Pervasive and Mobile Computing 45 (2018) 19–34

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

# Enabling interactive mobile simulations through distributed reduced models

Christoph Dibak [a,*], Bernard Haasdonk [b], Andreas Schmidt [b], Frank Dürr [a], Kurt Rothermel [a]

[a] Institute of Parallel and Distributed Systems, University of Stuttgart, Stuttgart, Germany
[b] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Stuttgart, Germany

ARTICLE INFO

*Article history:*
Available online 13 February 2018

*Keywords:*
Augmented reality
Mobile computing
Mobile cyber–physical systems
Mobile simulations

ABSTRACT

Currently, various hardware and software companies are developing augmented reality devices, most prominently Microsoft with its Hololens. Besides gaming, such devices can be used for serious pervasive applications, like interactive mobile simulations to support engineers in the field. Interactive simulations have high demands on resources, which the mobile device alone is unable to satisfy. Therefore, we propose a framework to support mobile simulations by distributing the computation between the mobile device and a remote server based on the reduced basis method. Evaluations show that we can speed-up the numerical computation by over 131 times while using 73 times less energy.



## 1. Introduction

Recently, several hardware and software companies announced or already released augmented reality devices, such as Microsoft's Hololens, Sony's SmartEyeglass, or Epson's Moverio Pro. Using such devices allows the user to see the real world augmented with additional information, such as the state of the object the user is looking at or even holograms displaying 3D scenes over the real world.

One interesting application of such devices is to display the result of numerical simulations. Having simulation results ubiquitously available supports engineers or decision makers in the field [1–3]. As an example for such pervasive applications, consider an engineer who has to find a solution for placing a hot exhaust tube during deployment of a machine in a factory. To this end, the engineer uses her augmented reality device, which directly shows the heat of the surface of the tube as if the machine were operational. She can adjust the airflow surrounding the tube by changing parameters. The application enables her to see the heat even in complex regions, e.g., in bends, and she can place the tube according to surrounding materials. Other applications for mobile simulations are the visualization of simulation results based on readings from nearby sensors for Internet of Things applications or simulations on drones in order to exploit the wind-field for energy-efficient flight routes [4].

The main challenge for the computation of simulation results on mobile devices is the computational complexity. While it is feasible to visualize even complex 3D data using dedicated GPUs on the mobile device, calculating this data is still challenging due to slow processors and limited energy resources of mobile devices. Therefore, we present an approach to support interactive simulations on resource-constrained mobile devices by distributing computations between the mobile device and a server infrastructure. Moreover, the programmer should be able to re-use existing simulation code. Such

* Corresponding author.
*E-mail addresses:* dibak@ipvs.uni-stuttgart.de (C. Dibak), haasdonk@mathematik.uni-stuttgart.de (B. Haasdonk), schmidta@mathematik.uni-stuttgart.de (A. Schmidt), duerr@ipvs.uni-stuttgart.de (F. Dürr), rothermel@ipvs.uni-stuttgart.de (K. Rothermel).

https://doi.org/10.1016/j.pmcj.2018.02.002
1574-1192/

code is typically optimized for the server architecture, which might include special hardware, such as additional GPUs, or vector processing instruction sets, not available on the mobile device. Besides the computational aspect, the communication overhead is critical for an efficient distributed execution on mobile devices and a remote server. Therefore, we also need to minimize the amount of data communicated between mobile device and server.

To support numerical simulations on mobile devices, we present a middleware in this paper that allows for the efficient distribution of simulations between a mobile device and remote server infrastructure. This middleware is based on the so-called Reduced Basis Method (RBM) that allows for reducing the complexity of simulations by pre-calculating an optimized basis reducing the dimensionality of the simulation problem. The basic idea of applying the RBM to distributed mobile simulations is to calculate the reduced basis remotely on a server and to transfer the basis to the mobile device. Using the reduced basis, the mobile device is able to perform simulations locally with reduced computational overhead to calculate approximate solutions, where the reduction of quality of approximate solutions is well-defined and bounded.

Using the RBM for efficient distributed simulations is not completely novel and has already been proposed in [5]. However, approaches so far are rather static. They calculate and deploy a basis once, which is then used for all subsequent simulations. In contrast, our approach allows for the dynamic adaptation of the basis during runtime.

In detail, the contributions of this article are: (1) identification of a new class of serious augmented reality applications, namely interactive mobile simulations; (2) presentation of a mobile simulation middleware that supports the programmer in implementing such applications; (3) two approaches to enable mobile interactive simulations with and without a-priori known parameter restrictions; (4) two approaches optimizing the bottleneck on mobile devices, which is reading data from internal storage; (5) one approach to improve the pre-calculation step of the RBM for our mobile approach; (6) real-world evaluations, including comparison of run-time and energy consumption of the methods, showing that our approaches are up to over 131 times faster and consume 73 times less energy compared to offloading everything to a server.

This work is an extension of our prior work [6], in which we presented two approaches to enable mobile interactive simulations and one approach to reduce the amount of data to be read from internal storage. In this work, we extend our previous work by another approach for reducing the read operations which is built on top of the previously introduced approach. Additionally, we present a novel approach for the optimized pre-calculation on the server that further reduces the energy cost during runtime on the mobile device.

The rest of the paper is structured as follows. Section 2 provides background on the RBM followed by the system model in Section 3. Section 4 introduces a basic approach, which will be modified to the adaptive approach in Section 5. In order to reduce the number of snapshots during the computation, we present the subspace approach and the reorder approach in Sections 6 and 7. Section 8 introduces a new approach for pre-processing on the server in order to improve the reorder approach. Section 9 evaluates our approaches, before we discuss related work in Section 10 and conclude the paper.

## 2. The reduced basis method

Our approach to enable interactive mobile simulations utilizes the Reduced Basis Method (RBM) to solve complex numerical problems by calculating parts of the simulation on a remote server infrastructure. In order to better understand the problem to be solved and the solution, we first give a brief introduction to the numerical problems to be solved and the RBM in this section, before we explain the approach in the following section. A more in depth explanation of RBM can be found in [7–9].

### *2.1. The full numerical problem*

Simulations predict the behavior of a system based on a model. Commonly, such models are described using differential equations. Such equations need to be discretized, which leads to large algebraic equations of the form $A \cdot u = f$, where $A$ is a given matrix, $f$ is a given vector called the right-hand side, and $u$ is the unknown solution. We call this equation the *full problem*.

Simulation models contain parameters describing different properties of the system that can be changed. Such parameters can be used to interact with the system, e.g., to insert sensor readings or user input. To express the dependency on parameters, we formulate the full problem as

$$\mathbf{A}(\mu) \cdot \mathbf{u}(\mu) = \mathbf{f}(\mu) \tag{1}$$

where $\mu$ is a vector including all parameters of the simulation.

### *2.2. Parameter separable matrices*

The essential part of RBM is the parameter separability of the matrix $A(\mu)$ and the right-hand side $f(\mu)$. Parameter separability of a vector or matrix $M$ is given if we know scalar functions $\theta_i$ that map from the parameter space to real numbers and matrices $M_i$ that have the same shape as $M$ such that

$$\mathbf{M}(\mu) = \sum_i \theta_i(\mu)\mathbf{M}_i. \tag{2}$$

Such a representation can be derived from the model equation or using Empirical Interpolation Methods [10].