Augmented Reality Visualization of Numerical Simulations in Urban Environments

Sebastian Ritterbusch, Staffan Ronnås, Irina Waltschläger, Philipp Gerstner, and Vincent Heuveline
Engineering Mathematics and Computing Lab (EMCL)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{sebastian.ritterbusch, staffan.ronnas, vincent.heuveline}@kit.edu, {irina.waltschlaeger, philipp.gerstner}@student.kit.edu

***Abstract*—Visualizations of large simulations are not only computationally intensive but also difficult for the viewer to interpret, due to the huge amount of data to be processed. In this work, we present a novel Augmented Reality visualization method, which enables simulations based on current city model data to be presented with localized real-world images. Test scenarios of urban wind flow and fine dust simulations illustrate the benefits of mobile Augmented Reality visualizations, both in terms of selection of data relevant to the user and facilitation of comprehensible access to simulation results.**

***Keywords*-Scientific Visualization, Augmented Reality, Numerical Simulation, Urban Airflow, Geographical Information Systems.**

I. Introduction

Numerical simulation and interactive 3D visualization has today become an essential tool in many applications, including industrial design, studies of the environment and meteorology, and medical engineering. The increasing performance of computers has played an important role for the applicability of numerical simulation but has also led to a rapid growth in the amount of data to be processed. At present, the use of simulation software and the interpretation of visualization results usually require dedicated expertise. The large amount of data available leads to two problems for the end-user, which are discussed in this paper extending [1]. On the one hand, handling and selection of the appropriate data requires a suitable user interface. On the other hand, the amount of perceptible information is limited, and thus visualizations of large data sets need very intuitive methods to be understandable.

The use of Augmented Reality (AR) is aiming at the extension of human senses for delivering contextual information in an optimized way [2], [3]. For the visual sense, a difference of traditional imaging of virtual information to augmented imaging is the direct correspondence of virtual objects to reality. By exploitation of this additional and seamless information channel, the quality of information representation is strongly enhanced. This generally improves the analysis and comprehension of virtual data, but also opens new aspects for validation. This is especially true for AR visualizations of numerical simulations in living environments, where a manual comparison of results in the form of a visualization in a virtual world with reality may be tedious, and even misleading for an uninformed viewer. For instance, we make use of higher-order elements or artificial boundary conditions to better represent reality [4], [5], but for which highly specialized visualization methods would be needed to represent the data in its full fidelity [6]. When representing the results in the context of reality, the evaluation of the chosen model is simplified, and the results are represented more appropriately in the actual surroundings instead of an arbitrarily complex model thereof.

Numerical simulations in many domains can benefit from AR visualizations. Besides analysis of urban airflow and a forecast of fine dust distribution as presented in this paper, examples include noise propagation [7], urban climate simulation [8], and human crowd simulation [9]. The general feasibility of simulations in living environments and AR visualization was strongly promoted by the introduction and increasing role of Geographical Information Systems (GIS) for urban planning [10]. Their improved accuracy joined with the increasing performance of computing systems are making accurate large scale urban simulations feasible. We present the results of the joint work with the city council of Karlsruhe for simulations in an urban environment as an illustrative example setting, with focus on the advantages of mobile AR visualization of large numerical simulations. The proposed visualization method, whose development started with the *Science to Go* project, serves as a technology for solving problems of large scale data visualizations. Additionally, it also opens the path to making results of numerical simulations accessible to decision makers and to the citizen at large, both from the technical and the comprehension perspective. The general availability of smartphones and tablets equipped with GPS, cameras and graphical capabilities fulfills the technical requirements on the client side for implementing the presented visualization method. This allows for an intuitive exploration of large scale simulations. The ongoing standardization process of GIS for city modeling in the CityGML consortium [11] enables standardized simulation and visualization services for world-wide use based on the presented method in the future.

This work is an extension of [1] with a more in-depth description and discussion of the method, the application to a new scenario and simulations, as well as a description of further research into solutions for accurate visual alignment of AR visualizations using active markers.

In this paper, we first present previous papers and projects, which relate to the proposed concept. This is followed by a description of the visualization method, with details on the needed steps of pre-processing, simulation, AR visualization, interaction, and the client-server framework. The text ends with the conclusion and acknowledgments of partners and funding for the project.

II. Related Work

The Touring machine [12] was one of the first mobile solutions for AR illustrating the potential of enhancing real life images in real-time for exploration of the urban environment. The approach was to display information overlays on the camera image, which is still popular in AR applications of today [13], [14]. This concept is well suited to presenting textual or illustrative information, such as designation of points of interest, or augmented objects on top of printed markers. But this does not directly apply to immersive AR visualization of simulation results in the living environment around the viewer as presented in this paper.

The availability of dedicated graphical processing units on mobile devices has led to AR visualizations of pre-defined 3D objects [15], which have been found beneficial in laboratory setups [16]. This is the basis for visualization of 3D structures representing the results of simulations. The use of AR visualization for environmental data is presented in the HYDROSYS framework [17], which provides a method to combine measurements and simulation data with geographic information. Similar to the work presented in this paper, that framework emphasizes the need for simulation information on-site. The conceptional need for combining simulation results with data from geographic information systems is also a driving force for the CityGML project [10], which has applications to natural disaster management.

AR visualization of urban air flow phenomena in an indoor virtual reality laboratory setting based on physical mock-up building blocks is presented in [18]. The general aim of that work is similar to the one presented here, but it is focused on the interaction with objects in the visualization, and does not treat the aspect of remote visualization on mobile devices.

A related domain is that of map generation through interpolation of geographically localized, sparse data. A sophisticated algorithm for this type of problem is proposed in [19], which could conceivably also be used as a source of data for the visualization method presented in this work. In the applications presented here, the focus is on the use of data obtained through numerical simulation.

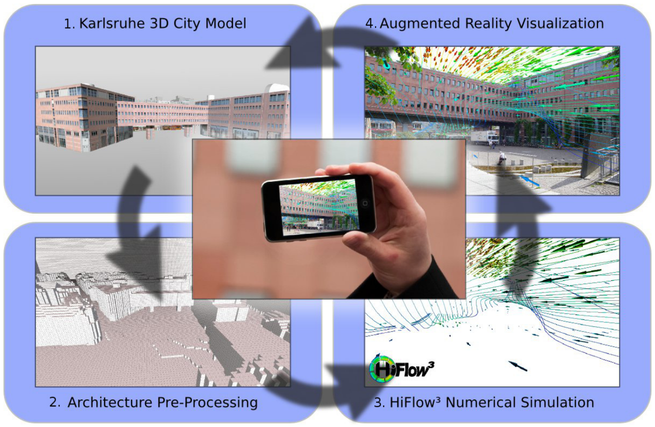
1. Karlsruhe 3D City Model
4. Augmented Reality Visualization
2. Architecture Pre-Processing
3. HiFlow³ Numerical Simulation
HiFlow³

Figure 1. Augmented Reality simulation and visualization workflow.

The simulations that are presented in this work concern wind flow and particle distribution in urban environments. This setting has previously been investigated in several works, including [20] and [21]. In contrast to those papers, we employ a simplified model, which does not include the effects of wind turbulence. This reduces the computational costs, while still delivering results that serve to illustrate the potential of the AR visualization method. It is also advantageous in cases where the outcome of numerical simulations has to be related to the real surroundings, such as for the placement of mini wind turbines in urban spaces, which does not only depend on the optimal wind conditions as discussed in [22] and [23], but also their fit into the city scape.

III. Visualization Method

The problem of creating AR visualizations of scientific data is demanding in several aspects, and its solution must necessarily combine a range of techniques from different fields, including geometric modeling, numerical simulation, computer graphics and network programming, as illustrated in Figure 1. In this section, we describe the method that we have developed to achieve this goal. First, we outline the problems that were identified in the early phases of development. Next, we describe two scenarios, which are used to illustrate the use of the method. In the remainder of the section, we provide details on various aspects of the techniques that were used, including the construction and discretization of a virtual geometry, modeling and numerical simulation, AR visualization, interface for user interaction, and a framework for distribution of the compute load.

A. Identification of Problems

To obtain a clear understanding of the steps required to create AR visualizations of scientific data, we have identified and analyzed the main problems associated with this task. As with any AR implementation, the first challenge is to construct a virtual geometry. In this work, we have focused on use cases in an exterior urban setting, but the proposed concept could also be applied in large open areas as well as inside buildings.

The next challenge is to create datasets that are suitable to visualize in the AR rendering. In this work, we are interested in displaying solutions of numerical simulations of physical phenomena, such as wind flow, noise, temperature or particle concentrations. The process to compute these solutions is largely manual: one has to determine a suitable mathematical model, formulate a precise and well-posed problem, choose an appropriate numerical method, and perform discretizations of the equations as well as the geometry. Furthermore, one must acquire the necessary input data such as material properties, boundary values and initial conditions. Ideally, all these steps would be automated, but at the present state of research, at least the steps up to and including the discretization require some human intervention.

Once a dataset has been computed for the virtual geometry, one has to combine it with the real-world geometry, based on the position and orientation of the user. The major difficulties in this context are the alignment of the virtual and real geometries, and the combination of the computed dataset and the current camera view.

AR visualization is by nature interactive, and should permit the user to control the displayed data in various ways, not only by moving the camera. Furthermore, it is not always evident how visualizations of scientific data, and its associated uncertainties, should be interpreted. An important challenge is how to present data in such a way that it can be correctly understood by non-experts.

The final problem that we identified is the need for substantial compute power, both for the numerical simulation and for visualization of the results. Although the capabilities of handheld devices is steadily increasing, the processor within a single mobile phone is not able to solve three-dimensional fluid flow problems with reasonable accuracy within acceptable time limits. Hence, a distributed architecture is needed, which allows remote access to numerical simulations on powerful hardware.

The method proposed in this work is an attempt to address all these problems. We discuss the extent to which we consider our solution successful, as well as the open problems that remain, in Sections IV and V.

B. Scenarios

In order to demonstrate the capabilities of our visualization method, we define two test scenarios, each consisting of a specific numerical simulation in a specified place. Figure 2 shows the location of these sites on a map of the city of Karlsruhe. These scenarios are primarily meant to illustrate how AR visualizations of scientific data are useful, and to provide datasets upon which the various data processing and visualization techniques can be tested. The accurate simulation of the physical processes that we have chosen is generally a difficult and time-consuming problem, which is not the main focus of this work. For this reason, the models have been simplified, and the input parameters has been chosen in such a way as to make it possible to obtain the data in a short amount of time, at the expense of accuracy and physical realism of the results. The computations performed and the simplifications that were made, are described in detail later in this section.

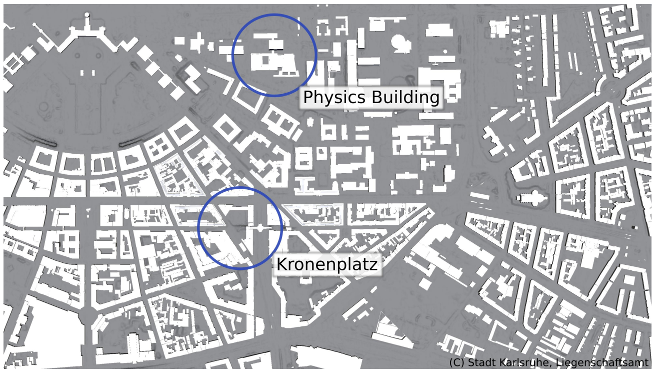

Figure 2. Map of Karlsruhe with places corresponding to scenarios.

The first scenario that we consider is wind simulation around the building that hosts the Department of Mathematics of the Karlsruhe Institute of Technology (KIT). It is located at the Kronenplatz square in Karlsruhe. We use synthetic data to determine a plausible wind velocity flow on the boundary of the domain, and solve the incompressible Navier-Stokes equations to obtain the solution in the entire domain.

The second scenario concerns the spread of fine dust particles in the vicinity of the Physics building on the campus of KIT. In a first step, we again compute a velocity field around the buildings as in the first scenario; and then solve a model for the transport of microscopic particles suspended in the air based on this velocity field.

C. Virtual Geometry

A numerical simulation can be viewed as the combination of a mathematical description of the physical phenomenon to be simulated, a numerical method to solve the problem, and a computational domain describing the space in which the simulation is performed. While the first two aspects are discussed in literature, and actively researched in computational sciences, the third aspect traditionally receives less attention for living environments. Understandably, this is due to the fact, that the effort of performing measurements of buildings is too large compared to the value of individual numerical simulations. Furthermore, the alignment with real world coordinates as needed for AR applications is an additional requirement. A solution to this problem is to derive the computational domain from other data sources, performing additional steps to convert the geometrical description to a

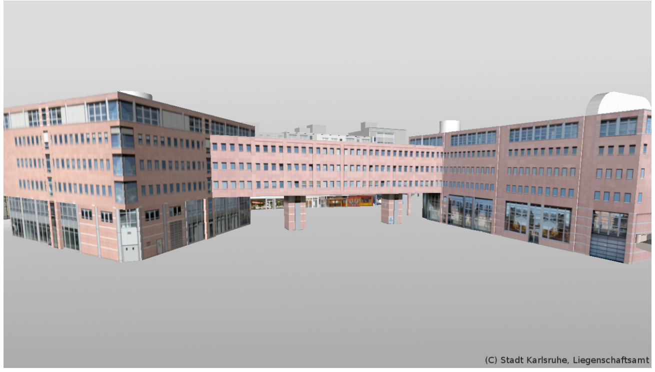

Figure 3. Photo-realistic building in the Karlsruhe 3D city model.

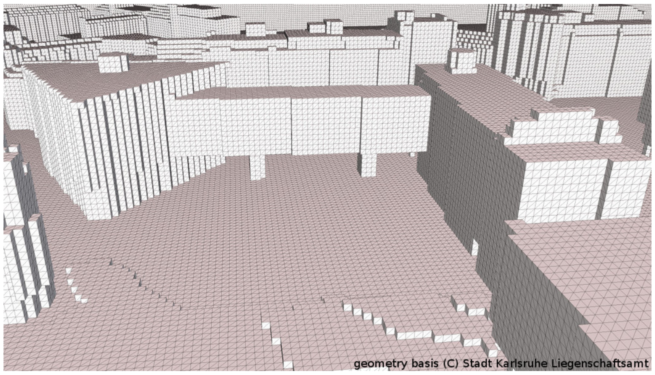

Figure 4. Computational geometry based on the Karlsruhe 3D City Model.

suitable computational domain. This approach is followed and explained in this text, based on a GIS urban model.

The project "3D-Stadtmodell Karlsruhe" [24] was started in 2002 as an improved database of geographic information to meet the demands of the local administration. It consists of several data sets of varying purpose, coverage, accuracy and detail, starting with a terrain model without buildings, and including large brick models for the cityscape, up to a photo-realistic model, as seen in Figure 3. All data sets are expressed in a global Cartesian coordinate system, such as Gauß-Krüger or Universal Transverse Mercator (UTM) coordinates, for alignment with the real world. The city model is currently progressing towards an integration into a CityGML [10] based representation.

Since none of the models were created for use by numerical simulation software, extensive pre-processing steps were necessary. In general, two or three models have to be combined to create a suitable computational domain, as seen in Figure 4. Special care was necessary to deal with model enhancements that had been made mainly for visual effects. For instance, there were closed window panes in garages facing the outside world on both sides with zero width, which are very significant for wind flow simulations around buildings. Although such irregularities could be avoided by imposing strict conditions on the city models, in general we cannot expect available city models to conform to these conditions, since they were originally created for visual planning. To avoid problems arising from these kinds of artifacts, an emphasis was put on the use of robust and efficient region growing methods that are well known from medical applications such as the realistic computational fluid dynamics simulations of the nose and lungs (see, e.g., [25], [26]).

The chosen approach approximates the geometry by discretization into voxels of pre-defined size. On the one hand, this avoids problems around very small details, that would require a high level of detail in the computational domain. This would lead to an increase of the computational effort a lot and a decrease the numerical stability, without necessarily yielding large gains in accuracy. On the other hand, the actual discrepancy between a given model and its approximation is easily controllable by the size of voxels, offering the choice between accuracy and computing time in advance.

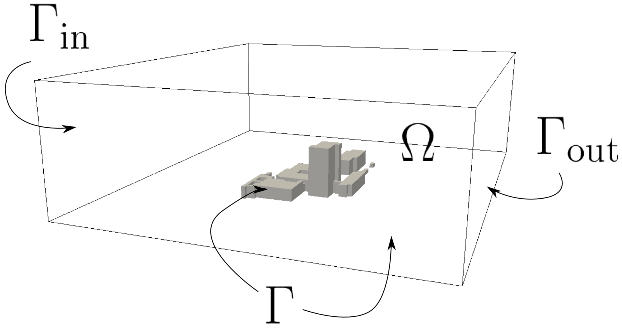

Figure 5. Schematic description of computational domain and boundary conditions for wind flow model.

Another challenge for enabling widespread use of numerical simulations in urban environments is the scarcity of highly accurate city models. This condition can be weakened to the availability of high resolution models in the main areas of interest, since widely available low accuracy models are sufficient for the necessary peripheral simulation in the surrounding area. In spite of the varying detail of the models, the very accurate geographic alignment offers the opportunity for an automated data source selection and pre-processing workflow.

D. Wind Flow Simulation

In both scenarios, we want to compute the flow of the wind around isolated buildings in the city. For this, we employ a simulation that solves a standard model based on the instationary version of the incompressible Navier-Stokes equations (see, e.g., [27]) in a sufficiently large computational domain Ω surrounding the area of interest. We apply suitable artificial boundary conditions for the assumed wind flow conditions, thereby neglecting the impact of surrounding buildings outside the domain. Since air can be considered incompressible for speeds much lower than the speed of sound, these equations provide an accurate description of the behavior of the air flow.

The model is formulated as an initial boundary value problem for a set of partial differential equations, which describe the time evolution of the velocity $\vec{u}(\vec{x},t)$ and the pressure $p(\vec{x},t)$, both of which are functions of position $\vec{x} \in \Omega$ and time t in an interval $[0,T)$. The problem is stated in (1), where the first equation is derived from the principle of conservation of momentum, and the second from that of conservation of mass. The derivation of these equations make use of the fact that air can be considered to be a Newtonian fluid.

$$\begin{aligned} \partial_t \vec{u} + (\vec{u}\cdot\nabla)\vec{u} &= -\frac{1}{\rho_F}\nabla p + \nu\Delta\vec{u}, && \text{in } \Omega\times(0,T), \\ \nabla\cdot\vec{u} &= 0, && \text{in } \Omega\times(0,T), \\ \vec{u} &= \vec{u}^{in}, && \text{in } \Gamma_{\text{in}}\times(0,T), \\ (-\mathcal{I}p + \nu\nabla\vec{u})\cdot\vec{n} &= 0, && \text{in } \Gamma_{\text{out}}\times(0,T), \\ \vec{u} &= 0, && \text{in } \Gamma\times(0,T), \\ \vec{u}(\vec{x},0) &= \vec{u}_0(\vec{x}), && \text{in } \Omega\,. \end{aligned} \tag{1}$$

Here, the parameters ρ_F and ν correspond to the density and kinematic viscosity of air, which are both assumed to be constant. Since we solve the equations on a truncated domain, the solution has to be prescribed on the boundary. Figure 5 shows a schematic overview of the boundary conditions. At the walls of buildings as well as on the ground, the velocity is set to zero, which corresponds to so-called *no-slip* boundary conditions. This part of the boundary is denoted Γ in (1). On one side of the domain, Γ_{in}, we prescribe a fixed velocity $\vec{u}^{in}$. Since this velocity is not known exactly for a given situation, we need to make an assumption about it. A common model for the general behavior of the lowest layer of the atmosphere (also called the *Prandtl layer*) is to assume that the speed grows logarithmically with the height z above ground [28], [29]. This corresponds to the following expression:

$$\vec{u}^{in}(z) = -\frac{U}{\kappa}\left(\ln\left(\frac{z}{z_0}\right)\right)\vec{n}_{\text{in}}, \tag{2}$$

where U is an estimated average wind speed, $\kappa \approx 0.4$ is the von-Kármán constant, and z_0 is a measure of the roughness, and corresponds to the height above the ground where the velocity becomes zero. The vector $\vec{n}_{\text{in}}$ is the outward unit normal on Γ_{in}. In the lack of wind profile measurements, also a simplified model with a linear profile can be considered:

$$\vec{u}^{in}(z) = -U\left(\frac{z}{z_1}\right)\vec{n}_{\text{in}}, \tag{3}$$

where U in an estimated average wind speed at height z_1. In the simulations, the second approach was adopted, and the parameters were chosen to be arbitrary, but reasonable values, which are shown in Table I. In future work, one could imagine to base the boundary values on current solutions of the lowest layers in weather forecasting models, such as the global model GME [30] or the regional model COSMO [31].

Table I
VALUES OF THE PARAMETERS USED IN THE WIND FLOW SIMULATIONS.

Parameter	Assumed value
Kinematic viscosity ν	0.001 m^2/s
Density ρ_F	1.2041 kg/m^3
Max. inflow speed U	10 m/s
Height z_1	150 m

Here, we have chosen the approach of using fixed values of the velocity on the sides (Dirichlet boundary conditions), for example to set the known wind profile [32]. This offers the chance of using an exterior flow condition on the top plane [33], which can be used to significantly reduce the required size of the computational domain. Another approach for choosing suitable conditions would be to consider a city with regularly aligned blocks and using a lid driven simulation with cyclic boundary conditions on the sides with sufficient height, as in [34].

On the remaining part of the boundary, denoted by Γ_{out}, a relation between pressure and velocity is imposed, which corresponds to an outflow. This so-called *do-nothing* condition appears naturally in the weak formulation that is used for the finite element discretization, and is easy to work with since it does not require any special treatment in the discretization.

The kinematic viscosity ν in (1) describes roughly the thickness of the fluid. It plays an important role via the *Reynolds number*, a dimensionless quantity that characterizes the behavior of the flow with respect to turbulence. It is defined as $\text{Re} = \nu^{-1}|\vec{u}|L$, where L is the characteristic length scale of the problem. When Re is large, the flow has a turbulent character, which requires highly sophisticated methods for its solution. With realistic values of $\nu \approx 10^{-5}$ m^2/s for the type of geometries and flow speeds that we are considering, Re would certainly lie in this regime. Investigations such as those described in [20] and [21] show that this type of turbulence computation is within the possibility of present simulation technology. However, to avoid the additional expense of performing such computations for this scenario, we have chosen to use a larger value of ν. The value for this and the other parameters that were used in the simulations are listed in Table I.

We discretize this mathematical model using a finite element method based on a standard weak formulation of (1). We follow the discretization approach used in [35], with Q_2/Q_1 finite elements, which yields second order accuracy for the velocity field, and first order for the pressure. The solution of the nonlinear system of equations uses the

Newton method with a GMRES linear solver to compute the corrections. The GMRES method uses preconditioning by multilevel incomplete LU factorization through the ILU++ software package described in [36]. The implementation of the simulation is based on the finite element library HiFlow^3 [37].

E. Fine Dust Simulation

For the second scenario, we simulate the spread of fine dust particles in the air. This type of computation has several important applications, which include predicting the effect of pollution (heavy metals, smog, smoke), as well as estimating the transport of naturally occurring dust and pollen, both of which can be useful for instance in city planning. At low altitudes in urban areas, the occurrence of buildings strongly limits the transport of particles, and the question of deposition of particles becomes important. In the following, we describe a mathematical model for particle transport, which is derived from the work presented in [38].

We assume a set of non-interacting, spherical particles P_i, $i = 1, ..., N$, with radii r^i and masses m^i. In the following, a superscript i denotes that a quantity is related to particle P_i. Its position $\vec{x}^i(t)$ will evolve according to its velocity $\vec{u}^i(t)$ via the ordinary differential equation (ODE):

$$\frac{\mathrm{d}\vec{x}^i}{\mathrm{d}t}(t) = \vec{u}^i(t). \tag{4}$$

The velocity of a particle P_i is the sum of the velocity $\vec{u}_F$ of the air at x^i, and a velocity $\vec{u}^i_P$ that arises due to the total external force $\vec{F}^i(t)$ acting on the particle:

$$\vec{u}^i(t) = \vec{u}_F(\vec{x}^i(t)) + \vec{u}^i_P(t). \tag{5}$$

Figure 6 shows the two contributions to the particle velocity together with the forces that are accounted for in the model. The air velocity field $\vec{u}_F$ is obtained from a computation of the wind flow, as described in III-D. In general, this wind field varies in time, but for simplification, we have assumed that it is stationary in our model. One can think of this as an average over time of the possible wind fields; although in the computations, we have simply used the instantaneous solution at an arbitrary point in time.

The second part of the velocity $\vec{u}^i_P(t)$ is computed according to Newton's second law, which can be expressed as follows:

$$m^i \frac{\mathrm{d}\vec{u}^i_P}{\mathrm{d}t}(t) = \vec{F}^i(t). \tag{6}$$

The force acting on a particle is assumed to consist of three effects:

$$\vec{F}^i(t) = \vec{F}^i_{\text{grav}} + \vec{F}^i_{\text{pres}} + \vec{F}^i_{\text{drag}}. \tag{7}$$

Here, $\vec{F}^i_{\text{grav}} = -m^i g \vec{e}_z$ is the gravitational force, with $g \approx 9.81\,\mathrm{m\cdot s^{-2}}$ the gravity of earth, and $\vec{e}_z$ the upward vertical direction vector. $\vec{F}^i_{\text{pres}} = -\frac{4\pi}{3}(r^i)^3 \nabla p_F$ is the force that the air pressure p_F exerts on the spherical particle. Finally, $\vec{F}^i_{\text{drag}}$ corresponds to the friction force, which acts on the particle as it moves in the fluid. It is given by

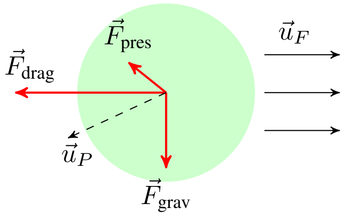

Figure 6. Schematic image of forces acting on a particle in the fine dust model.

$$\vec{F}^i_{\text{drag}} = -0.5 c^i(\mathrm{Re}^i)\rho_F A^i |\vec{u}^i_P| \vec{u}^i_P, \tag{8}$$

where c^i is the drag coefficient associated with, ρ_F the density of the fluid, and $A^i = \pi(r^i)^2$ the cross-sectional area of the particle perpendicular to the direction of motion.

The drag coefficient is determined in terms of the particle Reynolds number, which is defined as $\mathrm{Re}^i = \frac{|\vec{u}^i_P| r^i}{\nu_F}$, where nu_F is the kinematic viscosity of the fluid. An empiric law for the drag coefficient, which is known [39] to be valid for low values of Re^i is

$$c^i = \begin{cases} \frac{24}{\mathrm{Re}^i}, & \text{if } 0.0 < \mathrm{Re}_P \le 1.0, \\ \frac{24}{(\mathrm{Re}^i)^{0.646}}, & \text{if } 1.0 < \mathrm{Re}_P \le 400\,. \end{cases} \tag{9}$$

For the computation of Re^i, the kinematic viscosity and density of the fluids were chosen as $\nu_F = 1.71 \cdot 10^{-5}\mathrm{m}^2/\mathrm{s}$ and $\rho_F = 1.20\,\mathrm{kg/m}^3$, which corresponds to air at standard outside temperatures. We have further assumed for simplicity that all the particles have the same radius $r^i = 1.9 \cdot 10^{-5}\,\mathrm{m}$ and mass $m^i = 1.15 \cdot 10^{-10}\,\mathrm{kg}$: a more sophisticated method would be to assign this at random from a given distribution.

Altogether, the evolution of the particles is described by the $2N$ ODE (4) and (6), supplemented by initial conditions for $\vec{x}^i$ and $\vec{u}^i$ at $t = 0$. These conditions are typically chosen at random based on a distribution that corresponds to the specific situation at hand. For the velocity, another possible choice is to first determine the initial positions, and then to start the particles with the same velocity as the underlying fluid: $\vec{u}^i(0) = \vec{u}^i_F(x^i(0))$.

Many different methods exist for solving systems of ODE. In accordance with the wish to keep our procedure as simple as possible, we have chosen to use quite basic methods. The total time-interval $[0, T)$ is split into time steps of size Δt, and the solution is computed at the discrete times $t_n = n\Delta t$. For solving (4), the implicit Euler method is applied, which yields the following iterative method, for $n = 0, 1, \ldots, T/\Delta t$.

$$\vec{x}^i(t_{n+1}) = \vec{x}^i(t_n) + \Delta t \left(\vec{u}^i_P(t_{n+1}) + \vec{u}^i_F\left(\vec{x}^i(t_{n+1})\right)\right). \tag{10}$$

To avoid having to solve a nonlinear problem in this case, we assume that the fluid velocity varies slowly in space, and make the approximation $\vec{u}^i_F\left(\vec{x}^i(t_{n+1})\right) \approx \vec{u}^i_F\left(\vec{x}^i(t_n)\right)$, which yields the modified iteration step:

$$\vec{x}^i(t_{n+1}) = \vec{x}^i(t_n) + \Delta t \left(\vec{u}^i_P(t_{n+1}) + \vec{u}^i_F\left(\vec{x}^i(t_n)\right)\right). \tag{11}$$

This can be computed explicitly, once $u^i_P(t_{n+1})$ has been determined from the discretization of (6). Again, the implicit Euler method is used, giving the basic iteration:

$$\vec{u}^i_P(t_{n+1}) = \vec{u}^i_P(t_n) + \frac{\Delta t}{m^i}\Big(\vec{F}^i_{\text{drag}}(\vec{u}^i_P(t_{n+1})) + \vec{F}^i_{\text{pres}}(\vec{x}^i(t_{n+1})) + \vec{F}^i_{\text{grav}}\Big). \tag{12}$$

Similarly to above, it is assumed that the gradient of the fluid pressure varies slowly in space, so that the approximation $\vec{F}_{\text{pres}}(\vec{x}^i(t_{n+1})) \approx \vec{F}_{\text{pres}}(\vec{x}^i(t_n))$ can be made. The gravitational force is constant in both time and space. To treat the drag force in an accurate way, we keep the form as it is, and use a fixed point iteration to solve the resulting non-linear equation.

As was the case for the wind flow simulation, the model that we have used here has been simplified to make the computations easier, and to be able to arrive at a result with a limited effort. In particular, a more complete model would also take into account the effects of turbulence, and the resulting random variations in the particle force.

F. Augmented Reality Visualization

The problem of combining virtual objects with an image of reality is discussed in two steps. First of all, the general composition of virtual data with a photographic image is introduced, followed by approaches for the actual alignment of the virtual world with reality in the next subsection.

The visualization method is based on the accurate alignment of the viewer's position and the orientation of his camera view with the three-dimensional city model and the numerical simulation. In the setup considered here, only the graphics representing the flow field are to be embedded in the real-life image as seen in Figure 1, and therefore, the virtual city model and the computational mesh should not be visible. However, the simulation results that are covered by buildings in the city model must also be removed from the image. The approach we followed is to paint the background and city models completely in black. Therefore, the occluded simulation results are masked by the city model, which itself remains invisible, leading to a masked visualization as displayed in Figure 7. All black areas will then be treated as being transparent. Such a color-key method can be improved by rendering using an alpha-channel, but this generally requires more adaptions in the visualization software, and was not deemed necessary for these examples.

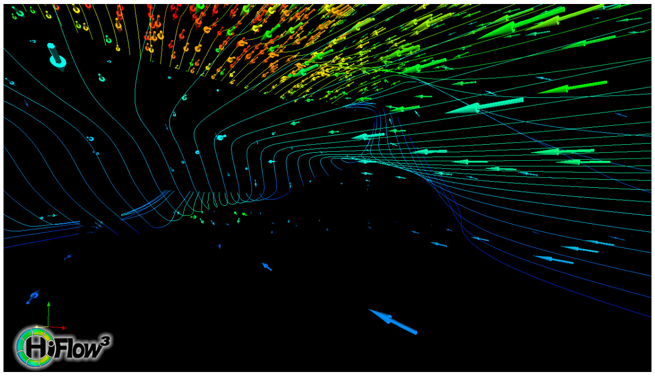

Figure 7. Masked numerical simulation visualization.

The masked visualization can then be composed onto the camera view leading to the augmented numerical simulation visualization in Figure 16, which was extended with the computational domain for illustration. The resulting image is very informative and gives insight into the simulation results. Since the displayed part of the simulation coincides with the viewer's position, the data selection is most intuitive and the full simulation can be explored by simply wandering around in the computational domain. A corresponding AR visualization for an isolated multi-component building in the Physics scenario is shown in Figure 17.

G. Interaction and User Interface

The AR visualization needs accurate positioning and orientation information. This is strongly linked to the user interface, in which the position in space and the view orientation defines the information the viewer wants to analyze. We will discuss the use of sensors of hand-held devices as a man-machine interface, its use for AR positioning and orientation, and an extension for accurate positioning and orientation using active markers.

The interaction and the user interface is crucial for usability and comprehension. The proposed model is to present the mobile device as a window to the AR and the results of the numerical simulation. This leads to challenges as outlined in [13] that can be addressed using sophisticated mathematical methods such as filtering, simulation, and parameter identification. Only the increasing computing power available in modern mobile devices such as smartphones and tablets enable the use of such costly algorithms in real-time the are necessary for responsive haptic user interfaces.

The camera view in space is defined by six parameters, the three-dimensional position and the three viewing angles. Therefore, at least six dimensions of sensor data are needed to control the user interface. Besides GPS, mobile devices of the latest generation contain spatial accelerometers as well as

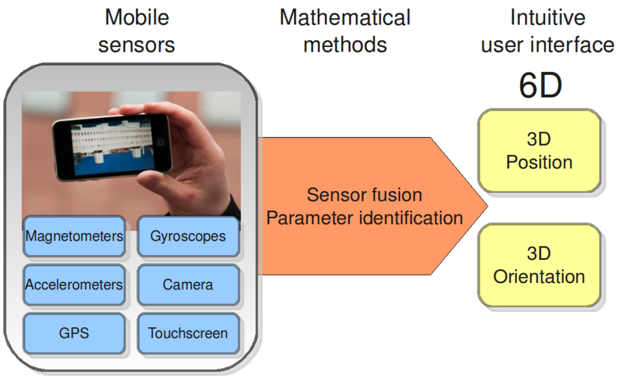

Figure 8. Mathematical methods enable intuitive user interfaces.

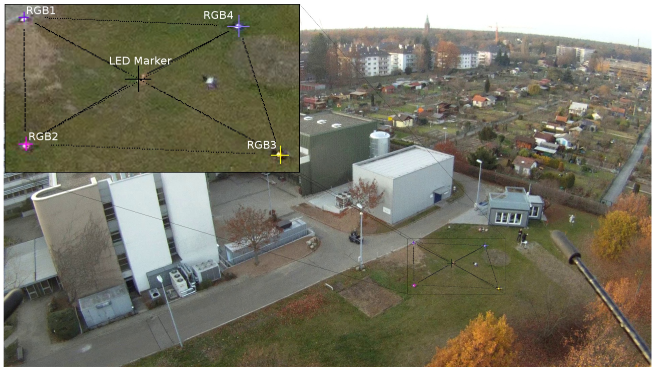

Figure 9. Marker detection from UAV camera view.

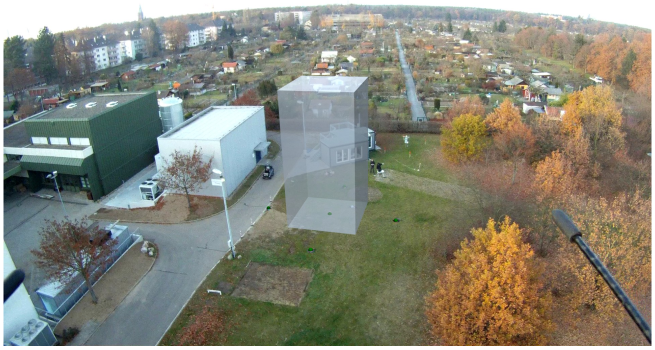

Figure 10. Augmented reality building visualization.

spatial magnetometers as a minimum. Taken together, they provide the necessary six degrees of freedom in the sensor data, enabling a new approach to an intuitive interface, which can be improved by any other additional sensors such as gyroscopes or camera based marker detection. Figure 8 illustrates that this step covers the real-time fusion of various sensor readings to gain the position and orientation information that is the basis for the AR visualization.

As evaluated in [40], the effective orientational accuracy of current mobile devices is about two to three degrees in heading, pitch and roll, and an absolute GPS position is accurate to at most 10 m. A typical horizontal field of view of a smartphone camera is 55 degrees, which means that the orientational error results in about 5 % on-screen distance error. The visual error induced by the positioning error depends on the viewing distance to the building. For 50 m distance, the angular error can add up to 16 degrees, for 100 m distance up to 8 degrees, yielding $15-30$ % on-screen distance errors. Therefore, user interaction is necessary to align the AR visualization with reality. Although the positioning errors seem to be dominant, they are less problematic once an alignment was successful, as relative GPS measurements are far more accurate.

An alternative approach is to take advantage of markers for augmented reality such as introduced by [41]. While this approach is well suited for small objects, it does not scale up to buildings. Therefore, it was proposed in [42] to introduce active markers for AR visualizations of buildings and simulations. Such markers are not only suited for ground-based AR visualizations, but also for visualizations from radio controlled multicopter aircrafts.

In Figure 9, we show the test setup from an unmanned areal vehicle (UAV) and the accurate detection of the active markers from the movie stream. This resulted in the AR visualization of a building model in Figure 10.

Interaction with a numerical simulation consists not only of moving around and changing the view; it is highly desirable to also offer access to visualization parameters, such as what quantities are displayed, the method used, and potentially to enable changing some simulation parameters. From the view of the user interface, the touchscreen interfaces of modern mobile devices offer endless possibilities for manipulation of visualization and simulation parameters. Another crucial issue is the interactivity that is offered to the user: the presented visualization needs to be updated frequently, but is limited by the available network bandwidth.

H. Client-Server Framework

In general, large scale numerical simulations and scientific visualization are resource-intensive, and require dedicated high-performance hardware. Although mobile devices are becoming increasingly powerful, there is still a large gap in performance between these devices and the clusters of thousands of servers that are typically used in scientific computing.

In order to enable interactive AR visualizations on mobile devices, we propose a client-server approach where the display and data selection is performed with a user interface on a mobile device, but the actual simulation results and visualization remains on a high-performance server infrastructure. As illustrated in Figure 11, the clients are connected to the visualization service on the servers by wireless or cellular networks, which are limited by the available bandwidth. In a direct image transport, a refresh

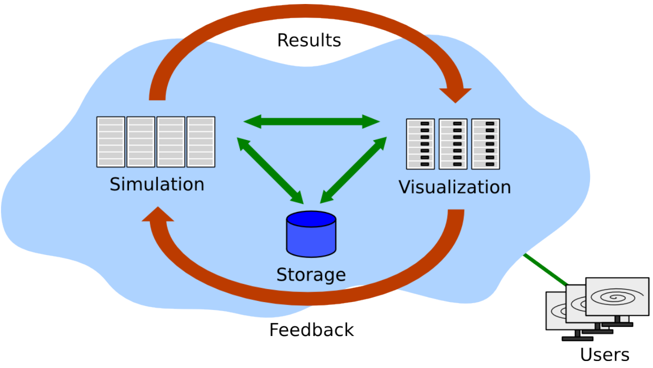

Figure 11. Interaction model.

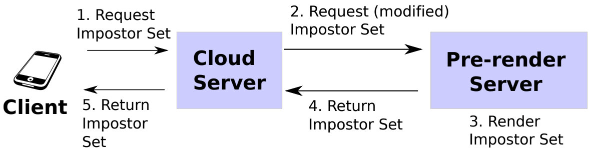

Figure 12. Schematic overview of the client-server framework.

rate of several frames per second is feasible on UMTS networks. But the interactivity is bound to latencies ranging from 100 ms to several seconds.

In computer gaming, there are similar requirements for interactivity as in scientific visualization. In [43], a platform is introduced which aims at providing 3D games even on handheld devices. They either transmit the OpenGL or DirectX commands directly to the client, or use a low-latency version of the H.264 encoder to transmit the visual information to the client. While the system aims at WLAN networks, the concept seems applicable to Long Term Evolution (LTE) mobile networks, that can provide peak bandwidths exceeding 100 Mbps in the downlink direction [44].

In AR applications, orientation and position changes are most common, and theoretically, the optimal approach would in this case be to transmit the full 3D model to the mobile client, to enable realtime interaction. But for large datasets, the mobile devices generally cannot meet the memory demands and GPU performance needed.

Therefore, the approach that we have adopted in the European Project *MobileViz* is to compute visually indistinguishable but reduced 3D models, which enable high refresh rates and low interaction latencies even when they are rendered on a mobile device. The reduced models consist of a set of impostors in the form of simple images, which are generated on the server for the current viewpoint, and then transmitted to the client, where they can be rendered at low cost. The details of this method are described in [45] and [46].

Figure 12 shows schematically how this type of rendering is embedded in our client-server framework. The server component of this framework is split into two parts. The *pre-rendering* service accepts incoming requests for visualizations of particular datasets, and generates the corresponding impostor images, possibly by using hardware dedicated to scientific visualization. The *cloud server* provides a web service, which accepts multiple concurrent incoming requests, and determines which impostors should be generated to fulfill these requests. The requests are then forwarded to the pre-rendering service. In order to keep the load on the pre-rendering server small, the cloud service caches already computed results, and determines the optimal parameters for the impostor rendering. To reduce the amount of computation, it can choose to return a slightly different view than what was requested, in order to make use of already existing data.

In order to give the user of the mobile device the possibility to interact with the visualization, and by extension also the numerical simulation, the *cloud server* will also interact with those components, to forward user requests to them via a specialized interface. Whereas a prototype implementation of the impostor-based rendering is already in place, the development of the aspects dealing with the interactivity is still on-going.

The architecture presented here can be understood in the context of *Mobile Cloud Computing*, where part of an application running on a mobile device is offloaded to a server infrastructure. This model of computing is undergoing rapid growth and offers several advantages, as described in for instance [47] and [48]. In the current work, we have partitioned the application statically between the mobile client and the cloud server. The interaction with the reduced visualization in the form of the impostors takes place on the client, and the actual compute-intensive rendering, on the server.

An alternative approach would be to employ a dynamic partitioning of the execution between server and client as suggested in [47], [48]. The decision of what part is executed where would then be determined by the capacity of the device and the quality of the network connection. A limitation to this approach is that the amount of data being visualized is often very large, and might therefore have to be kept on the cloud server.

IV. Results

In this section, we discuss the results of our tests with the presented methods.

A. Virtual Geometry

Based on 3D city models, our voxelization method is able to derive a computational domains for simulation in a robust way. We joined several data sets of various levels of detail to achieve the most accurate data basis, which was then mapped into voxels of given size. By this, the method can adapt the resulting model to the demanded accuracy, and at the same time filters out small artefacts or errors that would otherwise influence the simulation. The results are presented for two building complexes in Figures 4 and 5.

If a 3D city model is available, the presented method is automated, and delivers computational domains in a robust way. This could be improved by using a more general representation model than a voxel-based approach, but the aspects of robustness, resulting level of detail, and additional computational costs would need to be weighed against the potential benefits. A straight-forward compromise with small additional computational costs in this step, could be to use a hierarchy of voxels, such that the level of detail remains fixed, but larger areas can be covered by larger voxels, as long the numerical method and simulation software permits such selectively coarsened representations of the computational domain. This type of approach could significantly speed up the simulation.

Naturally, the method based on data from a GIS urban model will require additional consideration, as important aspects for simulation were not taken into account in the generation of the models. An example is given by thin glass panes, where both sides face the outside. This needed special treatment to prevent an air passage through this flat object where in reality, the air is blocked. Also additional information about the surface materials should be extracted from databases, to provide hints for which mathematical model should be employed on very smooth surfaces compared to rough planes.

Already in its simple form, however, our method was capable of providing usable computing domains for simulation, while leaving the world coordinate reference system intact, for later virtual reality visualization.

B. Wind Flow Simulation

We used our implementation of the simplified wind flow model described in Section III-D to generate data for the Kronenplatz and Physics scenarios. The same setup was used to treat both the case of the single isolated building in the former scenario, and the group of buildings in the latter. Visualizations with streamlines created using Paraview [49] are shown for the two scenarios in Figures 13 and 14.

For both scenarios, the results obtained are plausible, given the simplifying assumptions made for the model. The way the velocity fields are affected by the presence of buildings is qualitatively correct, which is sufficient to illustrate the functioning and utility of the AR visualization method.

In order to be appropriate for a real use case, the simulation would of course have to deliver data that reflects reality in a more accurate way. The corresponding model would have to use values of the material parameters deduced from measurements, and be modified to deal with the turbulence effects that would arise. Furthermore, the data for the boundary conditions would have to be chosen in a meaningful way. This could be done in several ways: through user input, local measurements, or, as mentioned in Section III-D, interpolation of meteorological data that is available at larger scales.

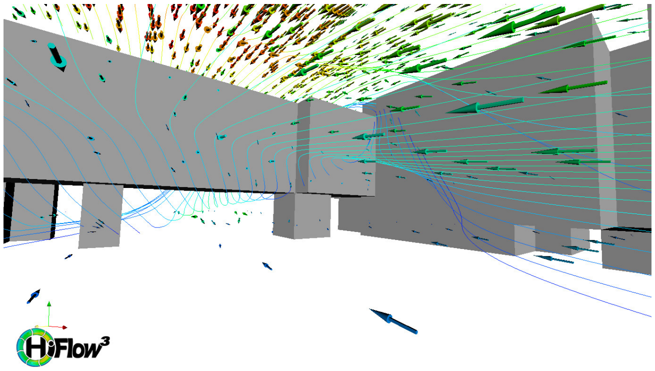

Figure 13. Visualization of computed wind flow field for the Kronenplatz scenario.

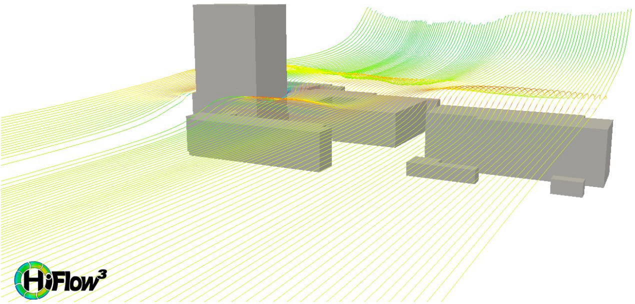

Figure 14. Visualization of computed wind flow field for the Physics buildings.

Naturally, so long as one can only obtain sparse and imprecise information about the current state of the wind, the accuracy of the simulation results will be limited. Therefore, it is important to be clear about the suitability of the simulation results in the context of specific use cases. We would expect that this type of simulation, together with the AR visualization method that we propose, find use for instance when assessing decisions in urban planning, or when evaluating risks associated with airborne pollution. In these cases, one can base the computations on sets of measurements taken over a long time, or averages thereof. Of course, the simulation cannot be expected to exceed the accuracy of the data describing the meteorological situation and the computational domain. Communicating the restrictions in accuracy to the user of the visualization remains an important open problem.

C. Particle Simulation

For the second scenario with the Physics building, we also implemented a numerical simulation for the spread of

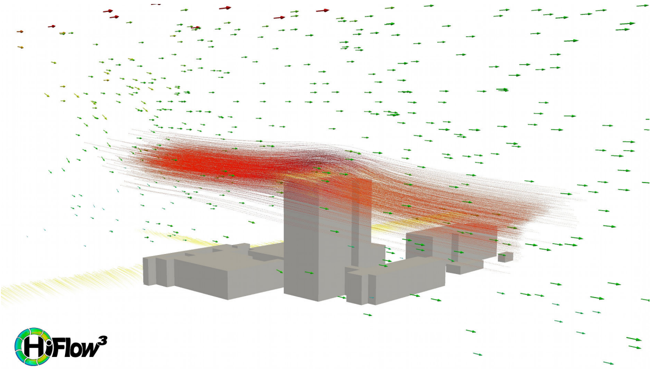

Figure 15. Visualization of fine dust particles distributed around the Physics buildings.

fine dust based on the mathematical model described in Section III-E. This simulation used the computed velocity field of the wind described in the previous section. We considered a setup with particles originating from a hypothetical chimney high up in the air, as well as along a hypothetical street passing by parallel to the buildings. The result of the simulation is shown in Figure 15, which displays the positions of the particles throughout the simulated time interval, in order to capture the entire simulation in one picture.

As was the case for the wind simulation, the results are of sufficient quality to illustrate the potential of particle simulations in conjunction with AR visualization, but cannot be considered an accurate representation of how particles would really behave in the atmosphere. The errors in the simulation are due both to the inaccuracies in the model for the wind flow and the simplifications that were made in the particle model. Additionally, the initial particle distribution is synthetically generated in this case, whereas a realistically relevant simulation would require measurements of this data.

We consider this type of simulation coupled with AR visualization to be applicable to for instance urban planning, evaluations the impact of pollution on the environment, and disaster planning.

D. AR Visualization of Wind Field and Particles

We combined the simulation data from the wind flow and particle computations for the Physics scenario into one image using the masking technique described in Section III-F. Figure 18 shows one such image, where the underlying photo was taken using a standard camera. This example illustrates how numerical results from several computations can be combined into one image, providing several pieces of information at once. The viewer gets an idea both about how the wind flows around the building, and how small particles might behave in this flow.

This image was created manually by aligning the computational geometry with the corresponding objects in the photo. The alignment is critical for the visualization, and not an easy task due to potential inaccuracies of the position and orientation information in the computational geometry, as well as additional camera parameters, such as field of view or distortions.

On a mobile device on-site, this information is available, at least approximately, and one can hope to obtain a rea-

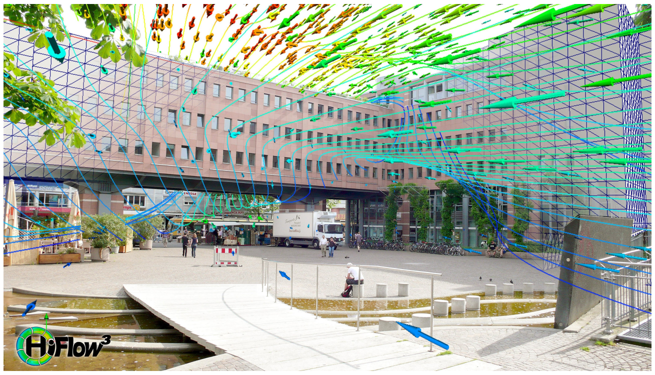

Figure 16. Enhanced Augmented Reality visualization of air flow.

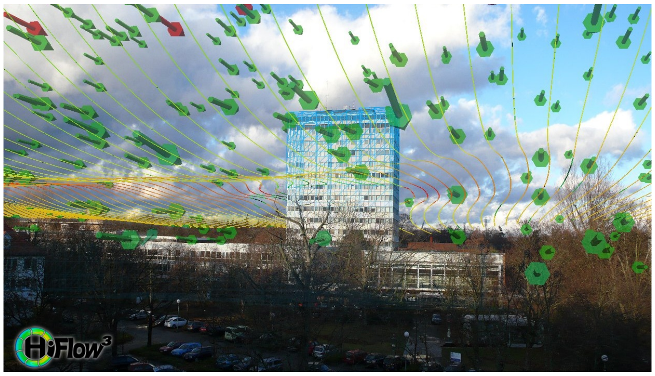

Figure 17. Augmented Reality visualization of air flow around isolated building from the Physics scenario.

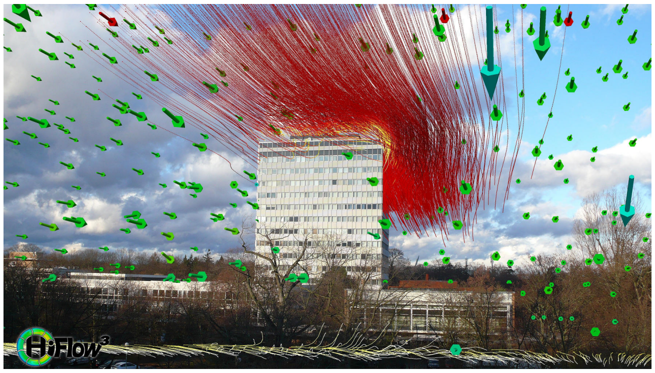

Figure 18. Augmented Reality visualization of velocity field and distributed fine dust particles around the Physics buildings, created using the masking technique.

sonably good fit between the simulated data and the camera image. Where high accuracy is required, one can compensate for errors in the position and orientation using a marker-based approach, as it was presented before.

The feasibility of such a solution was presented in Figures 9 and 10 for use with UAVs, which hints to a very promising application field of the presented visualization method in combination with flying cameras. This way, the simulation can be analyzed using the AR visualization also from above.

V. Conclusion

In this paper, we have presented a novel visualization method for large-scale scientific computing, illustrated by the examples of simulating urban air flow and fine dust distribution. The use of mobile devices opens the path to intuitive access to, and interaction with, numerical simulations that are highly comprehensible due the embedding in to the real-life camera view as AR visualizations. This is an answer to how sophisticated simulations can be made usable for non-scientists, as it is replacing the artificial and complex virtual representation of reality with a direct view of reality itself. However, this is not a complete solution, since the actual representation of simulation results needs to be understood correctly. This AR presentation aids greatly through the direct correspondence with reality, but there are other areas that require further investigation to find suitable imaging methods. For instance all numerical simulations are approximations with associated errors, both introduced by the computation itself, and by the limited accuracy of the measurements. Such uncertainties should be made obvious also to an uninformed viewer. Suitable visualization concepts for this is an open area of research.

An advantage of the method is the simplicity of selecting the data of interest and view orientation by just walking through the immersive simulation in reality and pointing the mobile device. Of course, this is limiting us to views from places, that the viewer can walk to. The general availability of UAV, combined with their ease of operation, is overcoming this issue to some extent.

We depend on the availability of a 3D city model of sufficient accuracy, in order to derive a computational domain in a robust way. All additional information that is included in the model, such as surface properties, can aid to improve the simulation quality. The introduction and adoption of a general standard such as the CityGML standard is of great help, but also offers the chance to integrate simulations into GIS databases. The work presented here, could improve the way in which such information is evaluated through AR visualizations.

The technical problem of exact alignment of real-world images with the virtual objects cannot yet be solved solely based on sensor measurements of mobile devices, but active markers can help solving this issue. This is a topic of ongoing research and development.

Another technical problem is to derive accurate information on the current conditions around the computing domain, such as the current weather conditions. Such information is available in databases from weather forecast agencies, but the resolution provided is on the order of kilometers, compared to the level of detail suitable for this visualization method that can go down the order of meters. We can expect the availability of higher resolution weather models in future, but suitable mathematical modeling for interpolation of surrounding weather conditions is a topic current research.

The proposed remote visualization method detailed in [45] and [46] is perfectly suited for displaying large stationary numerical simulations on mobile devices using the presented AR visualization method, due to its support for AR applications and its economic resource usage. By exploiting the increasing graphical performance of mobile devices, the scarce network bandwidth is utilized very efficiently. It is desirable to extend this method to instationary simulations as well, but the increased amount of data to be transmitted is limited by the traditionally small bandwidth available to mobile devices. There are promising approaches for periodic cases, but until there are new concepts for remote visualizations, increased bandwidth through new transmission standards look promising to solve this issue.

The development of the client-server framework for remote visualization enables the access to, and interaction with, large scientific datasets on mobile devices. Although still not completed, the design and prototype implementation of this framework is an important step towards realizing the goal of providing distributed visualization and simulation services over the Internet.

The presented AR visualization method is very general in its scope in the sense that it is usable for many application areas. It is expected to facilitate the use of numerical simulations by scientists as well as citizens and decision-makers. Furthermore, we are convinced that it can increase the impact and improve the communication of scientific results in interdisciplinary collaborations and to the general public.

VI. Acknowledgments

The *Karlsruhe Geometry* project is a collaboration of the Liegenschaftsamt of the city council of Karlsruhe with the Engineering Mathematics and Computing Lab (EMCL) and was supported by the KIT Competence Area for Information, Communication and Organization. The authors thank the Fraunhofer IOSB Karlsruhe and Building Lifecycle Management (BLM) at the KIT for execution of the UAV flights. The development of intuitive user interfaces for scientific applications on mobile devices was part of the *Science to Go* project, which received funding from the Apple Research & Technology Support (ARTS) programme. 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] V. Heuveline, S. Ritterbusch, and S. Ronnas, "Augmented reality for urban simulation visualization," in *Proceedings of The First International Conference on Advanced Communications and Computation INFOCOMP 2011*. Barcelona, Spain: IARIA, 2011, pp. 115–119.

[2] U. Neumann and A. Majoros, "Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance," in *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*. IEEE, 1998, pp. 4–11.

[3] R. T. Azuma *et al.*, "A survey of augmented reality," *Presence-Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[4] K. Gerdes, "A summary of infinite element formulations for exterior helmholtz problems," *Computer methods in applied mechanics and engineering*, vol. 164, no. 1, pp. 95–105, 1998.

[5] J. P. Wolf and C. Song, *Finite-element modelling of unbounded media*. Wiley Chichester, England, 1996.

[6] W. J. Schroeder, F. Bertel, M. Malaterre, D. Thompson, P. P. Pebay, R. O'Bara, and S. Tendulkar, "Methods and framework for visualizing higher-order finite elements," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 446–460, 2006.

[7] J. Kang, "Numerical modelling of the sound fields in urban streets with diffusely reflecting boundaries," *Journal of sound and vibration*, vol. 258, no. 5, pp. 793–813, 2002.

[8] A. J. Arnfield, "Two decades of urban climate research: a review of turbulence, exchanges of energy and water, and the urban heat island," *International Journal of Climatology*, vol. 23, no. 1, pp. 1–26, 2003.

[9] S. R. Musse and D. Thalmann, "Hierarchical model for real time simulation of virtual human crowds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 152–164, 2001.

[10] T. Kolbe, G. Gröger, and L. Plümer, "CityGML: Interoperable access to 3d city models," in *Geo-information for Disaster Management*, P. Oosterom, S. Zlatanova, and E. Fendel, Eds. Springer Berlin Heidelberg, 2005, pp. 883–899.

[11] T. H. Kolbe, "Representing and exchanging 3d city models with CityGML," in *Proceedings of the 3rd International Workshop on 3D Geo-Information, Lecture Notes in Geoinformation & Cartography*, J. Lee and S. Zlatanova, Eds. Seoul, Korea: Springer Verlag, 2009, p. 20.

[12] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A Touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment," in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, oct 1997, pp. 74 –81.

[13] J. B. Gotow, K. Zienkiewicz, J. White, and D. C. Schmidt, "Addressing challenges with augmented reality applications on smartphones," in *MOBILWARE*, 2010, pp. 129–143.

[14] D. Schmalstieg, T. Langlotz, and M. Billinghurst, "Augmented reality 2.0," in *Virtual Realities*, G. Brunnett, S. Coquillart, and G. Welch, Eds. Springer Vienna, 2011, pp. 13–37.

[15] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg, "Towards massively multi-user augmented reality on handheld devices," in *In Third International Conference on Pervasive Computing*, 2005.

[16] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001.

[17] A. Nurminen, E. Kruijff, and E. E. Veas, "Hydrosys - a mixed reality platform for on-site visualization of environmental data," in *W2GIS*, 2010, pp. 159–175.

[18] H. Graf, P. Santos, and A. Stork, "Augmented reality framework supporting conceptual urban planning and enhancing the awareness for environmental impact," in *Proceedings of the 2010 Spring Simulation Multiconference*. ACM, 2010, pp. 181:1–181:8.

[19] M. Hammoudeh, R. Newman, C. Dennett, and S. Mount, "Interpolation techniques for building a continuous map from discrete wireless sensor network data," *Wireless Communications and Mobile Computing*, 2011. [Online]. Available: http://dx.doi.org/10.1002/wcm.1139

[20] S. R. Hanna, M. J. Brown, F. E. Camelli, S. T. Chan, W. J. Coirier, S. Kim, O. R. Hansen, A. H. Huber, and R. M. Reynolds, "Detailed simulations of atmospheric flow and dispersion in downtown Manhattan: An application of five computational fluid dynamics models," *Bulletin of the American Meteorological Society*, vol. 87, no. 12, pp. 1713–1726, Dec 2006. [Online]. Available: http://dx.doi.org/10.1175/BAMS-87-12-1713

[21] P. Gousseau, B. Blocken, T. Stathopoulos, and G. van Heijst, "CFD simulation of near-field pollutant dispersion on a high-resolution grid: A case study by les and rans for a building group in downtown montreal," *Atmospheric Environment*, vol. 45, no. 2, pp. 428 – 438, 2011.

[22] J. S.-D. Muro, E. J. Macías, J. B. Barrero, and M. P. de la Parte, "Two-dimensional model of wind flow on buildings to optimize the implementation of mini wind turbines in urban spaces," in *International Conference on Renewable Energies and Power Quality*, 2010.

[23] F. Balduzzi, A. Bianchini, and L. Ferrari, "Microeolic turbines in the built environment: Influence of the installation site on the potential energy yield," *Renewable Energy*, vol. 45, pp. 163 – 174, 2012.

[24] T. Hauenstein, "Das 3D-Stadtmodell Karlsruhe," in *INTERGEO*, 2009. [Online]. Available: http://www.intergeo.de/archiv/2009/Hauenstein.pdf 29.7.2011

[25] M. J. Krause, “Fluid flow simulation and optimisation with lattice boltzmann methods on high performance computers: Application to the human respiratory system,” Ph.D. dissertation, Karlsruhe Institute of Technology (KIT), 2010.

[26] K. Inthavong, J. Wen, J. Tu, and Z. Tian, “From CT scans to CFD modelling - fluid and heat transfer in a realistic human nasal cavity,” *Engineering Applications of Computational Fluid Mechanics*, vol. 3, no. 3, pp. 321–335, 2009.

[27] J. H. Spurk and N. Aksel, *Fluid Mechanics*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2008.

[28] H. Kraus, *Die Atmosphäre der Erde: Eine Einführung in die Meteorologie*. Springer Berlin Heidelberg, 2004.

[29] D. Etling, *Theoretische Meteorologie: Eine Einführung*. Springer Berlin Heidelberg, 2008.

[30] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner, “The operational global icosahedral-hexagonal gridpoint model GME: Description and high-resolution tests,” *Monthly Weather Review*, vol. 130, no. 2, pp. 319–338, 2002.

[31] “Core documentation of the COSMO-model,” http://www.cosmo-model.org/content/model/documentation/core/default.htm (Accessed 2013-06-09).

[32] I. Waltschläger, “Randbedingungen zur Windsimulation im Stadtgebiet,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2011.

[33] V. Heuveline and P. Wittwer, “Adaptive boundary conditions for exterior stationary flows in three dimensions,” *Journal of Mathematical Fluid Mechanics*, vol. 12, no. 4, pp. 554–575, 2009.

[34] P. He, T. Katayama, T. Hayashi, J. Tsutsumi, J. Tanimoto, and I. Hosooka, “Numerical simulation of air flow in an urban area with regularly aligned blocks,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 67-68, pp. 281 – 291, 1997.

[35] V. John, G. Matthies, and J. Rang, “A comparison of time-discretization/linearization approaches for the incompressible Navier-Stokes equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44/47, pp. 5995 – 6010, 2006.

[36] J. Mayer, “A multilevel Crout ILU preconditioner with pivoting and row permutation,” *Numerical Linear Algebra with Applications*, vol. 14, no. 10, pp. 771–789, 2007. [Online]. Available: http://dx.doi.org/10.1002/nla.554

[37] H. Anzt, W. Augustin, M. Baumann, T. Gengenbach, T. Hahn, A. Helfrich-Schkarbanenko, V. Heuveline, E. Ketelaer, D. Lukarski, A. Nestler, S. Ritterbusch, S. Ronnas, M. Schick, M. Schmidtobreick, C. Subramanian, J.-P. Weiss, F. Wilhelm, and M. Wlotzka, “HiFlow3: A hardware-aware parallel finite element package,” in *Tools for High Performance Computing 2011*, H. Brunst, M. S. Müller, W. E. Nagel, and M. M. Resch, Eds. Springer Berlin Heidelberg, 2012, pp. 139–151.

[38] T. Gengenbach, “Numerical simulation of particle deposition in the human lung,” Ph.D. dissertation, Karlsruhe Institute of Technology, 2012.

[39] J. K. Comer, C. Kleinstreuer, and C. S. Kim, “Flow structures and particle deposition patterns in double-bifurcation airway models. Part 2. Aerosol transport and deposition,” *Journal of Fluid Mechanics*, vol. 435, pp. 55–80, 4 2001.

[40] M. K. Kirchhoefer, J. H. Chandler, and R. Wackrow, “Cultural heritage recording utilising low-cost close-range photogrammetry,” in *Proceedings of CIPA 23rd International Symposium*, 2011.

[41] H. Kato and M. Billinghurst, “Marker tracking and hmd calibration for a video-based augmented reality conferencing system,” *2nd IEEE and ACM International Workshop on Augmented Reality*, pp. 85–94, 1999.

[42] V. Koch, S. Ritterbusch, A. Kopmann, M. Mueller, T. Habel, and P. von Both, “Flying augmented reality,” in *Proceedings of the 29th eCAADe conference*, Ljubljana, Slovenia, 2011.

[43] A. Jurgelionis, P. Fechteler, P. Eisert, F. Bellotti, H. David, J. P. Laulajainen, R. Carmichael, V. Poulopoulos, A. Laikari, P. Peraelae, A. D. Gloria, and C. Bouras, “Platform for distributed 3d gaming,” *International Journal of Computer Games Technology*, vol. 2009, p. 15, 2009.

[44] E. Dahlman, H. Ekström, A. Furuskar, Y. Jading, J. B. Karlsson, M. Lundevall, and S. Parkvall, “The 3G long-term evolution - radio interface concepts and performance evaluation,” in *IEEE 63rd Vehicular Technology Conference*, vol. 1, 2006, pp. 137–141.

[45] A. Helfrich-Schkarbanenko, V. Heuveline, R. Reiner, and S. Ritterbusch, “Bandwidth-efficient parallel visualization for mobile devices,” in *The Second International Conference on Advanced Communications and Computation*. IARIA, 2012, pp. 106–112.

[46] V. Heuveline, M. Baumann, S. Ritterbusch, and R. Reiner, “Method and system for scene visualization,” Feb. 27 2013, WO Patent 2,013,026,719.

[47] D. Kovachev and R. Klamma, “Beyond the client-server architectures: A survey of mobile cloud techniques,” in *1st IEEE International Conference on Communications in China Workshops (ICCC)*, 2012, pp. 20–25.

[48] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[49] “ParaView - Open Source Scientific Visualization,” http://www.paraview.org/ (Accessed 2013-06-09).