Visual Human+Machine Learning

Raphael Fuchs Jürgen Waser Meister Eduard Gröller

Abstract— In this paper we describe a novel method to integrate interactive visual analysis and machine learning to support the insight generation of the user. The suggested approach combines the vast search and processing power of the computer with the superior reasoning and pattern recognition capabilities of the human user. An evolutionary search algorithm has been adapted to assist in the fuzzy logic formalization of hypotheses that aim at explaining features inside multivariate, volumetric data. Up to now, users solely rely on their knowledge and expertise when looking for explanatory theories. However, it often remains unclear whether the selected attribute ranges represent the real explanation for the feature of interest. Other selections hidden in the large number of data variables could potentially lead to similar features. Moreover, as simulation complexity grows, users are confronted with huge multidimensional data sets making it almost impossible to find meaningful hypotheses at all. We propose an interactive cycle of knowledge-based analysis and automatic hypothesis generation. Starting from initial hypotheses, created with linking and brushing, the user steers a heuristic search algorithm to look for alternative or related hypotheses. The results are analyzed in information visualization views that are linked to the volume rendering. Individual properties as well as global aggregates are visually presented to provide insight into the most relevant aspects of the generated hypotheses. This novel approach becomes computationally feasible due to a GPU implementation of the time-critical parts in the algorithm. A thorough evaluation of search times and noise sensitivity as well as a case study on data from the automotive domain substantiate the usefulness of the suggested approach.

Index Terms—Interactive Visual Analysis, Volumetric Data, Multiple Competing Hypotheses, Knowledge Discovery, Computer-assisted Multivariate Data Exploration, Curse of Dimensionality, Predictive Analysis, Genetic Algorithm

1 Introduction

Due to the increasing pace and complexity of modern simulation systems, engineers are often confronted with huge data sets that are difficult to make sense of. Usually, the results contain multivariate sets of physical attributes such as temperature or fluid velocity, often with hidden relations between them. Even for domain experts, it can be tedious to search for explanations for features in the data. In the past, machine learning (ML) algorithms have been adapted to assist in the exploration process. In the context of database systems, we find methods that produce logic rules which are valid for many items in the database. Based on labels inside the data set, these methods can build classifiers from the data. However, these methods require knowledge that is contained in the data itself. That is, the machine learns from the data only, producing hypotheses that are restricted to given relations between data items inside. Therefore these techniques are often not applicable in the field of engineering simulations and computational fluid dynamics (CFD). The simulation simply does not generate labels for the data items and a classification can only be found interactively during visual analysis. A mechanism is thus needed to let engineers add information that is based on their knowledge and experience. Interactive visual analysis (IVA) supports the formation of flexible user selections. The main idea of IVA is to depict various attributes using multiple views and to allow the engineer to interactively select (brush) a subset of the data in these views. All corresponding data items in linked renderings are highlighted as well, providing the analyst with information about the interplay of the attributes involved. As an example, consider a simulation of an industrial prototype where regions of high temperatures are found. In the temperature attribute view, the engineer selects the critical value ranges. During visual inspection in linked views, the user finds that the areas of critical temperature are related to regions of high vorticity values and low velocities. This way, the user has formed the hypothesis that 'critical temperature' corresponds to 'high vorticity values' AND 'low fluid velocities'.

- *Raphael Fuchs is with ETH Zurich, E-mail: raphael@inf.ethz.ch.*
- *Jürgen Waser is with VRVis Vienna, E-mail: jwaser@vrvis.at.*
- *Meister is with TU Vienna, E-mail: groeller@cg.tuwien.ac.at.*

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org .

When we present the general concepts of IVA to our application partners from research and industry, we are often confronted with scepticism. We can formulate the critical issues in the following questions (see also Figure 1):

1. How do I know if my selection is the only one that relates to the spatial feature? Are there alternative explanations?

2. Is there a reasonable hypothesis that corresponds to a spatial feature I am interested in?

Considering the infinite number of possible selections and logical combinations thereof, it soon becomes clear that this criticism is justified. This is often referred to as the *curse of dimensionality* for IVA which complicates reasoning when many attribute dimensions are available. To overcome these problems we propose a combination of IVA and machine learning. See Figure 2 for an illustration. We present a set of fitness criteria which filter the best hypotheses out of the vast search space using a genetic algorithm. While searching, both the machine and the human user are able to learn from and contribute to the current set of hypotheses. We present an interactive framework with adequate visualizations that enable the user to analyze and modify the generated hypotheses before feeding them back into the machine learning algorithm.

2 Related Work

Shneiderman [27] suggests to combine information visualization with data mining and gives additional recommendations for future research: novel methods should allow the user to specify what he is looking for, results should be easily reportable and the human responsibility should be respected. Keim et al. [14] describe the visual analytics process to be characterized by interaction between data, visualizations, models about the data, and the users in order to discover knowledge. They emphasize the importance of the human in the data analysis process. Seo and Shneiderman [26] present the rank-by-feature framework to find features in high-dimensional data. Keim [15] provides a taxonomy of visual data mining techniques and gives an insightful overview of visual data exploration techniques. He suggests that the generation of hypotheses can be done either by automatic techniques from statistics or with ML or visually. In this paper we show that these approaches can all work together.

Hao et al. [8] have presented a powerful method to find attributes correlating to a selected feature. They propose queries which allow to select hot spots within one attribute and to find correlated attributes.

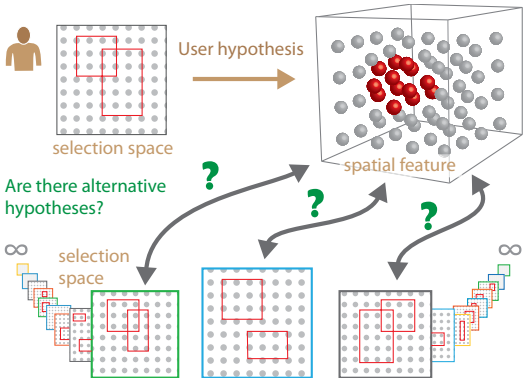

Fig. 1. Curse of dimensionality for IVA. In the space of all possible hypotheses it is not clear if the current one is the only candidate. This is a common problem engineers encounter when they use standard IVA methods (linking and brushing) for making sense of multidimensional simulation results. Secondly, searching for a hypothesis to explain a given spatial feature can be almost impossible for the user.

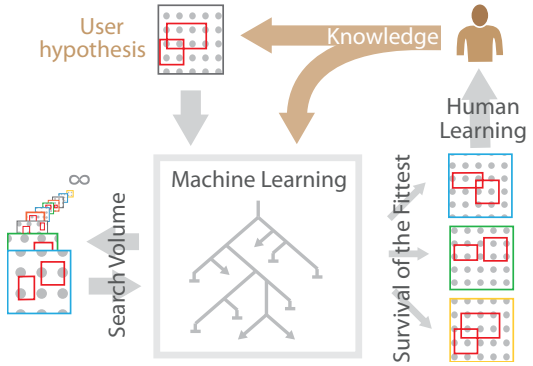

Fig. 2. We propose the extension of human learning with machine learning, comprising a genetic algorithm that efficiently searches for the best hypotheses available.

Yang et al. [31] present a system to extract hidden features in a data set interactively. They define distance metrics for the user hypotheses to filter out similar hypotheses using clustering. Müller et al. [20] discuss how principal component analysis can enhance the visualization with a focus on scatterplot matrices. Doleisch et al. [6] present the feature definition language which we have adopted for the interactive generation of user hypotheses. Kehrer et al. [13] apply this approach to create insight into weather simulation data. Rautek [24] uses a fuzzy rule base for the automatic generation of visualization parameters in volume renderings.

Hertzmann [10] gives an introduction to ML suitable for the visualization researcher. Chen [5] believes that the integration of ML and information visualization is a fruitful approach. In this spirit, Rossi [25] suggests to favor user intervention and control during ML over direct interaction with the visualization. Ward [30] describes how visual clues allow users to monitor dimension reduction and clustering techniques in order to check whether important information has been removed. Jänicke et al. [11] have presented an approach to find relevant regions inside large and complex data sets based on statistical complexity.

Laine [16] discusses three criteria for machine learning to be applicable in an industrial environment. These are supervised operation, robustness and understandability. He stresses that statistical significance is no measure for relation to the problem of interest and presents examples where statistical properties are not sufficient for knowledge generation. Ma [18] discusses seminal work that allows the user to select samples of relevance from which a machine learning algorithm builds a classifier which then segments the rest of the data (see also [28, 29]).

Another direction of research focuses on the improvement, verification or substantiation of data mining results with visualization and interaction. These works show how classifiers can be improved or analyzed by interaction and visualization. Examples are: decision tree construction, support vector machine algorithms and association rules - all improve when supported by an IVA environment [1, 22, 4, 19, 23].

3 Visual Learning

In this section we describe different aspects of the visual framework which allow to monitor and steer the ML algorithm. We start with a general overview and give an example. Then we discuss several important facts which are relevant when combining human and machine learning. Details are discussed in later sections.

The genetic algorithm can be considered to be learning from the population of available hypotheses which encapsulate the generated information about the data set. When the user or the machine interact with the hypotheses in the population, using editing, optimization or combined operations, the population grows while generating novel information. Based on fitness evaluation or user interaction, the low performing hypotheses or parts thereof are removed which is also an important learning step.

We structure the presented approach according to different types of interaction:

- Genes: When working with selections the user creates novel information which is treated by the algorithms as genes. In the presented framework a single selction is a gene. Handling of genes is discussed in Subsection 3.2.
- Hypotheses: Complex hypotheses are composed of selections. A hypothesis can be evaluated for each data item and linked to the volume rendering for inspection. Based on information on other hypotheses in the population and on his own background knowledge, the user can analyze the influence of the selections and recombine and modify the hypotheses before reiterating the heuristic search. The interaction with hypotheses is discussed in Subsection 3.3.
- Fitness: To exercise a global type of control over the development of the population, the user is able to specify different parameters such as the weights for the computation of fitness values (see Subsection 3.4).
- Optimization: It can be very difficult to maximize the fitness of a hypothesis interactively. In Subsection 3.5, we describe a solution to this problem and explain why interaction is still necessary.
- Global Parameters: These parameters steer the ML algorithm and control the rate of mutation, recombination and selection (see Section 5.

3.1 Considerations on Learning

In this section we present some facts we consider relevant when human and machine learning are to be combined.

Confirmation bias Cognitive science shows that there is a tendency to search for new information in a way that confirms the current hypothesis and to irrationally avoid information and interpretations which contradict prior beliefs [12].

No free lunch From theory we know that there is no universally good search/learning configuration [7]. Changing one parameter may improve the performance for some problems, but usually reduces it for others. Human knowledge is required to specify good parameters and to steer the machine learning algorithm into the right direction. For example the selection in Figure 3 can be locally optimal, but the user understands that disabling parts of the hypothesis makes sense nevertheless.

Strengths and weaknesses The most important strength of the human user is his ability to understand the problem at hand. Therefore he

can often decide when to perform directed search (optimization) and how to limit the search space. Furthermore, based on the visualization, he is often able to extract information from noisy or incomplete data. The user can give meaning to the features detected in the data. The machine on the other hand has many capabilities the user lacks: its work is inexpensive, fast and tireless. It is able to perform large searches and we know that heuristic search algorithms can perform quite well in large and unstructured search spaces.

No Labels Flow features can be fuzzy for multiple reasons. Often, their boundaries are not sharp. Larger features (such as vortices) are composed of smaller structures which can be features themselves. When exploring simulation data it is not pre-determined which data ranges are important. Only after the engineer has gained understanding of the situation we obtain labels for the data (e.g. 'too hot').

User Requirements The three central requirements are control, robustness and understandability. Control enables the engineer to define what he considers interesting and to steer the machine learning process in the right direction. Robustness is the insensitivity to small errors or deviations from assumptions. The third requirement is understandability. This means straightforward visualization of results and their textual representation to facilitate reporting.

3.2 Visual Genes

We call the smallest item which carries information a gene. In the present context, we consider a single one-dimensional fuzzy selection a gene. That is a tuple of the form $(fuzzy_{min}, full_{min}, full_{max}, fuzzy_{max}, a_i)$, where the first four values define a fuzzy selection and a_i the relevant attribute. Detailed information about selections is visible in small scatterplot icons. See Subsection 4.1 for a formal definition. These icons act as toggle buttons to allow deactivation of all selections in one attribute view. E.g. in Figure 3 (3) two genes have been switched off. For more specific interactions, the user can enlarge these scatterplots to edit, move, add or remove selections. In the first steps of data exploration, one of the most important tasks is to find the data attributes which are relevant for the problem at hand. Also, when a very large number of attributes is available, the heuristic algorithm will learn faster when only relevant attributes are considered. Therefore, we augment the user interface by a simple color-based clue describing which *attributes* are common in the gene pool. For each attribute we compute a frequency value and integrate this information as the background color into the display of hypotheses. For example in Figure 3 (2) the important attributes a_0 and a_1 are frequent in the gene pool and receive darker coloring than the others.

3.3 Visual Hypotheses

A hypothesis is a fuzzy-logic combination of selections. See Subsection 4.1 for a formal definition. Each hypothesis is visualized as a horizontal frame of interactive attribute views. Disjunctivly combined views (clauses) are placed side-by-side without separating space (see Figure 3 (2)). Clauses themselves are placed in separate frames (see Figure 3 (5)). A conjunction of clauses forms a hypothesis. The user hypothesis is indicated by a small icon on the left and further highlighted by a different background color. This way, the hypotheses in a population can be viewed and compared. To specify novel genes, selections from linked views can be transferred to the user hypothesis. To analyze spatial properties of a given hypothesis, the user can display the resulting fuzzy selection in a linked 3D rendering, where fuzzy membership values can be included into the transfer function, e.g. as a mapping of the degree of selection to opacity or color.

3.4 Visual Fitness

The goal is to find hypotheses that lead to features which are similar in physical space and different in selection space. These hypotheses should be as simple as possible. We therefore require measures for feature similarity, complexity and individuality to define the fitness of a hypothesis. See Section 4 for detailed explanations. Setting the weights of these measures provides control over what to look for at the current learning step.

High fitness values may result from good performance in one or several of the three components feature similarity, individuality and complexity. Since this is important information these three components are shown as different shades of green in the fitness bar to the left of each hypothesis. For example in Figure 3 the increase in fitness between (2) and (3) is due to the lower complexity (lowest dark green bar), whereas the difference between (3) and (4) is due to the better match between the features.

3.5 Visual Optimization

A crucial weakness of optimization algorithms which lack the support from background knowledge is the fact that they often cannot discriminate between local and global optima. Therefore they are prone to get stuck at local optima where the fitness of a hypothesis is relatively high but far from a global optimum. The question whether a local optimum is meaningful or not in the current context cannot be answered by the machine. On the other hand, even if including the local neighborhood only, an extensive search often can not be performed by the human user. For these reasons, we add an optimize button to each clause such that the user can decide when and where to search. We have implemented a simple hillclimbing operator which performs a gradient ascent towards the local optimum: The operator moves each boundary element of every selection and calculates the fitness for all changes. From all possible modifications, the one with the maximal improvement is accepted. This is done for decreasing step widths until the best improvement falls below some user specified margin. For example, consider a one-dimensional non-smooth selection $S = (s_{min}, s_{max})$ contained in a hypothesis. For a given step δs, the fitness of the hypothesis changes when S is replaced by $S_1 = (s_{min} - \delta s, s_{max})$, $S_2 = (s_{min} + \delta s, s_{max})$, $S_3 = (s_{min}, s_{max} - \delta s)$ or $S_4 = (s_{min}, s_{max} + \delta s)$.

3.6 The Vis 09 example

This example is based on a synthetic data set containing a well distinguishable feature ('Vis09') which is related to multiple attributes in the data (see Figure 3). Over a 3D domain, we define ten scalar attributes. Voxels outside the feature have random attribute values in the $[0,1]$ interval. The data values inside the feature are chosen such that the feature is describable by two different hypotheses. The first hypothesis comprises a selection on attributes a_0 and a_1. The second hypothesis is defined among attributes a_2, a_3, a_4, a_5, a_6 and a_7 which have to be combined in a specific way. The 'Vis' part of the feature can be selected by brushing the a_2 vs. a_3 scatterplot in conjunction with a_6 vs. a_7. The '09' part can be selected by brushing the a_4 vs. a_5 scatterplot in conjunction with a_6 vs. a_7. Neither a_2 vs. a_3 nor a_6 vs. a_7 alone result in a useful hypothesis. Therefore the 'Vis09' feature is of the form $((a_2 \wedge a_3) \vee (a_4 \wedge a_5)) \wedge (a_6 \wedge a_7)$. Such a feature is almost impossible to find for the human user, even though it is still rather simple from the viewpoint of the machine. The synthetic data set contains a third feature defined in attributes a_8, a_9 which is easily found and selected interactively. It has been arranged to contain the neighborhood of the 'Vis09' feature (see Figure 3(1)). In Figure 3(1) the user has figured out that attributes a_8 and a_9 might describe a feature of interest. This is a common situation where one attribute describes roughly what an engineer is looking for (such as low pressure is roughly related to vortices). The resulting feature in Figure 3(1) is not very sharp though. In Figure 3(2), the user runs the ML algorithm to search for alternative hypotheses. The best resulting hypothesis has low fitness but the user is able to see that the machine has found something valuable. In Figure 3(3) the user deactivates genes on other attributes by toggling the corresponding parts of the machine hypothesis, and the fitness increases. In Figure 3(4) the user has found a novel hypothesis. Interactive visual analysis allows the user to relate the novel hypothesis to the reference hypothesis and generate a good explanation even though there is some noise left.

At this point, the user has gained some information. However, it is still unclear if there are alternative hypotheses and so he triggers another search. In Figure 3(5) the algorithm reveals an alternative hypothesis. In Figure 3(6) the alternative consists of two clauses com-

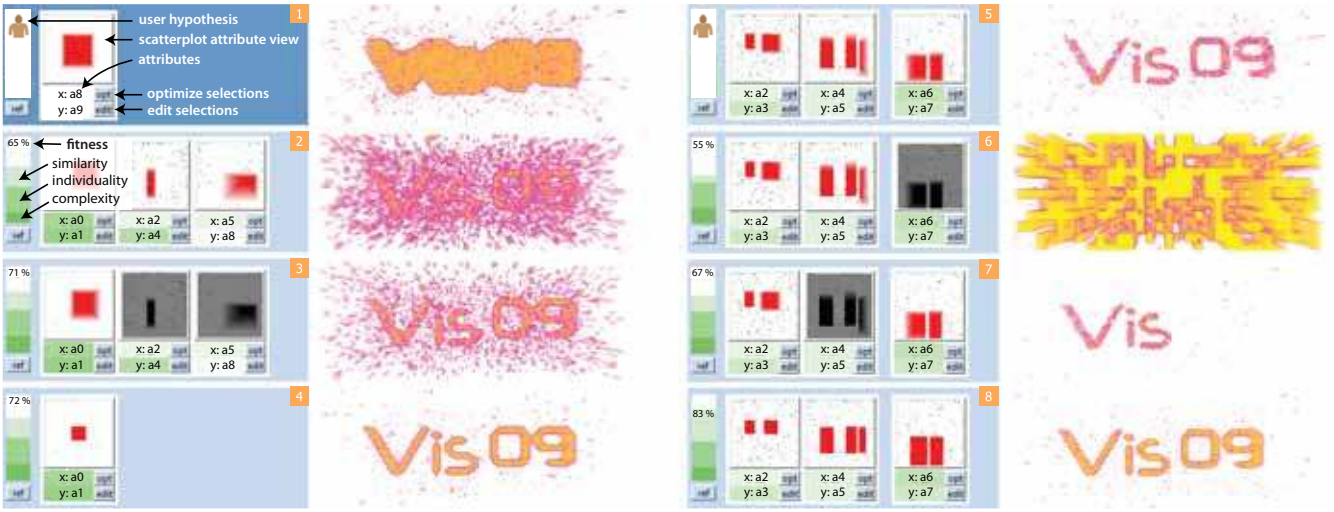

Fig. 3. Interaction example. (1) The user has the idea that attributes a_8 and a_9 describe a feature of interest. The resulting feature is shown in a rendering to the right. (2) The user runs the machine learning algorithm to search for hypotheses. The best resulting hypothesis has 65% fitness. The coloring (dark green) shows that attributes a_0 and a_1 are important in the entire population. (3) The user deactivates genes on other attributes and the fitness increases. (4) Via interactive visual analysis, the user improves the hypothesis. It is still unclear whether there are alternative hypotheses. (5) A larger search has found an alternative hypothesis. (6) The alternative explanation consists of two clauses combined by conjunction (AND). If one of the two clauses is missing, the feature vanishes. (7) The first clause contains the disjunction (OR) of multiple selections. Attributes a_2, a_3 select the shape "Vis" only. (8) Automatic local optimization via hill climbing finds the optimal selections.

bined by conjunction ($\wedge$). If one of the two clauses is missing, the feature vanishes as already discussed above. Figure 3(7): Based on the interactive framework, the user analyzes the impact of the different parts of the hypothesis and is able to understand the relation of the attributes to the 'Vis' and '09' parts of the feature. Figure 3(8): Finally, it is desirable to reduce the amount of noise in the selection. However, this is difficult do by hand. A simple optimization scheme helps the user to find the optimal selections in relation to the feature of interest.

4 Hypotheses

In this section we describe the formalization of hypotheses and measures to quantify their properties.

4.1 Definitions

A sample point in the n-dimensional data set is given by the attribute vector $\mathbf{a} = (a_1, \ldots, a_n)$. Each selection $S(\mathbf{a})$ is a mapping from $\mathbb{R}^n$ to fuzzy membership values in the range $[0,1]$. A data element a is assigned a value of $S(\mathbf{a}) = 1$ if it is fully selected. A view $V(\mathbf{a})$ is a mapping that presents a subspace of the attributes to the user and can have specific interactions to select data points interactively. The suggested method could potentially combine multiple data selection metaphors, but we focus on smooth brushing on two-dimensional scatter plots. Without loss of generality, we will restrict these to rectangular areas to simplify the discussion. A single smooth rectangular selection $S(\mathbf{a})$ can be defined by two rectangular areas $R_1 \subset R_2$ on the drawing area of the view (see transparent regions in layers of Figure 4). Each data element is then assigned a fuzzy membership value of 1 if it was mapped onto a position $p \in R_1$. A data element has fuzzy membership value of zero if it is mapped outside of R_2 and is assigned fractional membership value inside the region $R_2 \setminus R_1$. We define this fractional membership value as $d(p,\partial R_2)/(d(p,\partial R_1) + d(p,\partial R_2))$ with $d(x,R) := \min\{|x-r|, r \in R\}$ where ∂ is the boundary operator. This definition has the benefit of being directly extensible to non-rectangular selections. Users may specify multiple selections within a single view which are then combined as fuzzy disjunctions. For more complex hypotheses, different views can be connected to each other via fuzzy logic operators. The resulting terms are Boolean expressions which are transferred into conjunctive normal form. This modification is necessary, as we bias towards simple, efficient and interpretable terms for both the genetic algorithm and interactive visualizations. With this transformation at hand, we formally define the hypothesis $\hat{H}(\mathbf{a})$ as a conjunction of N clauses $C_i(\mathbf{a})$

$$\hat{H}(\mathbf{a}) := \bigwedge_{i=1}^{N} \underbrace{\bigvee_{j}^{M_i} S_{ij}(\mathbf{a})}_{clause\, C_i} \tag{1}$$

where each clause is a disjunction of M_i selections S_{ij}. The straightforward way to evaluate Equation 1 is to translate the operators to the point-wise operations 'minimum' and 'maximum'. This leads to the expression

$$H(\mathbf{a}) = \min_{i=0}^{N}(\max_{j=0}^{M_i} S_{ij}(\mathbf{a})). \tag{2}$$

Within a clause, a data point is selected according to the largest value in all contained selections. In the final hypothesis, each data point receives the minimum selection value of all clauses. Figure 4 illustrates the quantities defined so far. A clause can be regarded as a layer with transparent regions that correspond to the disjunctively combined selections. To visualize the selective effect of the clause, we imagine the data items as spheres that fall through such a layer. In analogy to a filter process, each sphere is then re-colored depending on the location it fell through. The color intensity corresponds to the fuzzy selection value. In this picture, a complete hypothesis is modeled as a stack of several overlapping layers. The search for an alternative hypothesis is thus equivalent to the search for a different stack of selective data filters. To qualify the resulting hypothesis we require new quantities defined in the next sections.

4.2 Feature Similarity

The feature similarity $\mathscr{F}(H_1, H_2)$ between two hypotheses H_1 and H_2 measures the point-wise distance between the feature membership values resulting from Equation 2. We distinguish between selected data items (fuzzy value > 0) denoted by $H_i^+ := \{\mathbf{a}|H(a) > 0\}$ and deselected items (fuzzy value$= 0$) that are given by $H_i^- := \{\mathbf{a}|H(a) = 0\}$ with $i = 1,2$, and compare these two sets independently. This is necessary since the number of points belonging to the feature (H_i^+) can be a lot smaller than the number of data points outside the feature (H_i^-).

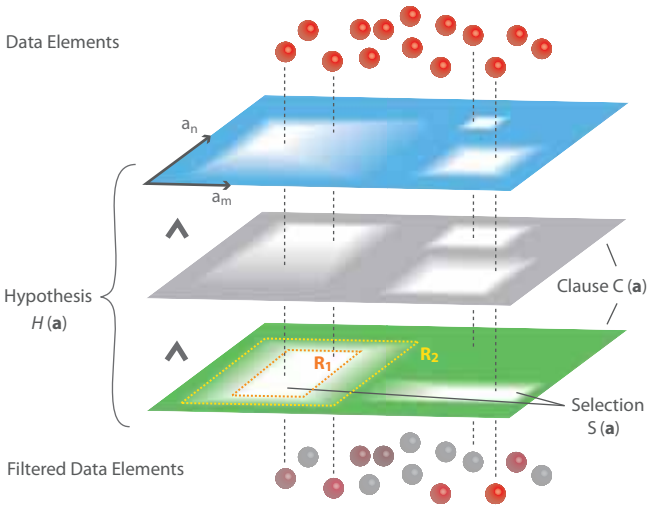

Fig. 4. Fuzzy logic hypotheses. The hypothesis is visualized as a stack of filter layers (clauses) with partially-transparent regions (fuzzy selections). The overlapping layers have a selective filter effect on the data items (spheres) above. While fully selected in the beginning (red), some of the spheres are being re-colored when falling through the stack. Less selected items receive higher desaturation, becoming grey if deselected.

Using this approach we can compute the directed similarity $\mathscr{F}_d$ between H_1 and H_2. The first term of $\mathscr{F}_d$ in Equation 3 accumulates the distance of all selected elements, whereas the second term accounts for the non-selected parts.

$$\mathscr{F}_d(H_i,H_j) = \frac{\sum_{H_i^+} 1-|H_i-H_j|}{|H_i^+|} \cdot \frac{\sum_{H_i^-} 1-|H_i-H_j|}{|H_i^-|}$$

$$\mathscr{F}(H_1,H_2) = \min(\mathscr{F}_d(H_1,H_2),\mathscr{F}_d(H_2,H_1)) \tag{3}$$

4.3 Complexity

The complexity of a hypothesis is a measures for the number of involved views v and the number of selections s in the hypothesis. More complex hypotheses are conceptually harder to understand. We thus need an expression, that allows for choosing upper limits for the number of views and selections, beyond which the complexity of a hypothesis strongly increases. Therefore, the complexity $\mathscr{C}(H)$ has the negative Gaussian shape

$$\mathscr{C}(H) := 1 - e^{-\left(\frac{s}{v\cdot\sigma_1}\right)^2 - \left(\frac{v}{\sigma_2}\right)^2} \tag{4}$$

where σ_1 and σ_2 control the number of selections (s) and views (v), respectively. We set $\sigma_1 = 10$ and $\sigma_2 = 5$ for all evaluations in this paper.

4.4 Individuality

The individuality measures how unique a hypothesis is in the set of all hypotheses in the current population. For this purpose, we require a function that compares two hypotheses according to their resemblance

$$\mathscr{R}(H_1,H_2) := \frac{1}{\max(\#H_1,\#H_2)} \cdot \sum_{i=0}^{\#H_1} \max_{j=0}^{\#H_2}(\mathscr{R}_c(C_i,C_j)) \tag{5}$$

where $\#H_i$, $i = 1,2$ refers to the number of clauses in a hypothesis. This definition is motivated by the 'stack of filters' understanding of a hypothesis as shown in Figure 4. From this viewpoint the resemblance of two hypotheses can be understood as the average resemblance of filters, where we always compare the two most resembling filters (clauses). The benefit of this approach is that two hypotheses

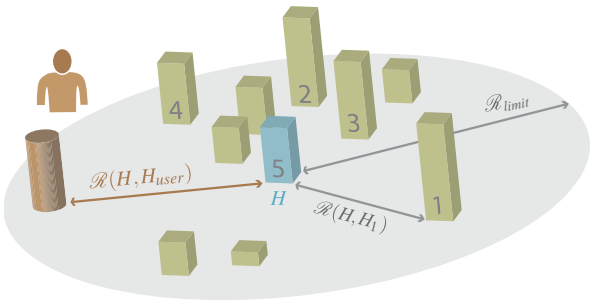

Fig. 5. Local ranking. For evaluating the rank of a hypothesis (grey number on the podiums), the current hypothesis (blue) is competing with its nearest neighbors in resemblance space. The higher its feature similarity (higher podium), the higher it's rank. The individuality is then computed as the resemblance to the user's hypothesis (brown), weighted with respect to the evaluated rank.

can have high resemblance if only a subset of their genes (selections) is similar. The resemblance $\mathscr{R}_c(C_1,C_2)$ between two clauses is

$$\mathscr{R}_c(C_1,C_2) := \frac{1}{\max(\#C_1,\#C_2)} \cdot \sum_{i=0}^{\#C_1} \max_{j=0}^{\#C_2}(\mathscr{R}_s(S_i,S_j)) \tag{6}$$

where $\#C_i$, $i = 1,2$, refers to the number of selections in a clause. A selection S as defined in Subsection 4.1 can be considered as a 2D fuzzy set with cardinality $|S| := \int_{\mathbb{R}\times\mathbb{R}} S(x)dx$. The resemblance of two selections $\mathscr{R}_s(S_1,S_2)$ is now defined as the cardinality of fuzzy set intersection of S_1 and S_2 divided by the cardinality of their fuzzy set union.

$$\mathscr{R}_s(S_1,S_2) := \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{7}$$

This can be understood as dividing the area of the intersection of S_1 and S_2 by the area of their union.

In order to allow the heuristic algorithm to create populations of sufficiently diverse hypotheses the following individuality ranking is of crucial importance. When there are multiple closely resembling hypotheses we want to rank them in a way that the one with the locally highest feature similarity also has highest individuality. Using Equations 5-7 we first compute the local neighborhood of a given hypothesis H which is defined as the list of hypotheses that are resembling H most. A hypothesis H' is contained in the local neighborhood of H if it fulfills the threshold condition $\mathscr{R}(H,H') \leq \mathscr{R}_{limit}$. This threshold is set to 0.6 in our evaluations. To calculate the individuality, all hypotheses in the local neighborhood are ranked by their feature similarity $\mathscr{F}$ with respect to the user's hypothesis. If the feature similarity of two hypotheses is equal, the less complex hypothesis receives higher rank. Figure 5 illustrates the local individuality ranking for a given hypothesis (blue podium). The resulting rank r is used to define the individuality $\mathscr{I}(H)$ as follows

$$\mathscr{I}(H) := (1-\mathscr{R}(H,H_{user})) \cdot (1-\varepsilon)^r \tag{8}$$

where the resemblance to the user hypothesis H_{user} is given special attention. The parameter ε controls the influence of the ranking value and is set to 0.25 in all our tests.

5 Machine Learning Algorithm

This section discusses our adaptation of a genetic search algorithm to find alternative hypotheses. A genetic algorithm consists of three basic steps: *selection*, *mutation* and *recombination*. In the selection process, the fittest individuals are chosen to be included in the next population without modification. A mutation step changes an individual randomly, thus widening the diversity of the population. Finally, in the reproduction step, individuals of a generation are recombined to produce members of the next generation.

The population size in our experiments contains 100 hypotheses. Ten are transferred via selection, 45 via mutation and 45 via reproduction. Setting these values can be important when the user wants different search strategies at different points in the analysis process. In the beginning, a configuration of high mutation rate, low selection and large population size might be good. Later, when several good candidate hypotheses are already present, low mutation rates and a very high recombination level can be appropriate to find interesting larger hypotheses consisting of the genes already present in the population.

5.1 Fitness and Selection

The fitness of an individual is defined in relation to the user hypothesis H_{user} and all other hypotheses of the population. We define the fitness of an individual H as

$$Fitness(H) = w_f \cdot \mathscr{F}(H, H_{user}) + w_i \cdot \mathscr{I}(H) - w_c \cdot \mathscr{C}(H). \quad (9)$$

where w_f, w_i and w_c weight the influence of feature similarity, individuality and complexity, respectively. For the evaluations in the paper, we set $w_f = 0.5$, $w_i = 0.3$ and $w_c = 0.2$. If the user is not completely sure about the shape and position of a given feature, there are two ways to take this into account: First, by a lower weight of the feature similarity component. Second, by reducing the fuzzy selection values in the reference hypothesis. We have found that overly complex hypotheses can require a large portion of the computational resources. We thus remove hypotheses of very high complexity scores without a complete fitness evaluation.

When an individual of the population has to be selected, we employ an acceptance-rejection method. This way, we achieve a distribution of random selections that is in accordance to the fitness values. A random integer i in the range from 0 to the size of the population and a random real value from the $f = [0,1]$ interval are generated. If the i-th individual in the population has fitness higher than f, this individual is selected. Otherwise the process is repeated.

5.2 Mutation

In standard definitions of genetic algorithms, mutations change bits of the string which defines the individual. We use a more semantic mutation operator to change the elements of a hypothesis. In the mutation step one individual is chosen as discussed in the previous section. A selection can mutate in four different ways: edit, move, crawl and replace. A selection is defined by a number of vertices (8 in the case of rectangles). The edit-mutation either selects one of the float values which describe the position of one vertex or it changes the data attribute a_i randomly. The move-mutation changes the position of all vertices of a selection in the same random direction. The crawl-mutation edits the fuzzy boundaries with higher probability than the sharp boundaries. The replace-mutation generates a novel random selection. We use equal probabilities for all four mutation methods.

The Boolean operators in a hypothesis mutate by reduction and flip. The reduction-mutation randomly removes one of the arguments of the selected operator. For example $\vee(s_1, s_2, s_3)$ could be reduced to $\vee(s_2, s_3)$. The flip operator changes a disjunction to a conjunction and vice versa. We do not use an insertion operation.

5.3 Recombination

In the reproduction step, two individuals are selected as discussed in Section 5.1 in order to be combined for generating new offspring. We suggest to represent each hypothesis as an operator tree (see also Section 7). For both selected individuals, a node in the operator tree is chosen and two offspring individuals are created by swapping the subtrees attached to the selected nodes. Care has to be taken that also the subtrees of these operators are split randomly such that the number of possible recombinations of the two individuals is not artificially limited. This approach benefits from a continuous evolution of hypotheses, which is in the spirit of interactive control over the evolutionary process.

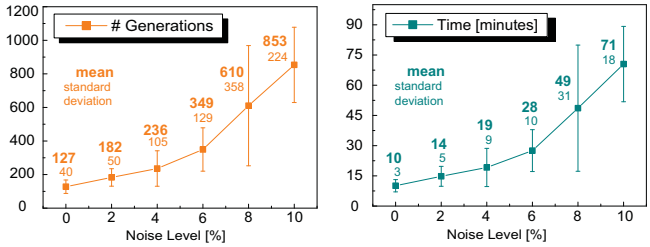

Fig. 6. Performance analysis: The search algorithm has been tested on synthetic data at different noise levels. The left graph shows the number of generations needed until a feature similarity of 0.8 has been reached. The right plot shows the related timing results.

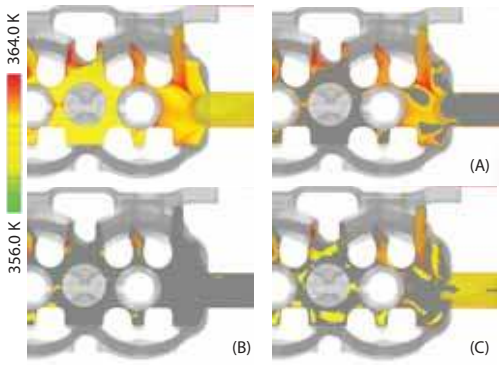

Fig. 8. Feature comparison: (top) Temperature distribution in a plane. (A-C) Selected areas correspond to the findings in Figure 7. We can see that the different variables explain different but partially overlapping subsets of the fluid.

6 Evaluation

In this section we describe evaluations of the visual human+machine learning approach.

6.1 Measurements

This section analyzes the number of generations and the time necessary to find a sharply defined feature inside synthetic volume data. We have set up a spherical feature in attributes a_0 and a_1 of a 10-dimensional data set that consists of 32^3 voxels. More exactly, for voxels with $v_x^2 + v_y^2 + v_z^2 < 1$ the attributes a_0, a_1 contain random values in the $[0, 0.1]$ interval, all other voxels contain random values in the $[0,1]$ interval, but either a_0 or a_1 has to be larger than 0.1. The alternative hypothesis to be found by the algorithm has been defined in two other attributes a_2 and a_3 in the same way. This (as well as the 'Vis09' example) can be considered as a worst case scenario since a slight deviation from the correct selection in a_2, a_3 performs poorly, whereas in practice, features in CFD data tend to be smoothly defined such that approximately correct hypotheses already perform well.

To study the robustness of the search algorithm, we have applied different noise levels to the attribute values. For noise level w we generate a random value r between -1 and 1 and modify the attribute value $a \leftarrow a + 0.01 \cdot w \cdot r$. Figure 6 shows the number of generations and time needed until a feature similarity of 0.8 has been reached. The algorithm has been rerun 100 times per setting to evaluate the error bars. While performing well at low noise levels and producing results after several minutes, the algorithm becomes less reliable the higher the noise level. This behavior can be expected, as the feature becomes less defined the more voxels contain random attribute values.

6.2 Case Study: Analysis of a Cooling Jacket

In the following subsection, we discuss an application of the suggested technique on a real world data set. A detailed description of the cool-

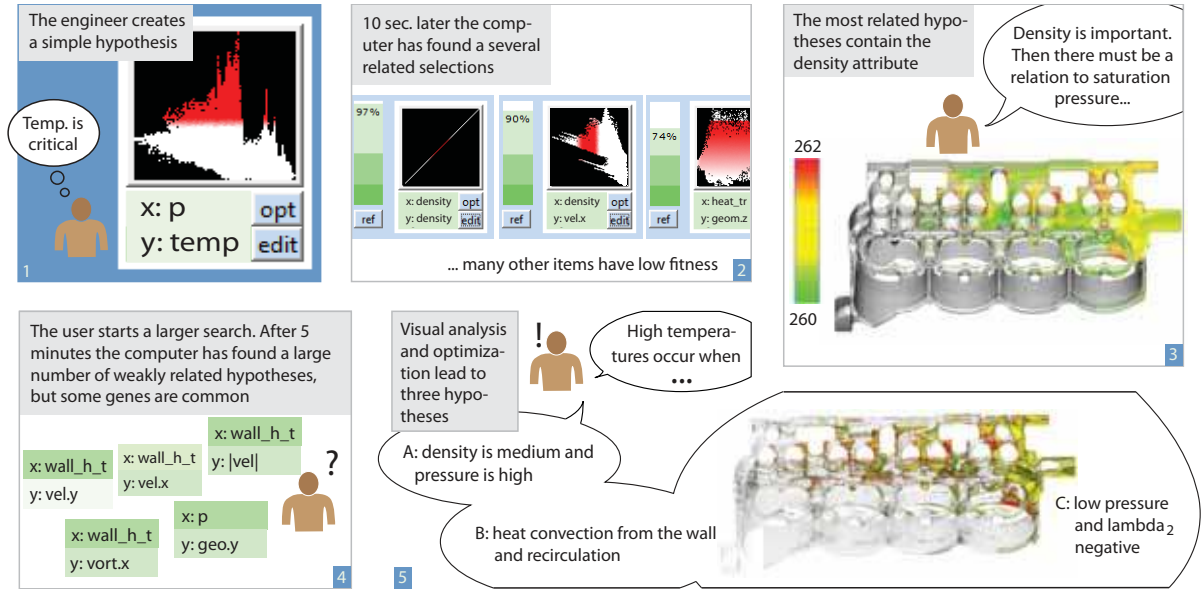

Fig. 7. Case study story: The engineer knows that there are critical temperatures inside the data set. An initial search shows that density and temperature are highly related since vaporized coolant has low density. The user adjusts his hypothesis to *hot AND vapor* and starts a larger search, but there is no single best solution. He analyzes the attribute ranges with the highest fitness and finds multiple hypotheses which can explain the hot temperatures at different locations in the data set. A: where pressure is high, the vapor could be compressed too much. B: where recirculation areas appear, the fluid transport could be hindered. C: vortex regions could trap hot fluid inside.

ing jacket can be found in previous work [17, 9, 3]. It is sufficient to know that the most important attribute of this data set is the fluid temperature. Many simulation variables (19 attributes) are related to the fluid temperature. Also, there are additional important derived attributes such as the λ_2 vortex detector or velocity, pressure and density gradients. In sum, there are 35 possible attributes to consider inside a data set consisting of approximately 1.5 million 3D cells.

We start the analysis with the assumption that high temperatures ranging from 363.0K to 372.151K with a fuzzy border of 1K correspond to a feature (see Figure 7 and 8). A quick initial search with 20 generations and a population of 50 individuals reveals a connection between high temperature values and a certain range of fluid densities. We use hill climbing to find the optimal density interval d and reach an almost exact (97%) feature similarity to the original hypothesis as seen in Figure 7 (2). Based on this information, the engineer might guess that the high temperatures are related to the parts of the fluid where high pressure hinders the hot parts of the fluid to expand. Therefore we extend our hypothesis to "high temperature and density in d", i.e. high saturation pressure (Figure 7 (3)). After a few minutes of interactive analysis it is clear that it is not possible to find related attribute ranges interactively.

We continue with a larger search and after 300 generations the machine has generated a large number of suggestions, but none of them has very high feature similarity. The visualization of global attribute frequencies shows that selections on certain attributes are very common in the population though, so we can still assume that there are hidden features to be found. We focus on the attributes related to the successful genes and deactivate all other genes from the best selections. Now, a visual comparison of the volume rendered features related to the three best hypotheses explains the non-perfect feature similarity: the generated hypotheses are related to three different but partially overlapping subsets of the fluid. See also Figure 8. Hypothesis A "density is medium and pressure is high" is actually the explanation the engineer was looking for, but it was impossible to verify this hypothesis as the only one. There are two other possible explanations B, C for different hot parts in the fluid. There are also sections where a lot of heat is transferred from the wall, but the material transport is jammed due to recirculating flow. Also, there are parts where vortices ("low pressure and $\lambda_2 < 0$") are trapping hot parts of the fluid. This combination of hypotheses is a novel result, which none of the previous case studies was able to generate since the interplay of six attributes in the data set (temperature, density, pressure, wall heat convection, inverse flow and λ_2) is almost impossible to find using interactive analysis alone.

7 Implementation

The presented approach becomes possible since the compute intensive parts of the fitness evaluation are implemented on the GPU. The operations which require the traversal of all data elements, i.e. the evaluation of hypotheses (Equation 2) and feature similarity (Equation 3) can be implemented in Nvidia's Cuda [21]. Hypothesis evaluation is a straightforward data traversal to evaluate a function consisting of min and max operations. For the required data handling the 'histogram' example available with Nvidia's Cuda SDK [21] is a good starting point. Using the Cuda API it is possible to stream chunks of data to the GPU such that the size of GPU memory is not a limiting factor.

The computation of feature similarity is a simple modification of the available 'reduction' example available in the Cuda SDK, where the computation of the sum is replaced by the term $\mathscr{F}_d$ from Equation 3. The computation of these values and the corresponding data handling require about 98% of the computation time. The genetic algorithm itself requires less than 1% of computation time and is implemented using the C++ standard template library. An operator tree based representation of hypotheses simplifies the otherwise complicated operations required by the evolutionary algorithm. Especially mutation, recombination and the translation of hypotheses to conjunctive normal form is simplified by this approach.

8 Conclusions, Problems and Future Work

The main goal of this paper is to answer the questions related to the verification and generation of hypotheses. The outcome is a framework for generating hypotheses by combining human and machine learning. Instead of replacing the inferential abilities of the human, the machine stimulates them by producing alternatives which can be evaluated by the human. A heuristic search algorithm which generates related features helps the engineer to evaluate his theories and find alternatives in the vast hypothesis space of multi-variate data. The

generated hypotheses are automatically added to the visualization and are available for further refinement and search steps.

In the presented framework, a hypothesis is simply a set of fuzzy selections together with boolean combinations. This simple format is easily transferred to text and facilitates reporting (e.g. "temperatures larger than 360 K coincide with densities in the range 0.3 to 0.5 kg/m^3 and negative velocities."). This is easier to understand than the weights of a neuronal net for example.

We consider the understandability of the generated hypotheses and the suggested incremental human+machine learning approach as the key selling point of the presented work. As Shneiderman has suggested, the approach presented in this paper actually allows the user to specify what he is looking for. Foremost, the visual human+machine learning approach respects the human responsibility. The integration of the machine learning step can actually prevent the user from missing important relations in the data.

There are several questions remaining. Since it is not possible to search the vast space of all possible hypotheses we cannot guarantee that we are not missing an interesting hypothesis. A statistical approach to measure the decreasing probability that important selections of a given complexity have been overlooked would be an important improvement, not only to the presented approach but also to IVA in general. Secondly, no user study has been performed. The results discussed in this paper have been generated with a minimum of interaction. We claim that the machine assistance reduces the required information analysis skills since a lot of work is done by the machine. We plan to evaluate this the future.

There are several improvements we propose for future work. First, we have discussed the evaluation of spatial features in three dimensions, but it would be possible to apply the presented technique in a more general setting, i.e., when features are not structured in space. Non-spatial features could benefit from the presented approach using information visualization techniques only. Second, the presented approach is not limited to one particular gene type. Other visualization parameters (e.g. a viewpoint or a transfer function) and user inputs could be included into the learning process. Third, in real world data, there can be a lot of samples which do not change the fitness of a given hypothesis. To improve performance it could be sufficient to evaluate the fitness only for a randomly selected subset of the data. Fourth, the selections presented in this paper are all based on brushing scatterplots. The presented approach is extensible to more complex selections which can be defined differently in various types of views and modified by appropriate mutation and recombination operators. Fifth, another interesting set of open questions is related to specialized visualizations of population development. The analysis could benefit from visualizations that show the ancestry relations of hypotheses and the structural similarities of the fittest individuals. This could provide information about the specific location of selections inside the operator tree of a hypothesis. Finally, the presented framework is limited to orthogonal projections. It would be interesting to study if the combination of the grand tour [2] with visual human+machine learning reveals additional meaningful hypotheses.

ACKNOWLEDGEMENTS

The authors thank Prof. Eyke Hüllermeier for his input. The project SemSeg acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042 and SimCT project (FFG Bridge, grant number 812136-SCK/KUG).

REFERENCES

[1] M. Ankerst. *Visual Data Mining*. PhD thesis, Faculty of Mathematics and Computer Science, University of Munich, 2000.

[2] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, 6(1):128–143, 1985.

[3] R. Bürger, P. Muigg, M. Ilcìk, H. Doleisch, and H. Hauser. Integrating local feature detectors in the interactive visual analysis of flow simulation data. In *Proc. EuroVis 2007*, pages 171–178, 2007.

[4] D. Caragea, D. Cook, H. Wickhamand, and V. Honavar. Visual methods for examining SVM classifiers. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, pages 136–153, 2008.

[5] C. Chen. Top 10 unsolved information visualization problems. *IEEE Comput. Graph. Appl.*, 25(4):12–16, 2005.

[6] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. VisSym 2003*, pages 239 – 248, 2003.

[7] T. M. English. Optimization is easy and learning is hard in the typical function. In *Proc. Congress on Evolutionary Computation: CEC00*, pages 924–931, 2000.

[8] M. Hao, U. Dayal, D. Keim, D. Morent, and J. Schneidewind. Intelligent visual analytics queries. In *Proc. VAST 2007.*, pages 91–98, 2007.

[9] H. Hauser, R. Laramee, and H. Doleisch. Topology-based versus feature-based flow analysis - challenges and an application. In *Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization*, 2007.

[10] A. Hertzmann. Machine learning for computer graphics: A manifesto and tutorial. In *Proc. Pacific Graphics 2003*, page 22. IEEE Computer Society, 2003.

[11] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):168–175, 2007.

[12] M. Jeng. A selected history of expectation bias in physics. *American Journal of Physics*, 74:578–583, 2006.

[13] J. Kehrer, F. Ladstadter, P. Muigg, H. Doleisch, A. Steiner, and H. Hauser. Hypothesis generation in climate research with interactive visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1579–1586, Nov.-Dec. 2008.

[14] D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. *Lecture Notes In Computer Science*, pages 76–90, 2008.

[15] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[16] S. Laine. *Using visualization, variable selection and feature extraction to learn from industrial data*. PhD thesis, TU Helsinki, 2003.

[17] R. Laramee and H. Hauser. Geometric flow visualization techniques for cfd simulation data. In *Proceedings of the 21st spring conference on Computer graphics*, 2005.

[18] K.-L. Ma. Machine learning to boost the next generation of visualization technology. *IEEE Comput. Graph. Appl.*, 27(5):6–9, 2007.

[19] T. May and J. Kohlhammer. Towards closing the analysis gap: Visual generation of decision supporting schemes from raw data. *Computer Graphics Forum*, 27(3):911–918, 2008.

[20] W. Müller, T. Nocke, and H. Schumann. Enhancing the visualization process with principal component analysis to support the exploration of trends. In *Asia Pacific Symposium on Information Visualisation*, volume 60, pages 121–130, 2006.

[21] Nvidia's CUDA. `www.nvidia.com/object/cuda_home.html`.

[22] F. Poulet. SVM and graphical algorithms: a cooperative approach. In *ICDM '04. Fourth IEEE International Conference on Data Mining, 2004.*, pages 499–502, 2004.

[23] F. Poulet. *Visual Data Mining*, chapter Towards Effective Visual Data Mining with Cooperative Approaches, pages 389–406. Springer, 2008.

[24] P. Rautek. *Semantic Visualization Mapping for Volume Illustration*. PhD thesis, Vienna University of Technology, 2009.

[25] F. Rossi. Visual data mining and machine learning. In *European Symposium on Artificial Neural Networks (ESANN2006)*, 2006.

[26] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.

[27] B. Shneiderman. Inventing discovery tools: combining information visualization with data mining. *Information Visualization*, 1(1):5–12, 2002.

[28] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.

[29] F.-Y. Tzeng and K.-L. Ma. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Proc. of the 2005 ACM/IEEE conference on Supercomputing*, page 6, 2005.

[30] M. Ward. Finding needles in large-scale multivariate data haystacks. *IEEE Comput. Graph. Appl.*, 24(5):16–19, 2004.

[31] D. Yang, E. A. Rundensteiner, and M. O. Ward. Analysis guided visual exploration of multivariate data. *In Visual Analytics Science and Technology, VAST 2007.*, pages 83–90, 2007.