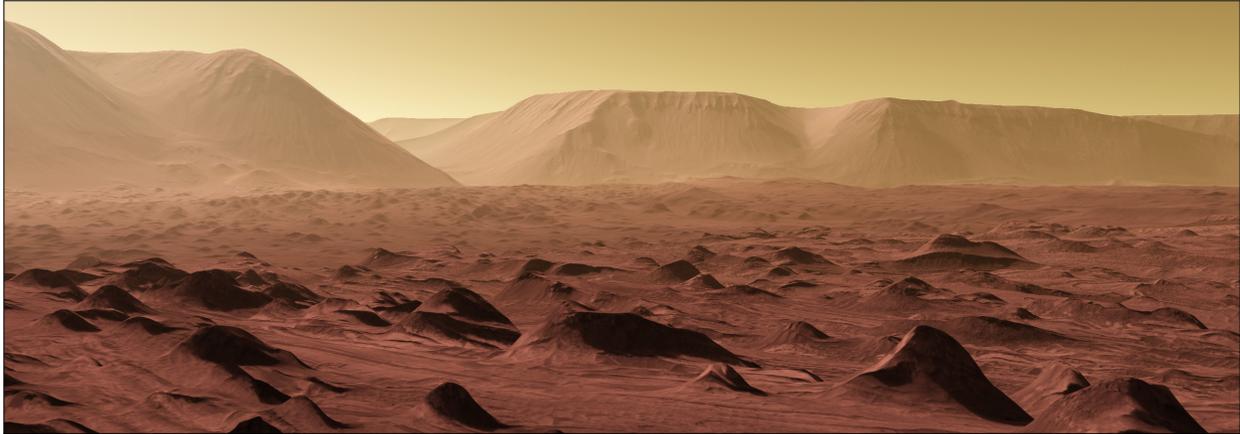


Globe Browsing: Contextualized Spatio-Temporal Planetary Surface Visualization

Karl Bladin, Emil Axelsson, Erik Broberg, Carter Emmart, Patric Ljung, Alexander Bock, *Member, IEEE*, and Anders Ynnerman, *Associate Member, IEEE*



Abstract—Results of planetary mapping are often shared openly for use in scientific research and mission planning. In its raw format, however, the data is not accessible to non-experts due to the difficulty in grasping the context and the intricate acquisition process. We present work on tailoring and integration of multiple data processing and visualization methods to interactively contextualize geospatial surface data of celestial bodies for use in science communication. As our approach handles dynamic data sources, streamed from online repositories, we are significantly shortening the time between discovery and dissemination of data and results. We describe the image acquisition pipeline, the pre-processing steps to derive a 2.5D terrain, and a chunked level-of-detail, out-of-core rendering approach to enable interactive exploration of global maps and high-resolution digital terrain models. The results are demonstrated for three different celestial bodies. The first case addresses high-resolution map data on the surface of Mars. A second case is showing dynamic processes, such as concurrent weather conditions on Earth that require temporal datasets. As a final example we use data from the New Horizons spacecraft which acquired images during a single flyby of Pluto. We visualize the acquisition process as well as the resulting surface data. Our work has been implemented in the OpenSpace software [8], which enables interactive presentations in a range of environments such as immersive dome theaters, interactive touch tables, and virtual reality headsets.

Index Terms—Astronomical visualization, globe rendering, public dissemination, science communication, space mission visualization

1 INTRODUCTION

Throughout history, the study of celestial bodies orbiting our Sun has always intrigued mankind and played a central role in mythology and culture, and contributed to the scientific discovery of the fundamental laws of physics. Since the invention of the telescope in the 17th century, our knowledge of the planets and other celestial bodies in our solar system has grown at an ever-increasing rate. The quest for more knowledge made a leap forward in 1962 when *National Aeronautics and Space Administration's* (NASA's) Mariner 2 flew by Venus and paved the path for a long-term, and still ongoing, systematic mapping of the solar system by spacecraft flying by or orbiting objects of interest. Another landmark in the exploration of the solar system was the

Viking program, launched in the year 1975, which gathered important information about Mars and its surface features from the two orbiting satellites and the landers put on the surface of the planet.

New Horizons' Pluto flyby in 2015 is yet another milestone as Pluto was the last planet¹ to be visited by a space probe. As a consequence of the spacecraft-based exploration of the solar system we now have detailed surface data from a range of celestial bodies orbiting the sun, including some located in the most remote places of the solar system.

This large collection of data has enabled numerous scientific discoveries about the geological structure of the planets and many of these have been based on the use of image processing and visualization. The work presented in this paper focuses on providing visualization tools and applications for communication of these discoveries, and the engineering efforts which enabled them. The developed application is an example of a new data-driven and interactive paradigm of science communication, which provides a means of bridging the gap between experts and the general public, but also serves as a useful tool for communication between teams of experts working on complex missions and scientific exploration. We are focusing on the use of open surface data made available by the different space agencies and research laboratories from which high-resolution and scientifically accurate representations of celestial bodies are created using geometry extraction and texturing,

-
- Karl Bladin, Emil Axelsson, Patric Ljung, Anders Ynnerman and Erik Broberg are with Linköping University. E-mail: { karl.bladin, emil.axelsson, patric.ljung, anders.ynnerman } @liu.se, eribr049@student.liu.se
 - Carter Emmart is with American Museum of Natural History. E-mail: carter@amnh.org.
 - Alexander Bock is with New York University and Linköping University. E-mail: alexander.bock@nyu.edu

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

¹When New Horizons was launched in 2006, Pluto was still defined as a planet. It has since been reclassified as a dwarf planet.

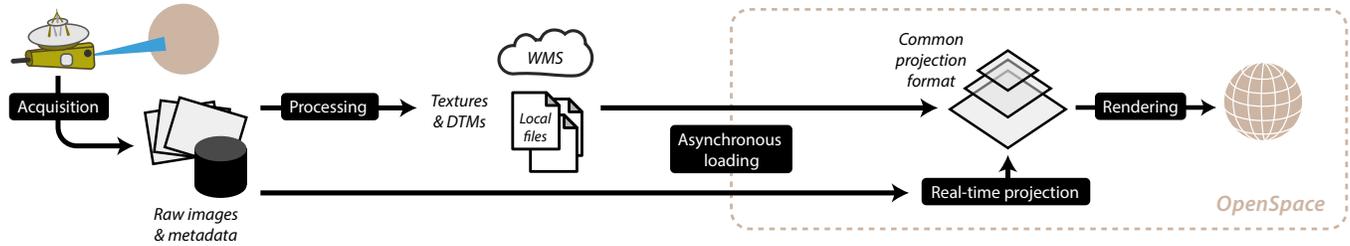


Fig. 1: Illustration of the processing pipeline of the presented application. Raw images acquired from spacecraft can be used as-is, by utilizing orientation information, and projecting images directly onto celestial bodies, or, they can be processed and made available as online global map resources. Our system supports both approaches, as in the cases with the Pluto flyby of New Horizons and the several missions to Mars. This puts high demands on the system to interactively render high-resolution images as well as reducing the time from image acquisition to dissemination.

enabling exploration in *Globe Browsing* sessions.

There are several technical challenges involved in accessing and preparing the data for interactive globe rendering which stem from aspects of collected data such as size, complexity, numerical precision and accuracy, need for curvature corrections, and incompleteness and variety of sources. Furthermore, the use of multiresolution visualization approaches to out-of-core data streaming from remote repositories and navigation models spanning from surfaces to interplanetary space requires dynamic data flow and navigation control.

We have chosen to address the challenges in the context of three representative application scenarios for different celestial bodies. As a first example we show how our system can enable interactive and seamless surface exploration of Mars with previously unprecedented resolution. We then show, as an example of temporal data, how daily images of the Earth, can be accessed and visualized from vantage points in interplanetary space as well as in close ups of regions of interest in order to communicate dynamic processes such as atmospheric events. Finally, we present how complex data gathering can be visualized by virtually following the New Horizons flyby of Pluto in 2015, and tapping into the data flows from the instruments on the probe, showing how high-resolution image mosaics are constructed, thus, dramatically reducing the time between discovery and dissemination.

At the core of our approach lies the gathering of surface image data and projecting it to a common projection format either as a preprocessing step or as an integrated part of the rendering process to visualize the data acquisition. The datasets are rendered as layers, projected on the surface of globes. Figure 1 presents an overview of this processing pipeline. In our work we followed four underpinning principles that guided the design and implementation;

- Visualization should be data-driven, and thus only data derived from the instruments on space probes should be used,
- Data from multiple instruments, and/or probes, should be merged to represent the aggregated knowledge of an object,
- The maximum available resolution in space and time should be accessible on demand for the areas of interest,
- Data should be as current as possible by enabling access to a range of curated and frequently updated repositories.

We provide a reference implementation of this system in the open source astrovisualization framework *OpenSpace*, which targets data exploration and science communication on a range of platforms, including large scale dome theaters, virtual reality headsets, and touch tables, and thus, enabling spatio-temporal navigation and contextualization of satellites and space probes together with celestial bodies.

Our contributions can be summarized as:

- A visualization pipeline and platform for contextualized multiresolution spatio-temporal data of celestial bodies,
- A clear focus on the ability to accurately visualize the acquisition of this data using the pipeline to show the accumulation of knowledge about celestial bodies.
- Three representative application scenarios demonstrating the utility and flexibility of the system.

2 RELATED WORK

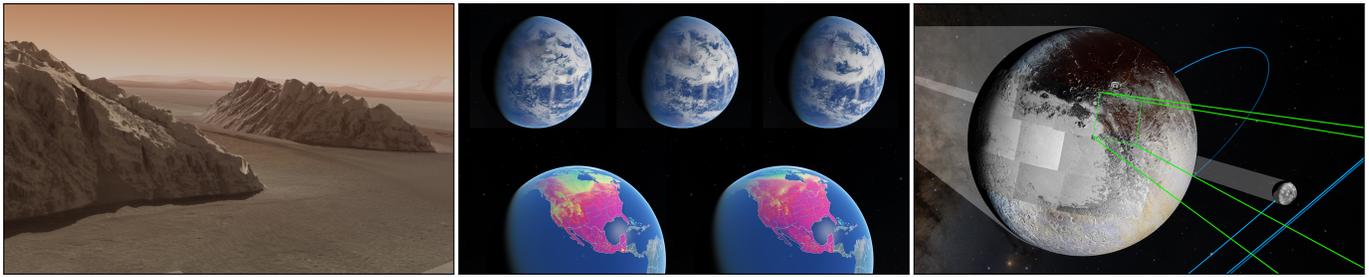
The presented application builds on technologies and methods spreading over four major topics, namely science communication, large-scale astronomical visualization, geographical information systems (GIS), and level-of-detail (LOD) rendering methods for (planetary scale) terrains. Our review of related work thus follows the same structure where we outline previous methods and approaches and conclude with commentaries on how it relates to the presented application.

Science communication — Visual communication and the use of animations and interactive techniques through computer graphics are at the core of communicating scientific findings and contextualizing information. It enables a wide appeal across ages and cultures. Virtual reality environments such as planetariums [37, 40] have always been in the focus of educators, but a shift towards supporting data inspection on consumer grade hardware is ongoing [48]. Notable examples of applications in which large scale-differences are visualized include movie productions, a classic example is *Powers of Ten* by Eames [18] from 1977, but also include current interactive software packages such as Uniview [33], DigiStar [21], and DigitalSky [54], which are all based on curated datasets containing information about our Universe [1].

Recent advances in graphics hardware and processing power in consumer hardware has enabled a broad use of actual scientific data in applications aimed for science communication for the public at science centers and elsewhere. One such story is presented by Ynnerman *et al.* [62]. The ability of providing the same data as scientists have used in their explorations to the layman user is currently a major change and a paradigm shift in science communication.

Astronomical visualization — One of the challenges in astrovisualization is dealing with positioning and navigation over extreme distances and there are several publications describing approaches to this problem, such as the notion of the ScaleGraph, introduced by Klashed *et al.* [33]. Hanson *et al.* [27] introduced the concept of *power scaled coordinates* (PSC) to handle the large numbers for astronomical distances, and subsequently Fu *et al.* [24] categorized the problem of large scale navigation and positioning. Fu and Hanson also introduced a depth buffer remapping to cover a wider range of distances than possible with a fixed point depth buffer and conventional near and far planes [23]. In our presented work we use the Dynamic Scene Graph, proposed by Axelsson *et al.* [6], as it allows for seamless navigation across large distances without loss of precision.

Relevant to the work presented here are a number of freely available astronomical software packages, such as, Eyes on the Solar System by NASA [29], also tools for planetary rendering in free software such as Blender [32, 47] can provide engaging experiences for a wide range of users. These solutions are focusing on general astronomical visualization and do not provide the out-of-core level-of-detail surface rendering required to visualize globes in high detail. Another related astrovisualization package is the *World Wide Telescope* (WWT) which brings together many data sources into a unified environment [3, 26]. Among these are static planetary datasets rendered using reprojection to the *Tessellated Octahedral Adaptive Subdivision Transform* (TOAST) format which poses issues on accuracy and in the availability of map



(a) High resolution terrain and texture on Mars

(b) Time varying datasets on Earth

(c) Image acquisition on Pluto

Fig. 2: Our system demonstrated with the three described scenarios. a) Light toned mounds in Ganges Chasma on Mars using a local HiRISE patch with 25 cm resolution and MOLA elevation with a global mosaic of CTX images and a global color basemap. b) The top row shows Earth reflectance animated over three days in June 2017. The color layer is a corrected reflectance global map taken by the VIIRS satellite. The bottom row shows the difference in maximum air temperature between June 1985 and 2015. Both layers have daily coverage provided by NASA GIBS. c) New Horizons image campaign of the Pluto system. The image shows Pluto and its moon Charon with the view frustum of New Horizons’ LORRI instrument. These scenarios are used in public presentations to explain their respective scientific discoveries to public audiences.

datasets due to flat tiles and reprojection requirements [7]. The projection is therefore more suitable for sky-tessellation as it does not require an ellipsoidal model and the dataset can be maintained in the same projection format.

There are several ongoing efforts providing access to planetary data through interactive globe visualization softwares. Cesium by Cozzi [14] is a JavaScript library which enables globe rendering in the web browser. A use case is NASA’s Mars Trek [49] which enables the user to toggle individual map layers and browse the surface of Mars in high detail. Google Earth, with the Mars feature, also enables layering of many of the datasets currently available from the ongoing missions at the planet as well as the vast amount of surface data for Earth enabling education in a wide range of subjects [51]. Similarly to NASA Worldwind, which enables tools to analyze and monitor weather patterns and visualize cities and terrain as well as temporal datasets and trajectories [28], the software is, however, not aimed towards contextualization by Universe-sized scale differences in a single unified environment.

Public outreach through web interfaces has increased drastically in recent years and we expect to see an increasing rate in flexibility by online software packages when reading and presenting scientific data immersively. Making use of the benefits of having native C++ as the core language, however, makes it possible to integrate supported libraries and software packages to enable hardware support and high performance. This is enabled using packages such as the Simple Graphics Cluster Toolkit (SGCT) [4], the Geospatial Data Abstraction Library (GDAL) [59], the SPICE toolkit [2] as well as the latest features of modern graphics APIs.

Geographic information systems (GIS) — GIS relies heavily on the ability to gather, transform, and visualize geospatial data. Texture maps and height maps; *digital elevation models* (DEMs) or *digital terrain models* (DTMs), are examples of such data and research and development of software solutions to handle different parts of GIS, such as rendering, have lead to several applications in space visualization.

A GIS software should be able to handle a vast amount of projection formats. When it comes to globe rendering, however, there are standards that are more common than others. Equirectangular geographic projection is by far the most commonly supported format used for unprojection of globes, it also plays a major role in applications outside of computer graphics, such as the *Global Positioning System* (GPS) [15].

Geographic projections suffers from well known issues near the poles such as oversampling and polar pinching [15]. The number of data products of this format, however, makes it useful for our purposes.

Level-of-detail globe rendering — LOD and multiresolution, out-of-core rendering methods are crucial when visualizing and navigating across large datasets [39]. By using lower resolution representations to render distant objects, more resources can be directed towards rendering salient objects. Cozzi and Ring [15] provide a comprehensive introduction to common methods used for globe rendering. Examples of earlier LOD techniques for globe rendering include the *real-time op-*

timally adapting meshes (ROAM), introduced by Duchaineau *et al.* [17] with improvements made by Hwa *et al.* [30]. Cignoni *et al.* [12] introduced P-BDAM as a planetary scale rendering solution requiring preprocessing and geometry generation. Although directed towards planetary visualization specifically, there were other methods such as chunked LOD [58] and geometry clip-maps [38] that better adapted to the evolution of graphics hardware, the ability to combine multiple data sources, and to generate geometry on the fly.

Kooima *et al.* [35] use a recursive subdivision of an icosahedron to construct the globe using a GPU oriented approach for vertex tessellation and texture atlases for reuse of tiles. Though the rendering algorithm nicely solves polar issues and exploit excellent triangle uniformity, restricted texture sampling precision is a limitation due to the calculation of texture coordinates being performed as an inverse projection using the global transform to the space of the projected map which is either geographic or polar stereographic. Since texture sampling is performed in the fragment shader with 32 bit floating point arithmetic, the resulting texture coordinates are unable to precisely represent imagery of details smaller than 2.39 meters for an Earth sized globe [35]. Spherical clip-maps, as presented by Claesen *et al.* [13], also do not align vertex grids with the textures which makes these methods better optimized for globes with smaller scale differences than we require.

Erikson *et al.* [20] describe a hierarchical LOD (HLOD) method generalized for three-dimensional geometries. For 2D geometries, such as chunks on a globe, each leaf node in a quad-tree corresponds to a part of the complete globe at a specific detail level. The quad-tree is dynamically updating its nodes by procedures known as *splitting* and *merging* so that high LOD is used only where it is needed.

Ulrich [58] proposed a chunked LOD technique where geometry generation of each chunk requires preprocessing in which the generation process is separated from the run-time selection of detail levels. Cozzi and Ring [15] define sub-processes of chunked LOD rendering as *selection* and *switching* which we adhere to. By defining the chunk geometry in the same hierarchical space as the texture tiles, texture sampling precision errors can be avoided [15], see Section 4.4.3.

We adopted the chunked LOD method due to its GPU friendly nature and implicit handling of polar issues that is common for geographically mapped geometry clip-maps methods. In contrast to Ulrich, we generate geometry dynamically on the GPU using vertex texture sampling, similar to Kooima *et al.* [35]. This enables dynamic loading of terrains as well as image-based level blending for textures and height maps.

Important optimization techniques for HLOD culling are described by Cozzi [16]. Horizon and view frustum culling are important procedures for our purposes to vastly reduce the number of chunks that need to be rendered without compromising image quality.

Stereoscopic reconstruction — Although it is only tangentially related to the system that we are proposing, it is valuable to place the many ongoing efforts in the reconstruction of geometry from multiple image sources into context. Koch [34] provided a framework to analyze a

scene using dense depth maps and constructing triangulated surfaces from surface points. This work spawned a large body of interesting research that are outside the scope of this manuscript. Wolf *et al.* [61] provide a useful overview over the field of using photogrammetry and stereoscopic reconstruction for terrains.

3 DATA SCENARIOS AND ACQUISITION

The available surface data is the underpinning resource upon which our framework is relying. In this section we therefore provide an in-depth presentation of the data behind the three application scenarios we present in this paper. We begin by describing the acquisition of terrain data with a high spatial resolution but limited temporal resolution for Mars (Section 3.1). We then give an overview of the dynamic time-varying data for Earth (Section 3.2). The section is concluded with a description of online image acquisition of a non-orbiting spacecraft, exemplified by New Horizons' flyby of Pluto (Section 3.3). Figure 2 shows examples of these three application scenarios.

3.1 Mars

So far 14 orbiters and 13 landers have been sent to our neighboring planet Mars, which is now the most intensively investigated planet beyond the Earth-Moon system. The most important large scale imaging campaign of Mars is carried out by the *Mars Reconnaissance Orbiter* (MRO), equipped with an array of cameras, spectrometers, and radar used to image the surface at different resolutions in part for scientific discovery, as well as for the use in landing site evaluation [63]. The satellite is in a polar orbit which means that, as the planet rotates underneath it, the instruments will eventually be able to perform measurements at every location on the planet.

The *Mars global digital image mosaic* (MDIM) is a global image dataset with a resolution of up to 240 meters/pixel [5]. The latest version of this image mosaic, MDIM 2.1 is compiled from Mariner 9 and Viking images with improved accuracy as a result of constraints from the *Mars Orbiter Laser Altimeter* (MOLA) data from the *Mars Global Surveyor* (MGS) spacecraft. The DEM, assembled from MOLA data, provides an offset from the *Areoid*, Mars' reference ellipsoid, to an average vertical precision of 100 meters with a resolution of about 450 meters/pixel [55]. This global height map provides sufficient resolution to serve as a background for other local terrain models.

The *Context Camera* (CTX) on the MRO provides gray-scale images of the planetary surface with about 6 meters/pixel resolution used for providing contextual information for the other instruments observing the Martian surface [63]. NASA Ames Research Center assembled a 5 TB global mosaic of CTX data with roughly 70% surface coverage. The CTX camera has mapped over 99% of the Martian surface after the MRO finished its 50,000th orbit in March 2017 [60].

The *HiRISE* camera on the MRO provides images with a spatial resolution of 25 cm/pixel that have been used to select and evaluate potential landing sites [43]. Both the CTX and HiRISE images can be used for stereoscopic reconstruction of DTM's that are available as local areas of interest on the surface (see Section 4.2). McEwen *et al.* [44] describes the HiRISE data available online and how the public can contribute in the choice of future target locations.

The main challenges in the Mars browsing application scenario arise from handling data products at various resolutions in a single context. This requires layer handling and the ability to seamlessly, precisely, and accurately render map data out-of-core with varying levels of detail.

3.2 Earth

Although Earth's thick atmosphere hinders the acquisition of detailed surface features, the relative ease of placing a satellite into orbit has resulted in hundreds of Earth observing satellites. Furthermore, a major design constraint for interplanetary spacecraft is the available bandwidth through the *Deep Space Network* (DSN), limiting the amount of data that can be recovered. This design constraint does not apply for Earth orbiting satellites and thus enables the creation of high spatial and temporal resolution imaging; in many cases these are global images which are updated daily, but more local and higher resolution satellites

exist, such as the geostationary GOES-16 which captures an image of the entire western hemisphere every 15 minutes at $10,000^2$ pixels.

We focused on three sun-synchronous satellites, described below, returning daily images of Earth, each covering the majority of Earth's surface area. The satellites perform scans in a large wavelength spectrum, covering both visible light as well as infrared light ($\approx 400 \text{ nm} - 14 \mu\text{m}$) and thus provide ample ground for public communication of studies of the surface, oceans, biosphere, and atmosphere. These datasets are available through a public interface provided by the NASA *Global Imagery Browse Services* (GIBS) that provides a unified interface to make NASA and *National Oceanic and Atmospheric Administration* (NOAA) imagery available to the general public [11].

The *Moderate Resolution Imaging Spectroradiometer* (MODIS) instrument has been used on the *Terra* and *Aqua* satellites since 1999 and 2000 respectively. Its resolution varies between 250 meters/pixel and 1 km/pixel [31, 52] and a global image has been generated daily since 2000. The data shows land boundaries, cloud movements, the study of phytoplankton, and global temperature studies, providing a unique view into the dynamic nature of our planet over this large timespan.

The *Visible Infrared Imaging Radiometer Suite* (VIIRS) instrument on the Suomi NPP satellite provides similar data to the MODIS instrument and has a spatial resolution between 375 and 750 meters/pixel since 2012 [53]. In addition, it has a pan-chromatic Day-Night-Band that is used to capture the night side of the Earth, thus creating the ability to study the development of cities through their light emission.

Another dataset is provided by the Center for Atmosphere Ocean Science at New York University and consists of radar measurements of glacial movements on Greenland. These measurements provide the velocity and height of glaciers as they calve and move into the Atlantic. This results in a time-varying DEM with a temporal resolution of 15 minutes.

3.3 Pluto

The images of the previous two scenarios were acquired by orbiting satellites providing the ability to produce images for an extended period of time. As these orbits are designed to have a low eccentricity, the instruments' surface resolution is constant throughout the orbit, which is very useful when reprojecting images to be used in global maps. However, not all spacecraft-based image acquisitions are performed from close to circular orbits. One recent example of a non-orbiting mapping campaign was performed in July 2015 by the New Horizons spacecraft as it was flying past the dwarf planet Pluto. Prior to the flyby, the best images of Pluto were available from Hubble with a resolution of about 161 km/pixel. The main image acquisition instrument on board New Horizons is the *Long Range Reconnaissance Imager* (LORRI), a gray-scale telescope. The *Ralph* instrument's primary data source is a Multispectral Visible Imaging Camera (MVIC) which enables the recovery of spectral information and thus provides color information for the LORRI images. The *Radio Science Experiment* (REX) is part of the radio communications system and has the ability to detect radio emissions either directly from Pluto's surface, thus measuring its surface temperature, or Earth-based radio signals that pass through Pluto's atmosphere and thus measure its composition. Between January and July, the surface resolution of the LORRI instrument gradually increased, reaching about 1 km/pixel on the day of the flyby and peaking at 50 meters/pixel during the time of closest approach.

The time varying resolutions provide an opportunity to show the incremental image acquisition, rather than using a static planetary map of varying resolution. Navigation and image data made available by the New Horizons team makes it possible to show the spacecraft's data acquisition process in the context of Pluto and its moons.

4 GLOBE BROWSING SYSTEM

In this section we describe aspects of the overall processing pipeline (see Figure 1) and the technical details required for the three representative application scenarios described in the previous section. As there are two distinct pathways into the rendering component, some of the scenarios utilize only parts of the described pipeline. More specifically, real-time projection does not utilize tiled textures.

The following sections elaborate on the individual components of our system that provide access to a large variety of imagery sources. First, the different ways of *Data Access* (Section 4.1) in our system is described. Second, we describe the *Data Preparation* (Section 4.2). If the images are not available in the correct georeferenced projection format or when detailed digital terrain models have to be generated, this is required as a preprocessing step. Third, we describe the use of *Real-Time Image Projection* (Section 4.3) used if navigational data and image times are available for specific mapping campaigns. Lastly, in Section 4.4, we describe the *Rendering System* used for our chunked LOD globe representation, utilizing the common projection format to result in the final three-dimensional globe.

In order to read as many data sources as possible without the need for additional reprojection steps, we chose to use a standardized common georeferenced map projection format; namely the equirectangular geographic projection [56]. From this georeferenced coordinate system, the renderer transforms vertices to achieve the correct mapping of the globe represented by a triaxial ellipsoid. This makes use of an inverse projection $(x, y, z)^T = P^{-1}(\phi, \theta)$ that maps each geodetic coordinate defined by a latitude (ϕ), and a longitude (θ) to a Cartesian coordinate on the surface of the reference ellipsoid in the model space of the celestial body as described by Cozzi and Ring [15].

4.1 Data Access

All of the data that can be accessed using our system has to be presented as image data. While this might place a constraint on the data types that can be visualized, it is in most cases possible to represent other interesting information as gray-scale images instead, for example as DTMs, where the pixel value corresponds to a height offset from a reference ellipsoid in the direction of the geodetic normal.

The most common access pattern for global image sources is a web-based LOD service called the *Web Map Service (WMS)* [41, 50]. While there are different dialects of this standard, it is possible to query a web server for an image covering a specific portion of a celestial body. Image tiles can either be abstracted away or be part of the access depending on the exact implementation of WMS. The access to these online datasets in our system is unified through the use of the *Geospatial Data Abstraction Library (GDAL)* [59], which enables access to a variety of WMS dialects through a single interface, thus drastically increasing the number of datasets that are accessible.

The GDAL interface also supports a vast amount of geospatial image data formats including common ones such as IMG, GeoTiff, JPEG2000, and Cube files along with standard image formats such as JPEG and PNG. This is useful to enable access to geographically regional datasets like the HiRISE patches made available from NASA [44].

In the case of iterative image projections, such as in the New Horizons scenario, the individual images can be loaded as they are produced and released by the science team. As these images are not georeferenced, real-time information about the spacecraft and its instruments are required in order to project these images, as described in Section 4.3.

4.2 Data Preparation

For the case of global image datasets, a variety of geographic projection formats are available. In order to support the layering of multiple data sources for a multimodal surface visualization, we chose to employ the most widely used format, the equirectangular geographic projection, as a single unified projection format for the globes in our system. If a global image dataset is not provided in equirectangular projection, we convert the dataset using GDAL.

The HiRISE patches from NASA are referenced in meters and not degrees as required by our equirectangular latitude-longitude rendering pipeline. Similarly, the exemplified time varying DEMs from the Center of Atmosphere Ocean Science are provided in EPSG:6326 datum which is used by the web mercator projection. These are examples of datasets that need to be reprojected. Through GDAL's notion of *virtual datasets*, local patches are treated as sparsely populated global maps. Virtual datasets are defined using XML specifications where properties such as data type, geographic extent, scale factors, offsets, and multiple raster bands can be specified for versatility.

To produce either local or global DTMs, some spacecraft use altimeters that are capable of measuring the height information directly. If they do not possess instrumentation that produces direct measurements of height information, DTMs can still often be reconstructed using stereoscopic pairs of images in stereophotogrammetry. The use of the NASA *Ames Stereo Pipeline (ASP)* stereogrammetry software [46] for content creation becomes a valuable tool in our pipeline for building DTMs from stereo pairs. ASP works in tandem with planetary mapping methods available through the *United States Geological Survey's (USGS), Integrated Software for Imagers and Spectrometers (ISIS)* [25]. CTX patches have successfully been generated from stereo pairs to match the MOLA base map of Mars [10, 42]. We exemplify this using a collection of CTX images covering West Candor Chasma in Valles Marineris on Mars. Furthermore, it has been shown to work on reconstructing high resolution DTM's using image pairs from the HiRISE camera resulting in a resolution of about 1 meter/pixel [36].

4.3 Real-Time Image Projection

In order to support the direct iterative visualization of spacecraft image acquisition, our system interfaces with various tools used in space mission planning and monitoring. The information these tools provide is used to reconstruct the location and orientation of a spacecraft, together with specifications about the instruments camera system, such as field-of-view and detector sizes. Utilizing this data, together with the timing at which a specific image was taken, we project images and store them in an equirectangular global image map of the entire globe. This map can then be used as a data source providing image data to the renderer.

The SPICE toolkit provided by NASA's *Navigation and Ancillary Information Facility (NAIF)* is used to load location and orientation data for planets, spacecraft, and instruments [2]. Image acquisition and exposure times are either directly provided by the mission scientists or available through NASA's *Planetary Data System (PDS)*, which curates information about all past and current space missions [45].

In order to map image data onto the equirectangular format, we combine position and navigation data from SPICE with image exposure timestamps from PDS. For individual instruments on the spacecraft, virtual camera frustums are constructed. For each image taken by an instrument, latitude-longitude values of the planetary surface are mapped to image coordinates. The images are projected to the globe using projective texturing as described by Everitt *et al.* [22]. Using this method, the globe's triaxial ellipsoid is rendered from the view point of the instrument. Then the image is applied to the projected globe with the correct geometric distortions. Thus the resulting images can be converted into an equirectangular global mosaic during the visualization of the acquisition.

4.4 Rendering System

The rendering system is primarily driven by the rendering of chunks, which are organized into a quad-tree. Each chunk covers a particular geographic area of the planetary surface. The processing pipeline is outlined in Figure 3. First, the chunk tree is processed to reflect the best fit for the given camera view. In this process the chunk tree is evaluated and chunks are either culled, split into child nodes, or merged. The resulting leaves are flagged for rendering, as detailed below in Section 4.4.1. Second, the system requests tile data from tile providers and will continue to render chunks with the currently available tiles, which we explain in Section 4.4.2. Finally, tiles and layers are combined in the fragment pipeline, presented in Section 4.4.3. Each chunk is rendered as a skirted regular grid mesh associated with a geodetic area and level of detail, see Figure 4. The grid size is constant per chunk and the height map layers will affect the altitude of the grid vertices.

Separating the representation used for the chunks (the geometry) from the individual tiles (the textures) enables dynamic loading as the geometry LOD can be updated independently from the image information and does not rely on the tile availability. This is important when multiple layers are combined as each may have its unique resolution. Figure 4 shows this exemplified on a single chunk that is rendered using three layers, each organized in a separate quad-tree.

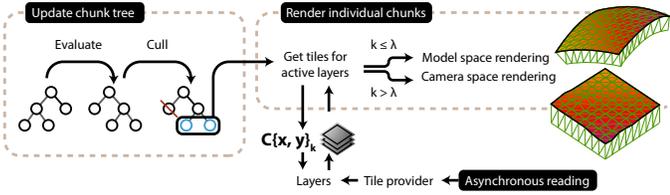


Fig. 3: Chunk tree processing and rendering pipeline overview. For each frame, the tree is evaluated and culled before rendering. Tiles are loaded asynchronously to avoid stalling the rendering pipeline.

4.4.1 Chunk Processing

Chunk nodes are organized in a quad-tree and defined as $C\{x, y\}_k$ for a geographic coordinate index (x, y) and detail level k . When $k = 1$, the globe is represented with two chunks, one for the Western and one for the Eastern hemisphere. Chunk tree evaluation involves selection of the desired LOD, limiting the level, and culling of chunks, as illustrated in Figure 3.

Chunk selection — Like Cozzi and Ring [15], we select chunks using a unique error metric. Our error metric however is calculated on the fly using the logarithm of the inverse distance from its closest point to the camera position or by an approximation of the chunk’s projected area on a sphere around the camera. The spherical projection is important for multi-display systems where there should be no difference in the error metric of a specific chunk rendered across neighboring screens.

Level limitation — In case a specific tile dataset does not contain any higher resolution tiles the maximum level of the chunk tree is limited. Since tile datasets can be sparse there is no way to know the highest detail level before the actual reading takes place. To avoid unnecessary splits of chunk nodes, they should not split unless there is higher resolution data available, that is, if a tile reading operation fails.

Chunk culling — To minimize the number of render calls, invisible chunks are culled. For horizon culling we use the maximum height of the chunk as bounding sphere. Chunks that are outside of the camera frustum are culled by testing a view space defined axis aligned bounding box of the chunk against the camera frustum. To perform correct culling, the height layer tiles do need some preprocessing before they can be used for rendering. More specifically, the minimum and maximum values of the tiles are used to specify a minimal bounding hexahedron for each chunk. These values are calculated out of core. The remaining chunk nodes are flagged as visible for rendering.

4.4.2 Tile Management

A chunk is affected by multiple tiles that have a texture representing a geodetic area. Tiles are uniquely defined using the notation $T_L\{x, y\}_k$, with the layer identifier L , the geographic tile index (x, y) and the detail level k . Tiles are organized in a quad-tree structure so that $T_L\{x, y\}_k$ covers the same area as the four child tiles $T_L\{2x, 2y\}_{k+1}$, $T_L\{2x, 2y+1\}_{k+1}$, $T_L\{2x+1, 2y\}_{k+1}$ and $T_L\{2x+1, 2y+1\}_{k+1}$. *Layers* provide different semantic meaning for groups of tiles. Such usage are for example color layers, color overlays, water masks, or height layers. The layers’ usage only impact the final rendering steps which means that all textures of the same resolution can be cached together and be reused for more recently requested tiles.

Each layer has a corresponding tile provider that the renderer will query using the tile index, representing the corresponding chunk to render. Tiles are requested for all active layers from the different layer groups. The ellipsoidal partitioning of the hierarchical tiles are aligned with the chunk partitioning, mapping the tile in a 1:1 relation except for a local offset and scaling. This enables texture sampling to be performed in the local space of each chunk, avoiding precision issues in texture coordinate calculation. Pixel padding, shown as borders around tiles in Figure 4, is also used to achieve seamless blending between tiles. As a tile provider receives requests for tiles from the rendering thread it will return a tile if present in the in-memory cache, otherwise an empty tile is provided and an asynchronous read request is put on a queue for the provider’s tile reading thread. The tile reader thread

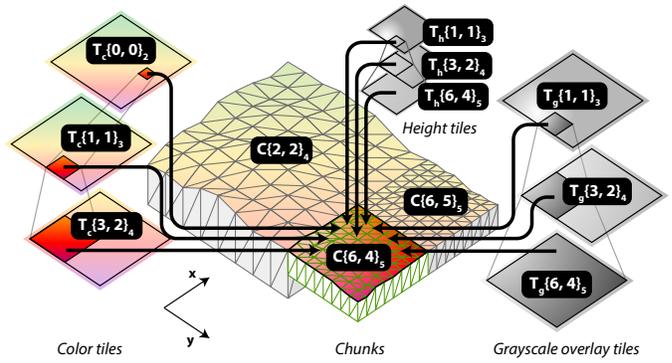


Fig. 4: Rendering chunk $C\{6, 4\}_5$ by combining the information from 9 tiles. For each of the three layer groups T_C (color), T_G (overlay), and T_H (height), the rendering of the main tile $T_C\{3, 2\}_3$, $T_G\{6, 4\}_4$, $T_H\{6, 4\}_4$ requires additional levels if level blending is enabled in order to provide interpolation between levels to avoid artifacts.

uses GDAL to read tile data from a source, GDAL is also caching data locally on disk and maintains an internal cache for recombining and remapping raw data tiles to cover the caller’s requested geographic area with pixels. Once the image data is ready, a tile is created and uploaded to GPU memory as a texture. The tile provider uses an in-memory *least recently used* (LRU) cache so that it can return a cached tile upon request. The LRU cache is shared by tile providers of the same layer group and keeps track of the total amount of tile data cached in system memory and on the GPU. When the cache is full, the tile texture of the least recently used tile will be re-used to create a new tile. Any other caching is performed outside of the rendering software.

4.4.3 Chunk Rendering

Given a valid chunk tree, each chunk that is flagged as visible is rendered individually. The geometry of the chunks is defined as a 2D regular skirted grid of constant size as seen in Figure 4. We avoid triangle decimation so that the same grid can be used for all chunks. This makes our chunk tree representation light weight and dynamic as vertex offsetting is performed in the vertex shader.

Each time a chunk needs to be rendered, all active layers will be queried for the most appropriate tiles. If the requested resolution level is not available, lower levels will be used. The renderer keeps the two immediately lower resolution levels available as well so that appropriate level blending can be applied, see Figure 4. We employ two types of rendering techniques for vertex positioning to account for both accuracy and precision.

Model-space rendering — Using the inverse projection $P^{-1}(\phi, \theta)$, vertices of a given chunk are unprojected from the georeferenced projection to the Cartesian model space of the globe. This operation is performed in single precision on the GPU and results in the chunk being accurately mapped to the curvature of the reference ellipsoid. Using this rendering method, the theoretical upper limit of precision will, for a Mars sized globe with radius $r = 3,390 \text{ km}$, be $p = \epsilon r \approx 0.2 \text{ m}$ where $\epsilon = 2^{-24}$ is the machine epsilon of a single precision IEEE floating point number [6]. Assuming a screen of width 1,920 pixels, the maximum horizontal length of a tile that covers half of the screen $w = 1,920/2$, and does not allow a one pixel error offset, is $l_{max} = pw \approx 194 \text{ m}$. Approximating the horizontal chunk length as $l = 2\pi r/2^k$, we get that the chunk level should not exceed $k_{max} = 16$ for model space rendering.

Camera-space rendering — To avoid having an upper bound on the chunk level so that the camera can move arbitrarily close to the surface without precision errors we use camera space rendering. The inverse projection $P^{-1}(\phi, \theta)$ can be performed in double precision on the CPU for the four corner points of the chunk which are then transformed to camera space in double precision and uploaded to the GPU in single precision. The rest of the vertices representing the chunk are then

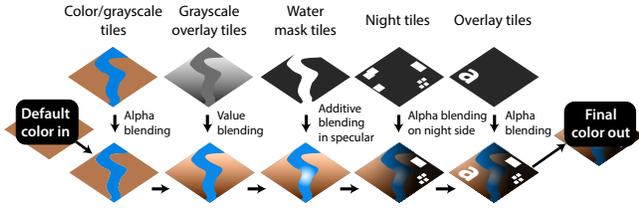


Fig. 5: Layer blending. Gray-scale overlays affect only the value component of the base color in HSV space. Water masks are used to change the specular reflection in the surface shading. Night tiles are useful for Earth where cities illuminate the side that is not facing the Sun. Geographical information can be added using overlay tiles.

obtained through bilinear interpolation. This method results in high precision rendering as the origin is practically moved to the center of the camera. The accuracy, however, is lower since the chunks are approximated as flat surfaces. We can calculate that the vertical error due the tiles being flat is $e_c = r - r \cos(2\pi/2^{k+1})$. With the constraint that the screen space error remains smaller than one pixel, i.e. $w_e/l < 1$, we obtain the minimum allowed chunk level for camera space rendering $k_{min} = 10$.

Combined model- and camera-space rendering — To account for both the accuracy, required for large curved chunks, and precision, required for smaller chunks close to the camera, we use a cutoff level, λ , to determine whether to use the model space or the camera space rendering as illustrated in Figure 3. Setting λ to a value between $k_{min} = 10$ and $k_{max} = 16$, satisfies both of the error criteria. These calculations are invariant of r which means that the same cutoff level can be used for globes of any size.

Fragment blending — To account for versatility and seamless level switching we perform three different types of fragment blending.

Layer group blending is performed to blend different layer groups by sampling and compositing each active layer group to get the final color, as shown in Figure 5. Examples are night layers which will be sampled on the night side of the globe, and water layers which are used to add a specular component to the final color.

Layer blending is performed so that many layers in the same layer group can be rendered together. Layers can be manually ordered and combined using per-layer settings such as opacity or blend mode. Furthermore, using alpha blending, local high-resolution patches can be rendered on top of other datasets with lower resolution.

Level blending is performed to smoothly transition between levels and avoid artifacts. It differs from Ulrich’s time-varying blending [58] in that we calculate an interpolation parameter for each fragment similar to Kooima *et al.* [35] depending on the camera distance that is used to interpolate between the tile of the current chunk level and the corresponding tiles of the two lower-level tiles, see Figure 4.

We use shader preprocessing and recompilation to opt out any unused blending option or layer as they are toggleable interactively.

4.4.4 Time Varying Datasets

Due to the abstraction of tile requests, we can define a layer type which can be updated in the rendering loop. For time varying datasets, a time stamp corresponding to the current simulation time is used to instantiate a tile provider for any given time step. We store a template defining the dataset without a specific time stamp. This template can be a file path or a GDAL WMS configuration that contains a time stamp key along with metadata defining time format, start time, end time, and temporal resolution. When issuing calls for tiles and the corresponding tile provider needs to be created, the time stamp key is replaced with a time stamp and a new dataset is opened.

As an example, a folder containing several image files can be used as a source for a temporal dataset as new files are opened once the simulation time reaches the next valid interval. If the dataset corresponding to the given time interval is already open, the temporal tile provider reads

the tile corresponding to the given tile index. Tiles from temporal tile layers do not differ from other tiles so they can be saved in the same cache and returned upon request.

Opening datasets and streaming tile data upon request can cause delay in tile loading if the time step of the simulation is significant. This can lead to a lot of reading within a short amount of time. Therefore, memory buffering becomes important for temporal datasets. Memory buffering can either be performed by pre-caching tiles when initializing the software or by browsing to the specific time intervals and let the tiles load as they are requested. Once in memory, the switching of tiles depending on time interval is not different from regular tile loading. Therefore, the layers can be animated by scrubbing the global simulation time forward and backward.

4.4.5 Atmosphere

Our application use an implementation of atmospheric effects as proposed by Elek [19]. The teaser image as well as Figure 2a and Figure 2b shows how it enables more realistic representations of Earth and Mars.

4.5 Multi Display Systems Rendering

The OpenSpace software uses a cluster rendering software package known as the *Simple Graphics Cluster Toolkit* (SGCT) [4] which supports image warping and blending for projection on dome displays and other clustered environments using several rendering machines (see Figure 6). Synchronization across rendering nodes is based on GenLocking at the graphics hardware level and SGCT furthermore provides support for synchronization of selected application data at frame rendering level, ensuring that all nodes in the cluster renders a given frame with the same camera and time state. Tile loading and chunk LOD selection is autonomous on each node, which visually can result in minor differences in the projector blending regions. However, due to the use of camera direction independent chunk selection algorithms, each rendering node will request the same tiles to be used across projector edges.

5 RESULTS

Here we present our results exemplified by the three application scenarios introduced in Section 3. We perform a globe browsing descent towards the surface of Mars measuring frame time and the size of the chunk tree. We also show how the Earth can be rendered with temporally varying textures of high resolution. Finally, we exemplify how our application is used to communicate the data acquisition process of the New Horizons flyby of the Pluto system.

5.1 Globe Browsing on Mars

In our test session, illustrated in Figure 7, Mars is rendered using the global Viking Color mosaic 2.1 as a base color layer. The global MOLA height layer is used as a base DTM. Furthermore, in order to achieve more detail, we include the Global CTX mosaic, as well as local patches that contain DTM and textures from the CTX and HiRISE camera.

The computer used for the measurements is an Intel Xeon E5-1620 3.60 GHz with 8 GB RAM and an NVIDIA GeForce GTX-780 GPU.



Fig. 6: Browsing the surface of Mars in a clustered environment rendering on the power wall at the University of Utah. The system uses eight rendering nodes, each connected to four screens.

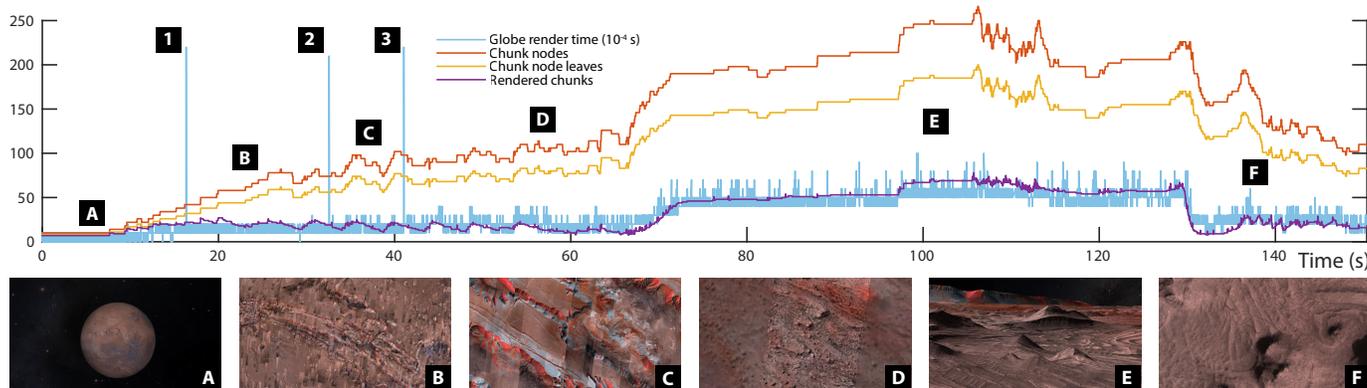


Fig. 7: Globe browsing on Mars: top graph shows size of the chunk tree along with the total render time. Atmosphere is not rendered. 1) Global CTX Mosaic enabled, 2) Local CTX patch enabled, 3) Local HiRISE patch enabled. A) Camera approaches Mars, B) Valles Marineris, C) West Candor Chasma, D) Layered rock outcrops in South West Candor Chasma, E) Camera tilted up, F) Camera rotated and then tilted down.

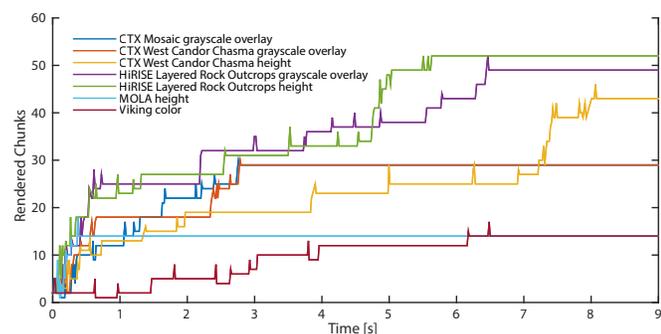


Fig. 8: Loading times for different data sources illustrated by the number of rendered chunks when enabling the layer as the camera is placed 50 meters from the surface facing the horizon.

Figure 7 shows how the frame time depends on the loading of data and the size of the chunk tree. When activating new layers, the frame time increases due to the need of shader recompilation. By descending to the surface, more of the chunk tree is traversed. However, the number of rendered chunks stays constant due to culling. Tilting the camera leads to new tile reading calls which in turn causes the chunk tree to grow over time as more tiles become available, see Figure 9. Following point E, the camera is rotated and thus decreasing the chunk tree due to unavailable tile data in areas previously invisible to the camera. As soon as the data is read and available, the chunk tree grows since it no longer is limited by the unavailable data. As soon as the camera is tilted down again, the number of rendered chunks decreases.

To further show how the different data products contribute to the chunk tree, Figure 8 illustrates the loading time for each of the Mars datasets with the camera positioned above the HiRISE patch. The final number of rendered chunks is directly dependent on the data source used due to the chunk level being limited by the availability of data and therefore differs between datasets. This scenario combines data sources stored on local WMS servers (CTX Mosaic and MOLA) and local patches stored on a spinning hard disk (HiRISE) and there is no significant difference between the loading times of the two.

The Mars globe browsing sessions have been used in a large number of public presentations to explain the history of Mars exploration, and the ability to contextualize events pertaining to the rovers on the surface. These public education events using the available Mars data have been ongoing for the past year and reached thousands of people using multi display systems and online streaming for public outreach. One such example of an event was a globe browsing session in the Hayden Planetarium seen by 400 people on site and live streamed on YouTube on June 5th 2017 entitled "Mars as never seen before".

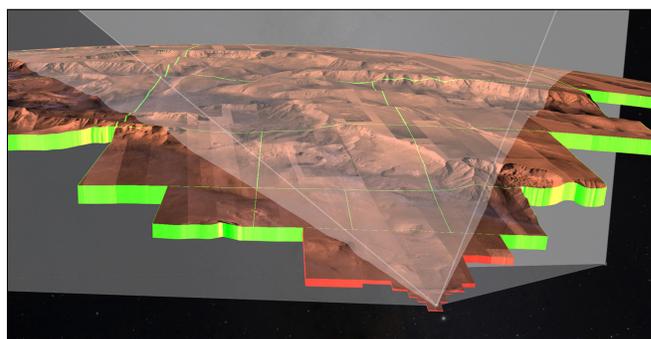


Fig. 9: Illustration of camera frustum and horizon culling. Model space rendering of chunks (green) is performed at higher levels, while camera space rendering (red) is performed at lower levels.

5.2 Temporal Datasets on Earth

In addition to ongoing public events related to the exploration of high resolution surface imagery on Mars, other public live shows have been held using our system to show dynamic processes on our own planet. On April 22nd 2017, coinciding with Earth day, an event connecting five locations around the world was held to discuss the impact of global warming on Earth with live audiences. This event consisted of multiple global NOAA datasets that show, among other parameters, precipitation, temperature, and the aerosol content of the atmosphere.

Moreover, to show how the Sun is directly affecting our planet, coinciding with Sun-Earth day, an event was held on June 27th 2017 at the Hayden Planetarium. By showing volumetric datasets from NASA's *Community Coordinated Modeling Center (CCMC)*, simulating solar activity visualized with time-varying field lines and ray casted volumes,

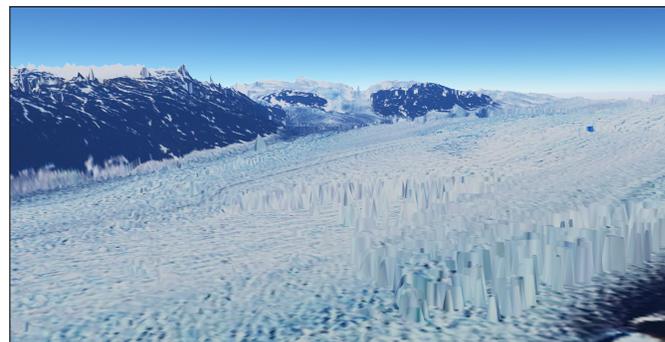


Fig. 10: Rendering of a single timestep of the Helheim glacier's movements in Greenland as acquired by radar measurements. With a cadence of 15 minutes, it becomes possible to observe the calving of the glacier.

in the same context as measured data in the form of global equirectangular time-varying datasets close to Earth, our software was used to visualize how the solar flares of 2012 related to the number of charged particles that impacted our planet.

By rendering time-varying DEMs in 3D, domain scientists working on generating and maintaining such datasets are able to contextualize their work. Further research into time-varying DEMs is necessary but we show already in Figure 10 the calving of the Helheim glacier ridge.

By accessing other imagery repositories such as the ones mentioned in Section 3.2 we are able to visualize hundreds of different scientific parameters animated over time. The list includes, but is not limited to, surface reflectance, cloud pressure, acids and oxides, temperatures, snow and ice coverage, soil and vegetation indices, chlorophyll levels, and more. Together with reference layers such as labels, border lines, coast lines, and roads these layers can easier be contextualized. Figure 2b shows corrected reflectance and maximum air temperature, both with a temporal resolution of one day.

By utilizing this data driven approach to animated planetary visualization we can show that our planet is far from static. In fact, it can act as a tool in educating the general public about the impact of human society on everyone's planet.

5.3 Visualizing Acquisition on Pluto

NASA's New Horizons mission flew by the Pluto system on July 14th, 2015 and took measurements with its seven instruments. Of special interest are the LORRI and *Ralph* instruments, that provided images of Pluto and its moons' surfaces, as well as REX, measuring Pluto's atmosphere. The user can follow the progression of the spacecraft over time and see the instrument activities. In addition to visualization of the instrument frustums and projected images, the measurement times for all instruments are presented to the user.

This mission visualization was presented to about 2,500 people during a public, global event with 13 different locations. During a 2h live show, which coincided with New Horizons closest approach to Pluto, experts on the mission team explained details of the desired outcome using OpenSpace as the source of contextualization for this information, see Figure 11. In addition to the live audience in the participating locations, a video stream of the event was made available. This was also later provided as a video-on-demand, called "Breakfast at Pluto". The event was described in the poster on public dissemination of space mission profiles by Bock *et al.* [9]. Some initial results from the New Horizons mission are given by Stern *et al.* [57]. The visualizations have also been used in several post flyby events showing the incremental update of our knowledge about this object to the general public.

Space missions, such as *New Horizons* or *Cassini*, are very expensive endeavors carried out in the public interest. These missions are planned years in advance in order to maximize the scientific output they produce. Due to this streamlining and the lack of available tools for the general

public, the spacecraft's actions during the scientific exploration can be difficult to understand and communicate. The New Horizons mission is an example of how our system allows mission scientists to explain their findings in context, and indeed in current time, and also allows them to preview different mission plans, while providing interesting content for the public engagement in science and technology.

6 CONCLUSION AND FUTURE WORK

In this paper we have presented an application which enables data-driven visual browsing of planetary surfaces with dynamic resolution in space and time. The application makes it possible to generate global views as well as virtual terrain fly-overs. One of the novel aspects of our work is that it contextualizes the surface data in a seamless astrovisualization environment making it possible to travel from one globe to another, or even visit spacecraft at past, current, or future locations. The platform used is the OpenSpace [8] software which builds on a dynamic scene graph and the capability to accurately position and visualize celestial bodies and spacecraft. Another key component presented is the capability to visualize the data acquisition process from the spacecraft perspective, letting the user appreciate the engineering efforts behind complex space exploration missions whilst appreciating the origin of the data which drives scientific discovery. With science communication as a primary target the software has already proven to create engaging immersive experiences in large scale dome theaters. We have developed a tool useful for scientists, who want to present their work in a contextualized manner, and for the general public who wish to explore accurate and open data representation of celestial bodies.

As future work we will further explore the seamless addition of detailed data from surface-based missions such as the Mars rovers. We will also add more data sources such as simulated data of abstract planetary properties. We also aim to add GIS tools to better aid scientists whereas the current main focus of the application is for public presentations. Intuitive and intelligent context dependent navigation models for multi-scale data is also an area with interesting challenges.

ACKNOWLEDGMENTS

We would like to acknowledge the Knut and Alice Wallenberg Foundation, the Swedish e-Science Research Center (SeRC), and ELLIIT for their support of this work. Parts of this work were supported by Vetenskapsrådet under award VR-2015-05462, NASA under award No NNX16AB93A, the Moore-Sloan Data Science Environment at New York University, and NSF awards CNS-1229185, CCF-1533564, CNS-1544753, and CNS-1730396. The authors would like to thank Ryan Boller, Lucian Plesea, and the NASA GIBS team for their help, and Jonathas Costa for the implementation of atmospheric effects. We would also like to thank the anonymous reviewers who provided valuable comments to improve the final quality of the paper.

REFERENCES

- [1] B. Abbott. The digital universe guide for partiview, 2006.
- [2] C. H. Acton. Ancillary data services of NASA's navigation and ancillary information facility. *Planetary and Space Science*, 44(1):65–70, 1996.
- [3] M. Ali, B. Chandramouli, J. Fay, C. Wong, S. Drucker, and B. S. Raman. Online visualization of geospatial stream data using the worldwide telescope. *Proceedings of the VLDB Endowment*, 4(12), 2011.
- [4] M. Andel, E. Sundén, and A. Bock. SGCT - simple graphics cluster toolkit, 2017. <https://c-student.itn.liu.se/wiki/develop:sgct:sgct>.
- [5] B. Archinal, R. Kirk, T. Duxbury, E. Lee, R. Sucharski, and D. Cook. Mars digital image model 2.1 control network. In *Lunar and Planetary Science Conference*, vol. 34, 2003.
- [6] E. Axelsson, J. Costa, C. Silva, C. Emmart, A. Bock, and A. Ynnerman. Dynamic scene graph: Enabling scaling, positioning, and navigation in the universe. In *EuroVis*. Eurographics Association, 2017.
- [7] G. B. Berriman and J. Good. The application of the montage image mosaic engine to the visualization of astronomical images. *Publications of the Astronomical Society of the Pacific*, 129(975):058006, 2017.
- [8] A. Bock, E. Axelsson, K. Bladin, J. Costa, G. Payne, M. Territo, J. Kilby, M. Kuznetsova, C. Emmart, and A. Ynnerman. OpenSpace: An open-source astrovisualization framework. *The Journal of Open Source Software*, 2(15), 2017.



Fig. 11: OpenSpace was used to visualize how New Horizons captured images of Pluto on July 14th, 2015. Sites from all over the world connected in a live event during the flyby. The image shows the New York site with host Neil deGrasse Tyson.

- [9] A. Bock, M. Marcinkowski, J. Kilby, C. Emmart, and A. Ynnerman. OpenSpace: Public dissemination of space mission profiles. *2015 IEEE Scientific Visualization Conference (SciVis)*, Oct 2015.
- [10] M. J. Broxton and L. J. Edwards. The ames stereo pipeline: Automated 3D surface reconstruction from orbital imagery. In *Lunar and planetary science conference*, vol. 39, p. 2419, 2008.
- [11] M. Cechini, K. Murphy, R. Boller, et al. Expanding access and usage of NASA near real-time imagery and data. In *AGU Fall Meeting Abstracts*, vol. 1, p. 04, 2013.
- [12] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Planet-sized batched dynamic adaptive meshes (P-BDAM). In *Proceedings of the 14th IEEE Visualization*, p. 20, 2003.
- [13] M. Clasen and H.-C. Hege. Terrain rendering using spherical clipmaps. In *Proceedings of the Eighth Joint Eurographics/IEEE VGTC conference on Visualization*, pp. 91–98. Eurographics Association, 2006.
- [14] P. Cozzi. Cesium: 3D maps on the web. FOSS4G NA, 05/2013, 2013.
- [15] P. Cozzi and K. Ring. *3D engine design for virtual globes*. CRC Press, 2011.
- [16] P. J. Cozzi. *Visibility driven out-of-core HLOD rendering*. PhD thesis, University of Pennsylvania, 2008.
- [17] M. Duchaineau, M. Wolinsky, D. E. Sigesti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein. Roaming terrain: real-time optimally adapting meshes. In *Proceedings of the 8th Conference on Visualization '97*, pp. 81–88. IEEE Computer Society Press, 1997.
- [18] C. Eames and R. Eames. *Powers of Ten*. IBM, 1977.
- [19] O. Elek. Rendering parametrizable planetary atmospheres with multiple scattering in real-time. In *Proceedings of the Central European Seminar on Computer Graphics*. Citeseer, 2009.
- [20] C. Erikson, D. Manocha, and W. V. Baxter. HLODs for faster display of large static and dynamic environments. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pp. 111–120. ACM, New York, NY, USA, 2001. doi: 10.1145/364338.364376
- [21] Evans & Sutherland. Digistar. [online] <http://www.es.com/Digistar/>.
- [22] C. Everitt, A. Rege, and C. Cebeonoyan. Hardware shadow mapping. *White paper, nVIDIA*, 2, 2001.
- [23] C.-W. Fu and A. J. Hanson. A transparently scalable visualization architecture for exploring the universe. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):108–121, 2007.
- [24] C.-W. Fu, A. J. Hanson, and E. A. Wernert. Navigation techniques for large-scale astronomical exploration. In *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006.
- [25] L. Gaddis, J. Anderson, K. Becker, T. Becker, D. Cook, and K. Edwards. An overview of the integrated software for imaging spectrometers (ISIS). In *Lunar and Planetary Science Conference*, vol. 28, p. 387, 1997.
- [26] A. Goodman, J. Fay, A. Muench, A. Pepe, P. Udomprasert, and C. Wong. Worldwide telescope in research and education. *arXiv preprint arXiv:1201.1285*, 2012.
- [27] A. J. Hanson, C.-W. Fu, and E. A. Wernert. Very large scale visualization methods for astrophysical data. In *Data Visualization*, pp. 115–124. Springer, 2000.
- [28] P. Hogan and F. Kuehnel. NASA world wind, open source 4D geospatial visualization platform: *.NET & Java* for education. In *AGU Fall Meeting Abstracts*, 2006.
- [29] K. J. Hussey. “Eyes On The Solar System” a real-time, 3D-interactive experience for planetaria and beyond. In *European Planetary Science Congress 2010*, vol. 1, p. 50, 2010.
- [30] L. M. Hwa, M. A. Duchaineau, and K. I. Joy. Real-time optimal adaptation for planetary geometry and texture: 4-8 tile hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):355–368, 2005.
- [31] C. Justice, J. Townshend, E. Vermote, E. Masuoka, R. Wolfe, N. Saleous, D. Roy, and J. Morisette. An overview of MODIS land data processing and product status. *Remote sensing of Environment*, 83(1):3–15, 2002.
- [32] B. R. Kent. Visualizing astronomical data with Blender. *Publications of the Astronomical Society of the Pacific*, 125(928):731, 2013.
- [33] S. Klashed, P. Hemingsson, C. Emmart, M. Cooper, and A. Ynnerman. Uniview - Visualizing the Universe. In *Eurographics - Areas Papers*. Eurographics Association, 2010.
- [34] R. Koch. 3-D surface reconstruction from stereoscopic image sequences. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 109–114, Jun 1995. doi: 10.1109/ICCV.1995.466799
- [35] R. Kooima, J. Leigh, A. Johnson, D. Roberts, M. SubbaRao, and T. A. DeFanti. Planetary-scale terrain composition. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):719–733, 2009.
- [36] R. Li, J. Hwangbo, Y. Chen, and K. Di. Rigorous photogrammetric processing of HiRISE stereo imagery for Mars topographic mapping. *IEEE Transactions on Geoscience and Remote Sensing*, 49(7):2558–2572, 2011.
- [37] C. T. Liu, B. Abbott, C. Emmart, et al. 3-D visualizations of massive astronomy datasets with a digital dome. In *Virtual Observatories of the Future*, vol. 225, p. 188, 2001.
- [38] F. Losasso and H. Hoppe. Geometry clipmaps: terrain rendering using nested regular grids. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 769–776. ACM, 2004.
- [39] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.
- [40] M. A. Magnor, P. Sen, J. Kniss, E. Angel, and S. Wenger. Progress in rendering and modeling for digital planetariums. In *Eurographics (Areas Papers)*, 2010.
- [41] J. Maso, K. Pomakis, and N. Julia. OpenGIS web map tile service implementation standard. *Open Geospatial Consortium Inc*, pp. 04–06, 2010.
- [42] D. P. Mayer and E. S. Kite. An integrated workflow for producing digital terrain models of Mars from CTX and HiRISE stereo data using the NASA Ames Stereo Pipeline. In *Lunar and Planetary Science Conference*, vol. 47, 2016.
- [43] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, et al. Mars reconnaissance orbiter’s high resolution imaging science experiment (HiRISE). *Journal of Geophysical Research: Planets*, 112(E5), 2007.
- [44] A. S. McEwen, R. Heyd, S. Sutton, et al. For the people: HiRISE data products. In *Lunar and Planetary Science Conference*, vol. 47, 2016.
- [45] S. K. McMahon. Overview of the planetary data system. *Planetary and Space Science*, 44(1):3–12, 1996.
- [46] Z. M. Moratto, M. J. Broxton, R. A. Beyer, M. Lundy, and K. Husmann. Ames Stereo Pipeline, NASA’s open source automated stereogrammetry software. In *Lunar and Planetary Science Conference*, vol. 41, 2010.
- [47] J. P. Naiman. Astrobend: An astrophysical visualization package for blender. *Astronomy and Computing*, 15:50–60, 2016.
- [48] A. Nakasone, H. Prendinger, S. Holland, P. Hut, J. Makino, and K. Miura. Astrosim: collaborative visualization of an astrophysics simulation in second life. *IEEE Computer Graphics and Applications*, 29(5), 2009.
- [49] NASA Jet Propulsion Lab. Mars Trek. <https://marstrek.jpl.nasa.gov/>.
- [50] Open Geospatial Consortium. OpenGIS web map server implementation specification. 2006.
- [51] T. C. Patterson. Google Earth as a (not just) geography education tool. *Journal of Geography*, 106(4):145–152, 2007.
- [52] V. V. Salomonson, W. L. Barnes, P. W. Maymon, H. E. Montgomery, and H. Ostrow. MODIS: Advanced facility instrument for studies of the Earth as a system. *IEEE Transactions on Geoscience and Remote Sensing*, 27(2):145–153, 1989.
- [53] C. F. Schuele, J. E. Clement, P. E. Ardanuy, C. Welsch, F. DeLuccia, and H. Swenson. NPOESS VIIRS sensor design overview. In *International symposium on optical science and technology*, pp. 11–23. International Society for Optics and Photonics, 2002.
- [54] Sky-Skan Inc. Digitalsky 2, 2016. <http://www.skyskan.com/products/ds>.
- [55] D. E. Smith, M. T. Zuber, H. V. Frey, et al. Mars orbiter laser altimeter: Experiment summary after the first year of global mapping of Mars. *Journal of Geophysical Research: Planets*, 106(E10):23689–23722, 2001.
- [56] J. P. Snyder. *Flattening the Earth: two thousand years of map projections*. University of Chicago Press, 1997.
- [57] S. A. Stern, F. Bagenal, K. Ennico, et al. The Pluto system: Initial results from its exploration by New Horizons. *Science*, 350(6258), 2015.
- [58] T. Ulrich. Rendering massive terrains using chunked level of detail control. *SIGGRAPH Course Notes*, 3(5), 2002.
- [59] F. Warmerdam. The geospatial data abstraction library. In *Open source approaches in spatial data handling*, pp. 87–104. Springer, 2008.
- [60] G. Webster. Prolific Mars orbiter completes 50,000 orbits, June 27 2017. [Online; posted March 29, 2017] <https://www.nasa.gov/feature/jpl/prolific-mars-orbiter-completes-50000-orbits>.
- [61] P. R. Wolf and B. A. Dewitt. *Elements of photogrammetry: with applications in GIS*, vol. 3, 2000.
- [62] A. Ynnerman, T. Rydell, D. Antoine, D. Hughes, A. Persson, and P. Ljung. Interactive visualization of 3D scanned mummies at public venues. *Communications of the ACM*, 59(12):72–81, 2016.
- [63] R. W. Zurek and S. E. Smrekar. An overview of the Mars Reconnaissance Orbiter (MRO) science mission. *Journal of Geophysical Research: Planets*, 112(E5), 2007.