DEVELOPING A GEODATABASE FOR EFFICIENT UAV-BASED AUTOMATIC CONTAINER CRANE INSPECTION

Mohammad Shafi Bajauri*, Ahmed Alamouri, Markus Gerke

Institute of Geodesy and Photogrammetry, Technische Universität Braunschweig, (m.bajauri, a.alamouri, m.gerke)@tu-braunschweig.de

Commission IV, WG IV/7

KEY WORDS: Geodatabase, UAV, Crane-Inspection, Structural Health Monitoring

ABSTRACT:

Inspection of container cranes is an important task to maintain the all-day operation in harbours. In practice, manually carrying out the inspection process is not the best solution due to the complexity, time consumption, and cost. The manual inspection often is performed by specialists who capture images of areas of interest and then check and analyse these images visually. Once a critical area is spotted, industrial climbers are hired to check the part in-situ. The visual pre-inspection should be reliable and accurate, because the in-situ inspection is expensive, but to miss defected areas might lead to a failure of the crane. From this point of view, we came up with the main idea of the joint research project ABC-Inspekt which embodies an exploratory investigation in providing an automatic inspection of container cranes using the technology of Unmanned Aerial Vehicles (UAV). Images, which are captured systematically and regularly, are managed in a database to facilitate access. The access is done by an operator, but also by an image analysis approach which automatically identifies possible defects and stores them back into the database. In this paper, we introduce the database schema, the web frontend for data storage, access, and viewing, and show intermediate results for the data processing workflow.

1. INTRODUCTION

The civil infrastructure consists of basic systems and amenities vital for societies, e.g. highways, airports, buildings, bridges, and ports. The civil infrastructure deteriorates over time due to usage, aging, and natural or human-based incidents. This brings up the necessity to regularly maintain these structures.
Efficient Structural Health Monitoring (SHM) of the civil infrastructures has become a topic of interest for academic researchers, government institutions, and private industries in recent years. SHM is considered an important process to ensure the safety and serviceability of civil infrastructures such as bridges and container cranes (Rao et al., 2021).
The SHM topic itself is vast and recent technological advancement has made it possible to apply new methods, deploy and use various tools and sensors ranging from Unmanned Aerial Vehicles (UAVs) to smartphones to fiber optics and acoustic sensors across different domains (Alokita et al., 2019; Ozer and Feng, 2020).
Container cranes are used both at inland and seaports to load and unload heavy containers. When maritime trades compose a huge portion of the export and import volume of any country, the smooth operation of the ports and container cranes is of utmost importance.

Currently, the SHM of container cranes is primarily carried out through manual visual inspection by human operators (Hoskere et al., 2020). However, although vital, manual inspection has several drawbacks. During the manual inspection, the container crane has to be stopped. The industrial climber would climb different parts of the crane and inspect it visually which results in the consumption of a lot of resources, time and eventually increases the cost. There are also places in the complex container crane structure that is hard to access and requires a lot of physical effort. The weather conditions at ports e.g. strong winds, gusts, fogginess, frost periods which often happen or change very fast also increase the safety concern of the operators.
Early detection of changes e.g. color irritations, crack formation, surface bulges, rust accumulation on the surface areas of the container cranes is important to ensure the normal operation of the crane and to avoid possible incidents or breakage which might result in human and monetary loss.
Such unfortunate incidents have already occurred. On May 15, 2015, a container crane broke during the loading of "Maersk Karachi" at the NTB container terminal in Bremerhaven. It shows the importance of early detection of frail points in the container cranes and avoiding or minimizing the chances of repetition of similar incidents.

The cost, time, resource consumption, and difficulties of manual inspection have brought the idea of using camera-equipped drones (UAV: Unmanned Aerial Vehicle) and capturing images in a well-planned manner. The captured photos are then analyzed by the human operators to detect the existence of possible damages to the cranes. Although drones make the inspection process easier and faster, the visual inspection of images is prone to the operator and human error and its reliability and certainty would depend on the professionality and experience of the operator. As the number of captured images increases over time, the process of visual inspection would become more cumbersome.

Due to the above-mentioned limitations and challenges in manual inspection, we came up with the idea of conducting a joint research project in cooperation with a harbour operator. Utilizing, the flexibility of the drone technology coupled with Machine Learning (ML) approaches and databases, our research aims to create an automatic. intelligent workflow that encapsulates all these components to detect the defects on

* Corresponding author

container crane surfaces in a time-series manner, thus, reducing the cost, time, and resources for container crane SHM. To read more about the ML methods utilized in our research project and achieved results you may refer to our previously published paper (Maboudi et al., 2021).

In the following sections of this paper, we will elaborate and focus on the workflow for efficient data management and show the achieved intermediate results.

2. CONCEPTS AND METHODS

2.1 Overall concept

To efficiently carry out the visual inspection, some constraints need to be taken into consideration. The process first starts with data capturing using drones. The pilots will fly the drone manually to capture images of the several areas of the crane which are susceptible to defects e.g. crane junctions and bolting areas. These areas of interest would be called "neuralgic areas" hereafter. The manual flight is due to the complexity of the crane structure and several regulative limitations on flight height, the drone distance to the crane, and the weather condition such as strong wind which is common in the seaports. In addition, due to GNSS outage, multipath issue, magnetic field disturbance, maintaining a constant Ground Sampling Distance (GSD), and overlap between images which would be discussed in detail in the following sections is difficult. Flight automation within this context and environment could be another topic of interest for further research.

The captured images which would be hundreds per flight need to be managed carefully. The images and their metadata of the single flight (single epoch) are stored in the file-system-based structure and database and are used for image processing tasks. However, we also aim to provide the users the opportunity, to have access to the previous images of earlier flights so they can be used for historic comparison (multi-epoch) of defect detection processes in the same neuralgic area. In addition, the detected defects or results from manual and automatic processing would also be stored in the database.

Working with databases is not really of interest to the users. One of the important goals of this research is to provide the users with an easy and interactive tool to store, query, and visualize the data. Hence, developing a Graphical User Interface (GUI) that is connected to the database in the backend is also part of the concept.
The main components of our research concept are shown in Figure 1.

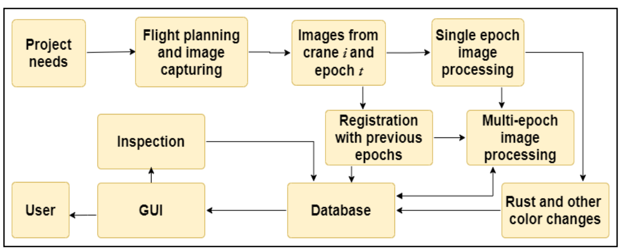

Figure 1. The overall concept of the project

2.2 Data capturing and pre-processing

Providing the base data about the crane area of interest is an important task because it directly affects the inspection process. From this point of view, a suggested solution for data capturing is to use drone technology. The flexible use of these devices enables capturing images that should have good quality and cover the studied area. In this project, the images play an important role on several levels. First of all, they are used for data acquisition, namely, to provide geometric information about the cranes. In addition, users/operators can get valuable basic data about the studied area based on non-geometric information like object color changes which are the main factors to be respected in the image analysis process. These considerations impose some requirements that should be respected by flight mission planning to capture images in a way that enables a reasonable inspection process in the context of image-based detection. The key requirements will be individually explained in the following

2.2.1 Flight planning

Several parameters have an influence on the actual flight planning:

Ground Sampling Distance (GSD)
Ground Sampling distance abbreviated as GSD plays an important role in object visibility and recognition in a digital image. GSD is also referred to as spatial resolution that reflects the distance between centers of two adjacent pixels measured on the ground or at the object, respectively.
In simple terms, GSD is the real size of the ground captured on a single pixel of the camera sensor usually expressed in cm/pix. The GSD is calculated using the relationship between the pixel size of the image sensor Ps and the image scale number m (Luhmann, 2017):

$$GSD = m \cdot Ps \qquad (1)$$

The flight height (or more generally the distance to object in our case) has a direct impact on the GSD values. In this joint research project, decisions about GSD, camera distance to the object, and lens size are crucial and directly affect the image processing results. To be able to recognize and detect the defects such as rust, color changes, irritations, etc. on crane surfaces automatically and intelligently, the following parameters values are required during the image capturing process:

Focal length (mm)	Distance to object (m)	GSD (mm/pix)	Footprint (m)
85	15	0.66	6×4
	25	1.10	11×7
50	15	1.13	11×7
	25	1.88	18×12
35	15	1.60	15×10
	25	2.68	26×17

Table 1. Parameters estimation based on three lenses of the camera Sony Alpha 7R IV with different distances to object

Image overlap
The photogrammetric bundle block adjustment requires a sufficient image overlap to realize a good linkage between images. In the context of flight planning, images can overlap in two directions: in the direction of flight called forward

overlap and between adjacent flight lines or so-called side overlap.

In practical applications, it is recommended to ensure an overlap in the flight direction (forward overlap) of about 60% or more and adjacent parallel flight strips (side overlap) about 30%. In practice, and especially with digital systems, a much higher forward overlap can be realized. Typically, 80% or more is possible. The side overlap is directly related to 1) the spacing of the flight strips, 2) the number of flight strips and 3) the flight time.

Conditions of visibility

For a reasonable detection of crane neuralgic areas, high object recognition is requested. In this context, we have to pay attention to visibility issues that are affected by factors like on the one hand weather conditions during the image capture process, and on the other hand the camera settings during flight missions; especially the depth of field, which needs to be high enough so we can acquire sharp images. For this reason, we list in a catalog to characterize the above-mentioned factors. Within this table, one can see that cloudy weather is the preferred condition for image capturing to minimize cast shadows in images. In contrast, dusty, foggy, steamy weather conditions may block the visibility of the areas of interest, hence, they are not recommended for flight missions.

Item		Ideal	Acceptable	Not recommended
Weather Conditions	Sunny		*	
	Cloudy	*		
	Foggy			*
	Dust			*
Depth of field	Low			*
	Middle		*	
	High	*		

Table 2. Catalog for visibility characterization

2.3 Data management

The captured images from different cranes and several neuralgic areas in a multi-epoch manner increase the complexity of data management. Over time, the amount and volume of data would increase and making it harder to keep track of the data.

Not only the raw images captured by the drone but also the results from the automatic defect detection process and the intermediate results e.g. annotations by the users generate new data that requires to be managed and integrated efficiently. As for photogrammetric purposes and acquiring the correct geometric and location information, the drone-based images need to be pre-processed. Huang et al. (2018) state that pre-processing of data results in the accumulation of the data that grows in complexity and needs to be managed and stored properly.

The data management concept is developed, considering, the drone-based imageries, the image processing results (e.g. detected defects), the intermediate results (e.g. annotations by the user) from several cranes and their neuralgic areas, all into a multi-epoch context. Besides, the concept is laid out in a way that could accommodate probable minor modifications and new requirements by the users.

In the following chapters, we describe how the data is stored and managed using a file-system-based structure and database.

The section would also include explanations for and details about the schema used for the database in our research.

2.3.1 Data structure and storage concept: The images captured by the drone are re-named uniquely and automatically using a Python-based tool developed for this specific task. The new names follow a pre-defined structure to include the EXIF and non-EXIF metadata. They contain the EXIF metadata of the date and time of captured image plus other non-EXIF necessary information such as crane type name (e.g. ZPMC), crane bridge name (e.g. CB1), the viewing side of the crane along the quay (e.g. Right), and neuralgic area name (e.g. A1 or B2). Because several images will be captured from a single neuralgic area during the flight, the 01, 02 … at the end of the image names are added to denote the sequential order in which images were captured. An example of re-named images is shown in Figure 2.

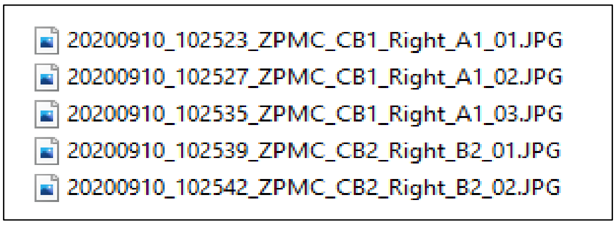

Figure 2. Example of the re-named images.

These re-named images are then stored in a pre-defined file system-based data server. The file system-based server follows a hierarchical structure where we go down from one image folder in the data server to crane type, crane bridge, the direction of images capture, and neuralgic areas/points subfolders. The same structure would be applied to image processing results.

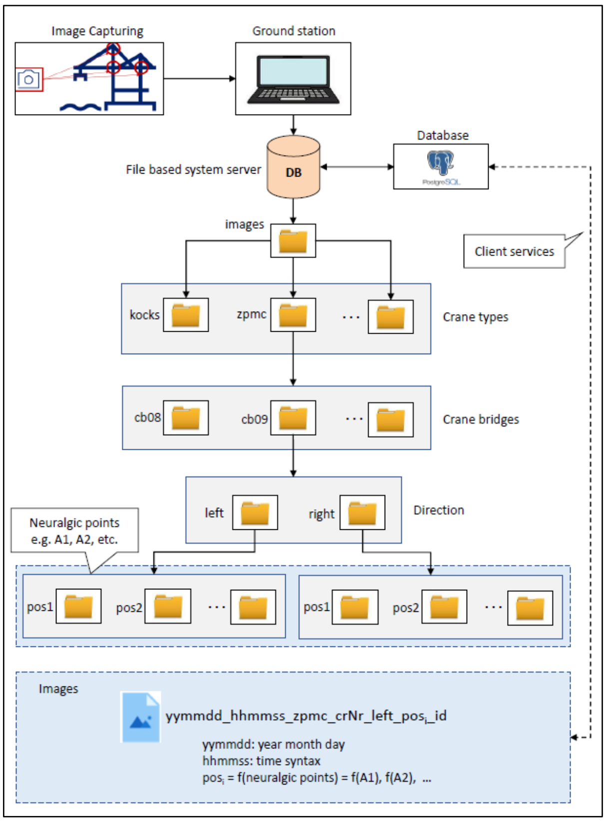

Figure 3. Concept of the data structure and image naming

The uniqueness of image names and storing them in relevant folders ensures the easy retrieval of the data later and avoids messing and confusing them with each other.

2.3.2 Database concept: A typical method to store large size drone-based imageries is to use a hybrid system where the images are stored in a file system and their metadata is referenced in a database management system (Fan et al., 2017). The benefit of this method is that drone-based images which are typically of larger sizes are not stored in the database, hence the database can perform faster. Technically, it is possible to store the images in most of the commonly used database management systems. However, this greatly impacts, the performance of the database in a negative way and can even cause database crashes, both during the storage and later during retrieval and query of the data. Sateesh and Das's (2019) experiments on storing large size images using file-system-based show that the approach performs 1.5 times faster than storing them inside the Oracle and MS SQL Server 2017 databases.
The high-resolution images we use in our research are an average size of 10 to 15 MBs. The number of these images grows to thousands as new images are captured over time. They will not be edited and will be accessible for read-only. The derived information would be vector layers or new images as well, hence, the file-system-based method is sufficient for our use. In addition, with the development of new technologies such as Solid-State Drive (SSD), the storage capacity of hard drives and their performance speed is growing quickly.

For the above-mentioned reasons, we decided to store the images in a separate file system-based multi-terabyte-sized data server and store their metadata and absolute storage paths in the relevant database tables. In the next section, the database schema utilized in our research project is explained.

2.3.2.1 **Snowflake schema**: Data modeling is an important task and one of the first steps that have to be taken when deciding the data structure. There are several logical questions that one has to keep in mind and try to answer while modeling the data. The questions are for example but are not limited to, what is my data? What is the data format, type, and how it is formulized or structured?
What are the relationships between the entities? How can the relationships be realized best and efficiently? What are the constraints in and among entities? What is the application of the data?

They are very important questions as answers to these questions lays the base for database schema sometimes also called database definition and database description. A database schema serves as a blueprint of the database and is often presented visually via diagrams. The schema which is an abstraction could be designed using specialized tools or just with a pencil and plain paper.
Considering the above-mentioned questions, the Snowflake database schema has been chosen for the data modeling in our research.

There are many other schemas e.g. flat model, hierarchical model, network model, the relational model which serve best for various types of data and needs. Each of them has its shortage and advantages. Different schemas have been developed through decades to fill the gaps and drawbacks of other schemas. Coronel and Morris (2016) list an overview of commonly used schema. Hence, after reviewing the literature and through examination of our data and understanding the inter-relationship of different database entities we realized that the snowflake schema is the best fit schema for our use.
The snowflake schema is one of the oldest database schemata and appeared during the '70s. You may refer to (Codd, 1971) as the earliest literature on it.

One of the aims of this type of schema is to avoid data duplication as much as possible. This is very helpful because it saves time and database storage capacity resulting in better performance. The data duplication may cause other issues and anomalies later in the database when the update or deletion of records is required.
This method of reducing or avoiding data duplication is known as "Database Normalization" and consists of several levels. Beside saving the time, it brings consistency and standard on storing the data. It is a well-researched and still evolving topic among database experts and researchers.
The snowflake schema not only provides us the opportunity, to avoid data duplication, redundancy, and dependency in database tables/columns, it also assists in data integrity and consistency. This creates space in database design for flexibility.

The snowflake schema is indeed a variation of another schema called "Star Schema" (van der Lans, 2012). In the snowflake schema, the main table is known as the "Fact Table" is connected to several other tables known as "Dimension Tables" via foreign keys. The main information is stored in the fact table and the other independent relevant entities are stored within separate dimensions.

The prime difference between star and snowflake is that, in a star schema, there is a fact table connected to several dimension tables but the dimensions themselves cannot have sub-dimensions. In snowflake schema, however, the structure can become more complicated as the dimensions are allowed to have sub-dimensions. Figure 4 shows the general structure of the snowflake schema. Although the diagram shows only one fact table, there is no limit on the number of fact, dimension, and sub-dimension tables in a snowflake schema, as it depends on what type of data would be stored in the database and how many independent entities exist in our data and how they are related, but generally, it is recommended to not add more dimensions and sub-dimensions and make the schema complicated unless it is necessary.

In addition, in both star and snowflake schemas, the independent dimensions e.g. dimension 1 and dimension 3 can not have relationships between themselves. This rule applies to the sub-dimensions as well.

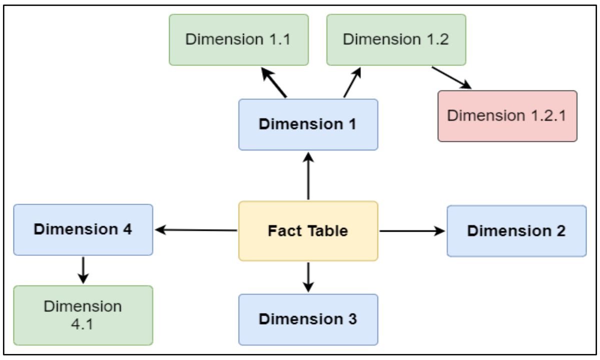

Figure 4. A Snowflake schema diagram

In short, the snowflake schema provides a multi-dimensional and hierarchical style design where each dimension stores distinctive information that is not directly related to other dimensions (van der Lans, 2012). This gives us the flexibility to update or delete data values in one dimension without affecting the other dimensions. Data normalization and integrity are also achieved as a result.

3. IMPLEMENTATION

At HHLA's Container Terminal Tollerort (CTT) in Hamburg seaport, the site where we conduct our research and implement the concepts, 14 container cranes manufactured in different years exist. The cranes are used throughout the year to load and unload heavyweight containers. It is a vital port not only for the German economy but also for Europe. The normal and undisrupted operation is very important. This is not feasible without the regular inspection of the cranes. In the following section, the achieved result so far from our research on the development of a geodatabase for efficient drone-based automatic container cranes inspection is presented.

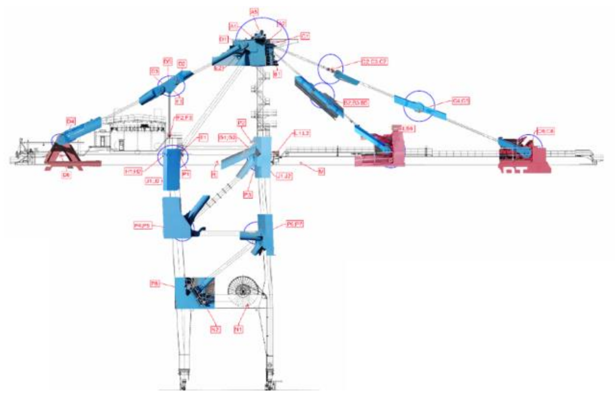

Figure 5. Schematic view of one crane bridge from crane type (ZMPC) with 13 neuralgic/critical areas

3.1 Data management results

Throughout the research, the aim is to use the available open-source solutions in the market and minimize the use of commercial tools as much as possible. Following this goal, for the geo-database development, the well-known PostgreSQL database development software is used.

It has been chosen because PostgreSQL has many extensions and plugins that make it powerful in different domains. Particularly for us, the PostGIS extension is of interest. Using this plugin enables us to store the geographic data e.g. coordinates of the images captured by the drone plus the orientation data in an efficient method. The PostGIS transforms a normal database into a geodatabase where the users can issue spatial queries. The queries would perform faster since the spatial indices created using PostGIS extension accelerate the process. It also supports vector and raster data (Marquez, 2015).

PostgreSQL is a tool that has been in development for many years and has become mature. It has compatibility with other tools such as QGIS. In addition, PostgreSQL follows the paradigm of a Relational Database Management Systems (RDBMS), which serves best the database schema concept presented in section 2.3.2.1.

The snowflake schema discussed earlier adopted for the database schema of our research is shown in Figure 6 and is implemented with the use of PostgreSQL and PostGIS extension. The database schema consists of two fact tables shown in yellow connected to each other namely *cranes_table* and *epoch_table*. They are used to store the information about cranes and each epoch, respectively. The *crane_table* is connected to several blue-coloured dimensions tables such as crane_type_table. crane_bridge_table, *neuralgic_points*.

The independent information about each crane like the neuralgic point/area and crane bridge names are stored within these dimensions. They are only referenced in using foreign keys (denoted with the *fk* prefix in the fact table) as attributes in the fact table. Hence, a modification, for example, if it is needed to update the neuralgic area name from A1 to A01, it is not required to update every record in the fact table. A one-time modification of A1 in the *neuralgic_points* table or dimension would update all the records in the database automatically.

The dimension may include their own attributes, for example, the rough shape description of neuralgic area e.g. circular, rectangular, etc can be stored as an attribute or column as well even after the database has been populated with the data.

An important detail regarding the data integrity is that if a record is inserted into the database with attributes that have not been already stored in the dimensions, for example, North instead of Left and Right, the insertion will fail and will not proceed. This prevents anomalies and stops incomplete, wrong, and unwanted records from being stored in the database.

On the other hand, the input images (drone-based images), the output data from the image analysis process (detected defects), and the annotation by the user/operator are all related and connected to the *epoch_table* fact table and each epoch would have an identifier number as the primary key.

An interesting detail is also the path attribute of the data. These are the physical storage paths of data such as the drone-based images, image processing results in form of file-based structure in the data sever discussed in 2.3.1.

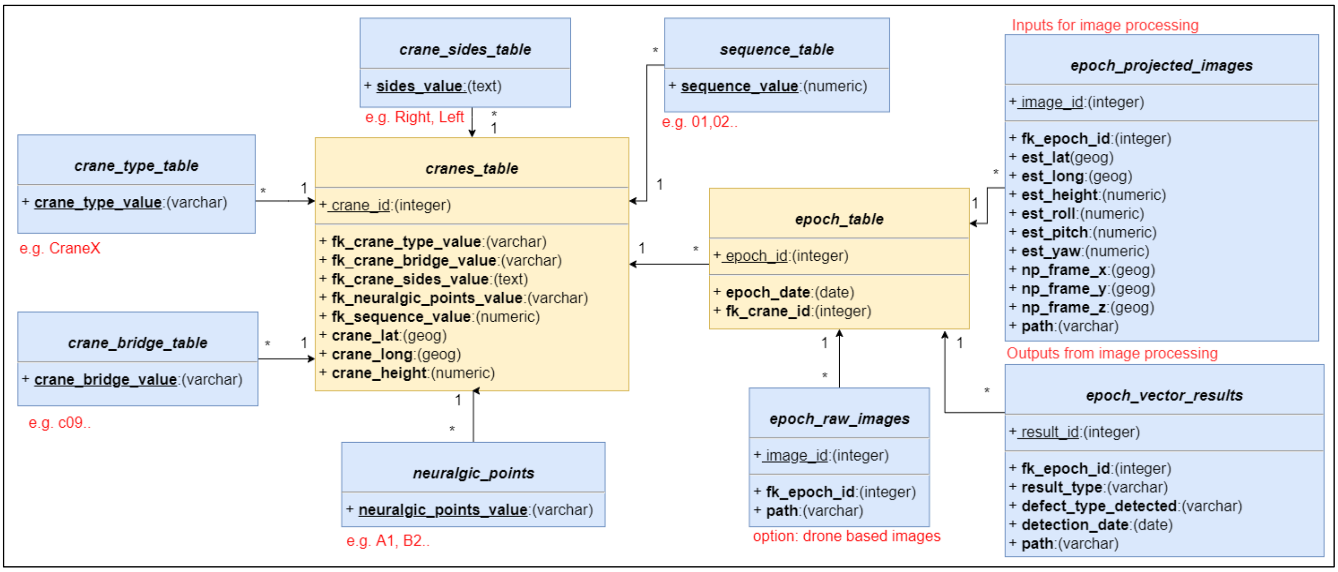

Figure 6. Database schema/structure

3.2 Graphical user interface

An easy-to-learn and easy-to-work graphical User Interface (GUI) is one of the main goals of our research. The importance becomes obvious as we know that human operators and users of the GUI are not professionals in photogrammetry, databases, or machine learning topics. They would only prefer to store, retrieve and view the data quickly and easily without having to learn the science and technical details behind the process of automatic defect detection or databases.

Therefore, a web-based user interface using *JavaScript, HTML, Bootstrap, PHP, CesiumJS, and Marker.js* tools and the relevant respective programming languages have been developed.
For security reasons, the user would be first required to register and then log in using their credentials. The GUI consists of three main tabs namely add data, retrieve data, and data in 2D/3D view. The functionality of the tabs is already clear from their names.

Using the add data tab, the users can add the data to be stored in the database tables. The data retrieve tab lets the users search for the data such as drone-based images and the image analysis results based on various search parameters and filters. For example, it is possible to search drone-based images based on the crane name type, crane bridge, the crane side (e.g. left, right), neuralgic area. The user also has the option to also add the date constraints for his/her search. A typical search query could look like searching for all images from crane X, crane bridge Y, left side, neuralgic area A1 from January 1st to the end of March 2021. A screenshot of GUI is shown in Figure 8. As a search result, the user will be provided with the lists of the returned images, their metadata such as longitude, latitude, height, roll, pitch, yaw, and storage paths. The user will also be given the option to view the returned results in 2D (the image itself) and 3D views. This will redirect the user to the 3rd tab to view the data.

In the 2D view, it is possible to just view the image or annotate it using the *Marker.js* tool. The annotation may include some shape drawing, commenting, writing a note. Storing the annotation within the database for future use by the user is under development.

CesiumJS is a virtual globe tool that is similar to Google Earth with the advantage that we can add our data such as images, point clouds, and shapefiles. The 3D view option integrated into the GUI enables the users to view the drone-based images and their orientation as it was captured with the drone in a 3D virtual globe. The 3D model of the crane and visualization of the data in CesiumJS provides the users with some contextual information for example how the drone/images were oriented during the capture or how far or close are the images relative to the crane. This will help in choosing an optimal image among many for further processes.

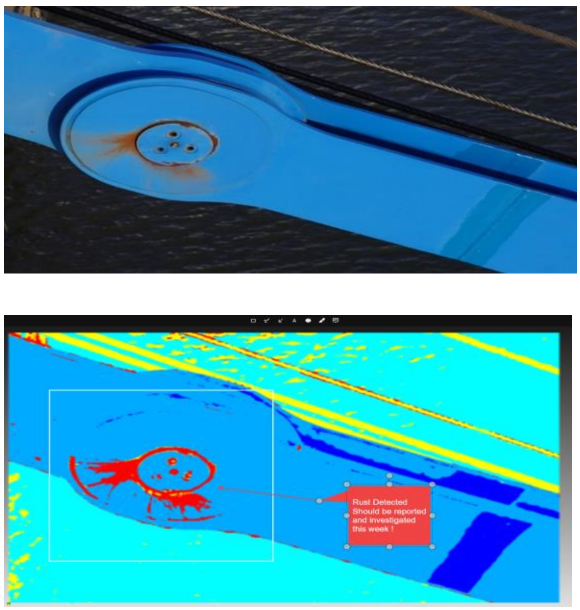

Figure 7. An example of a crane neuralgic area image captured by UAV (above) and its annotated result showing the rust detected in red (below) using ML methods. Both are stored in the geodatabase and can be retrieved by users.

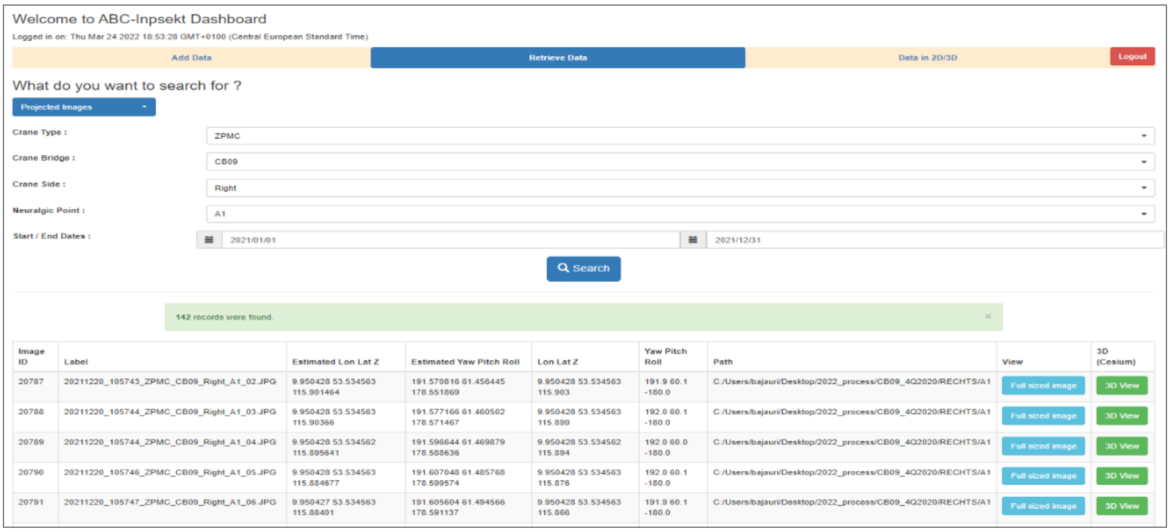

Figure 8. The internet browser-based GUI to add, query and view data (e.g. UAV imageries and crane defects detected using Machine Learning approaches).

3.3 Access and store the data with scripts

One other alternative for GUI is to access or store the data using scripts. To chain up the database access and storing process with the automatic ML-based defect detection processes in our project where primarily the Python programming language is used, code snippets have been written. These snippets would wrap up the SQL language into Python programming language using the psycopg2 library and the operator can query the stored data in the PostgreSQL database (e.g. drone-based images) based on the same modifiable search parameters that are available in the GUI. It will also enable pushing or storing the derived information and results back to the database without having to open the GUI. This enables a fully automatic workflow.

4. CONCLUSIONS

Structural health monitoring of the container cranes is vital for ensuring the normal operation of the cranes in seaports. This research project aims to present an automatic drone and machine learning-based multi-epoch approach as an alternative to the current manual expensive, risky, and time-consuming inspection method which is carried out by technical climbers. The complex 3D structure of the crane poses a challenge not only for the crane climbers but also for the drone capturing mission. Hence, careful flight mission planning is important to follow.

Hundreds of images are acquired from 14 cranes at different times of the year, the intermediate results from defect detection processes and the annotations by the users create the need for a well-structured and well-planned management. To this end, a geodatabase utilizing the snowflake schema has been developed using PostgreSQL and its PostGIS extension. The data themselves are stored in a file system-based structure in a data server and their storage paths in addition to the other attributes are stored in database tables. This approach enhances the database performance greatly.

A web-based user interface provides the user means to interact with the data without having to learn new skills and knowledge. Using the interface, the users can store new data, retrieve stored data, view the data in 2D and 3D, and annotate it with a few clicks.

The results presented in this paper are not final as we are still working on taking the advantage of geometric and spatial information available to further optimize the process of automatic crane inspection. One interesting and important idea under investigation is choosing the best image from a set of images taken during a flight for further process in image analysis. This choice of optimal image may be made based on the image and neuralgic area planes' normal comparison, the percentage of coverage of a neuralgic area in each image and finally the image characteristics such as brightness, saturation. The quantitative and qualitative evaluation and analysis of our proposed method is another future task.

ACKNOWLEDGEMENTS

This project is part of the Research Program "Innovative Port Technologies" (IHATEC) which is funded by the German Federal Ministry of Transport and Digital Infrastructure (BMVI). The authors would like to thank HHLA Sky and HHLA CTT for their support. The project ABC-Inspekt has funding id 19H19007B.

REFERENCES

Alokita, S., Rahul, V., Jayakrishna, K., Kar, V.R., Rajesh, M., Thirumalini, S., Manikandan, M., 2019. Recent advances and trends in structural health monitoring. Struct. Heal. Monit. Biocomposites, Fibre-Reinforced Compos. Hybrid Compos. 53–73. https://doi.org/10.1016/B978-0-08-102291-7.00004-6

Codd, E.F., 1971. Normalized data base structure, in: Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control - SIGFIDET '71. Association for Computing Machinery

(ACM), New York, New York, USA. https://doi.org/10.1145/1734714.1734716

Coronel, C., Morris, S., 2016. Systems, Database Design, Implementation, and Management, 12th ed. CENGAGE Learning, Boston, USA.

Fan, J., Yan, J., Ma, Y., Wang, L., 2017. Big data integration in remote sensing across a distributed metadata-based spatial infrastructure. Remote Sens. 10, 7. https://doi.org/10.3390/rs10010007

Hoskere, V., Narazaki, Y., Hoang, T.A., Spencer, B.F., 2020. MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure. J. Civ. Struct. Heal. Monit. https://doi.org/10.1007/s13349-020-00409-0

Huang, Y., Chen, Z.X., Yu, T., Huang, X.Z., Gu, X.F., 2018. Agricultural remote sensing big data: Management and applications. J. Integr. Agric. 17, 1915–1931. https://doi.org/10.1016/S2095-3119(17)61859-8

Luhmann, T., 2017. Nahbereichsphotogrammetrie, Grundlagen – Methoden – Beispiele, 4th ed. Wichmann, Berlin.

Maboudi, M., Alamouri, A., De Arriba López, V., Bajauri, M.S., Berger, C., Gerke, M., 2021. Drone-based container crane inspection: Concept, challenges and preliminary results. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. V-1–2021, 121–128. https://doi.org/10.5194/isprs-annals-V-1-2021-121-2021

Marquez, A., 2015. PostGIS Essentials. Packt Publishing Ltd.

Ozer, E., Feng, M.Q., 2020. Structural health monitoring. Start-Up Creat. 345–367. https://doi.org/10.1016/B978-0-12-819946-6.00013-8

Rao, A.S., Nguyen, T., Palaniswami, M., Ngo, T., 2021. Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. Struct. Heal. Monit. 20, 2124–2142. https://doi.org/10.1177/1475921720965445

Sateesh, B., Das, M.S., 2019. Performance analysis of database operations on large medical image datasets. J. Anal. Comput. 12.

van der Lans, R.F., 2012. Business Intelligence and Data Warehousing, in: Data Virtualization for Business Intelligence Systems. Elsevier, pp. 27–57. https://doi.org/10.1016/B978-0-12-394425-2.00002-2