Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries

Lars Gräning, Stefan Menzel, Martina Hasenjäger, Thomas Bihrer, Markus Olhofer, Bernhard Sendhoff

2008

Preprint:

This is an accepted article published in Journal of Mathematical Modelling and Algorithms, JMMA (2008). The final authenticated version is available online at: https://doi.org/[DOI not available]

Noname manuscript No.
(will be inserted by the editor)

Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries

Lars Graening · Stefan Menzel · Martina Hasenjäger · Thomas Bihrer · Markus Olhofer · Bernhard Sendhoff

Received: date / Accepted: date

Abstract Applying numerical optimisation methods in the field of aerodynamic design optimisation normally leads to a huge amount of heterogeneous design data. While most often only the most promising results are investigated and used to drive further optimisations, general methods for investigating the entire design dataset are rare. We propose methods that allow the extraction of comprehensible knowledge from aerodynamic design data represented by discrete unstructured surface meshes. The knowledge is prepared in a way that is usable for guiding further computational as well as manual design and optimisation processes. A displacement measure is suggested in order to investigate local differences between designs. This measure provides information on the amount and direction of surface modifications. Using the displacement data in conjunction with statistical methods or data mining techniques provides meaningful knowledge from the dataset at hand. The theoretical concepts have been applied to a data set of 3D turbine stator blade geometries. The results have been verified by means of modifying the turbine blade geometry using direct manipulation of free form deformation (DMFFD) techniques. The performance of the deformed blade design has been calculated by running computational fluid dynamic (CFD) simulations. It is shown that the suggested framework provides reasonable results which can directly be transformed into design modifications in order to guide the design process.

Keywords Knowledge extraction · Data mining · Decision tree · Rule extraction · Knowledge Verification · 3D turbine blade · Surface mesh representation · Design optimisation · Computational fluid dynamics · Direct manipulation of free form deformation

L. Graening, S. Menzel, M. Hasenjäger, T. Bihrer, M. Olhofer, B. Sendhoff
Honda Research Institute Europe GmbH
Carl-Legien-Strasse 30
D-63073 Offenbach/Main, Germany
E-mail: (lars.graening, stefan.menzel, martina.hasenjaeger, thomas.bihrer, markus.olhofer, bernhard.sendhoff)@honda-ri.de

1 Introduction

In the field of 3D aerodynamic shape optimisation, a large amount of geometric and flow field data is generated during the design process that usually encompasses several optimisation runs, manual design phases and experiments. Typically, only the most promising results with regard to one or more possibly competing performance indices are exploited to define the overall result of the design process. However, a lot of information about the process and the problem at hand that can be condensed into comprehensive rules is hidden in all of the data. Even poorly performing shapes can provide interesting insight into the interrelation between geometry and performance. The knowledge extracted from the data set at hand can be prepared in such a way that it is on the one hand usable by the engineer and on the other hand by a follow-up computational design and optimisation process. This type of knowledge extraction is the major focus of the present paper.

The paper is organised as follows. In the remainder of the first section methods for guiding computational and human centred design optimisation processes are reviewed. This is followed by the specification of unstructured surface meshes that are used as a common geometric representation of 3D designs. In Section 2, we will outline the new displacement measure, which captures local differences between designs. The extraction of knowledge based on displacement and performance data is described in Section 3. The theoretic concepts have been applied to the design of a 3D turbine blade geometry and the results are discussed in Section 4. In the last section, we summarise the paper and highlight open issues.

1.1 Guiding Aerodynamic Design Optimisation

Different techniques have been researched in order to make use of already generated design data for guiding further optimisations with the objective of improving the aerodynamic properties of the design at hand. Response surface methodology [27] is one of the classical approaches in this domain where the parameters of a regression model are estimated based on previously made experiments. It is expected that the regression model reflects the true trend in mapping the design variables to the performance numbers. The estimated trend is used to generate new designs that are likely to improve the aerodynamic performance. In computational optimisation techniques like evolutionary algorithms (EA) it is common to use information from already generated designs to guide the optimisation, e.g. using step size adaptation techniques like covariance matrix adaptation [8] in order to control the variation of the design parameters during the optimisation process. Another class of computational optimisation algorithms that are derived from evolutionary algorithms uses a probabilistic approach to guide the optimisation, namely estimation of distribution algorithms (EDA) [21]. In EDA the crossover and mutation operators are replaced by learning and sampling the probability distributions (e.g. using Bayesian networks) of the best individuals out of a population in order to exploit the considered system parameters. Typically the evaluation time is the most limiting factor for exploring large parts of the design space. In computational optimisation so-called surrogate models (also termed as meta-models) are used to estimate the performance numbers thus replacing time-consuming high fidelity simulations [16]. Different kinds of surrogate models have been applied, e.g. artificial neural networks (ANN), support vector machines (SVM), Kriging Models, etc.. The existing data is

used to update the parameters of those models. Further techniques for incorporating prior knowledge as well as knowledge acquired during the optimisation process are summarised by Jin [17]. Keane and Nair [18] reviewed computer-aided design techniques for analysing and optimising data applied to the aerospace domain.

Beyond those techniques, methods for extracting human readable knowledge from aerodynamic design data are rare. The main purpose of knowledge extraction in this domain is to guide human centred design optimisation by improving the understanding of basic design principles related to the design problem at hand. Obayashi et al. [28] have addressed the extraction of knowledge from a given data set in order to gain some insights into the relationship between geometry and performance measurements. They have used self-organising maps (SOM) [19] in order to find groups of similar designs and for multi-criteria performance improvements and tradeoffs. Besides SOM, Chiba et al. [4] have investigated the analysis of variance technique (ANOVA) in order to identify the most important design parameters. Their methods have been applied to supersonic wing design but the data and design parameter sets are generated by one computational optimisation algorithm and were therefore well-defined. If there is no common representation, like in the case the data are from different design processes, including manual design processes involving data from CAD systems, their methods can hardly be applied.

1.2 Surface Representation

Different geometric representations [29] make it difficult or even impossible to analyse the whole data set based on one homogeneous parameterisation. If different optimisation runs have been performed with different design parameters, one first has to find an adequate representation that captures all shape variations and that can be applied to various data mining techniques. Therefore, we suggest the use of unstructured surface meshes as a general representation. Each optimisation might be a manual or a computational process and is possibly based on different shape descriptors. The majority of shape representations are convertible to unstructured surface meshes, see e.g. [14], [32].

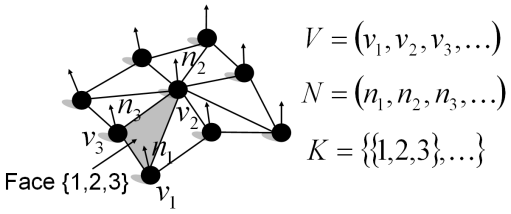

Fig. 1 Illustration of the specification of a triangular surface mesh $\mathcal{M} : (\mathcal{V}, \mathcal{N}, \mathcal{K})$ with $\mathcal{V}$ containing a set of vertices, $\mathcal{N}$ capturing normal vectors at the position of each vertex and $\mathcal{K}$ specifying the polygonal faces that built up the mesh.

For the description of the surface mesh we start with the mathematical framework given in [1]. It is assumed that the shape of a 3D design is described using a polygonal surface mesh $\mathcal{M}$, which is a partially linear approximation of the contour of the design. We postulate that each mesh $\mathcal{M}$ consists of a list of vertices $\mathcal{V}$, a complex $\mathcal{K}$ and a list

of normal vectors $\mathcal{N}$. The vertex list $\mathcal{V} = (\mathbf{v}_1, ..., \mathbf{v}_n)$ describes the geometric position of the vertices in $\mathbb{R}^3$, $\mathbf{v}_i = (x_1, x_2, x_3)^T$. A vertex can be seen as a sample point of the contour of the design. Each face of the polygonal surface mesh is defined by simplices of the form $\{i_1, i_2, i_3, ..., i_\mu\}$ where $i_l, l \in [1..n]$ are indices pointing to vertices that enclose the polygonal face. Fig. 1 illustrates a triangular surface mesh where the number of vertices, which have been used to form each polygon, is set to 3. In addition to the vertex list, a list of normal vectors $\mathcal{N} = (\mathbf{n}_1, ..., \mathbf{n}_n)$ with $\mathbf{n}_i = (n_1, n_2, n_3)^T$ is given. Each normal vector $\mathbf{n}_i$ has a defined direction perpendicular to the surface mesh and provides local curvature information at the position of vertex $\mathbf{v}_i$.

The unstructured surface mesh as a general representation allows the analysis of local shape modifications and their influence on the performance value(s) independent of the parametric representation that has been used during the design and optimisation process.

2 Displacement Measurement

Under the assumption that the surface triangulation results in surface meshes for which the location and the number of vertices is sufficiently precise to capture the characteristic changes of all designs in the given data set, the displacement is measured between each vertex on the reference design and each corresponding vertex on the modified design. Mesh refinement and simplification algorithms, e.g. [31,7], allow a further optimisation of the triangulation. In order to measure the displacement between two vertices of different surface meshes, the correspondence problem has to be solved and an appropriate metric has to be found for measuring the amount and the direction of the displacement between the pair of vertices. The resulting displacement data is the basis for knowledge extraction from the design data set, see Section 3.

2.1 Identifying Corresponding Vertices

An appropriate identification of corresponding vertices is essential for measuring the correct displacement between two surfaces. Wrong estimates will lead to an error in the displacement measurement and hence to errors in any further knowledge extraction step. Let $\mathcal{M}_r$ and $\mathcal{M}_m$ be two unstructured surface meshes, where $\mathcal{M}_r$ is a baseline or reference mesh and $\mathcal{M}_m$ is the target or modified mesh. The main objective in solving the correspondence problem is to find an appropriate function f which assigns to each vertex $\mathbf{v}_i^r \in \mathcal{V}_r$ from the reference mesh $\mathcal{M}_r$ a corresponding vertex $\mathbf{v}_j^m \in \mathcal{V}_m$ from the target mesh $\mathcal{M}_m$. More formally this can be written as:

$$f : \mathcal{V}_r \rightarrow \mathcal{V}_m, \quad \mathcal{V}_r \in \mathcal{M}_r, \quad \mathcal{V}_m \in \mathcal{M}_m \tag{1}$$

So far there exists no universal transfer function f that leads to robust solutions without providing any prior knowledge or assumption concerning the design data set at hand. A more specific function has to be found which is tuned to the design data characteristics. It is assumed that no rigid design modification has been applied to any of the designs. All designs are located in a universal position. We rely on the fact that the differences between designs can completely be expressed in terms of local non-rigid design modifications.

The following objective function is used in order to assign a corresponding vertex $\mathbf{v}_j^m$ to each vertex $\mathbf{v}_i^r$:

$$f(\mathbf{v}_i^r) = \min_{j \in n_m} \Big\{ |\mathbf{v}_i^r - \mathbf{v}_j^m| \cdot (2 - \mathbf{n}_i^r \circ \mathbf{n}_j^m) \Big\}, \tag{2}$$

where n_m equals the number of vertices that make up the surface mesh of the target design, $|\mathbf{v}_i^r - \mathbf{v}_j^m|$ defines the Euclidean distance and $\mathbf{n}_i^r \circ \mathbf{n}_j^m$ measures the difference between the normal vectors of the vertices. The objective function has been defined so that the vertex $\mathbf{v}_j^m$ that is closest to $\mathbf{v}_i^r$ and has a similar normal vector is assigned as the corresponding vertex. On the one hand if the normal vectors have the same orientation the scalar product equals one and hence the function relies completely on the spatial distance between vertices. If on the other hand the normal vectors are perpendicular (pointing in opposite direction) the objective function is two times (three times) the Euclidean distance. The objective function is a simplification of the one used by Wang [33]. He additionally added a term that measures the difference in the Gaussian curvature, which was mainly due to the application to the analysis of brain images. In general, it is possible that one vertex from mesh $\mathcal{M}_m$ is assigned to more than one a vertex of mesh $\mathcal{M}_r$. An overview of different techniques for estimating corresponding vertices is given in [22].

2.2 Definition

The displacement measure should describe the position of a vertex with respect to another design. One way to capture this information is to use the difference vector $\mathbf{s}_{ij} = \mathbf{v}_i^r - \mathbf{v}_j^m$, which is the difference between vertex $\mathbf{v}_i^r$ of mesh $\mathcal{M}_r$ and the corresponding vertex $\mathbf{v}_j^m$ of mesh $\mathcal{M}_m$. The difference vector clearly captures the displacement between $\mathbf{v}_i^r$ and $\mathbf{v}_j^m$. However, the difference vector is sensitive to possible errors resulting from wrong estimations of the corresponding vertices or from different sampling methods of the surfaces of the geometries. Furthermore, the difference vector requires $d = 3$ parameters for describing the displacement of one vertex in $\mathbb{R}^3$. Thus, to capture the displacement between two complete surface meshes the number of parameters is $3 \cdot n_r$, where n_r equals the number of vertices that belong to the reference mesh. To overcome the disadvantages of the difference vector, we suggest the following displacement measure:

$$\delta_{i,j}^{r,m} = \delta(\mathbf{v}_i^r, \mathbf{v}_j^m) = (\mathbf{v}_j^m - \mathbf{v}_i^r) \circ \mathbf{n}_i^r, \delta \in (-\infty, +\infty) \tag{3}$$

The displacement measure is defined as the projection of the difference vector $\mathbf{s}_{ij} = (\mathbf{v}_i^r - \mathbf{v}_j^m)$ onto the normal vector $\mathbf{n}_i^r$ of vertex $\mathbf{v}_i^r$ of the reference design $\mathcal{M}_r$. The absolute value of the displacement measure provides information on the amount of vertex modification while the sign of the displacement measure in conjuction with the normal vector of the vertex provides information on the direction of the vertex modification. The normal vector $\mathbf{n}_i^r$ points towards the normal or positive direction of vertex modification.

2.3 Major Properties

The displacement measure is by definition a non-symmetric vector quantity containing both the magnitude and the direction of vertex modification. If the modified vertex

lies above the tangential plane described by the normal vector of the reference vertex, the displacement measure is positive, see Fig. 2 a). Whereas if the vertex lies below the tangential plane, Fig. 2 b), the displacement measure is negative. In the special case where the modified vertex is located within the tangential plane, the displacement measure is zero as shown in Fig. 2 c), which makes sense because we have to assume that the difference is a result of a different triangulation. If the reference vertex has been modified along the line described by the normal vector, the amount of the displacement measure equals the Euclidean distance between the reference and the modified vertex.

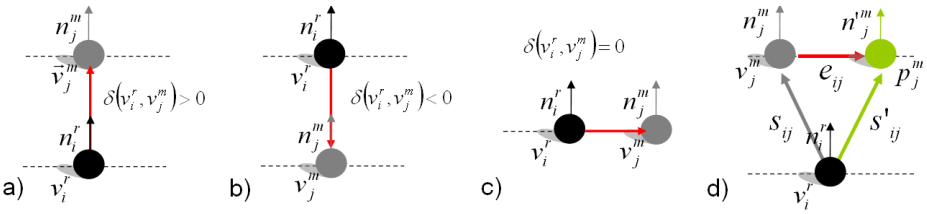

Fig. 2 Examples of the displacement measure. Figures a) and b) illustrate that a vertex displacement parallel (anti-parallel) to the normal direction results in a positive (negative) displacement value. A displacement perpendicular to the normal vector results in a displacement value of zero, as shown in c). Figure d) illustrates the error when calculating the displacement measure, which results from the discretisation of the surface and the error when estimating corresponding points.

As Fig. 2 d) indicates, the displacement value contains an error, which is mainly the result of the discretisation of the surface using triangulation and of the correspondence problem. Formally, this can be written as

$$\delta_{i,j}^{r,m} = (\mathbf{s}_{ij} + \mathbf{e}_{ij}) \circ \mathbf{n}_i^r = \mathbf{s}_{ij} \circ \mathbf{n}_i^r + \mathbf{e}_{ij} \circ \mathbf{n}_i^r, \tag{4}$$

where $\mathbf{e}_{ij}$ describes the error between the ideal displacement value and the measured displacement value. Under the assumption that the curvature of both surfaces $\mathcal{M}_r$ and $\mathcal{M}_m$ is similar at the position of the corresponding vertices it follows that $\mathbf{n}_i^r \approx \mathbf{n}_j^m$. Then, the error term from equation 4 simplify as:

$$\mathbf{e}_{ij} \circ \mathbf{n}_i^r \approx \mathbf{e}_{ij} \circ \mathbf{n}_j^m. \tag{5}$$

With $\mathbf{e}_{ij} \circ \mathbf{n}_j^m = |\mathbf{e}_{ij}| \cos(\angle(\mathbf{e}_{ij}, \mathbf{n}_j^m))$, if additionally a smooth surface or a small error $|\mathbf{e}_{ij}|$ is assumed, $\mathbf{e}_{ij}$ is perpendicular to $\mathbf{n}_j^m$ and hence $\cos(\angle(\mathbf{e}_{ij}, \mathbf{n}_j^m)) \approx 0$. Thus the error term becomes zero. Therefore, the displacement measure is relatively insensitive to small errors arising from the surface triangulation or from an incorrect estimation of corresponding points.

Another advantage of the displacement measure compared with the difference vector is that only n_r parameters are required for the description of the differences between two unstructured surface meshes, where n_r equals the number of vertices of the reference mesh.

3 Knowledge Extraction from Design Data

In aerodynamic design optimisation the main goal is to find three-dimensional shapes, that are optimal for specific performance measurements, like aerodynamic drag or

lift, under specific constraints, e.g. manufacturing limitations. In general, during the optimisation process a large number of shapes are generated and evaluated based on different representations and parameterisations. The results are heterogeneous design data sets from which only a very small number of designs are used in the end to determine the optimal shape (or a set of optimal shapes) that is processed further, e.g. in rapid prototyping devices for experiments. As we noted in the introduction, we aim at exploiting the information contained in the large remaining part of the data set. In this section, we describe how the displacement measure in conjunction with statistical and data mining methods can be used in order to extract meaningful information (knowledge) from heterogeneous design data sets. Figure 3 provides an overview of the entire knowledge extraction framework.

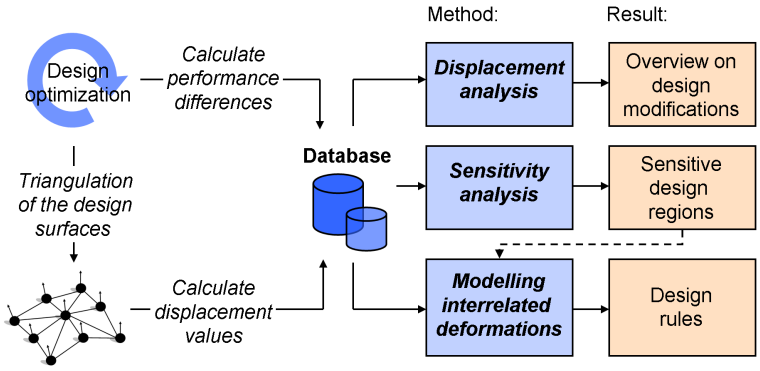

Fig. 3 Overview of the knowledge extraction framework for analysing design data resulting from the design optimisation process.

3.1 Displacement Analysis

Analysing local modifications in form of vertex displacement helps to gain some insight into the exploration of the design space. Two measures are suggested: the relative mean vertex displacement that provides information on how a vertex has been modified with respect to one reference design and the overall displacement variance that identifies the vertices that have been modified most frequently.

3.1.1 Relative Mean Vertex Displacement

In order to get information on local design modifications with respect to one baseline design, we define the *relative mean vertex displacement* :

$$\overline{\delta}_i^r = \frac{1}{N-1} \sum_{m=1, m \neq r}^{N} \delta_{i,j}^{r,m} \tag{6}$$

Given a data set of N unstructured surface meshes $\overline{\delta}_i^r$ evaluates the mean displacement of vertex $\mathbf{v}_j^m$ of all meshes $\mathcal{M}_m$, $m \in [1 \dots N]$ from the vertex $\mathbf{v}_i^r$ of the reference

mesh $\mathcal{M}_r$. Here, like in the following we assume that $\mathbf{v}_j^m$ is the corresponding vertex to $\mathbf{v}_i^r$. $\overline{\delta}_i^r$ provides information on how far a reference vertex has been modified along its normal vector with respect to the whole data set. If $\overline{\delta}_i^r > 0$, the vertex $\mathbf{v}_i^m$ has been modified parallel to the normal vector of the vertex and $\overline{\delta}_i^r < 0$ indicates a modification anti-parallel to the normal direction of the vertex. If $\overline{\delta}_i^r = 0$, the vertex has not been modified, each displacement value was zero due to modifications orthogonal to the normal vector or the modifications around the reference vertex in the data set have cancelled each other out. If there are outliers that affect the calculation of $\overline{\delta}_i^r$, we recommend to use the median instead of the mean in order to retrieve the desired information.

3.1.2 Overall Displacement Variance

In order to calculate $\overline{\delta}_i^r$, the baseline mesh r must be given. An alternative would be to calculate the mean displacement between all possible shape combinations in the data set. However, this is not a good alternative, because if the normal vectors of corresponding vertices are similar, it holds that $\delta_{i,j}^{r,m} \approx -\delta_{i,j}^{m,r}$ and as a result such a measure would always tend to zero.

In order to get an overview over the variations of local design modifications, an *overall displacement variance* is defined as follows:

$$\begin{aligned} \sigma_{\delta_i} &= \sqrt{\frac{1}{N(N-1)} \sum_{r=1}^{N} \sum_{m=1, m\neq r}^{N} (\delta_{i,j}^{r,m} - \overline{\delta}_i)^2} \\ \sigma_{\delta_i} &\approx \sqrt{\frac{2}{N(N-1)} \cdot \sum_{r=1}^{N} \sum_{m=r+1}^{N} (\delta_{i,j}^{r,m})^2} \end{aligned} \tag{7}$$

This measure describes the strength and the frequency of local design modifications based on the whole data set. Following our argument above, we can set $\sum_{r=1}^{n} \overline{\delta}_i \approx 0$.

3.2 Sensitivity Analysis

Sensitivity analysis relates the displacement measure to variations of the corresponding performance values.

3.2.1 Relative Vertex Correlation Coefficient

The *relative vertex correlation coefficient* R_i^r from Equation 8 formalises the linear relation between local modifications in form of vertex displacements and performance values with respect to a chosen reference design. $\phi^{r,m} = f^m - f^r$ is the performance difference between two designs r and m. $\overline{\phi}^r$ is the mean value of the performance differences with respect to the reference design r.

$$R_i^r = \frac{\sum_{m=1, m\neq r}^{N} (\delta_{i,j}^{r,m} - \overline{\delta}_i^r)(\phi^{r,m} - \overline{\phi}^r)}{(N-1)\sigma_{\delta_i^r}\sigma_{\phi^r}} \tag{8}$$

$$\sigma_{\delta_i^r} = \sqrt{\frac{1}{N-1}\sum_{m=1,m\neq r}^{N}(\delta_{i,j}^{r,m} - \overline{\delta}_i^r)^2}, \qquad \overline{\phi}^r = \frac{1}{N-1}\sum_{m=1,m\neq r}^{N}\phi^{r,m}$$

$R_i^r > 0$ indicates that moving the vertex parallel to the normal vector is most likely to improve the performance of the design and vice versa. Again two situations lead to a vanishing R_i^r value. Firstly, the obvious explanation is that an (anti)-parallel modification of the vertex has no effect on the performance measure. Secondly, if the vertex is already located in an optimal position, every modification will reduce the performance and R_i^r will also be close to zero. In order to distinguish between the two cases, one could fit a linear model to the displacement and performance difference pairs and calculate the residual of the linear model. This residual provides information on the uncertainty of the correlation coefficient. Of course, the uncertainty of the correlation coefficient might also result from noisy data or non-linear relations between displacement measure and performance differences.

3.2.2 Vertex Sensitivity

In order to identify vertices that are sensitive to performance changes based on the whole data set without referring to one baseline shape, the Pearson correlation coefficient [12], is calculated based on all pairwise design comparisons. As $\phi^{r,m} = -\phi^{m,r}$, calculating the mean value for all performance differences results in $\overline{\phi} = 0$. We define the *overall vertex correlation coefficient* as follows (assuming again $\sum_{r=1}^{N}\overline{\delta}_i \approx 0$):

$$R_i = \cdot\frac{\sum_{r=1}^{N}\sum_{m=1,m\neq r}^{N}\delta_{i,j}^{r,m}\phi^{r,m}}{N(N-1)\sigma_{\delta_i}\sigma_\phi} \tag{9}$$

$$\sigma_\phi = \sqrt{\frac{2}{N(N-1)}\cdot\sum_{r=1}^{N}\sum_{m=r+1}^{N}(\phi^{r,m})^2}$$

The overall vertex correlation coefficient captures the linear relationship between the displacement and performance changes. In order to be less sensitive to outliers or noise in the data, it is reasonable to apply the Spearman rank based coefficient [12] instead of the Pearson correlation coefficient. Since the overall vertex correlation is linear, information is provided to distinguish between those vertices which are more likely to improve the performance by moving them parallel to the direction of the normal vector and those which improve the performance when moving them anti-parallel to the direction of the normal vector.

In aerodynamic design optimisation the interrelation between design modifications and performance changes is often highly non-linear. In order to capture also non-linearities, one could use information based measures like *mutual information* [6] to determine the sensitivity of vertices. The disadvantage of non-linear methods like mutual information is that the information to predict the direction of design improvement with respect to the normal vector is lost.

3.3 Modelling and Analysing Interrelated Deformations

For the calculation of the measures described above, the displacement of each vertex is considered independent of the others. Especially in aerodynamics, the interrelation between distant vertices or design regions and their joint influence on the performance plays an important role. In this section, special characteristics for the extraction of knowledge in form of associative rules based on data from unstructured surface meshes are discussed. The rules describe the relation between the displacement of distant vertices and their joint influence on the performance criteria. Modelling the interrelation between input variables is achieved by applying well known modelling techniques like Fuzzy rule induction, Bayesian networks, decision trees or others to the data set, for an overview of these techniques see e.g. [24].

3.3.1 Dimensionality Reduction

In general, the number of input parameters must be kept small for most modelling techniques in order to produce a small set of interpretable and manageable association rules. For the shape representation of section 1.2, the number of inputs equals the number of vertices n, which is large in practice. Therefore, a reduction of the number of input parameters is strongly required. In order to reduce the number of parameters it is reasonable to combine neighbouring vertices to form n_c design regions where $n_c \ll n$. During a design or optimisation process, it is unlikely that only single vertices are modified. Rather entire design regions of a certain extent are considered for shaping new designs.

Our suggested procedure for dimensionality reduction by means of identifying local design regions is summarised in Algorithm 1 and explained in more detail as follows. In a first step, vertices that do not seem to contribute on the performance of the designs are filtered out from the entire set of vertices. Therefore, the overall vertex correlation coefficient, R_i, is calculated for each of the n vertices as formulised in Eq. 9. The vertices which seem to have no impact on the performance value are removed by applying a threshold λ to R_i. Based on psychological research, done by Cohen [5], the threshold is set to $\lambda = \pm 0.3$. Cohen states that a small or no correlation is observed for a correlation coefficients with an absolute value smaller or equal to 0.3. The classification of the correlation coefficient by Cohen is used just as a rule of thumb but helps us to specify an appropriate threshold. The filtering based on the overall vertex correlation coefficient results in a set of vertices that can be separated into two sets $\mathcal{R}_+$ and $\mathcal{R}_-$. $\mathcal{R}_+$ contains vertices where $R_i > 0$ while $\mathcal{R}_-$ contains vertices with a correlation value $R_i < 0$. Finally, in order to form the desired design regions, a KMeans clustering algorithm [23] is applied to each of the two resulting sets of vertices. Performing the clustering on each set separately ensures that vertices that have a strictly different impact on the performance are assigned to different design regions. The final clustering splits the set of vertices based on a predefined distance measure into clusters. Generally, the (squared) Euclidean distance is used. Thus, distant vertices are assigned to separate clusters and neighbouring vertices to one and the same cluster (design region). The gap statistic [11] is used to overcome the problem of selecting an appropriate number of clusters in advance. Once the design regions have been identified the vertices closest to the cluster centres are considered for modelling and for the extraction of design rules.

Algorithm 1 Identification of design regions for dimensionality reduction

for i, $0 \leq i \leq n$ **do**
  calculate R_i based on Eq. 9
  if $+\lambda < R_i \leq 1.0$ **then**
    $\mathcal{R}_+ \Leftarrow \mathcal{R}_+ \cup \{\mathbf{v_i}\}$
  else if $-1.0 < R_i \leq -\lambda$ **then**
    $\mathcal{R}_- \Leftarrow \mathcal{R}_- \cup \{\mathbf{v_i}\}$
  end if
  $\mathcal{C}_+ \Leftarrow Gap(KMeans(\mathcal{R}_+))$
  $\mathcal{C}_- \Leftarrow Gap(KMeans(\mathcal{R}_-))$
  $\mathcal{C} \Leftarrow \mathcal{C}_+ \cup \mathcal{C}_-, \quad \mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}, \quad \mathcal{C}_i = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_i}\}$
  $\mathcal{CC} \Leftarrow ClusterCentres(\mathcal{C}), \quad \mathcal{CC} = \{\mathbf{v}_{CC1}, \mathbf{v}_{CC2}, \ldots, \mathbf{v}_{CCn_c}\}$

  return $\mathcal{CC}$
end for

3.3.2 Rule Induction

Rule induction is one of the fundamental and most often applied tools in the field of data mining and machine learning. Rules are easy to interpret by the engineer and hence raise his/her understanding of the system in hand. In aerodynamics the influence of one region of the design on the performance often strongly depends on the shape of the remaining design regions. Our driving force is to extract knowledge describing the complex relation between design regions and its performance number(s). It is important that the aerodynamic engineer is able to use the rules for the further development of new designs.

In the present framework rule extraction techniques are applied to the vertices closest to the centre of the design regions, $\mathbf{v}_{CCj} \in \mathcal{CC}$, that are the result of Algorithm 1. We basically distinguish between *absolute* and *relative design rules*. Absolute design rules describe the interrelation between the absolute positions of the vertices and their joint influence on the performance. An absolute design rule might look as follows:

$$IF \quad \mathbf{v}_{CC2_x} > 1.2 \quad AND \quad \mathbf{v}_{CC7_y} < -0.2 \quad THEN \quad f = 12.2$$

This rule states that a change in the x coordinate of the vertex $\mathbf{v}_{CC2}$ above 1.2 and a change in the y coordinate of vertex $\mathbf{v}_{CC7}$ below -0.2 is expected to result in a performance number of $f = 12.2$. Absolute design rules have two main disadvantages. Firstly, the number of parameters for rule extraction is three times the number of considered vertices, n_c. This directly increases the number of possible rules that are generated from the data set. Secondly, it is difficult to directly use the extracted rules to generate new designs due to the fact that the rules provide no information about the position of the vertices which are not used for modelling.

Design rules that are generated based on displacement data describe the interrelation of vertices and their influence on the performance relative to a predefined reference design. An example of a relative design rule is the following:

$$IF \quad \delta^{r,m}_{CC2} > 0 \quad AND \quad \delta^{r,m}_{CC7} < 0 \quad THEN \quad \phi^{r,m} < 0$$

From this rule it is expected that a modification of the vertex $\mathbf{v}_{CC2}$ towards its normal direction in conjunction with a modification of vertex $\mathbf{v}_{CC7}$ against the direction of the normal vector will result in a reduced performance value. For relative

design rules the number of parameters equals the number of vertices which are taken into account for rule extraction. Hence the number of possible design rules is small compared to absolute design rules. The reference design the relative rules relate to is usable as an initial shape for generating new designs. What means that the extraction of relative design rules can directly be integrated into the design process. Apart from the design rules, the elongation of the cluster centre provides a hint on the size of the design region considered for design modifications. Dependening on the requirements the parametric input for the design rules is real-valued or restricted to the sign of the displacement.

4 Application to 3D Turbine Blade Design Data

In order to demonstrate the feasibility of the theoretical concepts, the suggested methods have been applied to the data of an ultra-low aspect ratio transonic turbine stator blade of a small Honda turbofan engine. The data set, used for knowledge extraction, result from several computational optimisation runs where evolutionary strategies have been applied [9]. The different optimisations were driven by two objectives, considering the total mass averaged pressure loss and the maximum variation of the pitch-wise static outlet pressure. Other criteria like low pressure drag, low heat transfer, low material stress were not considered in the optimisation of the blade. In order to estimate the performance numbers, computational fluid dynamic (CFD) simulations have been applied. The CFD solver used is a parallelised 3D in-house Navier Stokes flow solver, called HSTAR3d [2]. The control point coordinates of a B-Spline representation of the blade geometry were used as parameters for the optimisation. For detailed information on the optimisation techniques and the parametrisation of the blade geometry, the reader is referred to [9] and [10].

In the present case the evolutionary optimisation runs provide the data for the data analysis. In general the suggested algorithms for knowledge extraction are not restricted to data from computational optimisations. The data also might come from manual optimisation processes where the performance has been determined under real physical conditions, e.g. with wind tunnel experiments. Furthermore, the analysis is independent from the geometric representation of the turbine blade, as long as the triangular surface mesh can be retrieved from the used representation. In the following analysis, the influence of the different turbine blade geometries on the total pressure loss ω is under investigation. The analysis techniques described in this paper are restricted to single performance numbers only. Multiple performance criteria can be considered by calculating a weighted sum of the different performance numbers. More information concerning the calculation of the pressure loss ω from the CFD simulation results are provided by Kuno and Sonoda [20].

Finally the data set under investigation contains 200 blade geometries and their corresponding total mass averaged pressure loss values.

4.1 The Turbine Blade Geometry

The structure of the axial gas turbine is rotationally symmetric. Eight equally shaped blades ($NB = 8$) are uniformally distributed around the hub of the turbine stator section. Thus, each of the 200 different stator blade geometries from the data set defines

together with the hub and the casing a complete stator section of the gas turbine. The axial length of the hub is the same for all blade geometries. Figure 4 a) illustrates the shape of the initial turbine blade. This turbine blade was the starting point for the different optimisations and has been selected as reference design for the following analysis. Additionally, in Fig. 4 the most important parts of the blade are named.

The parameters of the reference stator blade are those of a High Pressure (HP) turbine in the Honda HF118 turbofan engine. In the real engine the stator blade has an impingement cooling, but for research purposes the blade without cooling was used to demonstrate the feasibility of optimisation and knowledge extraction methods, focusing on the aerodynamic performance. The pitch to chord ratio of the blade is the inverse of the stage solidity which has a value of 0.706. The blade has a zero inlet metal angle, an isentropic exit Mach number of 1.04, an outlet metal angle of 72.80 degrees and the Zweifel loading coefficient has been chosen to be 0.80.

In a pre-processing step, the unstructured surface meshes of the 200 blade geometries have been generated. Given the B-Spline representation of the blades, the bounding surface of the blade geometry has been triangulated. In general the triangulation depends on the used representation and possibly depends on the application as well. In the present case a uniform triangulation has been chosen, where the minimum edge length of the triangles is large enough to represent all significant modifications of the 200 blade designs. The hub and tip cross-sections are ignored for the triangulation because there is no aerodynamic flow. The resulting triangulation of the reference blade is shown in Fig. 4 a). Each of the generated surface meshes consists of 1200 vertices. A normal vector is assigned to each vertex pointing towards the outside of the blade contour. Based on the unstructured surface meshes and their related performance numbers, all pairwise comparisons are generated. From the pairwise comparisons the displacement of corresponding vertices and the performance differences between blade designs have been calculated. The database subject to knowledge extraction comprises $200 \times 200 \times 1200$ displacement and 200×200 performance difference values.

4.2 An Engineering Approach to Knowledge Verification

The verification of the retrieved knowledge is essential in order to demonstrate the feasibility of the applied techniques. A standard way in machine learning is to split the entire data set into a training and a test or verification set. The application of this technique to the design data set is not advisable. The number of designs in the data set is often quite small. Thus, one cannot guarantee that the desired deformations (required to verify a certain hypothesis) are correctly represented in the test data set. In this paper, we apply systematic deformations to the reference blade and verify the expected results by re-calculating the performance of the deformed design. Each re-calculation requires the simulation of the fluid dynamics, which makes the verification more time consuming. Nevertheless, this kind of verification allows the test of hypotheses generated from the extracted knowledge.

In order to perform the desired deformations to the reference design *direct manipulation* of *free-form deformation* (DMFFD) has been applied [13, 25]. DMFFD is an extension of the standard *free-form deformation* (FFD) method [30]. It has two major advantages for the present task which will become apparent after a brief explanation of the method.

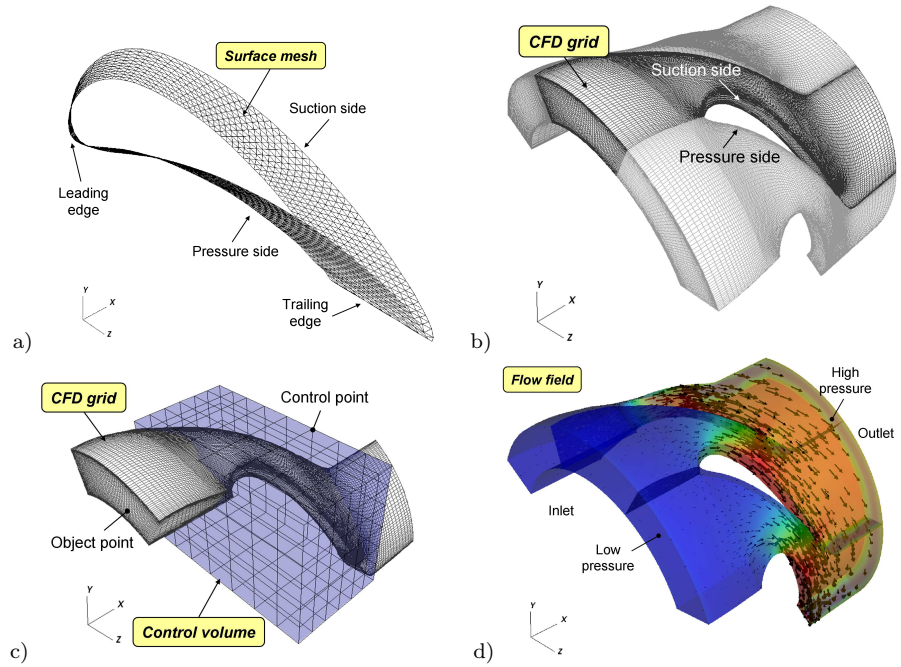

Fig. 4 Illustration of the reference blade with a) its triangular surface mesh, b) its CFD grid, c) the control volume which is needed to deform the blade surface as well as the CFD grid using DMFFD and d) the flow field resulting from the CFD simulation. The coordinate system is chosen so that x is the streamwise, y the pitchwise and z the radial direction. The illustration of the flow field shows the pressure loss (colour-coded) relative to the inlet total pressure as well as the velocity (vector field) of the stationary flow. The pressure loss ranges from 0.000 (blue) to 0.643 (red) and the velocity from almost 0.000 Mach (small arrow) to 1.296 Mach (large arrow).

FFD has been introduced in the field of computer graphics and represents variations of a baseline object. The object has to be embedded into a trivariate spline by transforming the object points to the spline coordinate system. The spline is defined by a control volume which has to be set up. By moving the control points of the spline the volume is deformed as well as all objects within the control volume.

In the direct manipulation of FFD the modifications are specified by the displacement of object handles instead of the control points. These object handles can be set arbitrary within the control volume. Based on their movement the positions of the control points are calculated.

In the present context, the DMFFD method is particularly suited to apply the extracted design rules to the reference design for two reasons. Firstly, highly sensitive vertices or vertices included in the premise of a design rule can be selected as object handles and displaced according to the rule. Furthermore, the adjacent vertices are displaced continuously and thus a displacement of a whole design region is realized. The size of these concurrently displaced areas depends on the granularity of the control lattice. Secondly, with respect to CFD a time-consuming re-meshing is omitted since the deformations affect the surface mesh as well as the CFD grid. For further details

on deformation methods and their application to shape and design optimisation the reader is referred to [26,25] and references therein.

The calculation of the quality of the deformed blade is done using the 3D flow solver HSTAR3d [2]. The CFD simulation requires the generation of a CFD grid which has been generated by discretising the space between two blades. Fig. 4 b) illustrates the CFD grid of the initial turbine blade. The flow is the same between every two blades. Hence one has to simulate the flow only for one out of eight sections. The CFD grid has to be deformed according to the modifications of the surface mesh of the blade. A control volume containing $7 \times 7 \times 9$ control points, Fig. 4 c), has been placed around the CFD grid. In order to verify the results from the knowledge extraction procedure, the following steps are carried out:

Step 1 Transform the vertices of the initial surface mesh (Fig. 4 a)) so that they are embedded in the control volume which encapsulates the CFD grid (Fig. 4 c)). Therefore, transform the vertices to their corresponding position related to the orientation of the CFD grid. This requires to group the vertices into those which belong to the suction side and those which belong to the pressure side of the blade.

Step 2 Once the transformation is done, select the desired vertices from the transformed surface mesh and determine the new position of the vertices based on the design rules.

Step 3 Adapt the control points representing the control volume so that the induced transformations of the DMFFD method describes the required transition to the new position of the vertices specified in Step 2.

Step 4 Deform the CFD grid based on the transformations done in Step 3.

Step 5 Calculate the flow (Fig. 4 d))and the aerodynamic performance based on the deformed CFD grid in Step 4.

4.3 Displacement and Sensitivity Analysis

Given the turbine blade data set the relative mean vertex displacement $\overline{\delta}_i^r$, the overall displacement variance σ_{δ_i} and the relative vertex correlation coefficient R_i^r are calculated. The values are colour-coded and mapped onto the surface of the blade, as illustrated in Fig. 5. $\overline{\delta}_i^r$ as well as R_i^r are calculated relative to the reference blade illustrated in Fig. 4. Fig. 5 a) provides a quick visual impression on the mean difference of the reference blade to the remaining blade designs of the data set. A negative $\overline{\delta}_i^r$ displacement indicates deformation towards the inside while positive values of $\overline{\delta}_i^r$ indicate deformations towards the outside of the blade contour. Looking at the displacement values at the suction side (Fig. 5 a) right) and the pressure side (Fig. 5 a) left) respectively, one can see that most blades within the data set are much thinner nearby the suction side leading edge and slightly thicker at parts of the pressure side involving the pressure side trailing edge compared to the reference blade. The values of the overall displacement variance, Fig. 5 b), highlight those regions which have been frequently modified and those which have not been deformed at all. The values are transformed to logarithmic scale to visualise also small variations. Vertices that are not deformed at all might be interesting for the designer who can model those untouched regions to test their influence on the performance of the blade.

The linear influence of the already deformed design regions on the performance is illustrated in Fig. 5 c). Blue regions indicate a negative correlation ($R_i^r < 0$) between

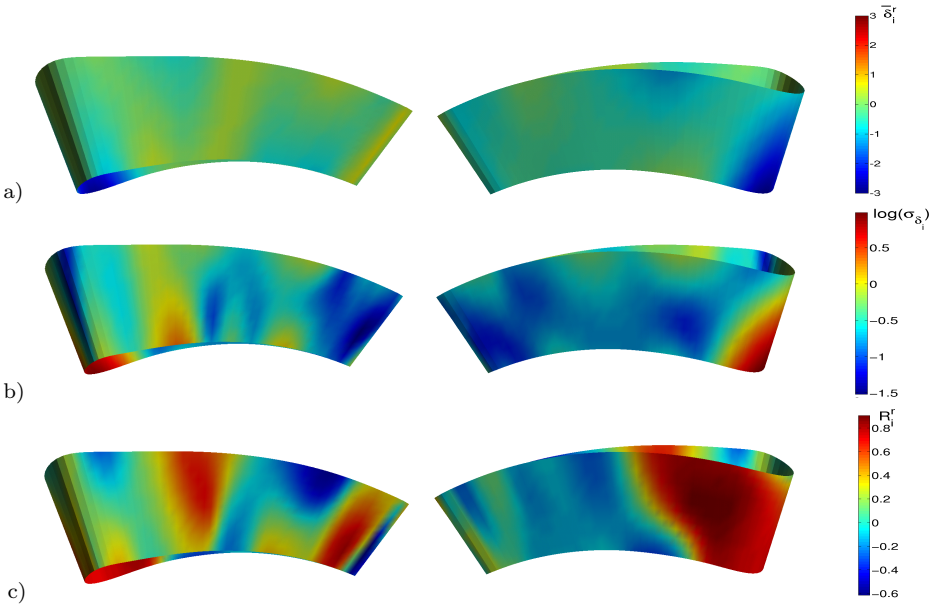

Fig. 5 Illustration of the results of the displacement and sensitivity analysis: a) relative mean vertex displacement, b) overall displacement variance and c) relative vertex correlation coefficient. The left column shows the pressure side and the right column the suction side respectively.

the displacement and the performance difference. This is interpreted as follows. Moving a vertex i from a bluish region along the normal direction ($\delta_i^{r,m} > 0$), towards the outside of the blade surface, will result in a decrease in pressure loss ($\phi^{r,m} < 0$) what leads to an increased blade performance. Positive correlations ($R_i^r > 0$) are visualised with a reddish colour. Positive correlations indicate that a displacement of the vertices towards the outside of the contour will result in an increase of the pressure loss and hence will lead to a decrease of the performance. The vertices around the leading edge of the blade seem to be highly sensitive to performance changes, considering a linear analysis. The sensitivity analysis lead to the conclusion that deforming the surface of the suction side at the leading edge towards the outside of the blade ($\delta_i^{r,m} > 0$) will increase the pressure loss ω and thus decrease the performance. Vice versa, deforming the leading edge to the inside of the blade is expected to increase the performance.

In order to directly verify whether the displacements of the vertices around the leading edge are linearly correlated to changes in the pressure loss, a vertex ($\mathbf{v}_{CC7}$, see Fig.7 b)) from the surface mesh close to the leading edge has been chosen and DMFFD has been applied. Thus, only vertices around the leading edge are deformed while the position of the remaining vertices remains constant. The displacement values $\delta_{CC7}^{r,m} \in \{-4, -3, -2, -1, 1, 2, 3\}$ are used. A value of $\delta_{CC7}^{r,m} = 1$ corresponds to a displacement of approximately 6% of the axial chordlength of the blade, measured at the hub section. The shape of the 3D surface mesh has been deformed based on DMFFD. Additionally, the differences in the pressure loss between the reference and the modified design $\phi^{r,m} = \omega^r - \omega^m$ are calculated. Fig. 6 summarises the results, which confirm the expected linear relationship between the deformation of the leading edge and the pressure loss. The reference blade and two experiments A and B are marked within

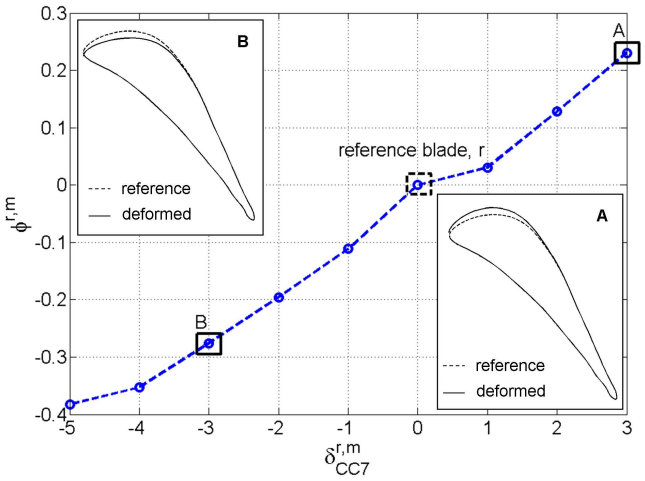

Fig. 6 Results of the verification, showing the linear relationship between the deformation of the leading edge and the resulting performance differences. A and B show the deformations of the blade at the 2D hub cross section for a displacement of 3.0 and −3.0 respectively.

Fig. 6. For A and B the deformations are visualised on the 2D hub cross-section given a displacement of $\delta_{CC7} = 3.0$ and $\delta_{CC7} = -3.0$ respectively.

4.4 Rule Induction

In order to extract meaningful design rules from the data set we first reduce the number of considered vertices as described in Algorithm 1. The sensitivity value R_i is calculated for each of the $n_r = 1200$ vertices. The threshold $\lambda = \pm 0.3$ is assigned to the correlation values in order to filter the most sensitive vertices from the blade surface. The two resulting sets of vertices $\mathcal{R}_+$ and $\mathcal{R}_-$ are illustrated in Fig. 7 a). Blue coloured regions highlight vertices with $R_i \leq -0.3$ while red coloured regions highlight those with $R_i \geq 0.3$. The remaining vertices ($-0.3 < R_i < 0.3$) are not considered for clustering and rule induction. KMeans clustering together with the gap statistic is applied to the two sets of sensitive vertices to generate the design regions. The vertices closest to the resulting centres of the design regions, $\mathbf{v}^r_{CCj}, j \in [1 \dots n_c]$, are marked on the blade surface which is shown in Fig. 7 b).

The sign of the displacement values from the vertices $\mathbf{v}^r_{CCj}$ as well as the sign of the performance differences $\phi^{r,m}$ are considered for the extraction of design rules. A decision tree is generated in order to retrieve a set of comprehensible design rules which are visualised in a tree like structure. The resulting decision tree is shown in Fig. 8. Because the decision tree is generated based on the displacement values it depicts relative design rules. The tree has been generated by a recursive partitioning of the input space with respect to the Gini index [24]. The Gini index is a measure of statistical dispersion and is used to select the attribute upon which to split at each branch of the tree. Each node of the tree lists information on the expected consequences specifying the likelihood for increasing or decreasing the performance number. The root node provides information on the distribution of the performance differences in the initial

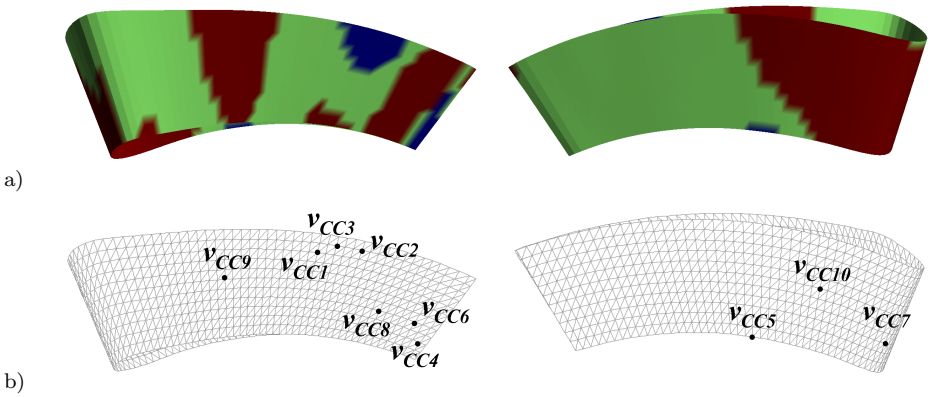

Fig. 7 Illustration of the results from dimensionality reduction. The filtered vertices are highlighted in a). A red colour is assigned to vertices with $R_i \geq 0.3$ while vertices with $R_i \leq -0.3$ are shown in blue. The vertices closest to the cluster centres, resulting from the clustering of the filtered vertices are marked in b).

data set. It can be seen that quite a large number of blade designs (84%) show a better performance than the chosen reference design. The branches represent a conjunction of vertex displacements that entail a certain consequence. In order to keep the number of rules manageable the growth of the tree has been limited to a predefined size. One advantage of decision trees is that they can easily be converted into a set of IF-THEN-rules. Exemplarily, the following two rules are extracted based on the branches marked with **A** and **B** in Fig. 8:

$$\mathbf{A}: IF \quad \delta^{r,A}_{CC8} < 0 \quad AND \quad \delta^{r,A}_{CC10} < 0 \quad AND \quad \delta^{r,A}_{CC7} < 0 \quad THEN \quad \phi^{r,A} < 0$$
$$\mathbf{B}: IF \quad \delta^{r,B}_{CC8} > 0 \quad AND \quad \delta^{r,B}_{CC7} < 0 \quad THEN \quad \phi^{r,B} > 0.$$

Rule **A** suggests a strategy, where the involved vertices are moved against the direction of their normal vector in order to decrease the pressure loss and hence to improve the quality of the blade. While rule **A** is expected, rule **B** seems to be more interesting. This rule indicates that an interrelated displacement of vertices around $\mathbf{v}_{CC8}$ towards the outside of the blade together with a displacement of the vertices around $\mathbf{v}_{CC7}$ to the inside of the blade will increase the pressure loss. This seems to be contradictory to the experiments which are done in section 4.3. There it has been shown that deforming $\mathbf{v}_{CC7}$ towards the inside of the blade decreases the pressure loss and consequently increases the performance of the blade. From this one can conclude that the latter statement only holds if $\mathbf{v}_{CC8}$ is not deformed towards the outside of the blade.

In order to verify the reliability of classification trees, DMFFD is applied in order to deform the reference blade. Two deformed designs are generated based on the rules **A** and **B**. With respect to rule **A**, the following displacements are assigned to the corresponding vertices on the reference design: $\delta^{r,A}_{CC8} = -2.0$, $\delta^{r,A}_{CC10} = -2.0$ and $\delta^{r,A}_{CC7} = -2.0$. DMFFD deforms the remaining vertices accordingly. The displacements of the deformed blade surface are illustrated in Fig. 9 a). It can be seen that apart from the vertex used as object handle the displacement nicely tapers off. The re-calculation

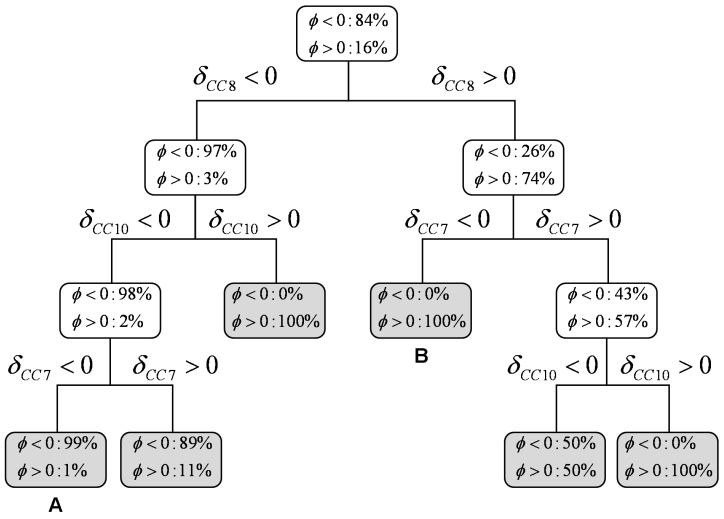

Fig. 8 Decision tree resulting from the 3D turbine blade data set. The vertices close to the cluster centres are used as features for modelling the tree. Each node contains the probabilities for increasing as well as decreasing the performance number. **A** and **B** mark branches (design rules) verified by means of DMFFD.

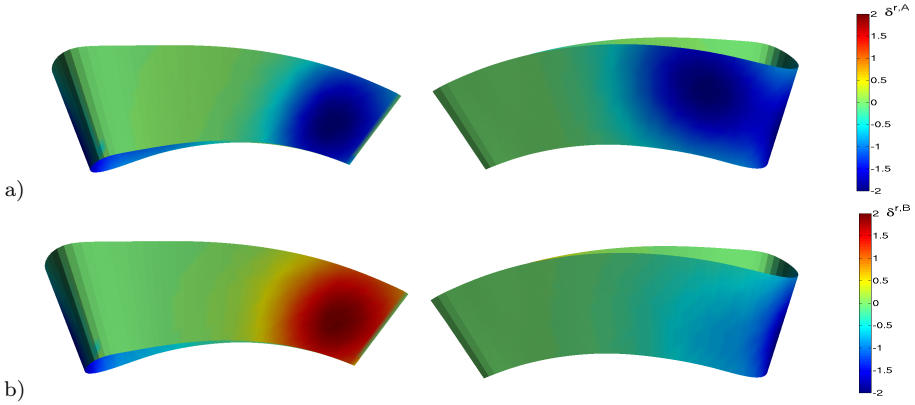

Fig. 9 Illustration of the applied deformations based on rule **A** a) and **B** b). The displacement values are colour-coded and mapped onto the surface of the reference blade.

of the performance of the deformed blade shows a reduction in the pressure loss of $\phi^{r,A} = \omega^r - \omega^A = -10.50\%$ compared with the performance of the initial blade. The result confirms the expected outcome predicted from the design rule.

For rule **B** the following displacements are assigned: $\delta^{r,B}_{CC8} = 2.0$, $\delta^{r,B}_{CC7} = -2.0$. The resulting deformations after applying DMFFD are illustrated in Fig. 9 b). The re-calculation of the performance results in an increase in the pressure loss of $\phi^{r,B} = 5.68\%$. This result confirms the prediction of the design rule **B** and consolidates our conclusions that a deformation at the leading edge only leads to a decrease of the pressure loss if the pressure side trailing edge is not thickened.

5 Summary

In this paper, we investigated the task of knowledge extraction from heterogeneous data sets that usually result from aerodynamic shape optimisation processes. The main contribution described is the formulation of a displacement measure that acts on a generalised shape representation – the unstructured surface mesh. Only a general shape representation allows to investigate design data independently from any of the representations used during the design and optimisation process. Based on the displacement and performance data set a framework has been presented that comprises a number of approaches for displacement analysis, sensitivity analysis, dimensionality reduction and rule extraction.

In order to demonstrate the feasibility of the suggested framework, we applied the proposed methods to a data set taken from the optimisation of a 3D turbine blade. Decision trees have been formulated to generate a set of comprehensive design rules which refer to a pre-defined blade design. The reliability of the used methods within the domain of aerodynamic design data has been demonstrated. The hypotheses retrieved from the sensitivity analysis and the decision tree have successfully been confirmed by actively deforming the reference blade using DMFFD and calculating their aerodynamic quality.

One main goal of further research is to use the information from the data set in order to improve the ongoing optimisation process, e.g. by specifying parameters of the optimisation algorithms or by increasing the generalisation capabilities and reducing the approximation errors of surrogate models [16].

6 Acknowledgement

The authors would like to thank T. Sonoda from the Honda Wako Nishi R&D Center and T. Arima from the Honda Wako Research Center for their valuable support regarding turbomechanical aspects of our work and for providing us the CFD simulation software HSTAR3d. Furthermore, the authors like to thank the members from NuTech Solutions, a Netezza company, for their valuable contributions to our work.

References

1. M. Alexa, Recent Advances in Mesh Morphing, *Computer Graphics Forum*, vol. 21, no. 2, pp. 173–197, 2002
2. T. Arima, T. Sonoda, M. Shirotori, A. Tamura, K. Kikuchi, A Numerical Investigation of Transonic Axial Compressor Rotor Flow Using a Low-Reynolds-Number $k - \epsilon$ Turbulence Model, *ASME Journal of Turbomachinery*, vol. 121, no. 1, pp. 44–58, 1999
3. L. Breiman, J.H. Friedman, R.A.Olshen and C.J. Stone, *Classification and Regression Trees*, California, USA, Wadsworth, Inc., 1984
4. Kazuhisa Chiba, Shinkyu Jeong, Shigeru Obayashi and Hiroyuki Morino, Data Mining for Multidisciplinary Design Space of Regional-Jet Wing, *IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2333–2340, 2005
5. J. Cohen, *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1988
6. T. M. Cover, J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., Wiley Series in Telecommunication, 1991
7. M. Garland, P. S. Heckbert, Surface Simplification Using Quadratic Error Metrics, *Proceedings of the 24th annual Conference on Computer Graphics and Interactive Techniques*, pp. 209–216, 1997

8. N. Hansen, A. Ostermeier, Completely Derandomized Self-adaptation in Evolutionary Strategies, *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001
9. M. Hasenjäger, B. Sendhoff, T. Sonoda and T. Arima, Three dimensional aerodynamic optimization for an ultra-low aspect ratio transonic turbine stator blade, in *Proceedings of the ASME Turbo Expo*, ASME Paper No. GT2005-68680, 2005.
10. M. Hasenjäger, B. Sendhoff, T. Sonoda and T. Arima, Three dimensional evolutionary aerodynamic design optimisation using single and multi-objective approaches, in *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*, EUROGEN, Munich, 2005.
11. T. Hastie, R. Tibshirani, G. Walther, Estimating the Number of Clusters in a Dataset via the Gap Statistic, Tech. Rep., Dept. of Statistics, Stanford University, 2000
12. T. Hill, P. Lewicki, STATISTICS Methods and Applications, StatSoft, Tulsa, OK, 2007
13. W. M. Hsu, J. F. Hughes, H. Kaufman, Direct Manipulation of Free-Form Deformations, *International Conference on Computer Graphics and Interactive Techniques*, in Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, pp. 177–184, July, 1992
14. Y. Ito and K. Nakahashi, Surface Triangulation for Polygonal Models Based on CAD Data, *Intern. Journal for Numerical Methods in Fluids*, vol. 39, no. 1, pp. 75–96, April 18th, 2002
15. V. Jain and H. Zhang, Robust 3D Shape Correspondence in the Spectral Domain, *IEEE Intern. Conf. on Shape Modeling and Applications, SMI'06*, pp. 118–129, June 14th-16th, Matsushima Japan, 2006
16. Y. Jin, A Comprehensive Survey of Fitness Approximation in Evolutionary Computation, *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005
17. Y. Jin, *Knowledge Incorporation in Evolutionary Computation*, Springer, 2005
18. A. J. Keane and P. B. Nair, *Computational Approach for Aerospace Design*, Wiley, 2005
19. T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, Heidelberg, 1995
20. N. Kuno, T. Sonoda, Flow Characteristics in a Transonic Ultra-Low-Aspect-Ratio Axial Turbine Van, *Journal of Propulsion and Power*, vol. 20, no. 4, pp. 596–603, AIAA, 2004
21. P. Larranaga, J. A. Lozano, *Estimation of Distribution Algorithms*, Kluwer Academic Publisher, 2002
22. P. Laskov, C. Kambhamettu, Comparison of 3D Algorithms for Non-rigid Motion and Correspondence Estimation, *Proceedings of the British Machine Vision Conference*, BMVC 2001, Manchester, UK, 2001
23. S. P. Lloyed, Least Squares Quantization in PCM, *IEEE Transaction on Information Theory*, vol. 28, no.2, pp. 129–137, 1982
24. *The Data Mining and Knowledge Discovery Handbook*, Edited by Oded Maimon and Lior Rokach, Springer, 2005
25. S. Menzel, M. Olhofer, B. Sendhoff, Direct Manipulation of Free Form Deformation in Evolutionary Design Optimisation, *International Conference on Parallel Problem Solving From Nature*, PPSN, ed. T. P. Runarsson et al., Springer, pp. 352–361, 2006
26. S. Menzel, B. Sendhoff, Representing the Change – Free Form Deformation for Evolutionary Design Optimisation, *Evolutionary Computation in Practice*, ed. T. Yu, D. Davis, C. Baydar, R. Roy, Springer Berlin, 2008
27. R. H. Meyers and D. C. Montgomery, *Response Surface Methodology*, Wiley-Interscience, 1995
28. S. Obayashi and D. Sasaki, Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map, *In Proc. Second Intern. Conf. on Evolutionary Multi-Criterion Optimization*, EMO 2003, vol. 2632/2003, pp. 71, Faro, Portugal, 2003
29. J. Samareh, A Survey of Shape Parametrization Techniques, *Intern. Forum on Aeroelasticity and Structural Dynamics Conf.*, 1999
30. T. W. Sederberg, S. R. Parry, Free-form deformation of solid geometric models, *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151–160, 1986
31. J. R. Shewchuk, *Delaunay Refinement Mesh Generation*, Ph.D. thesis, Carnegie-Mellon Univ., School of Computer Science, 1997
32. K. Shimada, Anisotropic Triangular Meshing of Parametric Surfaces via Close Packing of Ellipsoidal Bubbles, *Intern. Journal of Computational Geometries and Applications*, vol. 10, no. 4, pp. 400–424, 2000
33. Y. Wang, B. S. Peterson, L. H. Staib, Shape-based 3D Surface Correspondence using Geodesic and Local Geometry, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 644–651, 2000