



3D Remote Assistance for NDT Inspections

Jörg Rehbein¹ · Sebastian-Johannes Lorenz¹ · Jens Holtmannspötter¹ · Bernd Valeske²

Received: 26 July 2023 / Accepted: 23 October 2023 / Published online: 22 December 2023
© The Author(s) 2023

Abstract

In this work, we present a system architecture that is especially designed for remote ultrasound testing inspections (UT). The system itself is realized using a real-time session service and a web service based on a REST API (REST: Representational State Transfer; API: application programming interface). This web service is used to store data persistently, e.g., sample geometries, raw nondestructive testing (NDT) data and derived inspection results in a shared dataspace. In the current development state, the results consist of textures mapped onto the sample geometry. This approach allows us to display the UT results directly on the real sample using mixed reality technologies. We also implemented a feature to assist the inspector remotely by making use of the availability of this digital representation. Hence, it is necessary to share additional data like the current UT-signal, temporary position marks, user and device positions, etc. between the different participants. A real-time distribution of this highly dynamic data is required to create an effective assistance environment. Therefore, a separate session service is used. The inspection data generated in this temporary session can also be transferred to the afore mentioned dataspace to be saved persistently. The system features mixed reality visualization for the inspector and optionally a virtual reality or a 3D Desktop environment for one or more remote assistants.

Keywords Non-destructive testing (NDT) · NDE 4.0 · Ultrasonic testing (UT) · Augmented/virtual/mixed reality · Real-time visualization · Digital twin · Remote assistance

1 Introduction

Ultrasound testing is a common method to ensure safety and reliability in aeronautics. Many research projects aim to develop new concepts and processes for NDT by integrating new technologies [1–5]. In our previous publication [6], we presented a technology demonstrator that can create a link between UT data and the digital representation of an inspected object. Using this connection, we achieved a real-time visualization of the NDT data directly on the sample with mixed reality systems.

A time-consuming part of NDT processes is creating necessary documentation of the inspection results. A classical documentation (e.g., drawing on the aircraft structure, scanning to PDF) will become obsolete, if all the recorded data is permanently saved in a digital form. The data generated

by our technology demonstrator is suitable to be saved in a cloud-like environment. The overall goal is to implement digital twin concepts into a system, capable to store all the generated NDT data.

Another problem is the requirement of experienced inspection staff. To be able to concentrate human resources and therefore to reduce time consuming travels and financial expenses, we are presenting a concept for a location-independent UT remote assistance system.

This approach requires the system to be extended with two main capabilities: Firstly, the system must be able to permanently store inspection raw data and results in a cloud-like environment. Secondly, the system must be able to distribute the whole inspection in real-time over the internet to distant locations.

All these approaches including digital twins, digital process transformation and integration of new technologies are part of fourth revolution of nondestructive evaluation [7, 8]. With the presented work we show how a shared dataspace may be created and contribute to a more efficient workflow. As the creation of a digital data ecosystem is a proposed goal

✉ Jörg Rehbein
JoergRehbein@bundeswehr.org

¹ Wehrwissenschaftliches Institut Für Werk- Und Betriebsstoffe (WIWeB), Institutsweg 1, 85435 Erding, Germany

² Fraunhofer-Institut Für Zerstörungsfreie Prüfverfahren IZFP, Campus E3 1, 66123 Saarbrücken, Germany

of the concept would we like to see our approach as a step towards NDE 4.0.

2 Related Research

The technical background and results presented in this work are based on our previous publication on 3D-Visualization of Ultrasonic NDT Data Using Mixed Reality [6]. In this paper, we described a technological demonstrator that is capable to visualize ultrasonic NDT data in real-time in a mixed reality environment [9]. The UT signal is processed and color coded in a first step. In a second step, the UT data is transmitted to a game engine, where the signal is drawn into the texture of a virtual 3D model of the inspected object. In order to track the spatial position and rotation of the UT sensor, we used an optical tracking system [10]. Since the recorded UT data is permanently linked to its correct three-dimensional position on the 3D model, this procedure may be suitable for digital twin applications. From our experience, it is also extremely useful, as it provides immediate feedback to the inspector and reduces the amount of manual documentation of the inspection.

Current assistance systems for NDT applications can be divided into two different approaches helping the inspector. The first method consists of technical systems that provide additional information (e.g. visual, acoustic) with the user in a direct feedback loop. The content is usually made of data coming from sensors, spatial positions, device settings, etc. The presented demonstrator in our previous publication can be categorized as such a system. Another example for a system that keeps the inspector in a feedback loop is the SmartInspect system developed by Fraunhofer IZFP [11–13].

The second method, that can be used to provide assistance for the inspector are remote systems, that allow a distant human expert to remotely join the inspection session. This work focuses on the second method. At the same time, it is built on visual technology developed for the first method.

Meyendorf et al. proposed a first approach for a remote assisted NDE inspection in 2017 [14]. They use common video conferencing technologies like Skype for transferring the desktop from the inspector's location to the remote assistant's location.

Another approach is presented by Westerkamp et al. [15]. They developed an online maintenance assistance system (OMA), that can be used for various applications including UT. They extended traditional video conference systems with features that are needed for remote maintenance systems like screenshot creation that can be saved in a shared workspace. From our understanding, the NDT systems are also integrated making use of video streams to provide compatibility to various NDT systems.

What known systems currently do not provide is an individually tailored remote assistance system for each NDT technique. All the UT system data is currently transferred using video streams. Three-dimensional information is not shared at all. Thus, the reference to a virtual representation of the real object does not exist.

The UK National Aerospace NDT Board has released guidelines and recommendations for assisted remote inspection systems [16]. They generally distinguish between synchronous and asynchronous inspections. In asynchronous inspections, the final evaluation of the NDT data and contextualization does not happen at the same time, the data is created. In contrast, synchronous evaluation takes place immediately. They propose three user roles: the operator, the verifier and the inspector, where the two last ones can work asynchronously. They recommend well trained staff for the remote NDT process, as well as written work instructions for remote NDT processes including an adequate risk assessment.

3 System Architecture

The presented work in this paper describes the architectural extensions and developed features for the existing system. The extensions can be split up in two interacting parts:

As proposed in the outlook of our previous paper, the first part describes how NDT data recorded by the system can be stored permanently in a database. This database is accessible from a network, so that it can be made ubiquitously available. Previously, it was only possible to save the recorded data locally.

The second part describes how real-time data of an ongoing inspection is distributed over the internet. This is necessary to realize the new real-time assistance function of the technology demonstrator.

3.1 Shared Dataspace

In order to save meta data of 3D geometries, NDT raw data and the NDT results, a SQL database has been created. The actual binary data is saved outside of the database on the hard drive of the server. The meta data of each file type have 1:n relations as described: NDT measurements are referenced to a 3D geometry file and as a NDT result is derived from a measurement. Each entry it is also referenced to the specific file. Additional information is saved to the data, e.g., the type of the NDT result. This information indicates how to process the linked binary file – similar to a file extension. These data fields also allow us to implement further NDT modalities or result types in the future.

Although we are aware of DICONDE [17, 18], and the efforts of the community to establish it as a common database

Fig. 1 Database structure

digital_twins		ndt_measurements		ndt_results	
field name	type	field name	type	field name	type
id	varchar(128)	id	varchar(128)	id	varchar(128)
name	varchar(128)	digital_twin_id	varchar(128)	ndt_measurement_id	varchar(128)
3d_model_filename	varchar(128)	name	varchar(128)	name	varchar(128)
3d_model_filetype	enum('OBJ')	datetime	datetime	date	datetime
		type	enum('UT','EC')	type	enum('UNWRAPPED_TEXTURE')
		filename	varchar(16)	filename	varchar(16)

Table 1 Example endpoints of the REST API to manage the 3D models and the corresponding meta data

HTTP Method	API Endpoint	Function
GET	digitaltwins/	Get a list of all data sets
GET	/digitaltwins/{id}	Provide standard CRUD functionalities
POST	/digitaltwins	
PUT	/digitaltwins/{id}	
DELETE	/digitaltwins/{id}	
POST	/digitaltwins/{id}/3d_model/obj	Provide functionalities to upload 3D model data; Currently only OBJ files are supported
DELETE	/digitaltwins/{id}/3d_model	
GET	/digitaltwins/{id}/3d_model	

The endpoints for NDT measurements and NDT results are created similarly

standard, we have chosen a simpler but more adaptable solution for our project. For a real inspection much more meta information (e.g., calibration data, qualification of the inspector, device types) needs to be stored in order to ensure the quality and reliability of the results. For our testing environment, we decided to omit this additional information. Figure 1 shows the information that is stored in the database.

In the current implementation, the 3D geometry of the specimen is saved as OBJ file. The OBJ file format was chosen as it can store necessary UV data for our texture mapping method [6]. The NDT measurements are saved in a custom file format. In this binary format, all A-scans are stored together with their corresponding spatial coordinates (position and orientation for each measurement point). The 'filename' database fields refer to the binary files. These measurement points are referenced to the origin of the 3D model of the investigated object. As the testing setup is only used for verification, equipment information is not stored. For NDT results, we are currently using one method so save the measurement: In case of the *UNWRAPPED_TEXTURE* type for NDT results, the saved file is a PNG image. A file format providing a transparency channel is required, as the texture would also occlude not inspected areas on the inspected object.

The database can be accessed through a HTTP REST-API [19]. Those APIs are often used to access databases or other resources in the internet. Our API implementation provides creating, reading, changing/updating and deleting (CRUD) functionalities for each of the three file types. Additionally for each file entry, the binary data on the file system (OBJ,

raw measurement data and PNG) can be stored and accessed through the web API.

In order to access the database, a client library has been created. This library allows any C# application to interact with the remote database, including the game engine running the virtual and augmented reality (AR/VR) applications. We chose C# as it is the main scripting language for the used game engine. In the current state, the game engine is able to load the 3D geometry from the server that corresponds to the real-world object to be investigated. The NDT measurements and NDT results are created as described in [6]. After successfully completing the NDT inspection and creating the files locally, the game engine is able to send the data to the web service to store it permanently. An excerpt of all API endpoints can be found in Table 1.

We designed this API in order to show an example of how our generated NDT data could be integrated in a Digital Twin environment. Thus, the endpoints are all including the key "digitaltwin", although we did not implement a whole Digital Twin infrastructure.

RESTful APIs are very common in today's internet applications. Even Microsoft's Azure platform does provide a REST API to connect to their Digital Twin environment [20]. Thus, our implementation is also based on this approach to show the general feasibility and compatibility with latest standards.

3.2 Real-Time Sessions for Remote Assistance

Mixed reality technologies improve the inspection process in various ways. In the current demonstrator augmented reality is part of the real-time feedback loop for the inspector. For the remote assistant, virtual reality can be used to provide a better impression of the distant scene. In order to connect these two technologies to a distributed system, it is necessary to implement a real-time service.

Although the 3D geometry data and the gathered NDT data can be distributed using the technology described in the previous section, this architecture is not suitable to share real-time data during an inspection. As this is a necessary functionality for real-time assistance, a separate service has been created.

The real-time service is based on a standard TCP/IP connection. When a client program connects to the server, it firstly tells the server its role and what data it expects to receive. Roles that can be chosen are: inspector, assistant and device. There can only be one inspector in a single session, but as many assistants as necessary. The device role is used for equipment that does not share appropriate position information, e.g., cameras or the UT frontend software. Available transmission data to be selected is:

3.2.1 Participant Positions

This setting defines if the server should send participant position updates to the client. These updates also include the view direction. This may not be suitable for clients with the user role 'device'. An updated user list that is shared on connection or disconnection events, is always being shared among the clients.

3.2.2 Spatially Tracked UT Stream Packet

The client program that is running on the inspector's pc concurrently merges real-time UT sensor data with its spatial position. The spatial information is drawn from the used tracking system. The whole corresponding UT signal (A-scan) is added. This also includes defined time gates, activated triggers, coupling detection information, assigned colors and the UT device settings that were used to create the UT signal (for further information, see [6]). This packet data can only be sent to the server from the client having the inspector role.

3.2.3 List of Position Marks

In order to facilitate communication between the inspector and the remote assistants, we implemented a rudimentary feature to share position marks that can be placed in 3D space. Every client can add position marks and to clear the shared list

of position marks. Position marks are currently represented by three-dimensional arrows.

3.2.4 UT Settings

A set of settings for the UT hardware is shared over the service as well. Every client application has the permission to update the settings and thus change values. This provides the possibility for the assistants to remotely adjust the UT settings for the on-site inspector and increases their capability of control. If changed, the UT settings are automatically applied by the UT hardware. They include signal recording rate, the signal offset, hard- and software preamplification, the signal transformation and the time gates. The set of settings is copied to every recorded UT signal (see 3.2.2).

3.2.5 Recorded UT spots

Generating the C- or D-scan images works as follows: The signal processing procedure assigns a false color to the signal. The position of the UT sensor is mapped into UV texture coordinates. This information is used to draw a colored spot into the texture of the 3D model.

This information containing the generated C/D-scan needs to be shared across the clients. In order to reduce the required bandwidth, not the whole texture or UT signal is sent to the server. A reduced subset of information is sent to allow the reconstruction of the texture on the remote assistant clients. The sent information comprises of UV texture coordinates and the signal color and a unique identifier. This information is sufficient to accomplish complete client-side false color image reconstruction.

3.2.6 Inspector View and Overview Camera Stream

In complex cases a complete virtual representation of the NDT inspection may not be sufficient for a successful remote assistance. Therefore, to further facilitate the remote assistance intend, we added the functionality to distribute two independent video streams from the inspector to the clients. The first is meant to be the inspector view and the second one can be used by an individually placed overview camera. This camera may record the whole scene, consisting of the area to be inspected and the inspector him or herself. As AR and MR headsets usually have an integrated camera available, we intend to use this built-in device for the inspector view camera. Often, this camera's video stream includes the virtual objects displayed on the AR/MR displays making it suitable for us to be streamed over our NDT service. Initially we implemented an MJPEG-stream, but in order to reduce the required bandwidth, we are currently using a H.264-like video encoding technology, however, almost any video streaming encoding that can be split up into chunks can

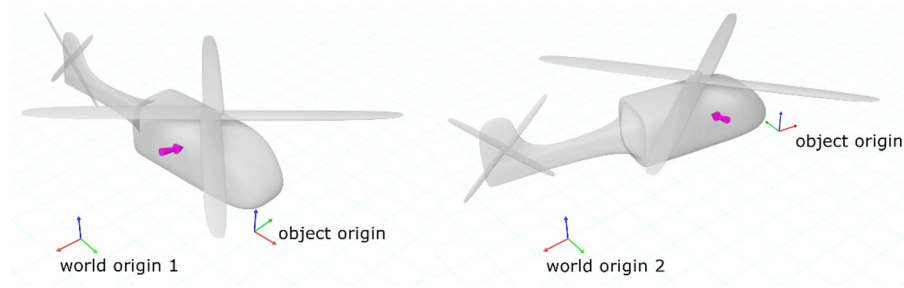


Fig. 2 Location information is always shared relatively to the 3D model coordinate system. Because of that, the purple marking always remains at the correct position on the 3D

object, independent of its position relative to the corresponding world origin. The object coordinate system can be freely moved within the working space

be used in the future to reduce required bandwidth. We do not transfer a separate video stream for the A-scan as mentioned in [14], as we share the original UT signal in a digital form.

3.2.7 Further Information About the Real-Time Service

There is additional information that is always being shared disregarding the role of the client program or the selected data. As mentioned above, a list of all connected clients is always shared between the clients. Additionally, the identifiers (IDs) of the investigated part (3D model), NDT measurement and NDT result are always shared over the network. These identifiers refer to the IDs of the shared dataspace web service. This gives every client application the ability to download data from the web service individually and reduces the load on the real-time service. Multiple servers providing the mirrored persistent data as common for cloud services may be used as well [21].

All clients may use different tracking systems - or in case of a desktop application - do not even have one. Therefore, they have different coordinate systems and thus, different world origins. This makes it difficult to individually share correct position information. To address this problem, all shared spatial information is shared relatively to the investigated object's origin Fig. 2 illustrates how location information is shared among the participants.

The server caches information sent to it. This saves bandwidth and reduces the need to wait for the next update from the other clients. This includes all current positional information, relevant identifiers, recorded UT spots and positions marks. This data is automatically sent to a new connected client application.

When the real-time assisted UT inspection has successfully been completed, the inspector can upload the raw measurement data and the result data to persistently store it on the web service. This result data can later be examined - also in an AR/VR multi user session for further evaluation.

In summary, every client device is able to download persistent data as well as participate in the real-time data exchange process.

Figure 3 shows the principal system architecture.

We did not implement an audio connection as this service can act relatively independent of our system and does not have a direct influence on the shared data. We are currently using common VoIP services. Nevertheless, the occurring data traffic will be considered for bandwidth measurements.

4 Security

Concerning security, we prepared the system for modern data protection methods. For the shared dataspace web service infrastructure, all the data traffic is running over a HTTP connection. This means it can easily be extended using HTTPS providing data encryption functionality. We did not include authentication as we are only running this service in a local network. Upgrading it with security functionalities would not have had any benefits for the project. Real-world implementations may also make use of blockchain technologies with the aim to prevent tampering of the stored data.

The real-time service is based on a TLSv1.2 (Transport Layer Security) transport stream. Thus, all exchanged data is automatically encrypted by default. Additionally, the protocol provides a data field for authentication to prevent unwanted access: "authKey". For every data packet the server or client receives, the authKey is checked. The packet will be discarded if the authKey does not match the predefined one. As the system is designed as a cooperative system, all participants share the same rights. Thus, all clients share the same authKey. This data field may later also be used for a security token to provide individual authentication and permission management.

Fig. 3 Principal system architecture: The link between both services is created by providing a shared list of relevant IDs of the Web Service. Every Client may individually download necessary data from the Web Service

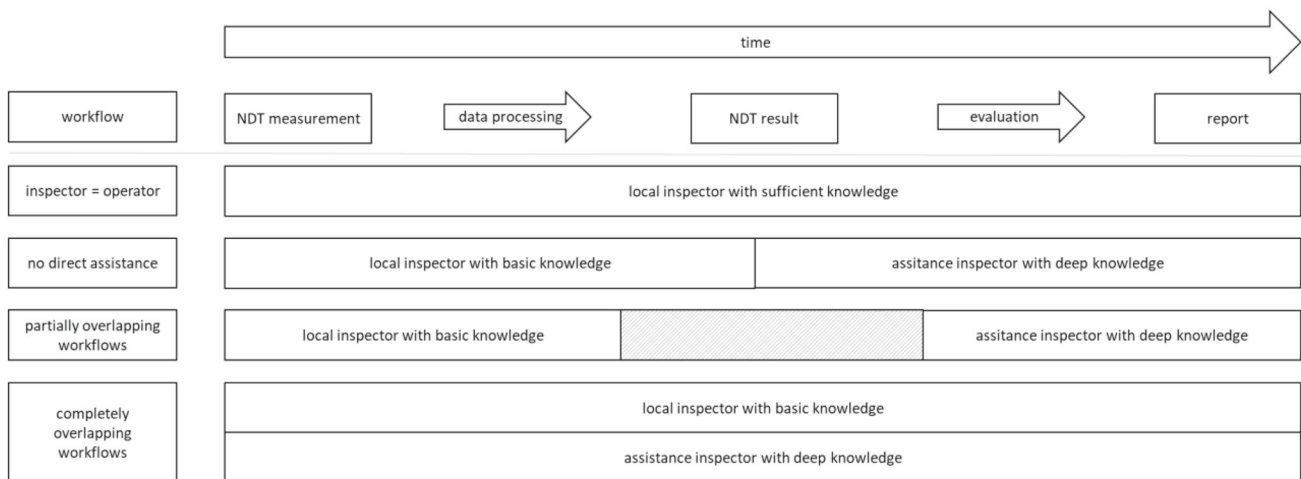
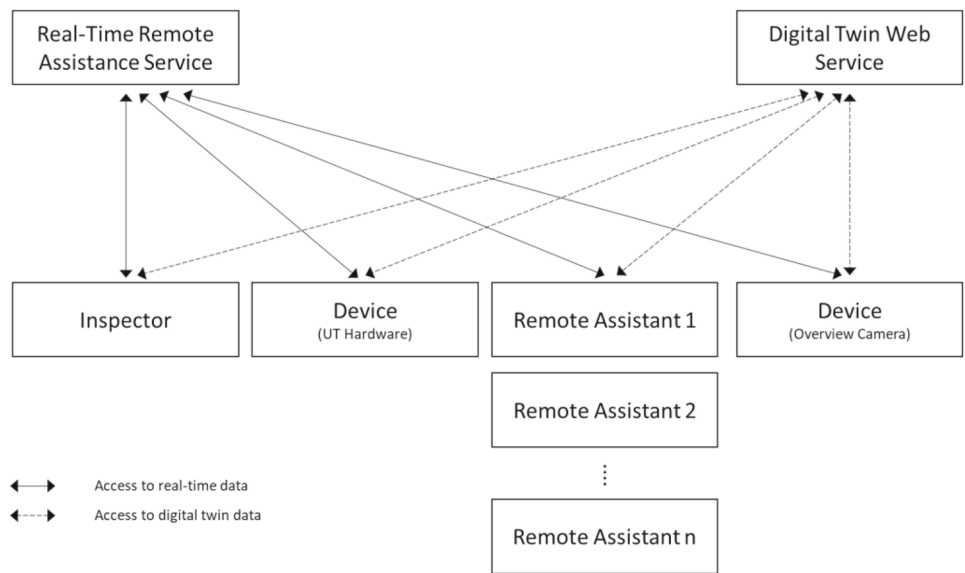


Fig. 4 Proposed workflows for the described system

5 Anticipated Workflow

The system basically provides a range of possible levels of assistance integration. These are schematically shown in Fig. 4. All proposed assistance levels are based on the same workflow. It begins with the traditional NDT process in which the raw data is being generated. This raw measurement data is processed and displayed on the 3D model. In our approach, the data processing happens simultaneously to the recording (For further information on that process, see [6]). The final report may later be derived from the NDT result. Each resulting set of NDT data can be uploaded, as well as downloaded from the shared dataspac.

In the first assistance level, there is only one inspector active, having sufficient knowledge to perform the whole NDE inspection independently. He operates the system on

his own and creates NDT measurement raw data as well as the NDT result. This is the basic workflow as described in [6] without the need of further assistance for the inspector. The NDT measurement raw data and the NDT result is finally transferred to the persistent web service. Therefore, it is also possible to time-independently review the collected NDT data.

The second assistance level lets the local inspector with basic knowledge still record all the NDT data without assistance. The second experienced distant inspector becomes active when all required data has already been collected. He is able to interpret the automatically generated NDT result. If necessary, he would also be able to derive a new NDT result from the raw data in the NDT measurement with changed settings, e.g. time gates. As described, this procedure consists of two sequential working steps of the two inspectors.

Table 2 Developed client applications

Location	Client application role	Application type	Used hardware
Inspection site	Inspector	3D mixed reality	Optical tracking system MR glasses Tracked UT sensor Pointing device UT sample
	Inspector UT frontend	2D desktop	UT-USB interface
	Camera client	2D desktop	Webcam
Remote assistant site	Remote assistant with desktop client	3D desktop	–
	Remote assistant with VR headset	3D virtual reality	VR headset

This assistance level would be an option if the local inspector lacks knowledge mainly in details of the inspected object. The shaded area in assistance level 2 indicates shows the flexible point of time for the handover.

The third assistance level allows both, the local inspector and the remote assistance inspector, to work simultaneously on the NDE data acquisition. In this setting, the remote assistant can actively vary parameters and directly change the data recording settings. The following steps in the workflow can also be taken together. The third level provides a full remote assistance service. It is suitable for situations, where the local inspector additionally to the second level also lacks in knowledge about the NDT process itself.

To achieve the desired functionalities, we developed several client applications, which are described in Table 2.

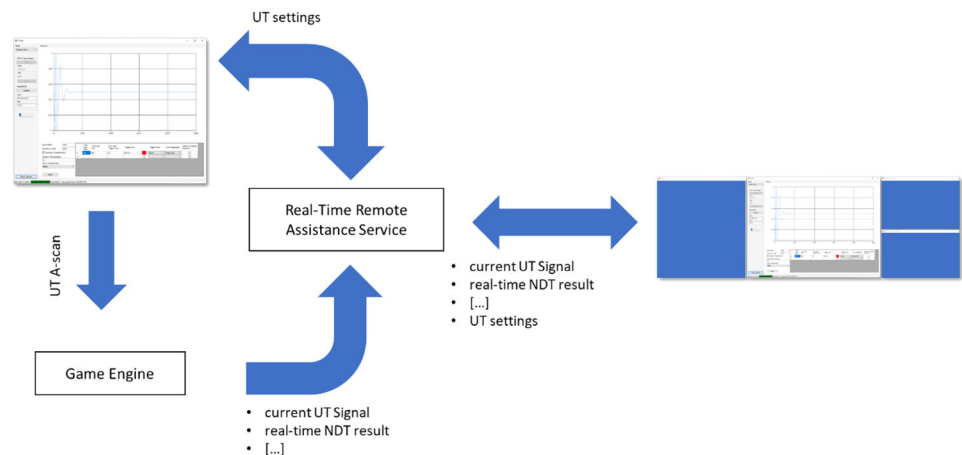
The inspector application and the UT frontend are running the necessary software parts to perform the mixed reality NDT inspection. In order to allow the remote assistant to

change parameters, the UT frontend has a separate connection to the real-time service. The inspector himself can do this via directly adjusting parameters in the frontend software or in the game engine. Optionally, there can be added two camera clients on the inspection site to provide the video streaming functionality. Typically, these are two independent applications. The network architecture for the real-time service is shown in Fig. 5.

The remote assistant may either use the 3D Desktop client to analyze the geometrical model or the virtual reality client to view it in XR mode.

The only current limitation of the system is, that due to development capacities, we were not able to implement the display of encoded video streams in the game engine. We assume, a company with sufficient development staff will be able to get this implemented. For our future test environment including the potential user group, we will fall back to a MJPEG-like video transfer.

Fig. 5 Implementation of the feedback channel for the UT parameters; left: inspector applications; right: remote assistant application



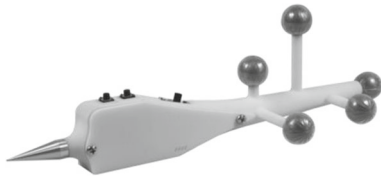


Fig. 6 Pointing device that is currently used to place marks in the working space. It is connected to the game engine using a WIFI network connection

6 Results

We have extended our system demonstrator in a way that allows us to assist a UT inspection remotely. All necessary data is being shared using a two-server architecture. The first server holds static data and the second one distributes dynamical information.

At the time of writing this publication we have implemented three types of applications. The first implementation uses a mixed reality headset [9] and an optical tracking system [10]. This application is meant to run on the inspector's pc to record NDT data. Additional input hardware to place marks has been designed (see Fig. 6). The first remote assistant application is created for remote assistance using virtual reality. We successfully tested this client application with an HTC VIVE headset [22]. We did also implement a standard Desktop application, showing the whole scene from a first-person view. In both remote assistant applications, the A-scan is currently displayed on an individually placeable user interface (UI) canvas. Audio communication is currently realized using a third-party voice-over-IP (VoIP) application.

Figure 7 shows two screenshots of an ongoing inspection. On the left-hand side (inspector), the figure shows the colored AR overlay. Additionally, an A-scan is shown. On the right-hand side (remote assistant), the whole scene is rendered in a VR headset, including the virtual test object and the same A-scan that is shown on the inspector's side. The current positions of the session participants are indicated by a yellow avatar. In both screenshots, a marking arrow is shown that has been placed by the remote assistant. The inspected test object

is the same helicopter tail structure as shown in our precedent publication [6].

As all spatial information is transmitted relative to the 3D model's origin, every client can locally place it regarding their needs. This may be necessary, e.g., if the remote assistance needs to be given from a confined space. The inspector can regardlessly align the 3D model to the real-world object. All marks, participant locations etc. stay relative to the 3D model.

Every participant can adjust the UT settings as they are distributed over the network.

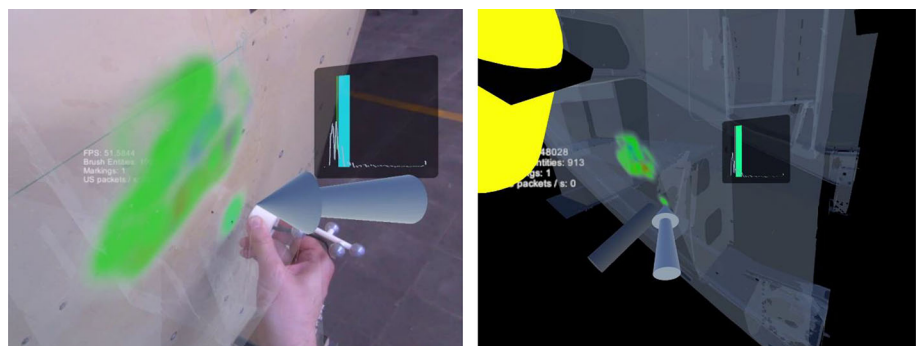
For the technology demonstrator we used the mentioned hardware (HMDs and tracking system). Generally, it is possible to replace any of these components with minor adjustments of the system.

6.1 Bandwidth Analysis for the Remote Assistance Functionality

In order to get an impression of the required bandwidth of the system and to identify the high data consuming parts of the network protocol, a detailed analysis was performed. For a simple test scene, all transferred data has been recorded including the corresponding time stamp and saved to disk on the server. This provides the possibility to analyze the data later. It would additionally be possible to re-feed it into the system. Figure 8 gives an overview of the recorded scene. In this experiment, a carbon fiber reinforced plastics (CFRP) plate was inspected. The plate is prepared with parallel teflon stripes to simulate delaminations. It is mounted in a trackable frame. Figure 9 shows the 3D model used for this scene, including the projected texture (see [6] for detailed information). The texture is colored in blue where the teflon stripes are located.

The required Bandwidth has been analyzed in two steps. At first, the occurrences of each message type were counted (Fig. 10) and plotted over time (Fig. 11). In a second step, the actual used bandwidth for each message type has been calculated (Fig. 12) and plotted over time (Fig. 13). The following charts and graphs show the results of the procedure.

Fig. 7 Screenshots of an ongoing remote assisted inspection



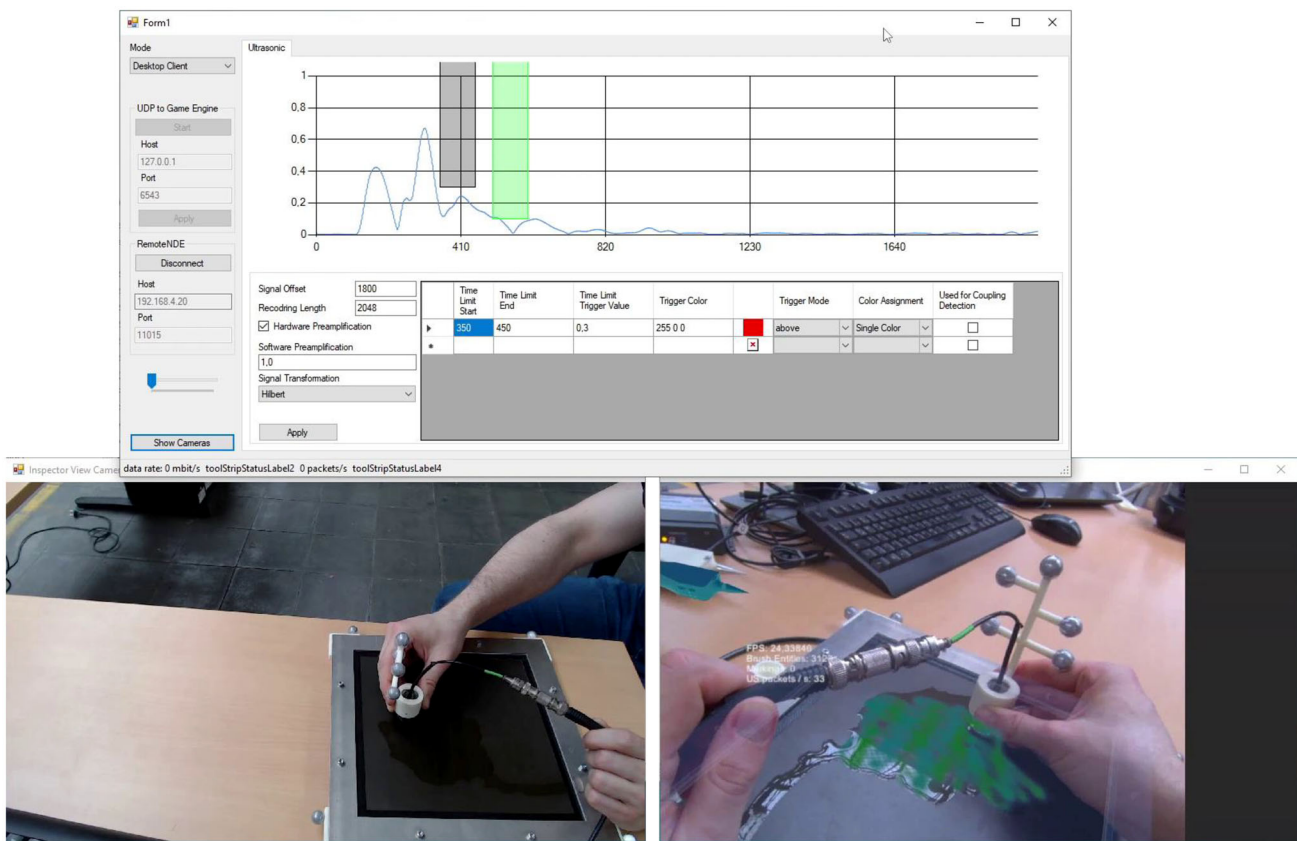


Fig. 8 Overview of the recorded Scene on remote assistant site (desktop mode) – The Main Window is showing the transmitted A-scan. The two bottom windows are showing the overview camera (left) and the inspector view camera (right)

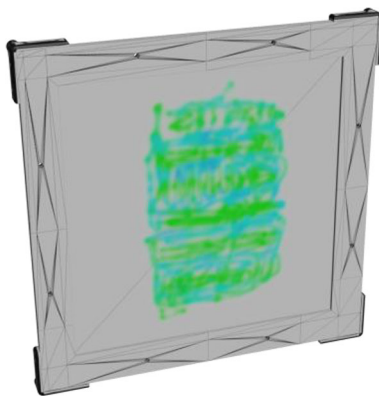


Fig. 9 Underlying 3D model and projected texture

The video encoder settings for both video streams were set to the following configuration: resolution: 1920×1080 pixels; codec: $\times 264$ (ultrafast, zerolatency); keyframe interval: 100; bitrate 3 Mbit.

Although more message types are existing, for the analysis, just a set of five message types has been considered. These are `UserPositionUpdate`, `CompleteBrushEntitiesSnapshot`, `CurrentSpatiallyTrackedUSStreamPacketUpdate`, `InspectorViewCameraFrameData`, `OverviewCameraFrameData`. All other existing messages were not taken into account, as they are only sent on very specific events and have no significant data size. These are e.g.: `Client/Server Hello/Goodbye` messages, `ReceiveOptionsUpdate`, `UserlistUpdate`, `MarkingListUpdate`, `ID` updates, `UTSettingsUpdate`.

The prefix in the message type stands for the transmission direction of the message. C2S means client to server and S2C server to client respectively. X2X means, this message type is used for both directions. Typically, the message is just forwarded to the other clients based on their receive option settings. For this analysis, the prefix has no relevance, as all recorded messages result of incoming traffic.

The traffic analysis clearly shows, that the video streams are taking the largest part of the transferred data (see Fig. 13). Both video streams take about two thirds of the used bandwidth. Replacing them by AR/VR technologies may lead to a significant improvement in terms of required bandwidth. The

Fig. 10 Bandwidth analysis - Message occurrences

Message Occurrences (total 6818)

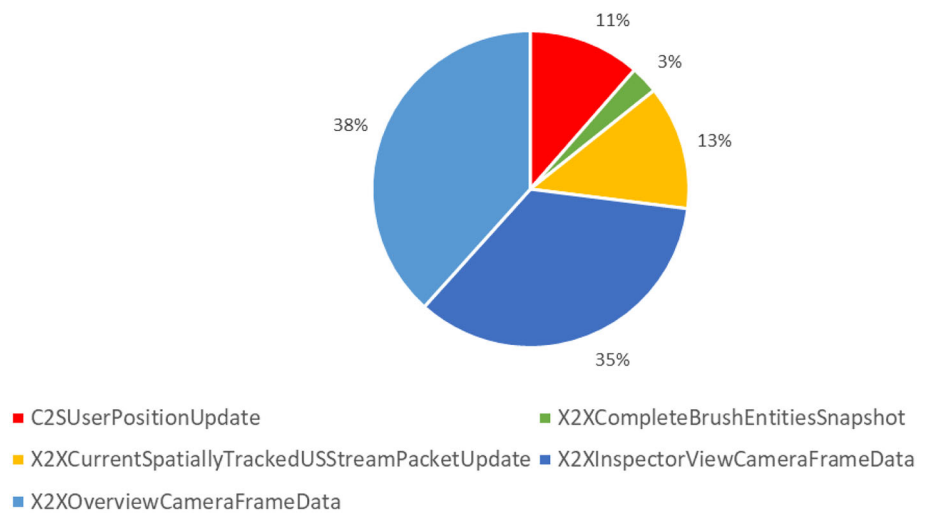
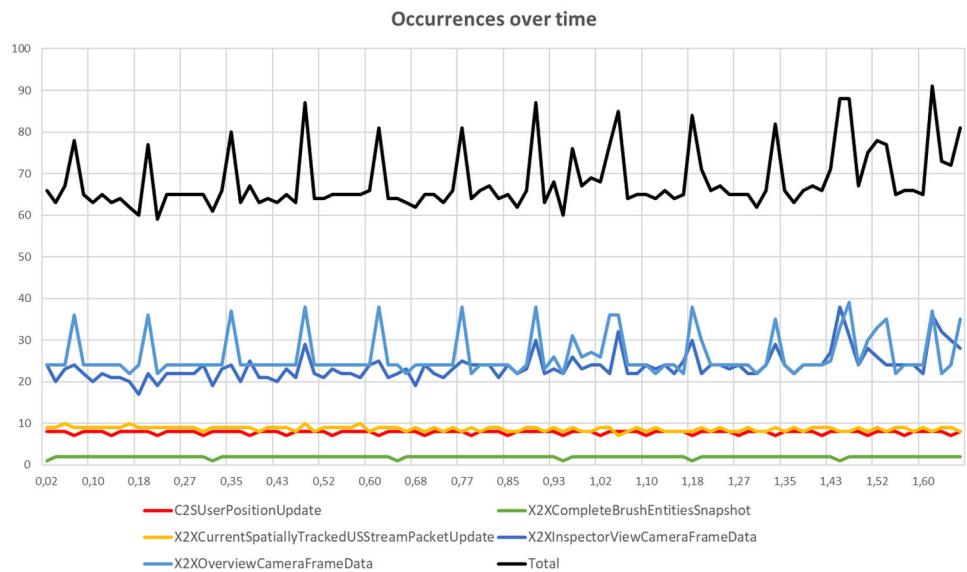


Fig. 11 Bandwidth analysis - Message occurrences over time



required bandwidth for the video streams also depends on the image complexity. Because of this, the overview camera consumes slightly more bandwidth than the inspector view camera, which is constantly missing out a grey bar on the right-hand side, because of the aspect ratio not being 16:9.

The user position updates do often occur, but they do not have a significant influence on the bandwidth and the system performance. About 11% of all received messages are position updates, but the total amount of all data is just about 0.08% overall.

Figure 11 displaying message occurrences over time shows, that the messages are sent at relatively constant rates, except for two message types: The first exception is caused by the key frames produced by the video encoders.

This can be seen as peaks in the graph. The other exception is the BrushEntitiesSnapshot message type, containing all necessary information to enable the receiving client the reconstruction of the texture information. Currently, the complete information is bundled into one message and sent to the server. Thus, the message size is constantly growing. At the end of the experiment, it contained more than 3254 colored spots representing the texture information. In a future step, the server will be enabled to cache texture information, that has already been transmitted.

The used bandwidth never exceeded 700 KB/s (5.600 kbps), which should be low enough to be handled by most internet connections. We do not expect it to significantly increase depending on the number of users in a session. The only part in the network that is required to have a greater

Fig. 12 Bandwidth analysis - Transferred data

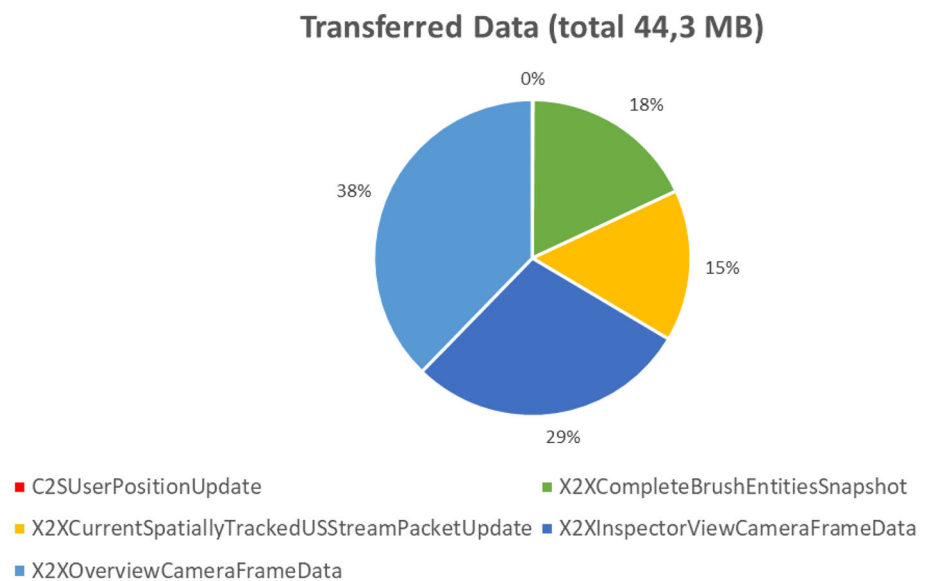
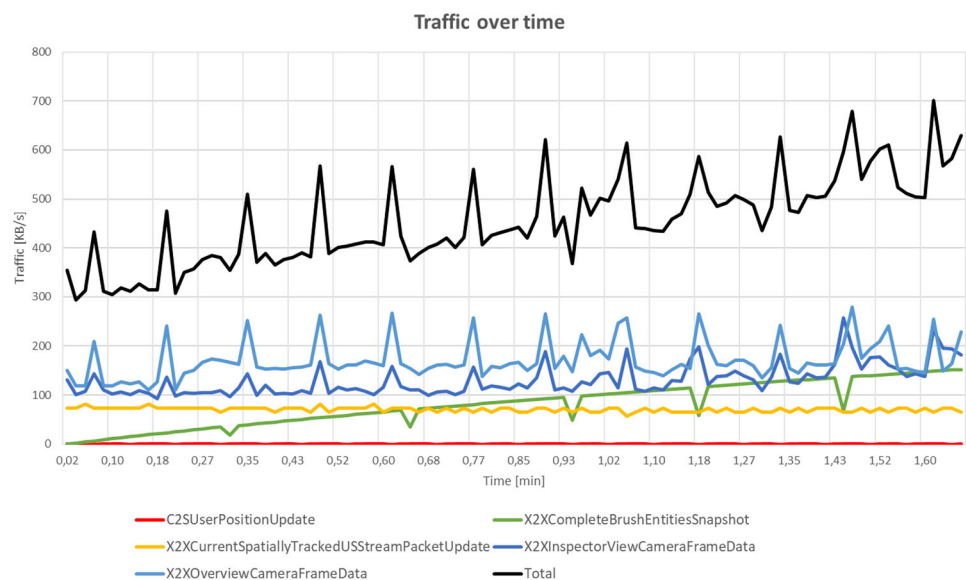


Fig. 13 Bandwidth analysis - Traffic over time



bandwidth is the server as it distributes all the data to the remote assistants. We have successfully tested the system with two remote assistants over an internet connection without noticeable limitations compared to local tests. The server was located close to Berlin, whereas the inspector and remote assistants were in Erding, close to Munich. Additionally, it should be mentioned, that all analyzed data is net data, meaning that the originated traffic may be a bit higher due to encryption overhead. We do currently not compress the transmitted UT data. For a first step towards compression, we were able to reduce the transmitted UT stream size by 8% making use of the deflate algorithm [23].

7 Discussion

The described system architecture allows the realization of NDT remote assistance across distant locations. Current remote assistance technologies are based on video streams [14, 15]. Our implementation makes use of AR/VR technologies and the integration into a shared dataspace. This offers a lot of flexibility in terms of the type of application that is running on the assistant's pc and the evaluation process.

The remote assistant can choose what type of application fits the best to the particular inspection task. He can stay with the current state by just using video streams. If he wants to get a better imagination of the whole scene, he may use a virtual reality headset. If the assistant needs to do adjustments to the UT hardware, all basic settings can be accessed remotely.

As the real-time service has a direct reference to the 3D model, all gathered data is implicitly suitable to be saved in the web service. Additional flexibility is arising, as data analysis does not need to be done at the same location, consecutively or even by the same person. The data may be recorded by a level 1 inspector and saved to the web service as raw data. This raw data can later be downloaded and reviewed by an experienced inspector. He then creates the final NDT result file and uploads it to the shared dataspace. It may also be possible to completely outsource the final task of evaluation to further reduce costs of having company internal level 3 inspectors. Those external experts may also assist during the actual inspection process.

8 Outlook and Future Work

Currently there is no user interface optimized for adjusting NDT settings from within a virtual or mixed reality application. Functionalities like mode switching, starting and stopping of the data recording is currently realized using keyboard shortcuts. Basic UT settings can be made using a standard 2D GUI (graphical user interface). One of the next work packages will include realizing a suitable user interface for each application type: the inspector, the desktop remote assistant and the VR-based remote assistant clients.

The system is currently also limited to ultrasound testing, but regarding the database structure it should be simple to add further modalities do the web service. The real-time service instead, has specialized functionalities that make it more difficult to implement other NDT modalities according to their requirements. The system presumably needs to be changed to add further NDT modalities.

The next work package we will address is to test the different application types with experienced NDT inspection staff of the German armed forces. The purpose will be to investigate the potential benefits of the remote assistance system. We plan to compare the state of the art, video stream based remote assistance with our 3D remote assistance system.

The network analysis revealed a possible bottleneck for large scale measurements: As currently the complete texture information is transferred, the used bandwidth is concurrently growing. In order to only having to send new recordings, this will be fixed with implementing server-side caching.

Acknowledgements Not applicable

Author Contributions JR and SL wrote the main manuscript text. JR prepared and created the figures. The manuscript has been reviewed by JH and BV.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors have no relevant financial interests to disclose.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

Ethical Approval All ethical COPE guidelines have been considered.

Consent to Participate All authors consent to participate in this work.

Consent for Publication All authors consent to publish this work.

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

References

1. Meyer, J., Rehbein, J., de Freese, J., et al.: Visualisation of ultrasonic testing data using augmented reality. 7th International Symposium on NDT in Aerospace (2015)
2. Deppe, R., Nemitz, O., Herder, J.: Augmented reality for supporting manual non-destructive ultrasonic testing of metal pipes and plates. *Virtuelle und Erweiterte Realität – 15. Workshop der GI-Fachgruppe VR/AR:45–52*. (2018). <https://doi.org/10.2370/9783844062151>
3. Fraunhofer, I.Z.F.P.: 3D-SmartInspect: intelligente Prüfüberwachung und -dokumentation durch optisches Trackingsystem für Ultraschall- und Wirbelstromprüfung. <https://www.izfp.fraunhofer.de/content/dam/izfp/de/documents/aktuelles/2017/3D-SmartInspect-de.pdf> (2017). Accessed 22 Sep 2021
4. Valeske, B., Osman, A., Römer, F., et al.: Next Generation NDE Sensor Systems as IIoT Elements of Industry 4.0. *Res. Nondestruct. Eval.* **31**, 340–369 (2020). <https://doi.org/10.1080/09349847.2020.1841862>
5. Prado, G., Silva, P., Simomura, F., et al.: SHM with augmented reality for aircraft maintenance. 31st Congress of the International Council of the Aeronautical Sciences (2018)
6. Rehbein, J., Lorenz, S.-J., Holtmannspötter, J., et al.: 3D-visualization of ultrasonic NDT data using mixed reality. *J. Nondestruct. Eval. Nondestruct. Eval.* (2022). <https://doi.org/10.1007/s10921-022-00860-7>
7. Vrana, J., Singh, R.: NDE 4.0-A design thinking perspective. *J. Nondestruct. Eval. Nondestruct. Eval.* **40**, 8 (2021). <https://doi.org/10.1007/s10921-020-00735-9>
8. Vrana, J.: The core of the fourth revolutions: industrial internet of things, digital twin, and cyber-physical loops. *J. Nondestruct. Eval. Nondestruct. Eval.* (2021). <https://doi.org/10.1007/s10921-021-00777-7>
9. Varjo, H.Q.: <https://varjo.com> (2021). Accessed 16 Sep 2021

10. Advanced Realtime Tracking GmbH & Co. KG: ARTTRACK5. <https://ar-tracking.com/en/product-program/arttrack5> (2021). Accessed 16 Sep 2021
11. Valeske, B., Lugin, S., Schwender, T.: IZFP-SmartInspect: Prüf-Assistenzsystem für die multi-modale Handprüfung im Zeichen der Digitalisierung – Möglichkeiten für ZfP 4.0 durch Echtzeit-Ergebnisvisualisierung und Augmented Reality. DGZfP DACH-Jahrestagung 2019, Friedrichshafen (2019)
12. Lugin, S., Osman, A., Valeske, B.: Verfahren und Anordnung zur überwachten, räumlich-aufgelösten Prüfung eines dreidimensionalen Objektes (DE 10 2015 211 025 A1 2016.12.22). <https://patents.google.com/patent/DE102015211025A1/de?qoq=DE102015211025A1>. (2015)
13. Fraunhofer IZFP: SmartInspect video presentation. https://www.izfp.fraunhofer.de/content/dam/izfp/en/videos/smart_inspection_solutions.mp4 (2019). Accessed 05 Jun 2023
14. Meyendorf, N., Schallert, R., Pal, S., et al.: Using Remote NDE, including External Experts in the Inspection Process, to Enhance Reliability and address Today's NDE Challenges. 7th European–American Workshop on Reliability of NDE (2017)
15. Westerkamp, C., Behrens, R., Kolesnikow, A., et al.: Online Maintenance Assistance (OMA) - remote control of service and lab experiments. In: International Conference on Remote Engineering and Virtual Instrumentation (REV), pp 353–356. (2014)
16. UK National Aerospace NDT Board: UK NANDTB POLICY ON REMOTE NDT (2022)
17. Martin, J., Leinenbach, F., Engert, D., et al.: DICONDE in der ZfP 2023
18. E07 Committee Practice for Digital Imaging and Communication in Nondestructive Evaluation (DICONDE)
19. Red Hat: What is a REST API? <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (2023). Accessed 12 Apr 2023
20. Microsoft: Azure Digital Twins APIs and SDKs - Azure digital twins. <https://learn.microsoft.com/en-us/azure/digital-twins/concepts-apis-sdks> (2022). Accessed 03 Nov 2022
21. Altomare, F.: What is a Content Delivery Network? CDN Explained. GlobalDots (2023)
22. HTC Corporation: VIVE Pro. <https://www.vive.com/de/product/vive-pro/> (2021). Accessed 16 Sep 2021
23. Deutsch, P.: DEFLATE Compressed Data Format Specification version 1.3. RFC Editor (1996)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.