**Proceedings of the ASME 2021 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2021
August 17-20, 2021, Virtual Conference, USA**

IDETC2021-71329

VISUALIZING MODEL-BASED PRODUCT DEFINITIONS IN AUGMENTED REALITY

Teodor Vernica[†♠], **Robert Lipman**[‡], **William Z. Bernstein**[‡*]
[†]Associate, Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA
[♠]Aarhus University, Department of Computer Science, 8200 Aarhus N, Denmark
[‡]Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA
Email: {teodor.vernica, robert.lipman, william.bernstein}@nist.gov

ABSTRACT

Augmented reality (AR) technologies present immense potential for the design and manufacturing communities. 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 Product Manufacturing Information (PMI) with three-dimensional (3D) model data into an AR scene. We demonstrate our pipeline by interacting with annotated parts while continuously tracking their pose and orientation. Our work provides insight on how to address fundamental issues related to interoperability between domain-specific models and AR systems.

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 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. Though 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 this work, we investigate the feasibility of porting standards-compliant product definitions, including product manufacturing information (PMI) annotations, into AR environments. Figure 1 provides an example of a model-based product definition, displayed in 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 [5], 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 PMI information into AR, we developed a pipeline that automatically leverages the standard product model as a reference. We validate our approach by using several parts in an assembly. To conclude, we enumerate challenges faced in the integration and use of the models in AR.

Our primary use case is overlaying standards-based in-

---

[*] Address all correspondence to this author.

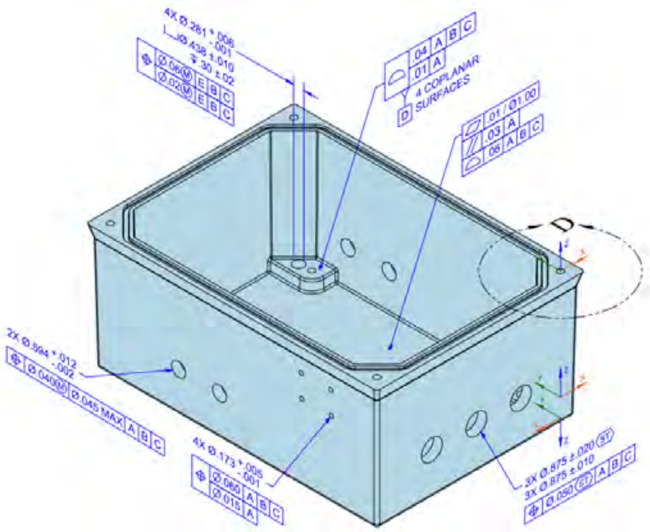


Figure 1: Example of a CAD model with PMI annotations[1].

spection data onto design information, including GD&T 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 those 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 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 (STEP) is a neutral representation of product data used for manufacturing [6]. STEP files facilitate interoperability between different CAD software and are used to represent PMI and other information vital for the smart manufacturing digital thread. STEP is maintained by the International Organization of Standardization[2] (ISO) and is actively developed to meet the requirements of the engineering community.

STEP AP242 [7], referred to as Managed Model Based 3D Engineering, covers the scopes of AP203 and AP214 and contains new capabilities for computer-interpretation of manufacturing and assembly information, including surface finish, manufacturing process information, and tolerances [8, 9].

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 [5] and ISO 1101:2012 [10] are the industry standards for the syntax and semantics of GD&T.

Though 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 AR engines

While AR is proving to have numerous industrial use cases, developing applications requires significant time investment. 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 such as Unity[3]. 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, Unity does not 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 therefore hinder experimentation and use of these new technologies.

With the increase in adoption of Industrial AR solutions [11], 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. Of important note are PiXYZ's studio[4]

---

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

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

[3] https://unity.com/

[4] https://www.pixyz-software.com/studio/

and plugin[5]. PiXYZ studio is a data preparation tool that enables the conversion and optimization of 3D CAD data to more lightweight tessellated representations, better suited for visualization purposes. PiXYZ plugin is a Unity plugin that provides a more direct integration and support for additional features, such as the ability to import point cloud data. Given the lack of native support, Unity has partnered with PiXYZ and endorses the plugin as the way to incorporate CAD data inside the game engine [12].

While commercial products, such as PiXYZ and others, provide robust and easy-to-use solutions for development, and have previously been successfully leveraged by researchers [13, 14], 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 Unity.

2.3 Use of PMI within AR

While limited, previous work has used AR as a tool for visualizing manufacturing data. Urbas et al. [13] 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. In contrast, we showcase a means to achieve the same goal by leveraging standards-based open-source tools. Section 4.1 showcases our AR application, which validates our approach.

Fiorentino et al. [15] 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. 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 [16]. In our work, the focus is 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.

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.

To achieve this vision, a standards-based approach is necessary. In our approach, we rely on standards to overcome data interoperability challenges. Standards, specifically standard data representations, provide mechanisms for data interchange between disparate computer-aided engineering (CAE) 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 [17]. 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[6], which inputs a STEP Part 21 (P21) file [18] (or a STEP instance file) and outputs an X3D file including annotations and views per the user's request. The exact part geometry is converted to faceted geometry for X3D, an ISO standard [19]. The NIST STEP File Analyzer and Viewer[7] (SFA) uses the translator for enhanced visualization of geometry and graphical PMI on a web browser. 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 [20].

Our approach is fully standards-based and open-source. 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

---

[5] https://www.pixyz-software.com/plugin/

[6] https://github.com/usnistgov/STP2X3D/

[7] https://www.nist.gov/services-resources/software/step-file-analyzer-and-viewer

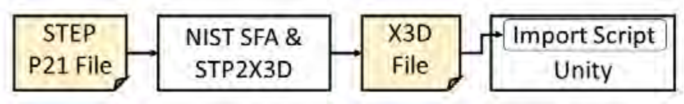


Figure 2: Implemented pipeline.

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.

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 eXtensible Markup Language (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 represents our implemented pipeline.

The import script is (1) 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.

```
<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: Example IndexFaceSet element.

Listing 1 shows 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 correspond 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).

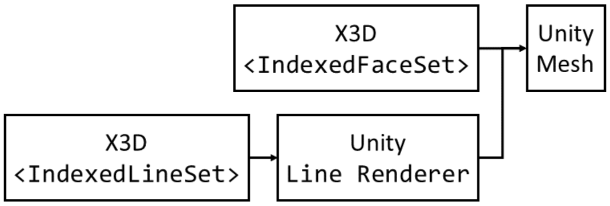


Figure 3: Mapping of elements from X3D to Unity.

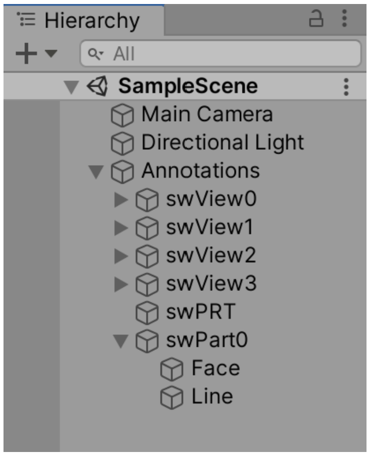


Figure 4: Example of generated Unity hierarchy.

Since the number of indexes of `<IndexedLineSet>` elements is not necessarily a multiple of 3 (e.g., `<IndexedLineSet coordIndex=’0 1 -1’>`), 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 `Annotations`. 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 `Annotations` game object in this case. Meshes generated from `<IndexedFaceSet>` and `<IndexedLineSet>` elements are then generated as children of each corresponding view or part geometry in the hierarchy (`Face`, `Line`). Preserving this structure allows views, part geometry and meshes 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-

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

frame (`Line`) of the geometry (`swPart0`) is superimposed over the physical piece, providing additional context.

The import script, is not a fully fledged X3D importer, as only relevant elements are being translated to equivalent Unity counterparts. But 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.

4.1 Augmented Reality application

By leveraging the previously described pipeline (Fig. 2), we developed an AR application[8] which uses model-based tracking to register PMI annotations extracted from the as-designed CAD model, to their physical counterparts, as shown in Figs. 6 & 7.

The annotations are separated into views. 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. 8. The toggles are automatically generated based on the currently tracked model. Figure 8a 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 8b 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.

To visualize the PMI annotations in AR, we leveraged PTC's Vuforia[10] framework due to its robust tool set. The recent addition of model-based tracking allows physical 3D objects to be used as tangible user interfaces [21]. 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.

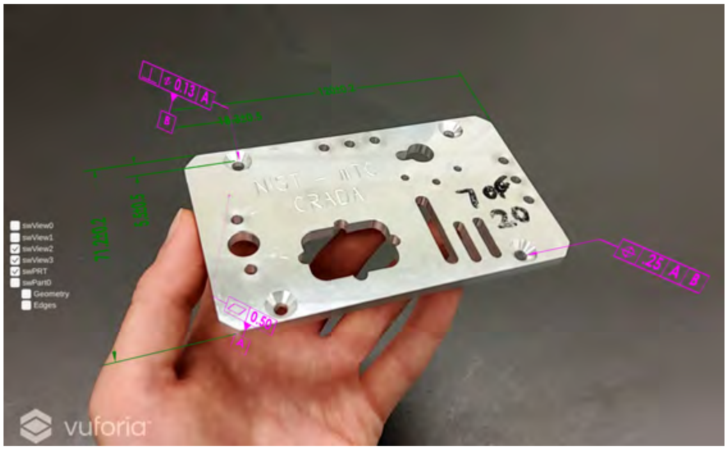


Figure 6: PMI annotations superimposed on the machined Plate[9] part from the NIST dataset. View 1 (magenta) shows datum definitions. View 2 (green) shows hole definitions.

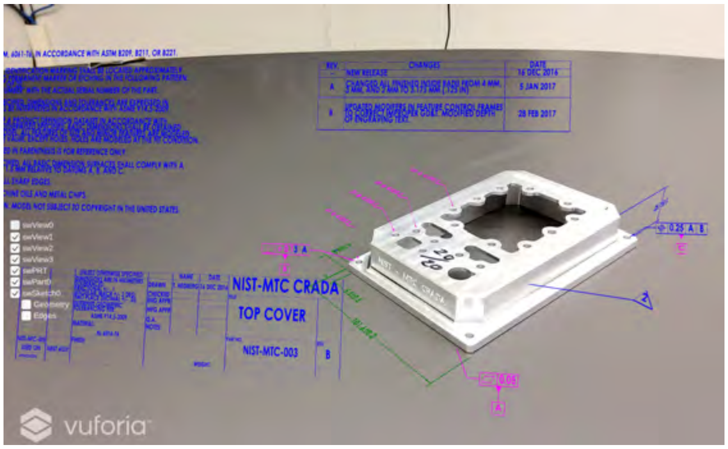


Figure 7: AR view of the Cover[11] part. View 1 (blue) presents notes and titles. View 2 (magenta) shows datum and hole definitions and View 3 (green) displays boundary definitions.

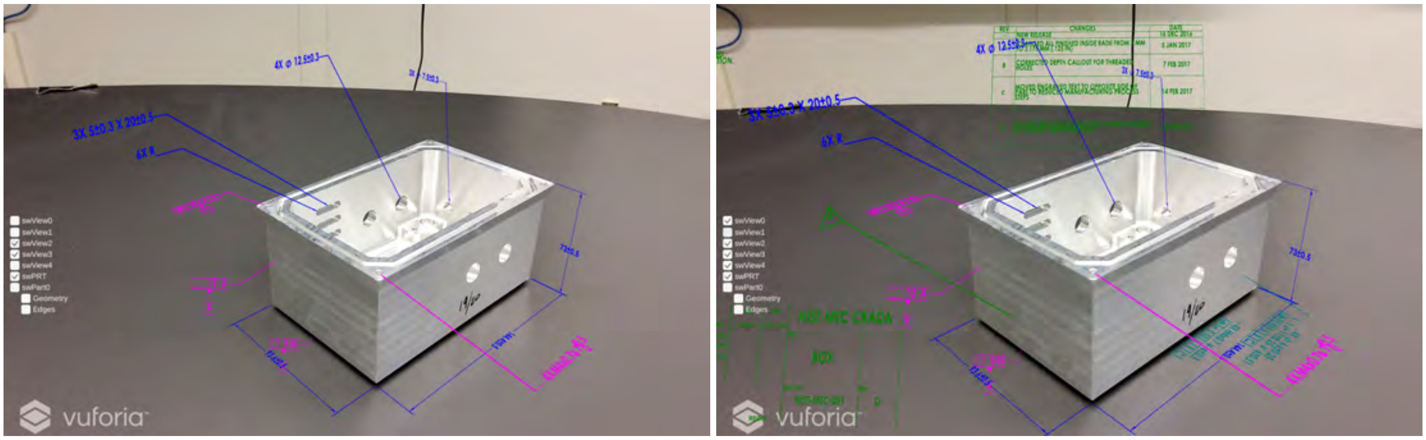
---

[8] See demo at https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-video.html

[9] https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-plate.html

[10] https://www.ptc.com/en/products/vuforia

[11] https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-cover.html



(a) (b)

Figure 8: AR views of the Box[12] from the NIST Box Assembly dataset, with different views toggled.

To validate our approach, we imported and tracked three STEP files that make-up the NIST *Design, Manufacturing, and Inspection Data for a Box Assembly* dataset[13]. The publicly available dataset consists of three CAD Models and associated data collected during the fabrication process of the parts. The application was deployed to an Android tablet, running on the Android 10 operating system.

5 Pipeline issues

The presented workflow requires manual intervention when translating files from STEP to X3D using SFA. The process can be 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.

Augmented reality tracking libraries generally provide a visual representation of the tracked model within the game engine. In doing so, digital augmentations can be positioned relative to a visual aid in the 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 present. 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 this 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. This is 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.

6 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 further investigate.

- *Spatially contextualizing annotations with respect to the decimated part model*: In the native design model, PMI annotations are fixed to the features to which they characterize. However, once the model is tessellated into a mesh for AR presentation, annotations are no longer coupled to a closed-body feature. Instead, the placement of annotations relies on the coordinate-based placement of annotations. To ensure proper placement, it is necessary to properly segment the decimated model to contextualize annotation placement.
- *Coordinating contextual views of annotations*: The STEP

---

[12] https://pages.nist.gov/CAD-PMI-Testing/NIST-AR-box.html
[13] https://github.com/usnistgov/smstestbed/tree/master/tdp/mtc

data structure affords context views of annotations, which could be thought of 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.

- *Registering other digital thread data representation in the same spatial context*: Relating inspection data back to design models in AR is a desired use case. For example, non-destructive 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.

For tracking parts in the presented AR application, we leveraged a commercial tracking toolkit. To accomplish a fully open-source solution, a recently released open-source tracking toolkit[14] could fulfill this requirement.

7 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. 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 information contextualized to the native design model. Proper registration of such data will enable more scalable industrial AR applications. As a result, future work will center around spatially contextualizing downstream data onto design models.

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, 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.

Acknowledgements

We thank the NIST-internal and conference reviewers for their valuable feedback. We also would like to acknowledge Prof. Soonjo Kwon, who developed the STEP to X3D Translator (STP2X3D).

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.

REFERENCES

[1] Egger, J., and Masood, T., 2020. "Augmented reality in support of intelligent manufacturing–a systematic literature review". *Computers & Industrial Engineering,* ***140***, p. 106195.

[2] Pedone, G., and Mezgár, I., 2018. "Model similarity evidence and interoperability affinity in cloud-ready industry 4.0 technologies". *Computers in Industry,* ***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". *Advanced Engineering Informatics,* ***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". In ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection.

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

[6] Pratt, M. J., Anderson, B. D., and Ranger, T., 2005. "Towards the standardized exchange of parameterized feature-based cad models". *Computer-Aided Design,* ***37***(12), pp. 1251–1265.

[7] 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.

[8] Feeney, A. B., Frechette, S. P., and Srinivasan, V., 2015. "A portrait of an iso step tolerancing standard as an enabler of

---

[14] https://github.com/usnistgov/TrackingExpertPlus

smart manufacturing systems". *Journal of Computing and Information Science in Engineering,* ***15***(2).

[9] 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.

[10] ISO 1101:2012, 2012. *Geometrical product specifications (GPS) – Geometrical tolerancing – Tolerances of form, orientation, location, and run-out*. International Organization for Standardization, Geneva, Switzerland.

[11] Hall, S., and Takahashi, R., 2017. "Augmented and virtual reality: The promise and peril of immersive technologies". *Media & Entertainment*, October.

[12] Martin, E., 2019. Get to the point: Pixyz 2019.2, Dec.

[13] Urbas, U., Vukašinović, N., et al., 2019. "Displaying product manufacturing information in augmented reality for inspection". *Procedia CIRP,* ***81***, pp. 832–837.

[14] Kleverud, F., 2018. "Interactive visualization of cad data in real time rendering". Master's thesis.

[15] Fiorentino, M., Monno, G., and Uva, A., 2009. "Tangible digital master for product lifecycle management in augmented reality". *International Journal on Interactive Design and Manufacturing (IJIDeM),* ***3***(2), pp. 121–129.

[16] Fiorentino, M., Uva, A. E., and Monno, G., 2011. "Product manufacturing information management in interactive augmented technical drawings". In World Conference on Innovative Virtual Reality, Vol. 44328, pp. 113–122.

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

[18] 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.

[19] ISO/IEC 19775:2013, 2013. *Information technology — Computer graphics, image processing and environmental data representation — Extensible 3D (X3D)*. International Organization for Standardization, Geneva, Switzerland.

[20] Stefan, L., Hermon, S., and Faka, M., 2018. "Prototyping 3d virtual learning environments with x3d-based content and visualization tools". *BRAIN. Broad Research in Artificial Intelligence and Neuroscience,* ***9***, pp. 6–20.

[21] Ishii, H., 2008. "The tangible user interface and its evolution". *Communications of the ACM,* ***51***(6), pp. 32–36.