VOIDGA:  
A View-Approximation Oriented Image Database Generation Approach

Jonas Lukasczyk*  
TU Kaiserslautern

Eric Kinner†  
TU Kaiserslautern

James Ahrens‡  
Los Alamos National Laboratory

Heike Leitte§  
TU Kaiserslautern

Christoph Garth¶  
TU Kaiserslautern

ABSTRACT

In this work, we propose a novel view-approximation oriented image database generation approach (VOIDGA) that enables the adequate generation of arbitrary views. Our approach utilizes Depth Image Based Rendering (DIBR) techniques to derive novel views based on a set of depth images. In contrast to approaches that store a huge amount of images to cover a wide range of possible view directions, VOIDGA identifies and stores only those images that significantly contribute to the overall view-approximation quality while bounding the resulting approximation error. This further reduces the size of image databases and the number of images that need to be processed by DIBR algorithms. We demonstrate VOIDGA on several challenging real-world examples, and compare our approximations against ground truth renderings using two image-based metrics.

Keywords:  Image Database, Depth Image Based Rendering, Geometry Reconstruction, View Approximation

1 INTRODUCTION

The increasing size and complexity of datasets make it necessary to reduce the amount of stored information while still supporting effective data exploration through interactive visual interfaces. Especially in the case of extreme scale simulations, it is often impossible to interactively render—or even store—an acceptable set of simulation states for post hoc analysis due to bandwidth and I/O constraints. To address this issue, Ahrens et al. [1] proposed the ParaView Cinema concept as an image-based approach for the post hoc exploration of simulation output. In their approach, an image database is created in situ consisting of color and depth images of simulation elements taken from various camera positions. Such databases are several orders of magnitude smaller than the simulation data they are derived from, and they enable the real-time exploration of extreme scale simulations by querying and compositing images from the database. Rudimentary image database viewers facilitate basic camera movement by simply snapping to the closest available camera position [2, 23, 32]. As a principled limitation, these viewers cannot visualize viewpoints that had not been foreseen and specified during database generation. However, depth image based rendering (DIBR) algorithms enable the approximation of novel views based on existing database elements, which supports unconstrained camera interaction for visual exploration (Fig. 1). Such algorithms, however, introduce approximation errors that depend on the quality of the used DIBR technique and the input depth images. It is also not clear which and how many images are needed to adequately approximate a wide range of novel views.

In this paper, we address these issues by taking the first steps towards leveraging the information stored in an image database to its full potential. Specifically, we propose a novel view-approximation oriented image database generation approach (VOIDGA) that determines a minimal set of input depth images that enable the approximation of new views within a certain error bound. The core concept of VOIDGA is to identify and store images that significantly contribute to the overall view-approximation quality, while at the same time discarding images that can already be adequately approximated. This yields much smaller image databases than the ones produced by current state-of-the-art implementations which uniformly sample
images on a spherical grid (Fig. 2). This also results in a reduced set of images that need to be processed by DIBR algorithms while still guaranteeing a minimum approximation quality.

We demonstrate the effectiveness of VOIDGA on several real-world examples of varying complexity, including strongly jagged surfaces and line geometry which represent worst-case scenarios for depth image based geometry approximation algorithms. To assess the quality of the resulting approximations, we employ two image-based metrics to measure the perceived and actual shape distortion between ground truth images and the approximations.

Furthermore, our approach can act as an optional addition to existing image databases, and can be employed independently from other database dimensions, such as time or parameter values. To summarize, the contributions of this work are:

- A survey of literature on IBR methods suited for scientific image database element approximation.
- A detailed description of a DIBR implementation for the approximation of arbitrary views for Cinema databases.
- A novel view-approximation oriented image database generation approach (VOIDGA) that derives a minimal set of images to approximate arbitrary views within a certain error bound.
- The demonstration that this approach allows fluid camera interaction with acceptable errors, which are examined qualitatively and quantitatively using two image-based metrics over several databases created from real-world use cases.

2 RELATED WORK

Image Based Rendering (IBR) methods and their integrated geometry approximation algorithms have been extensively studied in the context of remote rendering [3, 6, 17], image-based meshing [7, 9, 27, 31, 43, 44], 3D video processing [26, 38], and many more. In remote rendering, they significantly reduce server and bandwidth load by enabling clients to extrapolate new views based on already transmitted images without additional requests to the server [6]. As soon as the camera from the client diverges too much, the server generates and sends new depth images to the client. This is especially useful if the visualizations require computational or data intensive procedures. In 3D video processing, they allow to post hoc create stereoscopic images based on video-plus-depth footage [38]. They are also used to mesh objects based on multiple photographs, which enables photorealistic texture mapping [34, 39], the digital archiving of cultural heritage [44], and the complete reconstruction of indoor as well as outdoor environments [27, 43].

Shum and Kang [35] point out that all these methods require either implicit [4, 5, 9, 21, 42, 44], explicit [6, 27, 30, 31], or no [18, 25] geometry information to create novel views based on feature registration, geometry approximation, or plenoptic functions, respectively. Implicit geometry approximation algorithms interpolate between images by detecting and tracking features—such as the optical flow [4, 21, 42] or SIFT [9]—which creates visually appealing transitions between different views. However, the interpolated images do not necessarily have to coincide with reality, and often exhibit ghosting and warping artifacts [37]. IBR algorithms that use no geometry information interpret large, dense sets of images as two-dimensional slices of the four-dimensional light field function [18, 25]. All images are used to approximate the light field which is subsequently sampled to generate novel views. This produces high quality results as long as the light field approximation is good enough, but this requires a huge amount of images (1k+) and even compressed representations [18] do not scale for non-static scenes.

On the other side of the IBR spectrum are algorithms that explicitly derive the implied geometry of depicted objects based on depth images. These images can be obtained from sensors [27, 43], estimators [10, 19, 20, 22], or directly from the rendering pipeline [1]. As it is straightforward to generate Cinema databases that contain precise depth images of 3D rendered objects—such as iso-surfaces, streamlines, and particle trails—we focus in this paper on Depth Image Based-Rendering (DIBR) techniques. This does not mean, however, that the other approaches do not have merit or are inapplicable for Cinema databases. In the following, we present an overview across the development of DIBR techniques, which is also the basis of our exemplary implementation. Like other DIBR methods, our implementation is based on the fact that each pixel of the depth image corresponds to a 3D point on the depicted surface (Fig. 3a and b). These points can be computed by inverting the projection that was used to generate the depth image [6, 27, 30, 31], which yields a set of independent points in 3D space. A simple way to render the resulting locations is to represent them as a point cloud, called splatting [28, 31, 36, 46] (Fig. 3b). However, this creates gaps between points; especially when the depth image has a low resolution. The gaps can be filled by increasing the point size (which leads to a strong divergence from the original surface), or by increasing the point number (which requires high-res depth images). Another way to solve this problem is to continuously fill these gaps. For this it is necessary to link neighboring points of the depth image by creating a surface patch between them, i.e., derive a triangulation based on the depth image. As a first step, one can create two triangles between four neighboring pixels to create a piecewise linear approximation of the surface between the points (Fig. 3c). This fills all gaps, but also creates surface patches between pixels with very different depth values (Fig. 3c). This is known in the DIBR literature as the depth discontinuity [26, 38, 45]. A trivial solution to this problem is to use a distance threshold to discard distorted triangles (Fig. 3d). Unfortunately, there exists no threshold value that will always produce the best results as this value strongly depends on the smoothness of the depicted object. Moreover, removing such triangles creates gaps again that must be either filled by a variant of splatting [26], or by incorporating the implied geometry from multiple depth images [5, 6, 27] (Fig. 4).

Our algorithm follows the previously described outline, but is optimized for the view approximation of Cinema databases as these pose additional challenges. Most prominently, it is not possible to derive new image artifacts once the simulation has been built since this would require rerunning the simulation. Therefore, it is necessary to determine in situ various database parameters (e.g., the image resolution and sampling rate) that will later enable the approximation of views within a certain error threshold. To assess the quality of the approximated views, we use two image metrics: the Depth Difference (DD) [15], and the Multi-Scale Structural Similarity Metric (MS-SSIM) [41]. An advantage of these image-based metrics is that they fit well in the context of Cinema databases, and that they are independent of the actual data representation. Euclidean-based mesh distortion metrics—including the Hausdorff Distance,
To avoid recomputing the 3D world positions of all depth image pixels each time the camera is changed, we cache the positions of each depth image $D_i$ in a new position texture $P_i$. In contrast to the depth image that stores at each pixel the depth value, the position texture stores at each pixel the 3D world position of its corresponding depth image pixel. If the depth image was generated by an orthographic camera then the position of each depth image pixel can be calculated by

$$p(u,v,d) = c_p + u \cdot c_w \cdot (c_d - c_u) + v \cdot c_h \cdot c_u + d \cdot c_d$$

where $c_p$, $c_w$, and $c_h$ are the camera position, width, and height in world coordinates; $c_u$ and $c_d$ are the normalized camera up and direction vector; $u, v \in [-0.5, 0.5]$ are the coordinates of the pixel in image space; and $d$ is the depth value of the pixel in world coordinates. We use an orthographic over a perspective camera model to avoid projective distortions [15]. Since each pixel can be processed individually, we compute all positions of $D_i$ in one pixel shader pass that stores the locations in a new framebuffer $P_i$. In our experiments, this process takes roughly 0.02s for each $1024^2$ depth image while not utilizing parallelism.

3.1.3 Stage 3: Forward Warps

In this stage, we reproject the locations stored in the texture cache $P$ using the current camera setting. First, we clear color and depth channels of the framebuffer $F$, and then process each texture in the cache individually to incorporate its depicted surface into the overall view approximation.

Figure 3: Illustration of the forward mapping of a single 40x40 depth image (a) for the viscous finger dataset. The resulting points can either be directly visualized by splatting (b), or by approximating the surface between the points (c) and (d)). Splatting (b) creates gaps that need to be filled either by increasing the point number (i.e., image resolution) or the point size. The surface approximation creates piecewise-linear patches between vertices (c), but a distance threshold is needed to discard distorted triangles (d).

Figure 4: Composited approximations of the viscous finger dataset using four depth images with a resolution of either $40^2$ (left) or $1024^2$ (right). Colors encode the depth image that generates its corresponding surface patch. Depth images that depict the same part of a surface generate similar patches which causes z-fighting.
we render the resulting color for the entire point (Fig. 8a). For the
we remove more and more distorted triangles and thus create less
Pi vertex looks up its corresponding 3D location in
not compute any lighting in this stage, although it would be possible.
angle vertices (Fig. 8c). Note, in our current implementation we do
triangulation, we linearly interpolate the scalar values between tri-
data values are then rendered accord-
If an image database only consists of depth images then we sim-
• perform forward warp of P_i
• render diffuse and depth to F
Next, for each texture P_i, we perform an optimistic forward map-
all others are discarded. Thus, we use the new camera settings to
which depth image contributes which part of the resulting surface
we can take advantage of hardware acceleration. Fig. 4 illustrates
• position mesh for stage 2
• SSAO
• position mesh
• render diffuse and depth to F
Figure 5: Outline of our DIBR pipeline. First, we initialize the data-
structures that are used in later stages. We then build a texture
that stores the 3D world positions of the depth image pixels.
Upon camera rotation, we process the cache and reproject positions
using the new camera settings to write color and depth outputs in an
intermediate framebuffer. Finally, we post-process the framebuffer
using SSAO and FXAA.

Stage 1
(Initialization)
initialize:
• FBO cache P for stage 1
• position mesh for stage 2
• FBO F for stage 3
Stage 2
(Caching)
- clear FBO cache P
- for each depth image D_i:
  • create framebuffer P_i
  • store world positions of D_i in P_i
  • append P_i to P
Stage 3
(Forward Warps)
- clear color and depth buffer of F
- for each framebuffer P_i ∈ P:
  • perform forward warp of P_i
  • using position mesh
  • render diffuse and depth to F
Stage 4
(Post Processing)
render F to screen using:
• SSAO
• FXAA

3.1.4 Stage 4: Post Processing
As we do not compute any lighting at stage 3, we use a customized Screen Space Ambient Occlusion (SSAO) shader to compute the
global and local lighting. The global lighting greatly enhances
spatial perception by darkening regions that are next to areas which
are closer to the camera. For the local lighting, the SSAO shader
approximates the normal at a fragment and subsequently computes
its interaction with a point light source. The approximated normals
are also used to create a silhouette effect by emphasizing hard edges.
Furthermore, it is possible to adjust SSAO parameters such as the
radius of the depth sampling as well as the emphasis of shadows
and edges. The initial settings of these parameters can be used for
any scene, but users can perform a fine tuning to highlight specific
aspects of a model. Finally, the Fast Approximate Anti-Aliasing
(FXAA) shader smoothes jagged edges.

3.2 Quality Metrics
To assess the significance of individual depth images during the use
of VOIDGA (Sect. 3.3), and to validate our results (Sect. 4), we
compare the approximated views to ground truth renderings using the
Multi-Scale Structural Similarity Metric (MS-SSIM) [41] that
estimates the perceived image similarity, and the Average Depth
Difference (DD) [15] that measures the actual shape distortion.
The DD tries to capture the volumetric difference between two
objects and is computed in two stages. First, it computes the Average
Depth Difference (ADD) between two depth images by computing
the actual depth value difference per pixel. Hence, it is assumed that
the depth images have the same size, and that their values are
in the range [0, 1]. Similar to Cinema databases, this is done for
multiple camera angles that are positioned on vertices of a grid that
encapsulates the dataset and aim towards the object center. The DD
is then given as the average of all computed ADDs.
The MS-SSIM is modeled after the assumption that the human
visual system is highly adapted for extracting structural information
from 3D projections. Thus, a measure of the structural similarity
between images can provide a good estimate of the perceived image
quality [40,41]. In contrast to the original SSIM, the MS-SSIM itera-
tively downsamples the input images to determine the luminance and
contrast variations for varying resolutions. This allows to evaluate
the structural similarity between images more independently from
the actual image sizes. Similar to the ADD metric, we compute the
MS-SSIM for multiple camera positions and build the average to
evaluate the structural similarity across the entire approximation.
However, the ADD computes an error value from 0 (identical) to 1
(complete opposite), whereas the MS-SSIM computes a score from
0 (not similar) to 1 (identical).
3.3 VOIDGA

We propose a novel view-approximation oriented image database generation approach (VOIDGA): a greedy algorithm that iteratively refines a sampling grid and then only stores potential database elements (also called artifacts) that significantly contribute to the overall approximation quality. To this end, VOIDGA consists of three phases: the database backbone generation, the database refinement, and the database downsampling. To run VOIDGA completely automatically, users have to specify the maximum number of database artifacts, the initial (and thus maximum) artifact resolution, and the initial error tolerances. In the following, we demonstrate VOIDGA using our DIBR implementation, and the ADD and MS-SSIM metrics, but it is possible to use VOIDGA with any method or metric.

3.3.1 Backbone Generation

First, we normalize the dataset geometry according to the dimensions of the unit-cube with center at the origin, and then select a sampling grid structure. A common way to generate Cinema databases is to uniformly sample along a latitude-longitude parameterized sphere that encapsulates the dataset, where the cameras are positioned at the grid vertices and aim towards the center (Fig. 2). For image database viewers that simply snap to the next available artifact, this creates intuitive transitions as it seems like the camera rotates along the lat-lon axes. However, this grid causes an oversampling at the poles, and an undersampling at the equator (Fig. 2 left). Since our DIBR method is not restricted to the actual artifact locations, we propose to use an icosahedron as a sampling primitive. In contrast to a lat-lon grid, each icosahedron refinement uniformly creates new sampling positions that equally cover possible view angles (Fig. 2 right). As these positions are eventually added to the database, we effectively improve the approximation quality in each step.

To generate the database backbone (a small set of database artifacts that are the basis for the view approximation), we sample images on the 12 vertices of the unrefined icosahedron (intersection of red lines in Fig. 2, right). For each vertex of the grid, we generate a depth image with the initial resolution using an orthographic camera where the width and height are set to the icosahedron diameter. Thus, each artifact encapsulates the complete dataset, and the 12 locations already provide a good view angle coverage. It is also possible to add model-specific view angles—such as interior locations—to the backbone if such important angles are known. The next phase uses the resulting artifacts as a basis for the view approximation.

3.3.2 Database Refinement

In this phase, VOIDGA iteratively refines the sampling grid and only adds artifacts that significantly contribute to the global approximation quality. Hence, it is necessary to derive two depth images for a position: the depth image \( D \) of the ground truth geometry, and the depth image \( ˆD \) of the current view approximation using all available database artifacts. Note, the depth image \( D \) depends on the used DIBR algorithm (e.g., triangulation or splatting) and its respective parameters (e.g., the distance threshold and point size). In the case of our implementation, we initialize the DIBR pipeline by building the position texture cache using the database backbone.

To automatically tune the initial settings of the DIBR parameters, we derive both \( D \) and \( ˆD \) at the positions where the approximation error can be assumed to be maximal, i.e., at the triangle centers of the current icosahedron refinement. Next, VOIDGA finds a local approximation error minimum by iteratively increasing the DIBR parameters. To do this, it is only necessary to derive the new depth images \( ˆD \) and to compare them to the cached images \( D \) by computing the ADD and the MS-SSIM for each resulting pair. The ADD can be directly computed using the depth images, but the MS-SSIM compares color images. Since our DIBR pipeline is modular, we can simply feed the depth images individually into stage 4 of our rendering system to generate images that are equally shaded. As soon as the error increases, we stop the automatic tuning and communicate the current error and DIBR settings to the user. Although VOIDGA can run fully automatically, this gives users the option to directly compute the current approximation against the ground truth; either by directly contrasting the pairs, or by free camera movement as long as the ground truth can be rendered at interactive framerates. Moreover, users can adjust the database constraints and the error thresholds, which is especially useful if proper initial settings are unknown.

After the automatic tuning, VOIDGA refines the icosahedron grid which yields a set of potential database artifact locations at the new vertex positions. VOIDGA then derives for each position both depth images and computes the error metrics. Instead of storing all depth images \( D \) immediately in the database, VOIDGA discards all pairs that satisfy the error thresholds, and then only adds the ground truth image to the database that has the worst approximation quality. Next, it recomputes the depth images \( ˆD \) at the remaining positions with the current database, and then repeats the previously described process. After all current positions have been processed, VOIDGA again refines the icosahedron, performs the parameter tuning, and selects important samples. This process is repeated until either the maximum number of artifacts is reached, or all candidates satisfy the error thresholds. This scheme reduces the number of stored artifacts, while asserting a minimal approximation quality at the missing sampling locations. In our experiments, sequentially executing this process took roughly one minute. Note, however, deriving new depth images and their scores is embarrassingly parallel.

3.3.3 Database Downsampling

Finally, VOIDGA communicates to the user the impact of the artifact resolutions on the overall approximation quality and the used disk space. Obviously, a lower artifact resolution results in worse approximations, but the benefit of a significant disk space gain might be worth a slightly worse approximation quality. Note, for this stage it is not necessary to actually recompute the depth images \( D \) and \( ˆD \), instead they can be directly downsampled from the high-res images.

4 Results

In the following, we demonstrate the effectiveness of VOIDGA in conjunction with our DIBR implementation on several real-world examples of varying complexity. To validate the quality of the generated views, we examine the approximation error qualitatively and quantitatively based on two image similarity metrics.

4.1 Error Plots

We quantitatively evaluate the approximations generated by the proposed algorithm using the similarity metrics described in Sect. 3.2. Specifically, for a given resolution and number of database elements that were either generated uniformly (U) or via VOIDGA (V), we generate a total of 1,000 random viewing directions over the unit sphere, derive both the ground truth and the approximated view, and subsequently compute the difference between them based on the ADD and MS-SSIM metrics. Fig. 6 depicts the distributions of both metrics grouped first by dataset, then by image resolution, and finally by the different sampling methods. We proceed to demonstrate and discuss these results in detail for each of the three datasets.

4.2 Viscous Fingering

We first demonstrate our approach on datasets that exhibit large smooth surface areas. Specifically, we generated image databases for the simulation ensemble that was provided for the 2016 Scientific Visualization Contest [11]. These simulations model the process of viscous fingering: an instability phenomenon that occurs in porous media at the interface between two fluids of distinct viscosity. In the specific case of the contest dataset, the simulations model a cylinder that is filled with pure water and that contains an infinite salt supply.
at its top. As soon as the salt mixes with the water, the resulting solutions sink down to the bottom as they have a higher density than the surrounding water. Meanwhile, the solutions form structures with increased salt concentration value; called viscous fingers. We follow the approach of Lukasczyk et al. [24] to identify viscous fingers as isosurfaces of the salt concentration density field. Fig. 5 shows the ground truth isosurface geometry and the view approximations, where the viscous fingers and the salt supply are colored bright orange and dark gray, respectively. Quantitative results for this dataset are shown in Fig. 6, left column. For smooth surfaces as the ones found in this case study, triangulation outperforms the splatting technique. Not only does it exhibit a lower approximation error, but also achieves a higher frame rate. Splatting causes a warp of the original surface—i.e., creates an artificial width of the surfaces based on the point size—which causes the generated views to score lower on the image metrics. As shown in Fig. 6, VOIDGA uses fewer images (23) than the complete second icosahedron refinement (42), yet achieves similar error scores. Artifacts that depict the top of the salt supply are discarded by VOIDGA as the smooth surface of the supply can already be approximated by the database backbone.
Figure 7: Generated views for the ground water dataset using different rendering modes and parameters. The images show the path of water (red streamlines) through a karst limestone ground sample (gray) that was taken in south Florida. The dataset was provided by the Texas Advanced Computing Center (TACC) and the Florida International University. All views have been generated using 42 depth images with a resolution of either $512^2$ or $128^2$ pixels. As the approximations show a view angle that was not covered by the used depth images, the approximation error is largest in the cavities where no geometric information is available. Nevertheless, the outer structure is accurately reconstructed even for the relatively low number and resolution of the depth images.
4.3 Asteroid Impact

A second dataset exhibiting large and relatively smooth isocontours is part of a threat assessment study of asteroid ocean impacts [33] that was made publicly available for the 2018 scientific visualization contest [12]. The dataset consists of several extreme scale simulations that model different impact scenarios for varying impact angles, asteroid sizes, and heights of potential airbursts. We generated isosurfaces for the temperature and water density field for impact scenario ya31, i.e., no airburst event, an asteroid diameter of 250 meters, and an entry angle of 45 degrees. Fig. 1 depicts the temperature and water density isosurfaces for level 0.2 eV and 0.002 g/cm³ in orange and blue, respectively. The ground truth geometry consists of roughly three million triangles, while the generated view was derived based on only 12 depth images with a resolution of 512² pixels for each contour, i.e., 24 depth images with an uncompressed total size of 24MB. The images were chosen using VOIDGA to ensure approximation error bounds of 0.001 ADD and 0.97 MS-SSIM for the current view. The large surfaces are accurately approximated, while the base of the water vapor exhibits some approximation artifacts. Recognized, white crep inside the cavities, patches of small features due to the low pixel density of the used depth images. Splatting, on the other hand, will render a point at the location of a small feature as long as it is depicted by at least a single depth image pixel. However, the splatted surface suffers from gaps, and the point borders induce a rough surfaces appearance. Increasing the point size to fill these gaps results in an artificial surface warp that negatively impacts the overall approximation quality.

4.4 Groundwater

In this case study, we demonstrate that our DIBR algorithm can also approximate very complex surfaces with an acceptable error, and that our approach enables the composition of approximated and explicitly stored geometries. To this end, we generated a Cinema database for a karst limestone ground sample that was taken in south Florida. The ground sample was provided by the Texas Advanced Computing Center (TACC) and the Florida International University as a triangulated surface consisting of roughly 8 million triangles (gray surface of Fig. 7 top). Domain experts involved in this research are primarily interested in the propagation of ground water through the stone cavities (red streamlines of Fig. 7). This dataset is challenging for depth image based geometry approximation since the complex structure of the cavities occlude most of the interior geometry. To compensate, it is necessary to sample depth images on a dense grid. In the following, we show that even for the low number of samples chosen by VOIDGA it is possible to adequately reconstruct the outer shell of the stone. However, to demonstrate the effects of undersampling, we use only 42 depth images with a resolution of either 512² or 128² pixels that are sampled on a once subdivided icosahedron (Fig. 2b).

The rows 2-3 and 4-5 of Fig. 7 show approximated views that are generated via surface triangulation or splatting, respectively. The lighting of the complete scene is performed in the post processing shader. Screen space ambient occlusion greatly enhances the perception of the stone porosity and the spatial arrangement of the streamlines. With the sparse sampling the outer structure of the stone is still accurately reconstructed, white crep inside the cavities, as neighboring pixels are too distant of its neighbors at the price of warping the resulting surfaces. This is also reflected in the error metrics (Fig. 6, middle). Especially the MS-SSIM is very sensitive to the surface warps. Furthermore, VOIDGA uses far fewer artifacts (82) than the maximum refinement level (162) while achieving similar approximation errors. Since the geological sample has a relatively smooth backside, VOIDGA primarily stores images that depict the front.

An advantage of the modular design of the demonstrated DIBR algorithm is that the approximated geometry can be rendered together with other, non-approximated geometries. For instance, the red streamlines of Fig. 7 are explicitly stored geometries that are correctly composed with the approximated geometry. Based on this principle, extremely large simulation elements can be approximated by depth images, while specific features of smaller size can be stored explicitly.

4.5 Jet Streamlines

Sparse line geometry is another challenge for depth image based approximation techniques due to the strong depth variations of neighboring pixels. In the following, we demonstrate the quality of our view approximation for streamlines computed from a CFD Jet simulation. It models the injection of a jet into a medium at rest and the friction-based formation of vortical structures.

As expected, the approximations cause large errors for small image resolutions (Fig. 6, right column). Intuitively, the image resolutions are far too low to distinguish between individual streamlines. This is especially negative for the triangulation as neighboring streamlines are falsely connected via surface patches (Fig. 8, right). Although splatting can still produce convincing results even for low resolutions, the necessary large point size blots the streamlines (Fig. 8, left) which has a dramatic impact on the error metrics.

For this dataset, we used VOIDGA to generate a database with a focus on high quality depth approximations rather than image similarity, effectively de-emphasizing color reproduction. Therefore, we enforced a strict hardware MS-SSIM threshold (0.8). This bias towards depth image quality can also be observed in the respective error plots. Note that the VOIDGA database (72 images) and the maximum refinement level (162 images) achieve similar errors.

The colors of the streamlines encode their lifetime and are mapped post hoc. This requires to store for each depth image an additional floating point image that records at each pixel the lifetime of the depicted part of the streamlines. The ability to apply a color map post hoc on the approximated geometries demonstrates that our approach can be easily combined with the existing practice of image databases.
VOIDGA due to its modular design. Moreover, both of our DIBR approximations are fairly rudimentary DIBR approaches. Although they respectively [41]. Furthermore, both metrics evaluate the overall image quality, and thus neglect small but potentially important features.

Our geometry approximation methods (triangulation and splatting) are fairly rudimentary DIBR approaches. Although they require a minimal overhead and already produce acceptable results, more advanced DIBR methods are expected to produce higher quality approximations. Such techniques can easily be integrated into VOIDGA due to its modular design. Moreover, both of our DIBR implementations require parameters (the distance threshold and the point size) whose values have a significant impact on the resulting approximation quality. VOIDGA is capable of automatically finding suitable initial parameters, but the current tuning procedure can get stuck in local extrema. To solve this problem, we will integrate more advanced optimization techniques such as simulated annealing [13].

Naturally, the effectiveness of VOIDGA depends strongly on the used comparison metrics. While we have selected metrics that represent geometry representation (ADD) and image similarity (MS-SSIM), both are not without drawbacks. The most significant problem is their strong dependence on the background to foreground ratio. In effect, a larger background will result in better similarity scores which is not ideal for real-world settings. Moreover, the ADD is computed for normalized depth values, and therefore depends on the precision of the depth buffer. The MS-SSIM, on the other hand, requires input parameters that can currently only be chosen heuristically [41]. Furthermore, both metrics evaluate the overall image quality, and thus neglect small but potentially important features.

**5 LIMITATIONS**

The scope of this paper is to demonstrate the application of DIBR techniques in the context of Cinema databases, with the aim of understanding the value of these techniques for in situ and post hoc visualization. This does not only include the DIBR approximation of views, but also the smart generation of databases that guarantee a maximum approximation error. Advancing from this core concept towards a practical system requires several extensions.

First, as our entire pipeline is based on DIBR techniques, it is necessary to derive and store depth images. Thus, our approach does currently not support the approximation of volume renderings and transparent geometry. In this case, one needs to adapt other techniques such as image warping [16] or volumetric depth images [8].

Our geometry approximation methods (triangulation and splatting) are fairly rudimentary DIBR approaches. Although they require a minimal overhead and already produce acceptable results, more advanced DIBR methods are expected to produce higher quality approximations. Such techniques can easily be integrated into VOIDGA due to its modular design. Moreover, both of our DIBR implementations require parameters (the distance threshold and the point size) whose values have a significant impact on the resulting approximation quality. VOIDGA is capable of automatically finding suitable initial parameters, but the current tuning procedure can get stuck in local extrema. To solve this problem, we will integrate more advanced optimization techniques such as simulated annealing [13].

Naturally, the effectiveness of VOIDGA depends strongly on the used comparison metrics. While we have selected metrics that represent geometry representation (ADD) and image similarity (MS-SSIM), both are not without drawbacks. The most significant problem is their strong dependence on the background to foreground ratio. In effect, a larger background will result in better similarity scores which is not ideal for real-world settings. Moreover, the ADD is computed for normalized depth values, and therefore depends on the precision of the depth buffer. The MS-SSIM, on the other hand, requires input parameters that can currently only be chosen heuristically [41]. Furthermore, both metrics evaluate the overall image quality, and thus neglect small but potentially important features.

**6 CONCLUSION AND FUTURE WORK**

We presented a novel view-approximation oriented image database generation approach (VOIDGA) that determines and stores a minimal set of images for the generation of arbitrary views while bounding the maximum approximation error. As demonstrated on several challenging real-world examples, VOIDGA can reduce image database sizes and the number of images that need to be processed by DIBR methods. VOIDGA can also ensure that a disk space budget is used to its full potential, which stands to be useful for in situ visualization, but also for sharing visualization results where bandwidth usage is of importance.

We examined the resulting approximation error qualitatively and quantitatively via two image-based comparison metrics: the ADD and MS-SSIM. Based on our results, we found that even a relatively low number of database elements (~42) at medium resolution (512²) can already produce high quality approximations. Moreover, smooth surfaces can be well approximated by triangulations, whereas extremely jagged surfaces, sparse line-geometry, and low-resolution depth images are best approximated by splatting.

Towards adapting VOIDGA for production use, many improvements appear possible. As described in Sect. 5, we plan to integrate more advanced DIBR methods and other error metrics to further improve the resulting approximation quality. This is possible due to VOIDGA’s modular design. We also plan to investigate view-dependent resolutions and feature-based camera locations. For example, depictions of smooth surfaces could be stored at low resolution, whereas detailed surface variations and important features are depicted by high-res images. A combination with the work of Nouanesengsy et al. [29] appears fruitful.

**ACKNOWLEDGMENTS**

This research was funded by the German Research Foundation (DFG) within the International Research Training Group IRTG 2057 “Physical Modeling for Virtual Manufacturing Systems and Processes”. This work was also funded by a DOE ASCR program award, Drs. Lucy Nowell and Laura Biven, program managers.

Figure 8: Comparison between the generated views (left and right) and the ground truth (middle) for the jet dataset. To emphasize potential visual errors, the views have been approximated by using only the database backbone (12 depth images) with a resolution of either 256² or 512² pixels for the splatting or triangulation technique, respectively. The color map was applied post hoc and encodes the lifetime of individual streamlines. For sparse line geometry, the triangulated approximation exhibits large artifacts as it connects neighboring streamlines; falsely considering them to be a single surface. Splatting generates comprehensible results, but the points have to be relative large to create the impression of looking at continuous surface geometry. This bloats the streamlines, and therefore the resulting images score badly for the used metrics.
REFERENCES


