Advancing Formative Assessment in MCAD Education: The Visual Analytics of Parametric Feature-Based Solid Models

Original

Advancing Formative Assessment in MCAD Education: The Visual Analytics of Parametric Feature-Based Solid Models / Otto, H. E.; Mandorli, F.. - In: ADVANCED ENGINEERING INFORMATICS. - ISSN 1474-0346. - ELETTRONICO. - 48:(2021). [10.1016/j.aei.2021.101308]

Published
DOI:10.1016/j.aei.2021.101308

Advancing formative assessment in MCAD education: The visual analytics of parametric feature-based solid models

Harald E. Otto[1] · **Ferruccio Mandorli**[1]

Abstract

Advancing formative assessment in MCAD education is an important but difficult problem. Success in this endeavor requires feature-based MCAD model assessment to consider the quality of a model from various viewpoints. That includes the absolute criteria that are associated with technical domain knowledge and also criteria related to model deficiencies. For the latter, this entails assessing the results of wrong or inappropriately applied system commands, and of partial or entire modeling strategies. Here, an approach that combines the perceptual abilities, creativity, and domain knowledge of the human user with the computational power of current desktop computing has great potential to contribute to solving the problem. The aim of the current paper is two-fold. Firstly, it presents a novel approach to analyzing feature-based characteristics of MCAD models, an approach that is aimed at advancing formative assessment in the educational context. This approach is based on visual analytics and efforts to combine visualization, human factors, and data analytics. Secondly, it reports on the technical architecture and concrete implementation of a newly developed visualization environment for a software tool to enable and put into practice this novel MCAD model assessment approach. The development of this new visualization environment is based on an advanced visualization pipeline that employs radial visualization, while supporting dedicated user interaction techniques to facilitate analytical processes.

Keywords Strategic knowledge built-up · Formative feedback · Advanced visualization pipeline · Visual analysis of multivariate data · Radial visualization

1. Introduction

The increasing gap between student learning outcomes that are achieved with classic, though apparently outdated, teaching approaches in departments of science and engineering at institutions of higher education, and the vigorously rising demand for professionals with sophisticated skills and competencies in highly competitive markets led globally to the mobilization of various efforts to introduce changes in the way course curricula and teaching are designed and executed. In the context of computer-aided design (CAD) and, in particular, CAD education for mechanical engineering (MCAD), this translates, according to trends and studies, into a focus on the development and implementation of restructured curricula and alternative teaching approaches. These are more student centered and learning oriented, and thus are better structured to efficiently and effectively match actual student learning outcomes with skills and competencies related to, among other skills, spatial ability and mental visualization, cognitive model composition, meta-cognitive processes including planning, predicting, and revision, and modeling strategies (see also [11,28,98]).

✉ Ferruccio Mandorli
f.mandorli@staff.univpm.it

[1] Department of Industrial Engineering and Mathematical Sciences, Faculty of Engineering, Polytechnic University of Marche, I-60131 Ancona, Italy

Recent efforts to reform an actual MCAD course, which is currently a part of the curriculum for the Laurea degree in mechanical engineering at the institution represented by the authors, addressed, among other matters, the development of modeling competencies with particular reference to the strategic knowledge required to create well-designed usable MCAD models (cf. [68,69]). This major course-specific learning goal, i.e. development of MCAD competency, and in particular the strategic knowledge and modeling skills indispensable for producing well-designed MCAD models, requires better teaching techniques that reach beyond the usual lecture-based presentation of domain-specific factual knowledge with students mostly in the role of passive learners. Moreover, it especially requires assessment techniques and feedback which are capable of adequately and frequently measuring the gap between actual student learning outcomes as achieved and learning goals as pre-assigned, while also providing high quality and timely feedback for both teacher and students. Within this setting, and in the context of higher education, as outlined earlier, the assessment of student performance and results produced in CAD laboratory exercises and course assignments needs to be conducted in a computer-aided manner. This will support actual implementation, while also improving the scope and overall quality of formative assessment and feedback, but it requires new approaches and tools for feature-based solid model assessment.

The aim of the current paper is two-fold. Firstly, it presents a novel approach to analyzing feature-based characteristics of MCAD models, an approach that is aimed at advancing formative assessment in the educational context. This approach is based on visual analytics and endeavors to combine visualization, human factors, and data analytics, which will allow the analysis to benefit from the strengths of both human intelligence and computer-based data processing. This will support, as well as further, efforts to make data and information processing more transparent for the analytical discourse, and also to better synthesize information, derive insight and understanding, detect and validate what is expected, and discover the unexpected. Within the educational context outlined, the last is of particular importance because it can contribute to increasing the teacher's knowledge of what can go wrong during CAD laboratory exercises, and which kinds of deficiencies and errors students are actually capable of introducing into MCAD models. Consequently, the teacher will gain a better understanding of the difficulties students face while actually practicing MCAD modeling.

Secondly, this paper reports on the technical architecture and concrete implementation of a newly developed visualization environment module for a software tool to enable and put into practice this novel MCAD model assessment approach. The development of this new visualization environment is based on an advanced visualization pipeline that employs radial visualization utilizing optimized Kiviat diagrams, while supporting dedicated user interaction techniques to facilitate analytical processes. Regarding these processes, extra effort has been devoted to the design and implementation of effective and efficient data transformation modification and viewing support. In particular, a brushing mechanism has been incorporated based on cross-linking the visualization environment with the feature-based MCAD modeling environment. User interactions for the human-information discourse are explicitly supported by mechanisms that allow for data and information to be compared, categorized, and annotated interactively.

In this paper, first an overview is provided of current developments in CAD education and modeling with feature-based MCAD systems, along with some background on MCAD model assessment and visual analytics. Next, the problem space of feature-based MCAD model assessment within the educational context is analyzed to support the forming of concrete application domain related requirements for visual data and information representation, and user interactions to enable and facilitate visual analytics. Based on those requirements, a novel approach is introduced, employing a radial visualization of so-called *feature-based characteristics* and visual analytics to enhance, as well as make more efficient, the assessment of feature-based MCAD models within the educational context. Details of the framework, the concepts, and the system architecture of a visualization environment module prototype that was developed and implemented are then presented and discussed. This is followed by examples and discussions of results obtained and experiences gained from within the educational context of an actual MCAD course. Lastly, a brief summary of outcomes achieved so far and an overview of work currently in progress are provided, and some conclusions are drawn in the final section.

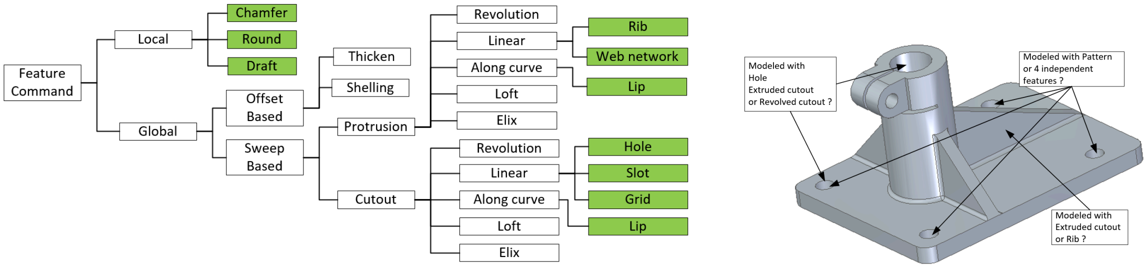
2. Background, scope, and motivation

2.1. Developments in CAD education

Issues and current shortcomings in CAD education, some of which have been previously outlined and are further discussed elsewhere in this paper, have been addressed within discipline-based educational research from several directions as follows. A recent approach to transforming and advancing adaptive expertise development in CAD education by integrating contextual exercises was presented in [56]. A study on the transfer of learning between 3D modeling systems is reported in [101], and provides results with implications for the design of educational programs in regard to delineating between system dependent and system independent skills and knowledge. Educational issues similar in this direction toward knowledge and expertise development that can transcend a particular CAD system are discussed, for example, in [104,108]. Efforts to address improvements in pedagogical approaches for 3D CAD are presented in [18], and these address CAD expertise development by supporting strategic knowledge development and the improvement of spatial ability. In particular, to support the latter, integration of a number of strategies has been proposed. These include pre-exposure to perceptual differentiation, experience with manipulative tasks, and the use of sketching. Work on a theoretical framework and heuristics for best practice is introduced in [79], addressing issues of developing both the capacity to generate cognitive models and the ability to decompose geometric elements, leading to better cognitive handling of modeling concepts as well as achieving design intent in the context of parametric modeling system oriented CAD pedagogy. Work addressing and reviewing educational issues of teaching aids based on augmented reality, cognitive model composition and meta-cognitive processes is reported, for example, in [22,32,47,61,71,76,103,105]. Topics covered include strategic knowledge and design intent, as well as recent empirical research. To translate the potential and benefit of those encouraging approaches into educational practice, however, also requires better structured and more frequent assessment and feedback than can be achieved with traditionally employed summative assessment and feedback techniques. Here, formative assessment (cf. [9,45,78]) and formative feedback [38,84] appear to offer a viable solution, and are increasingly regarded as promising and effective components within instructional practices currently proposed for reforming higher education in science and engineering. Unfortunately, within CAD education, dedicated techniques and tools are not yet available to support the implementation of formative assessment, in particular to assist learning goal and outcome oriented assessment of CAD models produced by students. Moreover, those frameworks and tools for CAD model analysis and evaluation that are available and deployed within commercial and industrial settings cannot be directly used in educational settings. This is due to differences in assessment criteria and evaluation goal settings, focusing mostly on issues related to application context, quality, and interoperability of CAD models (cf. discussions and commercial / industrial tool reviews in [3,33,107]).

2.2. Modeling with feature-based MCAD systems

Among the various definitions of features that can be found in the literature, we can define a feature as the inherent concept of a meaningful abstraction of elemental components of a mechanical part that are organized into categories such as ribs, holes, and slots that designers and engineers use when referring to products. Here, from an ontological viewpoint, a feature belongs to the domain of real objects (cf. [58]). Within the MCAD domain, those features of the domain of real objects are conceptually approached with what are called in this paper *full-features* and *shape-features*. Using an MCAD system, actual instances of full-features and shape-features are created by applying feature commands. Shape-features are comprised of general data that contain, for example, an identifier, a reference to a feature command history entry, etc., shape data containing all shape related parameters such as geometric dimensions and spatial location, and the feature shape that consists of all the topological and geometric entities used to form the shape that is the geometric representation. Full-features are comprised of the same data sets as shape-features, but additionally contain feature specific data relating to engineering and design properties of a particular type of feature. For example, full-features of type rib provide the feature specific rib thickness parameter, which is then, in an automatic manner, symmetrically applied in respect to the profile plane where the rib profile has been specified. Note that those feature specific data are also used to support consistent and effective MCAD model alteration, as is shown for concrete examples elsewhere in the paper. A structured overview of feature commands for the more specific full-features (colored in green) and the more general shape-features that can be found in most commercially available MCAD systems is shown in Fig. 1(a). To avoid ambiguity, in the remainder of this paper the domain or type / category is provided alongside the term feature when necessary. At this point, perhaps it should be made explicit that what has been said about features so far was aimed at providing a system neutral ontological as well as epistemological perspective in regard to the design / engineering meaning of features, their shape aspect, and related commands as provided by most modern MCAD systems. This endeavor, however, should not be mistaken as an attempt to create yet another feature definition and taxonomy.



(a) (b)

Fig. 1 Feature commands and feature shapes. From left to right: (a) structured overview of feature commands that are usually provided within modern MCAD systems, (b) individual feature shapes created with various feature commands.

With a feature-based MCAD system, the modeling process is based on the sequential application of feature commands. This is driven by the objective of adding full-features and shape-features to the MCAD model. First, the user selects the type of feature command that is considered most suitable for the modeling task. Next, the user interacts with the system interface to carry out the various steps of the modeling command. The sequence of executed feature commands is then stored in a kind of feature command history list. Feature commands can be logically subdivided into local and global commands (see again Fig. 1(a)). Local commands related to, for example, rounds and chamfers, are used to detail the local shape within an MCAD model. Global commands, such as cutouts and slots, are used to model the global shape. Global feature commands are executed by initially selecting a plane, then drawing a profile on the plane, and finally defining the extrusion constraints. Note that these global commands are actually the result of a sophisticated implementation of parameterized sweep and Boolean operations that were already present in CAD systems before the introduction of the feature-based modeling approach.

Due to the importance of the relationship between the feature shape and feature commands, some fundamental issues need to be made clear, as follows. Firstly, the shapes of the various features in the domain of real objects can be created within an MCAD system by employing a single shape-feature command. For example, the shape of a slot or a pocket can be created using just the command for an extruded cutout shape-feature. Secondly, there is no unambiguous mapping between the shape of a feature and the feature command applied to create its geometric representation within a CAD system. Hence, various feature commands can be applied during the modeling of the same shape of feature, as shown in Fig.1(b). Thirdly, as an unintended side effect, the feature shape created with one feature command can be altered or even deleted by the application of a following feature command. This usually results in the entry of a feature command in the history of the feature modeling sequence (general feature data) without a corresponding feature shape in the MCAD model. This situation renders the MCAD model in an inconsistent state in regard to the presence of a modeled feature, because some data, such as the entry of the executed feature command in the feature modeling sequence, are still present in the model, while other data, such as the feature shape, are missing.

2.3. MCAD model assessment and visual analytics

Within the context of CAD education for mechanical engineering, assessing MCAD models is quite a delicate and highly time-consuming activity, which requires, among many other competencies, the ability to discriminate efficiently between trivial errors, i.e. errors that have been committed by students due to carelessness and inadvertence while performing the exercise, and more serious errors, i.e. errors that have occurred due to a lack of knowledge and understanding of the domain subject. Moreover, due to the problematic correspondence between feature commands to create an MCAD model and the resulting shape of the MCAD model, where differences in the MCAD model creation do not necessarily lead to differences in the MCAD model shape, an alternative method of representing MCAD model properties that are different from the computer-rendered shape needs to be employed. In statistical analysis, a similar situation was described in [1], where differently structured data sets showed an identical outcome (statistical profile) from the viewpoint of basic descriptive statistics. However, they showed very different results when the method of interpreting the data was changed and an alternative graphical representation and visual analysis of the data sets was employed. Partly inspired by this example, those responsible for the approach presented in this paper determined that their first goal was to find an alternative graphical representation of MCAD model properties that could better support visual analysis and model assessment. Such an alternative graphical representation should be based not on topological and geometric characteristics, but on characteristics that are related to the means of creating feature-based MCAD models – that is the modeling command structure and its conceptual nature based on features. For this purpose, it was determined that understanding graphically represented MCAD model characteristics in a systematic and computer-aided manner would best be achieved by the adoption of visual analytics.

Visual analytics is a highly interdisciplinary area, building on several scientific fields: cognitive science, human perception research, information / scientific visualization, and data management. In particular, visual analytics combines the strength of human visual perception and cognitive capabilities with the data processing power of computer systems, thus enabling semi-automated analytical processes where man and machine cooperate to produce results based on effective and efficient use of their distinct and outstanding capabilities (see also [49,64,83,85,97]). Visual analytics is more than just visualization, as it combines data analysis with human factors and visualization technology, resulting in an integral approach to decision-making, with the goal of making the processing of data and information transparent for analytical discourse (cf. [48]). This is also emphasized by the fact that visual analytics gives priority to making sense of the data through the various iterations of data analytics (cf. [92]). Visual analytics is highly application oriented and thus strongly driven by requirements stemming from the individual application domain, as is shown in detail elsewhere in this paper. However, to enable visual analytics that are effective and efficient, the information visualization environment needs to support the user interactions required by visual analytics and provide an information display that is optimized for efficient human perception and is appropriately structured to allow for visual analysis of the patterns, data trends, etc., which are subject to exploration and assessment within the context of the work as presented in this paper. In the following section, these issues are addressed and discussed within the detailed description of the problem space, derived requirements, and formulation of a solution which is implemented as a prototype system.

3. Problem space, requirements, and approach

Visual analytics and information visualization combine user interaction and visual representation of data and information to facilitate analytical processes. These, in turn, are an integral part of the sense-making loop and decision-making. Therefore, efforts to create a solution space begin with analysis of the problem space. Related transitions toward requirements are then structured according to questions and goals relating to the value and functionality of the visual representations and interactions to be designed for the domain and application context, as outlined earlier. Those can be summarized as follows.

Firstly, which questions in regard to MCAD model assessment need to be answered where a fully automated solution is either not feasible / available yet or considerably inferior to an integrated semi-automatic approach? In particular, and related to that, which aspects of exercise requirements and known issues about CAD model deficiencies and shortcomings need to be located, identified, and verified, regarding their presence / absence in an MCAD model? Secondly, which circumstances have a considerable potential to enable and support detection and analysis of new, previously unknown issues and deficiencies related to MCAD models? Thirdly, which means are required within the context of MCAD model assessment in education to support the generation and documentation of which kinds of new information and knowledge? The last mentioned can be used immediately as input to a next instance in the sense-making loop, and/or added to the data and information space subject to analysis (see also externalizing of knowledge in [97], p. 436).

3.1. From problem space to requirements

By detailing what has been outlined above and applying it to the problem space of feature-based MCAD model assessment within the educational context, basic application domain related requirements can be formed for data and information representation and for user interactions. In what follows, a representative selection of those inquiries into the characteristics of the problem space and derived requirements is exemplified. These have been central during framework development and system design. It is important to recall that efforts, discussed in this paper, to enhance feature-based MCAD model assessment by integrating a software tool-based approach with visual analytics are aimed at supporting timely and high quality formative feedback. First of all, questions regarding the overall quality of an MCAD model, which represents a solution created by a student in response to an exercise assignment, need to be answered. Here, initial basic MCAD model assessment needs to address issues in several directions, which are, at various points, intertwined.

Firstly, attention must be given to the exercise requirements and projected learning outcomes. The validity of the MCAD model needs to be determined in terms of completeness in view of features (real objects domain) and the full-features and shape-features that are expected to be used to model them. Verification of the presence of all features of the real domain in an MCAD model, such as stiffening ribs and holes for bolt / nut or screw based fixtures, as stipulated by the exercise requirements, is most efficiently and effectively solved by a semi-automatic approach. This is especially true within the educational context, which is considerably different from an industrial / commercial context, as outlined elsewhere in this paper. For example, automatic verification will check the presence of features (real objects domain) according to the exercise requirements. This might include checking that the stiffening ribs were properly modeled using the correctly corresponding feature type, in this example a rib full-feature. Then unambiguous cases, where either no explicit corresponding feature type exists or an incorrect feature type has been used, would be analyzed by visually inspecting the respective visualizations. Note that in the case of an incorrect feature type used to model a design entity, conditions can be intertwined with those of an incorrect modeling context, as discussed later in this sub-section. This requires that the visual representation of feature-based characteristics of the MCAD model is capable of presenting a multi-dimensional / multivariate space based on each feature type (MCAD domain) and the number of actual feature entities of each feature type present in the MCAD model. Additionally, this representation of feature-based characteristics needs to be made available to the user in a manner linked to the graphical representation of the MCAD model within the CAD system that was used to create it. In particular, within such a linked view scenario, to support various user interactions during inspection and analysis, it is necessary for corresponding individual feature entities in both representations to be linked as well.

Secondly, in regard to exercise requirements and projected learning outcomes, the validity of a feature-based MCAD model in terms of the overall shape, as a result of the model's geometric / topological characteristics, needs to be assessed in view of the full-features and shape-features used to create it. Shape deviations in feature-based MCAD models, as known and frequently encountered within this context, can be attributed to particular errors which are typically committed by novices, and can be located and identified as follows. Extra features that are not supposed to be present in the MCAD model can be located and verified by visually inspecting simultaneously the MCAD model shape and the set of out-of-scope features within the two linked representations. Spatial interference between full-features and shape-features, caused by errors in the dimensioning and/or positioning during either the creation or the alteration of the feature geometry / topology, can also be effectively and efficiently located and identified by visual analysis of the MCAD model shape. Notice, however, that in this case the visual approach is limited in regard to the nature of the spatial interference (in particular size and type) in relation to the display resolution of the visualization environment and the functionality of system supported user interaction such as panning, zooming, and showing details on demand. Notice that several overall, as well as specific, feature-based MCAD model assessment criteria that are deployed within the assessment framework, such as the renaming of created full-features and shape-features, are not explicitly discussed here, as those are already implemented on a fully automated base within the CAD model assessment tool developed by the authors (cf. [69]). Here, assessment draws support from the framework and concepts of feature deficiencies and critical modeling situations developed by the authors and reported in [57,69].

Thirdly, feature-based MCAD models need to be analyzed in regard to particular deficiencies and shortcomings. Here, visual analysis is most effective and efficient, if the following requirements are met. For exploration and assessment of the actual modeling context of full-feature / shape-feature entities, a linked view is required of the representation of the feature-based characteristics and the MCAD model, as discussed earlier. However, the multi-dimensional representation space needs to be further structured to indicate whether a feature belongs to a feature type that is supposed to be present in the MCAD model, according to the exercise requirements, and thus represents an in-scope feature type, or is not supposed to be used, and thus represents an out-of-scope feature type. The presence of out-of-scope features can also indicate that particular features not required for the actual creation of the MCAD model have been used in a different context – for example as UNDO features or to recover from errors. Note that an unusually high number of entities of a certain feature type within in-scope features is sometimes also an indicator that a full-feature or shape-feature has been used within a modeling context that demands further analysis. However, in all those cases, to finalize assessment of the actual modeling context of a feature entity, relevant characteristics of the corresponding entity in the CAD model representation, such as the history and the current location within the feature modeling sequence, need to be made available to the user for inspection and analysis.

A common shortcoming of feature-based MCAD models created within the educational context is their low-level structure and low degree of robustness, issues that are also both adverse to model alterability. Reasons for that are various. For example, during MCAD model creation novices usually focus on the shape aspect and its implementation far more than on designing a proper modeling strategy and on how to incorporate aspects of the design intent. Moreover, students, due to being novices who are still in the process of learning and developing their skills, prefer to use the commands and modeling elements that they have mastered so far and are comfortable with, thus sometimes remaining somewhat reluctant to use more complex and challenging ones, although they are required to use these at some point within exercise work. Providing detailed high quality formative feedback regarding this shortcoming is both important and often quite demanding, and so is the assessment of MCAD models related to it. Instances of MCAD models with a low-level structure are usually indicated in the representation of the feature-based characteristics by an unusually low number of full-features and shape-features used to create the model. Note, however, that this condition is intertwined with the condition indicating incomplete MCAD models – that is, models where, according to the exercise requirements, some design entities, and consequently the features required to model them, are missing. Here, analysis also needs to take into account that a low number of features (MCAD domain) used to create the MCAD model can be an indicator that complex profiles have been used to create several features of the real domain at once instead of using individual full-features and shape-features as expected according to the exercise requirements and projected learning outcomes. Therefore, support of visual analysis within the assessment scenario outlined above requires, besides the linked view of representations and basic browsing and exploration functionality (as discussed elsewhere in this paper), also access to and interactions with profiles. These, in turn, need to be linked (through features) as entities with their geometric / topological characteristics in the MCAD model representation to entities in the representation of the feature-based characteristics.

During visual analytics, the visualization environment is used both for consuming existing data and information and for creating new information and knowledge. In case of the latter, to make the newly created information available as input for the next instances of visual analysis and/or additional downstream analysis performed at a time later, data structures and user interactions to document and store this existing data and information are required. To better maintain the integrity of data and information in regard to the proper functioning of epistemic user interactions (cf. [97]), while also supporting provenance of the newly created information, documentation and storing of the latter should be approached by replenishing entities considered relevant with notes and comments, and facilitating the creation of new types within the existing data and information. Within the application context as outlined earlier, this can be achieved most effectively, as well as efficiently, as follows. Firstly, data structures and interactions must be provided to the user for interactively annotating the individual entities within the representation of feature-based characteristics. In this manner, newly created information relating to knowledge and insight gained regarding the role of features and their characteristics, in view of the MCAD model assessment, can be documented and stored. Secondly, data structures and interactions must be provided to the user for interactively creating and editing model error and feature deficiency categories, which are then linked to individual entities within the representation of feature-based characteristics. This enables the documentation and recording of model errors and deficiencies in regard to full-features and shape-features known to be committed by students and which were found during model analysis. Moreover, it facilitates the documentation and archiving of information that led to new insights and perspectives. Note that, according to requirements regarding a linked view between the representation of feature-based characteristics and the MCAD model representation, data structure extensions and additional user interactions need to be integrated and linked across the data entities and generated visualizations of both representations.

3.2. From requirements to solution space formation

Using the requirements in regard to the problem space as presented above, layout design and base construction of the solution space can be approached as follows. To quantify and qualify multi-dimensional feature-based characteristics of MCAD models within an interactive computer-based visualization, an appropriate graphical method for displaying multivariate data in the form of a two-dimensional chart or diagram is required. Taking into account technological factors such as computer display size and resolution, and that graphical representations of feature-based characteristics and MCAD models need to be created in single-plot format and multi-plot format, which both have to be interlinked with the MCAD model representations, a space-efficient graphical method with a radial layout is most suited for the task. Moreover, radial charts and diagrams are often considered to be more aesthetic and natural, and therefore more memorable, than their linear counterparts (cf. [6,7,10,16,40,93]). Additionally, due to their cyclic structure, placing the first and last variable next to each other, their layout, compared to linear layouts, is more naturally fitting in view of the previously outlined requirement for the representation of feature-based characteristics regarding individual feature types. Another advantage of radial visualizations is their ability to provide a pictorial integrated representation of the whole, while also providing better visual cues for part-whole estimations and natural anchors relating to polar angles such as 0, π, etc. (see also [86]). Parts of these properties are also a contributing factor for the superiority of radial layouts in graphically showing outliers and commonality within data sets. Within radial visualization, the layout design can be organized using various encodings based on polygon (line), polar area (radial bars), and color.

Furthermore, factoring in the structure and user interaction requirements regarding the representation of feature-based characteristics, displaying multiple dimensions within a single chart or diagram in the form of closed polygonal profiles of a definite shape and size appears to be the most effective and efficient approach. This choice of encoding also assists in the avoidance of some known shortcomings, which make certain radial visualizations less effective in certain applications in terms of readability and support of exploratory user interactions (see also overview and discussions in [13,16,23,24]). These less effective applications are based on polar area encoding, such as *rose charts*, also known as *coxcomb charts* [12,21], radial charts and diagrams based on concentric rings or spirals [25,90], and pie charts (cf. [20,52]). The graphical method that was found to be most suited, while also having a promising potential for optimization in regard to visual encoding, layout design, and perception, is a line-based radial visualizations. This type of radial visualization is based on polygon encoding to display multivariate data in the form of a two-dimensional diagram and is known and referred to in the literature as a *radar* / *spider chart* [73,75,102], a *polar chart*, a *star plot* [24,26,109], or a *Kiviat diagram* [51]. In the remainder of this paper, this graphical method will be referred to as a Kiviat diagram.

4. Framework and system design

4.1. Information mapping and graphical representation

The basic graphical information display structure of the Kiviat diagram consists of a set of equiangular axes denoted by a_k, also sometimes called *spokes* (in the literature frequently used within references to star plots) radiating from a common center point, with each representing one data dimension. Within the context of this study, each individual axis a_k encodes one feature type denoted by FT_k together with a data point denoted by $\boldsymbol{p_k}$ located on it with a proportional distance from the diagram center that encodes data quantity (magnitude). This data quantity, which represents a feature count denoted by *FN*, is related to the number of feature entities of the feature type encoded by the axis which are verified to be present in a particular MCAD model. The feature scope denoted as *FC* determines if a feature type belongs to the set of in-scope features or the set of out-of-scope features. Now, feature-based characteristics, denoted as *FBC*, of the MCAD model can be graphically represented by connecting the data points $\boldsymbol{p_k}$ of all axes with straight line segments across all diagram sectors to form a closed polygon chain, that is, the polyline denoted by P_s.

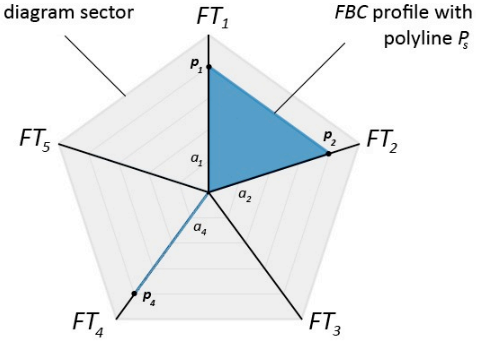


Fig. 2 Basic Kiviat diagram with one *FBC* profile.

This polyline then defines the geometries of property profiles regarding size, position, and shape within the diagram, which in turn forms a graphical representation of feature-based characteristics of the MCAD model according to requirements as outlined earlier. An illustrative example employing a Kiviat diagram based on such information mapping for feature-based characteristics consisting of just 5 generic feature types, denoted by FT_1, FT_2, FT_3, FT_4, and FT_5, is shown in Fig. 2. Notice that among those 5 generic feature types, 3 feature types, namely FT_1, FT_2, and FT_4, are defining in-scope features, while 2 feature types, namely FT_3 and FT_5, are defining out-of-scope features. Also note that concentric geometric structures such as circles or n-sided polygons, like the concentric regular pentagons shown in Fig. 2, can be added to the diagram as grid lines, to improve and make more efficient the visual judging and comparison of radial distances. This results in a layout that probably led to the naming of those diagrams as radar or spider charts.

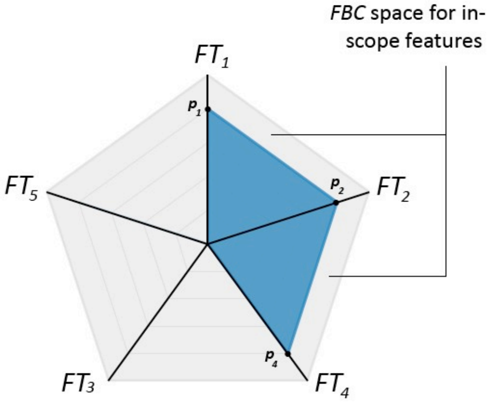


Fig. 3 Permutated Kiviat diagram with optimized *FBC* space and *FBC* profile.

An issue often pointed out in the literature is that there is a limit to the number of data sets which can be represented effectively on a radial diagram, and some representations may require a few hundred data sets (see also discussions in [53,109]). However, these considerations do not apply within the context of feature-based CAD models and graphical encoding as described earlier, because the number of feature types *FT* in MCAD models will always remain at least one order of magnitude below that limit. To optimize the explicit encoding of the feature scope *FC*, in regard to exercise requirements as discussed elsewhere in this paper, the order of the equiangular diagram axes needs to be rearranged and clustered in view of features considered to be out-of-scope and in-scope. In parallel with this task, the *FBC* profile needs to be optimized in regard to its polyline, which should represent a shape that can be recognized easily and quickly by the human visual perception system. This can be achieved by generating all possible circular permutations of the initial sequence $\{a_k\} = \{a_1,a_2,a_3,a_4,a_5\}$ of equiangular diagram axes and evaluating the resulting polylines taking into account some principles of Gestalt psychology (details are provided in the next sub-section). For the current simplified example, this process results in the permutated Kiviat diagram shown in Fig. 3, which is based on the sequence $\{a_k\} = \{a_1,a_2,a_4,a_3,a_5\}$ of equiangular diagram axes. The graphical layout of this permutated Kiviat diagram is now capable of efficiently and effectively representing a clustered *FBC* property space for both the in-scope features within the first two diagram segments and the out-of-scope features within the other diagram segments. The polyline of the profile also represents a more regular shape, in form similar to a tilted rhomboid, which is easy for the human visual perception system to recognize and identify. Structural differences and outliers, indicating deficiencies and errors in the MCAD model, can now be graphically presented in an efficient and effective manner, as shown in Fig. 4. In the Kiviat diagram shown in Fig. 4, the structural differences in the two *FBC* profiles can be recognized immediately. It can quickly be seen that there is a small difference in the proportion of FT_2 type features and a larger difference, amounting to a factor of 2, in the proportion of FT_1 type features. Also, the presence of FT_3 type features, that is out-of-scope features, can be noticed readily and instantaneously. Note that, in this simplified example, achieving the results as shown seems not to be that difficult. However, with real applications, where the number of circular permutations of diagram axes increases rapidly and the shape of the *FBC* profile sometimes tends to be intricate, finding an optimized solution becomes a non-trivial task.

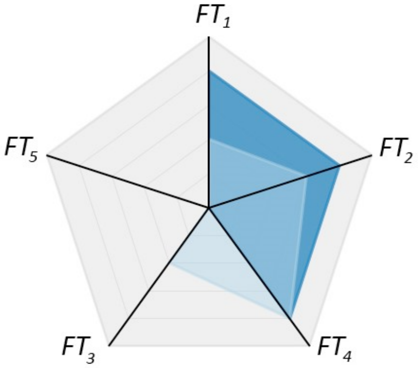


Fig. 4 Permutated Kiviat diagram with optimized *FBC* space and superimposed *FBC* profiles.

In general, as indicated in the previously outlined example, the mapping of information to graphical elements as outlined above can be further used to create multiple closed polygonal profiles that can be superimposed within one diagram to allow comparisons among several distinct data sets with common data dimensions and respective mappings. This allows, for example, a direct comparison of feature-based characteristics of MCAD model pairs created in two-part exercises aimed at skill development in MCAD model alteration. This approach is most effective when the area enclosed by one profile is entirely contained within another profile, or when transparency is used, allowing comparative analysis of relative areas without occlusion, as shown in Fig. 4. In what follows, further details are provided and various references given to basic and advanced literature regarding the actual visual encoding and layout design, and related strategies used to optimize the discriminability and perceptual visibility of the visualizations developed. Firstly, this includes *FBC* property space optimization, and, in particular, *FBC* profile optimization based on selected principles of Gestalt psychology. Secondly, it includes the development of an optimized color scheme designed as a system default, to ensure that the color combinations used in the visualization are universally legible with a design that is clear and accessible to both viewers with color impaired vision and viewers with full color vision. Thirdly, it includes the development of an improved layout design, which was further optimized based on the principles of the rule of thirds.

4.2. Visual encoding and layout design

4.2.1. Mapping and encoding of spatial properties

Data instances of the feature-based characteristics *FBC* of CAD models with their three components, that is, feature scope *FC*, feature type *FT*, and feature count *FN*, are mapped to a visual space using Kiviat diagrams as follows. Firstly, the entire circular graphical space is split into n equiangular axes a_k, which are drawn radially from the diagram center to its perimeter, where n is the data dimension determined by the n different feature types *FT*, with each diagram axis a_k, for $k = 1,2,3,\dots,n$, representing one feature type. Secondly, the feature count of *FBC* data instances for each *FT* is mapped to an entity position represented by a radial point $\boldsymbol{p_k}$ within the visual space, which is encoded as a distance r_k measured from the radial diagram center along the respective diagram axis a_k of the k-th feature type for which the feature count *FN* is encoded. Hence, the entity position $\boldsymbol{p_k}$ is defined by the pair (r, φ) of its coordinate values with the radial coordinate denoted by r and the angular coordinate denoted by φ, and computed as $r = \lambda r_k$ and $\varphi = (2\pi/n)(k-1)$, where λ is a linear coefficient used to normalize / adjust the graphical space (size) of the diagram in regard to the actual display medium. Thirdly, the layout of the radial diagram is sub-divided into two diagram sector clusters, with one sector cluster containing only encoded data entities of *FBC* data instances with *FC* indicating out-of-scope features, and the other sector cluster containing only encoded data entities of *FBC* data instances with an *FC* value indicative of in-scope features. Fourthly, all points $\boldsymbol{p_k}$ can be connected with straight line segments l_k to form a polyline denoted by P_s for $s = 1,2,3,\dots,m$, with m indicating the number of MCAD models. Here, the joint polyline P_s can be formulated as an alternating sequence of joint vertices, that is the $\boldsymbol{p_k}$ and links, that is the line segments l_k, expressed as $P_s = (\boldsymbol{p_1},l_1,\boldsymbol{p_2},l_2, \dots, l_n, \boldsymbol{p_{n+1}})$ with $\boldsymbol{p_1}$ and $\boldsymbol{p_{n+1}}$ sharing the same pair (r, φ) of coordinate values. This polyline P_s is the representative *FBC* profile, and thus the shape within the visual space that is graphically representing through the Kiviat diagram the feature-based characteristics of the s-th MCAD model.

4.2.2. Optimization of the diagram structure and property profile

Depending on the application context, there are various methods of optimizing the layout and property profile of radial visualizations. These methods include shape moments and less complex shape descriptors (cf. [54,72,110]) used as comparative metrics. In some cases, the diagram axes are rearranged, resulting in what is called in the literature a *permutated chart / diagram*. However, within the application context and data mapping / encoding as described earlier, the geometry of the *FBC* polyline needs to be optimized less in view of the technical issues which are important for automated processes, but more in regard to what is known about human visual perception and cognitive aspects related to it. It should be taken into account that, within the application context as outlined earlier, the reference CAD model, against which all CAD models created by students are assessed, is available as a reference solution for each exercise. Therefore, the issue of *FBC* polyline shape optimization related to axis reorganization can be approached by finding a spatial organization of the shape-defining elements, that is the joint vertices $\boldsymbol{p_k}$ and the linking line segments l_k, which can produce a shape easily and quickly recognized and understood by the human visual perception system. Note that, with this approach, which is limited to axis reorganization of the diagram, the original feature-based characteristics of the reference CAD model, or of any other CAD model subject to visualization, are not altered.

This approach has been pursued employing the *Prägnanz* principle of Gestalt psychology. In Gestalt psychology it is assumed that, when a group of objects is observed, perception of their entirety takes precedence over the perception of individual parts (see also [2,29]). Moreover, when individual parts are perceived, they are grouped according to certain rules of perceptual grouping, or, as Koffka put it, "The whole is other than the sum of the parts." ([50], p.176). Notice that here "other" does not necessarily mean "more" as is often claimed in the literature. Here the process of mentally forming patterns from simple basic rules is supporting an attempt to identify an outline or set of visual patterns and match it against shapes and objects that are known, in order to make sense of external visual stimuli. In other words, perceptual grouping attempts to describe the way the human visual system determines which parts and objects of external visual stimuli belong together to form a meaningful perceptual unit. In this regard, perceptual grouping can also be considered as one process by which diverse parts of a visual scene can be aggregated into higher-order structures. In Gestalt psychology and the study of perceptual grouping, the fundamental principle, referred to as *Prägnanz* (cf. [29,99,100]), implying conciseness and orderliness, but also known as the principle of *good Gestalt* (see also [94]), is based on all the concepts described in [99], and known as the original *factors* or *principles of perceptual grouping* such as similarity, proximity, and good continuation (cf. [15]). These principles not only allow us to predict the interpretation of how external visual stimuli are perceived, but also determine the best Gestalt possible based on what is visually given. The basic underlying assumption here is that a particular organization of graphical elements may be favored due to its being better than other organizations. Here, in particular, this means that the human perceptual system tends to favor an organization of perceptual groupings that allows for a regular, orderly, symmetrical, and also simple rather than complex, perceptual experience of the physical world. This was explained by one of the leading Gestalt psychologists in the words, "Of several geometrically possible organizations that one will actually occur which possesses the best, simplest and most stable shape." ([50], p.138). However, Prägnanz, like some other original basic grouping principles, is still without a clear and formal definition, which is partly because Gestalt psychology and its methods were largely based on demonstrations (see again discussions in [15,94]). Therefore, within the work presented in this paper, Prägnanz and related grouping principles were used as heuristics to approximate a good Gestalt for the shape representing feature-based characteristics based on the *FCB* polyline of the reference CAD model. This, in turn, results in a permutated Kiviat diagram that is improved in view of human visual perception through the rearrangement of the diagram axes. However, a more detailed discussion is beyond the scope of this paper.

4.2.3. Mapping and encoding of color

The design and creation of color sets, also in the literature referred to as *color schemes* and *color palettes*, which are used for the mapping and encoding of color during the visualization, can be approached using different strategies depending on the application field. For artistic and more general-purpose design applications, efforts are aimed at achieving color harmony and aesthetics as shown, for example, in [43,60,66]. Here harmonic color sets are usually created based on hue relations, which are derived from harmony principles in color theory and art (cf. [44,63]). However, a critical issue with such an approach is that recent results gained through empirical studies show that actual human judgments in view of color harmony do not necessarily correspond with those derived from color theory and art (for example, see discussions in [70,80]). For more technical, rather than artistic, applications such as visualization, different strategies can be pursued. Those can relate more to technology, such as display energy consumption [19], or perceptual issues such as discriminability, which was approached, for example, through optimization of perceptual visibility and perceptual distance [39]. Here also crowdsourcing and linguistics-based approaches ought to be mentioned. These aim to improve the modeling of color-term associations either by aiming at semantically meaningful color sets [55,81] or by fitting statistical models to human judgments [42,65]. Recently, promising approaches have employed a strategy that combines aspects of perceptual distance, name difference, and color preference, as reported, for example, in [36]. These provide a more balanced approach by taking into account strategy aspects related to both artistic and visualization applications. Within the visualization environment described in this paper, various settings for the mode of color scheme relate to particular strategies that were used for the color scheme design as follows. To optimize design flexibility and customizability allowing for a personal overall color preference, and in particular for color pairing preferences, while supporting individual choices of color harmony, particular color schemes can be created by the user. This can be achieved by directly defining the color scheme through either the input of the parameter values of a color specification system, for example in form of the additive primaries red-green-blue (RGB) used to produce emitted color, or by selecting particular colors using a color picker tool.

To optimize strategies based on discriminability and perceptual visibility as used in various visualization applications, while also attempting to encapsulate as much knowledge and insight as possible regarding the human visual perception system and the use of digital color in visualization (again see [64,88,97]), an optimized color scheme has been designed as a system default. However, it can be overwritten by selecting a different setting for the color scheme mode as outlined above. Taking into account that color vision impairment is probably the most widespread physiological impairment (cf. [8,62,67]), this color scheme has been designed to ensure that the color combinations used in the visualization are universally legible. With such a barrier-free design, the visualization is clear and accessible to both the color impaired and the viewer with full color vision. The current design is based on a customized color scheme for qualitative data, which employs differences in hue to represent differences in data type. This color scheme was created with the online tool *ColorBrewer2*[1] [14,37]. The current design encapsulates a color scheme sub-category that is one of the *paired color schemes*, "which present a series of lightness pairs for each hue" ([37], p. 31). This sub-category of the qualitative color scheme is necessary for the color encoding of the *FBC* polylines for pairs of CAD models that were created in two-part exercises. The legibility of the current customized color scheme design has been verified with a complementary software tool, namely *ColorOracle*[2] [46], an open-source simulator of color-impaired vision. The test results confirmed the legibility of the color scheme design for the most frequent color vision impairment related to forms of what is commonly referred to as red-green confusion (see also deuteranopia / deuteranomaly and protanopia / protanomaly in [8,62]). To integrate some aspects of strategies used in artistic and visualization applications, rather than aim for optimization, as described earlier, a third mode for the design and creation of the color scheme is provided. Here several pre-defined color schemes such as those from Microsoft Excel and Tableau, are linked to the visualization environment, from which the user can select individual colors to create a custom color scheme.

4.2.4. Optimization of the layout design

Within the visualization environment presented in this paper, the graphical layout design is based on Kiviat diagrams that are structured according to the data mapping and the visual encoding as described earlier. To accommodate the individual visualization scenarios as required for visual analysis of the graphical representation of feature-based characteristics of MCAD models, the graphical layout has been designed in regard to both single-plot format and multi-plot format. In single-plot format one diagram can accommodate up to three *FCB* polylines, which are then associated with an MCAD model pair from a two-part exercise and the reference CAD model. In the case of a one-part exercise, either a single *FCB* polyline is shown, or two *FCB* polylines are shown that are associated with an MCAD model and the reference CAD model respectively. To allow for adjustment of the visual appearance, particular elements of the layout can be toggled off and on. These include text-based descriptions along the diagram rim and the diagram axes indicating the feature type and the number of features, the radial concentric grid, and the straight lines emanating from the diagram center representing the equiangular axes. Also, the graphical representation of feature-based characteristics, in the form of the *FCB* polyline, can be changed into a filled polygon upon user request. Note that some aspects of the visualization layout design take into account the well-known composition recommendation commonly referred to as the *rule of thirds*, though in a broader context including visualization applications (cf. [7]). This has resulted, among other layout design decisions including those related to the dashboard, in limiting text entities to three typefaces, in an effort not to "… overlook the role that type plays in legibility, aesthetics, and meaning construction" ([7], p. 45). In multi-plot format many individual diagrams are generated within one visualization space, with each diagram representing feature-based characteristics of one MCAD model or of an MCAD model pair that may also include the reference CAD model, as requested by the user. To support fast and efficient visual identification of cases where out-of-scope features are present in a diagram, the default white background of this diagram is changed into either another diagram background color default (indicating the presence of out-of-scope features) from within the visualization environment or a user-defined diagram background color.

---

[1] https://colorbrewer2.org.

[2] https://colororacle.org.

4.3. Interaction techniques for visualization and analysis

4.3.1. Overview

If visual analytics is considered as an interactive and iterative dialogue between the user and the computer, then analysis as a process consists of a sequence of user interactions with and responses by the computer [48,83,87,89]. As visual analytics is not primarily about visually presenting information, design and implementation of visual aspects and interactive aspects are equally important to make visual analytics effective and efficient from the perspective of both the user and the visualization system. This approach has been successfully pursued in the field of human-computer interaction, where interaction design and user experience are major research themes. Unfortunately, this is different in the visualization field. Here research on how visualization systems can best support the analytical discourse, and the role of interaction in visualization and visual analytics (also see discussions in [27,91,95]), remains somewhat neglected and thus has not matured to the point where insight and recommendations can be compiled into a sound framework and translated into concrete applications in a straightforward manner. User interaction design for visualization and analysis as presented in this paper has been structured and organized according to the primary use. Therefore, this has been approached separately in regard to visualization, that is interactive modification of view and data transformation, and visual mappings, and in regard to analysis, that is the human-information discourse that is a higher-level user dialogue with the information to gain insight based on what has been visually represented and interactively modified (cf. [17,48,89,95]), as follows.

4.3.2. Modification of view transformation and data transformation

Interactive modification of the view transformation to navigate the visualization space has been designed in view of both single-plot format and multi-plot format for Kiviat diagrams. Interaction techniques allowing the user to directly select and highlight objects of interest are provided along with basic techniques for panning and zooming, as described, for example, in [64,87,97].

The design of interactive modification of data transformations to adjust the type and amount of data that is provided in the visualization space currently supports filtering based on query filters, details-on-demand [82] (sometimes also in the literature referred to as *drilling down)*, and coordinated / linked multiple data views based on linking and brushing, also referred to as *linked highlighting*, *cross-filtering*, and *cross-view brushing* (cf. [64,97]).

Currently, the interaction design for filtering within the visualization environment as developed and presented in this paper is based on the general approach of dynamic query filters, to enable the user to modify dynamically the number of entities in the data set that is displayed in the visualization. The query filters are structured in a manner similar to query filters presented in [5]. However, instead of using sliders to interactively control the query filters to filter out or add back data entities to the visual display, values to control the query filters need to be interactively inputted / selected, after which the visualization display is updated. This interaction design approach allows basically for item filtering, which is different from attribute filtering (cf. [106]). In the case of the latter, this results in a modification (usually reduction) of data dimensions in the filtered data set and the visual display, whereas in case of the former the number of entities in the data set is filtered according to their attribute values, that is, filtered along each data dimension within a multi-dimensional data set, thus modifying the number of data entities subject to visualization, while retaining all data dimensions. Within the current query design, item-based filtering across the multi-dimensional space of feature-based characteristics of MCAD models can be pursued along feature types, number of features, and feature scope.

Within information visualization and visual analysis, the importance of the interplay between the need for overview and the need to see details is clearly expressed in Shneiderman's still influential visual information seeking mantra, "Overview first, zoom and filter, then details-on-demand" ([82], p. 337). The last-mentioned is a high-level task abstracted as the selection of an item or group within a visualization, with details obtained when needed. This is usually approached by enabling a mouse click on or a hover query over a symbol or graphical mark from within the visualization space to open an information panel, which reveals further detailed information by showing additional attributes of the selected data entity. Within the current framework on interactive modification of data transformations, the information panel is designed as a floating pop-up window, which the user can interactively resize and reposition, depending on which portion of the data overview, that is the Kiviat diagrams in multi-plot format, needs to be visible for the current analysis in progress. This design is aimed at combining elements of semantic zooming [59] where the representation of the items is qualitatively different from simply being drawn larger than the version in the overview, with elements of interactive distortion techniques that preserve an overview of the data while showing details for the current focal point during drill-down operations. Note that details-on-demand tasks are supported by the background coloring of the Kiviat diagrams (described elsewhere in this paper) in a manner similar to what is known as *information scent* [74], as it provides visual clues suggesting where to zoom in, drill down, and analyze further.

Brushing consists of a data selection method that is usually specified in a direct manner by selecting data elements through combinations of mouse / cursor motions and button clicks in one view that are then highlighted in that view and also simultaneously in other views of the visualization space [4]. In the last three decades selection techniques and models for brushing have advanced considerably, and also include, for example, eye / head tracking, gestures within virtual reality environments, kernel density estimation, and neural networks (see also discussions in [30,96]). Brushing techniques can be based on screen space, data space, or structure space (cf. [31,41]), depending on where the interaction takes place. Within the current user interaction framework as presented in this paper, the design of linking and brushing, also in the literature referred to as *cross-filtering* and *cross view brushing* (cf. [64,97]), is defined in data space, because it allows for a data-aware selection in regard to features and their respective types. Once a set of features / feature types (MCAD domain) has been selected (brushed) in the view representing MCAD model features and feature-based characteristics that is visualized based on Kiviat diagrams, it will be simultaneously displayed in highlighted form in the cross-linked CAD model view of the MCAD modeling environment. Then analysis can proceed in parallel within the topological and geometrical context of feature-based MCAD models. The current design of linking and brushing supports two view modes, namely *compact view mode* (CVM) and *wide view mode* (WVM). The former is designed for a hardware setting with one physical display device, such as a laptop. The latter is designed for hardware settings with more than one physical display device, where individual cross-linked visualizations can be displayed across several physical display devices.

4.3.3. Modification of the visual mapping

Interactive modification of the visual mapping by the user is currently restricted to the alteration of certain parameter value settings that impact visual encoding in regard to color, spatial location, and shape of graphical entities and the layout of the radial diagrams, which also includes permutation of the diagram axes. This approach to user interaction may appear somewhat limited compared to techniques such as pivot tables and dataflow systems, where the user has a considerable amount of control over the interactive modification of mappings between the data and their visual representation. However, as the visualization environment presented in this paper was developed specifically for feature-based MCAD model assessment within the educational context, the design of the data mapping and visual encoding (presented elsewhere in this paper) has been optimized to facilitate effective and efficient support for visual analytics in this specific application field. This design approach for interactive modification of the visual mapping has been further pursued by encapsulating as much knowledge as possible on human-computer interaction, human visual perception, user interaction design, and multi-dimensional information visualization within the systematic default settings of the visualization environment.

4.3.4. Interactions for the human-information discourse

User interactions for the human-information discourse are currently designed to support processes related to comparing data, categorizing data, and annotating data. Comparing data in regard to feature-based characteristics is supported in single-plot format by allowing up to three *FCB* polylines to be graphically represented simultaneously within one Kiviat diagram. This allows for comparing feature-based characteristics of either a pair of MCAD models created during a two-part exercise or combinations of single MCAD models / MCAD model pair and the reference CAD model. In multi-plot format the data comparing support for the single-plot format is extended to allow for comparing feature-based characteristics in a similar manner across an entire exercise, including all student-created MCAD models. Data categorization is supported by allowing the user to interactively assign features (MCAD domain) of an MCAD model to either a known or a newly created model deficiency category in regard to the errors, deficiencies, and shortcomings of an MCAD model which have been detected during the analysis process in view of particular modeling situations and exercise requirements. This results not only in newly gained knowledge and insight, but also in its integrated digitized documentation, and thus considerably increased potential to be input for an iterative step within the sense-making loop and its structures aimed at supporting the whole knowledge discovery and MCAD model assessment process (see also discussions on analytic and synthesized knowledge in [34,95]). Currently, data annotation is also designed to support integrated digitized documentation of synthesized and newly created knowledge and its provenance in the form of textual annotations (see also produce goals in [64]; modern integrated visualization systems in [95]; externalizing of knowledge in [97]). Annotation is captured interactively through direct user input and becomes part of the data in a manner similar to a data attribute. In this way, it can be used explicitly and without technical difficulty for visualization and analysis later on, or in some other downstream tasks. Annotation can be used to capture knowledge and insight in view of previous assessment results and thus represents a kind of meta-knowledge. It can also be used to document additional findings and circumstantial facts relating to the analysis and the CAD model assessment. Moreover, annotation can be used as a kind of analytic provenance, documenting certain aspects of the advances and progress of the analysis process, and, related to it, the newly created knowledge and insight (cf. [35,77]). Notice, however, that, although provenance and related tracking techniques have recently gained interest within visualization system development, provenance that is related to insight and analytical findings can still be achieved only by using manual annotation.

5. Development and implementation

5.1. System implementation

As the prototype implementation of the visualization environment module needs to be integrated with previous work of the authors on software assessment tool development (cf. [69]), data management and transformation within the visualization pipeline have to operate through the CAD model and feature entity (CMFE) repository that in turn facilitates the import from and export to different parametric feature-based solid modeling environments. Within the CAD model feature entity (CMFE) inventory these are then compiled, together with results, into the model entity analysis reports.

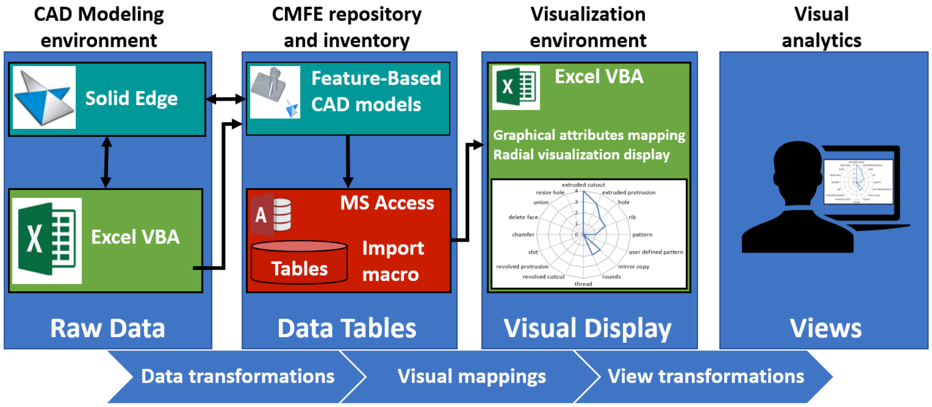


Fig. 5 Overview of implemented system components and their disposition within the information visualization pipeline.

The newly developed prototype implementation features a technical architecture that leverages API-based functionality provided by commercially available CAD systems to support a modular and highly cohesive system architecture. In the current implementation, the CAD modeling environment deploys a commercially available parametric feature-based solid modeling system, namely *SolidEdge*[3] from Siemens AG. At present, the CMFE repository is compiled by extracting CAD data from the *SolidEdge* part models using Visual Basic for Applications (VBA) functions. This extracted data is then further processed and stored in structured *Excel* files. Next, those structured *Excel* files are imported into the Microsoft *Access* RDBMS (relational database management system) by means of macros, to facilitate the creation and build-up of the CMFE inventory. Currently, the modularized visualization environment as shown in Fig. 5 is implemented using *Excel*, the VBA environment, and a data pipeline to the CMFE inventory that is channeled through compiled subsets of query reports.

5.2. Data set and format compilation

To process and visualize data correctly, compilation, export, import, and filtering within the information visualization pipeline in regard to the CMFE repository / inventory are organized as follows. All feature-based solid models that have been created by students are compiled and stored in the CMFE repository. This repository is structurally sub-divided into various sets of folders, with one set of folders for each exercise or course assignment. During the compilation process, information on feature entities and their related properties and meaningful characteristics, such as feature type, shape defining topology and geometry, is extracted from the parametric feature-based solid models, codified, and stored in the form of structured files, with one file for each model. Data on parametric feature-based model entities and their properties and characteristics stored in the model repository are processed and imported into the CMFE inventory. This inventory provides a lattice-based data structure, which is structurally organized as various linked entity tables. Data compiled from CAD models associated with a particular exercise or course assignment are assigned to one particular cluster of entity tables. It should be noted that table entries for each feature entity in the model repository contain also an identifier-based link, which connects them to the geometric modeling system. Note that this link mechanism is essential for enabling the implementation of a cross-linked view supporting linking and brushing.

5.3. Interface components and user interactions

The visualization environment prototype as developed and implemented consists of several individual interface components. There is a main visualization window, which is a kind of visualization canvas that provides an interactive viewport into the two-dimensional graphical representation of feature-based characteristics of MCAD models based on radial visualization in single-plot mode and multi-plot mode. During visual analytics users can navigate the entire main visualization window through panning and zooming using basic mouse interactions. A three-part information / interaction panel, which is implemented as a set of floating windows that can be interactively resized, re-positioned, and switched on / off by the user, provides detailed information on demand in regard to feature-based characteristics of MCAD models. This panel also allows for user interaction to enable annotation and categorization of data entities. A cross-linked view to the CAD environment facilitating linking and brushing, where selected data entities from within the main visualization window are shown in highlighted form in the CAD environment, represents another interface component, which not only enables users to obtain more detailed information but also enables an additional linked viewport into another data entity representation space. Finally, the interface component regarding the centralized configuration area is implemented in the form of a dashboard, which allows the user to view and manipulate basic settings of the visualization environment by single click selections, on / off buttons, and variable fields.

---

[3] https://solidedge.siemens.com.

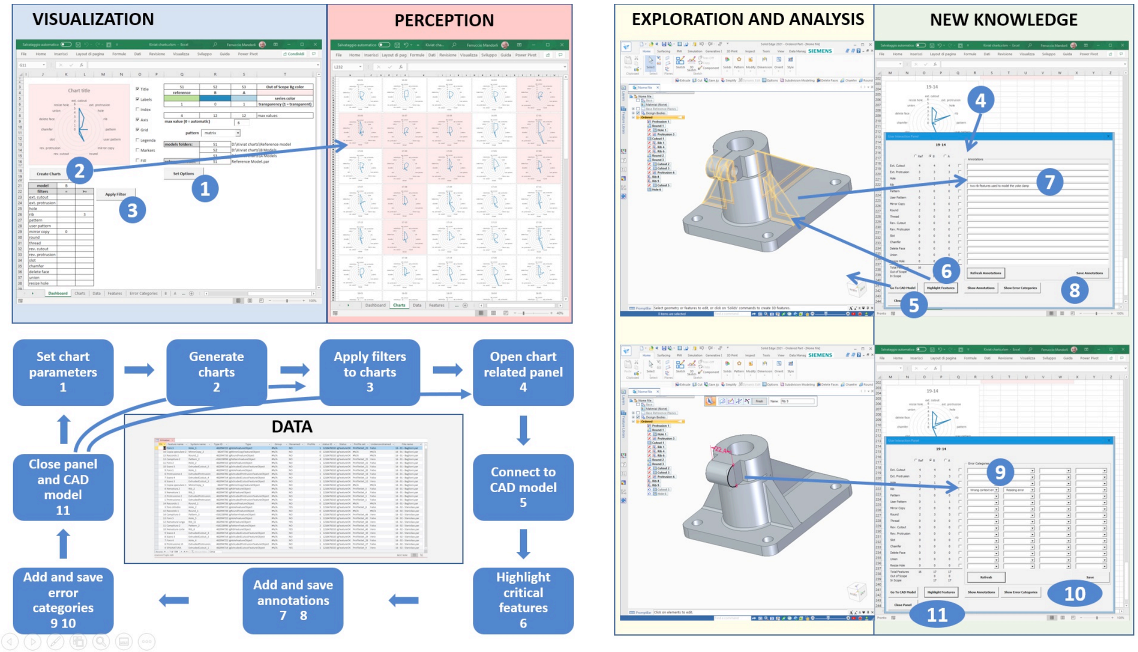


Fig. 6 Example of user interactions and related interface components during data visualization and exploration within the sense-making loop of visual analytics.

To provide an overview on the visual appearance of the visualization environment and the manner in which user interactions pertaining to visual analytics are supported by this prototype implementation, a section of an actual analysis and sense-making session is presented. This section was indeed executed as a segment of the MCAD model analysis example in connection with the diagram shown in Fig. 9(a), which is a part of the presentation and discussion of sample applications in the next section. Some selected individual user interactions and the operational steps involved in regard to the related interface components and cross-linked views of the visualization environment, as well as the MCAD modeling environment, which are shown in Fig. 6, are as follows.

After some semi-automated analysis based on filters and queries in regard to basic assessment criteria of MCAD models within the educational context as described in [69], data can be transformed and visualized to help gain further knowledge and insight. Here, exploration and discovery are structured through processes of the sense-making loop, while being supported by visual analytics. Within the visualization pipeline described earlier, pre-processed and compiled data from the CMFE inventory (see again Fig. 5) can be used as input for visualization. Generally, the first user interactions involve the setting of basic visualization parameters and the initial generation of Kiviat diagrams within a data display in multi-plot mode. Also, some filters, whenever deemed necessary, can be applied to adjust the data space being visualized. Those initial interactions can be performed using the dashboard of the visualization environment, as shown on the upper left-hand side in Fig. 6.

Next, through perception in regard to the human visual system, pre-attentive processing, human cognitive abilities, and the analysis goal pursued, first areas of interest within the data display can be quickly and efficiently identified and selected. In this example, with the goal of detecting outliers during visual analytics aimed at recognizing and making sense of trends and patterns, a Kiviat diagram depicting an unusually high number of rib features (see also Fig. 9(a)) was selected.

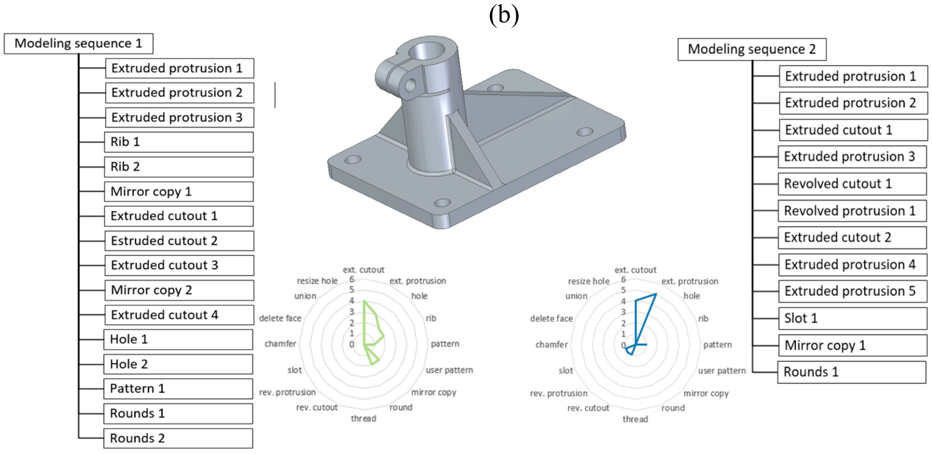
To further explore and analyze this situation, while also trying to build a more complete mental model of the data entities and the related feature-based modeling context, a cross-linked view to the MCAD modeling environment was established. Linking and data space brushing across these coordinated multiple views resulted in a corresponding visual representation of the actual MCAD model associated with the selected diagram, and within it the highlighted features subject to analysis, namely the five ribs, as shown in Fig. 6. Further exploration and analysis revealed that two rib full-features had been used by mistake to model a part of the yoke clamp head (see also lower part within the block on exploration and analysis shown in Fig. 6). This kind of MCAD model deficiency, which was created by a student, was neither apparent to the teacher nor ever encountered in any related previous modeling exercise. Hence, it represents, from the perspective of the teacher and the MCAD model assessor, new knowledge and insight on the kinds of errors that can be committed by students in this particular exercise and feature-based modeling context. Subsequent user interactions to document this newly gained knowledge and insight by manually annotating those data entities, that is, the wrongly used rib features, and also forming a new error category with those data entities associated with it, are supported by different parts of the user interaction panel as shown on the right-hand side in Fig. 6. At this point the importance of manual user-driven data annotation needs be pointed out again, due to its second role as a kind of analytic provenance, documenting certain aspects of the advances and progress of the analysis process, and, related to it, the newly created knowledge and insight.

Note that the size and spatial position of individual windows of the visualization environment and the MCAD modeling environment can be freely arranged by the user employing currently supported viewing modes (CVM / WVM) in combination with hardware settings consisting of one or several physical display devices. The layout of the interface components and cross-linked views as illustrated in Fig. 6 is just one example chosen to accommodate, as well as to spatially optimize, the figure layout as shown.

6. Sample applications from within the educational context

6.1. Overview

In this section, presentation and discussion of selected material will be confined to one exercise consisting of two segments. This exercise relates to the modeling of a bolted yoke clamp mechanism for rod fastening, with stiffening ribs, and a likewise bolted rectangular mounting base. In general, there are several feature command sequences that can result in a valid MCAD model that conforms to the shape and geometry required for the exercise. However, within the educational context of this exercise, students are required to take into account elements of modeling guidelines and best practice as taught in the course lectures. This includes, for example, creating the full volume first, then all the cutouts, and leaving the creation of rounds until the end of the modeling sequence, as shown in Fig. 7(a).



(a) (c)

Fig. 7 Feature-based MCAD model created with various modeling approaches. From left to right: (a) correct and recommended feature command sequence and related Kiviat diagram, (b) rendered shape of the feature-based MCAD model, (c) deficient feature command sequence and related Kiviat diagram.

Among the various modeling strategies possible, one that is effective, efficient, and considered to be adequate as a reference within the given educational exercise context leads to a model creation that requires the application of 16 feature commands (see again Fig. 7(a)). An example of an alternative, though deficient, modeling strategy using just 12 feature commands to create the same MCAD model shape (see Fig. 7(b)) is shown in Fig. 7(c). The actual number and type of features used in the two modeling sequences are shown in Table 1. Note that, although the model shape created with this alternative sequence of feature commands is identical to the previous modeling example (see again Fig. 7(a)), the entire modeling outcome, and most importantly, the basic feature-based characteristics of the MCAD model, not only differ, as can be seen in the related Kiviat diagrams, but exhibit several errors and shortcomings.

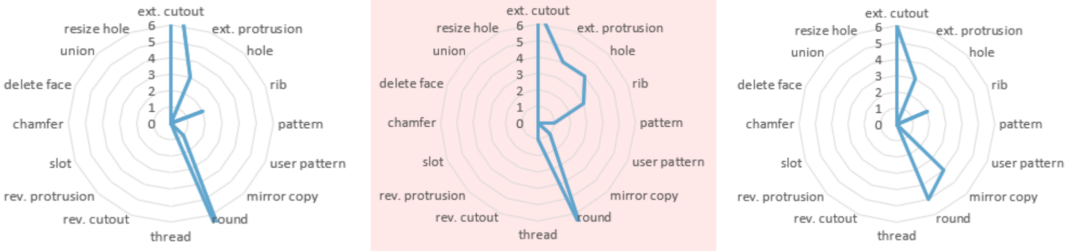
Table 1
Number of in-scope features and out-of-scope features within feature-based MCAD models that were created with various modeling approaches.

Feature Type (*FT*)	**Modeling Sequence 1**	**Modeling Sequence 2**
Extruded Cutout	4	3
Extruded Protrusion	3	3
Hole	2	0
Mirror Copy	1	1
Pattern	1	0
Revolved Cutout	0	1
Revolved Protrusion	0	1
Rib	2	0
Rounds	2	1
Slot	0	1

More details on the description of the CAD modeling exercise, the reference modeling approach, the reference CAD model, and the reference CAD model deficiencies can be found in [69]. Note that within the given context those reference structures are used as a means of embodiment of information and knowledge about important facets of the basic goals, outcomes, and concepts and methods that are relevant for each individual exercise. This foundation is of considerable value for various purposes, especially for the assessment of produced outcomes.

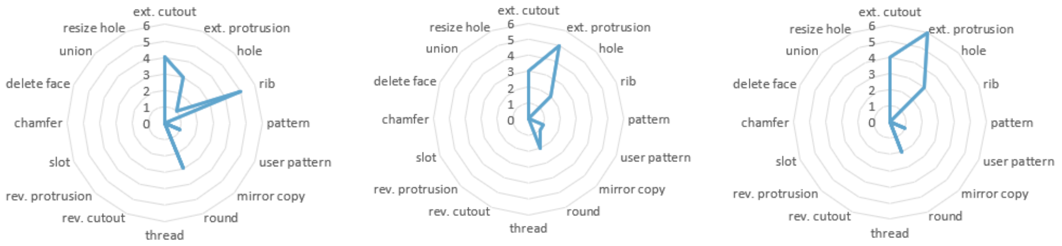
6.2. Trends and patterns in data sets

Visual analysis of the display in multi-plot mode immediately reveals a series of outliers, that is Kiviat diagrams of related CAD models where an unusually large number of features was used for their creation. Examples of cases that show large numbers, up to 75% above the average count, for both round features and extruded cutout features are depicted in Fig. 8(a) and Fig. 8(b). Notice that the diagram depicted in Fig. 8(b) has a colored background, indicating that out-of-scope features are also present. An example of cases where a large number of mirror copy features was found, along with a number of round features and extruded cutout features above the average count, is shown in Fig. 8(c). Further data exploration revealed that these patterns usually occur in CAD models with an *FN* value that is at least one-third above the average count. Furthermore, usually those CAD models have not been altered and resubmitted for the second exercise segment. An example of cases where a large number of rib features was found is shown in Fig. 9(a). However, it was found that those cases were mostly linked to CAD models containing an average count of features, and that they may or may not have been altered and resubmitted for the second exercise segment.



(a) (b) (c)

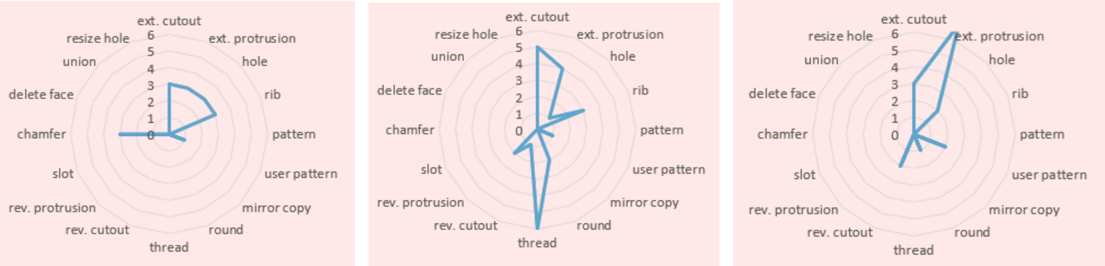
Fig. 8 Example of a selection of Kiviat diagrams taken from a data display in multi-plot mode. From left to right: (a) large number of both round features and extruded cutout features, (b) large number of both round features and extruded cutout features, (c) large number of mirror copy features and number of round features above the average count.



(a) (b) (c)

Fig. 9 Example of a selection of Kiviat diagrams taken from a data display in multi-plot mode. From left to right: (a) number of rib features above the average count and mirror copy features missing, (b) number of extruded cutout features below the average count, number of extruded protrusion features above the average count, and rib features missing, (c) number of extruded protrusion features above the average count and rib features missing.

Visual analysis of the display in multi-plot mode also revealed another series of outliers, that is Kiviat diagrams of related MCAD models where particular types of features (MCAD domain) were missing, even though, within this exercise, those feature types were deemed necessary for the proper creation of the MCAD models. For example, typical cases where rib full-features were found to be missing in the MCAD model are shown in Fig. 9(b) and Fig. 9(c). Notice that those cases are easy to recognize visually, as they always appear graphically in the first quadrant of the diagram in the form of a quadrilateral that is shaped similarly to a tilted rhomboid. Further data exploration revealed that those cases usually occurred in CAD models with numbers of extruded protrusion features above the average count. This initial finding was consistent with results of the detailed analysis and model assessment, which showed that in CAD models without rib full-features, the ribs (real objects domain) were modeled by using extruded protrusion shape-features. Visual analysis was also very effective and efficient in quickly revealing an overview of all cases of MCAD models that contained out-of-scope features (MCAD domain). Moreover, some aspects of the nature of those out-of-scope features could immediately be recognized based on their graphical representation in the second and third quadrants of the diagrams. For example, in Fig. 10(a) a case is shown where several chamfer full-features had been used in a CAD model that lacked any round full-features. Further detailed analysis was required to find out whether, among other shortcomings, those chamfer full-features were actually used to model rounds (real object domain). Further examples of cases where CAD models contained out-of-scope features (MCAD domain), such as thread full-features, or revolved cutout shape-features, are depicted in Fig. 10(b) and Fig. 10(c). As each of those cases indicated the presence of deficiencies in the related CAD models, a more detailed analysis was required to enable formation of the formative feedback. For the case depicted in Fig.10(c), a fully annotated example of such a detailed analysis is provided in the sub-section below.



(a) (b) (c)

Fig. 10 Example of a selection of Kiviat diagrams taken from a data display in multi-plot mode. From left to right: (a) chamfer features are present, (b) revolved protrusion features and thread features are present, (c) revolved cutout features are present.

The various visualization settings permit up to three *FBC* profiles to be graphically represented within one diagram, as either line-based contours or color-filled areas. Also, direct comparative visual analysis is explicitly supported. This is important for scenarios where feature-based characteristics of CAD models are compared with their counterparts in the reference CAD model and/or the different versions of individual MCAD models that were created in two-segment exercises.

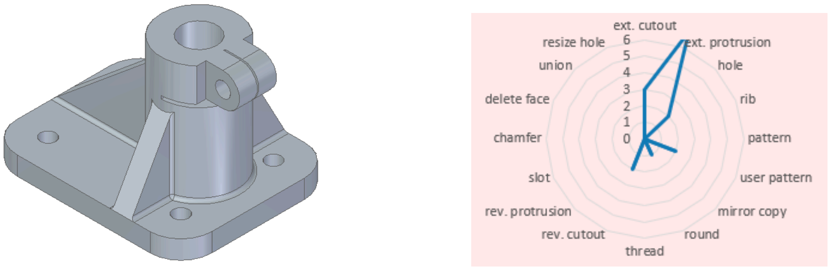
(a) (b) (c) (d)

Fig. 11 Example of a selection of Kiviat diagrams taken from a data display in single-plot mode with various visualization settings. From left to right: (a) single line-based *FBC* profile of a MCAD model, (b) single area-based *FBC* profile of an MCAD model, (c) superimposed line-based *FBC* profile of an MCAD model pair, (d) superimposed line-based *FBC* profile of an MCAD model pair and the reference model.

An example relating to this is shown in Fig. 11(a), which depicts a diagram of a CAD model that was created and submitted for the first exercise segment. It clearly depicts several deficiencies and shortcomings, such as missing rib full-features, a number of extruded cutout shape-features which is only about half of the average count, about 5 times as many round full-features as necessary, and more hole full-features than required. In Fig. 11(b) and Fig. 11(c) the final modeling result and the quantitative and qualitative improvements of the same CAD model, which was altered and resubmitted for the second exercise segment, are quickly and clearly visible. Among these are, for example, modification and adjustment of the application, and thus also quantity, of round full-features and hole full-features used to create the MCAD model. Finally, a look at Fig. 11(d) confirms that most of the alterations, quantitative as well as qualitative, that were made to the MCAD model can be considered amendments that improved the overall quality by bringing its feature-based characteristics closer to one example of a suggested good MCAD modeling exercise solution. However, despite the improvements, some deficiencies and shortcomings, such as the missing rib full-features, obviously still prevail in the resubmitted MCAD model. For those issues, a more detailed analysis is required, before formative feedback can be offered. The kind of analysis and assessment briefly outlined in the examples above integrates graphical representation of qualitative and quantitative aspects of MCAD model characteristics with explorative and analytical visual methods. Note that this is very difficult, and in some cases almost impossible, to achieve exclusively with automated query and filter based methods, especially taking into account the limited resources available for teaching at institutions of higher education and the steadily increasing number of students in MCAD education.

6.3. Making sense of data and the compiling / documenting of newly obtained knowledge

By obtaining more detailed information and an additional view by drilling down and making use of the cross-view link and brushing, particular MCAD models can be analyzed in more detail as follows. To keep the example transparent, while conveying all the points considered essential, the focus will be on one MCAD model that is considered a typical case representative and which has already been included in the previous assessment, with its Kiviat diagram shown in Fig. 10(c)).

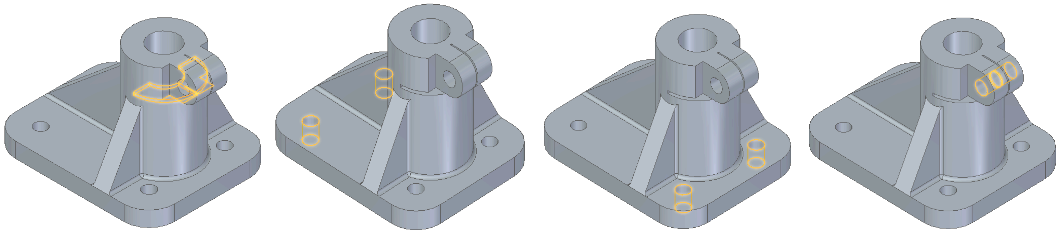


(a) (b)

Fig. 12 Example of actual MCAD model containing several deficiencies. From left to right: (a) rendered shape of the MCAD model, (b) related Kiviat diagram in single-plot mode.

6.3.1. Assessment details and feedback formation

Visual analysis of the Kiviat diagram for the MCAD model, as shown in Fig. 12(b), immediately reveals several deficiencies and shortcomings, which, however, are neither obvious nor even visible to the naked eye in the rendered shape of the MCAD model, as shown in Fig. 12(a). First, out-of-scope features (MCAD domain), in the form of two revolved cutout shape-features, are present. Second, rib full-features are missing. Third, mirror copy features (MCAD domain) are missing. Fourth, two user pattern features (MCAD domain) are present, though normally only one such feature (MCAD domain) is required. Fifth, this CAD model contains more extruded protrusion shape-features than are normally necessary to properly create it.



(a) (b) (c) (d)

Fig. 13 Example of actual MCAD model with features highlighted. From left to right: (a) two revolved cutout features, (b) first user pattern, (c) second user pattern, (d) two hole features.

These first general observations provide both pointers to where to drill down in the data sets and obtain more details for making sense of these observations, and also offer an approximate first base for individual formative assessment and its derived formative feedback. In the case of the former, cross-view linking and brushing was used to highlight and investigate relevant particular sections of the MCAD model, and revealed the following.

First, in the case of the two revolved cutout shape-features, as shown in Fig. 13(a), where one extruded cutout shape-feature should have been used instead, a closer look at why two revolved cutouts had been used instead of one revealed an incorrect application of the symmetry option within the modeling command. Here formative feedback must take into account that the student needs to be reminded that, according to both the exercise requirements and the projected learning outcomes which accompany each exercise and are provided to the students, revolved cutout features are considered out-of-scope features (for this exercise) and therefore should not be used. It should further be pointed out that if the extruded profile of the second revolved cutout shape-feature is not correctly related through associativity options to its counterpart in the first revolved cutout shape-feature, a parameter change in the latter can cause a wrong profile generation in the former.

Second, in the case of the three missing rib full-features, analysis of modeling details revealed that the actual geometry and topology of those (see again Fig. 12(a)) had been created with three extruded protrusion shape-features, but without the symmetry option invoked. This modeling situation is also related to the case of the missing mirror features (MCAD domain), where one mirror feature should have been used to model the second lateral rib (real objects domain) in an appropriate and consistent manner. It also explains why this CAD model contains more extruded protrusion shape-features than normally required, because they were also used to individually model each of the three ribs (real objects domain). Here, formative feedback needs to address issues relevant for cases where a specific feature (MCAD domain), such as a rib full-feature, is required. The use of a general feature (MCAD domain), such as an extruded protrusion shape-feature, should have been avoided, because of the potential detrimental impact on both MCAD model alterability and the proper use of MCAD models in other computer-aided engineering applications. For example, the student needs to be reminded of the symmetry option, which, in the case of a rib full-feature, automatically translates any changes in the rib feature thickness parameter into a correct symmetric adjustment of its geometry in respect to its reference plane. However, using an extruded protrusion without symmetry requires the user to simultaneously change in a coherent manner both the location of the reference plane and the extrusion distance, in order to achieve a proper and consistent MCAD model alteration. In the case of computer-aided generation of mechanical engineering drawings, using MCAD models that contain any extruded protrusion shape-features that should have been created with rib full-features leads to deficient results. This is because modern MCAD systems are capable of both recognizing rib full-features and correctly translating their graphical representation in longitudinal cross-sections of a drawing into non-hatched regions, whereas extruded protrusion shape-features always result in hatched regions.

Third, in the case of the two user pattern features (MCAD domain), analysis of details revealed that each was created as a user pattern based on two hole full-features as shown in Fig. 13(b) and Fig. 13(c). In this case formative feedback needs to make the student aware that this modeling strategy has no benefit and should be avoided for the following reasons. In general, pattern features represent a regular repetition of particular shape elements which are spatially positioned either along a curve or within a matrix formation. In the current modeling context, in regard to the first exercise segment, the purpose of using the pattern feature is to learn how to group a set of hole full-features into a single pattern feature to obtain effective and efficient access to the feature profile while positioning individual hole full-features using geometric dimensions and constraints. In this manner MCAD model alteration, which is the subject of the second exercise segment, can be conducted not only faster and more efficaciously, but also in a more robust bearing than in the case of individual hole full-features.

Fourth, after analyzing all the cases outlined above, it needs to be determined what the two hole full-features have been used for, and in which part of the MCAD model they were used. Here, analysis of details revealed that each hole full-feature was used to individually create each of the two holes (real object domain) within the yoke clamp head, as shown in Fig. 13(d).

6.3.2. Annotating and categorizing relevant data and information

To document findings and new knowledge within the current implementation of the visualization environment, data-aware and machine-readable metadata, in the form of textual annotations, can be created through manual actions by the user and these can be associated with the visualization elements. Note that, although, in some cases within MCAD modeling, features may be missing in the MCAD model, the data entities used within the visualization and conveying of this information are not. Therefore, a consistent data-aware annotation can be achieved both for features (MCAD domain) that are present in the MCAD model and for those that are required but found to be missing. As annotations are used within formative assessment of CAD models, which in turn contributes to the formation of formative feedback, the goal and application contexts are more specific than with general data exploration.

***FT*: revolved cutout** ***FC*: out-of-scope**
2 revolved cutout features have been created instead of 1, error in the application of the symmetry option, error in the modeling command execution, error in the selection of feature type

***FT*: extruded protrusion** ***FC*: in-scope**
3 extruded protrusion features have been used to create the ribs instead of using rib features,
no symmetry options invoked, error in the design strategy, error in the selection of feature type

***FT*: hole** ***FC*: in-scope**
2 hole features were used each to create the 2 holes of the yoke clamp head,
only 1 hole feature is required, probable cause deficient modeling sequence as
volume-removing features were added before all volume-adding features were present,
shortcoming in the design strategy

***FT*: user pattern** ***FC*: in-scope**
2 user patterns used for creating a 2 x 2 hole pattern, instead of using 1 (user) pattern feature with
4 hole features, error in the design strategy, new knowledge on subdivision error of user pattern,
newly created subcategory for user pattern error
. . .

Fig. 14 Example of a selection of data-aware textual annotations.

In this regard, to make annotations more effective and efficient, they are structured according to what analysis suggests a student has done wrongly or has falsely omitted to do, and what should have been done instead during model creation. They are also structured according to detected CAD model deficiencies and modeling errors, and possible causes and sources of errors, with, where applicable, an indication of which annotation parts comprise new knowledge and insight, perhaps along with references to previous and newly created error categories and sub-categories. For example, the analysis result in the case of the two user patterns (MCAD domain) as outlined above, where an incorrect sub-division of the user pattern was found, represents new knowledge for assessment, because this kind of error regarding user pattern features was neither known nor encountered in any previous exercise. This new knowledge is captured by the creation of a new error sub-category for user patterns and as a part of the annotation associated with the user pattern feature entity of the visualization. A selection of the annotations for the above example is shown in Fig. 14.

Conclusions

Although the concepts and workings of formative assessment are now widely known, its actual implementation and advancement in educational practice, in particular in the context of MCAD education, still poses a problem difficult to solve. The assessment of a high number of MCAD models in a short time poses a considerable challenge, especially when they need to be assessed in regard to various criteria, sometimes as complex as assessing modeling strategies. Such challenges can no longer be approached without the assistance of various software tools and computer-based support. This represents a task that is far from trivial, as assessment requires not only the detection and identification of deficiencies that in many cases do not violate general normative knowledge about feature-based modeling and geometric modeling, but also knowledge about the modeling goals and how they have been translated into actions. Here an approach that combines the perceptual abilities, creativity, and domain knowledge of the human user with the computational power of current desktop computing has great potential to make a major contribution to solving the problem. In this paper, the approach, structures, and framework developed and used for the design and actual implementation of a visualization environment have been presented and discussed. The approach and prototype system developed are aimed at enabling visual analysis of feature-based characteristics of CAD models within the context of MCAD education. A compiled selection of examples has been given to illustrate the translation and application of central concepts of the visual analytics framework and the visualization environment and how these relate to and interact with exercise-specific learning goals and outcomes and the assessment of actual feature-based CAD models as created by students according to concrete exercise requirements.

Test and evaluation of the prototype implementation of the visualization environment produced valuable theoretical and empirical results. These were compiled and translated into improvements in the reformed MCAD course and pointers for future work, some examples of which are as follows. Improvements in the MCAD course that were considered significant from a pedagogical viewpoint have mostly been related to newly gained insight and knowledge, which can also be considered a form of feedback for the teacher. This kind of feedback has been compiled based on deficiencies and errors that were found in MCAD models during analysis and assessment, but had not been encountered in any previous exercise. For example, newly gained insight on the errors and shortcomings in using the revolved cutout feature and the user pattern feature, as described in detail elsewhere in the paper, led to an enhancement and fine-tuning of the course lecture material on both the conceptual structure of the revolved cutout feature, as well as the user pattern feature, and the correct use of the related feature commands. Also, the part of the course that required the modeling of a pattern of holes as a part of the mechanical fixture design within the exercise reported in this paper has been modified. Those modifications and improvements will take effect in the coming academic year. Valuable pointers for future work with promising potential and synergy were related, among other matters, to the provenance of insight in regard to manual error categorization and annotation, with the latter also discussed in detail in the paper. For example, the dimension and nature of externalizing knowledge and insight should be expanded. These are compiled and documented in the form of annotations and error categories within the visualization environment, both during and after visual analysis and assessment. Firstly, to make the manual process of annotation more efficient, pre-defined elements of documented knowledge and insight derived from successful annotations of similar, previously encountered cases should be made available to the user during the annotation process. Secondly, knowledge and insight documented by means of annotations and error categories should be made available to the user in an efficient and adequate manner to facilitate systematic support for the scaffolding of formative feedback. Here future work will involve the design and implementation of additional data structures and procedures within the visualization environment and the wider architecture of the feature-based CAD model assessment module, including its repository and inventory, so that it can capture, compile, store, and retrieve knowledge and insight that has been manually documented during annotation and error categorization.

Currently, preparations are underway to fully integrate the experimental prototype of the visualization environment with an integrated semi-automatic software tool for CAD model assessment that was developed by the authors and has just been fully deployed to support formative assessment and feedback within the recently reformed MCAD course. As far as the authors are aware, no software tools currently exist within MCAD education that explicitly support feature-based MCAD model assessment in a manner adequate to the problem, nor is there any systematic approach to be found in the literature that employs visual analytics in the educational context with a visualization environment that explicitly supports MCAD model assessment. Hopefully, the field of CAD education, and in particular the MCAD education community, can benefit from the various contributions of the work presented in this paper, which range from a conceptual outline of how to systematically employ visual analytics as a solution within the context of MCAD education to a detailed description of the development and implementation of the visualization environment and the user interaction support.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Anscombe, F. (1973). Graphs in statistical analysis. *American Statistician*, 27(1), 17-21.

[2] Arnheim, R. (2004). Gestalt psychology. In W. E. Craighead, C. B. Nemeroff (eds.), *The Concise Corsini Encyclopedia of Psychology and Behavioral Science*, (pp. 401-402), Hoboken, NJ, USA: John Wiley and Sons.

[3] Barequet, G., Duncan, C. A., Kumar, S. (2006). RSVP: A geometric toolkit for controlled repair of solid models. *IEEE Transactions on Visualization and Computer Graphics*, 4 (2), 162 - 177. http://dx.doi.org/10.1109/2945.694983

[4] Becker, A., Cleveland, S. (1987). Brushing scatterplots. *Technometrics*, 29(2), 127-142. https://doi.org/10.2307/1269768

[5] Bederson, B. B., Shneiderman, B. (eds.) (2003). The Craft of Information Visualization: Readings and Reflections. San Francisco, CA, USA: Morgan-Kaufmann.

[6] Benoît, G. (2016). The 'beautiful' in information: Thoughts about visual literacy and aesthetics. *Journal of Visual Literacy*, 35(1), 60-78. https://doi.org/10.1080/1051144X.2016.1205831

[7] Benoît, G. (2019). *Introduction to Information Visualization: Transforming Data into Meaningful Information*. Lanham, MD, USA: Rowman and Littlefield.

[8] Birch, J. (2001). *Diagnosis of Defective Colour Vision*. Boston, MA, USA: Butterworth-Heinemann.

[9] Black, P., Wiliam, D. (2003). In Praise of educational research: Formative assessment, *British Educational Research Journal*, 29 (5), 623 – 637. https://doi.org/10.1080/0141192032000133721

[10] Borkin, M. A., Vo, A. A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., Pfister, H. (2013). What makes a visualization memorable?. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2306-2315. https://doi.org/10.1109/TVCG.2013.234

[11] Branoff, T. J., Hartman, N. W., Wiebe, E. N. (2003). Constraint-based solid modelling: What do employers want our students to know?, *Engineering Design Graphics Journal*, 67(1), 2003, 6 -11.

[12] Brasseur, L. (2005). Florence nightingale's visual rhetoric in the rose diagrams. *Technical Communication Quarterly*, 14(2), 161-182. https://doi.org/1207/s15427625tcq1402_3

[13] Brehmer, M., Lee, B., Isenberg, P., Choe, E. K. (2019). Visualizing ranges over time on mobile phones: A task-based crowdsourced evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 619-629. https://doi.org/10.1109/TVCG.2018.2865234

[14] Brewer, C. A., Harrower, M. (2009/13). *Online Tool ColorBrewer2*. https://colorbrewer2.org, last accessed on June 19th 2020.

[15] Brooks, J. L. (2015). Traditional and new principles of perceptual grouping. In J. Wagemans (ed.), *Oxford Handbook of Perceptual Organization*, (pp. 57-87), New York, NY, USA, Oxford University Press.

[16] Burch, M., Weiskopf, D. (2014). On the benefits and drawbacks of radial diagrams. In W. Huang (ed.), *Handbook of Human Centric Visualization*, (pp.429-451), New York, NY, USA: Springer.

[17] Card, S. K., Mackinlay, J. D., Shneiderman, B., (eds.) (1999). *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan-Kaufmann.

[18] Chester, I. (2007). Teaching for CAD expertise. *International Journal of Technology and Design Education, 17*(1), 23–35.

[19] Chuang, J., Weiskopf, D., Möller, T. (2009). Energy aware color sets. *Computer Graphics Forum*, 28 (2), 203–211. https://doi.org/10.1111/j.1467-8659.2009.01359.x

[20] Cleveland, W. S. (1994). *The Elements of Graphing Data*. Summit, NJ, USA: Hobart Press.

[21] Cohen, I. B. (1984). Florence nightingale. *Scientific American*, 250(3), 128-137.

[22] Cukovic, S., Devedzic, G., Ghionea, I., Fiorentino, M., Subburaj, K. (2016). Engineering design education for Industry 4.0: Implementation of augmented reality concept in teaching CAD courses. In *Proceedings of the International Conference on Augmented Reality for Technical Entrepreneurs*, April 1-2, Bucharest, Romania, (pp. 11–16).

[23] Diehl, S., Beck, F., Burch, M. (2010). Uncovering strength and weakness of radial visualizations – an empirical approach. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 935-942. https://doi.org/10.1109/tvcg.2010.209

[24] Draper, G. M., Livnat, Y., Riesenfeld, R. F. (2009). A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(5), 759-776. https://doi.org/10.1109/ TVCG.2009.23

[25] Drocourt, R., Borgo, R., Scharrer, K., Murray, T., Bevan, S., Chen, M. (2011). Temporal visualization of boundary-based geo-information using radial projection. *Computer Graphics Forum*, 30(3), 981-990. https://doi.org/10.1111/j.1467-8659.2011.01947.x

[26] Elmqvist, N., Stasko, J., Tsigas, P. (2008). DataMeadow: A visual canvas for analysis of large-scale multivariate data. *Information Visualization*, 7(1), 18-33. https://doi.org/10.1057/palgrave.ivs.9500170

[27] Elmqvist, N., Vande Moere, A., Jetter, H.-C., Cernea, D., Reiterer, H., Jankun-Kelly, T. J. (2011). Fluid interaction for information visualization. *Information Visualization*, 10(4), 327-340. https://doi.org/10.1177/1473871611413180

[28] Elvira, Q., Imants, J., Dankbaar, B., Segers, M. (2017). Designing education for professional expertise development. *Scandinavian Journal of Educational Research*, *61*(2), 187–204. https://doi.org/10.1080/00313831.2015.1119729

[29] Eysenck, M. (2018). Fundamentals in Cognition. Hove, UK: Psychology Press.

[30] Fan, C., Hauser, H. (2019). On KDE-based brushing in scatterplots and how it compares to CNN-based brushing. In *Proceedings of the Machine Learning Methods in Visualization for Big Data*, June 3, Porto, Portugal, (pp. 1-5). https://doi.org/10.2312/mlvis.20191157

[31] Fua, Y.-H., Ward, M. O., Rundensteiner, E. A. (2000). Structure-based brushes A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(2), 150-159. https://doi.org/10.1109/2945.856996

[32] Garikano, X., Garmendia, M., Manso, A. P., Solaberrieta, E. (2018). Strategic knowledge-based approach for CAD modelling learning, *International Journal of Technology and Design Education*, 1-13. https://doi.org/10.1007/s10798-018-9472-1

[33] Gonzáles-Lluch, C., Company, P., Contero, M., Camba, J. D. (2017). A survey on 3D CAD model quality assurance and testing tools. *Computer-Aided Design*, *83*, 64 - 79. https://doi.org/ 10.1016/j.cad.2016.10.003

[34] Gotz, D., Zhou, M. X., Aggarwal, V. (2006). Interactive visual synthesis of analytic knowledge. In *Proceedings of the IEEE Symposium on Visual Analytics and Technology*, October 31 – November 2, Baltimore, MD, USA, (pp. 51-58). https://doi.org/10.1109/VAST.2006.261430

[35] Gotz, D., Zhou, M. X. (2009). Characterizing users: Visual analytic activity for insight provenance. *Information Visualization*, 8(1), 42-55. https://doi.org/10.1057/ivs.2008.31

[36] Gramazio, C. C., Laidlaw, D. H., Schloss, K. B. (2017). Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23 (1), 521–530. https://doi.org/10.1109/TVCG.2016.2598918

[37] Harrower, M., Brewer, C. A. (2003). ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1), 27-37. https://doi.org/10.1179/000870403235002042

[38] Hattie, J., Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81-112. https://doi.org/10.3102/003465430298487

[39] Healey, C. G.: (1996). Choosing effective colours for data visualization. In *Proceedings of the IEEE Visualization*, Oct 27 - Nov 1, San Francisco, CA, USA, (pp. 263–270). https://doi.org/10.1109/VISUAL.1996.568118

[40] Heer, J., Bostock, M., Ogievetsky, V. (2010). A tour through the visualization zoo. *Communications of the ACM*, 53(6), 59-67. https://doi.org/10.1145/1743546.1743567

[41] Heer, J., Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4), 45-54. https://doi.org/10.1145/2133806.2133821

[42] Heer, J., Stone, M. (2012). Color naming models for color selections, image editing and palette design. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, May 5 - 10, Austin, TX, USA, (pp. 1007-1016). https://doi.org/10.1145/2207676.2208547

[43] Hu, G., Pan, Z., Zhang, M., Chen, D., Yang, W., Chen, J. (2014). An interactive method for generating harmonious color schemes. *Color Research and Application*, 39(1), 70–78. https://doi.org/10.1002/col.21762

[44] Itten, J. (1961). *The Art of Color*. New York, NY, USA: Van Nostrand Reinhold.

[45] Irons, A. (2008). *Enhancing learning through formative assessment and feedback*. Abingdon, UK: Routledge.

[46] Jenny, B., Kelso, N. V. (2007). Color Design for the color vision impaired. *Cartographic Perspectives*, 58, 61-67. https://doi.org/10.14714/CP58270

[47] Kabouridis, G., Giannopoulos, G. I., Tsirikas, S. A. (2015). On the development of course interconnections within a mechanical engineering training programme via single CAD/CAM/CAE software, *World Transactions on Engineering and Technology Education*, 13(3), 335-341.

[48] Keim, D., Andrieko, G., Fekete, J.-D., Görg, C. (2008). Visual analytics: Definition, process, and challenges. In A. Kerren, J. T. Stasko, J.-D. Fekete, C. North (eds.), *Information Visualization*, LNCS 4950, (pp. 154-175), Berlin/Heidelberg, Germany: Springer.

[49] Kielman, J., Thomas, J., May, R. (2009). Foundations and frontiers of visual analytics. *Information Visualization*, 8(4), 239-246. https://doi.org/10.1057/ivs.2009.25

[50] Koffka, K. (1935). Principles of Gestalt Psychology. New York, NY, USA: Hardcourt, Brace and Company.

[51] Kolence, K. W., Kiviat, P. J. (1973) Software unit profiles and Kiviat figures. *ACM SIGMETRICS Performance Evaluation Review*, 2(3), 2-12.https://doi.org/10.1145/1041613.1041614

[52] Kosslyn, S. (2006). *Graph Design for Eye and Mind*. New York, NY, USA: Oxford University Press.

[53] Lehmann, D. J., Theisel, H. (2013). Orthographic star coordinates. I*EEE Transactions on Visualization and Computer Graphics*, 19(12), 2615–2624.

[54] Leu, J.-G. (1991). Computing a shape's moments from its boundary. *Pattern Recognition*, 24(10), 949-957. https://doi.org/10.1016/0031-3203(91)90092-J

[55] Lin, S., Fortuna, J., Kulkarni, C., Stone, M., Heer, J. (2013). Selecting semantically-resonant colors for data visualization. *Computer Graphics Forum*, 32(3pt4), 401-410. https://doi.org/10.1111/cgf.12127

[56] Liu, K., Peng, X., McGary, P., Yalvac, B., Ozturk, E., Johnson, M. D., Valverde, L. (2015). Integration of contextual exercises in computer-aided design education, *Computer-Aided Design and Applications*, 12 (S1), 13 - 21. https://doi.org/10.1080/16864360.2015.1077071

[57] Mandorli, F., Otto, H. E. (2013). Negative knowledge and a novel approach to support MCAD education. *Computer-Aided Design and Applications, 10(6),* 1007-1020. https://doi.org/10.3722/cadaps.2013.1007-1020

[58] Mandorli, F., Borgo, S., Wiejak, P. (2020). From form features to semantic features in existing MCAD: An ontological approach. *Advanced Engineering Informatics*, 44, https://doi.org/10.1016/j.aei.2020.101088

[59] McLachlan, P., Munzner, T., Koutsofius, E., North, S. (2008). LiveRAC – Interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 5 - 10, Florence, Italy, (pp. 1483-1492). https://doi.org/10.1145/1357054.1357286

[60] Meier, B. J., Spalter, A. M., Karelitz, D. B. (2004). Interactive color palette tools. *IEEE Computer Graphics and Applications*, 24(3), 64–72. https://doi.org/10.1109/MCG.20041297012

[61] Menary, G. H., Robinson, T. T. (2011). Novel approaches for teaching and assessing CAD. *In Proceedings of the 17th International Conference on Engineering Education*, August 21-26, Belfast, Northern Ireland, UK, (Paper-No.: MO.P.SA 11.228).

[62] Mollon, J. D., Pokorny, J., Knoblauch, K. (2003). *Normal and Defective Colour Vision*. New York, NY, USA: Oxford University Press.

[63] Munsell, A. H. (1969). *A Grammar of Color*. New York, NY, USA: Van Nostrand Reinhold.

[64] Munzner, T. (2014). *Visualization, analysis and design*. Boca Raton, FL, USA: CRC Press.

[65] Mylonas, D., MacDonald, L., Wuerger, S. (2010). Towards an online color naming model. In *Proceedings of Color Imaging Conference*, November 8 - 12, San Antonio, TX, USA, (pp. 140-144).

[66] O'Donovan, P., Agarwala, A., Hertzmann, (2011). Color compatibility from large datasets. *ACM Transactions on Graphics*, 30(4), 63: 1–12. https://doi.org/10.1145/2010324.1964958

[67] Olson, J. M., Brewer, C. A. (1997). An evaluation of color-selections to accommodate map users with color-vision impairments. *Annals of the Association of American Geographers*, 87(1), 103-134. https://doi.org/10.1111/0004-5608.00043

[68] Otto, H. E., Mandorli, F. (2018). A framework for negative knowledge to support hybrid geometric modeling education for product engineering. *Journal of Computational Design and Engineering*, 5, 80-93. https://doi.org/10.1016/j.jcde.2017.11.006

[69] Otto, H. E., Mandorli, F. (2021). Parametric feature-based solid model deficiency identification to support learning outcomes assessment in CAD education. *Computer-Aided Design and Applications, 18(2),* 411-442. https://doi.org/10.14733/cadaps.2021.411-442

[70] Ou, L.-C., Chong, P., Luo, M. R., Minchew, C. (2011). Additivity of colour harmony. *Color Research and Application*, 36(5), 355–372. https://doi.org/10.1002/col.20624

[71] Peng, X., McGary, P., Johnson, M., Yalvac, B., Ozturk, E. (2012). Assessing novice CAD model creation and alteration. *Computer-Aided Design and Applications*, PACE(2), 9-19. https://doi.org/10.3722/cadaps.2012.PACE.9-19

[72] Peura, M., Iivarinen, J. (1997). Efficiency of simple shape descriptors. In C. Arcelli, L. P. Cordella, G. Sanniti di Baja (eds.), *Advances in Visual Form Analysis*, (pp. 443-451), Singapore: World Scientific.

[73] Pinzger, M., Gall, H., Fischer, M., Lanza, M.: (2005). Visualizing Multiple Evolution Metrics. In *Proceedings of the ACM Symposium on Software Visualization*, May 14-15, St. Louis, MO, USA, (pp. 67–75). https://doi.org/10.1145/1056018.1056027

[74] Pirolli, P., Card, S. K. (1995). Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, May 7 - 11, Denver, CO, USA, (pp. 51-58). https://doi.org/10.1145/223904.223911

[75] Porter, M. M., Niksiar, P. (2018). Multidimensional mechanics: Performance mapping of natural biological systems using permutated radar charts. *PLoS One*, 13(9), 1-18. https://doi.org/10.1371/journal.pone.0204309

[76] Pütz, C. (2017). Teaching CAD with a pedagogical system ranging from videos to individual tutoring, *Journal of Industrial Design and Engineering Graphics*, 12 (1), 21–26.

[77] Ragan, E. D., Endert, A., Sanyal, J., Chen, J. (2015). Characterizing Provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 31-40. https://doi.org/10.1109/tvcg.2015.2467551

[78] Roos, B., Hamilton, D. (2010). Formative assessment: A cybernetic viewpoint. *Assessment in Education*, 12 (1), 2010, 7 – 20. https://doi.org/10.1080/0969594042000333887

[79] Rynne, A., Gaughran, W. (2008). Cognitive modeling strategies for optimum design intent in parametric modeling, *Computers in Education Journal*, 18(3), 2008, 55-68.

[80] Schloss, K. B., Palmer, S. E. (2011). Aesthetic response to color combinations: Preferences, harmony, and similarity. *Perception and Psychophysics*, 73(2), 551–571. https://doi.org/10.3758/s13414-010-0027-0

[81] Setlur, V., Stone, M. (2016). A linguistic approach to categorical color assignment for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 698–707. https://doi.org/10.1109/TVCG.2015.2467471

[82] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualization. In *Proceedings of the IEEE Symposium on Visual Languages*, September 3 - 6, Boulder, CO, USA, (pp. 336-343). https://doi.org/10.1109/VL.1996.545307

[83] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N. (2018). *Designing the User Interface: Strategies for effective Human-Computer Interaction*. Harlow, Essex, UK: Pearson.

[84] Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78 (1), 153 - 189.

[85] Sosulski, K. (2019). *Data Visualization made Simple: Insights into becoming Visual*. Abingdon, Oxon, UK: Routledge/Taylor and Francis.

[86] Spence, I. (2005). No humble pie: The origins and usage of a statistical chart. *Journal of Educational and Behavioral Statistics*, 30(4), 353-368. https://doi.org/10.3102/10769986030004353

[87] Spence, R. (2014). *Information Visualization*. Cham, Switzerland: Springer.

[88] Stone, M. C. (2003). *A Field Guide to Digital Color*. Boca Raton, FL, USA: A. K. Peters / CRC Press.

[89] Thomas, J. J., Cook, K. A., (eds.) (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Silver Spring, MD, USA: IEEE Computer Society Press.

[90] Tominski, , Schumann, H. (2008). Enhanced interactive spiral display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interaction*, November 27-28, 2008 Stockholm, Sweden, (pp. 53-56).

[91] Turkay, C., Kaya, E., Balcisoy, S., Hauser, H. (2017). Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 131-140. https://doi.org/10.1109/TVCG.2016.2598470

[92] van Wijk, J. J. (2005). The value of visualization. In *Proceedings of IEEE Visualization*, October 23-28, Minneapolis, MN, USA, (pp. 79-86).

[93] Waldner, M., Diehl, A., Gracanin, D., Spletchna, R., Delrieux, C., Matković, K., (2019). A comparison of radial and linear charts for visualizing daily patterns. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 1033-1042. https://doi.org/10.1109/TVCG.2019.2934784

[94] Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., von der Heydt, R. (2012). A century of Gestalt psychology in visual perception: Perceptual grouping and figure-ground organization. *Psychological Bulletin*, (138)6, 1172-1217. https://doi.org/10.1037/a0029333

[95] Ward, M., Grinstein, G., Keim, D. (2015). *Interactive Data Visualization: Foundations, Techniques, and Applications*. Boca Raton, FL, USA: CDC Press.

[96] Ward, M. (2018). Linking and brushing. In L. Liu, M. T. Özsu, (eds.), *Encyclopedia of Database Systems*, (pp. 2101-2105), New York, NY, USA: Springer.

[97] Ware, C. (2020). *Information visualization: Perception for design*. Cambridge, MA, USA: Elsevier.

[98] Weaver, G. C., Burgess, W. D., Childress, A. L., Slakey, L. (eds.) (2016). *Transforming Institutions: Undergraduate STEM Education for the 21st Century*, West Lafayette, IN, USA: Purdue University Press.

[99] Wertheimer, M. (1938). Laws of organization in perceptual form. In W. D. Ellis (ed.), *A Source Book of Gestalt Psychology*, (pp. 71-88), Gouldsboro, ME, USA: Gestalt Journal Press.

[100] Wertheimer, M. (2012). Investigations on Gestalt principles. In L. Spillmann (ed.), *On Perceived Motion and Figural Organization*, Cambridge, MA: MIT Press.

[101] Wiebe, E. N. (2003). Transfer of learning between 3D modeling Systems, *Engineering Design Graphics Journal*, 67 (3), 2003, 15 - 28.

[102] Wilkinson, L. (2005). *The Grammar of Graphics*. New York, NY, USA: Springer.

[103] Wong, Y. L., Siu, K. W. M. (2018). Assessing computer-aided design skills. In M. Khosrow-Pour (ed.), *Encyclopedia of Information Science and Technology*, Volume 10, (pp. 7382-7391), IGI Global, Hershey, PA, USA.

[104] Xue, D.: Teaching CAD in mechanical and manufacturing engineering programs – An experience at University of Calgary (2005). In *Proceedings of the Canadian Design Engineering Network (CDEN) Conference*, Kaninaskis, Alberta, Canada, (Paper-No. 9).

[105] Yan, C., Yu, B., Hui, Y., Zhou, F. (2017). CAD education of college students with engineering practice and team work, *EURASIA Journal of Mathematics, Science, and Technology Education*, 13(7), 5569-5576. https://doi.org/10.12973/eurasia.2017.01010a

[106] Yang, J., Peng, W., Ward, M. O., Rundensteiner, E. A. (2003). Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization*, October 19 - 21, Seattle, WA, USA, (pp. 105-112). https://doi.org/10.1109/INFVIS.2003.1249015

[107] Yang, J., Han, S. (2006). Repairing CAD model errors based on the design history. *Computer-Aided Design*, 38, 627 - 640. https://doi.org/10.1016/j.cad.2006.02.007

[108] Ye, X., Peng, W., Chen, Z., Cai, Y.-Y. (2004). Today's students, tomorrow's engineers: An Iindustrial perspective on CAD education, *Computer-Aided Design*, 36(14), 1451-1460. https://doi.org/10.1016/j.cad.2003.11.006

[109] Zanabria, G. G., Nonato, L. G., Gomez-Nieto, E. (2016). iStar(i*): An interactive star coordinates approach for high-dimensional data exploration. Computers and Graphics, 60, 107-118. https://doi.org/10.1016/j.cag.2016.08.007

[110] Zhang, D., Lu, G. (2004). Review of shape representation and description techniques. Pattern Recognition, 37(1), 1-19. https://doi.org/10.1016/j.patcog.2003.07.008