is based on the mascon model. The mascon model divides a body into simple and small elements known as mascons (mass concentration). Mascons are typically modeled as uniform homogeneous cubes or spheres. We concentrate the mass of the elements in their center, so the gravitational field is determined as a collection of point sources. The resulting gravity is then obtained by integrating the mass elements.

Similar to the work in [7], the primary principle of our method is to employ a polydisperse sphere packing, where the spheres are all contained within the shape but allowed to have varying radii while not overlapping. Due to the greedy nature of the Protosphere algorithm [16], it always places the largest spheres first that fit into the largest voids, following the idea of Apollonian sphere packings. This approach usually leads to high packing densities ($> 90\%$) for asteroid-like bodies.

Our key idea is based on the assumption that the smaller spheres deeper inside the body are less critical for an accurate gravitational field than small spheres close to the surface. Also, the gravity field on the surface has the most significant deviations. So our goal is to place *more* spheres close to the surface (see Fig. 1) while we allow larger voids in the center region of the body.

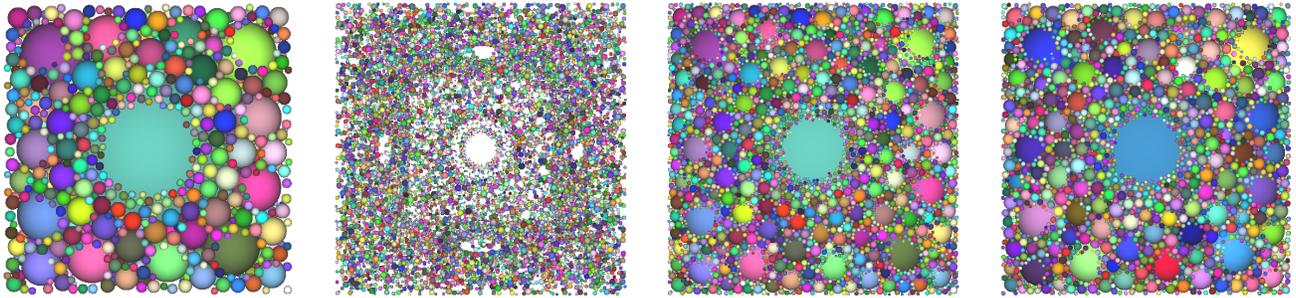
Therefore we have modified the Protosphere algorithm to start classically with the greedy filling until a user-defined packing density is reached. After this, we only place spheres to represent the surface details close to the border of the polygonal model.

Algorithm 1 Adaptive(PackingDensity ρ , Threshold λ)

```
if  $\rho < \lambda$  then
   $c \leftarrow$  select a random cell
else
   $c \leftarrow$  select randomly a cell at the border
end if
place the largest possible sphere in the cell  $c$ 
```

In detail, we control our adaptive sphere packing with a hyperparameter λ , which acts as a packing density threshold. Achieving a good coverage of a body’s volume usually leads to a more accurate gravity field. So we do not want to withdraw that idea completely; this is why we let the standard packing algorithm run till packing density reaches λ . Then we place spheres near the border while preferring the largest spheres as usual. In the end, the algorithm terminates if the target number of spheres for the sphere packing is reached.

Overall, this adaptive sphere packing achieves less volume



(a) **First: Greedy packing while below threshold (λ). Largest spheres get placed.** (b) **Second: Greedy packing near the surface till the goal number of spheres reached.** (c) **Our algorithm applied on: Adaptive sphere packing with packing density of 88.2% and 20k spheres.** (d) **Reasonable alternative: Full greedy packing with packing density of 89.24% and 20k spheres.**

Figure 1: Adaptive polydisperse sphere packing (20k spheres, $\lambda = 85\%$) to model a more accurate gravity field of a cube (1c). It allows us to reduce the error by 12%, with manageable computational time compared to similar method (1d, [5]). Packing density threshold λ controls the number of spheres near the surface. Our algorithm (Section 3) assigns additional mass to the spheres to match a body’s total mass.

coverage after λ is reached. A lower λ means less packing density and vice versa because we start to prefer the location of a sphere rather than filling globally large voids. To set the parameter, we use the packing density of the standard Protosphere algorithm as guidance. We usually set the hyperparameter around 1% till 5% below the reference.

Mass Assignment

If the body’s density is known, we can simply assign the masses to the spheres in the body. Since the spheres do not cover the whole volume of the body entirely due to the voids, we would end up with a mass that is lower than the actual total mass of the body. The packing density ρ accurately reflects the mass we lose in percent ($100\% - \rho$). The work in [5] presented several heuristics that consider the empty volume as a mass and assign spheres additional mass where the delta percentage volume increase method (DPVI) performed best. Smaller spheres tend to get more mass than larger spheres, based on the assumption that the bulk density in regions with smaller spheres is lower than in regions with larger spheres.

We have tried to use the heuristics for our new adaptive sphere packings, but the error for the gravitational acceleration on the surface generally increased. The reason for this is that the voids are not evenly distributed anymore. We can get a slight improvement with a very small λ . However, we can even further reduce the error with a new approach.

Our new approach is based on sampling the voids. Our goal is to associate the empty volume with the spheres to reduce the error for the gravitational field. Therefore we sample the empty volume and associate it with the closest spheres. Such an approach has two challenges: performance or precision and the metric used for association.

In order to obtain a good performance, we use uniform grids as an acceleration data structure. This enables us to find the closest sphere for each sampling point quickly.

In detail, we start with the initialization of the grid, which is described in Algo. 2.

Actually, we use two uniform grids; one for the triangles and one for the sphere packing, which we insert into the grid. The memory limits the cell size for the triangles grid. Choosing

Algorithm 2 InitGrid(Triangles, Spheres)

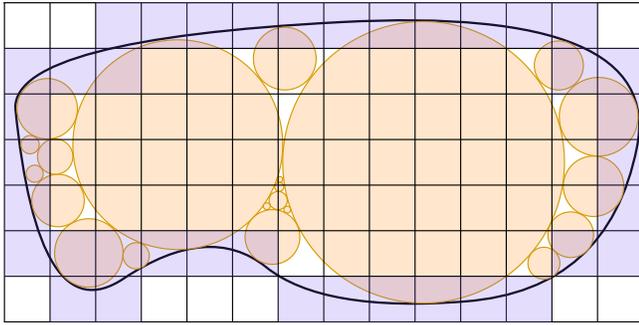
```

g ← create uniform grid and inset triangles/spheres
for all triangles t do
    mark cells of g intersected by t as border cells
end for
for all cells in g do
    generate random ray and traverse grid along that ray
    count intersections of ray with triangles c
    if ray hits vertex or edge of a triangle: then
        abandon ray and repeat with another (random) ray
    end if
    if c is even then
        mark cell as outside
    else
        mark cell as inside
    end if
end for

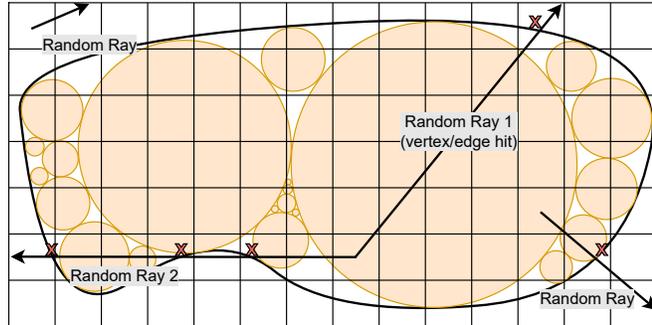
```

them as small as possible has led to the best performance. However, this was not true for the spheres. Here we worked with fewer cells per axis, achieving better performance. Cells containing triangles are marked as *border* cells (Fig. 2a). These cells catch the silhouette of the body and contain the volume inside and outside the body. We also determine the *outside* and *inside* cells, which definitely contain no triangles and lie far outside or deeper inside the body. So these cells contain either ultimately the body’s volume or no volume of the body. To classify them, we shoot a random ray from the cell center. Then we traverse the cells along the ray and test for ray and triangle intersection. With the number of intersections, we determine if the cell is completely inside or outside the body (Fig. 2b). We can see an example of marked cells in Fig. 2c. If a ray intersects a vertex or edge of a triangle, it is unclear how to count this type of intersection. There exist non-trivial methods to analyze such cases, yet it is often easier to shoot another random ray.

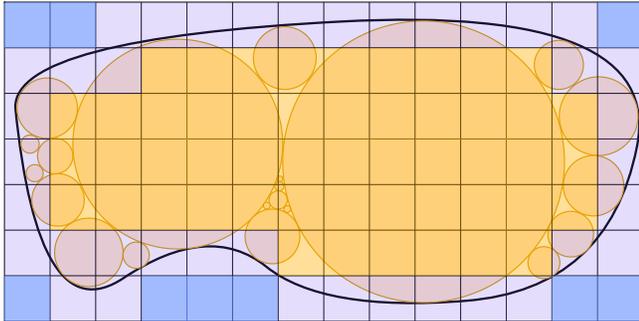
After the grid initialization, we start with the actual sampling of the empty volume (see. Algo. 3). We traverse the bounding box of the body in regular steps (Fig. 4a). The point can lie in three types of cells: outside, border, or inside. If the point lies outside, it is discarded. We can be sure that inside points lie inside the body. So we only need to check if a sphere



(a) Creating a uniform grid and inserting triangles and spheres. Also marking all cells containing triangles as border-type (light purple).



(b) Shooting random rays from non-border cell centers. Traversing along the ray through cells and counting the ray and triangle intersections. Even hits \rightarrow cell is outside; Odd \rightarrow cell is inside. Random ray shooting is repeated for vertex or edge hits.

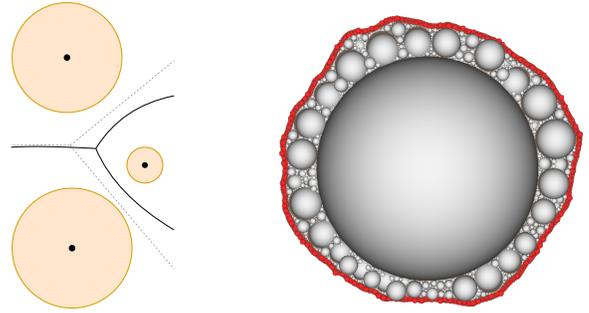


(c) Resulting segmentation of the body into cells. Cell are marked as follow: Outside is royal blue, border is light purple and inside cells are yellow.

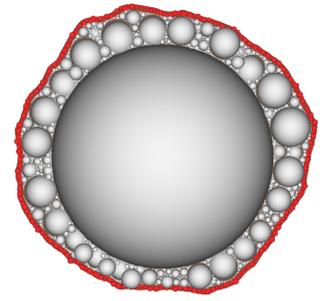
Figure 2: Acceleration data structure to speed up the sampling algorithm. It allows discarding many points that are certainly inside or outside the body's volume.

overlaps the point. If so, the point is discarded. However, for the border cells, we need to check if the point is inside. We do this by shooting a random ray, as before, for the cell centers in the initialization step. If it results in the point being inside, we must test for overlap with spheres.

For each sampling point, we search the closest sphere. As a metric, we use the Euclidean distance to the surface. We have tried different metrics, but this has delivered the best results. For example, if we have used the distance to the sphere centers, the larger spheres got less mass assigned, which in the end led to a larger error for the gravity field (Fig. 3a). We search the nearest sphere in the current cell for the sampling point. If there are no spheres, we search in



(a) Voronoi Diagram with point and circle set. Straight dashed grey lines divide space in the point set and hyperbolic arcs in the circle set. The larger spheres can get more volume or mass around assigned in the circle set.



(b) Sphere packings allow the modeling of inhomogenous density distributions. Cross-section of Benu test case modeled with a regolith layer (red spheres) and a core (gray spheres) and a total of 800k spheres.

Figure 3: Sphere packings only cover part of the volume, which must match an asteroid's total mass. Different metrics for area (mass) assignment are shown in 3a for a sphere packing (3b).

neighbor cells (blue cells, see Fig. 4b). If we find a minimal distance to a sphere, we use this distance as a search circle to ensure that no spheres are closer to the sampling point (purple circles, see Fig. 4b). Each sampling point adds weight to the sphere. We compute the volume that is not covered by the triangle mesh and sphere packing. Knowing the density, we can compute the mass for this volume. This mass is then added to the sphere's mass as a fraction of the weight and the total sum of weights.

Algorithm 3 MassAssignment(Stepsize, Grid)

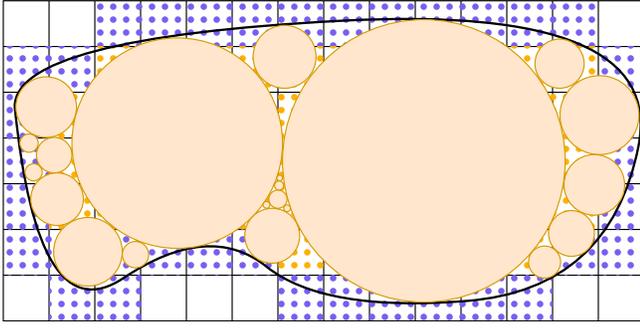
```

sample the bounding box with step size
for all sampling points  $p$  do
  if  $p$  in outside cell then
    continue
  else if  $p$  in border cell then
    test if  $p$  is inside with ray cast
    if  $p$  is not inside then
      continue
    end if
  end if
   $d \leftarrow$  search in current cell nearest sphere distance
  if no spheres in cell then
     $d \leftarrow$  search min.  $d$  in neighbor cells
  end if
  search in radius  $d$  for possible closer spheres
  add weight for nearest sphere
end for
for all spheres  $s$  do
   $s.mass \leftarrow \frac{s.weight}{sumWeights} \cdot uncoveredMass$ 
end for

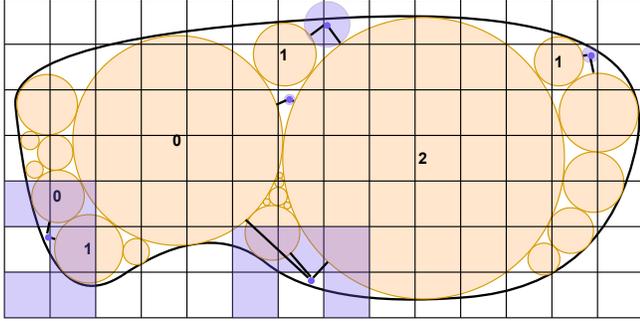
```

Mass Optimized Packings

If the gravitational field for a body is known (e.g., by measurements or by using the polygonal method), we can even further optimize the mass distribution to the spheres. This can be formulated as an optimization problem. Since our spheres have a fixed position in space, we only have the mass as an unknown, which is a linear optimization. We can solve it with the least squares method, which minimizes the error in Eq. 1.



(a) Sample empty volume in the body in regular steps. Points lying in outside marked cells are discarded. Inside points (yellow) are checked with spheres and border points (purple) need an additional inside test.



(b) Searching nearest sphere (distance to the surface) for a sampling point. Search spheres in the current cell or neighbor cells (purple cells). If the minimum distance is found, search in this minimal distance circle (purple circle). Points add weights to spheres.

Figure 4: The nearest sampling points add weight to a sphere. At the end: The sum of weights adds a fraction of mass to a sphere.

$$\arg \min_{\theta} \|X\theta - Y\|_2^2, \quad X \in \mathbb{R}^{n \times m}, Y \in \mathbb{R}^m \quad (1)$$

We take the gravitational potential (Y) as the target value. In our examples, we compute the gravitational potential with the polyhedral method, which results in a m -dimensional vector. We have many masses and points where we compute the gravitational potential. These inputs are represented mathematically by an $n \times m$ matrix X . The rows are the samples around the body, and the columns encode the masses or unknowns (θ). It is possible to determine the solution analytically through the normal equations:

$$\theta = (X^T X)^{-1} X^T Y \quad (2)$$

However, we would receive negative masses or densities. Therefore we constrain θ to be non-negative. We used a Python wrapper for the FORTRAN non-negative least squares solver [22]. For our scenario, it performed better than other solvers. We implemented a standard gradient descent and coordinate descent algorithm and tried other solvers like FNNLS [23]. To summarize: The solvers were faster but delivered a significantly less accurate solution. Therefore we kept using the Python wrapper [22].

4. RESULTS

We have used three test cases to evaluate our adaptive sphere packing. In the first two cases, we assume a homogeneous density, and in the last case an inhomogeneous density:

- Homogeneous Cube
- Homogeneous Phobos
- Inhomogeneous Bennu

We compare the gravitational acceleration g on the body's surface for each case. Since the error decreases with distance, we measure the largest possible errors here, i.e. directly on the surface. Actually, we placed our evaluation points at the center of each triangle, representing the body's surface.

We either compute the gravitational acceleration with the polyhedral method or analytically (cube case). The error is computed as the relative mean squared error for each evaluation point on the surface:

$$\sqrt{\frac{1}{n} \sum_{i=0}^n \left(\frac{g_{mascon_i} - g_{pm_i}}{g_{pm_i}} \cdot 100\% \right)^2} \quad (3)$$

For two reasons, the previously presented mascon sphere packing is chosen as a second reference. First, it is the most similar approach, and second, it delivers accurate results in general and better results with fewer spheres if compared to similar approaches. A relative error reduction is computed between our new adaptive sphere packing method and the similar method (standard packing and heuristic [7]):

$$\frac{RMS_{standard} - RMS_{adaptive}}{RMS_{adaptive}} \cdot 100\% \quad (4)$$

Our results were computed with double-precision floating-point numbers with an Intel® Core™ i7-12700K with 32GB RAM and a NVIDIA GeForce RTX™ 3080 Ti (CUDA) GPU. The computation for the sampling algorithm (to assign the masses in the voids to the spheres) was done in C++ in a single thread. We note that our sampling algorithm could be easily parallelized to further reduce computation time. Gravitational acceleration was computed on the GPU as in [7].

Homogeneous Model

Our first test case is a cube with an edge length of 2km. We assume a homogeneous density of $1\text{g}/\text{cm}^3$. The cube model consists of around $49k$ triangles, which corresponds to the number of evaluation points. Different ranges of spheres were targeted, from $2.5k$ to $80k$, and for each, we have chosen four packing density thresholds. The packing density thresholds are a few percent below the packing density of the standard packing. Fig. 5 shows that the largest improvement of around 12% is reached with $20k$ spheres and the threshold set to 85%. Other numbers of spheres that are below or higher tend to decrease the improvement. For both homogeneous cases we have used $G = 6.674 \cdot 10^{-11} \frac{Nm^2}{kg^2}$ as the gravitational constant.

The second homogeneous case is the Phobos model. It consists of around $274k$ triangles, and we assume a density of $1.86\text{g}/\text{cm}^3$ for our computations. We began again with a total number of $2.5k$ spheres and increased it up to $1280k$ spheres (see Fig. 6). We expected the improvement to decrease, with

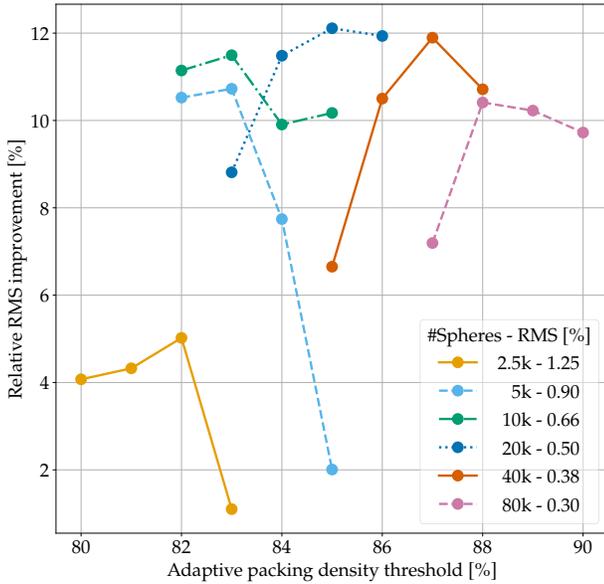


Figure 5: Homogeneous Cube. Relative error improvement with an adaptive sphere packing. Different numbers of spheres with different hyperparameters are compared. The lowest computed RMS is shown.

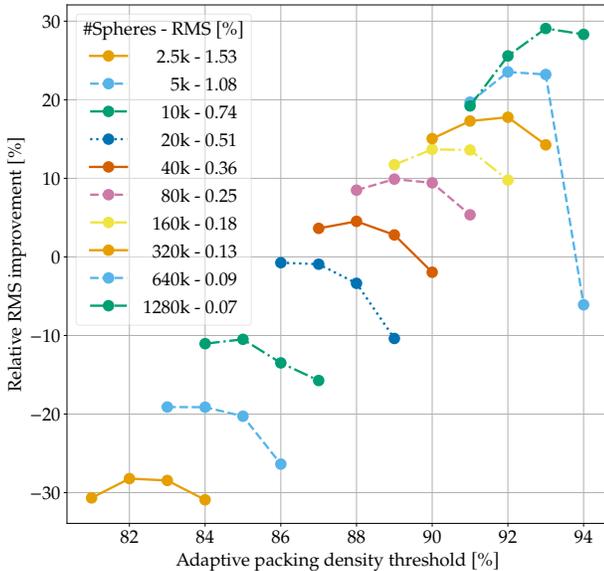


Figure 6: Homogeneous Phobos. Relative error improvement with an adaptive sphere packing. A different number of spheres with different hyperparameters are compared. The lowest computed RMS is shown.

a similar number of spheres, as in the cube model. However, the improvement seems not to decrease with an increasing number of spheres. We expect the deeper spheres to become more critical with a high number of spheres since the packing algorithm puts more spheres near the border. This hypothesis might still be true, and maybe many more spheres are needed. Another argument is that the surface is way more complex than the cube. With that, we also have more evaluation points on the surface. Hence, more spheres are needed near the surface for our sampling algorithm. It could explain why we receive a worse result with fewer spheres.

Model	Packing Density [%]		Error [%]	
	Core	Regolith	RMS	Improvement
20k	49.32	91.08	0.54	
A1 20k	51.00	88.38	0.46	14.891
A2 20k	51.20	88.94	0.45	16.023
A3 20k	50.67	90.65	0.46	14.890
40k	56.32	92.22	0.40	
A1 40k	58.43	88.82	0.34	15.335
A2 40k	58.59	89.37	0.34	17.030
A3 40k	58.69	91.13	0.33	18.879
100k	64.19	93.56	0.25	
A1 100k	65.38	89.33	0.25	0.713
A2 100k	65.60	89.88	0.23	7.250
A3 100k	66.03	91.66	0.21	14.264
800k	75.98	95.70	0.11	
A1 800k	76.18	89.96	0.16	-42.460
A2 800k	76.27	90.52	0.13	-16.618
A3 800k	76.40	91.40	0.11	-2.343

Table 1: Error for inhomogeneous Bennu modeled with a regolith layer. The core and Regolith columns indicate the packing density for the core and shell. Models with standard greedy packing and our new adaptive packings (A) are compared. The error is computed with RMS, and the relative error reduction is shown for comparison.

Inhomogenous Model

Finally, we tested our algorithms with an inhomogeneous density distribution. To do that, we used the Bennu model and modeled a regolith layer with the 3D modeling software Blender. A high-resolution mesh (inner core) was created, with a constant distance of 10m below the surface. We subtracted 0.250g/cm^3 from the mean density, which resulted in the a regolith density of 1.016g/cm^3 . Leftover mass resulted in a density of 1.301g/cm^3 for the core to keep the total mass of the body constant. For the gravitational constant, we have used $G = 6.667 \cdot 10^{-11} \frac{Nm^2}{kg^2}$.

Different adaptive packings were created for the evaluation and are shown in Table 1. We see that the best results were achieved with 40k spheres of almost 20% improvement. Yet, the improvement seems to have disappeared at 800k spheres. With a complex surface, we expect the same behavior as for Phobos. However, our regolith layer model is similar to the adaptive sphere packing. It leads the standard packing algorithm to cover much volume in the regolith layer, so we also have a better approximation near the surface. We guess this minimizes the effect of our adaptive sphere packing with a high number of spheres.

Error Analysis and Performance

A visualization of the different sphere packings can help to understand the results better: Fig. 7 shows the internal density distribution. The methods are ordered by the columns. The first method is the standard packing with the original heuristic DPVI, and the second is our new method. The third column shows the optimization method. The two rows represent the different test cases: Cube and Bennu.

To highlight the difference better, the results for the cube are in the same color range for the first two methods. Our approach distributes the mass more equally in the volume, leading to a more homogeneous image and better results. However, it is noticeable that with our method, there is more

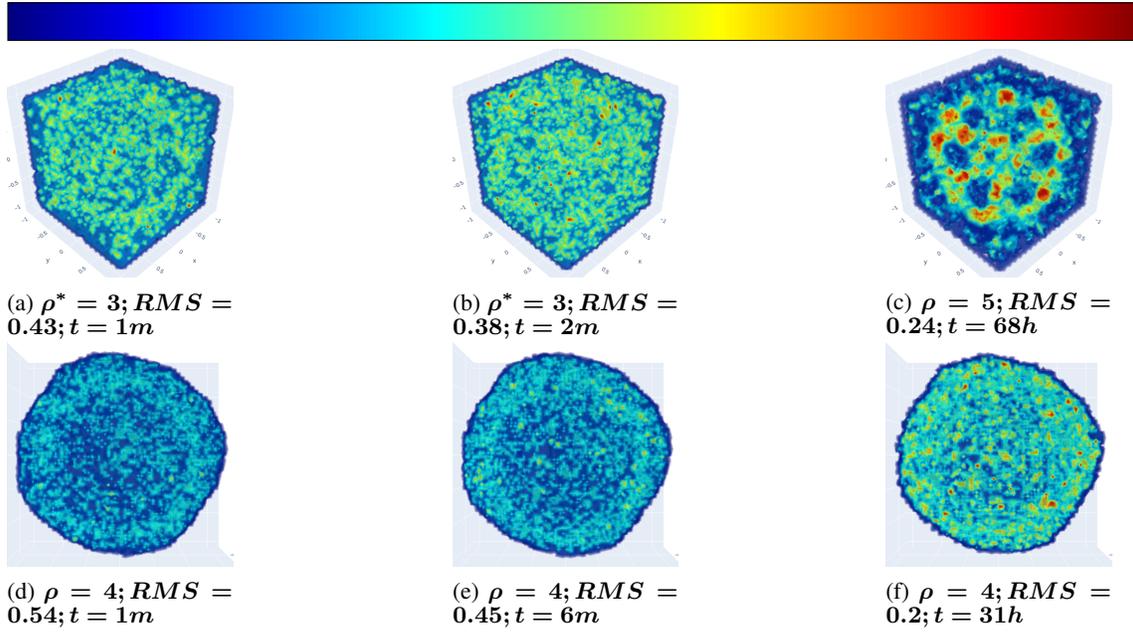


Figure 7: Internal density distribution rendered with isosurfaces from a sampled 60x60x60 density grid. Spheres are scaled up to match the mean density. Overlap areas result in higher density (red). Columns represent method: Standard packing with heuristic[5], Adaptive packing with sampling, Optimizing standard packing. Rows represent cases: Cube-40k and Benu-20k. ρ is the max density, RMS is the error, and t is the time needed to assign masses. ρ^* has adapted color scale.

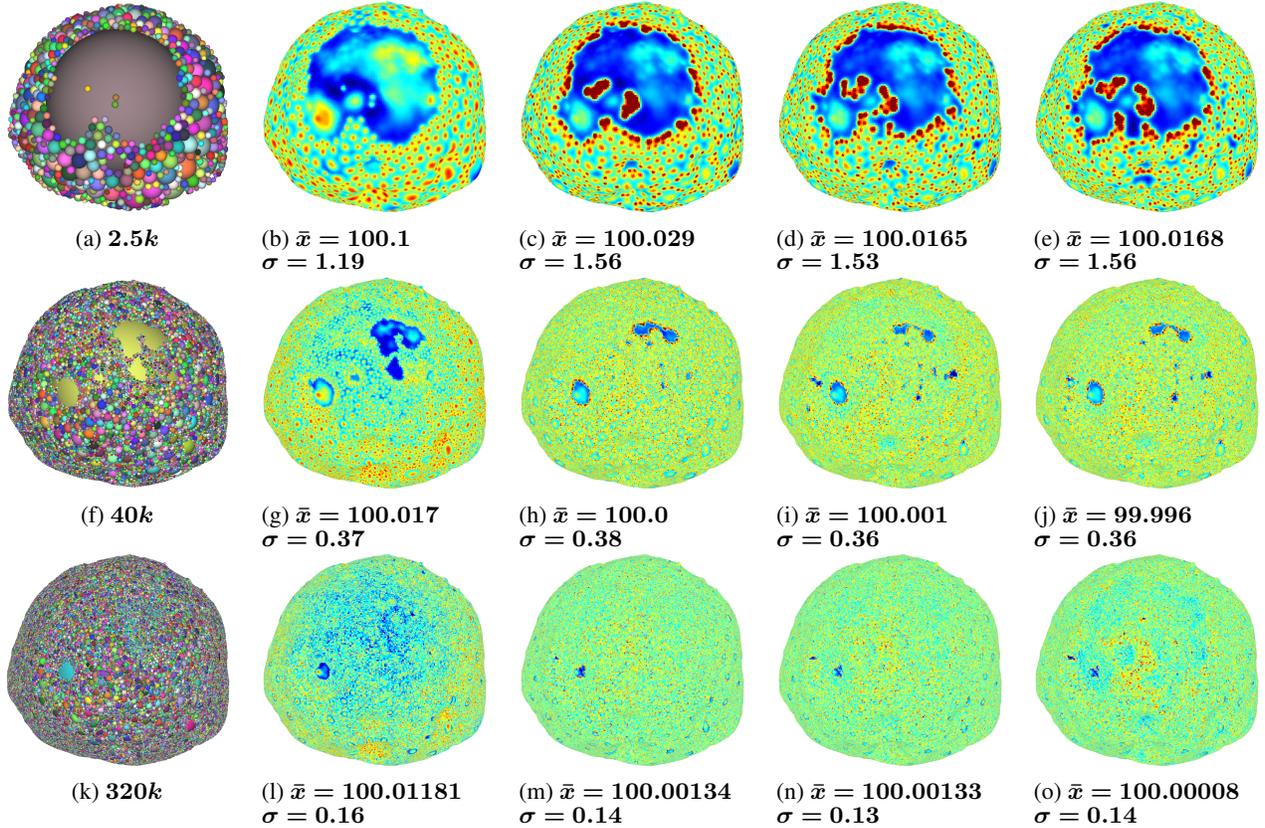


Figure 8: Relative error in percent on the Phobos surface is shown using the polyhedral method as a reference. Values above 100% are overestimated (red), and values below are underestimated (blue). The second column shows the similar method [5]. The third column starts with adaptive sphere packing with an increasing number of spheres at the surface from left to right. The color range is row-wise equal.

overshooting near the surface. This is visible by the high density near the surface in Fig. 7b. Uncovered volume near the surface does not mean we can just place more spheres near the surface to solve this issue; however, it helps. There seems to be an optimal threshold for our sampling algorithm (see Figs. 5, 6). These observations are similar for the Benu case (see Figs. 7d, 7e) as well as for the Phobos case.

Mass Optimized Sphere Packing

Finally, we have computed optimized mass distributions according to Section 3. Therefore we used a non-negative least squares solver, and the results are visible in the last column. For the cube, the error was reduced by 44%, and similar high error reductions were reached for Phobos and Benu. A notable pattern of the solver is that it works with high mass concentrations deeper inside the body. For the cube, it seems to create a rounded cube composed of edges in Fig. 7c. A thin and dense shell seems to appear for Benu in Fig. 7f. Very similar approaches are usually used for inverse gravity modeling [19]. However, different mass distributions can produce the same gravity field. Because of that, we achieved relatively inhomogeneous mass distributions. We think that the pattern the optimizer produces could be used to develop a better metric or extension, i.e., depending on the distance to the mesh surface. For now, we used the distance from the sampling point to the sphere’s surface (see the circle set in Fig. 3a), yet an extension could also include the sphere’s distance from the body surface and try to assign deeper spheres more mass.

Optimizing the mass distribution also has some drawbacks, mainly the computation time and the extensive memory usage. Fig. 7 shows the computation time for all cases. The computation time to compute the actual gravitational acceleration is running completely on a GPU, hence it is the same order of magnitude as in [7]. As previously stated, the mass assignment algorithm’s performance depends on the number of sampling points. Only a negligible amount depends on the number of spheres. There are fewer sampling points inside the body if we have more spheres, thus shorter run-time. We observed that the computation time does not vary significantly between the heuristic and the sampling (minutes). However, the optimization (hours and days) is a massive difference. Nevertheless, less input gravity data or fewer mascons to optimize can lead to better solutions with a justifiable effort.

Because we see a noticeable overshoot near the surface, we have visualized the gravitational acceleration error on the surface for Phobos (viewing the front face, see Fig. 8). Again, the error was computed as a relative error to the polyhedral method. Overestimates are colored red, and the underestimates are blue. Each row represents a total number of spheres, whereas the second column shows the previous approach [5]. Our approach starts from the third column, where we increase the number of spheres near the surface. For $2.5k$ spheres, we can see that our approach overshoots generally and pretty heavily (Figs. 8c, 8d, 8e). The issue seems to be that there is a lot of uncovered volume near the surface that is assigned to the small surface spheres that then stick out of the surface. These mascons are significant for the error since they are near the surface and thus near our evaluation points. The error \bar{x} is consistently over 100%, and the standard deviation seems relatively high. With $40k$ spheres, the smaller spheres start to cover the large sphere. This is even more visible for our approach (Figs. 8h, 8i, 8j). Here, the previous pattern is also visible on a smaller scale. Smaller spheres around larger spheres near the surface get

more mass assigned. These are the small red circles, where in the center, there is a larger sphere below the surface. Larger spheres seem to get less mass assigned, compared to the previous approach (compare Figs. 8l, 8m). The above observations also seem to be true for $320k$, where we achieved a smoother surface gravity. Overall our λ -parameter or number of spheres near the surface can lead to an overall overshoot or undershoot. There is more fluctuation with fewer spheres. For all our test cases, we have reached errors close to or below 1%, which is essential for practical applications. In terms of mission planning and theoretical astrodynamics, any adequate mathematical representation with accuracy near or within 1% of the homogeneous polyhedron may be fully considered, as emphasized by [15].

5. CONCLUSION AND FUTURE WORK

In this study, we have presented a new method to model the mass of irregular-shaped and inhomogeneous bodies. First, we pack non-uniform spheres inside the body and control the number of spheres near the surface through a hyperparameter. Second, we introduce a sampling algorithm, which distributes the missing mass of the voids to the spheres. The goal is to assign the empty volume around the spheres as a mass to the spheres. Therefore we sample the uncovered volume, and each sampling point adds weight to its nearest sphere. We accelerate the computation through a uniform grid as a spatial data structure.

We have used three test cases to evaluate our method: a synthetic cube and polygonal models of Phobos and Benu. We computed the gravitational acceleration using the polyhedral method and measured the relative mean squared error. Additionally, we have compared it to a similar state-of-the-art approach. The strengths and weaknesses were analyzed. We conclude that our method can reduce the error by almost 30% if compared to a similar method and thus allow us to model more accurate gravitational fields.

In the future, developing other metrics that do not lead to local anomalies would be interesting (i.e., avoiding high-density peaks). Another approach could be a filter that dynamically smooths those regions. Optimizers show there is still some room for reduction. Moreover, the general approach of representing mascons as polydisperse sphere packings can be used to also solve the inverse problem, i.e. computing a mass distribution from a measured gravitational field. Spheres could allow the modeling of different material mixtures more elegantly, which could be an advantage over the other methods.

ACKNOWLEDGMENTS

M.N. acknowledges funding from the Foundation of German Business (sdw) and the Royal Observatory of Belgium (ROB) PhD grants. The work by Bundeswehr University was carried out in the frame of project KaNaRiA-NaKoRa, financed by the German Ministry of Economy and Energy and administered by the German Aerospace Center, Space Administration (DLR, Deutsches Zentrum für Luft- und Raumfahrt, FKZ 50NA1915). The work on the part of University of Bremen presented in this paper was partially funded by DLR under grant 50NA1916.

REFERENCES

- [1] P. Michel, M. Küppers, A. C. Bagatin, B. Carry, S. Charnoz, J. d. Leon, A. Fitzsimmons, P. Gordo, S. F. Green, A. Hérique, M. Juzzi, Karatekin, T. Kohout, M. Lazzarin, N. Murdoch, T. Okada, E. Palomba, P. Pravec, C. Snodgrass, P. Tortora, K. Tsiganis, S. Ulamec, J.-B. Vincent, K. Wünnemann, Y. Zhang, S. D. Raducan, E. Dotto, N. Chabot, A. F. Cheng, A. Rivkin, O. Barnouin, C. Ernst, A. Stickle, D. C. Richardson, C. Thomas, M. Arakawa, H. Miyamoto, A. Nakamura, S. Sugita, M. Yoshikawa, P. Abell, E. Asphaug, R.-L. Ballouz, W. F. Bottke, D. S. Lauretta, K. J. Walsh, P. Martino, and I. Carnelli, “The ESA Hera Mission: Detailed Characterization of the DART Impact Outcome and of the Binary Asteroid (65803) Didymos,” *The Planetary Science Journal*, vol. 3, p. 160, Jul. 2022.
- [2] H. F. Levison, C. B. Olkin, K. S. Noll, S. Marchi, J. F. Bell, III, E. Bierhaus, R. Binzel, W. Bottke, D. Britt, M. Brown, M. Buie, P. Christensen, J. Emery, W. Grundy, V. E. Hamilton, C. Howett, S. Mottola, M. Pätzold, D. Reuter, J. Spencer, T. S. Statler, S. A. Stern, J. Sunshine, H. Weaver, and I. Wong, “Lucy Mission to the Trojan Asteroids: Science Goals,” *The Planetary Science Journal*, vol. 2, p. 171, Oct. 2021.
- [3] A. Probst, G. G. Peytavi, B. Eissfeller, and R. Foerstner, “Mission concept selection for an asteroid mining mission,” *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 88, no. 3, pp. 458–470, 2016.
- [4] M. Okabe, “Analytical expressions for gravity anomalies due to homogeneous polyhedral bodies and translations into magnetic anomalies,” *Geophysics*, vol. 44, no. 4, pp. 730–741, 04 1979. [Online]. Available: <https://doi.org/10.1190/1.1440973>
- [5] A. Srinivas, R. Weller, and G. Zachmann, “Fast and accurate simulation of gravitational field of irregular-shaped bodies using polydisperse sphere packings.” in *ICAT-EGVE*, 2017, pp. 213–220.
- [6] R. A. Werner and D. J. Scheeres, “Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia,” *Celestial Mechanics and Dynamical Astronomy*, vol. 65, no. 3, pp. 313–344, 1996.
- [7] H. Meißenhelther, M. Noeker, T. Andert, R. Weller, B. Haser, Ö. Karatekin, B. Ritter, M. Hofacker, L. Balestrero Machado, and G. Zachmann, “Efficient and Accurate Methods for Computing the Gravitational Field of Irregular-Shaped Bodies,” in *2022 IEEE Aerospace Conference*, 2022, pp. 1–17.
- [8] Y. Takahashi and D. Scheeres, “Morphology driven density distribution estimation for small bodies,” *Icarus*, vol. 233, pp. 179–193, 2014.
- [9] M. Noeker, H. Meißenhelther, T. Andert, R. Weller, Ö. Karatekin, and B. Haser, “Comparing methods for gravitational computation: Studying the effect of inhomogeneities,” *Copernicus Meetings*, Tech. Rep., 2022.
- [10] M. Noeker and Ö. Karatekin, “The wedge-pentahedra method (WPM): Topographic reduction of local terrain in the context of solar system surface gravimetry and robotic exploration,” *Frontiers in Space Technologies*, vol. 3, 2022. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frspt.2022.982873>
- [11] R. A. Werner, “Spherical harmonic coefficients for the potential of a constant-density polyhedron,” *Computers & Geosciences*, vol. 23, no. 10, pp. 1071–1077, 1997.
- [12] Y. Takahashi, D. J. Scheeres, and R. A. Werner, “Surface gravity fields for asteroids and comets,” *Journal of guidance, control, and dynamics*, vol. 36, no. 2, pp. 362–374, 2013.
- [13] Y. Takahashi and D. Scheeres, “Small body surface gravity fields via spherical harmonic expansions,” *Celestial Mechanics and Dynamical Astronomy*, vol. 119, no. 2, pp. 169–206, 2014.
- [14] M. Antoni, “A review of different mascon approaches for regional gravity field modelling since 1968,” *History of Geo- and Space Sciences*, vol. 13, no. 2, pp. 205–217, 2022. [Online]. Available: <https://hgss.copernicus.org/articles/13/205/2022/>
- [15] S. Tardivel, “The limits of the mascons approximation of the homogeneous polyhedron,” in *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5261.
- [16] R. Weller and G. Zachmann, “Protosphere: A gpu-assisted prototype guided sphere packing algorithm for arbitrary objects,” in *ACM SIGGRAPH ASIA 2010 Sketches*, 2010, pp. 1–2.
- [17] T. Chanut, S. Aljbaae, and V. Carruba, “Mascon gravitation model using a shaped polyhedral source,” *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 4, pp. 3742–3749, 2015.
- [18] A. Rathinam and A. G. Dempster, “Octree-based mascon model for small body gravity fields,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 11, pp. 2557–2567, 2019.
- [19] D. Scheeres, B. Khushalani, and R. A. Werner, “Estimating asteroid density distributions from shape and gravity information,” *Planetary and Space Science*, vol. 48, no. 10, pp. 965–971, 2000.
- [20] A. Colagrossi, F. Ferrari, M. Lavagna, K. Howell *et al.*, “Dynamical evolution about asteroids with high fidelity gravity field and perturbations modeling,” in *Advances in the Astronautical Sciences (Proceedings of the AIAA/AAS Astrodynamics Specialist Conference)*, vol. 156, 2015, pp. 885–903.
- [21] R. P. Russell and N. Arora, “Global point mascon models for simple, accurate, and parallel geopotential computation,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 5, pp. 1568–1581, 2012.
- [22] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1995.
- [23] R. Bro and S. De Jong, “A fast non-negativity-constrained least squares algorithm,” *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 11, no. 5, pp. 393–401, 1997.

BIOGRAPHY



Hermann Meißenhelger received his B.S. and M.S. in computer science from University of Bremen. He joined the institute Computer Graphics and Virtual Reality at the University of Bremen in 2019. Currently, he works as a research associate at that institute. His scientific interests are packing problems and collision detection.



René Weller is a postdoctoral research assistant at the Computer Graphics Group at the University of Bremen, where he also received his Ph.D. degree in computer science in 2012. He is working in the field of computer graphics and virtual reality with a specialization in collision detection algorithms.



Matthias Noeker received his M.Sc. degree in Aerospace Engineering (TU Delft) in 2018, his B.Sc. degree in Mechanical Engineering (University of Siegen) in 2016, and his BEng degree in Mechanical and Manufacturing Engineering (University of Portsmouth) in 2015. He is currently a PhD candidate at the Royal Observatory of Belgium and at UCLouvain, working on the GRASS gravimeter development and gravity modelling.



Tom Andert received his diploma in 2004 and his Ph.D. in 2010 in Geophysics from the University of Cologne, Germany. He is currently a senior researcher and leads the Radio Science research group at the Institute of Space Technology and Space Applications (ISTA) at the Bundeswehr University in Munich. His scientific interests include among others measuring and interpreting the gravity field of small solar system bodies using radio science data from interplanetary missions.



Gabriel Zachmann is a full professor with the computer science department at the University of Bremen, Germany. He is the head of the visual computing group, focusing mostly on computer graphics and virtual reality. Prior to that, he was a professor for 7 years at the Technical University of Clausthal, Germany, where he established a research group for computer graphics. He has worked in virtual reality, geometric computing, and visual computing in general for 25 years.