A universal method to compare parts from STEP files

Nishant Ojal1 · Brian Giera1 · Kyle T. Devlugt1 · Adam W. Jaycox1 · Alexander Blum1

Received: 19 January 2022 / Accepted: 14 June 2022 / Published online: 16 July 2022 © The Author(s) 2022

Abstract
Model Based Definition (MBD) captures the complete specification of a part in digital form and leverages (at least) the universal “Standard for the Exchange of Product” (STEP) file format. MBD has revolutionized manufacturing due to time and cost savings associated with containing all engineering data within a single digital source. This work presents a novel method to transform digital definitions in any given STEP file into a tensor-like structure that is unique for each part and can be used to regenerate the original STEP file completely. Resulting STEP tensors are amenable to part comparison based on various part specifications in a general and straightforward manner. Here, part similarity is evaluated among sets of parts according to specific geometry, material composition, and design intent. Importantly, specification similarity can be quantified using only the tensors’ structure. As such, this approach is not limited to families of geometric shapes, part types, or fabrication methods; nor does it require any prior knowledge about the parts being compared.

Keywords Model Based Definitions · Similarity assessment · STEP file · Manufacturing process planning · ISO 10303

Introduction
Model Based Definition (MBD) represents a comprehensive information source for the entire manufacturing chain across the enterprise (Alemanni et al., 2011) and is essential for achieving the smart manufacturing paradigm. MBD can specify all requirements for manufacturing part(s). In addition to the geometric information defining the part/assembly, it contains Product and Manufacturing Information (PMI) that includes Geometric Dimensions & Tolerances (GD&T), surface finish, material specifications data, and the like. This unification of data improves overall efficiency and eliminates redundancies, such as conversion to 2D drawings, feeding PMI in manufacturing softwares and using multiple data formats. MBD in the form of ISO 10303 defined STEP file standard (ISO, 2021) is a widely used neutral product data format. STEP files serve as a file-exchange format, which is compatible with most of the computer aided designed (CAD) software packages, open-sourced (such as FreeCAD, OpenSCAD, SALMONE, etc.), or proprietary (Autodesk, SOLIDWORKS, CATIA, Creo, etc.). It provides data exchange compatibility across the entire manufacturing chain, starting from design using CAD, analysis using computer-aided engineering, manufacturing using computer-aided manufacturing and inspection using Coordinate measuring machines. Due to its universality and versatility, STEP files are used for part specification for any type of manufacturing approach, e.g., traditional, additive, and/or advance manufacturing systems, and at all manufacturing scales.

The success of manufacturing process planning depends on how reliably a part can be fabricated and inspected using a series of production steps. Developing these production sequences for a new part or product can be a time intensive and expensive process. The starting place for this development generally involves investigating prior examples with similar geometries, materials, etc. and using what is learned as a guide. Current approaches to part comparison/similarity evaluation use only the geometry of parts. The underly-
ing methods being used for this evaluation include global feature-based, graph-based, hint-based, hybrid, volumetric decomposition, and machine learning-based methods. The Literature Review section below elaborates on each of these methods and the corresponding references. Collectively, these methods suffer from a combination of insufficient degrees of robustness and generalization and may also incur significant computational costs. This works aims to overcome these challenges by leveraging the information present in MBD in STEP format and hence further increasing the production utility of STEP files. A major highlight of this work is how information embedded within the STEP file can be extracted systematically and combined into a form amenable to computationally inexpensive matrix operations. As such, part similarity can be determined from various specification types, e.g., geometric data, material properties, GD&T, and combinations thereof.

This paper takes a novel approach of encoding the information of MBD in STEP format into a tensor-like structure, termed a “STEP tensor.” This information is embedded in STEP files through a nested structure of keywords and user-specified arguments for those keywords. By traversing through this root-branch-leaf hierarchy, this data is populated into STEP tensor. STEP tensor is unique for each part and contains all necessary details to reconstruct the STEP file. Using these characteristics of STEP tensors, the paper demonstrates the value of this novel encoding in context of manufacturing. By applying computationally inexpensive matrix operations to STEP tensors, the similarity evaluation of a new part with prior parts in the database is conducted. For this assessment, comparison of geometry and material of parts are performed in this work. The application of this approach in manufacturing process design is presented in this paper with simple examples. However, this can be applied to complex processes and even to other domains such as part design and inventory management as well.

**Literature review**

Solid models are represented predominantly in three representations, namely constructive solid geometry (CSG), boundary representation (BRep), and spatial subdivision (Homann & Rossignac, 1996). Most CAD systems use BRep representation for representing solid models. In BRep representation, an object is defined by the hierarchal connection of topological entities, such as vertex, edge, face, solid, etc. These topological entities contain information of geometrical entities, such as point, curve and surface etc. (Perzylo et al., 2015). STEP file is a neutral CAD format, which uses BRep representation. ISO 10303-42:2019 (ISO, 2019) defines the geometric and topological representation in a STEP file. Using this hierarchical and geometric information, researchers have proposed various approaches to identify the similarity of features and even compare three-dimensional shapes.

The applications of feature recognition or 3D shape comparison span multiple disciplines such as computer vision (Büker et al., 2001), artifact comparison (Koutsoudis et al., 2012) and molecular biology (Firdaus-Raih et al., 2014). In the field of manufacturing, feature recognition has immense potential applications such as, extraction of manufacturing features such as holes (for machining tool path generation), extraction of tolerance specifications (for machine selection required for manufacturing) and extraction of similar part from a database for manufacturing process design. This prior organizational knowledge reuse is a key step to reduce a new product development time and associated cost.

Current approaches for similarity comparison transform the 3D CAD object to a shape descriptor metric or “signature” of the object, and then distinguish the objects by a dissimilarity measure which computes the distance between a pair of descriptors (Tangelder & Veltkamp, 2008). Comprehensive reviews of the different approaches for 3D shape comparison have been presented (Shah et al., 2001; Iyer et al., 2005; Babic et al., 2008). The shape searching methods are broadly classified into following categories: Global feature-based methods, graph-based methods, hint-based methods, hybrid methods, volumetric decomposition methods and machine learning based methods. Global feature-based methods use global properties of the 3D model such as moments (Elad et al., 2001), Fourier descriptors (Kazhdan et al., 2003) and geometry parameters such as surface to volume ratio. These methods capture the averaged global description of shapes. However, they fail to discriminate among locally dissimilar shapes. Graph-based methods represent a part by relational data structure such as attributed adjacency graph (AAG). This data structure contains a detail in multiple levels and facilitates comparison of local geometry (Joshi & Chang, 1988). However, the pattern searching for a real-world part is computationally expensive. Also, complex forms where two or more features interact cannot be detected using these methods. Hint-based method (Vandenbrande & Requicha, 1994) attempts to identify the interacting feature by defining “a presence rule which asserts that a feature and its associated machining operation should leave a trace in the part boundary even when features intersect”. The presence of rules sacrifices the generality of the method, as they are feature specific (Shah et al., 2001). Hybrid methods are a combination of graph and hint-based methods (Gao & Shah, 1998). However, these methods still rely on predefined rules and hence cannot be generalized.

Volume decomposition methods are also capable of identifying interacting features. In these methods, the overall removable volume is identified by subtracting the part model from the stock model. This volume is decomposed into unit
volumes, which are merged into macro volumes based on common faces. The feature identification then reduces into classifying these macro volumes (Han et al., 2000). These methods are computationally intensive. This is because of the effect of local geometry getting cascaded globally across the part, which increases the cell decomposition complexity (Shah et al., 2001).

Machine learning based methods for feature recognition enable dynamic approach of knowledge acquisition, leading to performance improvement. Babyć et al. (2011) presents a review of Artificial neural network (ANN) techniques and other works use convolutional neural networks (Zhang et al., 2018) and (Shi et al., 2020) for feature recognition. All these approaches convert CAD models into a format suited for input to the machine learning techniques, based on methods such as, graph-based methods, contour-syntactic methods and heat persistence map etc. However, these machine learning techniques have some drawbacks while being used for feature recognition, such as challenges associated with identification of complex features and substantial resource required to train the models. Some recent studies, such as (Shi et al., 2020, 2022) are focussing on the identification of intersecting features and reducing the training resources to overcome these challenges.

In manufacturing arena, researchers have mainly focused on converting low level entities present in 3D models such as points, curves etc. to high levels features such as holes and pockets. The purpose of these studies is to use the extracted features to design the manufacturing process for the part. Bhandarkar and Nagi (2000) presented a feature extraction approach using STEP AP203. Using a rule-based methodology, this approach extracts the geometrical data within the STEP file and then translate this data into manufacturing features such as holes, pockets in STEP AP224 format. STEP Manufacturing team demonstrated milling of a part using STEP AP203 as input (Hardwick et al., 2013). They used FB Mach system (Feature-Based Machining Husk) for feature extraction to convert AP 203 to AP 224, and generated tool path required for milling. Bickel et al. (2021) have presented a machine learning based method to extract most similar part to a given input part from a database using similarity search. However, in addition to the complexity associated with this method, the comparison uses geometrical entities only. CAD data in form of a neutral format such as STEP files contain other information such as material property, geometric dimensions and tolerancing and user defined notes related to mechanical or thermal requirement of the part. This work introduces an approach to channel this enormous unexplored information within the CAD data by converting it to a machine-readable format termed the STEP tensor. A holistic comparison of parts (geometrical and material property data) is performed using this STEP tensor.

### Method

STEP files embed all specifications within an ASCII readable format. The encoding mechanism of STEP file is defined in ISO 10303-21 (ISO, 2016) with a schema written in EXPRESS language, defined in ISO 10303-11 (ISO, 2004). Specifically STEP 242 (“Managed Model based 3D engineering”) (ISO, 2020) has been used in this work. This ISO Application Protocol (AP) converges and replaces the earlier used AP 203 and AP 214 standards, with the addition of extended functionalities such as, 3D GD&T at assembly level and 3D shape quality. It captures both requirement data and metadata for a given part or assembly of parts. Requirement data, such as geometric, material, or other specifications, are mapped by a nested structure of keywords and user-specified arguments for those keywords that captures everything. Root keywords exist at the top of nested hierarchy of keyword(s). Underneath this root keyword are other keywords, termed as branch or leaf keywords, as well as arguments for those keywords. Leaf keywords have no branch associated with them but may still have arguments. For instance, the entire geometric specifications for an arbitrary part stem from the “MECHANICALDESIGN_GEOMETRICPRESENTATIONREPRESENTATION” keyword. These keywords are defined in ISO 10303-11. Along this root-branch-leaf hierarchy, the information related to the faces and their orientation, edge lengths, vertex points, etc. is captured by keyword(s) and arguments within a STEP file. This relationship can be represented as a tree-like structure, termed as a STEP tree. This STEP tree and the associated arguments are used to compare the parts in this work.

### STEP file structure

A STEP file may contain up to five sections (ISO, 2016) which, listed chronologically, are the Header, Anchor, Reference, Data, and Signature sections. The Header and Data sections are mandatory sections and the others are optional. Figure 1 depicts the content and purpose of each section in a STEP file. The Header section has the information regarding the context of the STEP file such as timestamp, file schema used, author’s name and organization, etc. The Anchor section contains the definition of external names for instances so that they can be referenced by other files. The Reference section defines instance names whose definitions are present in external files. The Signature section is placed at the end of a STEP file and indicates that the content is verified by that signee. The Data section contains the core information of part/assembly. The topological and geometrical entities are defined in this section and this work demonstrates how the Data section is analyzed in order to compare STEP files in a systematic and universal manner.
The manufacturing process for a given new part, discussed in details in Sect. 4.

A CAD model is a digital representation of a part that contains information required for defining a part’s geometry. It can be exported as a STEP file. The Data section of a STEP file contains the topological and geometrical information of the part. This information is organized into many lines called instances. Each instance has a unique numerical name such as “#123 = ”, followed by keyword(s) and associated argument(s) of each keyword. The arguments contain the keyword definition and typically contains references to many additional instances. The total number of these instances (present in arguments) scales with the complexity of part. The instance to the left of “=” sign is referred as a root and the instances to the right either as branches or leaves.

This hierarchy can be explained readily using an example of STEP file of a hemispherical shell with a hole punch shown in Fig. 2a. Since the entire STEP file contains 378 instances, only a small subset of instance lines is shown in Fig. 2b for clarity. For this object, the geometry is specified by the following instance

\[
\text{#141 = \textit{CLOSED\_SHELL}} \\
('\text{Hemisphere}', (#137, #138, #139, #140));
\]

with the \textit{CLOSED\_SHELL} keyword whose arguments are in parenthesis, e.g. ‘Hemisphere’, (#137, #138, #139, #140). In Equation 1, the instance \#141 is a root and the subordinate instances \#137, \#138, \#139, and \#140 are its branches. Continuing along this tree in this example, the instance \#196 calls keyword \textit{DIRECTION} with arguments ‘\textit{ref\_axis}’ and (1., 0., 0.) and, since no subsequent instance is called, \#196 is one leaf of many in this tree structure. The entire hierarchy that stems from instance root \#141 is mapped into a network graph in Fig. 2c. Each node in the graph labeled with its a unique instance number. The arrows point to subordinate branches or leaves, which can be invoked by multiple times by other branches within the tree. Furthermore, many unique root-branch-leaf structures exist within a given STEP file, stemming from different keyword instances. As such, these network diagrams can be very information dense and thus admittedly difficult to represent in a figure. However, though specific instance names referring to the same feature can vary between STEP files, the hierarchy and sets of nested keywords remain the same. Thus, instance labels within the plots are less important than the shape and structure of the network graphs.

Despite the inherent complexity of network diagrams, they still can be interpreted for similarity, at least on a qualitative basis. Figure 2d demonstrates this with trees derived from
Generating a database of STEP trees using STEP files for different parts. STEP trees for different objects, e.g. hemispherical object, cone, cylinder, or any object, are distinct. Part comparisons can be made using the structures of STEP trees to reveal the most similar part to a given new part.
two other example parts, namely a cone and a cylinder. Each of the “CLOSED SHELL” trees are generated via the same methodology as described above. Since the geometric specifications contained within the respective STEP files for these parts are different, the resulting trees are distinct. By evaluating these graph visualizations, the structure of this instance offers a means of comparing similarity between various parts. To gain quantitative insights of part similarity along with other information within the Data section, STEP tensors can be constructed from the network graphs. A STEP tensor is analogous to an adjacency matrix that maps the connectivity of keywords, while also storing information about arguments for the keywords. STEP tensors of the parts are all unique trees in a given Data. A Python based code is developed for generating STEP tensor and identifying the most similar part from a database.

The procedure for generating the STEP tensor data structure consists of two tasks. In the first task, the information within each instance in the Data section of the STEP file is extracted using parsing techniques. The keywords and their associated arguments are extracted. The root-branch-leaf hierarchy within the STEP file is also captured. Figure 3 walks through the process wherein the hierarchy of instances within Data section is parsed and (possibly nested) keywords and arguments are captured by the STEP tensor. For clarity, Fig. 3a and b show a subset of a Data section and resultings STEP tensors. In our code, these keywords are stored in a row vector and the position or index number of the keyword in the vector is taken as the keyword index number. Therefore, all STEP tensors are arrays of size \(2997 \times 2997\) and each entry, whose position is defined by \([\text{row index}, \text{column index}]\), is a list. The data in the row corresponds to the keyword index number of root and the data in the columns correspond to the keyword index number of associated branches or leaves. The diagonal elements of this array contain the count of occurrences of the roots, a list of instance names, followed by lists of associated arguments. Therefore, the off-diagonal elements contain the count of connections of roots with branches or leaves.

The second task, shown in Fig. 3c, consists of forming the STEP tensor, where the extracted parsed data of the previous step is stored. The STEP tensor stores a count of keyword connections of root with branches or leaves. According to the ISO standard, there exists an exhaustive set of 2997 different keywords within the STEP schema. In our code, these keywords are stored in a row vector and the position or index number of the keyword in the vector is taken as the keyword index number. Therefore, all STEP tensors are arrays of size \(2997 \times 2997\) and each entry, whose position is defined by \([\text{row index}, \text{column index}]\), is a list. The data in the row corresponds to the keyword index number of root and the data in the columns correspond to the keyword index number of associated branches or leaves. The diagonal elements of this array contain the count of occurrences of the roots, a list of instance names, followed by lists of associated arguments. Therefore, the off-diagonal elements contain the count of connections of roots with branches or leaves.

Figure 3c walks through an example of how this formation of the STEP tensor is performed on the instance given by Eq. (1). In this work, keyword indexes are assigned in the order they appear in the ISO standard. However, the actual ordering is arbitrary in the same way elementary matrix operations allow row switching. That said, analysis, comparison, etc. among a set of STEP tensors requires a consistent assignment schema. The index number for the keyword index “CLOSED SHELL” is 659 and for “ADVANCED_FACE” is 428. Here, the root keyword index 659 is connected to four branches of the keyword index 428. Hence, in the row keyword index 659 of the STEP tensor, the number 4 is stored in column keyword index 428 and root’s data “[1, [141]]” is stored in the column index 659. For this example keyword, the last entry is a blank list, i.e. there is no floating number associated with this keyword. Another instance of this STEP file in Fig. 3c,

\[
\#135 = \text{SPHERICAL\_SURFACE}('Sph', \#161, 15);
\]

invokes SPHERICAL\_SURFACE keyword, with index 2062. According to the aforementioned steps, the corresponding data stored in the STEP tensor is “[135], ['Sph'],[15]”. Similarly, ADVANCED\_FACE keyword (with index 428) appears four times in the Data section, thus STEP tensor entry [428, 428] is 4, followed by a list of instance names and the corresponding arguments.

To build the entire STEP tensor, the STEP file is systematically combed with the initial root as “MECHANICAL\_DESIGN\_GEOMETRIC\_PRESENTATION\_REPRESENTATION” keyword and then navigating through its branches and leaves present in its argument and so on. During this process, the information present in each root is extracted and populated into the STEP tensor. This process gradually builds up the STEP tensor and is complete when there are no branches left and the information present in all leaves are extracted.

**Generating forest matrix**

STEP tensor contains all information necessary to reconstruct the Data section of a STEP file in form of keywords and arguments, and root-branch-leaf tree structural relations between them. The Data section comprises of many “trees,” each tree starting with an instance and its connected downstream network of instances. To assimilate the richness of information present in the STEP file and hence in STEP tensor, it is useful to work with condensed data structures derived from all STEP tensors. A lower dimensional \(2997 \times 2997\) “forest matrix” can be created using the first element of each list of the STEP tensor, i.e. the number of times each keyword is used in a given STEP files. A forest matrix offers rapid qualitative insights, as it readily can be represented as an image. Figure 3d shows the forest matrix corresponding to the STEP tensor in Fig. 3c. Each (row, column) entry is color-coded by the frequency
Fig. 3 Data structure of STEP tensor: The structure of a STEP file (a) can be mapped into a STEP tree (b). (c) A STEP tensor contains a frequency count of the topological connections between different keywords and geometrical information of the part. (d) A STEP matrix, which comprises only frequency counts, is represented visually for qualitative analysis.

### (a) STEP file of a hemispherical shell with a hole.

```
#141=CLOSED_SHELL('Hemisphere', #137, #138, #139, #140);
...
#137=ADVANCED_FACE('Top', #66, #64, #135, F);
#138=ADVANCED_FACE('Bot', #67, #65, #63, T);
...
#66=FACE_OUTER_BOUND('Outer', #70, T);
#64=FACE_BOUND('Inner', #71, T);
#135=Spherical_SURFACE('Sph', #161, 15);
...
```

### (b) A subset of STEP tree of a STEP file.

| Keyword index | ADVANCED_FACE | CLOSED_SHELL |
|---------------|---------------|--------------|
| Column        |               |              |
| Row           | 0             | 428          |
|               | 4             | 659          |
|               | 1063          | 1064         |
|               | 2062          | 2996         |

Step tensor\([659, 659]\) = \([1, 141], [\{'Hemisphere\}', []]\)
Step tensor\([428, 428]\) = \([4, 137, 138, 139, 140], [\{'Top\}', []], \ldots\]
Step tensor\([2062, 2062]\) = \([1, 135], [\{'Sph\}', [15]]\]

* Numbers in STEP tensor represent count of Keyword connectivity.
* For clarity, 0 entries are left blank in STEP tensor.

### (c) An example of STEP tensor derived from STEP tree.

### (d) An example of visual representation of a Forest matrix.

Extracting material property

The build material(s) of a part plays an important role in determining its manufacturing process. Along with geometrical specifications, a STEP file also specifies the type and density of a part’s material. Equation (4) presents the instance lines that store the material name of the part.
This information of the material is extracted by combing with “PROPERTY_DEFINITION_REPRESENTATION” as the root and navigating through its branch and leaf keywords namely, “PROPERTY_DEFINITION”, “REPRESENTATION” and “DESCRIPTIVE_REPRESENTATION_ITEM”. In this example, the extracted material name is ‘Steel’. The value of density can be extracted by traversing through extra keywords namely, “MEASURE_REPRESENTATION_ITEM” and “DERIVED_UNIT_ELEMENT”. This information supplements with an extra dimension to the comparison of parts, besides the geometric comparison described earlier.

# Results and discussions

STEP tensors and hence Forest matrices can be built from STEP files using the procedure described in the Method section. This section demonstrates their utility in evaluating parts similarity according to geometry and material specification through several examples. These examples illustrate simple yet effective approach which can be implemented in manufacturing workflow and support in leveraging prior experiential manufacturing process information.

Forest matrices of two parts can be compared directly to reveal similarity between them, as demonstrated in Fig. 4. To illustrate this, a hemispherical shell and truncated pyramid are shown in Fig. 4a1 and b1, respectively. According to the procedure detailed in Fig. 3, their respective STEP forest matrices are generated in Fig. 4a2, b2, where unused keywords (whose row and column counts sum to zero) are hidden for clarity. Since the full (2997 × 2997) forest matrices have the same dimensions, they can be subtracted from each other, which is represented graphically in Fig. 4c. One key feature of this method is that the computational complexity and memory requirements are independent of the complexity of parts/assembly. For each case, the size of STEP tensor is always 2997 × 2997. The parts/assembly differ based on the value and location of the integer value stored in this tensor. The sparsity of the difference of forest matrices is directly proportional to the similarity between the parts. Furthermore, it is straightforward to derive a quantitative metric of the similarity. In general, two Forest matrices, e.g. \( F_1 \) and \( F_2 \), can be subtracted from each other in an element-by-element fashion to compute the squared difference:

\[
\sum_{\text{row}} \sum_{\text{column}} (F_1[\text{row}, \text{column}] - F_2[\text{row}, \text{column}])^2.
\]

Equation (5) provides a numerical value of the (dis)similarity between objects. For two identical objects, \( d^2(F_1, F_1) = 0 \). For the objects in Fig. 4a1, b1, the squared difference via Eq. (5) is \( d^2 = 14750 \).

When evaluating Eq. (5) among a set of distinct objects, the most similar pair would return the smallest value. This property can be exploited for interrogating a database of STEP tensors (and associated forest matrices) to extract important production trends and design insights. For instance, a pairwise comparison between a new part’s STEP tensor can be performed with all entries within this database to identify the most similar part(s) within the database. This has immense applications in manufacturing domain, especially in terms of organizing the manufacturing process data and in defining a starting point to manufacture the new part.

The identification of “the most similar part” can occur using a variety of specification(s) within each STEP file. Figure 5 illustrates the process of evaluating a database of parts for geometric similarity. While Fig. 5a1 shows this example using a set of five unique objects, the number and type of geometries is arbitrary, i.e. the technique is scalable and general. That is, for each part in the database, a STEP tensor is generated, as shown in Fig. 5a2 using the respective Forest matrices. These tensors (or subcomponents thereof) can be manipulated using standard matrix operations, e.g. Figure 4 and Eq. (5). After adding a new part (a hemisphere with a hole punch) to the database and computing its STEP tensor in Fig. 5b1, b2, the squared difference can be computed in a pairwise fashion with all other STEP tensors. When using only the portion of the STEP tensor that accounts for the geometric requirements, the object that returns the lowest squared difference value, \( d^2_{\text{min}} \), exhibits the most similar geometry to the new part. Figure 5c1, c2 shows intuitively that a hollow hemisphere with \( d^2_{\text{min}} = 7248 \) is the most similar part. This can inform the manufacturing design process in developing the production sequence of the new part by leveraging the prior experiential process knowledge of similar parts.

As just demonstrated, a comparison that uses only the geometric portion of the forest matrix, while powerful, may be too coarse for objects with the same overall geometry with different dimensions. This can have profound consequences in a manufacturing setting since fabricating the same object at different scales, e.g. 1 cm\(^3\) versus 1 m\(^3\), or different materials may require entirely different techniques. For instance, STEP
Fig. 4 Comparisons between two objects (a1, b1) can be made by subtracting their STEP file-derived forest matrices, represented graphically (a2, b2) by removing unused keywords. The resulting difference matrix offers rapid qualitative insights on similarity when viewed graphically, i.e. very similar objects yield a sparse difference matrix. Similarity also can be quantified via the squared sum of every element of the difference matrix per Eq. (5).

Fig. 5 Interrogating only the portion of the STEP file that defines geometric requirements of objects shown in (a1, b1) and pictorial representations of this portion of the Forest matrices (a2, b2), reveals the most similar object pairs (c1) via the most sparse difference matrix (c2) with the minimum squared difference value for the entire database, i.e. $d_{min}^2 = 7248$.

tensors contain information about the material properties of the parts, too. Consider two identical ingots made of steel and copper. Using the keywords described in Sect. 3.4, the ‘Steel’ and ‘Copper’ are extracted as the material name and so are the corresponding density values of 8 g/cm$^3$ and 8.94 g/cm$^3$ that live within the STEP files. This, or any, vital specification information can be used to make part families or select manufacturing processes to fabricate the parts. Thus, a more fine-tuned comparison that incorporates additional information in the STEP tensor is straightforward.

Continuing with the example above that used the topological (tree hierarchy) connection data to compare objects, assume additional hemispherical objects with a hole punch were to be fabricated that each had a different shell thickness. Since these hypothetical objects all have the same parameterized geometry, the difference matrices as computed with
Fig. 6 Comparison of hemispheres of different internal diameter (I.D.) using STEP tensors. The parts (Part 1 to Part 4) are compared to a reference part of I.D. = 20. Square differences clearly show the quantitative departure from the reference part.

| Reference Part | Part 1 | Part 2 | Part 3 | Part 4 |
|----------------|--------|--------|--------|--------|
| Outer diameter (O.D.) | 30     | 30     | 30     | 30     |
| Internal diameter (I.D.) | 20     | 10     | 15     | 22     | 25     |
| Squared difference | 0      | 1101   | 267    | 42     | 262    |

Fig. 7 Investigating the capture of design intent of an example part, shown in (a), using different modeling steps, shown in (b)–(d). The squared difference shows that the design intent is not captured by the STEP file. For this example, Fig. 6 illustrates a refined comparison by quantifying similarity using the information present in the STEP tensor. Again, squared difference is used when comparing the reference part (first column) to other parts arranged by increasing shell thickness. The summary table shows the most similar object to be Part 3, which has a shell thickness most approximate to the reference part and thus smallest non-zero squared difference, $d^2_{\text{min}} = 42$. Also, it is to be noted that the squared difference of the reference part compared to itself equals 0 by definition. This demonstration shows that a STEP-file derived similarity quantification is functional for manufacturing environments where minute part design iterations are frequent. As such, the refined analy-
sis in Fig. 6 can be useful in the rapid prototyping stage or in boutique manufacturing settings that leverage additive manufacturing platforms. Admittedly, Figs. 5 and 6 are proof-of-principle demonstrations using simple geometries, but the overall workflow of (1) performing a part comparison over an entire production data base to motivate a (2) more refined geometry-specific comparison is widely applicable due to its adaptability. For complex geometrical entities, e.g. defined by splines, further work needs to be done, which may leverage the complementary techniques detailed in Sect. 2.

Finally, the STEP tensor is used to investigate the modeling steps used to design a part. In keeping with the example part in Figs. 5, 6, and 7 compares four different design methods to create the same hemispherical shell with a hole punch. Figure 7b and c shows a wireframe that is revolved about its central axis that does and does not require a hole removed from the top, respectively. Figure 7d conducts a shell operation on a full sphere that is split half by a central plane before cutting a hole. Figure 7e splits a hollow sphere in half before cutting a hole from the top. Four forest matrices are generated from these STEP files and compared to the reference object. With the exception of Fig. 7c, this analysis reveals all methods are geometrically equivalent. In Fig. 7c, two additional vertex points are generated and their specification within the resulting STEP tensor is detected via non-zero $d^2$ value. From this, it can be concluded that the STEP file captures only the connections between the geometrical entities and not the steps followed to model the part. Since the modeling steps may inform the motivation behind a given product, such as GD&T and surface finish or non-financial interest in the subject matter or materials discussed in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Conclusions/future works

Model Based Definition (MBD) captures the complete specification of a part in digital form and leverages (at least) the universal “SStandard for the Exchange of Product” (STEP) file format. MBD has revolutionized manufacturing due to time and cost savings associated with containing all engineering data within a single digital source. This work presents a novel method of comparing parts using STEP 242, a neutral file format. Following are the contributions of this work.

1. A novel method of encoding the information present in the STEP 242 into called STEP tensors, is introduced. These tensors contain all the information present in the STEP files and are unique to a given part.
2. STEP tensors (or subcomponents thereof) can be manipulated using standard matrix operations. STEP tensors offer computationally cheap intercomparison among any set of parts in a general manner.
3. Intercomparison of parts can be used to leverage the prior knowledge of manufacturing process of similar parts. Additionally, the development of tools and analysis methods like those presented in this work can further leverage this data to improve the design and manufacturing processes. Increased focus on data collection, processing and utilization is a trend in the manufacturing community. This paper focuses primarily on geometry as a similarity metric, but STEP files contain a wealth of information about a given product.

Future work in this space could explore new applications of the STEP tensor (beyond part intercomparison), exploring new ways to integrate this tool into the design and manufacturing workflow. The STEP tensor could also be extended to include more PMI, such as GD&T and surface finish data. The natural extensibility of this method to include any information present in STEP file is a natural strength of the approach.

Acknowledgements This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-832725.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

Aleman, M., Destefanis, F., & Vezzetti, E. (2011). Model-based definition in the product lifecycle management scenario. The International Journal of Advanced Manufacturing Technology, 52(1), 1–14.

Babic, B., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. Computers in Industry, 59(4), 321–337.

Babić, B. R., Nešić, N., & Miljković, Z. (2011). Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25(3), 289–304.

Bhandarkar, M. P., & Nagi, R. (2000). Step-based feature extraction from step geometry for agile manufacturing. Computers in Industry, 41(1), 3–24.

Bickel, S., Cauer, C., Schleich, B., & Wartzack, S. (2021). Comparing cad part models for geometrical similarity: A concept using machine learning algorithms. Procedia, CIRP, 133–138.

Büker, U., Drüe, S., Götz, N., Hartmann, G., Kalkreuter, B., Stemmer, R., & Trapp, R. (2001). Vision-based control of an autonomous disassembly station. Robotics and Autonomous Systems, 35(3–4), 179–189.

Elad, M., Tal, A., & Ar, S. (2001). Content based retrieval of VRML objects—an iterative and interactive approach. In J.A. Jorge, N.M. Correia, H. Jones, & M.B. Kamegai (Eds.), Eurographics multimedia workshop. The Eurographics Association. https://doi.org/10.2312/EGMM/egmm01/107-118.

Firdaus-Raih, M., Hamdani, H. Y., Nadzirin, N., Ramlan, E. I., Willett, P., & Artyumiuk, P. J. (2014). Cognac: A web server for searching and annotating hydrogen-bonded base interactions in RNA three-dimensional structures. Nucleic Acids Research, 42(W1), W382–W388.

Gao, S., & Shah, J. J. (1998). Automatic recognition of interacting machining features based on minimal condition subgraph. Computer-Aided Design, 30(9), 727–739.

Han, J., Pratt, M., & Regli, W. C. (2000). Manufacturing feature recognition from solid models: A status report. IEEE Transactions on Robotics and Automation, 16(6), 782–796.

Hardwick, M., Zhao, Y. F., Proctor, F. M., Nashehi, A., Xu, X., & Venkatesh, S. (2013). A roadmap for step-NC-enabled interoperable manufacturing. The International Journal of Advanced Manufacturing Technology, 68(5–8), 1023–1037.

Hoffmann, C. M., & Rossignac, J. R. (1996). A road map to solid modeling. IEEE Transactions on Visualization and Computer Graphics, 2(1), 3–10.

ISO. (2004). Industrial automation systems and integration—Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual (Standard No. ISO 10303-11:2004). International Organization for Standardization. Retrieved from https://www.iso.org/standard/38047.html.

ISO. (2005). Industrial automation systems and integration—Product data representation and exchange—Part 55: Integrated generic resource: Procedural and hybrid representation (Standard No. ISO 10303-55:2005). International Organization for Standardization. Retrieved from https://www.iso.org/standard/38226.html.

ISO. (2016). Industrial automation systems and integration—Product data representation and exchange—Part 21: Implementation methods: Clear text encoding of the exchange structure Standard No. ISO 10303-21:2016. International Organization for Standardization. Retrieved from https://www.iso.org/standard/63141.html.

ISO. (2019). Industrial automation systems and integration—Product data representation and exchange—Part 42: Integrated generic resource: Geometric and topological representation (Standard No. ISO 10303-42:2019). International Organization for Standardization. Retrieved from https://www.iso.org/standard/78579.html.

ISO. (2020). Industrial automation systems and integration—Product data representation and exchange—Part 242: Application protocol: Managed model-based 3D engineering (Standard No. ISO 10303-242:2020). International Organization for Standardization. Retrieved from https://www.iso.org/standard/66654.html.

ISO. (2021). Industrial automation systems and integration—Product data representation and exchange – Part 1: Overview and fundamental principles (Standard No. ISO 10303-1:2021). International Organization for Standardization. Retrieved from https://www.iso.org/standard/72237.html.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., & Ramani, K. (2005). Three-dimensional shape searching: State-of-the-art review and future trends. Computer-Aided Design, 37(5), 509–530.

Joshi, S., & Chang, T.-C. (1998). Graph-based heuristics for recognition of machined features from a 3d solid model. Computer-Aided Design, 20(2), 58–66.

Kazhdan, M., Funkhouser, T., & Rusinkiewicz, S. (2003). Rotation invariant spherical harmonic representation of 3d shape descriptors. Symposium on Geometry Processing, 6, 156–164.

Koutsoudis, A., Makarona, C., & Pavlidis, G. (2012). Content-based navigation within virtual museums. Journal of Advanced Computer Science and Technology, 1(2), 73–81.

Perzylo, A., Somani, N., Rickert, M., & Knoll, A. (2015). An ontol- ogy for cad data and geometric constraints as a link between product models and semantic robot task descriptions. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Sys- tems (IROS) (pp. 4197–4203).

Shah, J. J., Anderson, D., Kim, Y. S., & Joshi, S. (2001). A discourse on geometric feature recognition from cad models. Journal of Computing & Information Science in Engineering, 1(1), 41–51.

Shi, P., Qi, Q., Qin, Y., Scott, P. J., & Jiang, X. (2020). Intersecting machining feature localization and recognition via single shot multibox detector. IEEE Transactions on Industrial Informatics, 17(5), 3292–3302.

Shi, P., Qi, Q., Qin, Y., Scott, P. J., & Jiang, X. (2022). Highly interacting machining feature recognition via small sample learning. Robotics and Computer-Integrated Manufacturing, 73, 102600.

Shi, Y., Zhang, Y., & Harik, R. (2020). Manufacturing feature recognition with a 2d convolutional neural network. CIRP Journal of Manufacturing Science and Technology, 30, 36–57.

Tangelder, J. W., & Velthamp, R. C. (2008). A survey of content based 3d shape retrieval methods. Multimedia Tools and Applications, 39(3), 441–471.

Vandenbrande, J. H., & Requicha, A. A. (1994). Geometric computation for the recognition of spatially interacting machining features. In Manufacturing Research and Technology (Vol. 20, pp. 83–106). Elsevier.

Zhang, Z., Jaisswal, P., & Rai, R. (2018). FeatureNet: Machining feature recognition based on 3d convolution neural network. Computer-Aided Design, 101, 12–22.

Publisher’s Note Springer Nature remains neutral with regard to jurisdic- tional claims in published maps and institutional affiliations.