Tangible Version Control: Exploring a Physical Object's Alternative Versions

Maximilian Letter
maximilian.letter@bht-berlin.de
Berlin University of Applied Sciences and Technology
Berlin, Germany

Katrin Wolf
katrin.wolf@bht-berlin.de
Berlin University of Applied Sciences and Technology
Berlin, Germany

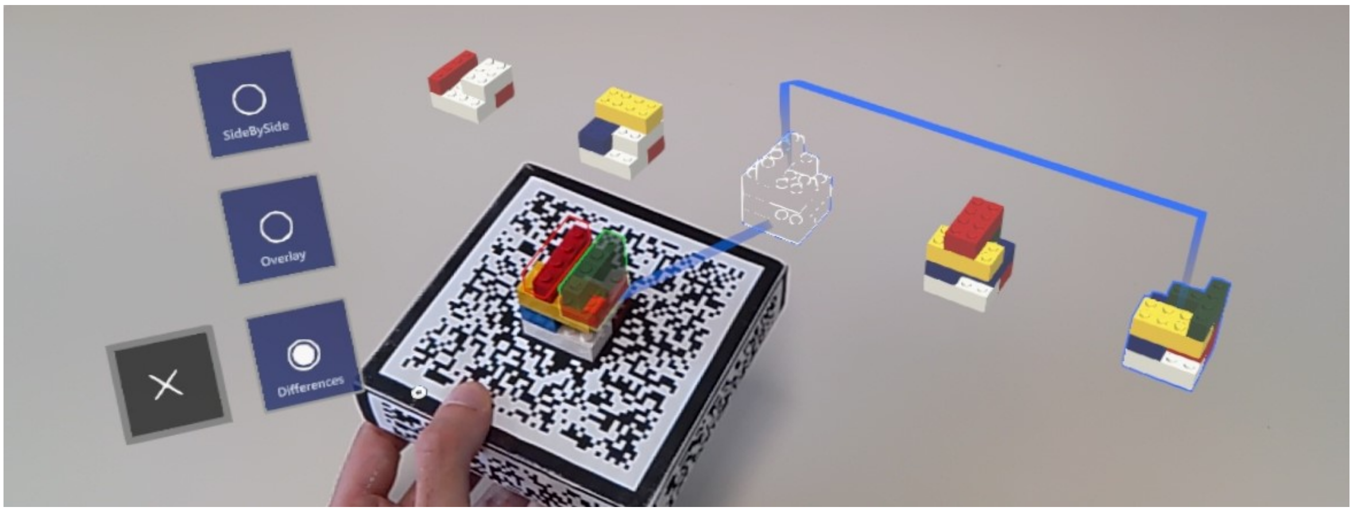

Figure 1: Prototype implementation of Tangible Version Control. Differences between a physical artifact and an alternative version of itself are displayed onto the object, a timeline of versions is visible in the background.

ABSTRACT

In iterative physical object creation, only the latest design state is manifested in the physical artifact, while information about previous versions are lost. This makes it challenging to keep track of changes and developments in iterative physical design. In this paper, we propose the concept of Tangible Version Control (TVC), inspired by the visualizations of traditional version control systems. In TVC, the real-world artifact itself is used for exploring its alternative versions in physical space, while comparisons to an alternative version are displayed seamlessly on the artifact with the use of augmented reality. Our implementation of TVC includes three different comparison modes, namely SideBySide, Overlay, and Differences. Furthermore, we discuss the anticipated use, opportunities, and challenges of using TVC in the future for individual users as well as for asynchronous collaborative work.

CCS CONCEPTS

• **Human-centered computing** → **Interaction design theory, concepts and paradigms**; **Mixed / augmented reality**.

KEYWORDS

Tangible User Interface, Physical Interaction Design, Seamless Interaction, Version Control, Augmented Reality, Asynchronous Collaboration

ACM Reference Format:
Maximilian Letter and Katrin Wolf. 2022. Tangible Version Control: Exploring a Physical Object's Alternative Versions. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3491101.3519686

1 MOTIVATION AND BACKGROUND

Designing and building three-dimensional physical objects can be done in two ways, that can be described as digital and physical design [12]. In the case of digital design, dedicated software is used, in which a user creates the object design completely virtually. As the content is digital, intermediate progress can be saved, duplicated, and shared. In the end, the user is required to either build the physical object after the blueprint they created or use a machine to fabricate the object. In physical design, however, objects are build directly in the real world and their design is iterated by modifying

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9156-6/22/04…$15.00
https://doi.org/10.1145/3491101.3519686

the physical object's appearance. While this approach is more natural to humans, as we have a natural understanding of the physical world [13], we cannot utilize the benefits of computer-supported design software. One missing feature is the saving of versions representing the design process, which are basically copies of the object at a certain point of time, as only the latest design is manifested in the physical object. We aim at bringing the benefit of exploring and comparing versions, to this date an exclusive function for digital design, into the real world, while preserving a tangible workflow with the physical artifact.

One form of saving interim states during design and development are version control systems (VCSs) [21]. Besides enabling the persistence of states and allowing users to comprehend and trace their progress, they allow for remote asynchronous collaboration between users. Numerous commercial VCS applications exist (e.g. Sourcetree[1], GitKraken[2], and GitHub Desktop[3]) to support the versioning process, to visualize a history of actions, and to compare different versions. In contrast to text-based content, the support of alternative digital document types, such as images or 3D models, is less established. Nevertheless, its use has been researched [4, 6–8, 31] and manifested in software products (e.g. GrabCAD[4] and onshape[5]). Notably, Doboš et al. proposed comparison algorithms for detecting discrepancies between models [6] and researched VCS-specific solutions in the context of 3D models [5, 7, 8]. However, all these commercial products and research projects are tailored to digital design workflows and not suitable for physical design.

One way of narrowing the gap between physical and digital design is to include the user and potential physical objects in an otherwise digital design process [11, 19, 24, 27]. A related approach is making fabrication more interactive by allowing users to interact with physical artifacts in-between or even during fabrication [17, 26, 29]. In some of those works, the authors used augmented reality (AR) to display upcoming changes during fabrication to the user [17, 26]. However, the augmented information was displayed statically at the location where the fabrication happened, either with a see-through virtual reality headset [17] or on a projection screen as a part of the machine [26]. Notably, a work of Weichel et al. additionally supported a form of versioning [26]. For this purpose, objects were digitized via 3D-scanning inside the fabrication machine. The user could then go back and forth between different versions. The prototype added or subtracted material as necessary, in order to match the alternative version. As it is build around a machining system, it is costly, mostly stationary, and disjointed from working with the own hands while in the machine.

Another benefit of the establishment of a version history, in order to understand changes made to a design, is to enable asynchronous collaboration. Perteneder et al. researched into preserving the tangibility of a physical artifact while tapping into the benefits of virtualization [18]. A motorized turntable and a camera were used to digitize physical objects as a 360 degree collection of images. The physical object and its virtual representations could then be explored by co-located as well as by remote collaborators in a web interface. While enabling a form of VCS for physical objects, the representations of the object and its history of versions were disconnected, split between real world and web interface.

When regarding the transfer of a VCS into the real world, it becomes apparent, that the support for physical objects is extremely limited, besides referring to virtual twins on two-dimensional screens. In order to utilize the human's haptic interaction skills as well as natural understanding of the physical world [13], it is of great interest to explore VCSs in a real-world context in which a physical artifact is augmented by alternative versions of itself that can then be compared against. By doing so, we aim to reduce the gap between physical and virtual worlds, enabling creators to continue iterating physical designs without diverting their attention to secondary screens.

As we are confident that the required technologies for such an approach are already existent or available in the near future, we explore the design, interaction, and visualization of a version control for physical objects. The versioning and digitization process itself is not investigated in this work. To the best of our knowledge, the concept of a version control attached to physical artifacts and displayed in the real world is a novel approach, which has not been investigated yet. Our focus is on the comparison of different versions of an object to its current physical state, including a timeline for exploration along which comparisons can be triggered. Both aspects are depicted in Figure 1.

2 TANGIBLE VERSION CONTROL

To introduce the concept of a Tangible Version Control (TVC), we first describe an abstract scenario of working with physical objects in the field of iterative product design that displays a possible workflow with TVC. The scenario is futuristic in the sense that product design is mostly done completely digital nowadays.

2.1 Scenario and Workflows

Alex works as a product designer and is assigned with the task of developing a new physical product. With an existing product at hand, which could be a precursor or a blank, Alex begins to modify it by adding, removing, or exchanging components. While doing so, interim versions of the artifact are digitized. At some point, Alex feels like she has gone off track and needs to return to an earlier version. By using TVC, Alex compares her current physical version with an earlier state. When doing so, changes on the object are displayed virtually and can be used as instructions to build from one version to another. After rebuilding to an earlier state, Alex keeps on iterating the object. Alex can use TVC at any point to compare her current physical version to the version she previously discarded going for, in order to check if ideas might transfer to the new designs. Alex finishes work for the day and returns the next day to keep on working on the product. TVC is used to get an overview of iterations done so far, before she continues. While Alex was away, TVC served as an entry point for remote asynchronous collaboration between designers at the company. Kim, a coworker of Alex, is also assigned with the task of developing a new product that is supposed to fit in the same product line as Alex' product. Kim uses TVC to directly compare his current physical artifact to

[1]https://www.sourcetreeapp.com/
[2]https://www.gitkraken.com/
[3]https://desktop.github.com/
[4]https://grabcad.com/workbench
[5]https://www.onshape.com/en/features/data-management

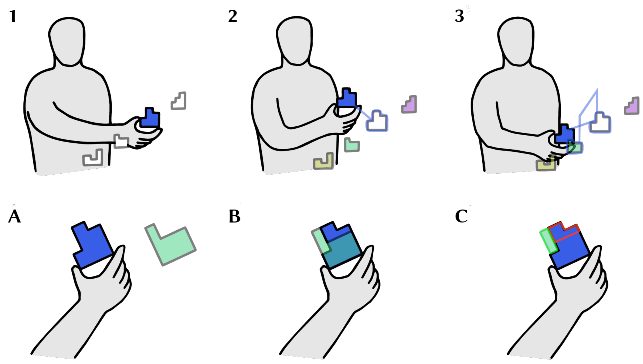

Figure 2: Implementation of the TVC concept. Upper row: timeline interactions. 1) The timeline is placed by positioning the physical artifact. 2) The relation between physical artifact and its virtual twin is displayed as a line. 3) By moving the physical artifact into another version, a comparison is triggered and highlighted in the timeline; Bottom row: comparison modes. A) *SideBySide*-mode, physical artifact and the compared against version are displayed next to each other. B) *Overlay*-mode, the compared against version is superimposed onto the physical artifact. C) *Differences*-mode, differing parts between the two versions are detected, highlighted, and animated on the physical artifact.

Alex' versions, without the need of having every physical version or his coworkers present on-site.

2.2 Concept

In order to enable such a scenario, our interaction concept of TVC attempts at merging information used in VCSs with the real world by using AR. Instead of abstracting information about alternative versions on a screen, it is visualized directly in the physical world. The user can explore a timeline of versions that is positioned in the real world as desired. The timeline represents a chronologically ordered history of versions, analogue to traditional VCS. It also contains a virtual twin of the physical object, which represents the current version. The timeline can be moved, placed, repositioned, and used for referencing the current comparison. By moving the physical artifact along the timeline after placement, the user interacts with alternative versions. When doing so, a comparison operation is started. Comparisons of versions are displayed on top of the artifact and take both, the visuals of the real world and the virtual world, in account. That way, the user can preview object states and potential changes before the object gets modified or fabricated [25]. On top, it serves as a guidance for assembling the object towards the alternative state [22]. For this purpose, parts either need to be removed, added, or exchanged. In the case of continuous material, the mass has to be reshaped as necessary. A visualization of the concept can be found in Figure 2, which covers the main interaction principles as well as our proposal of three modes for conveying a comparison, namely *SideBySide*, *Overlay*, and *Differences*.

3 PROTOTYPED INTERACTION DESIGN

To demonstrate the TVC concept, we prototyped key functionalities consisting of two logical parts, timeline and comparison. Some information that are typically present in a VCS, like the origin of changes and comments on a new version, are not regarded in this prototype, as described in Section 4.1. The implemented novelty of our concept is the application of VCS outside a screen environment, while including the physical object.

3.1 Timeline

The timeline is a completely virtual object, visualized in Figure 3. It represents information about the design process, in particular form factor differences, as well as a virtual twin of the physical object. During its placement, the timeline contents are displayed as transparent schemes with visible edges. The position of the timeline can be confirmed by interacting with a button hovering above the physical artifact. Once the timeline is placed, all versions, but the virtual twin, are rendered opaque and the object can be moved outside its virtual twin. To emphasize the belonging of physical artifact and virtual twin, a line is drawn from one to another when the physical object is near the timeline, fading out with increasing distance.

Moving the physical object into a version other than itself triggers a comparison between the two and the timeline is altered. The compared against version is highlighted by an outline and connected via a line to the virtual twin. By adding these indicators, a user can refer to the timeline at any time to recollect which version they are currently comparing against. Blue highlighting was chosen as a neutral attention grabber that would not be mistaken for the red and green highlighting colors used during the comparison.

3.2 Comparison

In TVC, the comparison between versions takes place in relation to the existing physical artifact. Hence, one component of the comparison is the physical object itself and the other one is the compared against virtual version. It must be kept in mind that other than fully digital content, the physical object cannot be altered but augmented. The artifact itself is not rendered in AR, as it is present and visible in the real world. Instead, a phantom model is used so that the object correctly occludes virtual content [14].

We propose three different modes for conveying a comparison between two versions, namely *SideBySide*, *Overlay*, and *Differences*, see Figure 4. We follow the three main principles for comparing visual structures Gleicher et al. identified in their work [10], namely juxtaposition (*SideBySide*), superposition (*Overlay*), and explicit encoding (*Differences*). Our representations are inspired by related work and applications that compare 3D models against each other [4, 6, 8]. During development, each mode was implemented with several alternative visualization techniques (*SideBySide* with changeable sides and pivot points; *Overlay* with different materials, such as transparent and wireframes; *Differences* with outlines only or blocks visualized in solid highlight color). The final visualizations were selected by multiple HCI researchers during an iterative design process. The implemented comparison modes are described in the following.

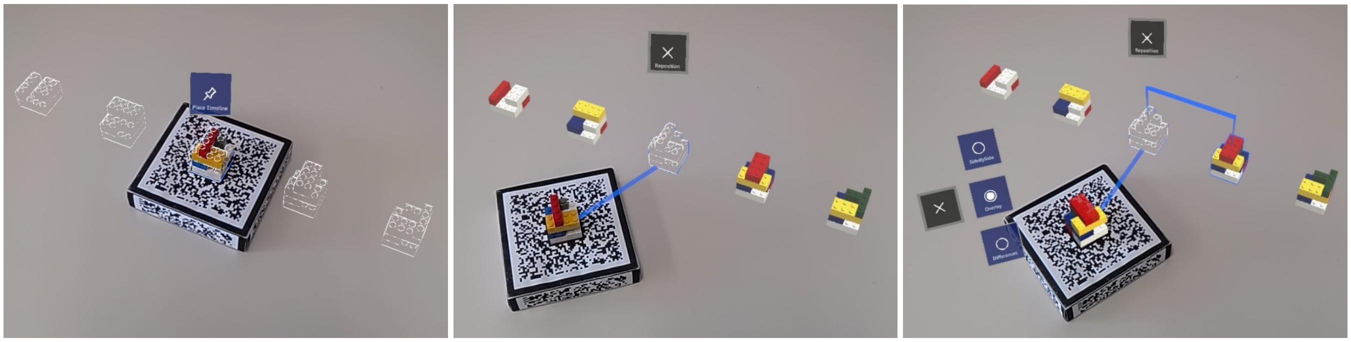

Figure 3: Timeline. Left: positioning the timeline by moving the physical artifact, schemes of versions are visualized as edges; Center: the timeline is placed, versions are rendered opaque, the virtual twin of the physical object is highlighted and emphasized by a line; Right: during a comparison between a version and the physical object, the corresponding versions are highlighted and linked in the timeline.

SideBySide The SideBySide condition emulates a comparison process of physical objects in the real world, in which two physical objects would be placed next to each other for comparison. In our implementation, the virtual comparison object floats to the right of the physical artifact. The two objects maintain the same distance and relative position to each other. The virtual comparison object mimics the artifact's internal rotation. This allows to have both objects fully in view while freely inspecting the objects from different angles by moving and rotating the physical artifact.

Overlay The Overlay condition uses the naive approach of superimposing an alternative version directly on top of the physical artifact. Therefore, the comparison object matches position and rotation of the physical artifact.

Differences Similar to the behavior proposed in the related work regarding 3D model comparison [4–6], as well as state-of-the-art VCS software, differences are automatically detected and highlighted. The differences are detected through a simple part-based comparison algorithm. It compares the two models part-wise for position, rotation, and descriptive names and classifies the parts as unchanged, modified, added, or removed. Relevant parts are then emphasized with color-coded outlines on top of the artifact. Parts that are added compared to the currently present physical object are highlighted in green, while removed parts are shown in red, and modified parts (e.g. the same part in a different color) transition between green and red. In order to let the user know what parts exactly were added to the marked positions, an opaque rendering of the parts is faded in and out. In the case of modified parts, the outline color is further transitioning from a red color, when the physical part is visible, to a green color, when the virtual version's part is visible.

3.3 Technology and Tracking

We developed our prototype for TVC in Unity[6] using version 2019.4. Basic functionalities, like object placement and UI elements, were adopted from the Mixed Reality Toolkit (MRTK)[7]. The versions shown in the timeline were pre-built in the Mecabricks workshop[8] and imported into Unity, as this prototype does not concern the digitization of physical objects. Tracking features were realized using the Vuforia SDK for Unity[9] in version 9.8. The prototype supports both versions of Microsoft HoloLens[10]. Our prototype implementation is publicly available[11].

As early experiments showed that the performance of model tracking[12] on HoloLens is poor, we instead used a trackable cube marker[13] as substitution for the prototype. The marker is cut out in the center, where the actual object is placed. Object and paper marker are then fixated on a stable plate that can be held without it bending. In addition to making the object tracking relatively stable, marker-based tracking allows for consistent tracking during modification of the physical artifact in order to match another version.

4 DISCUSSION

We presented the core concept of TVC as well as a first prototype implementation. Future prototypes, of course, can focus on various opportunities of physical objects' version control. We here discuss future functionalities and research directions that seem most reasonable to extend TVC, aiming at an increased scope as well as supporting collaborative and asynchronous work.

4.1 Future Functionalities

The proposed prototype is limited by the available technology and scope of this work, which offers multiple opportunities for improvement in the future. Instead of marker tracking, the tracking of objects in TVC is imagined to work with model tracking. That way, artificial markers could be omitted and objects without a flat basis would be supported. Further, the creation of new versions, which is a crucial aspect of a VCS, was not investigated in the prototype and needs to be addressed at one point. An optimal scenario would be the scanning and reconstruction [3] of the physical object with the head worn device itself, possibly enabled by the use of

[6] https://unity.com/
[7] https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity
[8] https://mecabricks.com/en/workshop
[9] https://developer.vuforia.com/downloads/sdk
[10] https://www.microsoft.com/en-us/hololens
[11] https://github.com/MaximilianLetter/TVC_public
[12] https://library.vuforia.com/objects/model-targets
[13] https://library.vuforia.com/objects/multi-targets

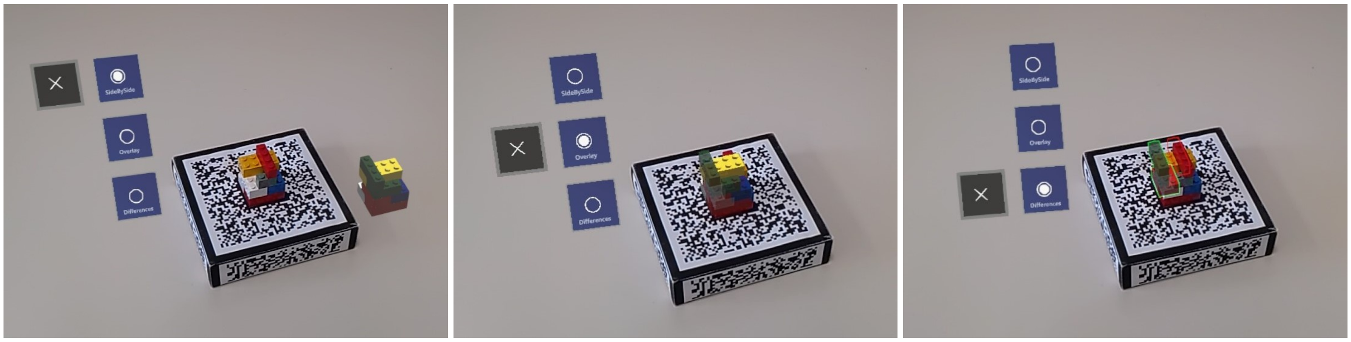

Figure 4: Comparison modes. *SideBySide*: physical artifact and compared against version are placed side by side, similar to the comparison of two physical objects; *Overlay*: the compared against version is superimposed onto the physical artifact; *Differences*: the differences between the physical artifact and the compared against version are detected and highlighted by outlines, added and modified parts are visualized as animated blocks that fade their transparency.

additional sensors, such as a depth camera [9], or via a software based approach, such as photogrammetry [20]. A currently feasible solution would be the use of external 3D scanning devices[14][15], in which the current physical object is inserted and digitized. An additional requirement for everyday use of TVC in the future is the assurance of conformity of physical artifact and virtual twin, as the mechanics of TVC crumble if this information is not aligned. In the following, we highlight some features of traditional VCSs that remain for further investigation as they might be interesting for TVC.

Authors and Comments: In VCS, each version usually comes with a reference to the author who is responsible for the new version, as well as a comment on its purpose. While it would be easy to include text information into TVC, clutter should be avoided and more sophisticated solutions might be achievable. One possible approach could be the use of auditory in- and output, which would allow using all display bandwidth for object presentation and hand interactions for working with the object. This feature is an essential development, as it is a further step to enable collaboration between users of TVC.

Extended Complexity: In our prototype, we regard the simple case of a timeline consisting of a single branch with multiple versions that are each made up out of a limited amount of predefined parts. It is of interest how to scale the concept of TVC to bigger repositories consisting out of multiple branches, each with many versions. This is especially important as physical space and user mental capabilities are limited. In addition, we are interested in how the concept translates to other materials that are more complicated than LEGO parts, for example a continuous material such as clay, as well as objects of bigger size, that do not fit the field of view of the user or the headset. To make TVC more general purpose, our naive part-based comparison function needs to be replaced by a more sophisticated algorithm in future work, for example one of the methods proposed by related work [4–6].

Functionality of Objects: Up to this point, TVC focuses exclusively on model appearance as the content of version control. An interesting extension would be the combination of appearance with functionality, which is for example manifested in code or represented through animations.

[14]https://matterandform.net/store/products/MFS1V2
[15] https://www.artec3d.com/portable-3d-scanners/artec-micro

4.2 Challenges and Opportunities of TVC

Some of the required functionalities discussed in Section 4.1 are heavily dependent on suitable hardware. One example is the choice of our AR device, where the HoloLens suffers from a narrow field of view as well as limited tracking capabilities. Another challenge is the identification of appropriate methods or devices for digitizing new versions based on a physical object in the TVC workflow. However, we are optimistic about the ongoing research into AR devices [2] that will allow for enhanced qualities in sensory capabilities [23, 28], optics solutions [1, 15, 30], and processing power.

While we aim to explore more scenarios and materials, there are some areas in which TVC appears suboptimal. One example is the use of immobile artifacts that cannot be moved, thereby defeating the interaction concept of TVC. Another example is working with materials that need to be permanently fused in between versions, for example by welding or gluing, as in these cases a transition to earlier versions might not be possible anymore.

Regarding the potential of TVC as a comprehensive system, we envision TVC as an approach that could help to move work on physical content from screens back into the physical world. We argue that maintaining a tangible workflow during staggered design iterations is a very valuable feature for makers, designers, and potential other professions. That way, the gap between physical and virtual information is brought closer together and access to object-relevant information becomes seamless, making it more natural for humans to explore and understand [13]. Conventional design methods, e.g. CAD plus VCS, have impregnable benefits in variability and accuracy that TVC cannot compensate because of its physicality. However, TVC can support other forms of creative design that are enabled by working in the physical world, as the medium used for design or creation majorly influences the outcome [16]. TVC might primarily find use in low level prototyping, where the ease of working with physical objects benefits design thinking as well as a documentation technology for more complex artifacts.

Further, TVC supports not just the design process, but the actual creation of artifacts that can then be used as molds, prototypes, or final products. We anticipate TVC being applied in work domains that already use physical prototyping, like fashion design, architecture, and landscape planning.

Further, we see potential in TVC to be used in collaborative scenarios. For this use case, newly created versions of an object would be managed in the cloud and be accessed by other collaborators, who then can compare their object's current physical state to the alternative versions. Additional synchronous features could be live information about which version is checked out by whose collaborator as well as if a new version is currently being created. Enabling remote and asynchronous collaboration would in conclusion reduce the need for extensive travel and thus, decrease human impact on the global climate.

With the proposal of TVC, we aim at giving an impulse about alternative takes on VCS in the context of physical objects. Further, we are confident that we were able to motivate further thinking and potential research into that direction and initiate a discussion about the challenges, opportunities, and promises of a version control that takes place in the real world.

5 CONCLUSION

In this work, we introduced Tangible Version Control (TVC), a novel interaction technique for exploring and comparing alternative versions of physical objects, using the tangible object itself as input device and projection space. The presented concept was implemented in a prototype application using augmented reality, marker-based tracking, and LEGO bricks as an exemplary physical object. We proposed three comparison modes, namely *SideBySide*, *Overlay*, and *Differences*, for displaying comparisons between the physical and a virtual version, as well as a timeline that displays all versions. We described the future development of TVC required to enable the possibilities that the concept promises. Further, the challenges and opportunities of this concept were discussed. While we acknowledge the benefits of fully digital design approaches, TVC provides an alternative way of iterating on physical objects that benefits from a natural and seamless interaction, while supporting an alternative form of creative work conducted in the physical world. We aim at contributing to the field of seamless interaction design by presenting a concept, accompanied by a prototype implementation, that allows for computer-supported work on physical objects in the real world.

REFERENCES

[1] Kiseung Bang, Changwon Jang, and Byoungho Lee. 2019. Curved holographic optical elements and applications for curved see-through displays. *Journal of Information Display* 20, 1 (2019), 9–23. https://doi.org/10.1080/15980316.2019.1570978 arXiv:https://doi.org/10.1080/15980316.2019.1570978

[2] Pietro Cipresso, Irene Alice Chicchi Giglioli, Mariano Alcañiz Raya, and Giuseppe Riva. 2018. The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature. *Frontiers in Psychology* 9 (2018), 2086. https://doi.org/10.3389/fpsyg.2018.02086

[3] Angela Dai and Matthias Nießner. 2019. Scan2mesh: From unstructured range scans to 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, CA, USA, 5574–5583.

[4] Jonathan D. Denning and Fabio Pellacini. 2013. MeshGit: diffing and merging meshes for polygonal modeling. *ACM Transactions on Graphics* 32, 4 (July 2013), 35:1–35:10. https://doi.org/10.1145/2461912.2461942

[5] Jozef Doboš, Niloy J Mitra, and Anthony Steed. 2014. 3D Timeline: Reverse engineering of a part-based provenance from consecutive 3D models. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, European Association for Computer Graphics, Geneve, Switzerland, 135–144.

[6] Jozef Doboš and Anthony Steed. 2012. 3D Diff: an interactive approach to mesh differencing and conflict resolution. In *SIGGRAPH Asia 2012 Technical Briefs (SA '12)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/2407746.2407766

[7] Jozef Doboš and Anthony Steed. 2012. 3D Revision Control Framework. In *Proceedings of the 17th International Conference on 3D Web Technology* (Los Angeles, California) *(Web3D '12)*. Association for Computing Machinery, New York, NY, USA, 121–129. https://doi.org/10.1145/2338714.2338736

[8] Jozef Doboš, Carmen Fan, Sebastian Friston, and Charence Wong. 2018. Screen space 3D diff: a fast and reliable method for real-time 3D differencing on the web. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology (Web3D '18)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3208806.3208809

[9] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 2015. 3D scanning deformable objects with a single RGBD sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Boston, MA, USA, 493–501.

[10] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309. https://doi.org/10.1177/1473871611416549

[11] Emrecan Gulay, Toni Kotnik, and Andrés Lucero. 2021. Exploring a Feedback-Oriented Design Process Through Curved Folding. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3411764.3445639

[12] Emrecan Gulay and Andrés Lucero. 2019. Integrated Workflows: Generating Feedback Between Digital and Physical Realms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300290

[13] Hiroshi Ishii. 2008. Tangible Bits: Beyond Pixels. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction* (Bonn, Germany) *(TEI '08)*. Association for Computing Machinery, New York, NY, USA, xv–xxv. https://doi.org/10.1145/1347390.1347392

[14] Denis Kalkofen, Christian Sandor, Sean White, and Dieter Schmalstieg. 2011. Visualization Techniques for Augmented Reality. In *Handbook of Augmented Reality*, Borko Furht (Ed.). Springer New York, New York, NY, 65–98. https://doi.org/10.1007/978-1-4614-0064-6_3

[15] Lu Lu, Taha Masood, and Barry Silverstein. 2021. Toward Lighter, Thinner AR/VR Systems. *Opt. Photon. News* 32, 7 (Jul 2021), 42–47. http://www.osa-opn.org/abstract.cfm?URI=opn-32-7-42

[16] Marshall McLuhan and Quentin Fiore. 1967. The medium is the message. *New York* 123, 1 (1967), 126–128.

[17] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174153

[18] Florian Perteneder, Eva-Maria Grossauer, Yan Xu, and Michael Haller. 2015. Catch-Up 360: Digital Benefits for Physical Artifacts. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (Stanford, California, USA) *(TEI '15)*. Association for Computing Machinery, New York, NY, USA, 105–108. https://doi.org/10.1145/2677199.2680564

[19] Patrick Reipschläger and Raimund Dachselt. 2019. DesignAR: Immersive 3D-Modeling Combining Augmented Reality with Interactive Displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19)*. Association for Computing Machinery, New York, NY, USA, 29–41. https://doi.org/10.1145/3343055.3359718

[20] Fabio Remondino, Alberto Guarnieri, and Antonio Vettore. 2005. 3D modeling of close-range objects: photogrammetry or laser scanning?. In *Videometrics VIII*, J.-Angelo Beraldin, Sabry F. El-Hakim, Armin Gruen, and James S. Walton (Eds.), Vol. 5665. International Society for Optics and Photonics, SPIE, San Jose, CA, United States, 216 – 225. https://doi.org/10.1117/12.586294

[21] Nayan B. Ruparelia. 2010. The History of Version Control. *SIGSOFT Softw. Eng. Notes* 35, 1 (Jan. 2010), 5–9. https://doi.org/10.1145/1668862.1668876

[22] Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. Drill Sergeant: Supporting Physical Construction Projects through an Ecosystem of Augmented Tools. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1607–1614. https://doi.org/10.1145/2851581.2892429

[23] Hossein Shahinian, Alyson Markos, Jayesh Navare, and Dmytro Zaytsev. 2019. Scanning depth sensor for see-through AR glasses. In *Optical Design Challenge 2019*, Bernard C. Kress (Ed.), Vol. 11040. International Society for Optics and Photonics, SPIE, San Francisco, CA, United States, 60 – 65. https://doi.org/10.1117/12.2523829

[24] Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. 2006. ModelCraft: capturing freehand annotations and edits on physical 3D models. In *Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/1166253.1166258

[25] Evgeny Stemasov, Tobias Wagner, Jan Gugenheimer, and Enrico Rukzio. 2020. Mix&Match: Towards Omitting Modelling Through In-situ Remixing of Model Repository Artifacts in Mixed Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376839

[26] Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. 2015. ReForm: Integrating Physical and Digital Design through Bidirectional Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/2807442.2807451

[27] Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. 2014. MixFab: a mixed-reality environment for personal fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3855–3864. https://doi.org/10.1145/2556288.2557090

[28] Min-Yu Wu, Pai-Wen Ting, Ya-Hui Tang, En-Te Chou, and Li-Chen Fu. 2020. Hand pose estimation in object-interaction based on deep learning for virtual reality applications. *Journal of Visual Communication and Image Representation* 70 (2020), 102802. https://doi.org/10.1016/j.jvcir.2020.102802

[29] Junichi Yamaoka and Yasuaki Kakehi. 2016. MiragePrinter: interactive fabrication on a 3D printer with a mid-air display. In *ACM SIGGRAPH 2016 Studio (SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/2929484.2929489

[30] Yue Zhang and Fengzhou Fang. 2019. Development of planar diffractive waveguides in optical see-through head-mounted displays. *Precision Engineering* 60 (2019), 482–496. https://doi.org/10.1016/j.precisioneng.2019.09.009

[31] Fabio Zünd, Steven Poulakos, Mubbasir Kapadia, and Robert W. Sumner. 2017. Story Version Control and Graphical Visualization for Collaborative Story Authoring. In *Proceedings of the 14th European Conference on Visual Media Production (CVMP 2017)* (London, United Kingdom) *(CVMP 2017)*. Association for Computing Machinery, New York, NY, USA, Article 10, 10 pages. https://doi.org/10.1145/3150165.3150175