Teodor Vernica
Systems Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899;
Department of Computer Science,
Aarhus University,
8200 Aarhus N, Denmark
e-mail: teodor.vernica@mpe.au.dk

Robert Lipman
Systems Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899
e-mail: robert.lipman@nist.gov

Thomas Kramer
Intelligent Systems Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899;
Department of Mechanical Engineering,
Catholic University of America,
Washington, DC 20064
e-mail: kramert@cua.edu

Soonjo Kwon[1]
Department of Mechanical System Engineering,
Kumoh National Institute of Technology,
Gumi, Gyeongbuk 39177, South Korea
e-mail: soonjo.kwon@kumoh.ac.kr

William Z. Bernstein[1]
Systems Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899
e-mail: william.bernstein@afresearchlab.com

Visualizing Standardized Model-Based Design and Inspection Data in Augmented Reality

Augmented reality (AR) has already helped manufacturers realize value across a variety of domains, including assistance in maintenance, process monitoring, and product assembly. However, coordinating traditional engineering data representations into AR systems without loss of context and information remains a challenge. A major barrier is the lack of interoperability between manufacturing-specific data models and AR-capable data representations. In response, we present a pipeline for porting standards-based design and inspection data into an AR scene. As a result, product manufacturing information with three-dimensional (3D) model data and corresponding inspection results are successfully overlaid onto a physical part. We demonstrate our pipeline by interacting with annotated parts while continuously tracking their pose and orientation. We then validate the pipeline by testing against six fully toleranced design models, accompanied by idealized inspection results. Our work (1) provides insight on how to address fundamental issues related to interoperability between domain-specific models and AR systems and (2) establishes an open software pipeline from which others can implement and further develop.
[DOI: 10.1115/1.4053154]

Keywords: industrial augmented reality, standards, interoperability, quality information framework, inspection, product manufacturing information, model-based definition, digital thread, QIF, PMI, MBD

1 Introduction

Augmented reality (AR) has become a valuable technology for manufacturing-based applications, including assistance in maintenance, process monitoring, and product assembly [1]. However, significant barriers exist to wider adoption of industrial AR including high development costs and a fundamental lack of interoperability [2]. Aimed at achieving data interoperability for smart manufacturing systems (SMS), the "digital thread" is conceptually useful for coordinating, aligning, and registering disparate data models across the product lifecycle including (but not limited to) product design, process planning, manufacturing execution, and part inspection [3].

Coordinating traditional engineering data representations into AR systems without loss of context and information remains a challenge. A major issue is the harmonization of standards within and across AR and SMS representations. For example, in previous work [4], we examined the integration issues between IndoorGML, a graph-based standard representation for modeling indoor spaces, with MTConnect, a standard for semantic interoperability of manufacturing assets. Although we were able to successfully generate a meaningful AR scene, we found semantic inconsistencies at the data-field level that can only be addressed by the standards development organizations (SDOs) themselves.

In industrial AR, most efforts have been put into several focused areas such as assembly [5,6], maintenance [7,8], product development [9], and manufacturing layout [10] as described in Ong et al. [11]. While previous studies generated application-specific AR scenes based on non-standard data for a specific task, few studies have focused on merging AR capabilities into the model-based integrated design and inspection frameworks [12]. Such AR-enabled integrated frameworks can bring substantial benefits to many enterprises by providing a means to connect the specific task-based efforts together through the digital thread.

In this work, we investigate the feasibility of porting standards-compliant product definitions, including product manufacturing information (PMI) annotations and inspection information, into AR environments. Figure 1 provides an example of a model-based product definition.[2] The figure presents an isometric view of a computer-aided design (CAD) model with various annotations. The annotations adhere to the American Society of Mechanical Engineers (ASME) Y14.5 standard [13], an authoritative guideline for the design language of geometric dimensioning and tolerancing (GD&T). Many manufacturers treat such representations as living documents for reference throughout the product lifecycle. Hence, it is critical to reference the original definition for additional uses.

To translate semantic PMI information into AR, we developed a pipeline that automatically leverages the standard product model as a reference. We demonstrate our approach by importing a three-part

[1]Corresponding authors.

Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received September 2, 2021; final manuscript received November 22, 2021; published online February 7, 2022. Assoc. Editor: Chih-Hsing Chu.

[2]https://go.usa.gov/mGVm

assembly with spatially anchored annotations into an AR presentation system. We then validate our approach by (1) directly interacting with the assembly components through a tablet and (2) testing the data mapping procedures on six other fully toleranced open-source design models with software-generated, nominal inspection data. We then enumerate challenges faced in the integration and use of the models in AR and present potential AR-assisted inspection-based use cases based on informal conversations with practitioners. These use cases can directly benefit from leveraging the presented pipeline.

Our primary use case is overlaying standards-based inspection data onto design information, including GD&T and inspection results information, in an AR-compliant environment. Specifically, this work is a first step in coordinating our previous study [3] that mapped standards-compliant inspection data to design information through knowledge graphs. Integrating such perspectives into an AR-capable environment is essential for realizing highly scalable industrial AR.

2 Background and Related Work

Industrial AR, or use of AR technologies in industrial practice, presents a number of domain-specific challenges. Many of these challenges relate to the compatibility of data representations across design/manufacturing-based use cases and AR presentation systems. For example, native design models are often represented as boundary-defined, three-dimensional (3D) models. To properly visualize such 3D models in AR, model simplification [14] is required, often in the form of mesh-based representations. Translation leads to an inherent loss of information and fidelity, such as data associated with fully defined geometric features. One of the casualties of this process is GD&T information, a critical component for on-demand part inspection.

Below, we review the relevant data representations that facilitate translation of PMI into AR, related work for presenting part inspection data within virtual environments, and shortcomings of existing approaches.

2.1 Relevant Data Representations and Standards. The STandard for the Exchange of Product model data (ISO 10303 STEP) is a neutral representation of product data used for the entire product lifecycle [15]. STEP files facilitate interoperability between different CAD software packages and are used to represent PMI and other information vital for the smart manufacturing digital thread. STEP is maintained by the International Organization for Standardization[3] (ISO) and is actively developed to meet the requirements of the engineering community. STEP AP242 [16], referred to as Managed Model Based 3D Engineering, covers the scopes of AP203 and AP214 and contains new capabilities for computer-interpretation of part and assembly information, including surface finish, manufacturing process information, and tolerances [17,18].

The annotations in Fig. 1 are linked to a CAD model's features (e.g., edges, holes, and faces) to provide a formal definition of product geometry and specifications. PMI annotations include GD&T information and non-geometric data, e.g., surface texture specifications, surface finish requirements, process notes, material specifications, and welding symbols. Since GD&T is a symbolic language meant to communicate information about manufactured parts, standardization is vital for the presentation of annotations to be properly governed. ASME Y14.5–2009 [13] and ISO 1101:2012 [19] are the industry standards for the syntax and semantics of GD&T.

Developed by the Digital Metrology Standards Consortium (DMSC), the Quality Information Framework (QIF) [20] supports model-based inspection (MBI) by providing data models and dictionaries in eXtensible Markup Language (XML) format to formally

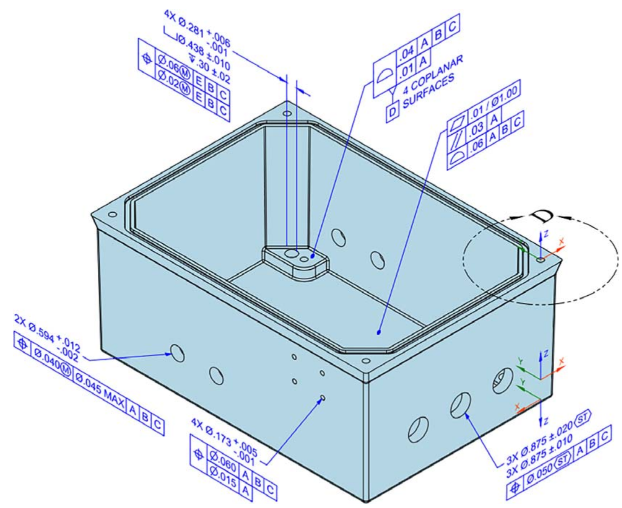

Fig. 1 Example of a CAD model with PMI annotations

define inspection plans, rules, and results. As STEP does, QIF provides data models to represent semantic PMI. However, QIF models features and characteristics in addition to tolerances. Recently, some commercial CAD vendors are starting to support the QIF format, signaling that QIF is spreading quickly throughout the design and manufacturing sectors.

STEP and QIF overlap in their model-based definitions, including part geometry, product structure, and GD&T. Trainer et al. [21] presented a mapping specification between the two standards to support Model-Based Enterprise (MBE) practices. Based on the specification, Kwon et al. [3] linked semantic PMI information both in STEP and QIF by using ontology and knowledge graphs to enrich a standards-based digital thread. Similarly, we link and visualize design and inspection data in AR scenes.

Although these standards exist, limited work addresses the porting of PMI into virtual environments, let alone AR, while still adhering to standard practices and guidelines.

2.2 Relating CAD Software to Augmented Reality Engines. While AR is proving to have numerous industrial use cases, developing applications requires significant time investment [1]. One fundamental barrier to quickening development time is the lack of interoperability between existing engineering data (e.g., CAD models) and the software used to develop AR applications.

AR applications are primarily developed in game engines. Game engines are traditionally used for video game development, but their powerful toolsets lend themselves to general software development including AR applications. However, game engines provide limited compatibility with engineering data, given that engineering work is curated in specialized software, with little overlap. By default, such game engines do not traditionally support any of the widely used boundary representation (B-rep) CAD formats, such as STEP. This incompatibility can hinder the use of existing models in AR environments and thus hinder experimentation and use of these new technologies.

With the increase in adoption of industrial AR solutions [22], commercial efforts have emerged to bridge the gap between engineering data and game engines. These solutions can vary from model translation tools and importers, to stand-alone AR visualization platforms. Some preparation tools claim to enable conversion and optimization of 3D CAD data to more lightweight tessellated representations, better suited for visualization purposes. These tools present themselves as plugins to specific game engines, providing more direct integration and support for additional features, such as the ability to import point cloud data.

While commercial products provide solutions for development and have previously been successfully leveraged by researchers

[3]ISO TC184/SC4 is the subcommittee responsible for STEP.

[23,24], open solutions can provide additional benefits. Standards-based open-source solutions are desirable in many settings because of the transparency with which the data is handled. To the best of our knowledge, there are currently no established open workflows or pipelines that can achieve automated integration between 3D CAD data and such game engines.

There are platform-specific commercial industrial Internet of things (IIoT) solutions that offer AR experiences on the market. For example, some recently released tools emphasize the standards-based connectivity among different IIoT devices which enables visualization of data from multiple sources in AR. However, the standards that these platforms focus on is at the protocol-level such as Hypertext Transfer Protocol (HTTP) and Simple Object Access Protocol (SOAP) not at the industrial data-level. It is thus challenging to find use cases that visualize different standard representations intertwined with each other in the same AR scene.

2.3 Use of Product Manufacturing Information Within Augmented Reality. A systematic method for augmentation is required to provide different manufacturing information in different tasks such as design specifications, setup, and fixture plans, and inspection information [11]. Fang et al. [25] highlighted the need for adoption of AR with a model-based definition (MBD) PMI model in the closed-loop dimensional quality management system.

AR-assisted inspection of mechanical and electrical parts has been a focus area among AR users. Polvi et al. [26] showed that an AR interface with 3D registered annotations resulted in faster times and fewer errors compared to non-registered annotations on static images for computer hardware inspection. Similarly, Runji and Lin [27] proposed a marker-less AR-aided printed circuit board (PCB) inspection, showing that a head-mounted display (HMD) device is more suitable than a handheld device for such a task. A recent study combined an AR-assisted inspection with a deep learning-based pin detection in aviation connector inspection [28].

While limited, previous work has used AR as a tool for visualizing manufacturing data. Urbas et al. [23] propose a method for part inspection that aids users in the measurement process by contextually visualizing PMI information in AR. They make use of the PiXYZ plugin to import the CAD data into Unity, including graphical PMI annotations. However, the implementation of the AR application itself is not realized. Moreover, Urbas et al. acknowledge that their method can only import graphical PMI and can not make use of semantic PMI. In contrast, we showcase a means to preserve semantic PMI in addition to the graphical representation by leveraging standards-based open-source tools. Section 4.1 showcases our AR application, partially validating our approach.

Fiorentino et al. [29] present the tangible digital master (TaDiMa) system. TaDiMa leverages markers embedded in technical drawings as tangible user interfaces to display Product Lifecycle Management (PLM) data, queried from a PLM database. Additionally, Fiorentino et al. [30] showcase two methods for reducing annotation clutter and readability. Finally, they propose a number of potential use case scenarios for their system. They explore additional use cases of a similar AR system in another paper. In our work, we focus on the data integration and the automated extraction and visualization of the data defined in the models. To deal with annotation clutter, we employ a two-dimensional (2D) user interface consisting of a list of toggles for each view defined in the STEP file.

Urbas et al. [31] developed an AR-aided inspection method that overlays deviations on the physical gear that is being inspected. They also enriched the inspection process with the automatic evaluation of the transferred PMI from the design data in STEP AP242. However, they did not consider the transfer of semantic PMI to AR nor did they use a standard format for inspection data. Additionally, their proposed pipeline for aligning and visualizing inspection data is not automated and requires significant human intervention throughout the process, making it unscalable. In contrast, we use the full capability of standard representations of semantic PMI defined in STEP AP242 (design) and QIF (inspection) to aid designers and inspectors in AR in an automated manner.

2.4 Takeaways From Existing Work. Realizing scalable and maintainable industrial AR experiences requires significant research and development. Such opportunities lie not only in fundamental research opportunities, such as understanding and improving asset tracking capabilities and enhancing worker engagement through more comfortable visualization modalities. However, to facilitate decision-making on the floor, it is still necessary to merge state-of-the-art AR technologies into existing engineering workflows.

The main challenge we aim to tackle is the lack of interoperability between platforms for developing AR experiences and upstream engineering data from CAD/computer-aided manufacturing (CAM) software. This triggers translations of engineering data into AR scenes, which results in a loss of semantic information generated during the MBD workflow. An AR-enabled integrated framework supporting the full MBD/MBE workflow could be a solution to this by allowing users to promptly respond to possible engineering change requests [32].

In our approach, we rely on standards to overcome data interoperability challenges between engineering data and AR engines. Standards, specifically standard data representations, provide mechanisms for data interchange between disparate computer-aided design, engineering, and manufacturing software. Next, we describe our technical approach, focusing on specific design decisions for producing a proof-of-concept.

3 Technical Approach

For design information, we assume that the STEP data model stores all as-designed geometry and other critical specifications, including GD&T annotations. Currently, AR engines and presentation systems cannot read STEP natively. Hence, it is necessary to translate the model into an AR-ready representation without losing context.

Model tessellation describes the process of translating a B-rep model into a triangular mesh, while model decimation refers to the process of reducing the number of polygons from an existing mesh [33]. Model tessellation and decimation is common for visualizing 3D models on lightweight devices, such as head-mounted displays. The magnitude of model decimation depends on a number of factors, including the computational power of the presentation device, e.g., tablet or head-mounted display.

For model tessellation, we leverage the NIST STP2X3D Translator,[4] which inputs a STEP Part 21 file [34] (or a STEP instance file) and outputs an X3D file. The exact part geometry is converted to faceted geometry for X3D, an ISO standard [35]. The NIST STEP File Analyzer and Viewer (SFA) [36] uses the STP2X3D translator for enhanced visualization of geometry in a web browser and separately translates graphical PMI, annotations, and views from a STEP file. For the purpose of the presented pipeline, shapes in X3D are represented by coordinates that are connected to create lines and faceted surfaces using `<IndexedLineSet>` and `<IndexedFaceSet>`, respectively. We provide additional explanation in Sec. 4. The X3D format offers significant opportunities in its ability to support lightweight visualization and readiness to link domain-specific information to geometric features [37].

X3D offers several components relevant for representing engineering design data. In addition, the advantages of XML format also hold in X3D. However, X3D is not as popular or widely accepted as other competing formats including its ancestor, Virtual Reality Markup Language (VRML) [38]. We are certain that any standard formats that can store and represent such design data will be able to serve the same purpose. COLLADA[5] and

[4]https://www.nist.gov/services-resources/software/step-x3d-translator
[5]https://www.khronos.org/collada/

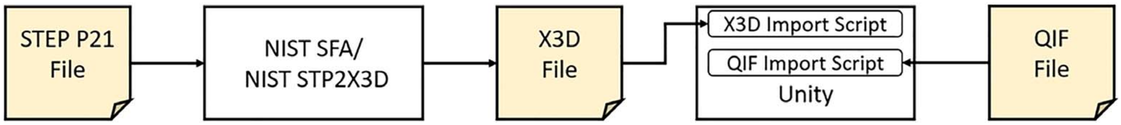

Fig. 2 Our pipeline to import a product definition including its PMI from STEP and QIF files into a Unity scene. The process of importing X3D files is further illustrated in Fig. 3, while the QIF import process is expanded in Fig. 6.

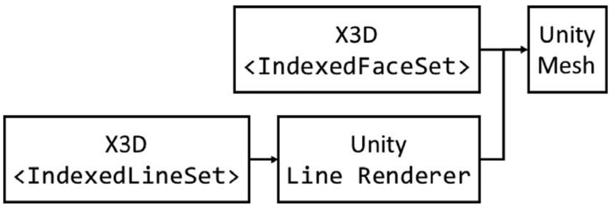

Fig. 3 Mapping of elements from X3D to Unity

glTF[6] are good candidates. Using separate formats for part geometry and PMI can be a viable option; however, divorcing the two seems to have less backing in the standards and industrial practice communities.

Our approach is fully standards-based and open-source.[7] As a reference throughout the pipeline, we leverage STEP to represent part geometry and graphical PMI annotations. Exact part geometry is represented by free-form surfaces and geometric primitives such as planes and cylinders. Examples of graphical PMI annotations for dimensions and geometric tolerances are shown in Fig. 1. Each annotation has a leader line that attaches it to the associated surfaces of the part. In the STEP file, graphical PMI is represented by lines and faceted surfaces.

To match semantic PMI in STEP and QIF files in AR, we match *name* values of GD&T items defined in both files assuming their names do not change over time. This is reasonable because it is likely that either (1) both files were generated in the same CAD system or (2) one file was translated into the other. However, errors could always be introduced during file translation or human intervention which could lead to mismatches. Using persistent identifiers, e.g., universally unique identifiers (UUIDs), instead of names would be an ideal alternative to name matching if each GD&T item was assigned a persistent identifier and used throughout the lifecycle. QIF developers recommend users to assign and use persistent identifiers to elements. In the STEP community, a recommended practice on this matter is currently under development.

4 Implementation Details

After a STEP file is converted to a mesh representation using the STP2X3D Translator, the X3D file must be imported into Unity. While Unity lacks native support for the X3D format, files can be easily parsed using generic XML parsers, such as the one built in Microsoft's.NET platform on which Unity runs. Therefore, an additional import script is required. Figure 2 conveys our implemented pipeline. In short, a product definition, defined as a STEP P21 File [16], is fed into the NIST STEP File Analyzer [36] recently updated with a STP to X3D converter.[3] Then, the resulting annotated X3D file along with its corresponding QIF instance file [20] is imported into Unity through scripts we developed.

The X3D import script (1) is attached to a game object in the scene hierarchy, (2) takes the X3D file generated by SFA as input, and (3) generates the respective geometries using Unity functions and components according to Fig. 3. For example, the vertices encoded within the `<IndexedLineSet>` X3D element are drawn using Unity's `Line Renderer` component. To facilitate the manipulation of each drawn line, such as its scaling, translation, and rotation, we convert each line into a mesh. `<IndexedFaceSet>` elements can directly generate a Unity mesh by using the encoded vertices together with the indexes encoded with the `coordIndex` attribute as the mesh triangles.

Listing 1 Example IndexFaceSet element

```
<IndexedFaceSet solid=′ false′
  coordIndex=′ 16 17 18 −1 … 16 18 19 −1′ >
  <Coordinate DEF=′ coord100′
  point=′ 4. 4.1 41.925 … 6.476 4.1 42.30′/>
</IndexedFaceSet>
```

Listing 1 provides an example `<IndexedFaceSet>` element. The element contains a `<Coordinate>` element as a child, which encodes coordinates through the `point` attribute. The coordinates can be used to generate a mesh in Unity with additional processing, as the mesh requires vertices as an array of type Vector3. The resulting array consisting of multiple Vector3 instances can then be passed to a new Unity mesh instance through its `vertices` property. To draw faces between vertices, indexes encoded in the `coordIndex` attribute of the `<IndexedFaceSet>` element are required. Indexes corresponding to each Vector3 value are separated by the –1 value and can be passed to a Unity mesh through its *triangles* property as an array of integers (excluding the –1 values).

Since the number of indexes of `<IndexedLineSet>` elements is not necessarily a multiple of 3 (e.g., `<IndexedLineSet coordIndex='0 1 -1'>` has only two, since –1 is excluded), it can not directly generate a mesh, as the indexes do not always define triangles expected by the mesh. Instead, the indexes and point coordinates can be passed to a Line Renderer Unity component through the `SetPosition` method. The resulting line can be converted into a mesh using the `BakeMesh` method.

Figure 4 shows an example scene hierarchy generated by the import script in Unity. In this example, the import script is attached to an empty game object named `Box Assembly - Plate`. Graphical PMI annotations are grouped in the X3D file by views, with each view containing different aspects of the part definition, such as tolerances and datum definitions. To preserve this structure, each view encoded in the X3D file generates a view game object in the scene hierarchy (`swView0`, `swView1`, etc.), as a child of the original object. The part geometry (`swPart0`) is encoded separately from the views and is also generated as a child of the original object, e.g., the `Box Assembly - Plate` game object in this case. Meshes generated from `<IndexedFaceSet>` and `<IndexedLineSet>` elements representing annotations are then generated as children of each corresponding view or part geometry in the hierarchy (e.g., `TAO 181 | datum | A`). Preserving this structure allows views, part geometry, and annotations to be manipulated individually within the game engine. Figure 5 illustrates the drawing of the part geometry with and without annotations. In this example, only the wire-frame of the geometry (`swPart0`) is superimposed over the physical piece, providing additional context.

With the latest developments[8] to STP2X3D and SFA, additional information can be preserved in the translation process from STEP

[6]https://www.khronos.org/gltf/
[7]https://github.com/usnistgov/AR-PMI

[8]This will be included in the upcoming releases of STP2X3D and SFA.

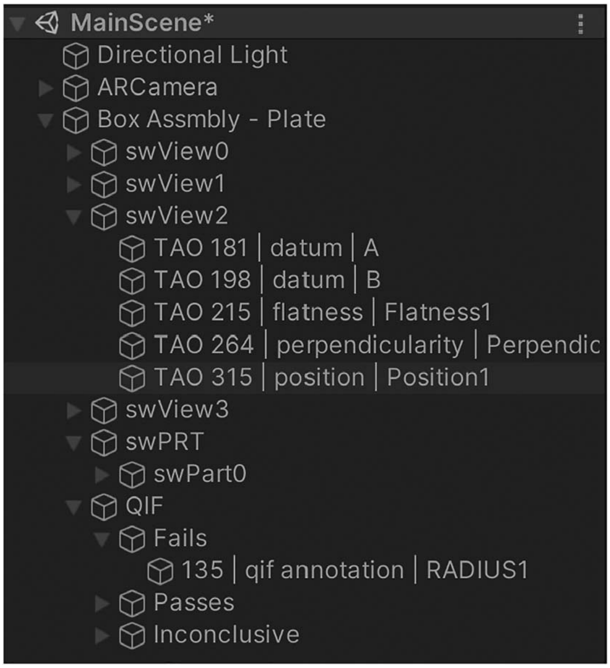

Fig. 4 Example of generated Unity hierarchy

to X3D. Particularly, features and surfaces defined in the STEP models can be delimited and related to corresponding annotations. This is done according to Fig. 6, where surfaces can be cross-referenced with annotations in a view, via the preserved feature names.

The import script is not a fully fledged X3D importer, as only relevant elements (e.g., `<Transform>`, `<Shape>`, `<Coordinate>`, etc.) are being processed, rather than the entire standard specification. That being said, additional support can be easily added, e.g., `<Viewpoint>` X3D elements can be mapped to a Unity camera object, and `<DirectionalLight>` elements can generate a Unity object characterizing directional light.

By preserving their names and ids in the translation process from STEP to X3D, individual annotations can be automatically correlated with characteristics measured in corresponding QIF files. To do this, we developed an additional Unity script that takes a QIF 3.0 file as input and, when attached to the same game object as the X3D import script, generates a new set of annotations based on the inspection results according to Fig. 6. The script parses the QIF file and reads all characteristic measurements which contain a `CharacteristicItemId`. The id can be used to read the corresponding characteristic item encoded within the *Characteristics* section of the QIF file. Characteristic items contain a `Name` element, which can be correlated to X3D annotations and surfaces using the same naming conventions.

As mentioned previously, we are relying on similar naming conventions for matching characteristics between STEP and the associated QIF files. However, we are not assuming these to be identical between the two formats, but rather that they can be matched through some pattern. For example, while a characteristic might be named *Linear Size.1* in STEP (and therefore in X3D), the corresponding QIF characteristic can be named *LINEAR_SIZE.1*. In a different example, the X3D annotation can be named *Feature Control Frame (24)* while the corresponding QIF characteristic is named *FEATURE_CONTROL_FRAME__24_*. In each case, a new function is required to match one pattern to the other. This, however, does not typically need to be done for each individual part. A set of models developed using the same CAD system will typically maintain the same pattern.

4.1 Augmented Reality Application. By leveraging the previously described pipeline (Fig. 2), we developed an AR application[9] which uses model-based tracking to register PMI annotations extracted from the as-designed CAD model, to their physical counterparts, as shown in Figs. 7 and 8.[10]

To test our approach, we imported and tracked three STEP files[11] that make up the NIST *Design, Manufacturing, and Inspection Data for a Box Assembly* dataset.[12] The publicly available dataset consists of three CAD Models and associated data collected during the fabrication process of the parts, including inspection data in the QIF format. The application was deployed to an Android tablet, running on the Android 10 operating system.

To spatially register and visualize the PMI annotations in AR, we leveraged PTC's Vuforia[13] framework. The recent addition of model-based tracking allows physical 3D objects to be used as tangible user interfaces [39]. This means that users can more naturally interact with digital information, such as CAD models, through a physical 3D representation of the model itself rather than through a 2D screen, with a mouse and keyboard as an input modality. Note that the AR framework itself is not tied to the importer, and thus other tracking solutions could be used based on need.

The annotations are separated into views, which are generally used in CAD modeling to contextually group together a set of relevant elements from the model. Views can be generally fit to purpose and can contain additional information such as the viewpoint. Here, each view is represented by a different color, as encoded in the X3D file by SFA. Each view and the part geometry can be toggled on and off, using the user interface shown in Fig. 9.[14] The toggles are automatically generated based on the currently tracked model. Figure 9(*a*) showcases two views simultaneously overlaid onto the box assembly. View 2 (magenta) represents datum and top hole definitions and View 3 (blue) represents the boundary and side hole definition. Figure 9(*b*) shows two additional views toggled on the same part: View 0 (green) displays notes and titles and View 4 (cyan) shows the bottom hole definition.

Figure 10 shows the results of leveraging the surface-annotation cross reference functionality described earlier. In this case, tapping an annotation in the AR view highlights the corresponding part surface that it references, while hiding all others for clarity. Similarly, tapping a surface on the part will highlight the corresponding annotations.

Figure 11 shows the QIF annotations resulting from the process of automatically relating inspection results to the X3D representation. The annotations are color-coded and separated in three layers that can be toggled on and off similar to the X3D views and part geometry. Annotations corresponding to characteristics that have passed every single measurement across multiple part instances are colored in green. Similarly, annotations corresponding to characteristics with failed measurement results are colored in red, and characteristics that have both passed and failed results across multiple part instances are labeled as inconclusive and colored in yellow.

While, we are only utilizing the *Results* QIF information model here, additional insights could be provided by similarly visualizing data from the *Statistics* information model. The scope of the statistics section is to capture summary statistical and derived results in addition to measurement data. The information can be stored as a variety of workpiece studies, e.g., First Article Inspection reports (FAIRs), Production Runs, and Gage Repeatability and Reproducibility studies. In the case of the Box Assembly dataset, additional data is encoded as part of a process capability study. The process capability study is meant to understand deviations of a specific process (i.e., machining) across multiple builds. Note that the open dataset reports on 20 builds for each of the three parts.

[9]https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-video.html
[10]https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-cover.html, https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-plate.html
[11]Note that SolidWorks MBD was used to export STEP AP242 files.
[12]https://github.com/usnistgov/smstestbed/tree/master/tdp/mtc
[13]https://www.ptc.com/en/products/vuforia
[14]https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-box.html

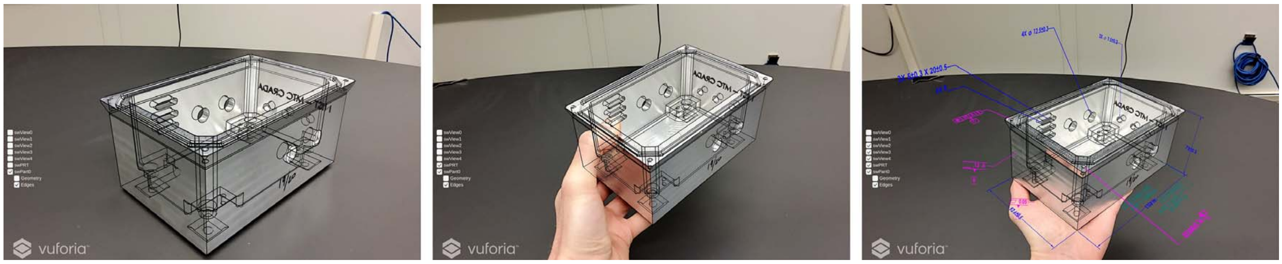

Fig. 5 AR view of the wire-frame of the part geometry overlaid on the physical part

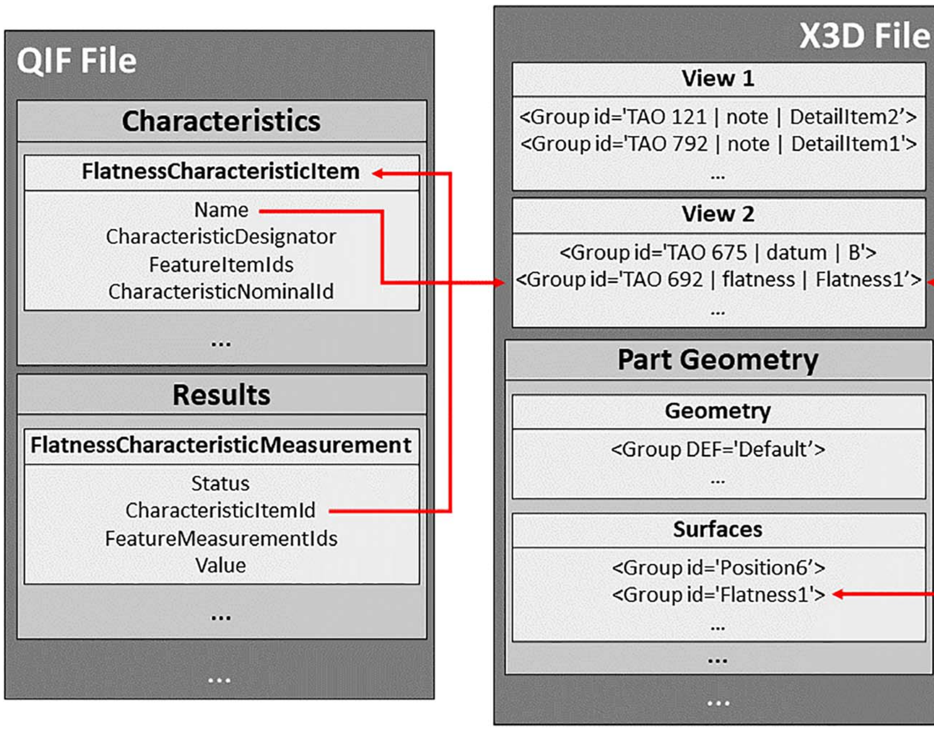

Fig. 6 Process of relating QIF results to X3D annotations and surfaces

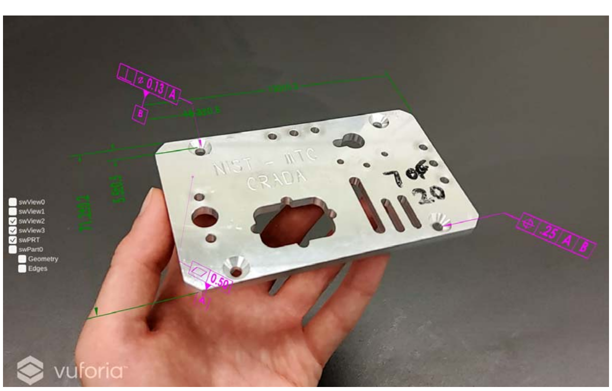

Fig. 7 PMI annotations superimposed on the machined Plate part from the NIST dataset. View 1 (magenta) shows datum definitions. View 2 (green) shows hole definitions.

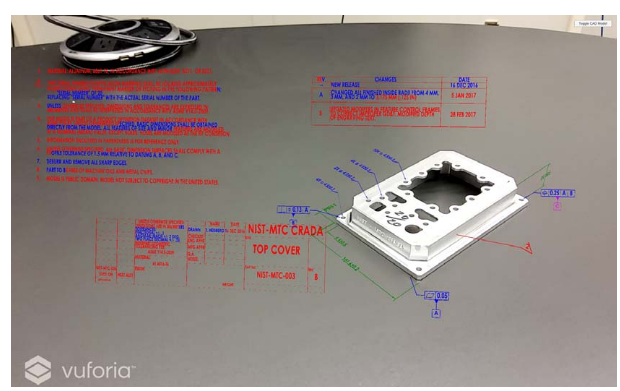

Fig. 8 AR view of the Cover part. View 1 (red) presents notes and titles. View 2 (blue) shows all PMI, View 3 (magenta) shows datum and hole definitions and View 4 (green) displays boundary definitions.

5 Validation

In addition to the Box Assembly dataset, we leveraged the NIST Fully Toleranced Test Cases (FTC) dataset[15] to validate our Unity import pipeline and automated characteristic matching. The test set was specifically developed to test the conformance of CAD systems to ASME standards for dimensions and tolerances [13] and are meant to be leveraged in testing of MBD and MBE workflows. The dataset consists of 11 different CAD test cases: five combined (or complex) test case (CTC) and six FTC. The dataset consists of a

[15]https://www.nist.gov/el/systems-integration-division-73400/mbe-pmi-validation-and-conformance-testing-project

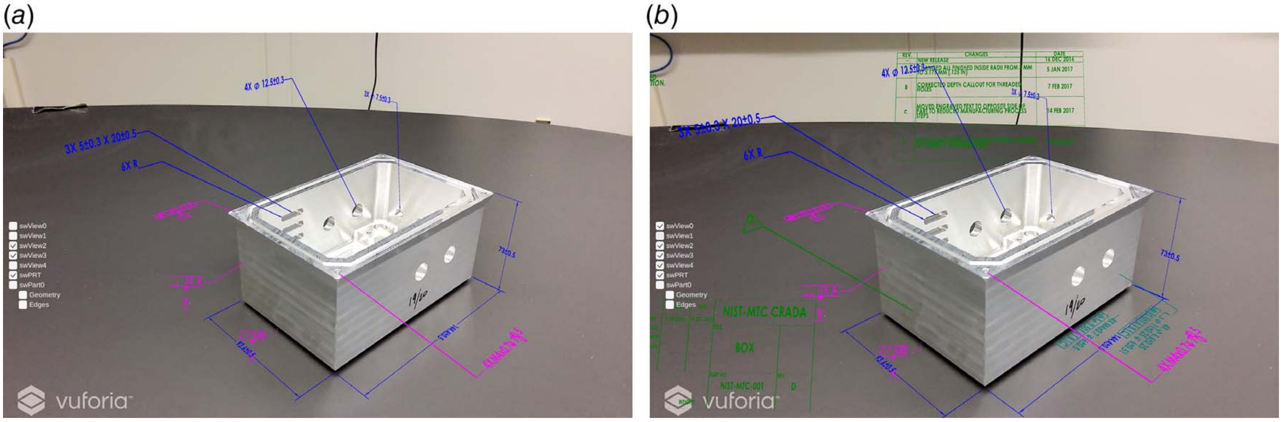

Fig. 9 AR views of the Box from the NIST Box Assembly dataset, with different views toggled

Fig. 10 Two different annotations highlighted with the corresponding surface on the fabricated part

total of 30 files, including variations of these 11 models, e.g., AP242 and AP203 versions of the same part.

The results of translating and importing the STEP models into Unity were visually inspected against the output of SFA, which was used as a ground truth reference, as exemplified in Fig. 12. 29 out of 30 files were imported correctly, with both part geometry and PMI annotations when available. In contrast to others, the 30th model present in the dataset (nist_ftc_08_asme1_ap242-e1-tg.stp) encodes a tessellated geometry rather than a B-rep representation. Given that the STP2X3D translator does not currently support tessellated geometries as an input, the output cannot be generated. In our case, this model is imported in Unity with correct annotations, but with incorrect normals in the part geometry.

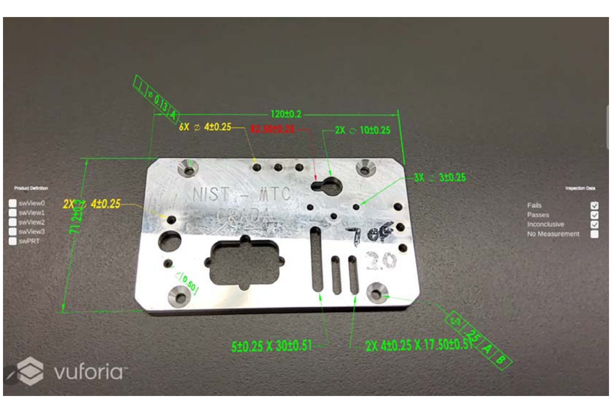

Fig. 11 QIF inspection results overlaid on the Plate part of the Box Assembly in AR. Annotations are color-coded based on characteristic measurement results across 20 part instances: green for passing every measurement, red for failing every measurement, and yellow for both failed and passed results across the multiple part instances.

Given that the CTC/FTC test cases were never physically fabricated, there exists no inspection data associated with them. As a result, we generated nominal inspection data with small Gaussian error based on the STEP models themselves via specialized software.[16] We were able to generate QIF files corresponding to models CTC 01, 03, 05, and FTC 06, 08, 09. We leveraged this data to validate the process of automatically relating QIF inspection results to the respective X3D model imported in Unity. Figure 13 shows the result of relating the corresponding QIF measurements to the FTC 06 test case as an example.

To validate the results, we compared the number of QIF characteristic items matched to an X3D annotation, to the total number of characteristic items enumerated in the "Results" section of the QIF file. If any characteristic items were not matched, they were individually inspected using SFA to find the cause. Table 1 shows the results of the validation process. All QIF characteristic items were successfully matched to a corresponding X3D annotation for the six CTC/FTC models. The Plate, Cover, and Box models had a number of annotations that were not able to be matched to the X3D model. The reasons are discussed in Sec. 6, but generally it

[16]Conducted via toolsets offered from Origin International Inc.

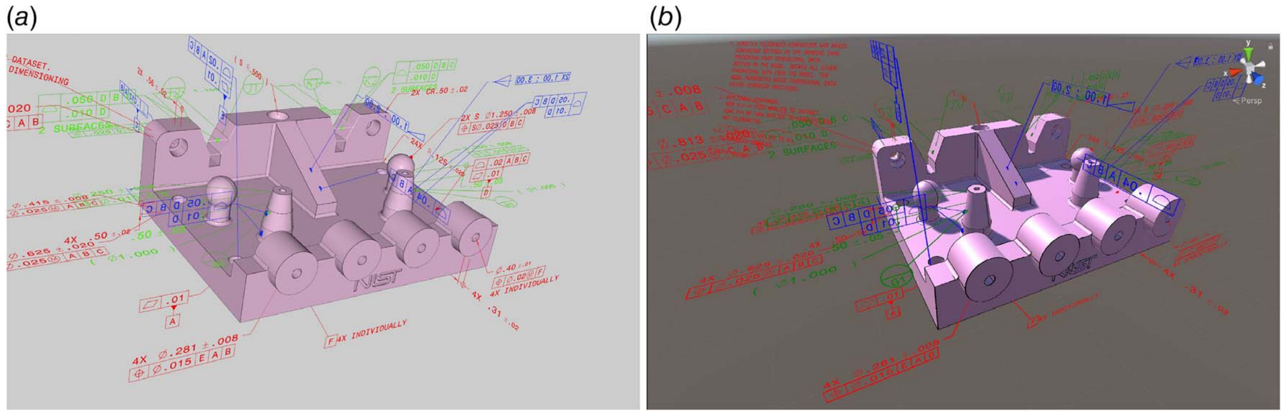

Fig. 12 Side-by-side comparison of the FTC 06 test case as visualized by SFA (*a*) and as imported in Unity (*b*)

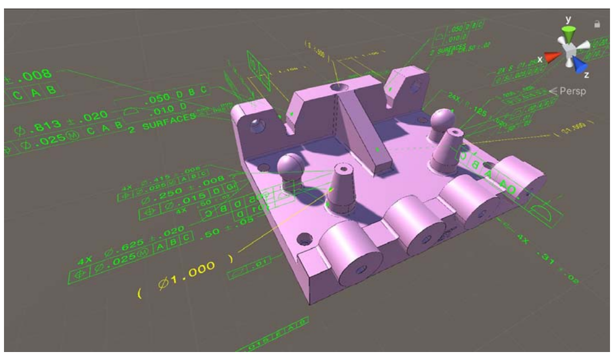

Fig. 13 Idealized nominal inspection data, i.e., QIF results, generated as annotations in Unity for the FTC 06 test case

is because, in some cases, the data violates our assumption that characteristics would have similar namings in both files.

6 Pipeline Issues and Limitations

The presented workflow requires manual intervention when translating files from STEP to X3D using SFA. The process can be more automated by using the command line version of SFA.

Additional human intervention, such as scaling and rotating, might be required once the model is imported inside the game engine. This is due to different units and coordinate system conventions used by software vendors. For example, Unity uses a left-handed y-up coordinate system, while another popular game engine, the Unreal Engine, uses a left-handed z-up convention. Additionally, by default, many 3D modeling programs use right-handed coordinate conventions. This means that a 3D model, created in one program, might be oriented and scaled differently depending on conventions used by the software to which it is imported. While these issues can be solved, it is usually done on a case-by-case basis. As a result, we consider automated scaling and rotating of the part object beyond the scope of this work.

Augmented reality tracking libraries generally provide a visual representation of the tracked model within a game engine. In doing so, digital augmentations can be positioned relative to a visual aid in the application development process. Hence, it is possible that the visual aid would be sensitive to the same challenges mentioned previously. In other words, the visual aid could be oriented or scaled differently than the X3D encoding. Additional work might be required to correctly align the imported X3D geometry and annotations to the provided visual aid within the game engine. To minimize this overhead, our importer attempts to automatically scale the X3D part geometry and annotations if such a visual aid is available. Our algorithm compares the dimensions of the two meshes and computes the scale difference between them. The X3D geometry is then scaled up or down by a computed factor. Note that we do not modify the scale of the visual aid to avoid any impacts on the tracking process.

While such enhancements would help eliminate some potential overhead, additional manual intervention might still be required once the data is brought into Unity during the application creation process. Again, we consider these improvements out of scope for this work, and might be dependent on other factors, such as the tracking libraries used, e.g., Vuforia's model target preparation process.

Because our method for correlating QIF inspection data to STEP PMI annotations relies on matching naming conventions, errors can incur when this assumption fails. Some causes for the QIF mismatches observed in the Box Assembly dataset include the following:

- *Incorrect feature namings in STEP files.* While most STEP annotations correctly preserve feature naming patterns, errors can occur in the export process of the STEP models from CAD systems (e.g., naming a feature *TXD6@Scheme2*). Given an arbitrary and non-descriptive name, these cannot be matched to QIF characteristics.
- *Inconsistencies in QIF Characteristic Item definitions.* In one case, a defined characteristic Item did not contain either a name or a designator. In another case, the characteristic name and characteristic designator were reversed. While these cannot be named errors, as these attributes are optional in QIF, they are inconsistent with the rest of the results and could be the result of human error.
- *General naming inconsistencies.* In some cases, characteristics defined in QIF are simply never mentioned in the corresponding STEP file. In some cases, this can be caused by the fact that files were generated using different software, and there is no guarantee that they will use the same naming conventions.

Table 1 Results of validating the process of matching QIF inspection results to the X3D model

Model	QIF characteristics	Matched characteristics
CTC 01	14	14
CTC 03	31	31
CTC 05	16	16
FTC 06	107	107
FTC 08	58	58
FTC 09	81	81
Plate	31	22
Cover	28	27
Box	47	36

In other cases, missing semantic information in the STEP files caused a mismatch between the two standards.

The issues present in the files could be fixed manually to yield better results. However, we did not alter the existing datasets, which were not specifically made to respect our assumption. Yet, the results confirm that our assumptions are reasonable. It is clear that leveraging persistent UUIDs to help coordinate the harmonization of data elements across industrial standards would be a more robust solution. Unfortunately, this is a difficult task, requiring harmonization and wide adoption from CAD system vendors [40,41].

7 Potential Use Cases of Our Product Manufacturing Information in Augmented Reality Pipeline

We developed three use case directions for visualizing product definitions and inspection results in AR, based on informal conversations with practicing metrologists.

7.1 Communicating Design Requirements Within Teams. AR can be useful for a team to create bubble drawings when planning inspection. Team members can walk through all setup instructions, such as fixture setup and part placement. Visualizing process-feature mappings in AR as tags can assist the team in root cause analysis (RCA). Similarly, showing statistical data attached to features will help cross-disciplinary teams, e.g., across design and manufacturing personnel, to visualize recent trends and anomalies in the data. Such information can assist in RCA. During team reviews, AR scenes can be enhanced with higher resolution color maps by superimposing deviations on a part as well as more information on the specific feature shown. This will be especially beneficial for reviewing FAIRs. Tasks described here are closely related to the results of this study.

7.2 Assisting Metrologists in Pre-Inspection Setup. By correlating annotations and the corresponding part characteristics, we can highlight regions of interest, such as datums, measurands, and reference frames, used by metrologists to position a part for inspection (similar to Fig. 10). Specifically, each part can have its fixtures produced depending upon measurement requirements. AR can be beneficial for designing and setting up fixtures for parts by quickening (and sometimes eliminating) manual tasks based on 2D drawings and pictures. Metrologists will also be benefited from overlaying information from a machinist's handbook in AR for quick look-ups. In addition, AR can help establish a connection to similar/relevant data from similar parts produced by the same or similar processes so that metrologists can visually refer to past data for the current setup.

7.3 Visualizing Inspection Results in Real-Time. Often, metrologists use technical drawings (or blueprints) as physical checklists to track the progress of an inspection. Rather than annotating directly on the paper drawing, our pipeline could support an application to keep track of the completed measurements. Especially for parts with many identical (or similar) features, such an application could significantly increase operator productivity and efficiency. Similar gains have already been realized in industrial practice for AR-assisted assembly [42].

Furthermore, as engineering teams mostly care about inspection fails, it will be useful to color-code failures especially when there are a significant number of similar features, e.g., holes, and to point-and-shoot failure reports overlaid onto the part. There are commercial tools that generate point cloud data scanned in real-time and register the points to the design model. Registration of the design with the physical part will be a simple extension assuming the design model is available. This is expected to be more feasible when using a non-contact 3D scanner. While expected to be technically challenging, inspection results could also be streamed in real time from a coordinate measuring machine and displayed in AR. Combining AR with real-time measurement devices based on computer vision techniques (e.g., measure by camera) could be one obvious future direction, yet would require additional technical advances.

8 Future Directions

Our proof-of-concept uncovers a number of research opportunities for a standards-compliant pipeline to be AR-ready with limited loss of information and context. Here, we present a number of research directions that we intend to investigate further.

- *Coordinating contextual views of annotations*: The STEP data structure affords contextual views of annotations, which could be thought of as layered data presentations based on user needs. The same capability could be leveraged to better handle AR-based presentation. The use of contextual views for AR is not well understood. As a result, there might exist opportunities for appending STEP with additional entities to facilitate better AR presentation. Considering complex product assemblies, additional research questions also remain, including (1) how well the presented pipeline translates to PMI for mated assemblies and (2) whether additional design considerations for interactivity with nested assembly views is warranted.
- *Registering additional digital thread data representations in the same spatial context*: Relating inspection data back to design models in AR is a desired use case. For example, nondestructive inspection (NDI) is leveraged for expensive parts, especially in high-mix, low-volume situations. Spatial registration between NDI data structures, such as those from X-ray computed tomography (XCT) scans for additive-manufactured parts, with native design models remains a challenge. Addressing automated registration of such data structures will facilitate unique AR use cases. Similar insights can be drawn from other manufacturing-related data, including point clouds derived from traditional probe-based measurement instruments and controller-reported data from machine tools.
- *Standardized exchange of scenes and animations in AR*: We used standard formats for design and inspection data, mapping and overlaying both in AR to assist users. Exchanging existing AR scenes, animation objects, and other related data between AR/Virtual Reality (VR) software platforms remains a challenge [11]. The standards community has recently attempted to address these challenges to reduce the fragmentation of AR development. Interested readers should direct their attention to the Khronos Group's OpenXR effort.[17] However, current efforts, such as OpenXR, do not fully address challenges tightly intertwined with industrial data standards. Additional effort is warranted to improve the maintainability and re-usability of industrial animations while preserving critical annotations, such as assembly mating requirements. Note that this issue touches beyond AR/VR development and is also dependent on the management of industrial data, such as properly storing spatial definitions of entities within various bills of materials (BOMs). Continued efforts in developing standardized methods to support interoperability between AR development and industrial AR will help close the gap.
- *Guidelines for industrial data visualization and interaction in AR*: Proper use of visual variables, e.g., size, color, and position of data entities, on two-dimensional interfaces is a well seeded research thrust in the information visualization (InfoVis) community. For example, researchers have provided very descriptive design guidelines for the proper use of visual variables [43]. However, for AR applications, the use of visual variables is not well understood. Similarly, while interaction methods are well understood in traditional 2D mediums, AR

[17] https://www.khronos.org/openxr

interaction methods are an ongoing research topic [44,45]. To address the potential inspection-related use cases described in Sec. 7, well-controlled user studies that provide a deeper understanding of proper data presentation are warranted.

9 Conclusion

We presented an automated approach for linking detailed design data, including PMI, within an AR experience. The lack of automated methods for coordinating PMI into game engines is a primary barrier for its use in AR. This technical gap motivates our work. Our technique is rooted in standards and leverages open tools (when available) throughout the pipeline. To realize the pipeline, two Unity scripts were developed, the X3D and QIF import scripts, in addition to modifications to the existing open-source tools, SFA and STP2X3D. Standards are critical for addressing interoperability challenges when coordinating domain-specific models with AR systems.

We envision this work will better facilitate integration of manufacturing and inspection information contextualized to the native design model. This paper is an extension of our existing work. We focused on improving the work by spatially contextualizing downstream data, i.e., inspection results in QIF, onto design model representations.

In summary, our pipeline simplifies AR scene creation for product model definitions. In the near term, the automated translation of native design models into Unity meshes is immediately useful for creating AR application prototypes more quickly. In the past, the amount of manual effort required to construct such functional prototypes has inhibited wider adoption of industrial AR. Our contributions can be leveraged across multiple use cases, including on-part inspection and assembly guidance.

Acknowledgment

We thank the NIST-internal and journal reviewers for their valuable feedback. We especially thank Bob Stone from Origin International for voluntarily producing the idealized nominal QIF data of the CTC/FTC test cases. Without his help, we would not be able to fully validate our presented pipeline. Through this validation process, we were able to improve the robustness of our work.

Disclaimer

The work presented in this paper is an official contribution of the National Institute of Standards and Technology (NIST) and not subject to copyright in the United States. Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST. Nor does it imply that the products identified are necessarily the best available for the purpose.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data and information that support the findings of this article are freely available.[18] The authors attest that all data for this study are included in the paper.

[18] See Note 7.

References

[1] Egger, J., and Masood, T., 2020, "Augmented Reality in Support of Intelligent Manufacturing—A Systematic Literature Review," Comput. Indus. Eng., **140**, p. 106195.

[2] Pedone, G., and Mezgár, I., 2018, "Model Similarity Evidence and Interoperability Affinity in Cloud-Ready Industry 4.0 Technologies," Comput. Indus., **100**, pp. 278–286.

[3] Kwon, S., Monnier, L. V., Barbau, R., and Bernstein, W. Z., 2020, "Enriching Standards-Based Digital Thread by Fusing As-Designed and As-Inspected Data Using Knowledge Graphs," Adv. Eng. Inform., **46**, p. 101102.

[4] Hanke, A., Vernica, T., and Bernstein, W. Z., 2020, "Linking Performance Data and Geospatial Information of Manufacturing Assets Through Standard Representations," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Online, Aug. 17–19.

[5] Wang, X., Ong, S. K., and Nee, A. Y., 2016, "A Comprehensive Survey of Augmented Reality Assembly Research," Adv. Manufact., **4**(1), pp. 1–22.

[6] Radkowski, R., 2016, "Object Tracking With a Range Camera for Augmented Reality Assembly Assistance," ASME J. Comput. Inf. Sci. Eng., **16**(1), p. 011004.

[7] Palmarini, R., Erkoyuncu, J. A., Roy, R., and Torabmostaedi, H., 2018, "A Systematic Review of Augmented Reality Applications in Maintenance," Rob. Comput.-Int. Manufact., **49**, pp. 215–228.

[8] Young, C., and Rai, R., 2021, "Automated Procedure Reconfiguration Framework for Augmented Reality-Guided Maintenance Applications," ASME J. Comput. Inf. Sci. Eng., **21**(6), p. 061011.

[9] Ong, S. K., Zhang, J., Shen, Y., and Nee, A. Y., 2011, "Augmented Reality in Product Development and Manufacturing" *Handbook of Augmented Reality*, B. Furht, ed., Springer, New York, pp. 651–669.

[10] Jiang, S., and Nee, A., 2013, "A Novel Facility Layout Planning and Optimization Methodology," CIRP. Ann., **62**(1), pp. 483–486.

[11] Ong, S., Yuan, M., and Nee, A., 2008, "Augmented Reality Applications in Manufacturing: A Survey," Int. J. Prod. Res., **46**(10), pp. 2707–2742.

[12] Liu, R., Duan, G.-j., and Liu, J., 2019, "A Framework for Model-Based Integrated Inspection," Int. J. Adv. Manuf. Technol., **103**(9), pp. 3643–3665.

[13] ASME Y14.5-2009, 2009. *Dimensioning and Tolerancing.* American Society of Mechanical Engineers, New York.

[14] Kwon, S., Mun, D., Kim, B. C., Han, S., and Suh, H.-W., 2019, "B-Rep Model Simplification Using Selective and Iterative Volume Decomposition to Obtain Finer Multi-Resolution Models," Comput.-Aided Design, **112**, pp. 23–34.

[15] Pratt, M. J., Anderson, B. D., and Ranger, T., 2005, "Towards the Standardized Exchange of Parameterized Feature-Based CAD Models," Comput.-Aided Design, **37**(12), pp. 1251–1265.

[16] ISO 10303-242:2014, 2014, *Industrial Automation Systems and Integration – Product Data Representation and Exchange - Part 242: Application Protocol: Managed Model-based 3D Engineering*, International Organization for Standardization, Geneva, Switzerland.

[17] Feeney, A. B., Frechette, S. P., and Srinivasan, V., 2015, "A Portrait of an ISO STEP Tolerancing Standard as an Enabler of Smart Manufacturing Systems," ASME J. Comput. Inf. Sci. Eng., **15**(2), p. 021001.

[18] AP242.org, 2009. Development of a Convergent Modular STEP Application Protocol Based on AP 203 and AP 214: STEP AP 242 –Managed Model Based 3D Engineering. Version 1.0.

[19] ISO 1101:2012, 2012, *Geometrical Product Specifications (GPS) –Geometrical Tolerancing – Tolerances of Form, Orientation, Location, and Run-out*, International Organization for Standardization, Geneva, Switzerland.

[20] ISO 23952:2020, 2020, *Industrial Automation Systems and Integration – Quality Information Framework (QIF) – An Integrated Model for Manufacturing Quality Information*, International Organization for Standardization, Geneva, Switzerland.

[21] Trainer, A., Feeney, A. B., and Hedberg, T. D., 2015, "Validation for Downstream Computer Aided Manufacturing and Coordinate Metrology Processes." NIST GCR 16-003.

[22] Hall, S., and Takahashi, R., 2017, "Augmented and Virtual Reality: The Promise and Peril of Immersive Technologies." Media & Entertainment.

[23] Urbas, U., Vrabic, R., and Vukašinović, N., 2019, "Displaying Product Manufacturing Information in Augmented Reality for Inspection," Proc. CIRP, **81**, pp. 832–837.

[24] Kleverud, F., 2018, "Interactive Visualization of CAD Data in Real Time Rendering," Master's Thesis, Department of Industrial and Materials Science, Chalmers University of Technology, Gothenburg, Sweden.

[25] Fang, F. Z., Li, Z., Arokiam, A., and Gorman, T., 2016, "Closed Loop PMI Driven Dimensional Quality Lifecycle Management Approach for Smart Manufacturing System," Proc. CIRP, **56**, pp. 614–619.

[26] Polvi, J., Taketomi, T., Moteki, A., Yoshitake, T., Fukuoka, T., Yamamoto, G., Sandor, C., and Kato, H., 2017, "Handheld Guides in Inspection Tasks: Augmented Reality Versus Picture," IEEE Trans. Visualiz. Comput. Graphics, **24**(7), pp. 2118–2128.

[27] Runji, J. M., and Lin, C. -Y., 2020, "Markerless Cooperative Augmented Reality-based Smart Manufacturing Double-check System: Case of Safe Pcba Inspection Following Automatic Optical Inspection," Rob. Comput.-Int. Manufact., **64**, p. 101957.

[28] Li, S., Zheng, P., and Zheng, L., 2020, "An Ar-Assisted Deep Learning-Based Approach for Automatic Inspection of Aviation Connectors," IEEE Trans. Indus. Inform., **17**(3), pp. 1721–1731.

[29] Fiorentino, M., Monno, G., and Uva, A., 2009, "Tangible Digital Master for Product Lifecycle Management in Augmented Reality," Int. J. Inter. Design Manufact. (IJIDeM), **3**(2), pp. 121–129.

[30] Fiorentino, M., Uva, A. E., and Monno, G., 2011, "Product Manufacturing Information Management in Interactive Augmented Technical Drawings," World Conference on Innovative Virtual Reality, Vol. **44328**, pp. 113–122.

[31] Urbas, U., Ariansyah, D., Erkoyuncu, J. A., and Vukainovi, N., 2021, "Augmented Reality Aided Inspection of Gears," Tehnički vjesnik, **28**(3), pp. 1032–1037.

[32] Sharp, M., Hedberg Jr., T., Bernstein, W., and Kwon, S., 2021, "Feasibility Study for An Automated Engineering Change Process," Int. J. Prod. Res., **59**(2), pp. 1–16.

[33] Schroeder, W. J., Zarge, J. A., and Lorensen, W. E., 1992, "Decimation of Triangle Meshes," Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp. 65–70.

[34] ISO 10303-21:2002, 2002, *Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure*, International Organization for Standardization, Geneva, Switzerland.

[35] ISO/IEC 19775:2013, 2013, *Information Technology–Computer Graphics, Image Processing and Environmental Data Representation–Extensible 3D (X3D)*, International Organization for Standardization, Geneva, Switzerland.

[36] Lipman, R., and Kwon, S., 2021, "STEP File Analyzer and Viewer User Guide (Update 7)". *Advanced Manufacturing Series (NIST AMS),* ***200-12***.

[37] Stefan, L., Hermon, S., and Faka, M., 2018, "Prototyping 3D Virtual Learning Environments With X3D-Based Content and Visualization Tools," BRAIN. Broad Res. Artifi. Intel. Neurosci., **9**, pp. 6–20.

[38] Consortium, V., 1997, ISO/IEC 14772-1 1997 the Virtual Reality Modeling Language (vrml97).

[39] Ishii, H., 2008, "The Tangible User Interface and Its Evolution," Commun. ACM, **51**(6), pp. 32–36.

[40] Sobel, W., and Eckenrode, R., 2017, Mxd final report project 14-10-01. avm standards development and promulgation. Technical Report, System Insights.

[41] Jauhar, T. A., Han, S., and Kwon, S., 2021, "Downstream Computer-aided Design, Engineering, and Manufacturing Integration Using Exchangeable Persistent Identifiers in Neutral Re-imported Computer-Aided Design Models," ASME J. Comput. Inf. Sci. Eng., **21**(1), p. 014501.

[42] Sanjiv, K., 2016, "How Augmented Reality Can Revolutionize Manufacturing." Industry Week.

[43] Carpendale, M. S. T., 2003, "Considering Visual Variables as a Basis for Information Visualisation".

[44] Regazzoni, D., Rizzi, C., and Vitali, A., 2018, "Virtual Reality Applications: Guidelines to Design Natural User Interface," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Quebec, Canada, Aug. 26–29, Vol. 51739, American Society of Mechanical Engineers, p. V01BT02A029.

[45] Ghazwani, Y., and Smith, S., 2020, "Interaction in Augmented Reality: Challenges to Enhance User Experience," Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations, pp. 39–44.