

Asterodrome: Force-of-Gravity Simulations in an Interactive Media Theater

Marcel Köster^{1,2}, Michael Schmitz¹, Soenke Zehle¹, and Burkhard Detzler¹

¹ xm:lab, Academy of Fine Arts Saar, Saarbrücken Germany

² German Research Center for Artificial Intelligence (DFKI), Saarbrücken Germany
marcel.koester@dfki.de, {m.schmitz,s.zehle}@xmlab.org,
detzler@hbksaar.de

Abstract. This paper presents an interactive entertainment experience deployed in a configurable surround projection environment that we called the *media theater*. This immersive installation is based on a real-time space simulator displaying several asteroid belts with large amounts of objects orbiting a sun. The sun acts as a dynamic center of gravitation that can be controlled by users with the help of a tracking system. Furthermore, it allows users to experiment with gravity in order to see how their interaction manipulates all objects in space.

Keywords: Virtual reality, interactive environment, interactive installation, physics simulation, n-body simulation, motion tracking, projection mapping.

1 Introduction and Motivation

The project described in this paper exemplifies our cross-institutional and cross-domain approach to design games and interfaces for media architectures. We conduct these experiments as part of a broader inquiry into the future of immersion, with a particular emphasis on immersive storytelling and trans-medial scenarios that unfold across platforms. This kind of research not only requires a multidisciplinary methodology but also the use of complementary research infrastructures. For instance, some of the common experiments we conduct are performed on media facades, in so called green boxes using a markerless motion-capture system and some in the newly created interactive installation introduced in this paper.

We designed a configurable media environment called *the media theater* (see Figure 1) for various exhibition setups, realized in form of a flexible prototyping environment using projection mapping techniques. A first use case of our setup is an enhanced and extended version of a real-time gravity simulation framework by Köster et al. [7] for interactive installations like media facades and virtual-reality environments. In this scenario, we designed and integrated multi-user interaction with several asteroid belts orbiting an adjustable sun.

In the field of related space simulations, many different applications have been developed in the past. Some applications focus on the visualization and

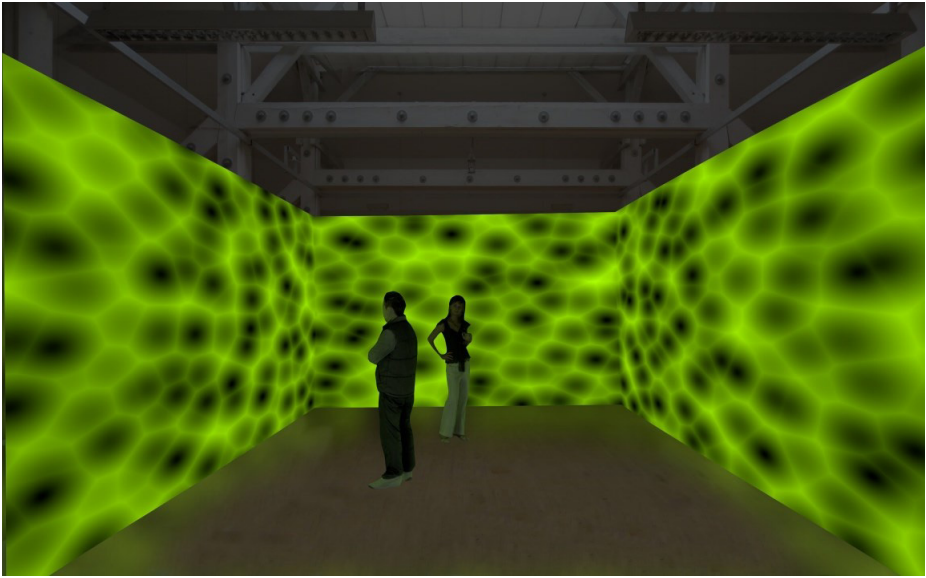


Fig. 1. Sample setup of the media theater (configuration *deep-u*) with five projectors (two on the left, one in the front and two on the right)

exploration of gathered data sets. An example for such a system is *Cassie 3D* [8]. It was invented by the NASA and allows users to discover the details behind the Saturn Cassini mission, but without allowing further explorations of the outer space. Other applications focus on the simulation of fictional scenarios and experiments with gravity, like the *Universe Sandbox* [5], which is most similar to our framework. It offers a gravity engine based on Newton's laws of gravitation. Users can play with the stars and can watch the consequences of their influences. However, all available space simulations rely on desktop-interaction concepts and do not offer support for sophisticated interactive installations.

2 Simulation Framework

As stated by Köster et al. [7], each media facade requires a unique adaption of a specific interaction scenario. This is necessary to take advantage of the capabilities of the facade and to adapt to the requirements. This statement also holds true for other interactive installations like the presented *media theater*. Examples for such requirements are specific hardware and projection-display setups, as well as the available interaction features.

Our simulation framework is designed for media facades and interactive installations. The framework can adapt to arbitrary projection setups and supports different interaction methods. The core ingredients are a parallel force-of-gravity solver and an appropriate renderer that can handle the desired setups.

2.1 Rendering

In our space setting, several thousand objects are commonly visible in parallel and need to be rendered in an appealing way. Moreover, each object in space should appear as a unique object in the scene. These requirements are satisfied using a custom dynamic LOD renderer. It makes heavy use of procedurally generated geometry (for planets and asteroids) and adaptive LOD groups for triangular meshes. The actual rendering process of objects of a certain group is then performed with the help of geometry instancing [10]. This technique allows for rendering of many objects with the same base geometry in an efficient way. The dynamically adapted LOD for near objects is realized with the help of on-the-fly tessellation capabilities combined with geometry-displacement mapping [4]. Depending on the camera setup, the current performance and the current areas of interest from the observers' perspectives, we can automatically switch to image-based alternatives like parallax mapping [11] or bump mapping [2] that create the illusion of a displacement of the object geometry.

2.2 Physics

The physics part of the framework realizes a straight-forward force-of-gravity simulation. The gravitational-force between two objects is given by Newton's law of gravitation. The computation of the required forces between n different objects (or bodies) implies that every body can influence all other bodies. The force of gravity that is applied to the i -th object of such a simulation [9] is given by

$$\mathbf{F}_i := Gm_i \cdot \sum_{j \in \{1, \dots, N\} \setminus \{i\}} \frac{m_j \hat{d}_{ij}}{|d_{ij}|^2}.$$

where

- G is the gravity constant $\approx 6.674 \cdot 10^{-11} \frac{m^3}{kg \cdot s^2}$,
- m_i is the mass of object i ,
- d_{ij} is the distance vector that points from object i to object j and
- \hat{d}_{ij} is the normalized distance vector.

Hence, we have to evaluate the forces F_i for all objects. An update of the acceleration vectors and the underlying object velocities and positions is performed once per step. However, in order to improve accuracy, multiple steps of the simulation per frame are performed. This avoids the occurrence of large force vectors, for instance.

In order to compute those vectors and performing the according position and velocity updates in real time, we can leverage the provided parallel computational power of the graphics-processing unit (GPU). The computation itself uses an hierarchical force-of-gravity algorithm related to the one of Barnes et al. [1]. If the number of objects falls below a certain threshold, we will adaptively switch to the direct $O(n^2)$ approach. However, we also perform several arithmetic and data-access pattern specific optimizations to improve the performance on a variety of different GPU accelerators.

For a proper realization of collisions, continuous collision detection based on spheres and ellipsoids is used. This kind of collision detection is perfectly suitable for the simulation: Nearly all objects in space can be roughly approximated by a sphere or an ellipsoid. Thus, collisions of two objects can be easily computed. It turned out that complex collision-detection functionality on the level of polygons or multiple spheres/ellipsoids cannot be distinguished from the approximated approach by an observer when using a large number of space objects. Furthermore, the collision logic can split or smash objects depending on their masses, velocity vectors and materials. If an object is smashed, it will be completely removed from the simulation. In the case of a subdivision operation, the object fragments will also be added to the internal simulation and will be treated as standalone objects.

3 Media Theater

The *media theater* was developed and installed as a configurable platform for various use cases: We aimed at supporting classic film screening situations as well as immersive interactive installations that employ a surround display plus ground projections. The theater is deployed in a large room under the roof with a pitched roof area and several roof beams that were used to frame the projection curtains and to fix projectors and cameras.

Six curtain elements shape a rectangular area with a 2:1 proportion (i.e. two curtain elements at the long sides and one curtain element for the short side). Each curtain element is the canvas of one projector and can be opened or hidden independently, such that a large number of configurations is supported.

Examples for possible setups are sketched in Figures 1 and 2. The first setup uses a so called *deep-u* format, in which visitors can enter a deep interactive installation. It is realized with two long projection curtains. Both sides use two projectors, whereas the rear uses a single projector. Additional bottom-projection from the top is also possible. In contrast to this configuration, the second setup focuses on the *long-u* format. It enables a more convenient and a more appealing setup for a larger number of observers and visitors (see Figure 3).

The general tracking capabilities feature arbitrary devices via a wireless network and motion tracking. In the case of external devices (like mobile phones), we rely on the support of the standard protocol for tangible user interfaces [6] (TUIO). This interaction support has been integrated to easily connect to the huge variety of available devices and applications. However, in order to provide a natural interaction concept and to animate a broad spectrum of observers and visitors to interact with a scenario, we also decided to integrate motion tracking. In our case, we make use of two depth cameras (two Asus Xtion Pro Live¹).

We decided to use infrared-based cameras due to their robustness against external influences (like dramatic changes in term of lighting).

¹ | http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

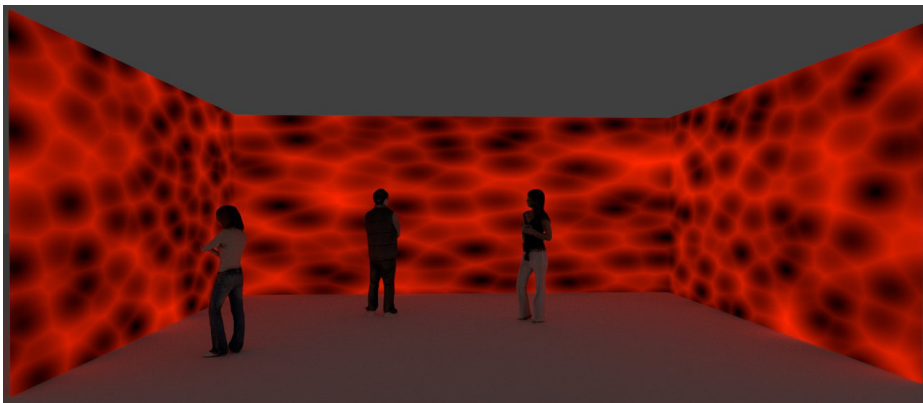


Fig. 2. Sample setup of the media theater (configuration *long-u*) with four projectors (one on the left, two in the front and one on the right)

Processing of raw-data information is performed by a so called interaction node that is an additional input-processing machine. It is connected via network to the actual simulation and rendering machines. This allows for high-performance and separate processing of large amounts of input information in form of CPU or GPU-based processing.

4 Use Case Deployment

The used configuration of the *media theater* is the *long-u* configuration (see Figure 2) with an additional bottom projection from the ceiling. This setup involves six projection surfaces: one on the left (plane 1), two in the front (plane 2), one on the right (plane 3) and two on the bottom (plane 4) (sketched in Figure 3).

The used scenario consists of three distinct (in the beginning) stable asteroid belts with different types of asteroids that are orbiting a central sun. We decided on different asteroid belts instead of a default solar system (like ours) since we want to ensure a continuously interesting simulation: A large number of objects in space appears overwhelming and fascinating. Furthermore, this variety of objects ensures continuous chain reactions and collision cascades. A common initial number of asteroids in the scope of the media theater is around $2^{17} = 131072$. This number can dramatically increase due to collisions and chain reactions. Only really small object fragments will be removed from the simulation upon collision, since they typically cannot be seen by an observer. After several interaction events and collisions in space, the average number of objects in the scene is around $280000 \approx 2^{18}$.

In contrast to previous work by Köster et al. [7], we also leverage temperature information combined with more detailed material specifications of space objects for rendering, mass computation and collision behavior. The previously used analogy for object sizes was: large objects in space imply a large mass, and



Fig. 3. Scenario setup with six projectors from a reference camera position

small objects imply a small mass. This slightly changes in the current setting. As soon as observers are able to definitely distinguish between light and heavy objects by their distinct appearance, we are able to slightly relax this constraint. For instance, the outer-most asteroid belt is the furthest one relative to the sun, and thus, could be hardly seen from an observer. Those asteroids can also become visible upon changing the material and temperature information to ice-like structures. The adapted sizes and the use of reflections and refractions from those asteroids will ensure a minimum visibility. A photo from the interactive installation can be seen in Figure 4.

4.1 Single- and Multi-User Interaction

As mentioned previously, the *media theater* features interaction via motion tracking from the ceiling. Two depth cameras capture every motion of visitors that are located inside of the interactive installation. Enabling user interaction for a single visitor is a non-problematic task. The user can be tracked according to the captured depth information from the interior of the installation. This location information can be used to manipulate and adjust the location of the central sun.

However, an interaction principle that allows for a natural and visitor-group-focused interaction is difficult to realize. In previous scenarios of media facades, we used multiple large-scale gravitational points in space (like multiple suns or larger planets) that could be controlled by users. This concept, however, makes a direct group-focused interaction hard, which would be a valuable choice for this setting.

Consequently, we decided to weight every visitor's influence according to the area he or she causes to appear in the depth image. People that are located close together will thus imply larger regions in the depth image. Furthermore, visitors can spread their arms to enlarge their captured regions, and therefore,

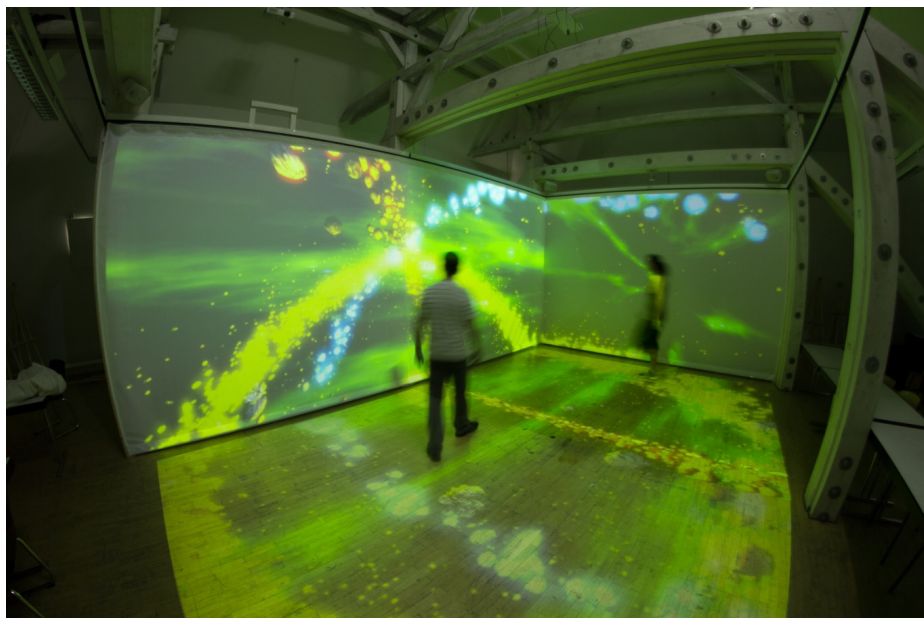


Fig. 4. Photo from the deployed scenario with the described virtual-camera configuration from Section 4.2. The photo was taken with the help of a wide-angle lens.

their weighting. The weighting factors, as well as the locations of the visitors are then used to control the sun. The target location of the sun can then be computed according to the mass center of all areas: The center of every detected visitor/group area is multiplied by its weighting factor, divided by the total weighting of all visitors and added to the target location.

In order to extract the required information from the depth images, we utilize the OpenCV library [3]. We subtract background information and try to find the largest connected regions, in which the depth differences are below a given threshold. By performing these processing steps we can determine the weights and locations of single and groups of visitors. Afterwards, we can compute the 2D location of the sun, interpolate the old location with the new location according to the elapsed time and can then send the target position to the simulation.

4.2 Projection Setup

The decision on a proper camera perspective that allows for an appealing and natural visualization for most observers is a sophisticated exercise. We decided on multiple virtual cameras oriented according to a visitor (height of the camera of about 1.7 meters above the ground) that is located in the center of the installation (around the location of the left human in Figure 3). This implies that we need four camera perspectives and have to render four images, one image with its corresponding camera perspective for each side. From a computer-graphics' perspective, we are rendering four images in an environment-mapping-based way.

This allows for a 3D-like immersive experience for visitors that are interacting with the simulation without the need for additional 3D equipment.

However, as shown in Figure 4, the perspective might not be optimal for observers that are not interacting with the installation. Experiments for adjusting and manipulating the camera positions and distortions to multiple observers on the fly would be an interesting extension in the future.

5 Conclusion

We presented an interactive installation called *media theater*. It supports many different configurations and interaction capabilities, like external devices based on TUIO or direct interaction via motion tracking. Due to its flexibility, it can be used as a rapid prototyping environment for a huge variety of interactive setups.

The installed real-time force-of-gravity simulation allows for experiments with gravity and appears fascinating due to the large number of presented space objects. Observers can directly interact with this scenario via the available motion-tracking features by manipulating a central sun. However, interaction is not limited to a specific amount of visitors, but also features interaction between multiple visitor groups. Furthermore, they are able to perceive a 3D-like experience in the center of the installation without the need for further 3D equipment.

In the future, we would like to experiment with dynamically adjusted camera perspectives to provide a multi-user 3D-like experience. Moreover, we plan to create further scenarios and to extend the interaction capabilities.

References

1. Barnes, J., Hut, P.: A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* 324 (1986)
2. Blinn, J.F.: Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph* (1978)
3. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000)
4. Cook, R.L.: Shade trees. In: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 223–231. ACM (1984)
5. Giant Army: Universe Sandbox (April 2015). <http://universesandbox.com/>
6. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: TUIO: A Protocol for Table-Top Tangible User Interfaces. In: *Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation* (2005)
7. Köster, M., Schmitz, M., Gehring, S.: Gravity Games - A Framework for Interactive Space Physics on Media Facades. In: *Proceedings of The International Symposium on Pervasive Displays, Perdis 2015* (2015)
8. NASA: Cassini (April 2015). <http://saturn.jpl.nasa.gov/multimedia/CASSIE/>
9. Nguyen, H.: GPU Gems 3. Addison-Wesley Professional (2007)
10. Pharr, M., Fernando, R.: GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation. Addison-Wesley Professional (2005)
11. Policarpo, F., Oliveira, M.M., Comba, J.L.D.: Real-time relief mapping on arbitrary polygonal surfaces. In: *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*. ACM (2005)