VI International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2015
B. Schrefler, E. Oñate and M. Papadrakakis(Eds)

ON IN-SITU VISUALIZATION FOR STRONGLY COUPLED PARTITIONED FLUID-STRUCTURE INTERACTION

OLIVER FERNANDES*, DAVID S. BLOM†, STEFFEN FREY*, ALEXANDER H. VAN ZUIJLEN†, HESTER BIJL† AND THOMAS ERTL*

*VISUS
University of Stuttgart
Allmandring 19, 70569 Stuttgart, Germany
e-mail: info@visus.uni-stuttgart.de, web page: http://www.visus.uni-stuttgart.de

†Delft University of Technology, Faculty of Aerospace Engineering
Kluyverweg 2, 2629HT Delft, The Netherlands
e-mail: d.s.blom@tudelft.nl, web page: http://www.lr.tudelft.nl

Key words: in-situ visualization, fluid-structure interaction, multi-physics

Abstract. We present an integrated in-situ visualization approach for partitioned multi-physics simulation of fluid-structure interaction. The simulation itself is treated as a black box and only the information at the fluid-structure interface is considered, and communicated between the fluid and solid solvers with a separate coupling tool. The visualization of the interface data is performed in conjunction with the fluid solver. Furthermore, we present new visualization techniques for the analysis of the interrelation of the two solvers , with emphasis on the involved error due to discretization in space and time and the reconstruction. Our visualization approach also enables the investigation of these errors with respect of their mutual influence on the two simulation codes and their space-time discretization. For efficient interactive visualization, we employ the concept of explorable spatiotemporal images, which also enables finite-time temporal navigation in an in-situ context. We demonstrate our overall approach and its utility by means of a fluid-structure simulation using OpenFOAM that is coupled by the preCICE software layer.

1 INTRODUCTION

Scientific computing and parallel visualization are at the cusp of the new era of exascale computing. In-situ visualization plays a key role in this transition, since the involved data rates are expected to be beyond the capabilities of future storage systems [23]. A central difficulty with in-situ visualization in parallel computing is to conform to bandwidth constraints. On the one hand, parallel scientific computing requires very high bandwidth,

e.g., for the communication between different simulation processes, on the other hand, high-resolution visualization at high frame rates is a prerequisite for successful research and development. While many simulations are intrinsically global and require substantial parallelization traffic, there is large potential for optimization in the visualization stage.

Since the available bandwidth typically does not allow for the transmission of the entire data stream to dedicated visualization nodes, the data reduction has to take place directly at the simulation nodes. In this paper, we present an approach to minimize the bandwidth load exerted by producing a view-dependent reduced representation of the data, before sending it to the visualization nodes.

For the scalar fields associated with the fluid, a common approach is to generate images using volume raycasting. In more detail, rays are sent from the viewpoint through the volume (typically one per pixel in the final image). Along each ray, scalar sample values are taken, color and opacity are assigned to these values by means of a so-called transfer function, and the final color of a ray (or a pixel) is generated by compositing these color values (e.g., [16]). In a coupled fluid-structure simulation, several data sets are available for analysis, and need to be handled by the visualization. To be able to get a concise view of the phenomena arising from interactions between the fluid and the structure, data from both solvers need to be integrated into the same visualization. This provides the possibility to better explore the mutual influence of the two simulation domains.

In this paper, we introduce an integrated approach for the in-situ visualization of both the volume stemming from the fluid solver and the structure surface. We then calculate the deviation introduced by the relaxation step and use it to visualize the the magnitude of the displacement on the representation generated for the structure. This enables the examination of how the quasi-Newton step impacts the calculation of the next time step for the different locations of the fluid-structure interface.

2 RELATED WORK

In case an incompressible flow is considered, sub-cycling of the fluid and solid solvers is necessary in order to minimize the introduced error at the fluid-structure interface due to the partitioning. Different coupling techniques can be applied, such as Aitken's method [19], vector extrapolation [20], Interface-GMRES(R) [27, 26] , and the interface quasi-Newton inverse least squares (IQN-ILS) technique [9, 8]. The IQN-ILS technique [9] is an efficient [10, 8] and robust black-box coupling algorithm for which convergence theorems are available in [17], and is consequently used in this contribution.

For in-situ visualization, typically reduced representations of the original data are generated, usually resulting in both reduced cost for storage/transfer as well as for rendering in comparison to the original. For this purpose, image-based or view-dependent techniques are often used as they allow for high compression rates while still maintaining high quality for a certain range of view configurations. Image-based rendering infers new images from existing ones, e.g., with changed lighting or camera configuration [36]. A number of techniques has been proposed to construct such different representations from

multiple views, like view-dependent texture maps [7], warping [24], light fields [21], etc. Other techniques use multiple images to synthesize new surface-based views of volume data (e.g., [4, 3]). Such techniques allow the adaptation of color and lighting parameters [18], or transfer functions [35].

Shade et al. [30] introduced LDIs (Layered Depth Images) that represent one camera view with multiple pixels along each line of sight. Reconstruction of geometric surfaces from LDIs has been discussed by Frey et al. [14]. In volume rendering, layer-based representations have been used to defer operations such as lighting and volume classification [29]. Such representations have also proven effective to cache results [22] or certain volumetric properties along view rays that can be later reused for efficient transfer function exploration. Also for deferred transfer function exploration, Tikhanova et al. [32] convert a small number of volume renderings to a multi-layered image representation. In another work, [33] use an intermediate volume data representation which encodes the distribution of samples along each ray. Shareef et al. [31] use image-based modeling to render unstructured grids based on parallel sampling rays and 2D texture slicing.

Frey et al. [15] propose Volumetric Depth Images (VDI) for view-dependent volume visualization as an extension of the Layered Depth Image (LDI) approach. Instead of only saving for each view ray of one camera configuration the depth and color values for a set of surfaces, as in LDIs, VDIs store so-called supersegments, each consisting of a depth range as well as composited color and opacity. On this basis, Fernandes et al. [13] introduce space-time VDIs (STVDIs) that provide an efficient space-time representation of VDI streams by exploiting inter-ray and inter-frame coherence, while still maintaining spatiotemporal flexibility. STVDIs provide the technical foundation for the approach proposed in the paper at hand and is outlined in more detail in Sec. 4.

3 SIMULATION SETUP

3.1 Fluid solver

The flow is governed by the incompressible Navier-Stokes equations. Therefore, the governing equations for the fluid are given by the balance of momentum

$$\rho^f \frac{\partial \boldsymbol{v}^f}{\partial t} + \rho^f \left(\nabla \boldsymbol{v}^f\right) \boldsymbol{v}^f = \nabla \cdot \boldsymbol{\sigma}^f, \tag{1}$$

and the balance of mass, which reduces to a divergence free constraint on the velocity field due to incompressibility:

$$\nabla \cdot \boldsymbol{v}^f = 0 \qquad \text{in } \Omega^f, \tag{2}$$

where the velocity field is denoted by $\boldsymbol{v}^f$, the pressure field is denoted by p^f, and ρ^f represents the density. A Newtonian fluid is considered, which leads to the constitutive equation for the stress tensor $\boldsymbol{\sigma}^f$

$$\boldsymbol{\sigma}^f = -p^f \boldsymbol{I} + \rho^f \nu^f \left(\nabla \boldsymbol{v}^f + \nabla \boldsymbol{v}^{f^T}\right), \tag{3}$$

for a given kinematic viscosity ν^f.

The foam-extend-3.1 package[1] is used for the simulations, a fork of the well known OpenFOAM package[2]. A coupled solution algorithm [5] is employed, instead of the well known PISO pressure-velocity coupling technique. Here, the continuity and momentum equation are solved in a fully coupled implicit manner, instead of a segregated approach. A second order backward differencing scheme (BDF2) is used to integrate the governing equations in time. The fluid mesh is deformed with radial basis function interpolation [6].

3.2 Structure solver

The configuration of the structure domain is described by the displacement $\boldsymbol{u}^s$. An elastic and compressible structure is assumed, and the governing equation is given by the balance of momentum

$$\rho^s \frac{\partial \boldsymbol{v}^s}{\partial t} + \rho^s \left(\nabla \boldsymbol{v}^s\right) \boldsymbol{v}^s = \nabla \cdot \boldsymbol{\sigma}^s + \rho^s \boldsymbol{g} \qquad \text{in } \Omega^s. \tag{4}$$

Equation (4) is modified to use the total Lagrangian description, i.e. with respect to the initial reference state Γ^s, resulting in

$$\rho^s \frac{\partial^2 \boldsymbol{u}^s}{\partial t^2} = \nabla \cdot \left(J \boldsymbol{\sigma}^s \boldsymbol{F}^{-T}\right) + \rho^s \boldsymbol{g} \qquad \text{in } \Omega^s, \tag{5}$$

where the deformation gradient tensor $\boldsymbol{F}$ is defined as $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}^s$, and the Jacobian J is the determinant of the deformation gradient tensor $\boldsymbol{F}$. By applying the constitutive law for the St. Venant-Kirchhoff material, the Cauchy stress tensor $\boldsymbol{\sigma}^s$ is found by applying

$$\boldsymbol{\sigma}^s = \frac{1}{J} \boldsymbol{F} \left(\lambda^s \left(\operatorname{tr} \boldsymbol{E}\right) \boldsymbol{I} + 2\mu^s \boldsymbol{E}\right) \boldsymbol{F}^T, \tag{6}$$

with $\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I}\right)$, and the shear modulus μ^s [1].

The structure solver is also implemented within the foam-extend-3.1 framework for ease of implementation. Therefore, a finite volume discretization is used instead of a finite element approach.

3.3 Fluid-structure interaction

With regards to the fluid-structure interaction problem, the fluid solver and solid solver are considered as black boxes. In other words, only the input and output information is accessible. Whether a compressible or incompressible, viscous or inviscid flow is considered does not influence the used methodology described in this paper. Also, different models

[1] `http://www.extend-project.de/`

[2] `http://www.openfoam.org/`

for the solid domain can be applied, since only the input and output information from the fluid solver and solid solver is considered to be accessible.

Therefore, at each time step the response of the fluid solver F_f is defined as

$$\boldsymbol{y} = F_f\left(\boldsymbol{x}\right), \tag{7}$$

where $\boldsymbol{x}$ denotes the displacement of the fluid-structure interface, and $\boldsymbol{y}$ denotes the force acting on the fluid-structure interface. The response of the structure solver F_s is consequently defined as

$$\boldsymbol{x} = F_s\left(\boldsymbol{y}\right). \tag{8}$$

Typically, at every time step the fixed point equation

$$\boldsymbol{x} = F_s \circ F_f\left(\boldsymbol{x}\right) \tag{9}$$

must be satisfied, which can also be written as the interface residual $\boldsymbol{R}$

$$\boldsymbol{R}\left(\boldsymbol{x}\right) = F_s \circ F_f\left(\boldsymbol{x}\right) - \boldsymbol{x}, \tag{10}$$

which is solved with a minimization or optimization procedure aimed to find the optimal solution $\boldsymbol{x}^\star$ such that

$$\boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}} \left|\left|\boldsymbol{R}\left(\boldsymbol{x}\right)\right|\right|_2. \tag{11}$$

At the fluid-structure interaction $\Gamma^{f\,s}$, the balance of stresses is enforced through

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{\sigma}^s \boldsymbol{n} \qquad \text{on } \Gamma^{f\,s}, \tag{12}$$

with the unit vector $\boldsymbol{n}$ normal to the fluid-structure interface $\Gamma^{f\,s}$, and the stress tensors $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$. Also, the no-slip condition is imposed at the fluid-structure interface so that the velocities $\boldsymbol{v}^f$ and $\boldsymbol{v}^s$ must be equal:

$$\boldsymbol{v}^f = \boldsymbol{v}^s \qquad \text{on } \Gamma^{f\,s}. \tag{13}$$

As mentioned in the introduction, standard approaches used to solve the strongly coupled fluid-structure interaction problem are the Gauss-Seidel method [12], fixed under-relaxation [2], Aitken under-relaxation [28], and the IQN-ILS method [9, 11]. Here, the IQN-ILS method is used to solve the interface problem.

4 IN-SITU VISUALIZATION OF COUPLED FLUID-STRUCTURE INTERACTION

In this paper, we use a typical in-situ visualization setup consisting of a parallel simulation environment with integrated visualization. The fluid simulation nodes generate the data, employing domain decomposition for parallel execution. On each compute node a reduced representation of the density scalar field is produced and prepared to be sent to the visualization nodes. Finally, the visualization nodes generate images from these reduced representations interactively upon user request. In this section, we first briefly outline the underlying VDI and STVDI [15, 13] (Sec. 4.1), and on this basis outline our approach for the visualization of interaction properties (Sec. 4.2).

4.1 Space-Time Volumetric Depth Images

A volumetric depth image (VDI) is a representation of a three-dimensional scalar dataset after applying a color transfer function [15]. It is generated internally from the data volume as viewed from a specific vantage point, but can be actually explored in full 3D, for all parts of the simulation data which were in the original view frustum. Being generated from a specific view point allows the user to choose more intuitively which parts of the volume he wants to investigate, and restrict data retrieval to the chosen subset.

Volumetric Depth Images Data Structure. A VDI is produced by ray casting the dataset along rays aimed at the region of interest within the dataset. In the following, we assume that rays are uniformly seeded on a two dimensional grid with the size $X \times Y$. Along each ray r_{xy}, where xy denotes its position on the regular grid, a number of Z samples are taken (typically spaced equidistantly), from front to back. The color transfer function is then applied to associate a scalar value with a user-defined color and opacity. Strictly speaking, a sample depicts the value belonging to a value of a distinct point in space in the volume. However, in the context of raycasting and a limited sampling rate, it can also be seen as representing a certain volume around this point. Conceptually, this is the extension of a pixel in image space to object space.

Storing each segment individually would produce a resampled representation R of the size $X \times Y \times Z$ for the volume covered by the rays. Instead, to reduce the size of such a representation, samples having similar color are merged along a ray into so-called supersegments. This can be done efficiently on-the-fly during while tracing the ray along Z. For these supersegments, the starting and ending parameter value on the ray r_{xy} are stored, as well as the merged color. This produces a list L_{xy} of supersegments for each ray, each with variable length, depending on how the algorithm was able to find mergeable samples. As a result, the size of the VDI representation R has been reduced considerably with respect to the original data, with only minimal loss (depending on the similarity metric), to a set of $X \times Y$ lists each containing $|L_{xy}|$ elements.

Space-Time Clustering. VDIs already reduce the data required to represent the simulation data of interest considerably. This can further be improved upon by exploiting coherency of data across rays, and if considering additional consecutive timesteps, even time. To achieve this, the supersegments for all rays are grouped to form so called space-time regions. Similar to the merging of segments into supersegments for VDIs, this is done by comparing the color of neighboring supersegments ,where neighboring means in adjacent rays or the same ray from an adjacent timestep. In detail, adjacent supersegments that are considered similar by a user defined color metric are assigned the same region id. A representative average color is calculated for an entire region and used in the following.

At this point, already a huge data reduction has been achieved, since the color data, originally stored per supersegment, only needs to be stored per region. Note that the loss in quality due to this can be controlled by the user by changing the parameters in the similarity metric. To also reduce the data which needs to be stored to represent

the locations of supersegments, we employ an algorithm which recalculates and sorts the geometrical data such that it can efficiently be encoded to a stream by an entropy encoder such as the zlib scheme. For more details on this algorithm, refer to the related work [13].

Rendering. On the visualization node, the stream is again decoded using the reverse process explained in Sec. 4.1. All supersegments in a region receive the region's color. The reconstructed VDI representation is now used to define frustums. For each supersegment a frustum is constructed by using the starting and ending point in depth along the generating ray, and using the distance between rays as width and height. These frustums then cover the entire original viewing volume, and are a good approximation to the original simulation data. Unlike a ray cast, these new representation can be built from comparably sparse data and be rendered at much higher speeds (see [15] for details).

4.2 Structural Data and Visualization of Interaction Properties

Many nodes computing the fluid domain solution also produce data for the structure, in form of their boundary. This boundary can be directly converted into a renderable mesh. Being only a two dimensional quantity described as a set of points and and a connecting topology, the structure surface description needs far less data to be transferred, and hence does not any special processing to reduce bandwidth load. To display information concerning the interaction at the coupling boundary of the fluid and the structure solver, we use the fluid solver boundary to define a surface representation of the structure.
On this surface, boundary related data can be displayed, which becomes interesting if the data has a non-uniform distribution on the 2D manifold of the structure surface.

As an example of a data varying across the surface, the deformation of the structure could be examined. The deformation consists of several factors involving both solvers and the coupling mechanism. With many different factors involved, approximations have to be made to accommodate both the numerics and the physics.
As mentioned in Sec. 3.3, a coupling scheme needs to be applied to the structure solvers solution, in order to be able to simulate an incompressible flow without suffering from instabilities due to the added mass effect.
The displacement introduced has an impact on the fluid solvers next iteration. To judge the deviation introduced by relaxation, we now consider the displacement of the surface before and after the relaxation step, and display the magnitude of the displacement vector for the surface of the structure.

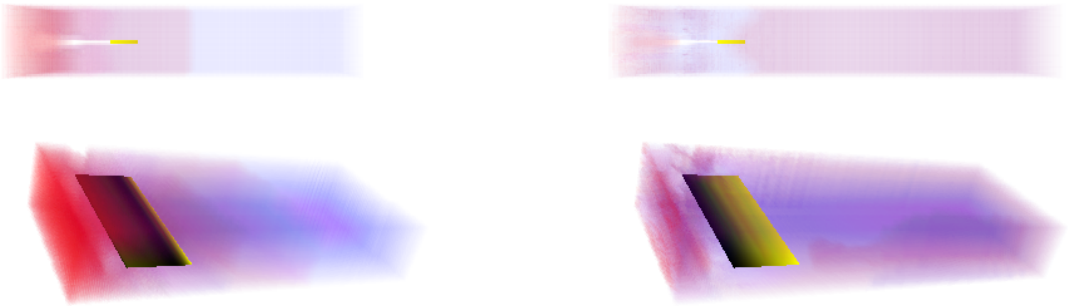
5 RESULTS

We demonstrate the integration of our technique in a distributed simulation setup, as already outlined briefly in 4 and 4.2.

The test case, originally proposed in [34], consists of a two-dimensional incompressible laminar flow around a fixed cylinder with an attached flexible cantilever. In [34], three different scenarios are presented with different fluid to structure density ratios. Here, the

fluid to structure ratio is set to $\rho^f/\rho^s = 1$, resulting in a strong coupling between the fluid and the structure. The reader is referred to [34] for further details on this fluid-structure interaction problem. An overview of the geometry is given by Figure 3.

In detail, the simulation outputs new time steps of the simulation in certain intervals (e.g., every 1000 time steps). This output is then directly fed to the VDI generation and spatiotemporal encoding/compression pipeline. This condensed representation is then communicated to our visualization node, that generates images from this representation. The table 2 shows the compression ratios determined during the simulation using the space-time VDI method. A good compression ratio of around 20% was achieved on average. The data sizes are compared to the output as written out for restarts by the solver.



(a) Timestep 0.1 (b) Timestep 3.3

Figure 1: Volume rendering of the pressure in cylinder flap testcase. The yellow color on the flap shows the distribution of the correction made by relaxation.

For the evaluation of the deviation visualization, we compared the output for the structure including the deviation, using several timesteps. The static parts of the structure (the cylinder) have not been displayed to reduce visual clutter. One can clearly see the spatial distrubution across the moveable surface, as shown in Fig. 1.

Timestep	Size (kBytes)	Ratio	CPU Time (s)
0.1	22	6.7%	4.2
1.4	53	16.3%	4.3
3.3	63	19.2%	4.3

Figure 2: The STVDI algorithm achieves a good compression ratio for different timesteps (simulation time). The timings refer to the slowest process performing the algorithm for a given timestep. Original data size: 326kByte.

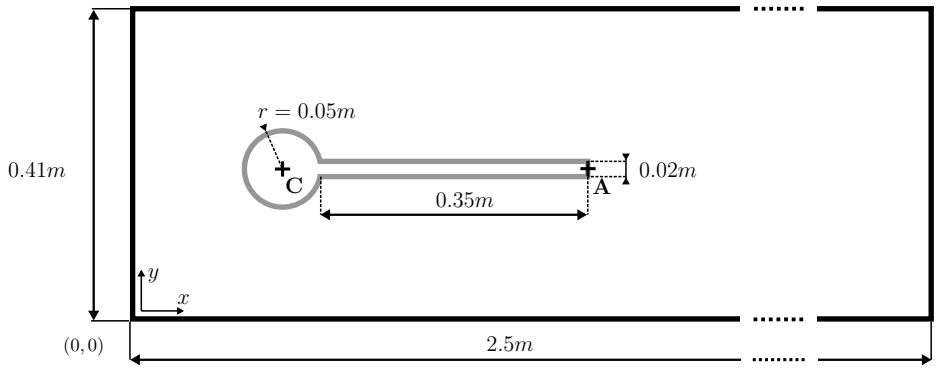


Figure 3: Geometry of the cylinder with an attached flap benchmark originally proposed in [34]. The figure shows a channel flow with a fixed cylinder and a flexible flap attached to the cylinder [25].

6 CONCLUSION

In this paper, we presented an integrated in-situ visualization approach for partitioned multi-physics simulation of fluid-structure interaction. In this context, we introduced new visualization techniques for the analysis of the interrelation of the two solvers, with emphasis on the involved error due to discretization in space and time. We showed that considerable compression can be gained by clustering in space-time and rearranging data for entropy encoding. For this, we employed the concept of explorable spatiotemporal images. Our visualization approach also enables the investigation of errors on the coupling interface due to a quasi-Newton relaxation step. We evaluated our overall approach with a fluid-structure simulation using an OpenFOAM implementation which allows for coupling by the preCICE software layer.

This method can also be extended to other metrics arising in the context of coupled systems, for example displaying coupling-layer internal interpolation error as part of the fluid-structure visualization. Further, the internal properties of the structure itself, as calculated in the structure solver, could be included to achieve a concise visualization.

REFERENCES

[1] P. Cardiff, A. Karač, and A. Ivanković. A large strain finite volume method for orthotropic bodies with general material orientations. *Computer Methods in Applied Mechanics and Engineering*, 268:318–335, January 2014.

[2] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering Computations*, 13(6):4–30, 1996.

[3] Baoquan Chen, Arie Kaufman, and Qingyu Tang. Image-based rendering of surfaces from volume data. In *Eurographics conference on Volume Graphics*, VG'01, pages 281–300. Eurographics Association, 2001.

[4] Jae-Jeong Choi and Yeong-Gil Shin. Efficient image-based rendering of volume data. In *Computer Graphics and Applications, 1998. Pacific Graphics '98. Sixth Pacific Conference on*, pages 70–78, 226, 1998.

[5] M. Darwish, I. Sraj, and F. Moukalled. A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics*, 228(1):180–201, 2009.

[6] A. de Boer, A. H. van Zuijlen, and H. Bijl. *Radial Basis Functions for Interface Interpolation and Mesh deformation*, volume 71 of *Lecture Notes in Computational Science and Engineering*, chapter 6, pages 143–178. Springer Berlin Heidelberg, 2010.

[7] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 11–20, 1996.

[8] J. Degroote. Partitioned Simulation of Fluid-Structure Interaction. *Archives of Computational Methods in Engineering*, 20(3):185–238, September 2013.

[9] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers and Structures*, 87(11-12):793–801, June 2009.

[10] J. Degroote and J. Vierendeels. Multi-solver algorithms for the partitioned simulation of fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2195–2210, June 2011.

[11] J. Degroote and J. Vierendeels. Multi-level quasi-Newton coupling algorithms for the partitioned simulation of fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 225-228:14–27, 2012.

[12] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182(3-4):499–515, February 2000.

[13] Oliver Fernandes, Steffen Frey, Filip Sadlo, and Thomas Ertl. Space-time volumetric depth images for in-situ visualization. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pages 59–65. IEEE, 2014.

[14] S. Frey, S. Filip, and T. Ertl. Mesh Generation From Layered Depth Images Using Isosurface Raycasting. In *ISVC '13: Proceedings of the 9th International Symposium on Advances in Visual Computing*, pages 373–383, Berlin, Heidelberg, 2013. Springer-Verlag.

[15] S. Frey, F. Sadlo, and T. Ertl. Explorable volumetric depth images from raycasting. In *Proceedings of the Conference on Graphics, Patterns and Images*, pages 123–130, Aug 2013.

[16] Markus Hadwiger, Patric Ljung, Christof Rezk Salama, and Timo Ropinski. Advanced illumination techniques for gpu volume raycasting. In *ACM SIGGRAPH ASIA 2008 courses*, SIGGRAPH Asia '08, pages 1:1–1:166, New York, NY, USA, 2008. ACM.

[17] Rob Haelterman, J. Degroote, Dirk van Heule, and J. Vierendeels. The Quasi-Newton Least Squares Method: A New and Fast Secant Method Analyzed for Linear Systems. *SIAM Journal on Numerical Analysis*, 47(3):2347–2368, 2009.

[18] Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. Generation of transfer functions with stochastic search techniques. In *Visualization*, pages 227–234, 1996.

[19] Ulrich Küttler and Wolfgang A. Wall. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.

[20] Ulrich Küttler and Wolfgang A. Wall. Vector Extrapolation for Strong Coupling Fluid-Structure Interaction Solvers. *Journal of Applied Mechanics*, 76(2), J. 2009.

[21] Marc Levoy and Pat Hanrahan. Light field rendering. In *23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 31–42, 1996.

[22] Eric J. Luke and Charles D. Hansen. Semotus visum: a flexible remote visualization framework. In *Proceedings of IEEE Visualization '02*, pages 61–68, 2002.

[23] Kwan-Liu Ma. In situ visualization at extreme scale: Challenges and opportunities. *IEEE Comput. Graph. Appl.*, 29(6):14–19, November 2009.

[24] Leonard McMillan and Gary Bishop. Plenoptic modeling: an image-based rendering system. In *22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 39–46, 1995.

[25] Miriam Mehl, Benjamin Uekermann, H. Bijl, David Stephen Blom, Bernhard Gatzhammer, and A. H. van Zuijlen. Parallel coupling numerics for partitioned fluid-structure interaction simulations. *SIAM Scientific Computing*, submitted 2013.

[26] C. Michler, E. H. van Brummelen, and R. de Borst. An interface Newton–Krylov solver for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, April 2005.

[27] Christian Michler, Harald van Brummelen, and René de Borst. An investigation of Interface-GMRES(R) for fluid–structure interaction problems with flutter and divergence. *Computational Mechanics*, 47(1):17–29, January 2011.

[28] D. P. Mok, W. A. Wall, and E. Ramm. Accelerated iterative substructure schemes for instationary fluid-structure interaction. In *First MIT Conference on Computational Fluid and Solid Mechanics*, pages 1325–1328, 2001.

[29] T. Ropinski, J. Prassni, F. Steinicke, and K. Hinrichs. Stroke-based transfer function design. In *IEEE/EG International Symposium on Volume and Point-Based Graphics*, pages 41–48, 2008.

[30] Jonathan Shade, Steven Gortler, Li-Wei He, and Richard Szeliski. Layered depth images. In *Proceedings on Computer Graphics and Interactive Techniques*, pages 231–242, 1998.

[31] Naeem Shareef, Teng-Yok Lee, Han-Wei Shen, and Klaus Mueller. An image-based modeling approach to GPU-based unstructured grid volume rendering. *Proceedings of Volume Graphics*, pages 31–38, 2006.

[32] A. Tikhonova, C. Correa, and Kwan-Liu Ma. Explorable images for visualizing volume data. In *IEEE Pacific Visualization Symposium*, pages 177–184, 2010.

[33] A. Tikhonova, C. D. Correa, and K.-L. Ma. An exploratory technique for coherent visualization of time-varying volume data. In *Proceedings of the Eurographics conference on Visualization*, pages 783–792, 2010.

[34] Stefan Turek and Jaroslav Hron. Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow. In Hans-Joachim Bungartz and Michael Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Modelling, Simulation, Optimisation*, pages 371–385. Springer Berlin Heidelberg, 2006.

[35] Yingcai Wu and Huamin Qu. Interactive transfer function design based on editing direct volume rendered images. *Visualization and Computer Graphics, IEEE Trans. on*, 13(5):1027–1040, 2007.

[36] Heung yeung Shum and Sing Bing Kang. A survey of image-based rendering techniques. In *In Videometrics, SPIE*, pages 2–16, 1999.