

# **Unsupervised Extraction of Design Components for a 3D parts-based Representation**

**Zdravko Bozakov, Lars Gräning, Stephan Hasler, Heiko  
Wersing, Stefan Menzel**

**2008**

**Preprint:**

This is an accepted article published in Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN). The final authenticated version is available online at: [https://doi.org/\[DOI not available\]](https://doi.org/[DOI not available])

# Unsupervised Extraction of Design Components for a 3D parts-based Representation

Zdravko Bozakov, Lars Graening, Stephan Hasler, Heiko Wersing and Stefan Menzel

**Abstract**—During CAD development and any kind of design optimisation over years a huge amount of geometries accumulate in a design department. To organize and structure these designs with respect to reusability, a hierarchical set of components on different scalings is extracted by the designers. This hierarchy allows to compose designs from several parts and to adapt the composition to the current task. Nevertheless, this hierarchy is imposed by humans and relies on their experiences. In the present paper a computational method is proposed for an unsupervised extraction of design components from a large repository of geometries. Methods known from the field of object and pattern recognition in images are transferred to the 3D design space to detect relevant features of geometries. The non-negative matrix factorization algorithm (NMF) is extended and tuned to the given task for an autonomous detection of design components. The results of the NMF additionally provide an overview on the distribution of these components in the design repository. The extracted components sum up in a parts-based representation which serves as a base for manual or computational design development or optimisation respectively.

## I. INTRODUCTION

WHEN designing products for the automotive or aerospace domain it is crucial to find geometries with optimal properties. Traditionally, manual design optimisation is a resource and timeconsuming process performed by large teams of engineers and designers. While physical tests can be partially replaced by computational simulation tools such as Finite Element (FE) or Computational Fluid Dynamics (CFD) solvers, design optimisation makes up a significant part of the development process, hence substantially increasing the development costs. Therefore, it is desirable to devise fully autonomous design optimisation environments supporting design development and, ideally, evolving the final shape of a product with no or minimal human interaction.

A crucial aspect is the choice of the representation of the object that is the target of optimisation. Recently, for complex designs deformation methods have received considerable attention [1], [2], [3]. Instead of parameterising the design directly, e.g. by spline curves/surfaces, transformations that are applied to an initially chosen base

design are encoded. This has the great advantage that the number and distribution of parameters is decoupled from the geometrical description. State-of-the-art free form deformation (FFD), which was originally developed for soft object animation, has been successfully applied in an evolutionary design optimisation framework to a stator blade of a jet turbine [4], [5]. However, despite of the various advantages of this technique, the search space of the algorithm is quite restricted to the area in which the initially chosen design is located. In particular topological changes to the design like holes are practically impossible to realize. To overcome this disadvantage a parts-based representation embedded in the FFD environment has to be developed which allows exchanging components of the design during the runtime of the optimisation. Consequently, the question arises of how to establish a repository of components on which the optimisation algorithm can rely on. Naturally, a repository of components is not only helpful in automated processes but also in manual design development.

Assuming a given large database consisting of 3D designs, which have been created during CAD development or design optimisation, in a straightforward way an experienced user could try to scan through it, select representative designs and decompose them into meaningful components manually. Consequently, this process is very timeconsuming and the results depend highly on the experience of the user as well as his definition of a component. To decouple this process from the users experience, in the present paper a framework is proposed which originates from the field of object recognition and pattern detection in images. The method strives for a fully autonomous and unsupervised process delivering a repository of components in a parts-based representation [6].

In the field of object recognition various algorithms have been proposed that can extract the building blocks of image sets. These components are used to represent the individual images more effectively and their presence helps to predict an image's content (object label). Depending on the concrete scenario, the algorithms can be constrained to look for components with more global or more local character. Some local methods, like the non-negative matrix factorization (NMF) [7], find components that directly reflect the constituent parts of an object and are therefore called parts-based methods. Because these methods usually do not make assumptions on the dimension of the data and because of the obvious analogy between a rigid object and a 3D design,

Zdravko Bozakov was with the Department of Electrical Engineering and Information Technology at the TU Darmstadt, Germany.

Lars Graening, Stephan Hasler, Heiko Wersing and Stefan Menzel are with the Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, D-63073 Offenbach/Main, Germany (e-mail: {lars.graening, stephan.hasler, heiko.wersing, stefan.menzel}@honda-ri.de).

these methods can be directly applied to the identification of the relevant parts of a 3D design in a pixel-like representation.

The paper is organized as follows. In section II a brief literature review is given on state-of-the-art algorithms for shape analysis and decomposition. Section III describes thoroughly the non-negative matrix factorization and its variations in the field of object recognition. A framework for the unsupervised extraction of components is presented in detail in section IV, followed by concluding remarks.

## II. REVIEWING ALGORITHMS FOR 3D SHAPE ANALYSIS AND SHAPE DECOMPOSITION

The process of extracting meaningful design parts from 3D shapes is not trivial. This is partly due to the fact that no universal definition of a design part exists. A human's idea of salient *features* which make up a part might strongly differ depending on the problem being evaluated. Computer based systems, on the other hand, derive completely different notions of a part. In the following, a brief overview of common 3D shape analysis and feature extraction techniques is presented. For further details the reader is referred to [8].

### A. 3D Shape Descriptors

The definition of adequate shape descriptors and the extraction of design features which encapsulate the required characteristics of a design is the main challenge for analyzing a repository of 3D designs. Design features may include statistical design properties such as moments, circularity or compactness. Alternative methods that use histograms of geometric statistics or statistics of frequency decomposition of shapes are mainly used for shape retrieval tasks. As an example, in [9] harmonic shape descriptors are suggested which use the amplitudes of spherical harmonic coefficients to generate a rotation invariant shape representation. Related to the task such representations are tuned for fast computing but miss the representation of details of the design. Furthermore, the resulting coefficients are hard to interpret for human beings.

### B. Clustering 3D Shapes

For partitioning a design repository it is often desirable to classify the content into a set of groups or clusters based on the similarity of feature vectors or shape descriptors. This is useful for determining distinct types of designs contained in a set as well as analyzing the distribution thereof.

A popular method which performs this type of clustering is the *K-Means* algorithm [10]. Given a number of  $N$ -dimensional data-points this iterative algorithm attempts to separate the data into  $K$  distinct clusters. The method is not guaranteed to converge to a global optimum. In fact the results of the algorithm strongly depend on the initial values. However, since the algorithm is very fast, it is common to execute it multiple times and select the results from the best run. A major drawback of the algorithm is the fact that the

number of clusters must be known in advance, thus making it unsuitable for unsupervised problems. Furthermore, the result of the clustering algorithm strongly depends on the choice of the correct metric for calculating the similarity between the shapes.

The *Principal component analysis* PCA, is a well established and widely used statistical technique for discovering the main features, or *principal components*, within complex data sets. The method is closely related to singular value decomposition. PCA has been successfully applied to various fields such as statistical analysis, lossy and lossless data compression, and face recognition. Given a set of  $N$ -dimensional points the PCA aims to re-express the data by finding a linear transformation of the coordinate system which results in an optimal representation of the data set in eigenvectors. Noise and redundancy are filtered out revealing hidden structure or features within the data. If all eigenvectors are used, a lossless reconstruction of the data set can be performed. However, if only the eigenvectors corresponding to the largest eigenvalues are used the number of dimensions required to represent the data set can be greatly reduced, at the cost of introducing only a small error.

### C. Segmentation of 3D Designs

For our application it is necessary to analyse repositories based not only on overall shape similarity, but also on specific features or parts contained within objects.

An approach called *convex decomposition*, initially developed for two dimensional polygons, has been used for decomposing three dimensional shapes. The aim is to segment the shape into a minimal number of convex polygons. The method has been studied extensively for 2D problems and optimal solutions have been suggested. A theoretical framework for the 3D case has been outlined in [11]. Unfortunately, the use of exact convex decomposition is only feasible for relatively simple shapes due to high computational costs. For shapes containing holes the problem is nondeterministic polynomial-time (NP) hard. Moreover, for complex polyhedrons, the decomposition typically results in a vast number of components, which are practically unmanageable. To alleviate this problem, approximate convex decomposition schemes have been proposed in [12] and [13], which decompose shapes into polygons which are allowed to exhibit a limited amount of concavity or group convex sub-components iteratively.

*Geometrical skeletons* can be used to represent geometries using a set of line segments. A graph-based representation can then be extracted from the skeleton. If a large number of skeletons are available, methods from graph theory are applied to find matching sub-graphs, which represent individual components present in multiple geometries [14].

An approach for shape decomposition that uses *fuzzy clustering for object segmentation* and *cuts* to extract regions corresponding to features called patches is presented in [15]. Based on facet distance information, a probability is

calculated that two facets belong to a certain patch. Since the method is computationally expensive, a decomposition of large models is accomplished by generating a simplified model on which the decomposition is performed and projecting the resulting patches onto the original model.

#### D. Image-based Feature Extraction

Due to the similar target of detecting patterns and relevant features in the field of object recognition and image processing, in the present paper we focus our analysis on the non-negative matrix factorization (NMF). NMF provides several characteristics that are also important for decomposing 3D objects into components. Based on a pixel representation, the NMF scans a database of images and decomposes the data set into relevant non-negative features and a coefficient matrix. The objective is to reconstruct the given database by these two matrices with a minimal error. Hence, this method fulfils two requirements for transferring it to the 3D object space. It allows an unsupervised execution of the algorithm and results in non-negative features, i.e. interpretable parts of the designs. The basic concept of the algorithm is described in detail in the next section.

### III. THE NON-NEGATIVE MATRIX FACTORIZATION FOR OBJECT DETECTION AND PATTERN RECOGNITION

#### A. Non-negative matrix factorization

The NMF algorithm is a powerful technique for decomposing large image repositories into their salient parts. The method gained popularity after it was presented in [7] for parts-based face recognition (see Fig. 1). The data set of faces is decomposed into relevant parts, like different kinds of noses, eyes, eyebrows etc. The activation matrix contains detailed information of how to combine the extracted features to reconstruct the original face image with a minimal error.

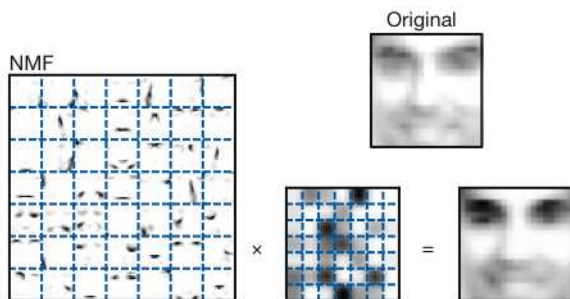


Fig. 1. Non-negative matrix factorization applied to face image data [7]. The feature and coefficient matrix allows the reconstruction of a face image which is contained in the original data set with minimal error.

Similarly to PCA, NMF can be used to reveal latent features in complex data sets. Unlike other rank reduction or matrix factorization techniques, such as PCA, independent component analysis (ICA) or vector quantization (VQ) a non-negativity constraint is imposed on the factor matrices. For many applications this facilitates the interpretation of the

features. A major advantage is that the algorithm generates additional information about the distribution of the parts in the database. As a drawback the number of desired base or feature vectors  $F$  must be specified beforehand.

The NMF objective is formulated as follows: given a non-negative matrix  $X \in \mathbb{R}^{N \times I}$  which contains in each column one of the  $N$ -dimensional images of the data set, find two non-negative matrices  $W \in \mathbb{R}^{N \times F}$  and  $H \in \mathbb{R}^{F \times I}$  in such a way that the product of both reconstructs the original data set with minimal square Euclidean distance.  $W$  denotes the feature matrix and  $H$  the coefficient matrix.

$$X \approx WH, \quad \|X - WH\|_2^2 \rightarrow \min \quad (1)$$

Initially two multiplicative algorithms for computing the NMF were proposed in [16] which have become a benchmark for the performance of all subsequently derived NMF variations. These algorithms are closely related to the method of gradient descent. In this paper, we will focus on the NMF based on the minimization of the square Euclidean distance between the original image set and the reconstruction which can be understood as a measure for the quality of the factorization or the reconstruction error (RE). Since it is not possible to compute a solution for  $W$  and  $H$  simultaneously, the objective function is solved by alternately applying the update rules (2) until convergence is reached. The elements of the matrices are initialized with random positive values. This initialization combined with the multiplicative nature of the algorithm guarantees that non-negativity is enforced. The multiplication in the update rule and the division are performed element-wise.

$$H_{ij} \leftarrow H_{ij} \frac{(W^T X)_{ij}}{(W^T WH)_{ij}}, \quad W_{ij} \leftarrow W_{ij} \frac{(XH^T)_{ij}}{(WHH^T)_{ij}} \quad (2)$$

To avoid divisions by zero, a small number  $10^{-9}$  is usually added to the denominator of the update rules. While the cost function (1) is convex in either  $W$  or  $H$  alone, it is not convex for both variables together. Therefore, the factorization produced by the NMF algorithm is not unique but can be any kind of stationary point, e.g. a local optimum or a saddle point.

#### B. Alternative NMF Formulations and Extensions

Several extensions have been derived from the NMF which are now explained in more detail. Enforcing orthogonality of the NMF base vectors, by introducing the constraint  $W^T W = I$ , results in the *Orthogonal NMF (ONMF)* which targets to extract non-overlapping features. The data set is decomposed into distinct parts which are physically interpretable. An extension of the NMF algorithm which incorporates this constraint has been presented in [17]. The constraint is imposed either in the feature or coefficient matrix or simultaneously in both.

An algorithm called *Local NMF (LNMF)* has been introduced in [18] which focuses on an improvement of the extraction of localized features. This extension of the standard algorithms became necessary because the parts-based character of the base vectors postulated by Lee and Seung is rarely achieved in practice. The algorithm targets three additional goals. Firstly, in order to minimize the number of feature vectors required to reconstruct the data set, the features should not unnecessarily be broken down. Secondly, in order to minimize redundancy among the feature vectors maximum orthogonality of the base vectors is required. The last constraint, maximum expressiveness, aims at enhancing the detection of features which contribute most to the reconstruction.

To improve the quality of detected features, an *NMF with Sparseness Constraints* has been proposed in [19] and [20]. The activation of sparse vectors follows a probability density which is “highly peaked at zero and has heavy tails” [19]. In other words the vector elements contain mostly zero values and only a few non-zero values. Enforcing sparseness results in a decomposition which uses only a few active elements to represent the entire data set and facilitates the interpretation of the derived features. For 3D design decomposition, a high degree of sparseness prevents the emergence of partially activated feature vector elements which do not have a physical interpretation. Similarly, sparseness of the coefficient matrix will result in nearly binary activations of the base vectors. Several methods have been devised to introduce a degree of sparsity on top of the original NMF formulation, which usually involve penalty coefficients which must be set manually. In [20] a desired degree of sparseness is enforced by augmenting the original NMF with a projection step.

A simple and yet very effective method for introducing sparsity in both the feature and the coefficient matrices was suggested in [21]. The original NMF factorization is extended to  $X \approx WSH$  by a constant, symmetrical matrix  $S$  which strongly depends on a smoothness factor  $\omega \in [0, 1]$ . For  $\omega = 0$  the method equals the standard NMF described above. Because  $S$  is constant the factorization can be written as  $X \approx (WS)H = W(SH)$  depending on whether the feature or coefficient matrix is currently being evaluated. Due to the multiplicative nature of the NMF, smoothing the feature vector will require the coefficient matrix to compensate the introduced error. This can only be achieved by adding zero values to the row vectors of the coefficient matrix, in other words  $H$  must become sparse. Conversely, smoothing the coefficient matrix will enforce sparsity in the feature vectors.

#### IV. A FRAMEWORK FOR THE UNSUPERVISED EXTRACTION OF COMPONENTS IN THE 3D DESIGN SPACE

This section focuses on the development of a framework for extracting relevant components from 3D objects. Before discussing the problem of determining the optimal number of features, the turbine test scenario is introduced.

##### A. Scenario: Decomposition of a Virtual Turbine

To evaluate the performance of the existing NMF algorithms in the context of design optimisation, a test scenario has been used. In this scenario turbine-like designs have been generated consisting of a central cylinder plus a variation of up to seven blades attached to it at seven possible distinct positions. Two types of blades were created, a straight and a bend one. All geometries were given in the standard STL-file format containing a triangulated 3D surface. Three example designs are depicted in Fig. 2.

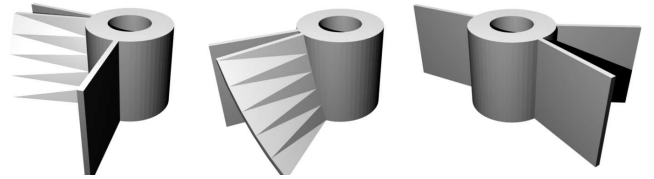


Fig. 2. Samples of base turbine designs.

To generate a large database of differently shaped turbines, five base turbines were defined. Each turbine was embedded in a free form deformation control volume which allowed small shape variations of the blades in tangential direction. From each base turbine 100 designs were generated by applying small random modifications to the blades. All 500 turbines were presented to the algorithm which had to calculate the optimal decomposition of the designs, i.e. the minimal number of parts which is required to reconstruct the database with a minimal RE.

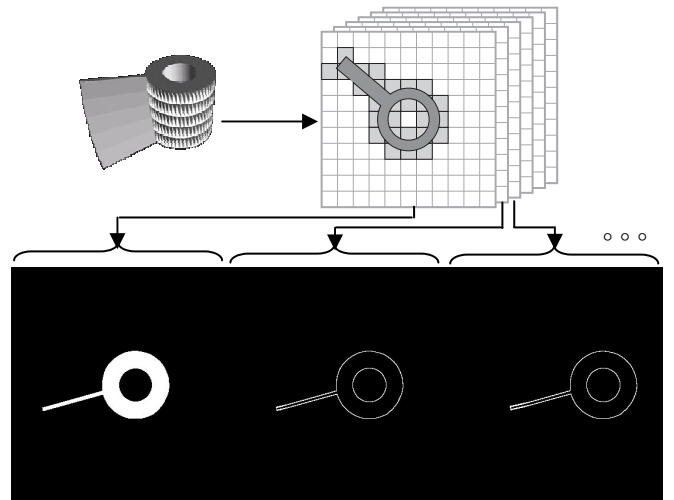


Fig. 3. Transformation of 3D designs into voxel space.

In order to apply image based feature recognition techniques to the 3D domain, polygon mesh based designs are converted to a voxel based representation. The resulting voxel model can be interpreted as a binary 2D image by reshaping the 3D voxel matrix into a series of slices and connecting them as depicted in Fig. 3. Thereafter, the application of 2D NMF algorithms becomes straightforward.

## B. Optimal Number of Features

The number of base vectors greatly influences the quality of the extracted features, as well as the feasibility of applying the NMF algorithms to a specific task in general. Since the method proposed in this paper aims at achieving a fully unsupervised design decomposition, it is essential to eliminate the need of a user interaction to specify the number of parts which should be extracted as it is required in the standard NMF algorithm [7].

One approach that we tested consists of finding a correlation between the RE and the specified number of features. Given a sufficiently large number of base vectors, the NMF algorithm should reconstruct the data set with an RE close to 0. Therefore, the idea is to execute multiple NMF runs, incrementing the number of feature vectors each time. It is expected, that the RE will improve significantly after each run until the optimal feature number is reached. After this point, adding more feature vectors should contribute only slightly for a further improvement of the RE. As a result a drop in the RE/feature curve should be noticeable which corresponds to the optimal feature number. Unfortunately, tests showed that implementing this method for complex data sets is difficult. The curves are quite smooth, making an unsupervised estimate of the feature number nearly impossible [6].

It has been observed that when specifying a large number of base vectors many of the tested NMF algorithms produced duplicate features containing only minor variations. The phenomenon occurs because all unconstrained NMF algorithms attempt to minimize the RE by fully utilizing all available base vectors. This observation motivates the idea to scan for similarity  $A = W^T W$  between the base vectors during the execution of the algorithm and reduce the number of features when redundancy is detected. The feature reduction step is either applied when the algorithm converges or after a predefined iteration number. For values of  $A_{i \neq j}$  which exceed a predefined similarity threshold  $T_{sim}$  the features are considered redundant, corresponding features are combined and the corresponding coefficient row is deleted. This method produces acceptable results, however some problems remain. Firstly, the optimal time at which the feature reduction step is performed is difficult to determine. If the reduction step is performed too early, too many vectors might be removed limiting the number of features found. Naturally, a suitable similarity threshold must be specified.

An alternative method for finding the optimal feature number is to incorporate strong sparsity constraints into the used algorithm. Ideally, the penalty added to get a sparse coefficient matrix should prevent features which are not absolutely necessary from being activated, i.e. the coefficient rows corresponding to these base vectors should contain only zero values. Experiments showed that the *nsNMF* [21] approach for enforcing sparsity produces very good results. The smoothing matrix can be easily incorporated into a wide range of NMF algorithms and sparseness is added

simultaneously to the coefficient matrix and the base vectors preventing the generation of partially activated feature vector elements. It is possible to adjust the amount of enforced sparseness by varying the parameter  $\omega$ . However, for the 3D model decomposition task at hand, a value of  $\omega = 0.5$  was found to produce very good results for a wide range of data sets. As a consequence, only an upper limit of the feature number must be selected. Base vectors which are not necessary to reconstruct the data set are automatically set to zero by the algorithm [6].

## C. Dilation of 3D Designs to reduce Misalignments

A problem arises when dealing with voxel images while using the Euclidean distance as a similarity measure. Due to the nature of the data set which includes a large number of voxel images with numerous minor deformations, many parts of the design may not overlap. While a human observer would classify two features translated by several pixels in respect to each other as belonging to the same group, mathematically they have a similarity value close to zero. The problem is depicted in Fig. 4.

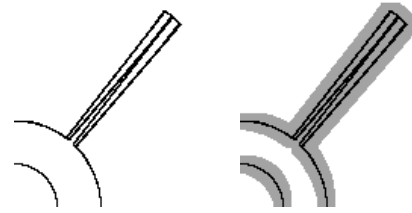


Fig. 4. Misaligned voxel designs and effect of dilation.

To ensure that similar features are detected as a single module by the NMF algorithm, a 3D morphological dilation step is applied to the voxel data set by iteratively dilating each design with a  $3 \times 3 \times 3$  kernel until a user specified maximum dilation number is reached. The dilation number is reduced during runtime when the algorithm converges. This reduction process is repeated several times, in order to extract thin features. As a consequence, the quality of extracted features improved. Nevertheless, it should be noted that the dilation number is highly task dependent. It has to be adapted according to the occurring design variations in the database and the chosen resolution of the voxel space [6].

## D. Method Prerequisites

In this section, an NMF based algorithm for analyzing large repositories containing geometrical data is presented. The proposed algorithm aims at extracting a minimal number of distinct sub-components present within the data set. The base vectors are not segmented more than absolutely necessary. By imposing an orthogonality constraint on the base vector set, the extracted features are unique and do not overlap, resulting in a minimal degree of redundancy within the feature-set.

As physical parts cannot be represented using partially activated voxels, ideally the extracted features should contain only binary values. This attribute is also crucial for

the transformation of the extracted features from the voxel domain back to a polygon based representation. The binarity constraint is enforced by requiring sparsity of the feature vectors and intermediate normalization steps.

Similarly, a very high degree of sparseness is imposed on the coefficient matrix in order to eliminate partly activated features. Moreover, this way the number of used base vectors is minimized.

Several algorithms satisfy one or more of the requirements mentioned above. The LNMF method [18] and feature orthogonal NMF algorithms in [17] enforce orthogonality constraints while simultaneously minimizing the number of active values in the matrices. However, sparsity is enforced only indirectly and not in the binary sense described above. Sparsity constraints can be added to either the feature or the coefficient matrix using a projected gradient approach as formulated in [20]. Although this approach produces adequate results, the constraints cannot be easily added to both factors at the same time. Additionally, to keep the amount of human interaction to a minimum it is desirable that the number of parameters to be configured is kept to a minimum. Algorithms which for example require a tuning of the step-size are therefore not feasible.

Summarizing, based on numerous experiments [6] our proposed method uses the orthogonal NMF [17] augmented by a smoothing matrix  $S$ , as it has been suggested in [21] for the original NMF. It was found that this combination produces results which are very close to the optimal case. At the same time, the implementation and derivation of the algorithms is straightforward.

#### E. Derivation of the Sparse Orthogonal NMF

In the following we propose an algorithm for the unsupervised extraction of components of 3D designs. To begin the derivation of the update rule for the feature matrix, we construct an extended objective function. This objective function which results in equation 3 is constructed using the Euclidean norm based NMF cost function extended by the constant smoothness matrix  $S$ , the orthogonality penalty term  $W^T W - I = 0$  and a Lagrangian multiplier  $\lambda$ . While the derived approach bares similarity to the tri-factorization proposed for clustering in [17], in our case  $S$  is a constant matrix which is used to impose sparseness in  $W$  and  $H$ . Orthogonality is enforced only in the feature matrix and the update step for the smoothing matrix vanishes. The derivation is analogous to the approach in [17] augmented by a smoothing matrix  $S$ .

$$\begin{aligned} L &= \|X - WSH\|_2^2 + \lambda \text{Tr}[W^T W - I] \\ &= \text{Tr}[(X - WSH)^T (X - WSH)] + \lambda \text{Tr}[W^T W - I] \end{aligned} \quad (3)$$

Using the gradient descent approach outlined by Lee and Seung the update rule for  $W$  is derived.

$$W \leftarrow W - \varepsilon \partial L / \partial W \quad (4)$$

Next, the derivative of  $L$  with respect to  $W$  is calculated and  $\varepsilon$  is set so that the additive components are eliminated.

$$\partial L / \partial W = -2XH^T S + 2WSHH^T S + 2W\lambda \quad (5)$$

$$\varepsilon = W / (WSHH^T S + \lambda W) \quad (6)$$

Substituting the equations above into equation 4 results in the update rule for  $W$ .

$$W_{ij} \leftarrow W_{ij} \frac{XH^T S}{W(SHH^T S + \lambda)} \quad (7)$$

The Lagrangian multiplier  $\lambda$  is approximated using the Karush-Kuhn-Tucker (KKT) condition  $\nabla_W L(W, \lambda) = 0$ . To ensure the non-negativity of the update, the inequality  $SHH^T S + \lambda \geq 0$  must hold, which is the case for:

$$-XH^T S + WHH^T S + W\lambda = 0 \quad (8)$$

$$-W^T XH^T S + W^T WHH^T S + W^T W\lambda = 0$$

$$\text{with } W^T W \approx I$$

$$W^T XH^T S - ISHH^T S = \lambda$$

Substituting  $\lambda$  into (7) results in the final form of the update rule. Additionally, the square root of the right hand side of (7) is taken in order to ensure convergence [17].

$$W_{ij} \leftarrow W_{ij} \sqrt{\frac{(XH^T S)_{ij}}{(W(W^T XH^T S))_{ij}}} \quad (9)$$

Because orthogonality of the coefficient matrix is not required, the standard NMF update step for  $H$  is utilized.  $W$  is substituted with  $WS$  to incorporate the sparsity constraint.

$$H_{ij} \leftarrow H_{ij} \frac{((WS)^T X)_{ij}}{((WS)^T (WS)H)_{ij}} \quad (10)$$

The initialization of the coefficient matrix and the base vectors has a strong influence on the results of the NMF. The effects of initialization have been summarized in [22]. The most basic approach is the initialization of both matrices with random non-negative values. However, this method typically leaves traces of noise in the feature vectors when factorizing binary voxel images. For our case a homogeneous initialization of the feature matrix with  $W_{ij} = 1$  provides very good results. An even more suitable alternative is to apply a fast clustering algorithm such as *K-means* to the input data, using each clusteroid to initialize a feature matrix column. As a consequence, all input voxel images are contained in one of the base vectors and therefore the probability that a feature is missed is minimized. On the other hand, large portions of each feature image which do not contain any information are fixed to zero, thus speeding up convergence. The likelihood of premature convergence to

a local optimum of the algorithm is greatly reduced. For the coefficient matrix a random initialization with a lower bound constraint  $H_{ij} > 0$  was used. The complete algorithm used in this paper is outlined in the following. The symbol  $\odot$  denotes element-wise multiplication, the symbol  $\oslash$  denotes element-wise division.

---

**Algorithm 1:** Sparse Orthogonal NMF Algorithm.

---

```

nz ← 1 × 10-20
D ← maxDilation
load image-set X
XD ← dilate3D(X,D)
L2 ← (I ⊙ (XDTXD))0.5
XD ← XDL2-1
initialize feature matrix W columns using K-means
initialize coefficient matrix H with random values
initialize smoothing matrix S according to α
repeat
  XD ← dilate3D(X,D)
  L2 ← (I ⊙ (XDTXD))0.5
  XD ← XDL2-1
repeat
  Ws ← (WS)T
  H ← H ⊙ (WsXD) ⊙ ((WsWsT)H + nz)
  Hs ← HTS
  W ← W ⊙ ((XDHs) ⊙ (W(WT(XDHs) + nz))0.5
  L2 ← (I ⊙ (WTW))0.5
  H ← L2H
  W ← WL2-1
until convergence
D ← D - 1
until (D < 0)

```

---

*F. Feature Extraction Results of a Turbine Test Scenario*

The tests presented in this paper were conducted on five turbine-like design groups. A cross-section showing the top of a sample of each design group is depicted in Fig. 5.

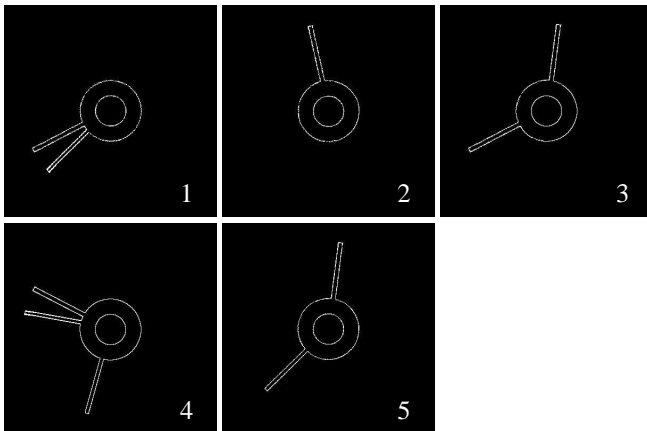


Fig. 5. Cross section at the top of the five base turbines.

For each design group 100 slightly deformed versions were generated using the standard FFD technique as described above resulting in a total number of 500 designs. To apply our proposed framework two parameters have to be specified explicitly. The maximum number of feature vectors was initially set to 15 and as smoothing factor a value of  $\omega = 0.5$  was used, as it indicated to provide robust results for different test setups. The convergence of Algorithm 1 is generally assumed when the RE ceases to decrease significantly, e.g. the first 3 digits do not change for 10 iterations. With respect to computation time, typical runs took approximately 2 hours on an AMD Opteron 250, 2.4 GHz system with 4 GB memory. The 3D results produced by the sparse orthogonal NMF combined with multistage dilation are depicted in Fig. 6. During runtime of Algorithm 1 the initial number of 15 features has automatically been reduced to 5. All 5 orthogonal features necessary to reconstruct the input data set with a minimal RE are extracted correctly. Moreover, the modules contain almost no artefacts and no voxels with low activation which strengthens the interpretability of each feature. The central cylinder is not extracted explicitly since it is present in all designs without any variation. As a result of the sparsity constraint only a minimal number of features is activated in the coefficient matrix partially shown in Fig. 7.

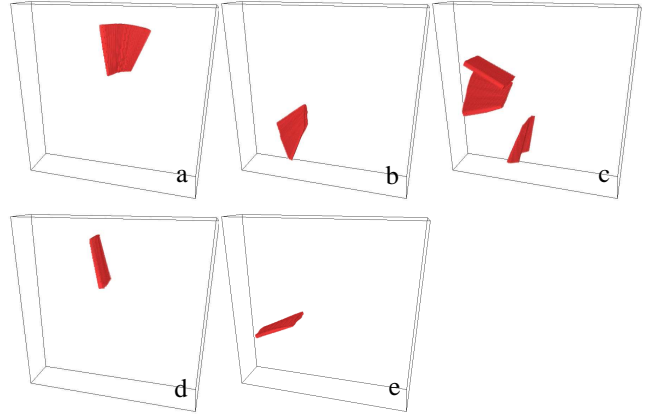


Fig. 6. Extracted features resulting from the Sparse Orthogonal NMF.



Fig. 7. Coefficient matrix corresponding to the extracted features in Fig. 6.

In Fig. 7 each activated pixel in each 15 pixel-sized column indicates which features are used to reconstruct a certain design. The level of activation is given by the brightness of the pixel colour. Hence, e.g. designs similar to base turbine 1 are composed of features b and e, and those close to turbine 3 of features a and e. To confirm the independency of the algorithm from random initialisations the scenario has been run multiple times. In all tests the same



features were found underlining the good reliability of the proposed method.

## V. CONCLUSIONS

In the present paper, we focused on a transfer of methods which have been developed in the field of object recognition to the field of 3D design development. The similar approach of an unsupervised detection of meaningful patterns in 2D image data sets motivated our proposed method for decomposing 3D designs. The NMF characterised by creating a parts-based representation consisting of non-negative components has been favoured. We propose a sparse orthogonal NMF algorithm, an augmentation of the orthogonal NMF with a constant smoothing matrix, which results in slightly modified update rules. This technique has been applied to voxelized 3D models. It has been shown that this method allows an unsupervised and autonomous decomposition of large numbers of 3D models stored in a data repository into salient components. As parameters, mainly the maximum number of features and the smoothing factor  $\omega$  have to be specified. For  $\omega$  we suggest 0.5 which has proven a good reliability. In addition to the decomposition of salient components, the presented method generates also information about the distribution of the parts within the database which allows an identification of outliers.

Furthermore, some constraints that are vital for a correct and meaningful decomposition are highlighted. These include feature vector orthogonality and a high degree of sparseness in both the feature and coefficient matrices. As a result, the problem of determining the correct number of parts is largely eliminated and only an upper limit on the number of desired features must be specified. Finally, the importance of voxel model dilation was emphasized in the context of module extraction which minimizes the number of features necessary for a complete representation of a data set.

Some aspects of the methods presented in this paper remain to be evaluated in future works. The possibility of incorporating rotation and position invariance of the extracted features [24], the handling of outliers and the coupling of local real world performances to local design features are some of them. Finally, the automatized integration of components into an FFD design optimisation framework remains the main target.

## ACKNOWLEDGMENT

The authors would like to thank Julian Eggert and Bernhard Sendhoff from the Honda Research Institute Europe GmbH for their valuable support during the project.

## REFERENCES

- [1] T. W. Sederberg and S. R. Parry, "Free-Form Deformation of Solid Geometric Models," in *Computer Graphics*, 20(4), pp. 151-160, 1986.
- [2] S. Coquillart, "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," in *Computer Graphics*, 24(4), pp. 187-196, 1990.
- [3] E. C. Perry, S. E. Benzley, M. Landon and R. Johnson, "Shape Optimization of Fluid Flow Systems," in *Proceedings of ASME FEDSM'00. 2000 ASME Fluids Engineering Summer Conference*, Boston, Massachusetts, 2000.
- [4] S. Menzel, M. Olhofer and B. Sendhoff, "Application of Free Form Deformation Techniques in Evolutionary Design Optimisation," in *Proceedings of the 6th World Congress on Multidisciplinary and Structural Optimisation*, Rio de Janeiro, 2005.
- [5] S. Menzel, M. Olhofer and B. Sendhoff, "Direct Manipulation of Free Form Deformation in Evolutionary Design Optimisation," in *Int. Conference on Parallel Problem Solving From Nature (PPSN)*, edited by T.P. Runarsson et al., Springer-Verlag, pp. 352-361, 2006.
- [6] Z. Bozakov, "Unsupervised Component Extraction for Design Optimisation using Feature Analysis," Master Thesis, Control Theory and Robotics Lab, Institute for Automatic Control, TU Darmstadt, 2007.
- [7] D. D. Lee and S. H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788-791, 1999.
- [8] T. Funkhouser, M. Kazhdan, P. Min and P. Shilane, "Shape-based Retrieval and Analysis of 3D Models," in *Communications of the ACM*, Vol. 48, No.6, pp. 58-64, June 2005.
- [9] M. Kazhdan, T. Funkhouser and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," in *Symposium on Geometry Processing*, pp. 167-175, June 2003.
- [10] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297, 1967.
- [11] B. M. Chazelle, "Convex decomposition of polyhedra," in *STOC '81: Proceedings of the thirteenth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, pp. 70-79, 1981.
- [12] J.-M. Lien and N. M. Amato, "Approximate convex decomposition of polygons," in *SCG '04: Proceedings of the 20th annual symposium on Computational geometry*, NY, USA: ACM Press, pp. 17-26, 2004.
- [13] J.-M. Lien and N. M. Amato, "Approximate convex decomposition," in *SCG '04: Proceedings of the 20th annual symposium on Computational geometry*. NY, USA: ACM Press, pp. 457-458, 2004.
- [14] H. Sundar, D. Silver, N. Gagvani, and S. Dickenson. Skeleton based shape matching and retrieval. In *SMI 2003*, pages 130-139, 2003.
- [15] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*. NY, USA: ACM Press, pp. 954-961, 2003.
- [16] D. D. Lee and S. H. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, pp. 556-562, 2000.
- [17] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA: ACM Press, pp. 126-135, 2006.
- [18] S. Li, X. Hou, H. J. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," *IEEE CVPR*, 2001.
- [19] P. O. Hoyer, "Non-negative sparse coding," in *Proc. IEEE on Neural Networks for Signal Processing*, pp. 557-565, Feb. 2002.
- [20] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457-1469, 2004.
- [21] A. Pascual-Montano, J. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, March 2006.
- [22] A. N. Langville, C. D. Meyer, R. Albright, J. Cox and D. Duling, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," *NCSU Technical Report Math 81706*, 2007.
- [23] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman and D. Dobkin, "A search engine for 3D Models," in *ACM Transactions on Graphics (TOG)*, pp. 83-105, January 2003.
- [24] J. Eggert, H. Wersing and E. Körner, "Transformation-invariant representation and NMF," in *Proc. of the IEEE International Joint Conference on Neural Networks IJCNN*, pp. 2535-2539, 2004.