



SINTEF

# Report

## InVizAR project

Augmented Reality visualization of invisible fields utilizing sensor data

### Author(s):

Costas Boletsis, Arne Lie, Ophelia Prillard, Karsten Husby, Jiaxin Li

### Report No:

2022:01481 - Unrestricted

### Client(s) (pos partner):

SINTEF Digital, Research Council of Norway

# Report

## InVizAR project

Augmented Reality visualization of invisible fields utilizing sensor data

**KEYWORDS**

augmented reality, data,  
sensors, visualization

**VERSION**

1.2

**DATE**

2023-01-16

**AUTHOR(S)**

Costas Boletsis, Arne Lie, Ophelia Prillard, Karsten Husby, Jiaxin Li

**CLIENT(S)**

SINTEF Digital, Research Council of Norway

**CLIENT'S REFERENCE**

N/A

**PROJECT NO.**

102026922-3

**NO. OF PAGES/APPENDICES**

27

**SUMMARY**

This is a report for the 2022 SEP Project InVizAR. InVizAR's main objective is to explore efficient and user-friendly AR visualizations of sensor data describing "invisible fields", like electromagnetic fields or radio signals. The focus here is on datasets with geospatial and temporal qualities. The ultimate goal of the project is to be able – through a use case – to design, develop, and present an AR visualization of invisible-fields' dataset that can be easily understood, interacted with, and enable users to reach to conclusions and make informed decisions. Moreover, a more practical goal of InVizAR is for the team involved in the project to get familiar with AR visualizations of sensor data coming from invisible fields.

**PREPARED BY**

Costas Boletsis

**SIGNATURE**

*Costas Boletsis*

Costas Boletsis (Jan 16, 2023 13:13 GMT+1)

**CHECKED BY**

Arne Lie

**SIGNATURE**

*Arne Lie*

Arne Lie (Jan 16, 2023 13:41 GMT+1)

**APPROVED BY**

Jan Håvard Skjetne

**SIGNATURE**

*Jan Håvard Skjetne*

**REPORT NO.**

2022:01481

**ISBN**

978-82-14-07971-5

**CLASSIFICATION**

Unrestricted

**CLASSIFICATION THIS PAGE**

Unrestricted

# Document history

---

VERSION	DATE	VERSION DESCRIPTION
1.1	2022-12-19	Version after editing and proofreading.
1.2	2023-01-16	Proofreading check.

---

# Table of contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Background .....	4
1.2	Project objective & motivation .....	4
1.3	Targeted use & visualization mode.....	5
1.4	Collaboration.....	7
1.5	Methodology.....	7
1.6	Functional requirements.....	8
<b>2</b>	<b>Use case &amp; dataset from sensor data .....</b>	<b>8</b>
2.1	Dataset description .....	8
2.2	Dataset syntax alternatives.....	10
<b>3</b>	<b>Dataset software .....</b>	<b>12</b>
3.1	Software documentation .....	12
3.2	Software inherited from ANDWIS.....	18
3.3	Further ideas for visualization .....	20
3.4	Git repository for making JSON files .....	21
<b>4</b>	<b>AR visualization prototype.....</b>	<b>22</b>
4.1	Design concept.....	22
4.2	Hardware and development .....	24
4.3	InVizAR prototype .....	25
4.4	Evaluation and conclusion .....	26

# 1 Introduction

## 1.1 Background

Devices like unmanned vehicles - maritime, ground, and aerial (drones) - and robots have a very important role in certain Internet-of-Things (IoT) settings, where they must collect sensor data remotely due to short range or directive signalling being used. The received signals are typically (invisible) electromagnetic fields carrying sensing data, such as Bluetooth or other radio signals, but can also be radar reflections, pure magnetic fields, or even camera imaging<sup>1</sup>. The sensor data need to be visualized based on their geospatial and time-based signatures. However, the fact that these data operate in four dimensions (longitude, latitude, elevation, and time) makes their visualization in 2D and 3D settings a challenging task<sup>2</sup>.

In our daily work, we often find it difficult to present the invisible data, such as electromagnetic or pure magnetic fields, to a broad audience, especially our clients, in an intuitive way so that they can understand the value of the data. For instance, localized diagnostic techniques, such as magnetic field non-destructive testing methodologies for structural monitoring of jacket platforms are very important for detecting early signs of deterioration and damage, thus preventing fatal accidents<sup>3</sup>. The visualization of such magnetic fields can define the parts that the diagnostic process covers, it can guide the look for and, ultimately, lead to the detection of structural flaws. A proper visualization is of the essence in this case, leading to better interpretation of the data, more informed decision-making and, finally, safety<sup>4</sup>.

Augmented Reality (AR) has been used to visualize sensor data since AR can better present the spatial qualities of the sensor data and overlay them on top of the real-world view as layers, so that they are better understood and lead to informed decision-making<sup>5</sup>. This functionality may also be beneficial for the visualization of invisible data so that users can interpret them in the best possible way. Overall, tackling the challenge of telling a detailed and complex story to a broad audience has now become possible by mixing the real and visible world with augmented depiction of invisible signals, either measured or modelled, while being able to collaborate with other users on these visualizations in 3D metaverse spaces<sup>5,6</sup>. The InVizAR project was created in an exploratory-research context to work towards that direction.

## 1.2 Project objective & motivation

InVizAR's main objective is to explore efficient and user-friendly AR visualizations of sensor data describing "invisible fields", like electromagnetic fields or radio signals. The focus here is on datasets with geospatial and temporal qualities. The ultimate goal of the project is to be able – through a use case – to design, develop, and present an AR visualization of invisible-fields' dataset that can be easily understood, interacted with, and enable users to reach to conclusions and make informed decisions. Moreover, a more practical

---

<sup>1</sup> Zhao, J., Fu, X., Yang, Z., & Xu, F. (2019). Radar-assisted uav detection and identification based on 5g in the internet of things. *Wireless Communications and Mobile Computing*, 2019.

<sup>2</sup> Nagarajan, M., Gomadam, K., Sheth, A. P., Ranabahu, A., Mutharaju, R., & Jadhav, A. (2009, October). Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences. In *International Conference on Web Information Systems Engineering* (pp. 539-553). Springer, Berlin, Heidelberg.

<sup>3</sup> Kot, P., Muradov, M., Gkantou, M., Kamaris, G. S., Hashim, K., & Yeboah, D. (2021). Recent advancements in non-destructive testing techniques for structural health monitoring. *Applied Sciences*, 11(6), 2750.

<sup>4</sup> Lee, C., & Park, S. (2015). Damage visualization of pipeline structures using laser-induced ultrasonic waves. *Structural Health Monitoring*, 14(5), 475-488.

<sup>5</sup> Kalkofen, D., Sandor, C., White, S., & Schmalstieg, D. (2011). Visualization techniques for augmented reality. In *Handbook of augmented reality* (pp. 65-98). Springer, New York, NY.

<sup>6</sup> Popescu, G. H., Ciurlău, C. F., & Stan, C. I. (2022). Virtual Workplaces in the Metaverse: Immersive Remote Collaboration Tools, Behavioral Predictive Analytics, and Extended Reality Technologies. *Psychosociological Issues in Human Resource Management*, 10(1).



goal of InVizAR is for the team involved in the project to get familiar with AR visualizations of sensor data coming from invisible fields.

The motivation behind this project, is typically found in projects where diverse sensor equipment is used to measure a physical phenomenon, e.g., a locally present electromagnetic field, and to blend these data with a known mathematical model of such field spreading and propagation, to obtain a depiction of this field covering a full 3D volume, typically as a hologram. The sensor data alone typically only covers a small subset of this 3D volume (i.e., sparse data) but can be used to calibrate the mathematical model for proper parameter settings. Finally, this processed data is visualized as a layer on top of the physical world, rendered by, e.g., a smartphone, tablet, or head-mounted displays (HMDs), such as the Microsoft HoloLens 2. The data is linked to the physical world by absolute geospatial coordinates, or relative to a spatial mapping performed of the physical world surroundings. In some cases, the sensor data may also be temporal, i.e., signals and fields that change strength and/or behavior over time.

This challenge is in line with many projects that need the following project workflow:

*Robots/drones with sensor technology payload => Data harvesting with spatiotemporal information => Visualization*

Use cases include: i) presenting complex industrial equipment maintenance status or new monitoring ideas to existing or potential customers, and ii) support the live feed of real time information to command-and-control or rescue personnel in challenging in-situ operations. In addition, such capabilities are a vital part of the creation of Digital Twins<sup>7</sup>.

The full scope of such a challenge is wide. 3D geospatial visualization is today possible using mixed reality equipment, such as Microsoft HoloLens; however, a conducted literature search revealed that despite of recent attempts in providing frameworks and workflows for such data renderings, it still involves a lot of hand coding and resource spending to achieve satisfactory results. On the far horizon of this project is the capability of offering augmented reality (AR) tools that will display 3D live measurements and calibrated mathematic models to end users in need for such capabilities, to understand physics and solve a multitude of different challenges. However, even though InVizAR contributes towards that goal, within the project's timeframe (that is, nine months), we can only address the "asynchronous" visualizations, i.e., visualizations that come from historical data, that concern the use of i), described above.

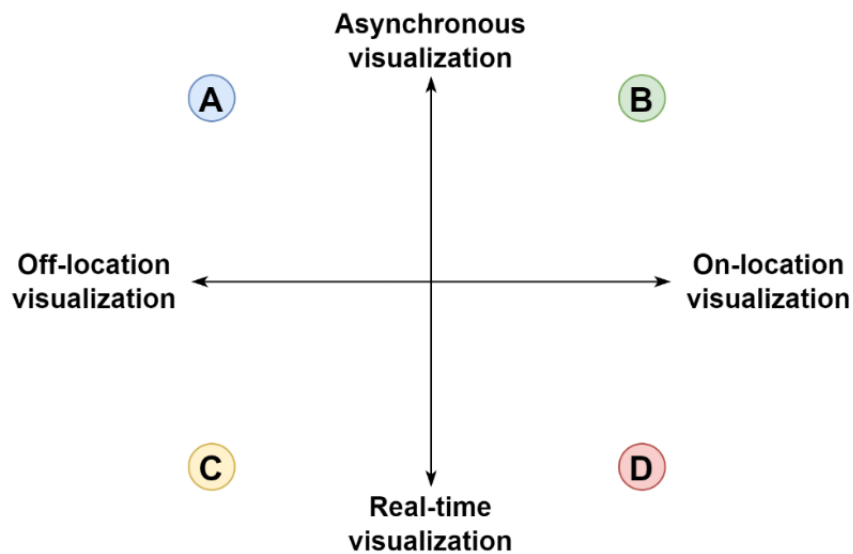
### 1.3 Targeted use & visualization mode

To better define the targeted visualization, we first identify the visualization modes that can take place in the field, based on a literature survey, and pinpoint the one that we consider appropriate for the intended use and effort in this project.

Figure 1 presents and simplifies the visualization modes, based on two axes. The horizontal one is about the spatial elements (location) of the visualization. With *on-location visualization*, we define the AR visualization that superimposes the 3D model on the real object. *Off-location visualization* takes place in another physical space than the one where the real object is placed (e.g., user's office). The vertical axis has to do with the timing of the data feed. *Asynchronous visualization* presents data that have been acquired through scanning before some time and a dataset has been formed. *Real-time visualization* concerns the live visualization of data that are coming in right at that moment when the scanning of the jacket platform takes place.

---

<sup>7</sup> 'What is a digital twin? | IBM'. <https://www.ibm.com/topics/what-is-a-digital-twin> (accessed Jun. 13, 2022).



**Figure 1.** The various visualization modes that can be utilized in our use case and in similar cases.

Based on these two axes, we identify four points that describe different uses and, thus, different visualization modes.

**Table 1.** Four potential visualization modes that we have identified based on previous experience with clients and related literature.

Visualization mode	Description
<b>(A): Asynchronous and off-location visualization</b>	This mode can be useful for communicating historical results remotely to someone else, e.g., co-worker, client <sup>8</sup> . Visualization can take place through: AR, VR, and 2D ways (the traditional way, it is happening now). Devices: phones/tablets, XR headsets, desktop PCs, dedicated probing devices.
<b>(B): Asynchronous and on-location visualization</b>	This mode is about visualizing a historical dataset in-situ, as a 3D model superimposed on the real object, in the AR view <sup>9,10,11</sup> . This is a dedicated AR visualization mode presenting contextual information <sup>12</sup> . Devices: phones/tablets, XR headsets.

<sup>8</sup> Gall, A., Gröller, E., & Heinzl, C. (2021, December). ImNDT: immersive workspace for the analysis of multidimensional material data from non-destructive testing. In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology (pp. 1-11).

<sup>9</sup> Bergamo, O., Campione, G., Donadello, S., & Russo, G. (2015). In-situ NDT testing procedure as an integral part of failure analysis of historical masonry arch bridges. *Engineering Failure Analysis*, 57, 31-55.

<sup>10</sup> Utzig, S., Kaps, R., Azeem, S. M., & Gerndt, A. (2019, March). Augmented reality for remote collaboration in aircraft maintenance tasks. In 2019 IEEE aerospace conference (pp. 1-10). IEEE.

<sup>11</sup> Ababsa, F. (2020, September). Augmented reality application in manufacturing industry: Maintenance and non-destructive testing (ndt) use cases. In International Conference on Augmented Reality, Virtual Reality and Computer Graphics (pp. 333-344). Springer, Cham.

<sup>12</sup> Kalkofen, D., Mendez, E., & Schmalstieg, D. (2007, November). Interactive focus and context visualization for augmented reality. In 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (pp. 191-201). IEEE.



<b>(C): Real-time and off-location visualization</b>	This mode can be useful for communicating probe results remotely to someone else, e.g., co-worker, client, in real-time, i.e., while the scanning takes place <sup>13</sup> . Visualization can take place through: AR, VR, and 2D ways. Devices: phones/tablets, XR headsets, desktop PCs, dedicated probing devices.
<b>(D): Real-time and on-location visualization</b>	This mode is about visualizing real-time probe measurements in-situ, as a 3D model superimposed on the real object, in the AR view <sup>13,14</sup> . This is a dedicated AR visualization mode presenting contextual information. Devices: phones/tablets, XR headsets.

For InVizAR, we target **(A)**, i.e., communicate the NDT results and visualization to third parties through the AR space. However, the vision of the project exists in the visualization modes **(B)** and, ultimately, **(D)**.

## 1.4 Collaboration

InVizAR is formed on a collaboration between the Communication Systems (Komsys) and Human-Computer Interaction (HCI) groups of the Sustainable Communication Technologies department of SINTEF Digital. The Komsys group holds expertise in electromagnetic fields, signal processing and communication, being involved in IPN projects like ANDWIS, while also running a related SEP2021 project, where live animation of the magnetic field vector was displayed correctly in (x,y,z) domain and in AR, relative to the view of a smartphone. The HCI group holds expertise in AR and VR visualization, user interface design and usability/user experience evaluation studies, being involved in project like Centre for Service Innovation (VR visualization of online services) and ARBIM (Augmented Reality & Intelligent Personal Assistants for Building Information Modelling).

- Komsys group: Bengt Holter, Karsten Husby, Arne Lie
- HCI group: Jan Håvard Skjetne, Costas Boletsis, Ophelia Prillard, Jiaxin Li

## 1.5 Methodology

In order to approach the main objective, a use-case-driven methodology was designed and utilized. To that end, the team used the expertise of the Komsys group members and their previous experiences on visualization needs in the sector. Therefore, the use case was defined by the team internally and based on experiences from previous projects and clients. The basic requirements for choosing an appropriate use case were: i) to have advanced visualization needs and ii) to provide an actual dataset coming from sensors on “invisible fields”. A game engine was decided to be used (Unity) for visualization purposes since that way we could utilize all the Extended Reality (XR) visualization capabilities offered by the engine and ensure the scalability of the project in the future in data volume and across XR technologies, if needed.

The next step was about formatting the dataset in a way that data can lead to a meaningful visualization. Therefore, an API structure (Section 3.2) had to be formulated for the data to be imported in the Unity game

<sup>13</sup> Sophian, A. (2021, June). Non-destructive testing (NDT) in industry 4.0: A brief review. In Proceeding International Conference on Science (ICST) (Vol. 2, pp. 1-9).

<sup>14</sup> Rehbein, J., Lorenz, S. J., Holtmannspötter, J., & Valeske, B. (2022). 3D-Visualization of Ultrasonic NDT Data Using Mixed Reality. *Journal of Nondestructive Evaluation*, 41(1), 1-11.



engine. Finally, a visualization metaphor had to be conceptually designed and implemented, facilitating the clear visualization of the sensor data and user-friendly interaction (Section 4). Due to the various expertise in the team, informal evaluations about the chosen AR visualization and its qualities would take place. The Komsys group would lead the evaluation of the data visualization as to the meaningful presentation of the data, while the HCI group would evaluate experiential qualities like user-friendliness, usability, and aesthetics.

## 1.6 Functional requirements

A set of functional requirements were set at the initial stages of the project, which would lead the design and development of the targeted visualization.

**Table 2.** The functional requirements for the targeted visualization in the InVizAR project.

Code	Functional requirements
FR1	Visualization should be produced by the project's available dataset.
FR2	Visualization should integrate interaction elements.
FR3	Visualization should be able to present different values at the same scene in a clear and comprehensible way.
FR4	Visualization should enable the presentation of and navigation among time series data.
FR5	Visualization's transparency values should be easily adjustable for not interfering with the real-world view.

## 2 Use case & dataset from sensor data

### 2.1 Dataset description

In order to start working on the software development and to fulfill FR1, there is a need for access to sensor data. There is also a need for complexity reductions compared to a reception of a live data stream. We also have the ambition to start working on data that is already connected to a SINTEF customer. Still, we want a non-trivial dataset, so that the visualization aspect has a certain challenge into *how* it shall be visualized. After a discussion we have concluded to work on a dataset that was recorded in the ANDWIS project for the client OceanTech Innovation AS. This dataset is thoroughly described in a paper<sup>15</sup>. In short it contains measurement data of specific fields recorded by an "NDT Eddy Current probe" (NDT: non-destructive testing), used to inspect offshore metal jacket platforms. Such platforms can be subject to material fatigue, especially at welding areas. An operator, typically, operates such a probe in challenging ocean environments, and would benefit a lot in having online augmented visualizations in her/his diving goggles. These visualizations should depict an analysis of what is being measured, with clear indications of any cracks in the jacket platform.

The actual acquired dataset was recorded onshore, and its setup is shown in Figure 2. The probe was moved seven times as shown in the figure, at seven different heights above the steel metal. In every recording, also seven different field frequencies are injected. Two different fields were recorded, as depicted in Figure 3. The paper<sup>15</sup> shows how the weighting of the different frequency response fields could be optimized in order

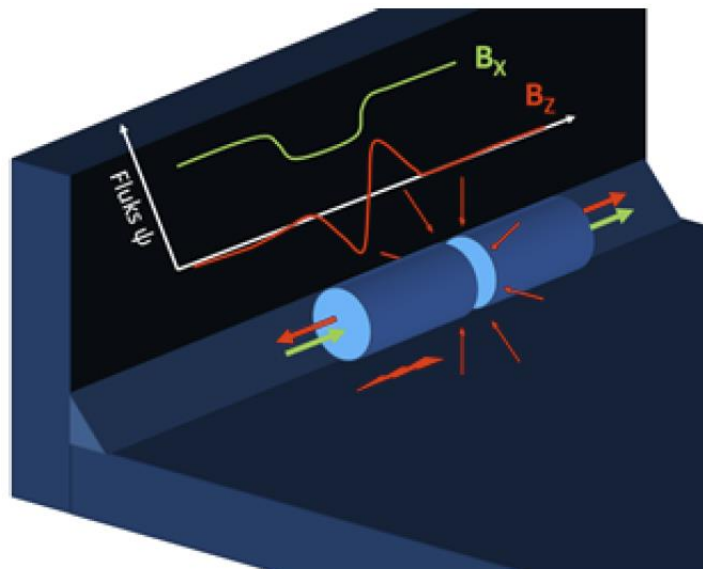
<sup>15</sup> K. Husby, C. E. Johnsen, and V. Ringset, 'Presenting a novel NDT Eddy Current probe for crack detection on rough welds and steel with variable permeability', in Proc. of IEEE SAS, Sundsvall, Sweden, 2022, p. 6.

to have measurements with the largest signal-to-noise (SNR) ratio, which is key to generate unambiguous crack indications. The  $B_z$  recorded values thus ended into the final processed data as  $B_M$ .

The dataset has one important missing property: the 3D geometry localization. Therefore, in the software, we have added artificial positions following the outer section of a tube. In addition, timestamps were added, assuming that a complete operation using the probe at a certain liftoff position took 125 seconds.



**Figure 2.** The recording of the actual dataset was performed onshore. The probe was moved along a welding in a zigzag movement, as indicated by the orange painted line.



**Figure 3.** The NDT Eddy current probe (cylinder) and the jacket platform welding of two parts positioned 90 degrees. The recording captures two complex flux fields:  $B_x$  and  $B_z$ . A crack is indicated as a red lightning marker.

The goal, in the future and outside the scope of InVizAR, is to have live reception of diverse external sensor data, e.g., streamed over WiFi directly or as a gateway for 4G/5G NB-IoT mobile networks, and that this set of data is encapsulated into a self-describing format, or a pre-described "raw" format. Further, if self-described, it should ideally comply with a language, a grammar, or an ontology, so that an interface to 2D or 3D rendering automatically understands the data, e.g., its meaning, its dimensions, and even to execute inference over a large set of data, without the need to create a tailored interface whenever the set of data changes. This topic is huge and relatively unexplored, it seems. Findings here include ontology suggestions like Liu et al.<sup>16</sup> and Elsaleh et al.<sup>17</sup>, and grammars like Vega and Vega-Lite<sup>18,19,20,21</sup>. To the authors knowledge, there seem to be no attempts so far to incorporate such formats to sensor data in the context of AR and visualization. It needs to be assessed if such incorporations can be beneficial. Such efforts are for the moment out-of-scope for this project.

## 2.2 Dataset syntax alternatives

For the purposes of this project, the formats of CSV (Comma Separated Values), XML (eXtensible Markup Language), and JSON (JavaScript Object Notation) are considered. The two latter are self-descriptive throughout the reception (CSV typically only has a single header that describes its data), and also allow for structured data with diverse dimensions to its siblings and children. JSON has overrun XML in many use cases the last years due to being lighter to stream and parse, and easier for human reading as well<sup>22</sup>. JSON, standardized by IETF in 2014<sup>23</sup>, has since 2016 got a sibling named GeoJSON<sup>24</sup>. These standards also include some grammar, i.e., specific characters and words that has special meaning. JSON itself only has structural characters as part of its grammar. Example:

```
{
  "book": [
    {
      "id": "01",
      "language": "Java",
      "edition": "third",
      "author": "Herbert Schildt"
    },
    {
      "id": "07",
      "language": "C++",
      "edition": "second",

```

<sup>16</sup> J. Liu, Y. Li, X. Tian, A. Sangaiah, and J. Wang, 'Towards Semantic Sensor Data: An Ontology Approach', *Sensors*, vol. 19, no. 5, p. 1193, Mar. 2019, doi: 10.3390/s19051193.

<sup>17</sup> T. Elsaleh, S. Enshaeifar, R. Rezvani, S. T. Acton, V. Janeiko, and M. Bermudez-Edo, 'IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams and Its Use with Data Analytics and Event Detection Services', *Sensors*, vol. 20, no. 4, p. 953, Feb. 2020, doi: 10.3390/s20040953.

<sup>18</sup> 'Vega'. <https://vega.github.io/> (accessed Jun. 11, 2022).

<sup>19</sup> 'A Visualization Grammar | Vega'. <https://vega.github.io/vega/> (accessed Jun. 11, 2022).

<sup>20</sup> 'A High-Level Grammar of Interactive Graphics | Vega-Lite'. <https://vega.github.io/vega-lite/> (accessed Jun. 11, 2022).

<sup>21</sup> A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, 'Vega-Lite: A Grammar of Interactive Graphics', *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 341–350, Jan. 2017, doi: 10.1109/TVCG.2016.2599030.

<sup>22</sup> 'A Deep Look at JSON vs. XML, Part 1: The History of Each | Toptal'. <https://www.toptal.com/web/json-vs-xml-part-1> (accessed Jun. 11, 2022).

<sup>23</sup> T. Bray, 'RFC 7159 - The JavaScript Object Notation (JSON) Data Interchange Format', Mar. 2014. <https://datatracker.ietf.org/doc/html/rfc7159> (accessed Jun. 11, 2022).

<sup>24</sup> H. Butler, M. Daly, and A. Doyle, 'RFC 7946 - The GeoJSON Format', Aug. 2016.

<https://datatracker.ietf.org/doc/html/rfc7946> (accessed Jun. 09, 2022).

```

    "author": "E.Balagurusamy"
  }
]
}

```

Where:

- curly brackets { and } enclose an *object*,
- square brackets [ and ] enclose an *array*,
- colon : is *name separator*, and
- comma , is *value separator*.
- Whitespace can be tolerated as (possibly multiple) space, tabs, newline and carriage return.

A GeoJSON object represents a Feature, or collection of Features, that has a Geometry with coordinates and a set of Properties, and is itself a JSON object. Thus, GeoJSON then also includes a grammar that must be used in order to comply to its standard. Example:

```

{
  "type": "FeatureCollection",
  "features": [{
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [102.0, 0.5]
    },
    "properties": {
      "prop0": "value0"
    }
  }, {
    "type": "Feature",
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [102.0, 0.0],
        [103.0, 1.0],
        [104.0, 0.0],
        [105.0, 1.0]
      ]
    },
    "properties": {
      "prop0": "value0",
      "prop1": 0.0
    }
  }, {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [100.0, 0.0],
          [101.0, 0.0],
          [101.0, 1.0],
          [100.0, 1.0],
          [100.0, 0.0]
        ]
      ]
    }
  }
]
}

```



```
    ],  
    },  
    "properties": {  
      "prop0": "value0",  
      "prop1": {  
        "this": "that"  
      }  
    }  
  }  
}]  
}
```

GeoJSON supports six different geometry object types: Point, MultiPoint, LineString, MultiLineString, Polygon and MultiPolygon. Many of these geometry types seem adequate for sensor data. The coordinate reference system for all GeoJSON coordinates is a geographic coordinate reference system, using the World Geodetic System 1984 (WGS 84) datum. Therefore, in our efforts to fulfill FR1, the GeoJSON was chosen as our preferred grammar, setting up the scene for what is described below.

## 3 Dataset software

### 3.1 Software documentation

The datasets coming from ANDWIS were stored in Matlab MAT files. In InVizAR, four main Matlab M-files scripts have been created:

- **makeAndwisJson.m**: imports the file dataset "Dora\_noise.mat" and outputs **andwisRawData.json**
- **run\_B\_M.m**: imports "B\_M.mat" and outputs **andwisProcessedData.json**
- **run\_B\_M\_expanded.m**: extended version that makes data in the period 2018-2022, one set per quarter, and creates one JSON file per year.
- **make\_tube.m**: generates artificial 3D coordinates to be used as missing data in the datasets, see Figure 4. Currently, the red curve is in absolute coordinates, meaning that they are not processed as an intersection of the tube.

The Matlab scripts puts the data into Matlab structs that resembles GeoJSON. Finally, when inputting these structures into the Matlab function `jsonencode()`, the outputted JSON will respect this structure. It is worth noting that the dataset `Dora_noise.mat` is a measured dataset conducted at the site and object as shown in Figure 2, while `B_M.mat` is conducted at a metal plate having a known 4 mm crack in it. The reason for this deviation is that the jacket platform (Figure 2) *do not have any cracks at all*. So, the processed set "B\_M.mat" contains a crack signature in its data, but similar type of signal spikes would have been processed in Figure 2 as well, if that object had similar type of crack. The crack signature detection is shown in Figure 5.

The GeoJSON coordinates are, according to the standard, in geodetic WGS84 format, i.e., decimal latitude/longitude. An optional third value is to be regarded as height above ground in unit meters. We thus need to place the tube center in Figure 4 in UTM coordinates, calculate the (x,y) coordinates of the half circle based on this easting/northing offset, and then transpose these to latitude/longitude. This is obtained by also including the following scripts from Matlab Central<sup>25</sup> (provided by Erwin Nindl and Rafael Palaciosin, 2013):

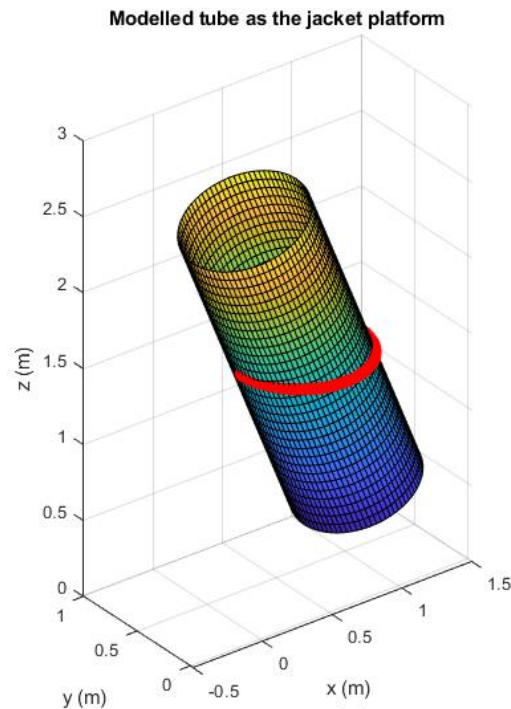
- **utm2deg.m**

<sup>25</sup> 'utm2lonlat - File Exchange - MATLAB Central'. <https://se.mathworks.com/matlabcentral/fileexchange/44242-utm2lonlat> (accessed Jun. 14, 2022).

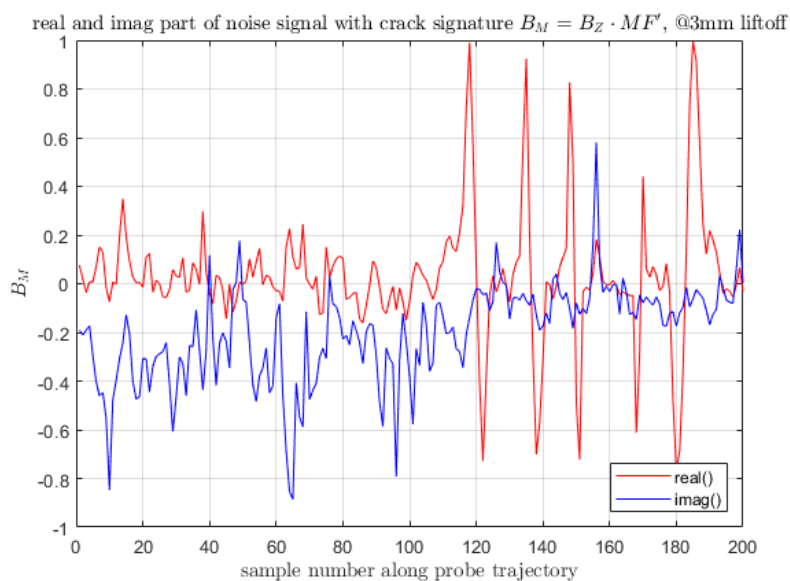


- `deg2utm.m`

We selected to place the origin of the coordinate system shown in Figure 4 at surface ground in the car parking space outside SINTEF in Strindveien 4, Trondheim, as shown in Figure 6. Google Earth shows that the UTM coordinates of this spot is 570383.88 (easting) and 7032422.39 (northing) relative to UTM zone 32 V. The elevation here (height above sea level) is about 38.5 meters: it is possible that this offset should be added to the z-value of GeoJSON. The final lines in `run_B_M.m` writes a KML file using the `kmlwriteline()` function in Matlab, using the latlong coordinates of the red intersection circle as input. Figure 7 proves that the coordinate handling is correct.



**Figure 4.** The modelled tube and half circle positions of the Eddy Current probe (red circles).



**Figure 5.** The real part of  $B_M$  carries the strongest capabilities in showing welding crack locations.

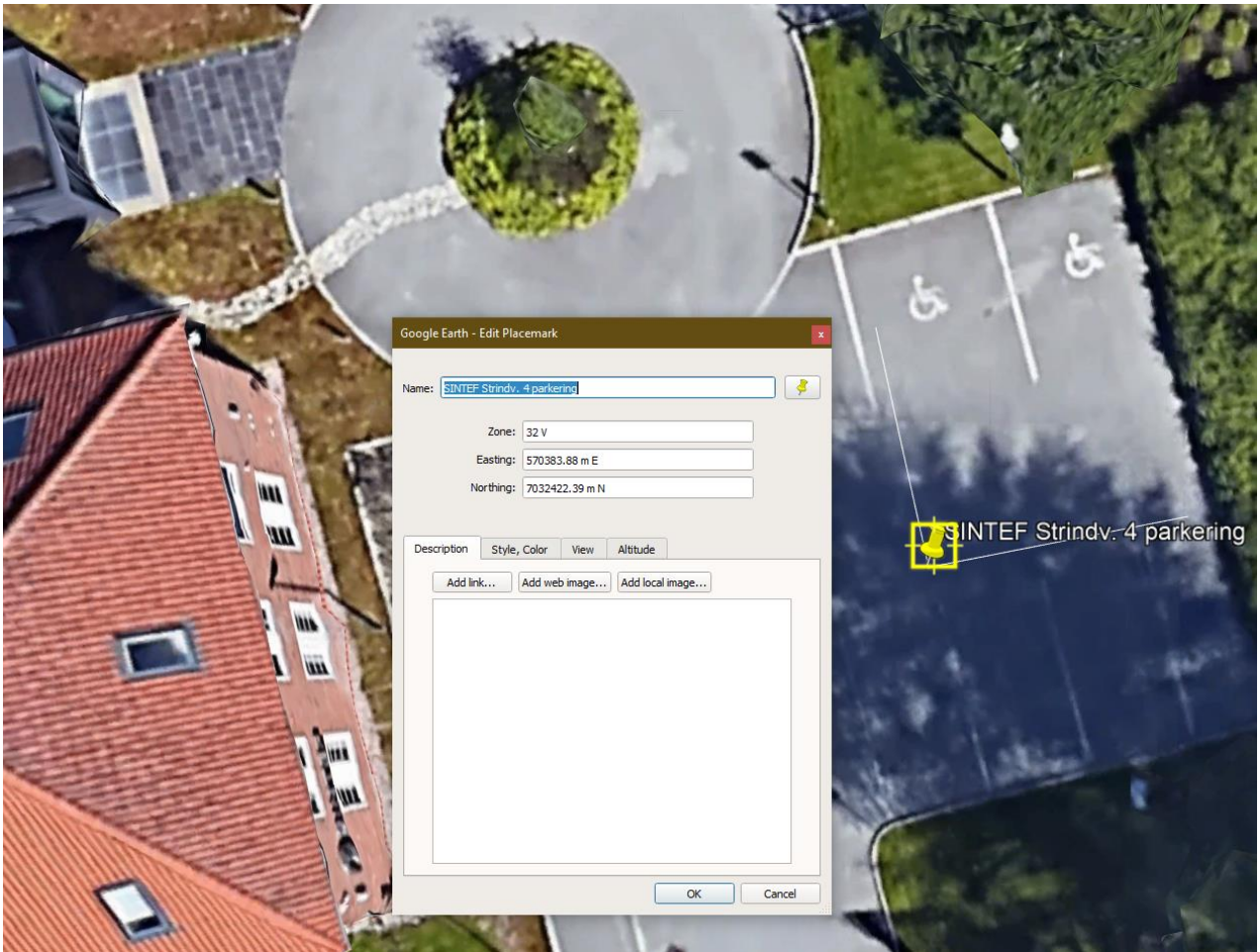


Figure 6. (x,y) origin placed at parking outside SINTEF in Strindveien 4, Trondheim.



Figure 7. Write to a KML file proves the latlon coordinates are handled correctly (red circle outside SINTEF building).

As can be seen from the following excerpts of the two JSON files, its structure resembles the GeoJSON format. The geometry:coordinates has a 3-element 3D point consisting of absolute polar WGS84 coordinates [longitude,latitude,'height above ground'], where the two first is decimal degrees, and the latter in decimal meters. The relative cartesian meter-based coordinates (as displayed in Figure 4) are recorded as properties:localPosition.

**andwisRawData.json:**

```
[
  {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [
        10.409599828432194,
        63.413383074877757,
        1.5001506592702698
      ]
    },
    "properties": {
      "localPosition": [
        0.20016228336130562,
        0.40018506203603038,
        1.5001506592702698
      ],
      "sampleNumber": 1,
      "timestamp": "2022-06-14 11:09:35.492",
      "frequencies": [
        "13 kHz",
        "19 kHz",
        "32 kHz",
        "52 kHz",
        "77 kHz",
        "123 kHz",
        "194 kHz"
      ],
      "liftoff": 0,
      "Bx_real": [
        -0.29917687245001218,
        -0.29656469421314496,
        -0.29637602782848749,
        -0.30126241584885305,
        -0.3092508318649097,
        -0.3180659438360528,
        -0.32360297984267955
      ],
      "Bx_imag": [
        0.030270485013214211,
        0.034701910045938776,
        0.040213821547212841,
        0.044310485392368849,
        0.042963847716882039,
        0.036091270287350957,
        0.026456968429828896
      ],
      "Bz_real": [
        0.003975801688038719,
```



```
0.0036109992325254222,  
0.0004001226467604792,  
-0.0012255801639684061,  
-0.0029668467537074892,  
-0.0052317643018351774,  
-0.0054082024817106453  
],  
"Bz_imag": [  
0.00700449413129888,  
0.0080338922445678442,  
0.010031631548546867,  
0.0098480367210163778,  
0.009269481974643062,  
0.00835514116761885,  
0.0097150234253938223  
]  
}  
},  
{  
"type": "Feature",  
"geometry": {  
"type": "Point",  
"coordinates": [  
10.409599820465948,  
63.413383013020862,  
1.5000864796492368  
]  
},  
"properties": {  
"localPosition": [  
0.19991619955271644,  
0.39328524331022546,  
1.5000864796492368  
],  
"sampleNumber": 2,  
...  
}
```

#### andwisProcessedData.json:

```
[  
{  
"type": "Feature",  
"geometry": {  
"type": "Point",  
"coordinates": [  
10.409599234234745,  
63.413383080855823,  
1.4998109990374255  
]  
},  
"properties": {  
"localPosition": [  
0.17048177310630078,  
0.40019823966291396,  
1.4998109990374255  
],  
"sampleNumber": 1,  
}
```

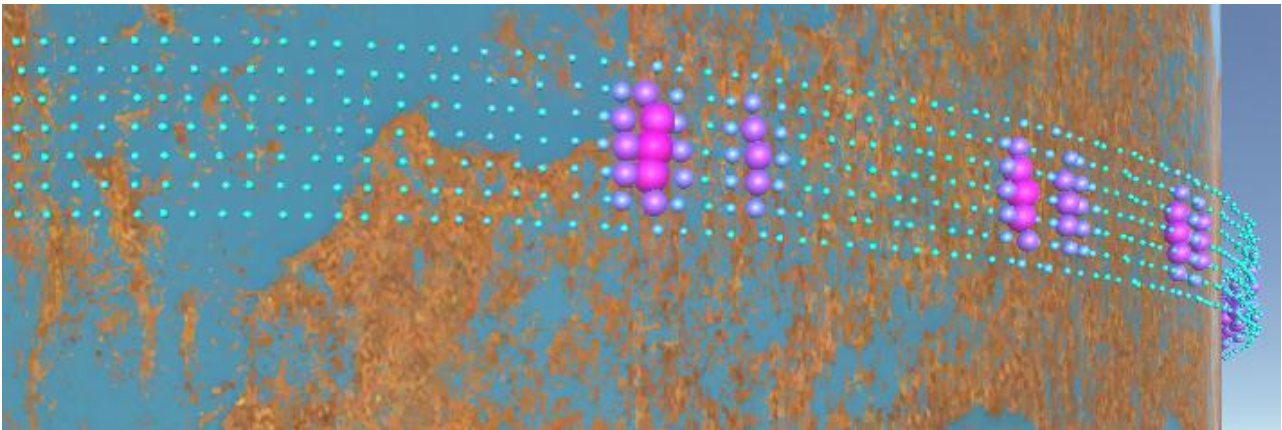


```
"timestamp": "2022-06-14 11:08:08.949",
"frequencies": [
  "13 kHz",
  "19 kHz",
  "32 kHz",
  "52 kHz",
  "77 kHz",
  "123 kHz",
  "194 kHz"
],
"MF": [
  0,
  0,
  0.85,
  0,
  0,
  0.75,
  0.91
],
"liftoff": 3,
"B_M_real": 0.076252519579397354,
"B_M_imag": -0.19126790401174562
}
},
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      10.409599218501315,
      63.413383021055381,
      1.5004738825700226
    ]
  },
  "properties": {
    "localPosition": [
      0.16984286522135827,
      0.39351897652142986,
      1.5004738825700226
    ]
  },
  "sampleNumber": 2,
  ...
```

In the final phase of the InVizAR project, it was decided to expand the set of data both in space and time:

- Space: it was envisioned that the measurements were not only performed along the curved side of the steel pipe, but along multiple runs along the curved steel pipe, in vertical offsets from each other (both above and below the original curve). The data values larger than absolute 0.3 were gradually faded towards each end. With three curves above and below the original dataset, the total dataset size was increased to factor 7x.
- Time: it was envisioned that the actual data recording was the last in a set of recordings started in 2018, and performed every 3 months. Thus, the 7x dataset size was increased to  $(7 \times 5 \times 4) = 140x$ .

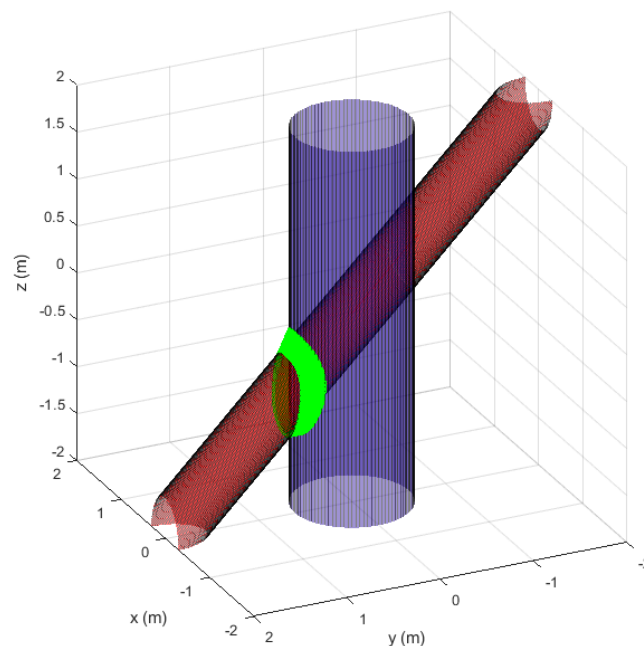




**Figure 8.** A visualization of the dataset when expanded to 7x – three additional curves above and below the original one.

### 3.2 Software inherited from ANDWIS

The InVizAR project (addressing and fulfilling FR1) was allowed to inherit some Matlab code from ANDWIS, thanks to Magnus Bjerkeng (Mathematics and Cybernetics, SINTEF). This software makes some detailed 3D calculations about the pipes and intersection area that need to be welded. This resulted in the depiction of Figure 9.



**Figure 9.** The ANDWIS advanced intersection of two pipes and the (green) welding area shown.

The welding area has a certain width and depth. The software calculates 3D points making multiple polygons to draw such areas, showing a full 360-degree welding. By dividing these 360 degrees into 200 polygons, we can anticipate that each recording is performed inside each polygon, and its visualization can benefit from the knowledge where this welding is located. In GeoJSON, we then switch to using type:Polygon, having five 3D coordinates, but where the first and last is the same coordinate. In this way we get closed polygons with four corners.

**andwisKyb\_ProcessedData.json:**

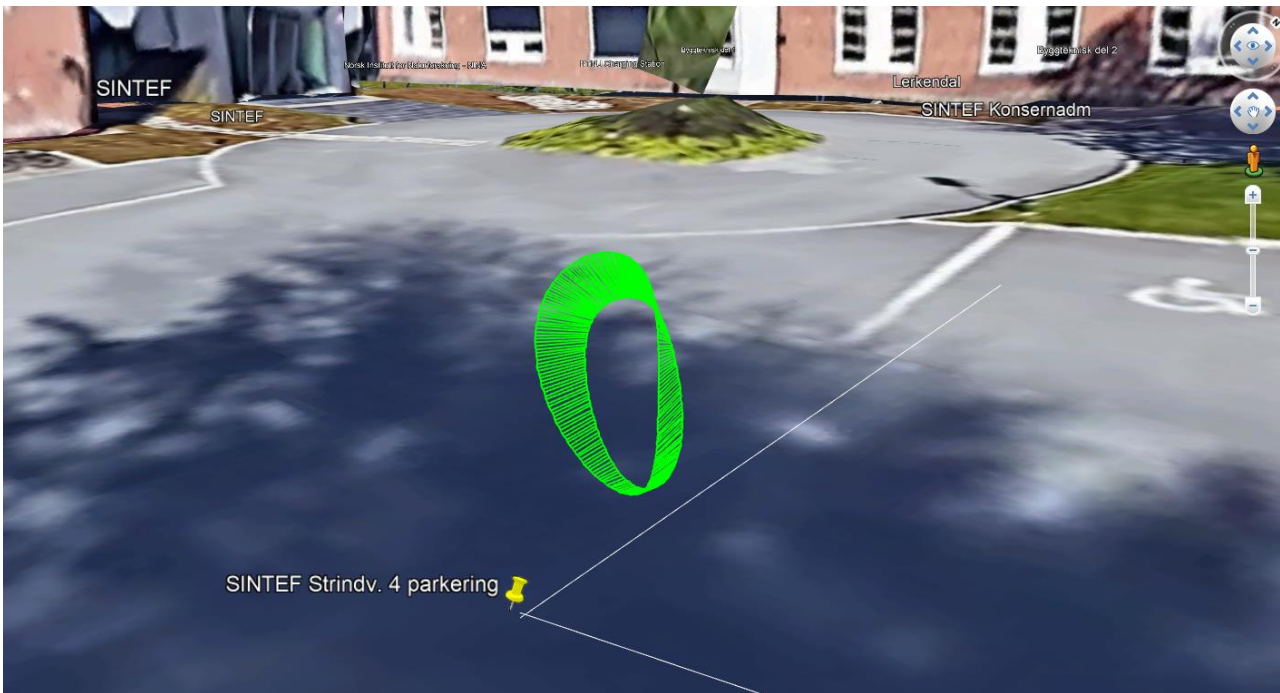
```
[
  {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          10.409595960170119,
          63.413385944918915,
          38.853280075062266
        ],
        [
          10.40959631424538,
          63.413385941663556,
          38.853979961911868
        ],
        [
          10.409596226750281,
          63.413384988381395,
          38.859825135461975
        ],
        [
          10.409595913442956,
          63.413384993275635,
          38.859352255469538
        ],
        [
          10.409595960170119,
          63.413385944918915,
          38.853280075062266
        ]
      ]
    },
    "properties": {
      "sampleNumber": 1,
      "timestamp": "2022-06-15 14:27:39.841",
      "frequencies": [
        "13 kHz",
        "19 kHz",
        "32 kHz",
        "52 kHz",
        "77 kHz",
        "123 kHz",
        "194 kHz"
      ]
    }
  },
  ...
]
```

The following software is added to the repository:

- **run\_B\_M\_v2.m:** InVizAR modifications of run\_B\_M.m to enable the use of the Kyb ANDWIS files
- from\_Kyb/Rx.m
- from\_Kyb/Rzyx.m
- from\_Kyb/line\_cylinder\_intersection.m
- from\_Kyb/objfun\_for\_seam.m

- `from_kyb/show_weld_seam_surface.m`: minor modification introduced to get a return struct that holds the needed coordinates.
- `from_kyb/t1_cylinder.m`
- `from_kyb/t2_cylinder.m`

The main script writes JSON data to `andwisKyb_ProcessedData.json`. Using the `kmlwriteline()` function we were able to also port the resulting data to KML and visualize it in Google Earth, as shown in Figure 10.

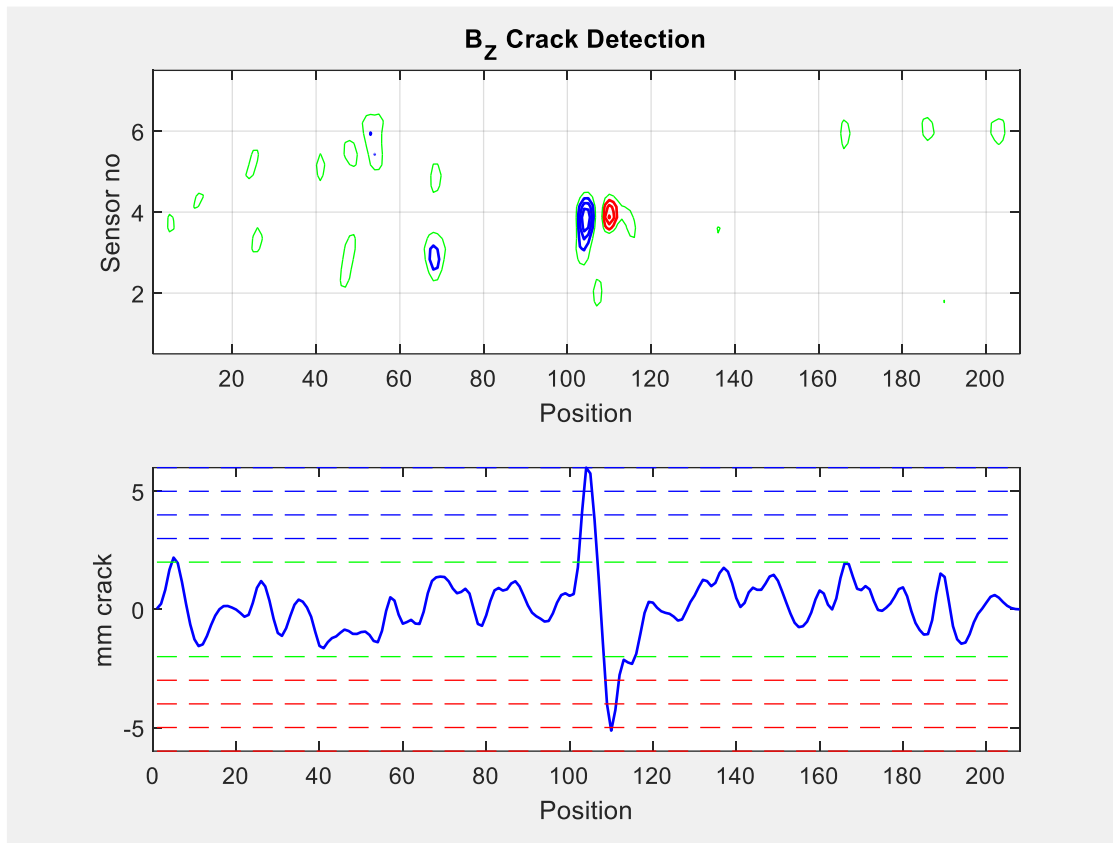


**Figure 10.** The welding polygons shown at absolute latlong reference.

### 3.3 Further ideas for visualization

The NDT Eddy probe has still ongoing development for improvement, and an ongoing project is making a probe that is attached to a robot, where the probe itself consists of seven distinct sensors placed along a line perpendicular to the movement direction of the probe. In this way, there is no need for an operator that performs a zigzag movement of the probe (as shown in Figure 2). In this case, seven "channels" are recorded, and the field intensity coming from all these sensors might be seen as a contour map, as shown in Figure 11 upper part. In the figure's lower part, the measured  $\Re(B_M)$  crack signature is calibrated and shown in another fashion that indicates the depth of the crack. This illustration is only meant to give ideas for how and what axis contexts a 3D hologram visualization can be challenged into.

The Matlab script used is named `ECA_visualization.m` and is provided by K. Husby from ANDWIS project.



**Figure 11.** Possibilities for visualization of crack detection when multiple probe sensors are used in parallel.

### 3.4 Git repository for making JSON files

```
$ git remote -v
origin  ssh://git@git.code.sintef.no/inar/json_testing.git (fetch)
origin  ssh://git@git.code.sintef.no/inar/json_testing.git (push)
```

```
$ git ls-files
.gitignore
ANDWIS_PAPER.pdf
B_M.mat
Dora_noise.mat
ECA_visualization.m
andwisInVizAR.kml
andwisKyb_InVizAR.kml
andwisKyb_ProcessedData.json
andwisProcessedData.json
andwisRawData.json
from_Kyb/Rx.m
from_Kyb/Rzyx.m
from_Kyb/line_cylinder_intersection.m
from_Kyb/objfun_for_seam.m
from_Kyb/show_weld_seam_surface.m
from_Kyb/t1_cylinder.m
from_Kyb/t2_cylinder.m
makeAndwisJSON.m
make_tube.m
run_B_M.m
run_B_M_v2.m
run_imag_B_noise.m
utm2lonlat/deg2utm.m
utm2lonlat/license.txt
utm2lonlat/utm2deg.m
```

```
$ git log --oneline
aead53d (HEAD -> master, origin/master) Initial commit ECA_visualization.m
```

675d908 Some modifications of the kyb files, and addition of a version 2 of the run\_B\_M in order to make use of the kyb files to calculate the welding area polygons.  
 2886827 Renamed \*cylinder\* to \*cylinder\*  
 96eb5fb Cleanup commented out code, and some indenting stuff.  
 867f48b Adding (initial commit) steel pipe math files from kyb (M. Bjerkgeng)  
 a445bb4 Adding the conversion scripts between UTM and decimal degrees.  
 d316118 The KML file of the intersection.  
 2ca2cee geometry.coordinates now in WGS84 latlon format, as it should be according to GeoJSON spec. run\_B\_M.m now also includes to write intersection line to a KML file for displaying in e.g. Google Earth.  
 791afee Shifted from cm to meters, and adding a offset input parameter for z-axis (height)  
 a900160 Cleanup of script comments and obsolete JSON files.  
 faec46e Renaming the JSON output files  
 451b0e1 <feat> Making run\_B\_M.m GeoJSON inspired as well, and fixing some small typo'es.  
 966d1cb <feat> JSON file format now is inspired by GeoJSON  
 6ca8f05 Improved plotting of text and precision.  
 407333c Added missing data, i.e., (x,y,z) coordinates that follow a tube with a certain liftoff. Added time offsets as well.  
 3486cca <feat> JSON format is now included as suggestion.  
 f09d9ff Initial Commit.

## 4 AR visualization prototype

### 4.1 Design concept

To define the type of the AR visualization, both literature and general searches took place, addressing the perspectives of AR visualization and NDT probe of jacket platforms. The aim was to identify base practices and detect any common practices that we could use for our use case. Followingly, the results of these searches are described.

Based on a literature survey on AR visualization techniques for data that demonstrate some sort of activity in invisible fields over space and time, we came across AR visualizations on datasets acquired by eye-/gaze-tracking<sup>27</sup>. This kind of data, demonstrate an activity (e.g., gazing at something) in the user's surrounding space, which takes place over time, and is visualized as a heatmap (Figure 12). At the same time, there are works in visualizing ultrasonic NDT data<sup>28</sup> and structural health monitoring for aircraft maintenance<sup>29</sup> using AR, which also use heatmaps/colormaps. Furthermore, a survey was conducted on existing or projected visualization techniques for NDT probes of jacket platforms. When analyzing the search results, we came across Creamform's Pipecheck solution<sup>30</sup>, which also visualized the results of an NDT probe as a heatmap (Figure 13).

Therefore, we chose to design the AR visualization of the use case's dataset as a heatmap (fulfilling FR3). The heatmap is a powerful tool for visualizing multi-dimensional data, where individual values can be expressed as colors<sup>31</sup>. "Heatmap is a popular method for visualizing matrix-like data by taking colors as the aesthetic elements. There are two major categories of heatmap visualization: spatial heatmap and grid

<sup>27</sup> Cochrane, T., Aiello, S., Cook, S., Aguayo, C., & Wilkinson, N. (2020). MESH360: a framework for designing MMR-enhanced clinical simulations. *Research in Learning Technology*, 28.

<sup>28</sup> Rehbein, J., Lorenz, S. J., Holtmannspötter, J., & Valeske, B. (2022). 3D-Visualization of Ultrasonic NDT Data Using Mixed Reality. *Journal of Nondestructive Evaluation*, 41(1), 1-11.

<sup>29</sup> Prado, G., Silva, P., Simomura, F., & Pereira, K. L. O. (2018). SHM with augmented reality for aircraft maintenance. In 31st Congress of the International Council of the Aeronautical Sciences.

<sup>30</sup> Creamform Pipecheck,

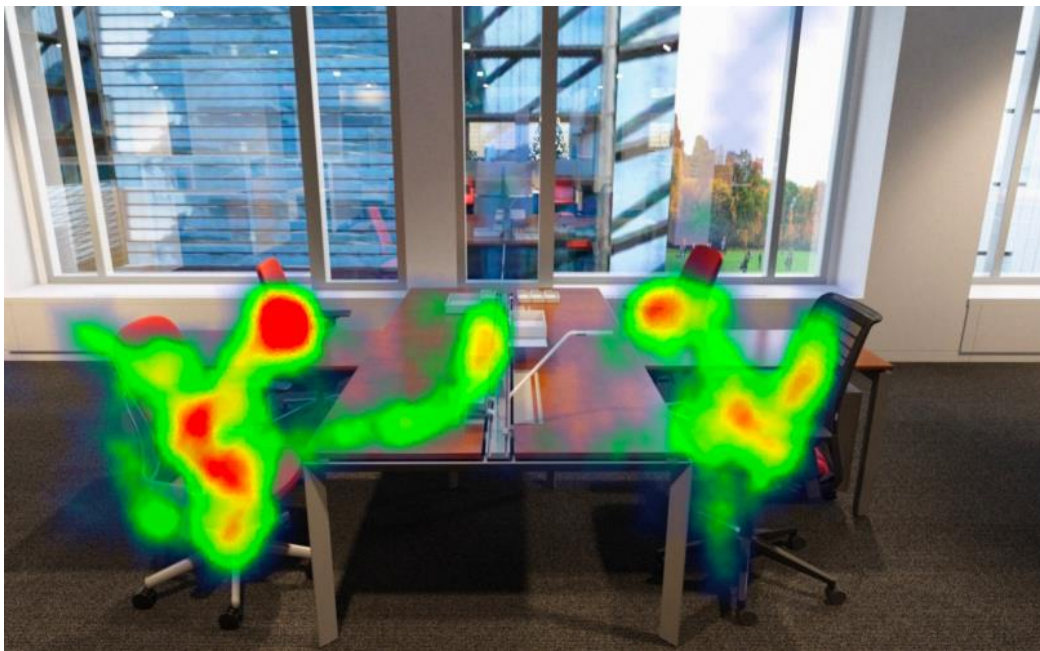
[https://www.creamform3d.com/sites/default/files/assets/ndt/2016/pipecheck\\_brochure\\_en\\_hq\\_20082015.pdf](https://www.creamform3d.com/sites/default/files/assets/ndt/2016/pipecheck_brochure_en_hq_20082015.pdf) (accessed Dec. 7, 2022)

<sup>31</sup> Mangiola, S., & Papenfuss, A. T. (2020). tidyHeatmap: an R package for modular heatmap production based on tidy principles. *Journal of Open Source Software*, 5(52), 2472.



heatmaps. The first category visualizes spatially distributed patterns, such as the global temperature distribution across the world, or the click activities on a web page from users... The second category is purely a rectangular layout of colored grids, where the two dimensions correspond to two types of variables<sup>32</sup>. In this work, we only address the first category due to the multi-dimensionality of our data. The works of Schloss et al.<sup>33</sup> on colormaps and of Liu et al.<sup>34</sup> on visualizing cylindrical coordinates with heatmaps (Figure 14) gave us a better understanding of the visualization type we are targeting for our use case.

The colors of the heatmap and the values' thresholds for those are going to be defined in the code. The use of sliders has been chosen as a user-friendly way of addressing FR4 and FR5, allowing users i) to easily navigate between datasets from the same jacket platform that were acquired through different NDT probes, over time, and ii) to manipulate visualization's transparency values. For visualization and, also, interaction purposes, the widely used Microsoft Hololens 2 headset was utilized, which has also been used in NDT-related use cases for visualization and interaction purposes<sup>10,11</sup> (thus addressing and fulfilling FR2).



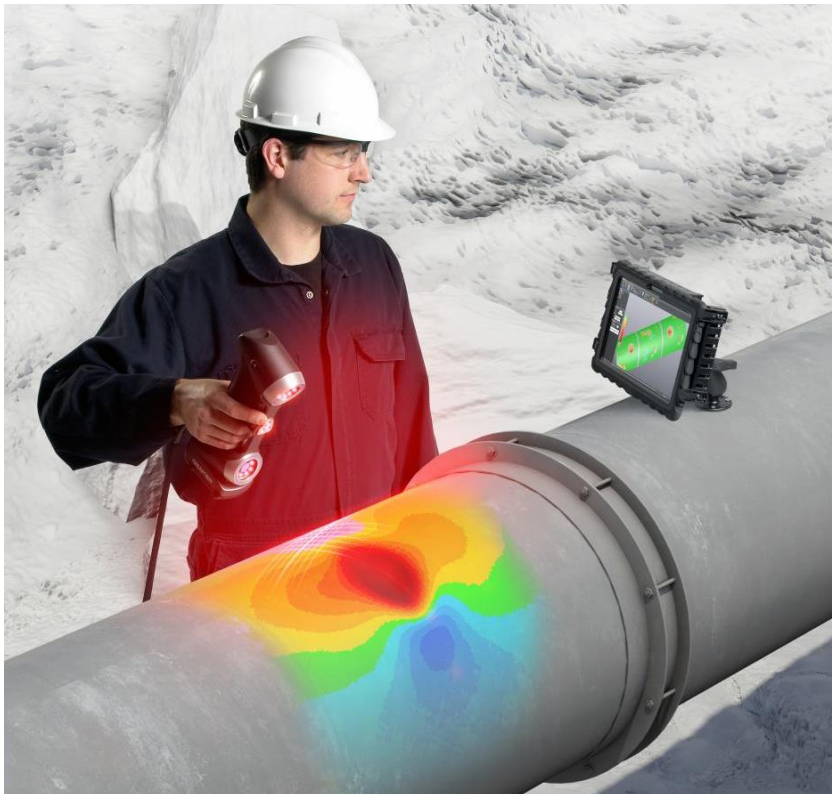
**Figure 12:** Heatmap visualization of data coming from gaze-tracking<sup>35</sup>.

<sup>32</sup> Gu, Z. (2022). Complex heatmap visualization. *iMeta*, 1(3), e43.

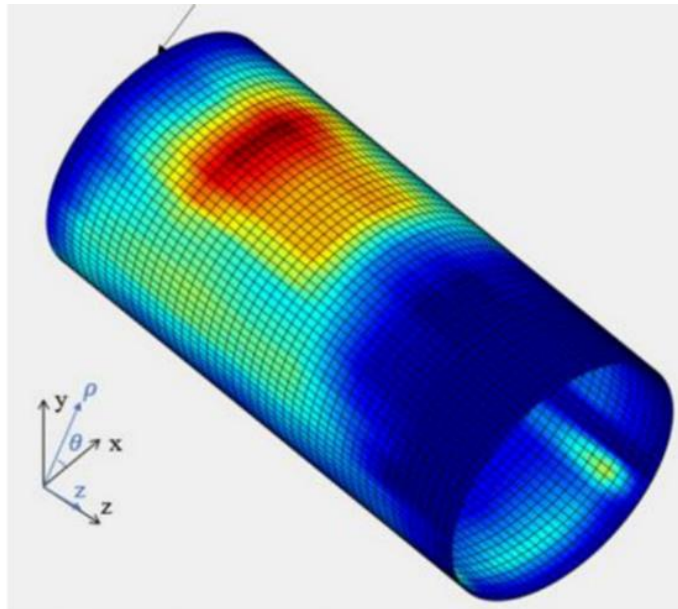
<sup>33</sup> Schloss, K. B., Gramazio, C. C., Silverman, A. T., Parker, M. L., & Wang, A. S. (2018). Mapping color to meaning in colormap data visualizations. *IEEE transactions on visualization and computer graphics*, 25(1), 810-819.

<sup>34</sup> Liu, Q., Liu, H., Huang, X., Pan, Y., Luo, C., & Sang, H. (2019). Inverse analysis approach to identify the loads on the external TBM shield surface and its application. *Rock Mechanics and Rock Engineering*, 52(9), 3241-3260.

<sup>35</sup> VRScout, "VR Heat Map Analytics Could Be Next Big Thing For Designers & Retailers", <https://vrscout.com/news/vr-heat-map-analytics-designers-retailers/> (accessed Dec. 7, 2022)



**Figure 13:** The Creafom Pipecheck solution for NDT probes, visualizing the results a heatmap<sup>30</sup>.



**Figure 14:** Heatmap visualization of cylindrical coordinates<sup>34</sup>. A sketch that was inspiring for our designed AR visualization on jacket platforms with cylindrical structure (pipes).

## 4.2 Hardware and development

The following hardware was used:

- A development PC running Windows 10 Enterprise 21H2 w/ OS build 19044.2006
  - The software packages listed below was installed on this computer

- Hololens 2 updated with Windows 10 Holographic 22H1 OS build 20348.1518 (September 2022)
  - Install "OpenXR Tools for Windows Mixed Reality" from Windows Store

In short, the following software packages was used and configured on the development PC:

- Unity / Unity Editor version 2020.3.35f1 LTS (and Unity HUB v3.3.0) with the following modules:
  - Universal Windows Platform Build Support (UWP)
  - WebGL Build Support
  - Unity Plus license
  - For collaboration in Unity between the HCI and KomSys groups, Plastic SCM was used.
- Microsoft Mixed Reality Feature Tool (MRTK)
- Visual Studio Community 2022 v17.3.4 (2019 version can also be used) with
  - .NET desktop development
  - Desktop development with C++
  - Universal Windows Platform (UWP) development
  - Game development with Unity
  - Windows 10 SDK 10.0.19041.0
- HoloLens 2 Emulator
  - Windows Holographic, version 22H1 September 2022 Update 10.0.20348.1518

Dr. Arne Lie of the KomSys group has created the very detailed guide “Unity download and configuration for InVizAR”, which can be accessed here: [Unity download and configuration for InVizAR.docx](#) . An application guide is available here: [UnityHoloLensApplicationGuide.docx](#) . The system architecture is visualized in the following Microsoft Visio file: [architecture.vsd](#) . The datasets can be accessed here [Data](#) . In case the files are not accessible, please contact Dr. Arne Lie (arne.lie@sintef.no) or Dr. Costas Boletsis (konstantinos.boletsis@sintef.no).

### 4.3 InVizAR prototype

The InVizAR prototype was developed as a feasibility demo, implementing the conceptual designs described above, it went through short development cycles, and it can be seen in action in the following videos: <https://hclilab.no/invizar2022/video/> There, two supplementary and alternative - as to their functionalities and visualization attributes – versions of the prototype are presented. The following features are presented:

- Close inspection and Animation ON
- ColorMap change
- Particle System activation
- Enabling of Bounds Control and move of pipe using Far Interactions
- Disabling of Bounds Control
- Particle System OFF and see how it dissolves away
- Particle System ON and how it displays one measured line after each other
- Opacity change
- Reset of Pipe to original position
- Test of Hide Pipe and Show Pipe buttons
- Tap on Point for additional frequency information
- Time Slider for navigating through historical datasets

The executable file (for the Microsoft HoloLens 2) and the source code can be accessed through the project's Git repository: `ssh://git@git.code.sintef.no/inar/invizar_final.git`. For access to the repository, please contact Dr. Arne Lie (arne.lie@sintef.no).

## 4.4 Evaluation and conclusion

The functional requirements, we set at the initial stages of the project have been fulfilled (Table 3) and led to the development of the InVizAR prototype. Both the design concept and the application were internally peer-reviewed and accepted by our teams' experts (Dr. Arne Lie, Dr. Karsten Husby) as to their suitability for communicating the project's ideas to potential clients. To further validate our designs, based on an external peer-review process, we also submit our work as an extended abstract for the 14th International Conference on Information Visualization Theory and Applications, with this report as supplementing material.

**Table 3.** The ways InVizAR addressed the functional requirements.

Code	Functional requirements	How they were fulfilled in InVizAR
FR1	Visualization should be produced by the project's available dataset.	For importing the available ANDWIS dataset to the Unity game engine, the GeoJSON standard was chosen, and a new API structure was defined.
FR2	Visualization should integrate interaction elements.	To that end, we used the Microsoft HoloLens headset that features efficient interaction techniques in its applications, right out of the box.
FR3	Visualization should be able to present different values at the same scene in a clear and comprehensible way.	The heatmap visualization type was utilized, while the colors that signify differences in frequencies and the frequencies' threshold values can be defined in the code.
FR4	Visualization should enable the presentation of and navigation among time series data.	A slider was implemented to navigate among various datasets.
FR5	Visualization's transparency values should be easily adjustable for not interfering with the real-world view.	A slider was implemented for adjusting transparency values.

Overall, through InVizAR, we strengthened our knowledge and competences on the use of heatmaps in AR settings for visualizing sensor data coming from invisible fields. Heatmaps can support the visualization of the multidimensionality of data, an attribute also communicable through AR. Naturally, the way heatmaps are applied and designed in AR settings is an ad-hoc process that depends on the use case. The results of this project can be used to inform the design of related tasks with similar needs and requirements. More specifically, in InVizAR, the following main results were produced:

- An **API structure** (based on the ANDWIS dataset, see Section 3.2) for importing NDT datasets in game engines.
- A **conceptual framework** of the potential **AR visualization modes of NDT data** (Figure 1 and Table 1).
- A high-fidelity, **prototype application** on a state-of-the-art AR visualization of frequencies, coming from a real-life NDT dataset.

Moreover, knowledge was exchanged between the Komsys and HCI groups, regarding setting up a Unity-based collaboration for producing visualizations, as well as examining the research topics of NDT probes, structural health monitoring, and jacket platforms.

Future work will address multi-user access to the AR visualization and the (B) and (D) modes of Table 1.