

Optimization and Evaluation of Bandwidth-Efficient Visualization for Mobile Devices

Andreas Helfrich-Schkarbanenko, Roman Reiner, Sebastian Ritterbusch, and Vincent Heuveline
Engineering Mathematics and Computing Lab (EMCL)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
 {andreas.helfrich-schkarbanenko, roman.reiner, sebastian.ritterbusch, vincent.heuveline}@kit.edu

Abstract—The visual analysis of large numerical simulations on mobile devices needs remote parallelizable visualization methods for low-bandwidth and high-latency networks. Based on a mathematical model for multi-layered planar impostor representation of arbitrary complex and unbounded scenes, we adapt an algorithm for optimal viewport placement from the theory of optimal experimental design. The results are evaluated in a realistic setting, proving the practical relevance of the theoretical findings, leading to a bandwidth-efficient remote visualization concept for high performance computing simulation results.

Keywords-Remote Visualization, Mobile Visualization, Optimal Experimental Design, Bandwidth Efficiency.

I. INTRODUCTION

Remote visualization is vital wherever local storage, data transfer rates or graphical capabilities are limited. Even though the capabilities of modern smartphones are increasing rapidly, without efficient visualization methods as introduced in [1] many desirable applications are impeded by limitations of the current hardware [2].

Image-based rendering techniques [3] are widely used to reduce the geometric complexity of virtual environments by replacing parts of a scene with a textured representation approximating the original geometry. Since these so-called *impostors* have a significantly simplified geometry, parallax errors [4] occur when rendering the approximation. An impostor is generated for an initial *viewport* (that is, a position and viewing direction) and is said to be *valid* as long as the visual difference to the (hypothetically rendered) original geometry is below a certain threshold.

In our application, these impostors are rendered remotely on render servers and streamed to a mobile device where they are used to approximate the scene. One substantial advantage of the impostor approach [5] is that the render time on the device only depends on the number of impostors and the resolution of the textures, not on the amount of data they display. As long as servers can generate and transfer the impostor textures sufficiently fast, every scene can be displayed remotely, regardless of its actual complexity. In this

setting, network bandwidth is the bottleneck and a careful analysis of bandwidth consumption becomes mandatory.

We develop a mathematical model that allows us to quantify the display error and propose an approximation method that proves to be optimal with respect to the derived error metric. We can show that our method significantly reduces the total amount of image data that needs to be transferred. The key aspects of our method are illustrated in Figure 1: In this simplified two-dimensional case, a traditional remote visualization using one layer would need at least 32 images to provide the same visual accuracy as one layer set of 5 images. This effect is amplified by each additional degree of freedom of the viewer. Based on the error metric that was already presented in [1], this paper extends the method described in [6] with respect to optimally chosen viewport sets locations for fixed numbers of layers, and evaluates the realistic performance of the concepts.

In the following Section II, we discuss related work. Then we introduce the underlying mathematical model in Section III, on which we derive the fundamental error metrics. In Section IV, this leads us to the optimal impostor placement and directly corresponding bounds for the visualization error of one impostor set. The practical outcome of the findings, using as many impostor sets as needed, is proven and evaluated theoretically in Section V. The general placement of viewports for impostor sets is solved by adaption of an algorithm from optimal experimental design to the visualization problem in Section VI. The proposed method is evaluated in Section VII in a realistic setting, which leads us to the conclusions in Section VIII.

II. RELATED WORK

A variety of image-based rendering techniques are reviewed in [5] and [3]. The first paper focuses mainly on techniques using planar impostors but also mentions more exotic approaches like depth images (planar impostors with per-pixel depth information) and light fields. These and other techniques, such as view morphing and view dependent textures, are examined in more detail in the second paper.

In the majority of cases, planar impostors stacked with increasing distance to the observer are used (see [4], [7],

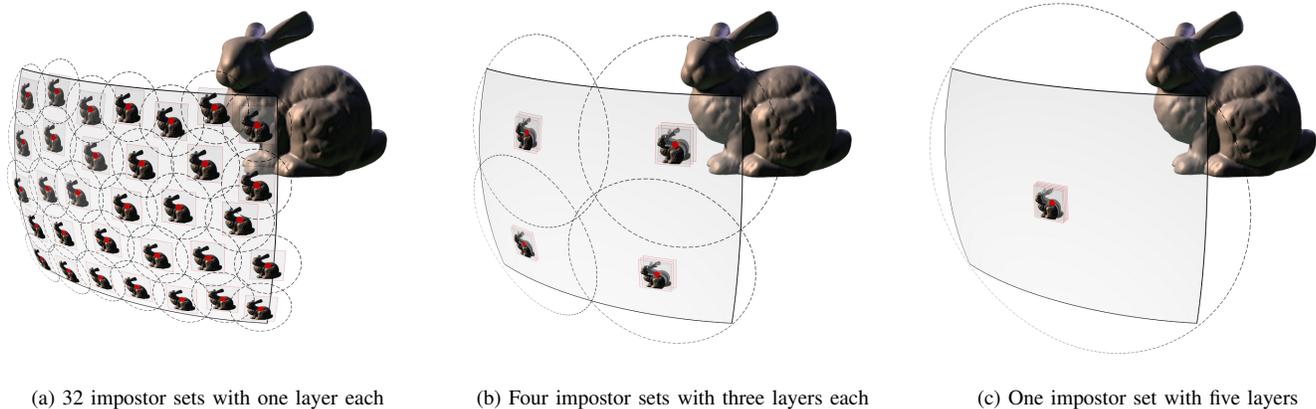


Figure 1. An impostor representation is only valid inside a small region around the initial viewport for which it was originally created. For observer viewports within this validity region (indicated by the dotted line) the display error does not exceed a given maximum value. To faithfully approximate the scene for all observer viewports inside the shaded area, several impostor sets have to be transmitted. The validity regions can be enlarged (while keeping the maximum error unaltered) by increasing the number of layers per impostor set. As the number of required impostor sets decreases faster than the number of layers per set increases, this significantly reduces the total number of layers needed to approximate the scene to a given accuracy .

[8]), usually to approximate distant parts of the scene or single objects. In contrast, our approach uses impostors to represent the full scene.

For large objects, different parts of continuous surfaces can end up on different impostors which makes them tear apart when viewed from a shallow angle. Avoiding this particular problem was one focus of the method developed in [4]. Another interesting use of planar impostors is [9], which treats the rendering of volume data on mobile phones.

Several approaches using geometrically more complex impostors can be found in [8], [10] and [11]. In [5], so-called *billboard clouds* are used to approximate the shape of an object using several intersecting planar impostors. While the impostor creation process for this approach is quite costly, the result allows examination from different viewing directions.

A very current example is Street Slide [12]. Street Slide sticks photos of front facades of urban environments to “panorama strips” that can be browsed by sliding sideways.

The need for accurate analysis of bandwidth and accuracy estimates is discussed in [5], [7], without further specifying how to choose which viewports to load. A more in-depth analysis on the subject of pre-fetching is given in [13] and [14]. The former defines a so-called benefit integral, indicating which parts of the scene – quality-wise – contribute most to the final image, the latter deals with rendering an indoor scene remotely. The task of remote rendering on mobile devices is addressed in [15] and [16], which mostly focuses on the technical aspects of the server-client communication.

Usually, depending on the complexity of the approximation, an impostor is either easy to generate but only valid inside a small region and thus needs to be updated very often, or it is valid inside a large domain but complex and

difficult to generate and display [3]. Since the former strains bandwidth and the latter strains render speed, any image-based rendering approach is usually a trade-off between these limiting factors.

III. VISUALIZATION MODEL AND ERROR METRICS

To begin with, a mathematical model describing viewports and projections thereon needs to be established, with which the rendering and approximation processes can be described. This yields an error function describing the maximum parallax error of a scene as a function of the observer movement, called *domain error*.

Finally, modeling the observer movement as a probability distribution, we can describe the expected value of this error. This *interaction error* will be the cost function that we intend to minimize.

A. Perspective projection

Using homogeneous coordinates and projective transformations [17], we can express perspective projection as a 4×4 matrix multiplication on the projective space \mathbb{P}^3 :

Definition 1. The perspective projection onto the plane $x_3 = d$ towards the origin is a function

$$\pi_d : \begin{cases} \mathbb{P}^3 \setminus \{(0, 0, 0, 1)^\top\} & \longrightarrow & \mathbb{P}^3 \\ x & \longmapsto & P_d x \end{cases}$$

with the parameter $d > 0$ defining the proximity of the projection plane.

From the intercept theorems, one can easily see that the perspective projection of a point $v = (v_1, v_2, v_3)^\top \in \mathbb{R}^3$, $v_3 \neq 0$ onto the plane $x_3 = d$ is given by

$(\frac{d}{v_3}v_1, \frac{d}{v_3}v_2, d)^\top$ which, using homogeneous coordinates, equals $(v_1, v_2, v_3, \frac{v_3}{d})^\top$. This yields the projection matrix

$$P_d := \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{array} \right).$$

B. Viewports

Any viewport can be described by five values $c_1, c_2, c_3 \in \mathbb{R}$, $\vartheta \in [-\pi/2, \pi/2]$, $\varphi \in [-\pi, \pi]$, defining an affine transformation χ , which is the combination of a translation by the vector $(c_1, c_2, c_3)^\top$ followed by a rotation around the x_1 -axis with the angle ϑ and a rotation around the x_2 -axis with the angle φ (cf. Figure 2). Actually, there is a sixth value which represents a rotation around the viewing direction. Such a rotation, however, does not change the image besides rotating it. We assume the rotation to be lossless, which is why we do not need it for our purposes.

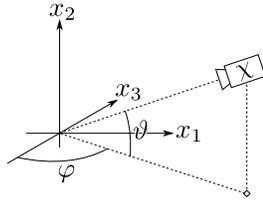


Figure 2. The angles φ and ϑ of a viewport χ

We condense all five values into a single vector $c := (c_1, c_2, c_3, \vartheta, \varphi)^\top$. When describing viewports, we will use this vector c and the associated transformation χ_c interchangeably. In particular, we will identify sets of viewports with subsets of \mathbb{R}^5 :

Definition 2. The set

$$X := \mathbb{R}^3 \times [-\pi/2, \pi/2] \times [-\pi, \pi] \subset \mathbb{R}^5$$

will be called the *viewport set*. For all practical purposes, however, we want to restrict to viewports inside a given set of *feasible viewports* $\Lambda \subset X$.

Projective matrix representations of χ_c and its inverse are

$$Q_c = \left(\begin{array}{cc|c} B_{\vartheta, \varphi} & B_{\vartheta, \varphi} c & \\ \hline 0 & 1 & \end{array} \right) \quad \text{and} \quad Q_c^{-1} = \left(\begin{array}{c|c} B_{\vartheta, \varphi}^\top & -c \\ \hline 0 & 1 \end{array} \right)$$

where

$$B_{\vartheta, \varphi} := \begin{pmatrix} \cos \varphi & -\sin \varphi \sin \vartheta & -\sin \varphi \cos \vartheta \\ 0 & \cos \vartheta & -\sin \vartheta \\ \sin \varphi & \cos \varphi \sin \vartheta & \cos \varphi \cos \vartheta \end{pmatrix}.$$

We can now calculate a matrix representation of a projection onto an arbitrary viewport, by combining the matrices above with the matrix representations of the default projection π_d .

Definition 3. Let χ be a viewport with an associated matrix representation Q and let π_χ denote a projection onto the viewport χ . Then, a matrix representation of π_χ is given by $P_{\chi, d} = QP_dQ^{-1}$, where P_d is the perspective projection matrix defined in Definition 1.

C. Rendering process

Let renderable objects be located in a domain Ω . We aim to simplify the scene by dividing Ω into m disjoint parts Ω_i called *cells*, replacing each with a planar representation of their contained objects. These so-called *impostors* will be created for the same initial viewport(s), that is, for a certain viewport we will create an *impostor set* with one impostor per cell, all for that particular viewport. This will be done for n initial viewports resulting in n impostor sets with m impostors each.

As long as the current viewport matches the initial viewport for which the impostors have been created, the impostor representation coincides with the image of the actual scene. Changing the viewport, however, will introduce parallax errors, since depth information is lost in the impostor creation process.

To determine this error, we will first regard a single cell Ω_i and a single vertex $v \in \Omega_i$. For a fixed initial viewport χ_1 we calculate the impostor representation \bar{v} of the actual point v . Then we consider a variable viewport χ and calculate the screen coordinates v' of v and \bar{v}' of \bar{v} as functions of the viewports χ and χ_1 (cf. Figure 3).

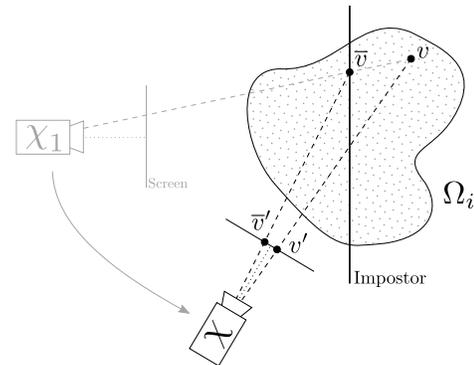


Figure 3. Rendering process for changed viewport

D. The domain error

If we reiterate the procedure above, we obtain two images for each point in Ω : one image of itself (v' , depending on χ) and one of its impostor representation (\bar{v}' , depending on both χ and χ_1). The screen distance of these two, measured in (sub-)pixels is called the *screen space error*. As we are not interested in the error of a single point, but rather in error functions expressing the error of the entire scene, for example the mean error or the maximum error, we aggregate the screen space error over all points in Ω . As the distribution

of vertices inside Ω is supposed to be unknown, we assume a uniform distribution and integrate the screen space error over the entire domain Ω . We will be using the maximum error which replaces the integral with a supremum.

Definition 4. Denote the number of cells with m . For an initial viewport χ_1 we define the *domain error*

$$\begin{aligned} D(\chi, \chi_1) &:= \sup_{v \in \Omega} \|v'(\chi) - \bar{v}'(\chi, \chi_1)\|_2 \\ &= \max_{0 \leq i \leq m} \left\{ \sup_{v \in \Omega_i} \|v'(\chi) - \bar{v}'(\chi, \chi_1)\|_2 \right\}. \end{aligned}$$

This domain error depends on a variable observer viewport χ and the fixed viewport χ_1 , for which the displayed impostor set was initially created. The dependence on χ implies that we cannot evaluate our impostor approximation without knowledge of the observer movement. Clearly, we want to optimize our setup a priori, and hence we need to find a way to evaluate it without knowledge of χ .

E. The interaction error

Assume that we have n impostor sets at hand for viewports $\chi_1, \dots, \chi_n \in \Lambda \subset X$. As before, we denote the observer's viewport with $\chi \in \Lambda$. Since we can choose from several impostor sets, we display that set whose initial viewport χ_k satisfies

$$D(\chi, \chi_k) = \min_{1 \leq j \leq n} D(\chi, \chi_j).$$

For $1 \leq k \leq n$ let Ξ_k denote that subset of Λ , on which $D(\chi, \chi_k)$ is the smallest of all domain errors:

$$\Xi_k := \left\{ \chi \in \Lambda \mid D(\chi, \chi_k) = \min_{1 \leq j \leq n} D(\chi, \chi_j) \right\}. \quad (1)$$

Next, we define a probability distribution P with an associated probability density function μ on Λ , for instance, a uniform distribution over Λ or a normal distribution around the current viewport χ . These distributions represent the probability for the respective viewport to occur, thus modeling the expected observer movement. We can then calculate the expected value of the error by integrating the domain error D over Λ with respect to the probability distribution P .

Definition 5. Let $n \geq 1$. We define the *interaction error* $I: \Lambda^n \rightarrow \mathbb{R}$, where

$$\begin{aligned} I(\chi_1, \dots, \chi_n) &:= \int_{\Lambda} \min_{1 \leq j \leq n} D(\chi, \chi_j) dP(\chi) \\ &= \sum_{j=1}^n \int_{\Xi_j} D(\chi, \chi_j) dP(\chi). \end{aligned} \quad (2)$$

The following Lemma shows that the interaction error will decrease as we add more viewports.

Lemma 1. Let $\chi_1, \dots, \chi_n \in \Lambda$. Then

$$I(\chi_1) \geq I(\chi_1, \chi_2) \geq \dots \geq I(\chi_1, \dots, \chi_n).$$

Proof: For $1 \leq k \leq n$, it is

$$\begin{aligned} I(\chi_1, \dots, \chi_k) &= \int_{\Lambda} \min_{1 \leq j \leq k} D(\chi, \chi_j) dP(\chi) \\ &\leq \int_{\Lambda} \min_{1 \leq j \leq k-1} D(\chi, \chi_j) dP(\chi) \\ &= I(\chi_1, \dots, \chi_{k-1}). \end{aligned}$$

■

IV. IMPOSTOR PLACEMENT AND ERROR BOUNDS

The efficiency of the proposed method is based on an optimal choice of initial viewports for the impostor sets, as well as an optimized cell partition for each set.

Theorem 2. Given renderable objects located in

$$\Omega := \left\{ (x_1, x_2, x_3, 1)^T \in \mathbb{P}^3 \mid 0 < a_0 < x_3 < a_{m+1} \leq \infty \right\},$$

the optimal cell boundaries for viewport translations are given by $a_i = (1/a_0 - i\delta)^{-1}$, $i = 1, \dots, m$ for a suitable $\delta(m) > 0$, and the optimal impostor placement with respect to the error metric is

$$d_i = \frac{2a_i a_{i+1}}{a_i + a_{i+1}}.$$

Note that m is finite even for domains with infinite depth, that is, when $a_{m+1} = \infty$ for which $d_m = 2a_m$.

Proof: For viewport translations the minimum of the domain error D with respect to the projection plane distance $d \in [a, b]$ can be found analytically. For details see [18, Theorem 3.2].

■

With this impostor placement, we have the following asymptotic behavior of the error with respect to viewport translations:

Theorem 3. For a fixed maximal screen space error $\varepsilon > 0$, the radius r of maximal permissible viewport change is proportional to the number of impostors per set m .

Proof: This property emerges during the proof of Theorem 2. For details see [18, Remark 3.5].

■

This Theorem shows that increasing the number of impostors per set will strongly decrease the interaction error, but the number of displayable impostors is bounded by the graphical capabilities of mobile devices. Due to such limitations, several impostors sets have to be transmitted.

Denote the number of impostor sets with n . Under certain assumptions we can show that the inspection error can be bounded by

$$C_1 n^{-1/5} \leq I(\chi_1, \dots, \chi_n) \leq C_2 n^{-1/5},$$

for constants $C_{1/2} = C_{1/2}(\Lambda, m)$. Proving these bounds will be the endeavor of the next section.

V. MODEL EVALUATION

Proposition 1. Using the \mathbb{R}^5 -parametrization of the viewport space, we can regard the domain error $D(\chi, \chi_k)$ as a continuous function $f : \mathbb{R}^5 \times \mathbb{R}^5 \rightarrow \mathbb{R}$ which, for moderate viewport changes, behaves almost linear.

More precisely, we can find positive constants a_1, \dots, a_5 and $\bar{a}_1, \dots, \bar{a}_5$ such that

$$\|A_1(x - y)\| \leq f(x, y) \leq \|A_2(x - y)\| \quad (3)$$

where $A_1 := \text{diag}(a_1, \dots, a_5)$ and $A_2 := \text{diag}(\bar{a}_1, \dots, \bar{a}_5)$.

Proposition 2. The matrices A_1 and A_2 depend on the number of cells m . For viewport translations they are proportional to m^{-1} as a direct consequence of Theorem 3.

Before proceeding, we need the following Lemmata.

Remark 1. In the following $A = B + C$ means that the set A is the direct sum of the sets B and C , that is, $A = B \cup C$ and $B \cap C = \emptyset$. In particular, $\text{vol}(A + B) = \text{vol}(A) + \text{vol}(B)$.

Similarly, $A = B - C$ means that $B = A + C$, that is, $C \subset B$ and $\text{vol}(B - C) = \text{vol}(B) - \text{vol}(C)$.

Lemma 4. Let G be a bounded, measurable, d -dimensional subset of \mathbb{R}^d and let B be a d -dimensional ball (with respect to a norm $\|\cdot\|$) of equal volume (cf. Figure 4a). Then

$$\int_G \|x\| dx \geq \int_B \|x\| dx.$$

Proof: Denote the radius of B with R . Due to $G = G \cap B + G \setminus B$ and $B = G \cap B + B \setminus G$, we can express G as $G = (B - B \setminus G) + G \setminus B$. As the volumes of G and B are equal, this also implies $\text{vol}(G \setminus B) = \text{vol}(B \setminus G)$.

Moreover, the distance from the origin to all points in $G \setminus B$ is larger than R while for all points in $B \setminus G$ it is smaller. Hence,

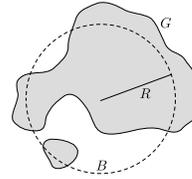
$$\int_{G \setminus B} \|x\| dx \geq \int_{G \setminus B} R dx = R \text{vol}(G \setminus B)$$

and, conversely,

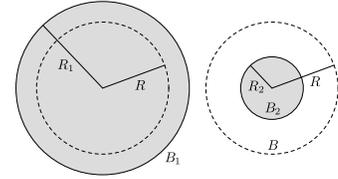
$$\int_{B \setminus G} \|x\| dx \leq \int_{B \setminus G} R dx = R \text{vol}(B \setminus G).$$

This implies

$$\begin{aligned} \int_G \|x\| dx &= \int_B \|x\| dx - \int_{B \setminus G} \|x\| dx + \int_{G \setminus B} \|x\| dx \\ &\geq \int_B \|x\| dx - \underbrace{R(\text{vol}(B \setminus G) - \text{vol}(G \setminus B))}_{=0}. \end{aligned}$$



(a) Lemma 4.



(b) Lemma 5.

Figure 4. Accompanying illustrations for the lemmata.

Lemma 5. Let B and B_1, \dots, B_n be d -dimensional balls (with respect to a norm $\|\cdot\|$), such that the volume of B is the arithmetic mean of the volumes of B_1, \dots, B_n . Then

$$\sum_{k=1}^n \int_{B_k} \|x\| dx \geq n \int_B \|x\| dx.$$

Proof: We first regard the case $n = 2$. Without loss of generality, let $R_1 \geq R \geq R_2$.

We define $G := (B_1 - B) + B_2$. Then, $\text{vol}(G) = \text{vol}(B_1) - \text{vol}(B) + \text{vol}(B_2) = \text{vol}(B)$ and Lemma 4 yields

$$\begin{aligned} \int_B \|x\| dx &\leq \int_G \|x\| dx \\ &= \int_{B_1} \|x\| dx - \int_B \|x\| dx + \int_{B_2} \|x\| dx. \end{aligned}$$

From this, the general case follows by induction. ■

Lemma 6. Let B be a 5-dimensional ball with radius R . Then

$$\int_B \|x\|_2 dx = \frac{4}{9} \pi^2 R^6.$$

Proof: Straightforward calculation using 5-dimensional polar coordinates. ■

With these Lemmata, we can prove the following estimation of the inspection error:

Theorem 7. Let Λ be bounded and assume a uniform distribution of observer viewports. Then, the interaction error can be bounded from below by

$$I(\chi_1, \dots, \chi_n) \geq C_1 n^{-1/5},$$

with the constant

$$C_1 := \frac{5}{6} \left(\frac{15}{8\pi^2} \det(A_1) \text{vol}(\Lambda) \right)^{1/5},$$

where $A_1 := \text{diag}(a_1, \dots, a_5)$ with constants $a_i > 0$ as in Proposition 1.

Proof: Let us first recall (1) and (2). Assuming a uniform distribution $\mu(\chi) = \text{vol}(\Lambda)^{-1}$ we can rewrite (2) as

$$I(\chi_1, \dots, \chi_n) = \text{vol}(\Lambda)^{-1} \sum_{k=1}^n \int_{\Xi_k} D(\chi, \chi_k) d\chi. \quad (4)$$

On the right-hand side, we have to evaluate n integrals of the form $\int_G f(x, y) dx$. Using (3) we define a transformation of coordinates $\Phi(x) := A_1(x - y)$ (which is the same for all n integrals) and obtain

$$\int_G f(x, y) dx \geq \int_G \|\Phi(x)\| dx = \frac{1}{\det(A_1)} \int_{\Phi(G)} \|x\| dx.$$

Applying this to (4) yields

$$I(\chi_1, \dots, \chi_n) \geq (\det(A_1)\text{vol}(\Lambda))^{-1} \sum_{k=1}^n \int_{\Phi_k(\Xi_k)} \|x\| dx. \tag{5}$$

Using Lemmata 4 and 5 (with $d = 5$), we obtain

$$\sum_{k=1}^n \int_{\Phi_k(\Xi_k)} \|x\| dx \geq \sum_{k=1}^n \int_{B_k} \|x\| dx \geq n \int_B \|x\| dx,$$

where

$$\begin{aligned} \text{vol}(B) &= \frac{1}{n} \sum_{k=1}^n \text{vol}(B_k) = \frac{1}{n} \sum_{k=1}^n \text{vol}(\Phi_k(\Xi_k)) \\ &= \frac{1}{n} \det(A_1)\text{vol}(\Lambda). \end{aligned} \tag{6}$$

With this, the estimation (5) yields

$$I(\chi_1, \dots, \chi_n) \geq (\det(A_1)\text{vol}(\Lambda))^{-1} n \int_B \|x\| dx \tag{7}$$

Now, we choose to use the Euclidean norm $\|\cdot\| = \|\cdot\|_2$ for which a 5-dimensional ball with radius R has the volume $\text{vol}(B) = \frac{8}{15}\pi^2 R^5$. Then, (6) implies

$$R = \left(\frac{15}{8n\pi^2} \det(A_1)\text{vol}(\Lambda) \right)^{1/5}.$$

Hence, using Lemma 6,

$$\int_B \|x\| dx = \frac{5}{6n} \det(A_1)\text{vol}(\Lambda) \left(\frac{15}{8n\pi^2} \det(A_1)\text{vol}(\Lambda) \right)^{1/5}.$$

Inserting this into (7) we finally obtain

$$I(\chi_1, \dots, \chi_n) \geq \frac{5}{6} \left(\frac{15}{8n\pi^2} \det(A_1)\text{vol}(\Lambda) \right)^{1/5}.$$

This theorem shows, that the efficiency of any choice of impostor sets cannot be better than the given estimate. The following theorem constructively proves, that a choice of impostor sets with the desired asymptotic dependence exists, that is, that this estimate is actually achievable.

Theorem 8. Let Λ be bounded with a uniform distribution and let $\tilde{\Lambda} \supset \Lambda$ be an enclosing cuboid. Then, there is a set of viewpoints χ_1, \dots, χ_n for which the interaction error satisfies

$$I(\chi_1, \dots, \chi_n) \leq C_2 n^{-1/5},$$

with the constant

$$C_2 := \frac{\pi^2 (\max\{\bar{a}_1, \dots, \bar{a}_5\} \text{diam}(\tilde{\Lambda}))^6}{36 \det(A_2)\text{vol}(\Lambda)},$$

where $A_2 := \text{diag}(\bar{a}_1, \dots, \bar{a}_5)$ with constants $\bar{a}_i > 0$ as in Proposition 1.

Proof: To begin with, we will prove the assertion for those n which are the fifth power of a whole number, that is, for $n^{1/5} \in \mathbb{N}$. The general case will be derived from this case later.

First, a bounded set Λ can be embedded into a cuboid $\tilde{\Lambda}$. For an n chosen as above, there is a regular decomposition of $\tilde{\Lambda}$ into five-dimensional cuboids Ξ_k with initial viewpoints χ_k at their respective centers.

Using the estimation $f(x, y) \leq \|A_2(x - y)\| = \|\Psi(x)\|$ with the same arguments as in the proof of Theorem 7, we obtain

$$\begin{aligned} I(\chi_1, \dots, \chi_n) &\leq \text{vol}(\Lambda)^{-1} \sum_{k=1}^n \int_{\Xi_k} D(\chi, \chi_k) d\chi \\ &\leq (\det(A_2)\text{vol}(\Lambda))^{-1} \sum_{k=1}^n \int_{\Psi_k(\Xi_k)} \|x\| dx \\ &\leq (\det(A_2)\text{vol}(\Lambda))^{-1} n \int_B \|x\| dx, \end{aligned} \tag{8}$$

where we used that all cuboids $\Psi_k(\Xi_k)$ are identical and can be embedded into a ball B in the last step. For this the radius needs to be at least

$$R = \frac{1}{2} \text{diam}(\Psi_k(\Xi_k)) \geq \max\{\bar{a}_1, \dots, \bar{a}_5\} \frac{\text{diam}(\tilde{\Lambda})}{2n^{1/5}}.$$

With this and Lemma 6 we finally obtain from (8)

$$I(\chi_1, \dots, \chi_n) \leq \frac{\pi^2 (\max\{\bar{a}_1, \dots, \bar{a}_5\} \text{diam}(\tilde{\Lambda}))^6}{72 \det(A_2)\text{vol}(\Lambda)} n^{-1/5}.$$

Now, for the general case, we divide $\tilde{\Lambda}$ into $\tilde{n} := \lfloor n^{1/5} \rfloor^5 \leq n$ cubes. This is possible because \tilde{n} is the fifth power of a whole number ($\tilde{n}^{1/5} \in \mathbb{N}$). Moreover,

$$\frac{\tilde{n}^{-1/5}}{n^{-1/5}} = \frac{n^{1/5}}{\lfloor n^{1/5} \rfloor} \leq \frac{\lfloor n^{1/5} \rfloor + 1}{\lfloor n^{1/5} \rfloor} = 1 + \frac{1}{\lfloor n^{1/5} \rfloor} \leq 2,$$

that is, $\tilde{n}^{-1/5} \leq 2n^{-1/5}$. Hence, by this and Lemma (1)

$$\begin{aligned} I(\chi_1, \dots, \chi_n) &\leq I(\chi_1, \dots, \chi_{\tilde{n}}) \\ &\leq \frac{\pi^2 (\max\{\bar{a}_1, \dots, \bar{a}_5\} \text{diam}(\tilde{\Lambda}))^6}{72 \det(A_2)\text{vol}(\Lambda)} \tilde{n}^{-1/5} \\ &\leq \frac{\pi^2 (\max\{\bar{a}_1, \dots, \bar{a}_5\} \text{diam}(\tilde{\Lambda}))^6}{36 \det(A_2)\text{vol}(\Lambda)} n^{-1/5}. \end{aligned}$$

Remark 2. As stated earlier, the matrices A_1, A_2 depend on the number of cells m . With the assumptions in Proposition 2, it follows that $I = \mathcal{O}(m^{-1}n^{-1/5})$.

VI. POSITIONING OPTIMIZATION

In general, the space decomposition into cuboids as utilized in Theorem 8 is far from being optimal. For a given number n of viewpoints, the optimal placement of viewpoints is a high dimensional optimization problem, similar to optimal experimental design (OED) problems.

OED provides techniques, that help to optimize the process of computing unknown parameters in experiments from measurements. The goal is to design the data collection process in such a way that the sensitivity of the measurements with respect to changes in the parameters is maximal, that is, the covariance of the measurement errors is to be minimized.

For the most part, this section follows the description in [19], mainly because of its conciseness. For more profound descriptions refer to (in order of extent) [20], [21], [22].

We assume a compact experimental region $\Omega \subset \mathbb{R}^d$ and denote the unknown parameters with $\theta = (\theta_1, \dots, \theta_p)^\top \in \Theta \subseteq \mathbb{R}^p$. Let $y_\theta(x)$ denote the outcome of the experiment at the location $x \in \Omega$. At fixed locations $\xi = (\xi_1, \dots, \xi_n)^\top$, $\xi_k \in \Omega$ we take measurements

$$z(\xi_k) = y_\theta(\xi_k) + \varepsilon_k \quad \text{for } k = 1, 2, \dots, n, \quad (9)$$

which are prone to measurement errors modeled as independently and identically distributed random variables ε_k with mean zero and variance σ^2 .

To begin with, we assume a linearized model $y_\theta(\xi_k) = f(\xi_k)^\top \theta$ with a response function $f = (f_1, \dots, f_p)^\top$, $f_k \in C(\Omega)$, which allows us to rewrite (9) in matrix notation as $Z = X\theta + \varepsilon$. where $Z = (z(\xi_1), \dots, z(\xi_n))^\top \in \mathbb{R}^n$,

$$X = \begin{pmatrix} f_1(\xi_1) & \cdots & f_p(\xi_1) \\ \vdots & \ddots & \vdots \\ f_1(\xi_n) & \cdots & f_p(\xi_n) \end{pmatrix} \in \mathbb{R}^{n \times p},$$

and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^\top \in \mathbb{R}^n$. If $X^\top X$ is regular, we can compute the least squares estimate (e.g., [20, Th. 1.2.1]) $\hat{\theta} = (X^\top X)^{-1} X^\top Z$. Then, at a point $x \in \Omega$, the predicted response is

$$\hat{z}(x) := f(x)^\top \hat{\theta} \quad (10)$$

with covariance

$$\text{cov}(\hat{z}(x)) = \sigma^2 f(x)^\top (X^\top X)^{-1} f(x) = f(x)^\top M^{-1} f(x), \quad (11)$$

where

$$M = M(\xi) := \frac{1}{\sigma^2} X^\top X. \quad (12)$$

The design problem is to find an *optimal design* ξ such that (10) is optimal in describing the actual experiment, that is, that $\text{cov}(\hat{z})$ is *minimal*, often implemented for example in optimization of the determinant of M for D-optimality, or of the eigenvalues of M denoted as E-optimality.

Any design ξ can be regarded as a measure on Ω [19, p. 16]: Suppose we are taking n measurements at the locations

ξ_1, \dots, ξ_n . Then we can interpret ξ as a probability measure on Ω if we define

$$\xi(x) := \frac{1}{n} \sum_{k=1}^n \delta_{\xi_k}(x).$$

For such a design ξ , we can define M by

$$m_{i,j}(\xi) := \frac{1}{\sigma^2} \int_{\Omega} f_i(x) f_j(x) d\xi$$

for $i, j = 1, \dots, p$, $M(\xi) := (m_{i,j}(\xi))$. Similarly, the variance function (11) can be generalized to

$$d(x, \xi) := f(x)^\top M(\xi)^{-1} f(x).$$

An algorithm due to Wynn, Mitchell and Miller for a fixed number n -point design optimization for D-optimality reads [19, p. 20]:

- 1) Begin with an arbitrary n -point design $\xi^{(0)}(n)$.
- 2) Find ξ_{n+1} such that

$$d(\xi_{n+1}, \xi^{(j)}(n)) = \max_{x \in \Omega} d(x, \xi^{(j)}(n+1))$$

and add ξ_{n+1} to the n -point design.

- 3) Find ξ_k such that

$$d(\xi_k, \xi^{(j)}(n+1)) = \min_{1 \leq i \leq n+1} d(\xi_i, \xi^{(j)}(n+1))$$

and remove ξ_k from the $(n+1)$ -point design.

- 4) Repeat steps 2 and 3 until the exchange does not result in an increase of $\det[M(\xi^{(j)}(n))]$.

This algorithm optimizes an n -point design by repeating the two following steps:

- Add the point of minimum covariance to the n -point design.
- Remove the point of maximum covariance from the $(n+1)$ -point design.

Note, that the covariances (and thereby the points added and removed in each step) depend on the current design.

This algorithm is adapted and extended to the viewpoint positioning optimization problem. Our goal is to find a viewpoint set $\{\chi_1, \dots, \chi_n\} \subset \Lambda$ for a given $n \in \mathbb{N}$, which minimizes the inspection error $I(\chi_1, \dots, \chi_n)$. Theorem 1 states, that adding any viewpoint will cause a decrease of the inspection error, while removing any viewpoint will increase it. Emulating the algorithm above, we start with an initial set of n viewpoints and hope to find an optimal viewpoint set by repeating the two following steps:

- Add that viewpoint for which the decrease of the inspection error is maximal.
- Remove that viewpoint for which the increase of the inspection error is minimal.

An implementation is given in

Algorithm 1

1. Begin with an arbitrary set of n viewpoints χ_1, \dots, χ_n

2. **repeat**
3. $I_0 \leftarrow I(\chi_1, \dots, \chi_n)$
4. $I_{\min} \leftarrow I_0$
5. **for** $\chi \in \Lambda$
6. **if** $I(\chi_1, \dots, \chi_n, \chi) < I_{\min}$
7. $\chi_{n+1} \leftarrow \chi$
8. $I_{\min} \leftarrow I(\chi_1, \dots, \chi_n, \chi)$
9. Add χ_{n+1} to the set of viewpoints
10. $I_{\min} \leftarrow I_0$
11. **for** $i \leftarrow 1$ **to** $n + 1$
12. **if** $I(\chi_1, \dots, \chi_{i-1}, \chi_{i+1}, \dots, \chi_{n+1}) < I_{\min}$
13. $k \leftarrow i$
14. $I_{\min} \leftarrow I(\chi_1, \dots, \chi_{k-1}, \chi_{k+1}, \dots, \chi_{n+1})$
15. Remove χ_k from the set of viewpoints
16. $(\chi_1, \dots, \chi_n) \leftarrow (\chi_1, \dots, \chi_{k-1}, \chi_{k+1}, \dots, \chi_{n+1})$
17. **until** $I_0 - I_{\min} < \kappa$

Lemma 9. Algorithm 1 is monotonically decreasing in I.

Proof: At the j -th iteration of the algorithm denote the set of starting viewpoints with X_j and the viewpoints added and removed in the intermediate steps with χ_{n+1} and χ_k respectively. Let further $X_{j+\frac{1}{2}} := X_j \cup \{\chi_{n+1}\}$ and $X_{j+1} := X_{j+\frac{1}{2}} \setminus \{\chi_k\}$. Then, $I(X_{j+1}) = I(X_{j+\frac{1}{2}} \setminus \{\chi_k\}) = \min [I(X_{j+\frac{1}{2}} \setminus \{\chi_i\})] \leq I(X_{j+\frac{1}{2}} \setminus \{\chi_{n+1}\}) = I((X_j \cup \{\chi_{n+1}\}) \setminus \{\chi_{n+1}\}) = I(X_j)$. ■

We implement the algorithm for our well-understood reduced problem of parallel viewport translations. This has several advantages. Firstly, it is rather simple to implement; secondly, since the domain error does not need to be calculated by integration, it has moderate calculation times; and lastly, since Λ is only two-dimensional, the results can be easily displayed.

We observe that, rather than converging to a design with n points, the algorithm, after a while, cyclically generates subsets of cardinality n of one design with $n + 1$ points (cf. Figure 5).

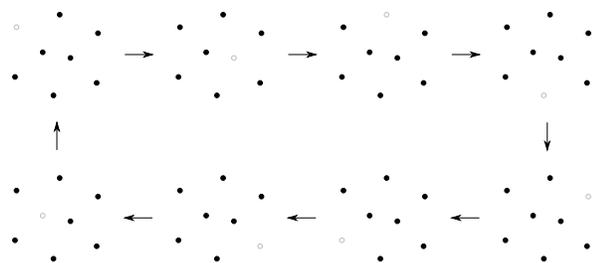


Figure 5. Cyclically generated subsets for $n = 7$

Further analysis shows, that at this point the first step of the algorithm always reproduces the same $(n + 1)$ -point design, from which in the second step one point is removed resulting in a subsets with n elements. That the

points are removed cyclically is due to the fact, that in our implementation of the algorithm, from several points of equal weakness the first one is removed, while new elements are always added at the end. Since, in a way, the algorithm converges to an $(n + 1)$ -point design, we can eradicate that problem by simply switching the two steps of the algorithm. Then the algorithm converges to a design with n -points, and the subsets of cardinality $n - 1$ are those generated between the steps. Once the optimal n -point design has been reached, the algorithm cyclically picks a point from the set, removes it from the design in the first step and immediately adds it again in the second step. Hence, the algorithm converges once n iterations in a row do not result in a different design.

With the two steps interchanged the algorithm does indeed converge to an n -point design as desired. However, especially for large n , it requires quite a lot of steps to turn the arbitrary initial design into a “reasonable” design which is then optimized further. Therefore, rather than starting with a random design, we hope to improve the algorithm by generating an initial design as follows.

Emulating the OED optimization algorithm by Federov, described in [19], we start with an empty design and successively add points, which are in some sense optimal, until we get to an n -point design. We do this by simply running the second step of the algorithm n times. This way, every point added to the initial design maximizes the decrease of the inspection error, resulting in an initial design which is rather good already. Note, that even though the point added in the k -th step in this manner is the optimal choice, the resulting k -point design is usually not optimal. For example, for $n = 3$ and a normal distribution, the first point will end up in the center and the second and third point will end up opposite of each other forming a line with the first point, while the optimal design would be three points forming an equilateral triangle around the center.

An updated version of our algorithm including the generation of an initial design is given below.

Algorithm 2

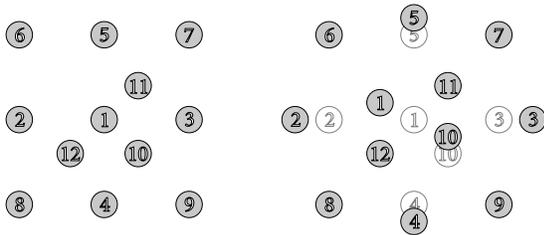
1. **for** $j \leftarrow 1$ **to** n
2. $I_{\min} \leftarrow \infty$
3. **for** $\chi \in \Lambda$
4. **if** $I(\chi_1, \dots, \chi_{j-1}, \chi) < I_{\min}$
5. $\chi_j \leftarrow \chi$
6. $I_{\min} \leftarrow I(\chi_1, \dots, \chi_{j-1}, \chi)$
7. Add χ_j to the set of viewpoints
8. **repeat**
9. $I_0 \leftarrow I(\chi_1, \dots, \chi_n)$
10. $I_{\min} \leftarrow I_0$
11. **for** $i \leftarrow 1$ **to** n
12. **if** $I(\chi_1, \dots, \chi_{i-1}, \chi_{i+1}, \dots, \chi_n) < I_{\min}$
13. $k \leftarrow i$
14. $I_{\min} \leftarrow I(\chi_1, \dots, \chi_{i-1}, \chi_{i+1}, \dots, \chi_n)$

15. Remove χ_k from the set of viewpoints
16. $(\chi_1, \dots, \chi_{n-1}) \leftarrow (\chi_1, \dots, \chi_{k-1}, \chi_{k+1}, \dots, \chi_n)$
17. $I_{\min} \leftarrow I_0$
18. **for** $\chi \in \Lambda$
19. **if** $I(\chi_1, \dots, \chi_{n-1}, \chi) < I_{\min}$
20. $\chi_n \leftarrow \chi$
21. $I_{\min} \leftarrow I(\chi_1, \dots, \chi_{n-1}, \chi)$
22. Add χ_n to the set of viewpoints
23. **if** $I_0 = I_{\min}$
24. $m \leftarrow m + 1$
25. **else**
26. $m \leftarrow 0$
27. **until** $m = n$

Lemma 10. After step 7, Algorithm 2 is monotonically decreasing in I .

Proof: At the j -th iteration of the algorithm denote the set of starting viewpoints with X_j and the viewpoints added and removed in the intermediate steps with χ_{n+1} and χ_k respectively. Let further $X_{j+\frac{1}{2}} := X_j \setminus \{\chi_k\}$ and $X_{j+1} := X_{j+\frac{1}{2}} \cup \{\chi_{n+1}\}$. Then, $I(X_{j+1}) = I(X_{j+\frac{1}{2}} \cup \{\chi_{n+1}\}) = \min_{\chi \in \Lambda} [I(X_{j+\frac{1}{2}} \cup \{\chi\})] \leq I(X_{j+\frac{1}{2}} \cup \{\chi_k\}) = I((X_j \setminus \{\chi_k\}) \cup \{\chi_k\}) = I(X_j)$. ■

The following Figure shows the resulting patterns of the initial and the optimal design. The numbering of the points reflects their order of appearance in phase 1, i.e., after step 7.



(a) Initial design after step 7. (b) Optimal design after step 27.

Figure 6. Testing Algorithm 2 for $n = 12$ and a normal distribution.

As long as the observer is not moving, pre-fetching the viewpoints as proposed by the algorithm results in an optimal set of data sets. However, in general, the optimal viewport distribution for any two different observer locations do not share any viewpoints. Hence, if we always pre-fetch those data sets which are optimal for the current observer location, we need to update *all* data sets whenever the observer is moving, rendering this approach useless. Instead, since the algorithm works by optimizing a given viewport distribution, it can be used adaptively. That is, rather than pre-fetching those data sets for the optimal distribution, we can update the current state, where all updates are in order of their importance. This way, even though the intermediate steps might not be optimal, every update is taking account of

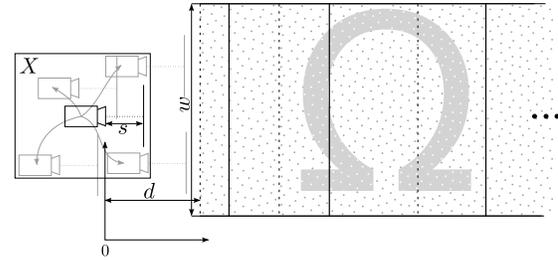


Figure 7. Test setup for evaluation

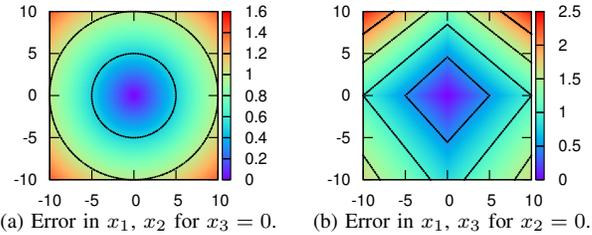


Figure 8. Screen space error distribution in pixels over view position change in pixels for 5-layer viewport centered in X .

the data sets already available on the device. An optimal distribution is only attained, if the observer is standing still long enough for all data sets to be updated to their optimal viewport.

VII. NUMERICAL TESTS

Since the positioning optimization is not trivial, we present the actual performance of the method in a test setup. In Figure 7, we see a half-infinite cylinder $\Omega \in \mathbb{R}^3$ with diameter w representing the visualization volume. The unmoved camera is placed in distance d from Ω and the visualization screen is fixed at distance s from the camera. When the camera is moving away from the center of viewport set X , the screen is moving with the camera, but Ω remains fixed.

In the following evaluation, the parameters $w = 100$, $d = s = 100$, and $X = [-10, 10]^3$ with uniform distribution were used. For illustration, the camera is not rotated, and thus we have a three dimensional viewport space and expect to achieve $I = \mathcal{O}(m^{-1}n^{-1/3})$ in this setup.

First of all, Figure 8 presents the error distribution for one viewport with 5 layers in two cut-planes through X for camera motion $x_3 = 0$ and $x_2 = 0$. Moving the camera in constant distance yields a symmetric error distribution, as it can be seen in Figure 8 (a). Moving towards Ω results in higher errors than moving away in Figure 8 (b), due to the displayed size increase of nearer objects.

Figures 9 and 10 each evaluate the average interaction error in pixels for an increasing number of viewpoints and layers. As expected, the evaluation of the error appears to be proportional to the inverse of the number of layers $I = \mathcal{O}(m^{-1})$, as presented in Theorem 3. The increase of

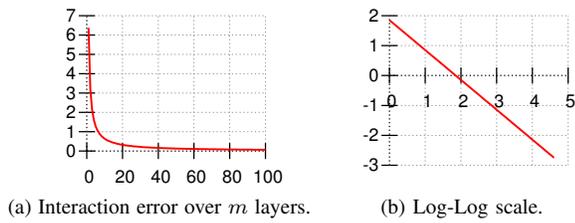


Figure 9. Interaction error for $n = 1$ viewport.

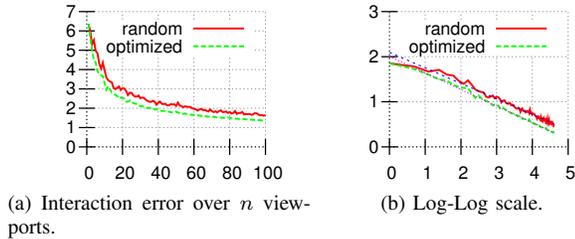


Figure 10. Interaction error for $m = 1$ layer.

viewports, on the other hand, involves choosing the locations of each reference viewport.

The theoretic results detail the asymptotic behavior, but they are quite rough for viewport numbers other than fifth power of whole numbers. Thus, the performance could deviate from our expectations for specific cases in real scenarios. But this is not the case, as Figure 10 illustrates the validity of the asymptotics for realistic cases.

Due to the computational costs for viewport position optimization, the results were compared to random distribution of viewports within X . To avoid unnecessarily skewed results for random placing, the first viewport was set to the center of X . In general, the optimized placement improves the performance, but both methods expose nearly the same order of convergence of $I \approx \mathcal{O}(n^{-1/3})$. The optimized curve also shows some irregularities due to geometric effects for certain viewport numbers, for example a cubic number of viewports can be placed more efficiently in a cube than any other number. The gain of optimization is significant, as for example the interaction error of 2 is reached for $n = 36$ in the optimized version, whereas the random method needs $n = 59$. Also the rate of convergence appears to be slightly worse, but considering the computational costs for optimization, a trade-off can be considered for real-time applications.

Figure 11 illustrates the pixel errors for 27 randomly chosen viewports with 27 layers each in two cut-planes through X for $x_3 = 0$ and $x_2 = 0$, corresponding to Figure 8. The first viewport located at $x_1 = x_2 = x_3 = 0$ is the only viewport placed on the cut-plane, all others were randomly distributed in X . Depending of the position, the scene will be visualized using the layers from the viewport

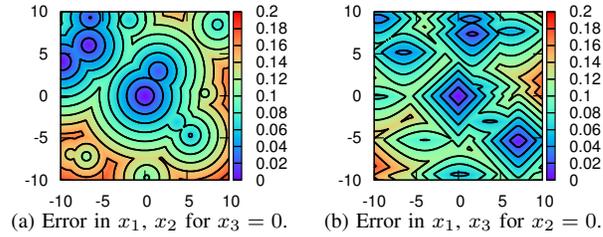


Figure 11. Screen space error distribution in pixels over view position changes in pixels for 27 random viewports with 27 layers.

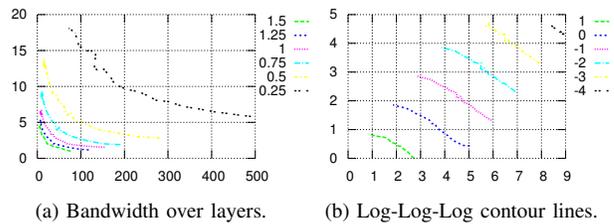


Figure 12. Error contour lines for number of total images over layers.

with lowest error contribution, leading to a continuous error function over X .

The actual performance of the method is evaluated in Figure 12, plotting the interaction error over the total bandwidth in dependence of number of layers needed. The total bandwidth is estimated by the number of images nm for n viewports each having m layers that need to be transmitted. There is a problem of sparse data for evaluation, for example 15 total images can result from either 3 viewports with 5 layers, and vice versa. This was overcome by using differing layer numbers during the tests, yielding the fractional layer number needed, for example 15 total images split on 2 viewports with 7.5 layers, in average. Figure 12 (a) denotes the total images on the x -axis, and average layers on the y -axis. The graphs are contour-lines of same interaction error. It is clearly visible, that both the increase of bandwidth and the increase in layers with constant bandwidth reduces the error. This clearly shows, that given a limited bandwidth, the interaction error can be as low as how many layers the output device can handle. Additionally, Figure 12 (b) presents the performance $I \approx \mathcal{O}(m^{-1}n^{-1/3})$ in the experiment, as it was predicted before.

VIII. CONCLUSION

In this paper, we developed a mathematical model which allows to measure, analyze and optimize the display error of image-based approximation techniques, presented an algorithm for viewport location optimization, and evaluated the performance of the method under realistic conditions. Both the error asymptotics derived for our method based on parallelized rendering, as well as the experimental results, show a clear advantage over traditional remote visualization

concepts like Virtual Network Computing (VNC) which, under ideal conditions, represent the scene by one image $m = 1$ without image warping, leading to doubled interaction errors than presented here.

In contrast to this, $m = 10$ impostors with $n = 1$ viewport cover the same volume of permissible viewports as $m = 1$ impostors for $n = 100000$ optimally chosen viewport sets. The latter is using a bandwidth of $\mathcal{O}(mn)$ that is 10000-fold higher. Comparing this to the bandwidth needed for transmission of impostors compared with their error contribution $\mathcal{O}(m^{-1}n^{-1/5})$, the method offers significant decrease of bandwidth consumption. By avoiding high network latencies, the user experiences low latency rendering.

The proposed method strongly benefits from graphical capabilities of clients, such as mobile devices, and will increase its efficiency for each new generation providing increased graphical performance. Due to the parallelization of server-sided image generation, and the proven efficiency thereof, the method is applicable to large and distributed data sets for visualization on mobile devices and thin clients, also including augmented reality applications [23].

ACKNOWLEDGMENT

The authors appreciate the support of the 'Federal Ministry of Education and Research' and 'Eurostars' within the Project E! 5643 MobileViz. The Eurostars Project is funded by the European Union.

REFERENCES

- [1] A. Helfrich-Schkarbanenko, V. Heuveline, R. Reiner, and S. Ritterbusch, "Bandwidth-efficient parallel visualization for mobile devices," in *INFOCOMP 2012, The Second International Conference on Advanced Communications and Computation*, 2012, pp. 106–112.
- [2] F. Lamberti and A. Sanna, "A streaming-based solution for remote visualization of 3D graphics on mobile devices," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 2, pp. 247–260, march-april 2007.
- [3] H.-Y. Shum and S. B. Kang, "A review of image-based rendering techniques," in *IEEE/SPIE Visual Communications and Image Processing*, 2000, pp. 2–13.
- [4] S. Jeschke and M. Wimmer, "An error metric for layered environment map impostors," Institute of Computer Graphics and Algorithms, Vienna University of Technology, Technical Report TR-186-2-02-04, 2002.
- [5] S. Jeschke, M. Wimmer, and W. Purgathofer, "Image-based representations for accelerated rendering of complex scenes," *Eurographics 2005 STAR Report*, vol. 1, 2005.
- [6] V. Heuveline, M. Baumann, S. Ritterbusch, and R. Reiner, "Method and system for scene visualization," Mar. 1 2013, wO Patent 2,013,026,719.
- [7] S. Jeschke, M. Wimmer, and H. Schuman, "Layered environment-map impostors for arbitrary scenes," *Graphics Interface*, pp. 1–8, May 2002.
- [8] W.-C. Wang, K.-Y. Li, X. Zheng, and E.-H. Wu, "Layered Textures for Image-Based Rendering," *Journal of Computer Science and Technology*, vol. 19, no. 5, pp. 633–642, September 2004.
- [9] M. Moser and D. Weiskopf, "Interactive volume rendering on mobile devices," in *13th Fall Workshop: Vision, Modeling, and Visualization 2008*, O. Deussen, D. Keim, and D. Saupe, Eds. Akademische Verlagsgesellschaft AKA GmbH, 2008, p. 217.
- [10] J. Cohen, D. Manocha, and M. Olano, "Simplifying polygonal models using successive mappings," in *VIS '97: Proceedings of the 8th Conference on Visualization '97*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1997, p. 395.
- [11] P. Debevec, Y. Yu, and G. Boshokov, "Efficient view-dependent image-based rendering with projective texture-mapping," University of California at Berkeley, Berkeley, CA, USA, Technical Report, 1998.
- [12] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street slide: browsing street level imagery," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–8, 2010.
- [13] J. Shade, D. Lischinski, D. H. Salesin, T. DeRose, and J. Snyder, "Hierarchical image caching for accelerated walk-throughs of complex environments," in *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM, 1996, pp. 75–82.
- [14] T. A. Funkhouser, "Database management for interactive display of large architectural models," in *GI '96: Proceedings of the Conference on Graphics Interface '96*. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 1996, pp. 1–8.
- [15] M. Hoffmann and J. Kohlhammer, "A Generic Framework for Using Interactive Visualization on Mobile Devices," *Communications in Computer and Information Science*, vol. 53, no. 4, pp. 131–142, 2009.
- [16] F. Lamberti, C. Zunino, A. Sanna, A. Fiume, and M. Maniezzo, "An accelerated remote graphics architecture for PDAs," in *Web3D '03: Proceedings of the Eighth International Conference on 3D Web Technology*. New York, NY, USA: ACM, 2003, pp. 55–61.
- [17] A. Beutelspacher and U. Rosenbaum, *Projective Geometry: From Foundations to Applications*. Cambridge University Press, February 1998.
- [18] R. Reiner, "Numerical Methods for Optimal Impostor Prefetching in Scientific Visualization," Diploma Thesis, Karlsruhe Institute of Technology, 2011.
- [19] R. C. St.John and N. R. Draper, "D-optimality for regression designs: a review," *Technometrics*, vol. 17, no. 1, pp. 15–23, 1975.
- [20] V. B. Melas, *Functional Approach to Optimal Experimental Design (Lecture Notes in Statistics)*. Springer-Verlag New York, Inc., 2005.

- [21] A. F. Emery and A. V. Nenarokomov, "Optimal experiment design," *Measurement Science and Technology*, vol. 9, no. 6, pp. 864–876, 1999.
- [22] D. Ucinski, *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.
- [23] V. Heuveline, S. Ritterbusch, and S. Ronnas, "Augmented reality for urban simulation visualization," in *INFOCOMP 2011, The First International Conference on Advanced Communications and Computation*. IARIA, 2011, pp. 115–119.