Machine Learning Approaches for Repositories of Numerical Simulation Results

Jochen Garcke[1,2], Rodrigo Iza Teran[1]

[1]Fraunhofer-Insitut SCAI, Numerical Data-Driven Prediction
Schloss Birlinghoven 53754 Sankt Augustin Germany
[2]Universität Bonn, Institute for Numerical Simulation
Wegelerstr 6 Bonn, Germany

1 Introduction

Finite element simulations of physical behavior are an essential tool in the development and improvement of industrial products for many years now [11]. Here, an approximate representation of the object by a mesh is employed and, based on a mathematical model of the physical process, a numerical simulation is performed.

For an accurate description such numerical approximations typically need to be very fine in the order of millions of unknowns in car crash, with larger simulations needed for some other applications.
The size of the resulting numerical data are huge, one simulation in time generates number of unknowns multiplied by the number of (saved) time steps numerical values. Furthermore, during the research and development process engineers easily generate several hundred variants of a specific finite element model that simulates different operating conditions of the product, multiplying the data once more. Post-processing software tools are readily available to display the 3D geometrical information of such a model and the results of one numerical simulation.

Engineers analyze and compare the different simulations using their own engineering knowledge, although this is generally limited to the simultaneous analysis of only a few simulations at a time. The complex structure of the data and its sheer size, the required 3D visualization of the geometry and the needed inspection of the associated design variables of each configuration prohibits a detailed comparative analysis of more than a few simulations by hand. There is need for a more efficient product development process which overcomes the current limitations. One approach consists in the use of a simulation data management system (SDM), where post processing quantities like the HIC index, intrusion, deformation energy and so on are managed in a convenient manner. This allows for querying requests about changes in previous designs as well as over the corresponding results on the basis of such post processing quantities. The use of an SDM system alleviates the problem of handling many large scale finite element simulations while keeping track of design changes. Nevertheless, the use of post processing quantities that are single scalar quantities or vectors does not allow an in-depth analysis of 3d deformations. Indeed an efficient and detailed analysis of this type for hundreds of design changes is nowadays a challenge in industrial practice.

The best known approach to tackle this data analysis challenge are Principal Component Analysis (PCA), to identify variation modes, and its counterpart classical Multidimensional Scaling (CMDS), for finding a low dimensional embedding. For several thousand simulations, corresponding to a specific product development phase, the PCA can recover the principal variations in just a few components.

The approach has already been successfully used for the analysis of numerical simulation data, see [1] [12], [13]. But in spite of their success so far, there are many situations where PCA or CMDS are not adequate. Being linear methodologies, they cannot cope with the presence of nonlinear correlations in the data, for example when dealing with time dependent highly nonlinear deformations.

Approaches that attempt to overcome these limitations have been proposed in recent years, see [2], [6]. In line of this research we propose to use nonlinear dimensionality reduction in the virtual product development process. Methods of machine learning are used that are known to be able to recover the so called intrinsic geometrical structure of the data in the presence of nonlinear correlations in the data.

Assuming the intrinsic geometry is low dimensional, recovering such nonlinear structures can be very effective for analyzing a number of simulations simultaneously. This work continues our research provided in [2] and [6] for a more realistic scenario in which a simulation data management system is assumed to be available to support the virtual product development process. Machine learning is demonstrated to enable a fast evaluation of not only post processing quantities but the complete 3D deformations of thousands of finite element design variants simultaneously. This improves considerably the usefulness of an SDM-System supporting a faster improved post processing of many simulation results.

In section 2 we provide a description of the use of a standard SDM in the virtual product development, section 3 discusses some concepts about nonlinear dimensionality reduction, the use of such methods in a SDM scenario is discussed in section 4. The application of the approach for the crash simulation bundle is presented in section 5 and we finally give an overview of the potential application of the methodology and discuss also further efforts in this area.

2 SDM in the Virtual Product Development

Analyzing the post-processing quantities derived from finite element information is today possible through the use of an SDM (Simulation Data Management System) System. Such system supports the engineer in the different steps of the evaluation of a virtual product through simulations:

1. Pre-processing – based on previous designs, a concept design is first set that will be subsequently be refined in several phases of the product design by changing the geometry, material data and other design related parameters in order to fulfill functional or regulatory constraints. A pre-processor will support the assembly process of the different design parts, re-meshing, setting of connection elements, welding points, loads, material parameters, etc. An input model is obtained for each model variation at the end of this step, in addition in this step a selection of the post-processing quantities, that have to be extracted after the simulation results are obtained from those input models, is chosen.

2. Solving – a SDM system will support delivering all required input models to a computing cluster and will trigger the execution of a parallel finite element solver.

3. Post-processing – the SDM system monitors the execution of the solver and when finished, it will trigger the extraction of the post-processing quantities selected in step 1.

4. Analysis – the SDM system provides the engineer all the extracted quantities through a client available to the engineer, based on those a decision process can be started for selecting the designs that fulfills functional requirements of regulatory constraints. Often this phase is also supported by the SDM allowing a comparative overview of different designs changes in correspondence to the simulation results.

An SDM system provides a convenient environment for product development, extending such systems for querying for specific post processing quantities or input variables is currently available. Data analysis capabilities can also be readily implemented in such environments for those quantities, nevertheless ideally a SDM system should support querying based also on geometrical objects like 3D geometries or deformations. Support for this type of analysis methods can be developed with the use of nonlinear dimensionality reduction which will be discussed next.

3 Nonlinear Dimensionality Reduction

Nonlinear dimensionality reduction is a very active area of research in machine learning in recent years, often known as manifold learning to emphasise the actual objective of such methods, namely the identification of low dimensional structures to represent high-dimensional information. Several methods have been introduced and we refer to [10], [14] for details about each of this methods.

For our setting each simulation can be treated as a point in a very high dimensional space given by the order of the discretization of the finite element mesh. The principal idea of these methods consist in the assumption that the data dimensionality is actually much lower and the issue is precisely to find this so called intrinsic dimensionality and the representation of the simulation in a coordinate system with such a number of dimensions. Intuitively this can be understood with a simple example of an image of a rotating object. The image is treated as an element in R^N where $N = b \times h$ is the size of the image in pixels, each rotated image is a high dimensional point in R^N but they all depend on the rotation angle, the angle is precisely the low dimensional coordinate and dimensionality reduction deals on how to find such coordinate (the rotation angle) based only on the pixels information of randomly sampled images. Once the angle is found, one says that the images are "located" on a low dimensional manifold (or space) given by the rotation angle.

To find such low dimensional structures, this work concentrates on so called kernel methods, which are based on the construction of a similarity matrix with coefficients calculated using a kernel function of the type $e^{-d(x,y)}$ where $x, y \in R^N$ are the simulations data sets and N can be very large (of the order of the total number of nodes on a mesh times the time steps of a numerical simulation). The distance function $d(x, y)$ is application dependent, being able to find an adequate distance between the high dimensional objects is critical for the success of the method. The kernel function is evaluated for all combinations of available datasets so that a matrix of dimension $m \times m$ is obtained where the number of simulations is m. The singular value decomposition (SVD) of the matrix is used to extract the eigenvectors and corresponding eigenvalues and it can be demonstrated that the intrinsic dimension can be estimated based on the decay of the eigenvalues, i.e. a low intrinsic dimension will have few large eigenvalues and the rest much smaller. The eigenvectors corresponding to largest eigenvalues are used as low dimensional coordinates for data analysis. In this work we use a specific variant of a kernel method called diffusion map, for details about the method see [9], [14].

Notice that in our analysis we use simulation data on a finite element mesh directly. A finite element mesh contains nodes and elements and we assume that the mesh connectivity is the same and use only the values defined at the nodes on the mesh as data set. If the mesh connectivity is different, a reference mesh can be used to map the simulation values from the slightly different numerical simulation meshes to it.

We would like to explain the mathematical principle of dimension reduction with a simple example in the next section.

3.1 Mathematical background for dimensionality reduction

Proposition:

Let S_α be a finite element model solution defined on a 1D mesh with N nodes and assume that the model depends on only one material parameter α and that this parameter has been changed m-times, obtaining a set of simulations $S_{\alpha_i}(r_1,...,r_N) \to R^N$, $i = 1,...,m$ which are represented as

$$x_i = S_{\alpha_i}(r_1,...,r_n), \quad i = 1,...,m .$$

Assertion: The simulations are located on a 1D curve that depends only on α.

Proof sketch:

The points are located in R^N. Since the models depend only on the parameter α there exists a mapping $u(\alpha): R \to R^N$ that takes points from R into R^N. Now a method for dimensionality reduction builds a matrix W with matrix components calculated as $w_{ij} = e^{-\|x_i - x_j\|^2} / \varepsilon, \; i, j = 1,...,m$, where ε is a parameter that represents the width of the Gauss function and $x_i = S_{\alpha_i}(r_1,...,r_N) \in \mathrm{R}^N$. The values of w_{ij} are further normalized and a discrete operator $Pf(x_i) = \sum_{j=1}^{M} p(x_i, x_j) f(x_j)$ can be constructed, where the $p(x_i, x_j)$ are the normalized component of the matrix. In [4] it has been shown that in the limit $m \to \infty, \; \varepsilon \to 0$, the discrete operator converges to a Laplace operator in the 1D space of the intrinsic parameter α.

We summarize the application of the method of dimensionality reduction for simulation as follows, The eigenvectors of the matrix operator converge under certain boundary conditions to the eigenfunctions of the operator $\frac{d^2 f}{d^2\alpha} = \lambda f$. Finally for a close curve with fixed end values and length $\alpha_{\max}$, the first non-zero eigenfunctions are given by $\cos(\frac{\alpha}{\alpha_{\max}})$. □

This shows in a simple way that even if the simulations are located in R^N , an intrinsic low dimensional structure can be found in only one dimension which is a function of the intrinsic material parameter α . That is one is able to identify the parameter α with the only information of the randomly sampled high dimensional simulations.

This result is very interesting from an engineering point of view. It suggests that using only randomly sampled simulation datasets some intrinsic parameters can be found along which a parametrization of all such simulations can be achieved. We summarize the steps of the method of dimensionality reduction for a set of simulation datasets:

Input data: m simulations $x_i \in R^N$

- Evaluate the matrix components $w(i,j) = e^{-d(x_i,x_j)^2} / \varepsilon, \quad i,j = 1,...,m$
- Normalize the matrix
- Solve the eigenvalue problem
- Choose eigenvectors corresponding to largest eigenvalues
- Choose a dimension r based on the decay of the eigenvalues

Output: r eigenvectors that represent the low dimensional coordinates spanning a space of dimension r .

The use of low dimensional coordinates obtained from a dimensionality reduction method allows the parametrization of all analyzed datasets. Each simulation corresponds to a point in this low dimensional space and the information gets organized according to the variation of some intrinsic parameters. The data is organized according to some principal trends, for example deformations that are geometrically similar will be located very near in this low dimensional space.

A very important observation is that post processing can be speed-up by the use of the method, since the task of evaluating and classifying many large 3D deformations according to their similarity can be easily done on a low dimensional space. How this could be done inside a SDM environment is highlighted in the next section.

3.2 A SDM Approach for Analyzing Numerical Simulation Data

We describe a general framework that allows for the analysis of high dimensional data from bundles of large finite element simulations.

Several steps are necessary and many of them can be supported by standard SDM systems directly while others need still to be developed. We assume that the big data is located in large computing cluster not at the engineer's desktop (see Figure 1):

1. Data extraction - the raw data for the analysis are obtained directly from the simulation, this contains several variables of different types such as scalars, vectors or tensors defined on nodes or elements of a finite element mesh. The SDM system can be easily be modified to extract those quantities for analysis after or during a simulation on a computing cluster.

2. Pre-processing - this step is usually necessary to cope with the huge data size (millions of nodes and elements) and its complex nature. A usual pre-processing step consists in the use of only subsets of the datasets in areas of interest (for example the supporting frontal beams of a car in a frontal crash), also sub-sampling and clustering can be employed. The data can also be

transformed in different ways in order to obtain a compact representation; for example using principal component analysis (PCA).

3. Dimensionality Reduction - in this step a low dimensional representation is obtained from the dataset that parametrizes the information. Several dimensionality reduction methods can be used in this framework.

4. Exploration - based on the low dimensional representation found after the dimensionality reduction. The simulation variables are organised in the low dimensional embedding space, i.e. each numerical simulation is represented as a point in the obtained low dimensional space. Due to the reduced dimensionality, the datasets can be efficiently explored to find similarities and the effect of input variables.

The first two steps in the analysis workflow involve costly data intensive computations. Preferably these take place in a parallel server infrastructure where the bulky data is stored (see Figure 1), therefore avoiding transfer of the big data and exploiting the parallel HPC resources.

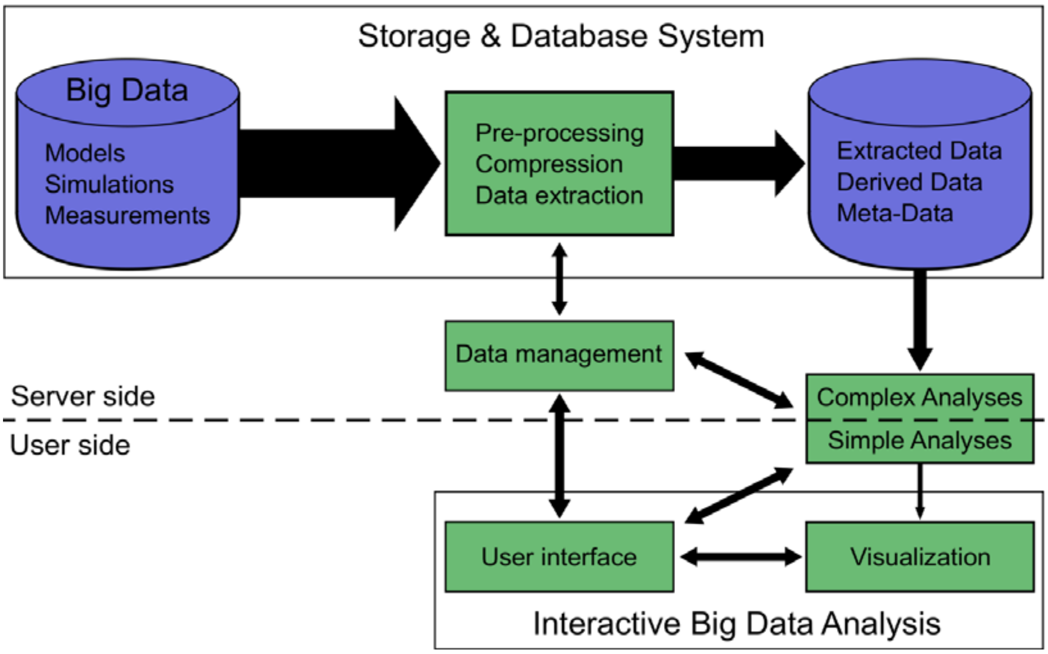

Fig.1: A schematic representation of an SDM system that used dimensionality reduction.

After processing and application of dimensionality reduction, the data exploration in step 3 should work interactively. A visualization framework should allow the navigation through the low dimensional representation and at the same time a second window of the framework should be able to update its content very fast with the representation of the 3D deformations corresponding to the low dimensional point that has been selected in the exploration step. Using such an interactive framework geometrical variants can be explored as well as deformations changes for time dependent problems.

A way to incorporate the information about the input parameters on the embedding can also be realized so that the impact of such changes in the deformations can be analysed.
This data exploration capability is not available on current SDM systems but it can be developed. An example of the exploration approach will be presented in section 4.

4 Robustness Analysis in Frontal Crash

For this case study we investigate a current Toyota model with 998,000 nodes and corresponding elements, which is publically available from the NCAC [7]. The position of the bumper was changed along a circle, see Figure 2 which results in an observably different crash behaviour during a frontal impact with a barrier.

A total of 243 numerical simulations where performed using LS-DYNA simulation software [8]. A total of 26 intermediate time steps were saved for each run.
The training dataset is saved in files in a binary format, so called post-processor software, can be used to read this data and extract (parts of) it. We use the software Animator [5] to extract specific components of a car or structure. The components we choose are the ones that are critical for the engineer during the analysis of the structural behaviour under crash (see Figure 3).

Note that in [2], [6] car crash data was analysed stemming from a simpler model from that repository, with more than one order of magnitude less grid points, to study the general analysis procedure. In this work we analyse simulation data in a size currently being used from a difficult engineering task, and further proceed to an interpretation of the results from an engineering perspective.

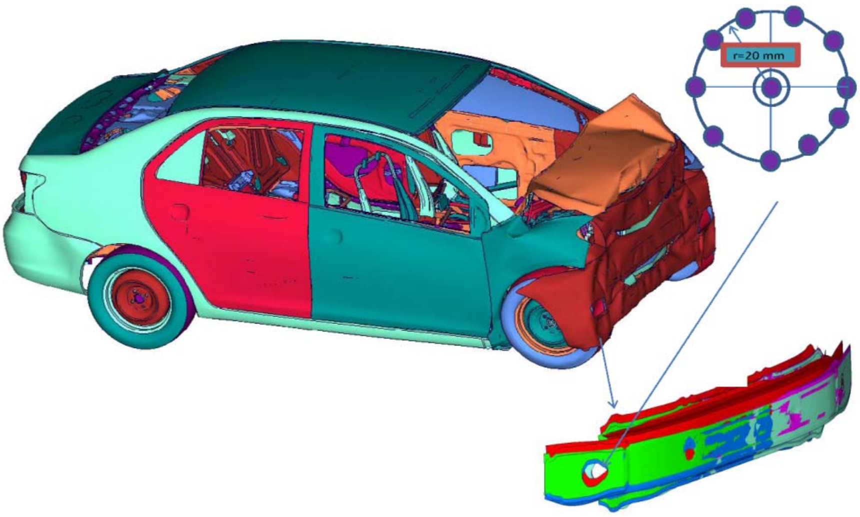

Fig.2: Variations of the bumper position along a circle.

We apply the workflow explained in section 3 to all simulations and extract in the preprocessing step as relevant structural parts the firewall and the structural beams, see Figure 3. The data analysis is performed at time step 14, when most of the crash affecting the frontal structure took place.

For calculating the entries of the similarity matrix we use as dataset $x \in R^N$ the resulting deformation at the mesh points of the selected parts. An arbitrary simulation is taken as reference and the distance function d used, is the Euclidean distance. Each simulation is a vector in R^N where N is the total number of nodes of the selected parts. The dimensionality reduction is then performed with these feature vectors.

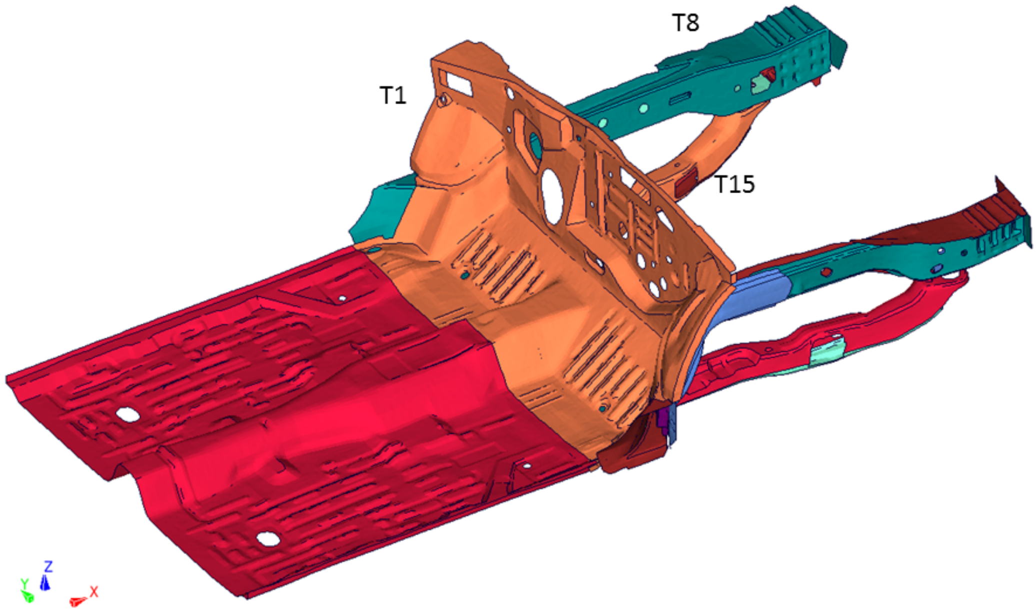

Fig.3: Structural parts relevant for the engineering analysis, selected are frontal beams and firewall.

For the exploration step of the workflow, each of the simulations is now represented as a point in a 3D plot (that is we choose $r = 3$ in the framework of section 3] which therefore gives a parametric representation of the 243 simulations in three dimensions.

To have a comparison with a standard linear approach we first performed a CMDS analysis for the same dataset using the first spectral coordinates as embedding for the exploration step, which is shown in Figure 4. Although simulations using a similar angle are arranged nearby, no distinct structures can be identified.

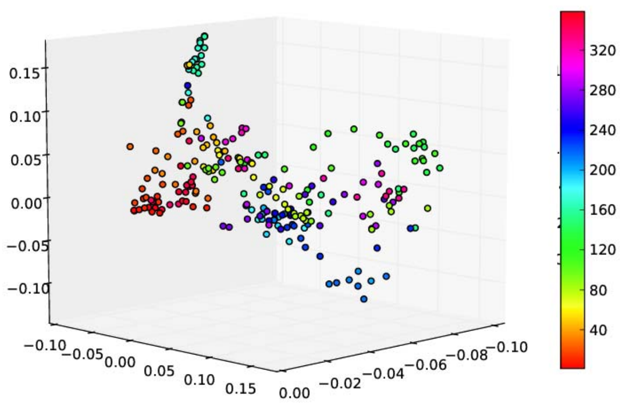

Fig.4: We show the CMDS embedding, where each simulation run is color coded with the angle of the bumper variation.

In Figure 5 the low dimensional embedding with a nonlinear dimension reduction (for this example diffusion maps) is shown. It can clearly be seen that the simulations are organized according to a certain type of deformation mode. The color of the deformed structure corresponds to the difference to the chosen reference model and the color of the points in the 3D plot corresponds to a specific angle for the bumper location in degrees.

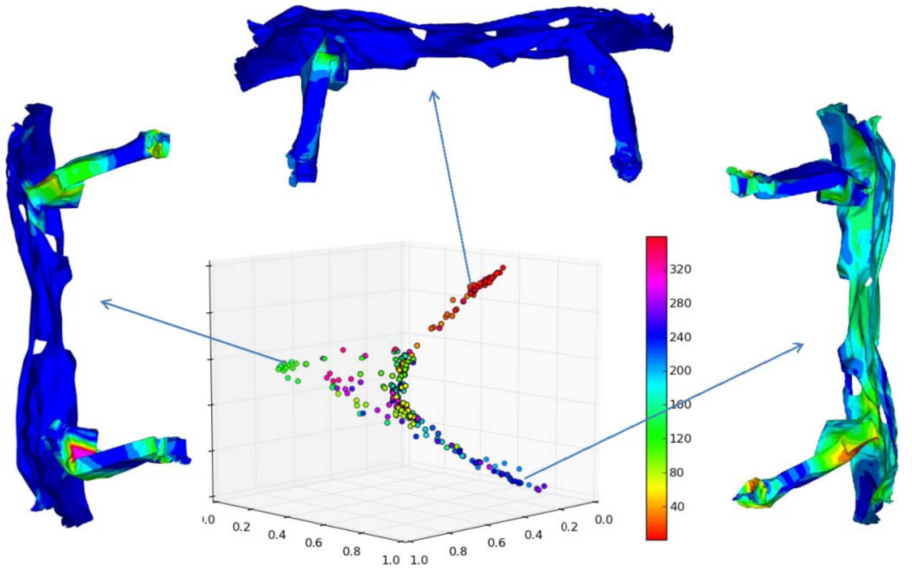

Fig.5: Embedding obtained by diffusion maps. Each simulation run is color coded with the angle of the bumper variation. Adjacent to the embedding plot are representative deformation modes for each cluster colored according to the difference from a reference configuration.

From the embedding plot for diffusion maps in Figure 5, it can be seen that at least three deformation modes can be associated to a range of positioning angles of the bumper. We extracted for clarity a beam from each mode in order to show the typical deformation in each mode and the angle dependence, see Figure 6.

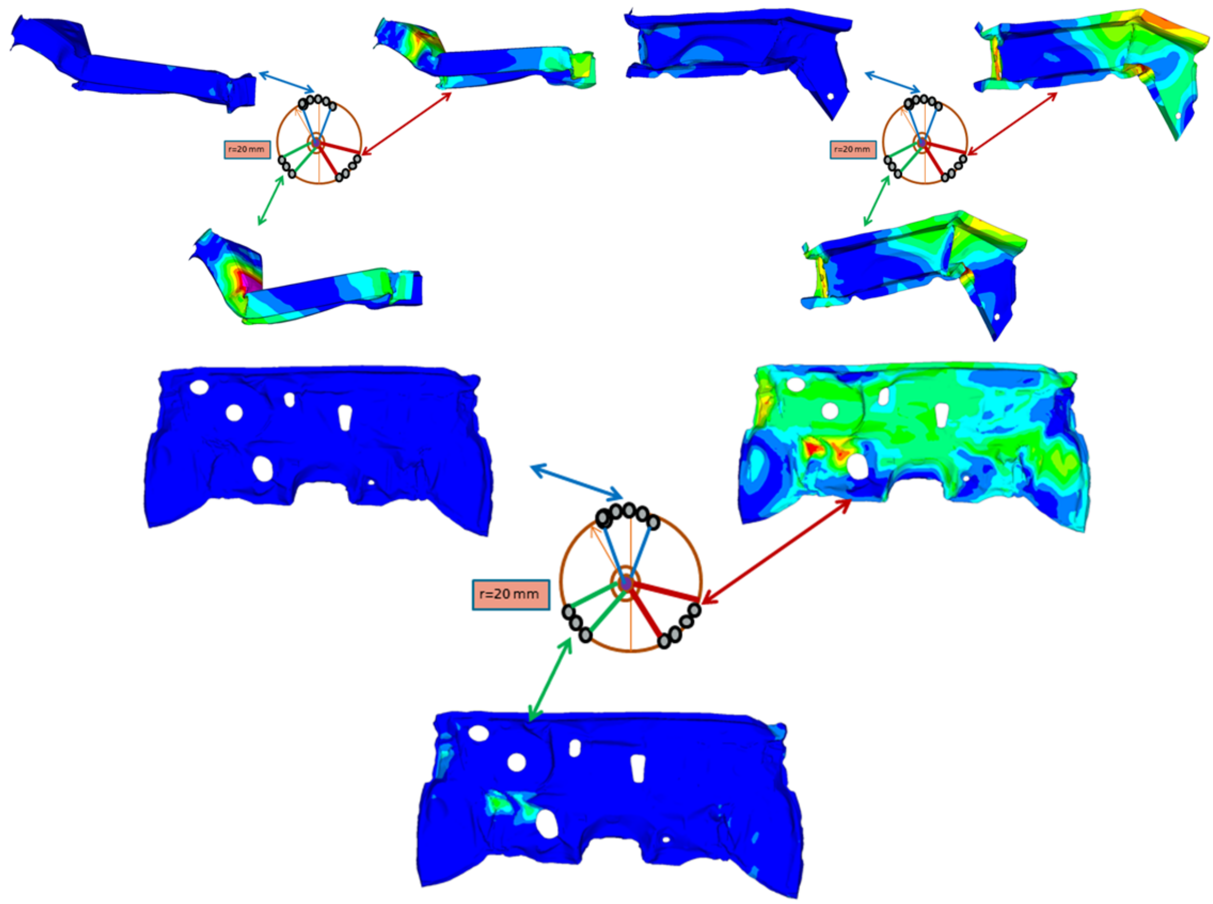

Fig.6: Bumper deformation modes derived from the diffusion maps embedding from Figure 5. Color code corresponds to differences from a reference simulation.

The computational time for data loading and preprocessing in an off-line-phase takes less than an hour, which is mostly due to the data extraction, and needs to take place once for each data set.
The actual data exploration afterwards can be done interactively. The selection of the corresponding simulations and the generation of the 3D pictures shown in Figure 5,6 took less than an hour. In comparison an engineer in charge of this project normally needs to analyze each of the deformation modes by hand (using Animator for example) and then classify them. Such task can require several days, if not a week for the amount of simulations involved.

5 Overview and Perspectives

The use of nonlinear dimensionality reduction inside an SDM simulation framework has been shown. The approach allows organizing simulation data sets along low dimensional structures that identify principal variation modes from a set of many simulations. A realistic large scale industrial example in crash simulation is used, whereby the position of a bumper is changed. We are able to show that different deformation modes in the structural beams of the car can be easily identified in a low dimensional representation.

Using nonlinear techniques reduces the complexity and time for investigating such large bundles of huge numerical simulation data. To deploy such an approach for the analysis of large scale simulation data in real-life environments, it will need to handle efficient data storage (including compression) which will take place on data servers in the future instead of workstations as nowadays is the practice in industry. It will also need efficient transfer of the (relevant) data between server and client, as well as efficient data processing for analysis and visualization procedures, like the one outlined in this work, on the server and the client.

It can be seen, that with the help of these new technologies a conceptually different approach to the analysis of the large data arising in the virtual product development process becomes possible. The approach allows an intuitive and interactive handling of the simulation results and provides to the development engineer simple possibilities for the comparative and concurrent examination of many simulation results.

Acknowledgements

The variations of the bumper positions were designed by Claudius Schoene of GNS mbH in the course of the KMU-innovative project FEMMINER supported by the BMBF. Part of the work was supported by the KMU-innovative project FEMMINER and the BMBF-supported projects SIMDATA-NL and VAVID.

6 Literature

[1] Ackermann S., Gaul L, Hanss M. and Hambrecht T.: “Principal Component Analysis for Detection of Globally Important Input Parameters in Nonlinear Finite Element Analysis”, Weimar Optimization and Stochastic Days 5.0, 2008

[2] Bohn B, Garcke J, Iza Teran R., Paprotny A., Peherstorfer U., Schepsmeier U. and Thole C. A.:“Analysis of car crash simulation data with nonlinear machine learning methods“, Proceedings oft he ICCS 2013, vol 18, Procedia Computer Science, Elsevier 2013, 621-630

[3] Braess D.: "Finite Elemente", Springer, 2013, 5 Edition.

[4] Coifmann, R. and Lafon S.: "Diffusion Maps", Applied and Computational Harmonic Analysis, 21(1), 2006, 5-30.

[5] GNS, GNS mbH.: “http://www.gns-mbh.com/”

[6] Iza Teran R.: "Enabling the analysis of finite element simulation bundles", International Journal for Uncertainty Quantification,(1), 2104.

[7] Kenneth O.: “Finite Element Model Archive, http://www.ncac.gwu/vml/models.htm/”

[8] LSTC, Livermore Software Technology.: “LSDYNA Keyword User’s Manual“, 2014.

[9] Lafon S., Lee A.: "Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning and Data Set Parametrization", IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9), 2006, 1393-1403.

[10] Lee J. and Verleysen M.: "Nonlinear Dimensionality Reduction", Springer, 2007.

[11] Meywerk M.: "CAE-Methoden in der Fahrzeugtechnik”, Springer, 2007.

[12] Mei L. and Thole C-A.: “Data analysis for parallel car-crash simulation results and model optimization”, Sim. Modelling Practice and Theory, 16(3), 2008, 329-337.

[13] Thole C-A., Nikitina L., Nikitin I. and Tanja Clees. :”Advanced mode analysis for crash simulation results”, Proceedings of the 9th LS:-DYNA Forum, 2010.

[14] Wang J., “Geometric Structure of High-Dimensional Data and Dimensionality Reduction", Springer 2011.