

# CAAP Final Report

**Date of Report:** August 14, 2024

**Recipient Organization:** University of Nebraska–Lincoln

**Prepared for:** *U.S. DOT Pipeline and Hazardous Materials Safety Administration*

**Annual Period: From** *(09/30/2019)* **to** *(09/29/2023)*

**Contract Number:** 693JK31950006CAAP.

**Project Title:**

An autonomous UAS inspection platform for high-efficiency 3D pipeline/route modeling/change-detection and gas leak detection-localization

**Prepared by:** Drs. Zhigang Shen and Zhexiong Shang, University of Nebraska-Lincoln

**Contact Information:**

Zhigang Shen, Ph.D.  
Professor  
The Durham School of Architectural Engineering and Construction  
College of Engineering  
University of Nebraska–Lincoln  
A405EE, Kiewit Hall  
1700 Vine St., Lincoln, NE 68588-0500  
Email: [shen@unl.edu](mailto:shen@unl.edu)  
Phone: (402) 472-9470

---

## Contents

List of Figures .....	iii
List of Tables .....	viii
List of Acronyms .....	ix
1 EXECUTIVE SUMMARY .....	1
1.1 Project Deliverables .....	1
1.2 Education Accomplishments (student engagement).....	3
1.3 Publications.....	4
2 INTRODUCTION .....	5
2.1 Problem Statement.....	5
2.2 Related Studies.....	7
2.3 The Project Objectives.....	9
3 EXPERIMENTAL PROGRAM.....	11
3.1 Experimental Design and Test Procedure.....	11
3.1.1 Task 1.....	11
3.1.2 Task 2.....	12
3.1.3 Task 3.....	38
3.1.4 Task 4.....	39
3.1.5 Task 5.....	39
4 RESULTS AND DISCUSSIONS.....	40
4.1 Task 1: Testing and the results of the prototype of pipeline inspection data management system.....	40
4.2 Task 2: Develop the quality-based 3D CPP algorithm(s).....	45

---

4.2.1	Experimental Results of Task 2.1: Topology-based 3D CPP Algorithm .....	45
4.2.2	Experimental Results of Task 2.2: 3D CPP Algorithm for Truss Structure .	49
4.3	Task 3: Develop a 3D profile change identification and quantification module ....	51
4.4	Task 4: Develop an infrared image processing module for leak detection.....	52
4.5	Task 5: Field Validation.....	58
4.5.1	Results - Pipelines.....	58
4.5.2	Results - Truss Structure.....	61
5	CONCLUSIONS.....	65
6	FUTURE WORK.....	68
7	LESSONS LEARNED.....	69
8	REFERENCES .....	70

---

## List of Figures

Figure 1 Iterative UAS pipeline facility inspection procedures.....	5
Figure 2: The system design of the developed application in Task 1.....	11
Figure 3 The system overview of the implemented 3D pipeline inspection data management system. ....	12
Figure 4. Overview of the proposed method for aerial 3D reconstruction.....	13
Figure 5 Workflow of the model preprocessing algorithm on a residential building.....	14
Figure 6 Each admissible camera view $\boldsymbol{v}_s$ must cover a neighborhood of the surfaces $\boldsymbol{\gamma}_s$ (colored in gray) centered at $\boldsymbol{s}$ (colored in blue) .....	15
Figure 7 (a) Top: the current model surface (in red) and the clustered neighbor surfaces (in blue); Bottom: the topology graph, denoting the current view (in red) and the clustered candidate matches (in blue); (b) surface selection when more than one surface is connected by an edge. Top: the condition where $\boldsymbol{sn1}$ is considered as the neighbor, while $\boldsymbol{sn2}$ is neglected due to the intersection between the line $\boldsymbol{p1}$ and $\boldsymbol{pn2}$ ; Bottom: both $\boldsymbol{sn1}$ and $\boldsymbol{sn2}$ are neglected due to the existence of intersections in both conditions. ....	16
Figure 8 Overview of the multi-view optimization (The camera rays in red conceptually denotes the best matches selected from the candidates (camera rays in black) at each iteration) .....	17
Figure 9 Evaluation of the approximated camera model: (a) the mesh geometry at a selected (in pink) and its neighbors (in blue and $\boldsymbol{k} = \mathbf{3}$ ); (b) visibility detection using the original camera model and the computation load; (c) visibility detection using the approximated camera and the computation load; (d) the observation quality of the original camera model; (e) the observation quality of the approximated camera model.....	20
Figure 10 Workflow of sampling complementary cameras for global reconstruction: (a) identify the model surface with insufficient reconstruction (in dashed box); (b) sample admissible cameras (in light green) based on multi-view stereo geometric criterion, and the camera view with the best observation quality is selected (in green); (c) merge the newly sampled camera into the optimal camera set .....	21

---

Figure 11 Workflow of the trajectory generation: (a) Find the shortest TSP path without considering the on-site obstacles; (2) Find the path segments collided with the obstacles and reroute the segments using RRT*; (3) Smooth the path into the flight trajectory using B-Spline curve interpolation. ....	21
Figure 12 The selected case studies in this study: (a) <i>Res.</i> , (b) <i>Com.</i> , (c) <i>Ind.</i> , and (d) <i>Church</i> ...	22
Figure 13. Overview of truss bridge reconstruction based on the proposed method. ....	25
Figure 14. Model preprocessing of truss structure .....	28
Figure 15 Workflow of the proposed view planning method. ....	28
Figure 16 The collection of oblique orientations (arrows in yellow) from the initial one (arrow in cyan) based on the combination of different $\alpha$ and $\beta$ . Highlighted is the selected combination based on the experiments. ....	30
Figure 17 The geometry between a set of viewpoints $vi$ and a surface point $p$ where $vi \in Vp$ , $i = 1, 2, 3, 4$ . ....	32
Figure 18 Collision-free path planning procedure: (a) connect start (dot in red) and end (dot in blue) waypoints with straight line, check if collisions exist (dot in green); (b) Perform informed RRT* (in green) to reroute the path (in purple); (c) smooth the collision-free path using B-spline curve interpolation .....	34
Figure 19 visibility detection of a viewpoint to a truss bridge: The surface points in red are covered by the camera FOV (i.e., viewing frustum) but not visible (i.e., occluded); The surface points in blue are visible by the camera view, but the incidence angles are too large which often result in poor image triangulation; The points in green are visible and triangulable that passes the visibility detection. ....	35
Figure 20 (a) synthetic truss bridge and the detailed views of the selected regions; (b) the abandoned truss bridge to be surveyed, which is restricted to human access due to safety concerns (the detailed view) .....	37
Figure 21 2D view of the terrestrial laser scanned truss bridge. The 27 red dots denote the positions at each scan, and the links indicate the registration between the scans. The colors of the	

---

links denote the overlapping quality between the scans (Green: at least 75% overlapping; Yellow: at least 50% overlapping).....	38
Figure 22. A demonstrative field inspection case using PRCI inspection data. ....	40
Figure 23. The implemented hypothetical defects.....	41
Figure 24. The point cloud model interface of PRCI field pipelines using UAS based photogrammetry.....	42
Figure 25. Point cloud models of before (left), after (middle), and 3D changes (right) of a hypothetical flood event.....	43
Figure 26. Color-coded defect conditions by year (demonstration purpose only) .....	43
Figure 27. Detailed data attached to the 3D clickable object .....	43
Figure 28. The Geo-referencing function of the system using Google Map .....	44
Figure 29 The input models (including the reconstruction from orbit flights and the manually designed 2.5D) and the computed camera trajectories (in orange) of the three structures (in columns) using different methods: <i>Top: Overhead, Middle: NBV, Bottom: Ours</i> .....	45
Figure 30 3D reconstruction of the three scenes (in columns) using different methods: <i>Top: Overhead, Middle: NBV, Bottom: Ours</i> . A challenging region for aerial reconstruction is chosen in each scene for the detailed comparison. ....	47
Figure 31 Detailed comparison at the highlighted region in <i>Res.</i> <i>Top: Overhead, Middle: NBV, Bottom: Ours</i> . (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-coded $Error_{Acc}$ map;(c) enlarged view of the color-coded $Error_{Comp}$ map. The unit of the color maps legend is meter.....	47
Figure 32 Detailed comparison at the highlighted region in <i>Com.</i> <i>Top: Overhead, Middle: NBV, Bottom: Ours</i> . (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-coded $Error_{Acc}$ map; (c) enlarged view of the color-coded $Error_{Comp}$ map. The unit of the color maps legend is meter.....	48
Figure 33 Detailed comparison at the highlighted region in <i>Ind.:</i> <i>Top: Overhead, Middle: NBV, Bottom: Ours</i> . (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-	

---

coded $Error_{Acc}$ map; (d) enlarged view of the color-coded $Error_{Comp}$ map. The unit of the color maps legend is meter.....	48
Figure 34 The viewpoints and the trajectories generated by different methods. ....	50
Figure 35 Qualitative comparison on the 3D reconstruction of the synthetic bridge between different methods. ....	51
Figure 36. An example of quantifying and locating large scale volume change along the pipeline route .....	52
Figure 37. Results of the Cloud Compare using the 3D profile changes of the riverbanks underneath the truss bridge described in Section 4.5.2.....	52
Figure 38. Left: outdoor propane tank gas leakage detected under different environmental conditions.....	53
Figure 39. The Propane Tank Used in the Experiments .....	54
Figure 40. Local weather conditions of the experiment .....	54
Figure 41. The Coleman propane bottle used in Experiment 2 .....	55
Figure 42. Originally planned UAS system.....	55
Figure 43. Basketball net mounted UAS data acquisition system.....	56
Figure 44. Aerial view of the experiment layout (from 16 feet above ground).....	56
Figure 45. Baseline frame .....	57
Figure 46. Detected gas leakage profiles from different times/frames.....	58
Figure 47. The field test facility at PRCI, Houston .....	59
Figure 48. UAS flight paths by three different methods (the automatic method is based on the path planning algorithm described in Section 3.1.2.1) .....	59
Figure 49. Prelim Visual quality comparison of photogrammetry-based 3D reconstruction using images from different path methods. ....	60
Figure 50. Prelim geometry quality comparison of photogrammetry-based 3D reconstruction using images from different path methods when compared to the laser-scanning results.....	61

---

---

Figure 51 The flight trajectories between the Zheng et al. [24] and our method. The colors denote the trajectories of different UAVs/flights for the mission execution.....	62
Figure 52 The full bridge view and the detailed views of the final reconstruction of the real-world truss bridge generated using Zheng et al., (ref) and our method. ....	63
Figure 53 (a) The values of $FQ$ and $V *$ when different $\sigma$ are selected; (b) The values of $FQE$ under different combinations of $\alpha, \beta$ . Noted the legend denotes $\alpha, \rho$ .....	64
Figure 54 An example of complex tank structure.....	66

---

## List of Tables

Table 1 Statistics of the selected structures: (from left to right) the areas, the height, the input type, the triangular surface number of the resampled mesh, as well as the reconstruction pipelines. ....	23
Table 2. Summary of the input parameters, the symbols, the descriptions, and the default values .....	26
Table 3 Statistic of the computed image views and the flight distance of the synthetic structures using the different methods.....	46
Table 4. Quantitative comparison of the reconstruction in terms of the model accuracy (Acc.) and the completeness (Comp.).....	49
Table 5. The number of images and the flight distance on the synthetic bridge .....	50
Table 6 Quantitative comparison of the synthetic truss reconstruction between different methods. ....	50
Table 7 Comparison of the statistics on flight execution and 3D reconstruction using different methods. ....	62
Table 8 Quantitative comparison of the real-world truss reconstruction.....	63

---

## List of Acronyms

ACVD	Approximated Centroidal Voronoi Diagrams.
AGL	Above Ground Level
APSO	Accelerated Particle Swarm Optimization.
BIM	Building Information Modeling.
CAD	Computer Aided Design.
CPP	High-Density Polyethylene.
CVRP	Capacitated Vehicle Routing Problem.
DPSO	Discrete Particle swarm optimization.
FOV	Field of View.
GNSS	Global Navigation Satellite System
GSD	Ground Sampling Distance
ICP	Iterative Closest Point.
LKH	Lin-Kernighan-Helsgaun.
MVS	Multi-View Stereo.
NBV	Next Best View.
NDD	Nearest Neighbor Distance
OBB-Tree	Object-oriented Bounding Box Tree
OGI	Optical Gas Imaging
PHMSA	Pipeline and Hazardous Materials Safety Administration.
PRCI	Pipeline Research Council International.
PSO	Particle Swarm Optimization.
RANSAC	Random Sample Consensus.
TLS	Terrestrial Laser Scanning.
TSP	Travelling Salesman Problem
UAS	Unmanned Aircraft System.
UAV	Unmanned Aerial Vehicle.
VTK	Visualization Tool Kit

---

# 1 EXECUTIVE SUMMARY

## 1.1 Project Deliverables

- **Task 1.** Develop a prototype of pipeline/tank inspection data management and the integration module.
  - 1.1. Design and implement the proposed module, which allows access, retrieve, and visualize inspection data (in image, point cloud, and text data formats). The 3D graphical user interface was implemented based on Google Maps API.
  - 1.2. Test and evaluate the proposed module using hypothetical georeferenced pipeline.
- ✓ **Originally proposed deliverable: A conference paper** to describe the developed application and the demo application.
- ✓ **Actual deliverable:** The proposed data management module was implemented, tested, and evaluated using hypothetical inspection data (hypothetical means that the defect data are real inspection data of the PRCI pipe but the locations of the defects are merged into one section of the real pipe for easy demonstration purpose without affecting the actual function of the application. The demonstrated locations of the defects in the 3D model are hypothetical). Details are described in section 4.1.

*A conference paper was not written* about the development (as originally planned) due to **1)** there being not enough technical challenges in the development to warrant a conference paper; and **2)** the Covid pandemic negatively affected conference availability.
- **Task 2.** Develop the quality-based 3D coverage path planning (CPP) algorithm.
  - 2.1. Identify and investigate the appropriate parameters and their value ranges to balance the UAS inspection efficiency and photogrammetry model quality. Identify appropriate optimization algorithms for pipeline/tank 3D coverage path planning.
  - 2.2. Implement and test the developed 3D CPP algorithm through simulations and outdoor flight tests.
- ✓ **Originally proposed deliverables: One** peer-reviewed **journal paper** and **one** peer-reviewed **conference paper** on the topic of quality-based 3D CPP.
- ✓ **Actual deliverables: Two** peer-reviewed high-quality **journal papers**, on the proposed technology/algorithms were submitted and published in 2022 and 2023 respectively. The two journal papers were cited widely in such a short period of time. Please see section 1.3 for the detailed publication accomplishments.

As indicated earlier, due to the pandemic no conference paper was written on the topic. But the team did **One invited Webinar** to ASCE T&DI UAS Committee on April 6,

---

2023, about the developed CPP technology.

Task 2 is the technical core foundation of the proposed project. Two 3D CPP algorithms were developed to deal with different types of inspected structures, including truss structure, which was a challenging task to have a full coverage inspection of truss structure due to the number of linear structure members and occlusion.

- **Task 3.** Develop a 3D profile change identification and quantification module.
  - 3.1. Identify and evaluate the out-of-the-box photogrammetry software in terms of its capability and accuracy in processing a large inspection pipeline dataset and create demonstrative before-and-after 3D pipeline/route models.
  - 3.2. Develop a 3D profile change identification and quantification module to allow automatic before-and-after event comparisons of 3D models to identify the change locations and change quantities. The changes can include land movement, third-party excavation, pipe displacement, scour erosion, etc.
  - ✓ **Originally proposed deliverables:** **One conference paper** on the topic about algorithm and coding of the application.
  - ✓ **Actual deliverables:** A free opensource software Cloud Compare was identified and evaluated (<https://www.danielgm.net/cc/>) to be used to identify 3D profile changes between and after potentially hazardous events to pipelines and the associated right-of-way. Since the application has a GNU General Public License (GPL), the application including the source code (<https://github.com/cloudcompare/cloudcompare>) is free to use for public. **As a result, no conference paper is needed** to report on the development of the topic, including the algorithm and the source code.
  
- **Task 4.** Develop an infrared image processing module enabling pipeline/tank leak detection and mapping on the geo-referenced 3D models.
  - 4.1. Develop a deep learning-based infrared image processing module to automate above-ground leak detection based on the Joule-Thomson effect, the gas spectral response effect, and the temperature difference between the gas and the ambient temperature.
  - 4.2. Map leaks. By fusing the infrared detecting results and the 3D pipeline model generated in Subtask 3.1, the detected leaks will be automatically mapped on the surface of the 3D pipeline model.
  - ✓ **Originally proposed deliverables:** (1) Validation results of the deep-learning-based detection model; (2) mapping algorithm codes; (3) **one conference paper** and **one journal paper** to report the outcomes.
  - ✓ **Actual deliverables:** (1) the originally proposed deep learning-based gas leakage

---

detection method was not developed due to the demonstrated effectiveness of the new method. Instead, a direct infrared image-based detection method was developed. (2) the deliverable of mapping leakage spots of the pipeline was merged and achieved in Task 1. A note is added at the end of Page 40. (3) a conference paper is still under development on the infrared image-based gas leakage detection technology based on the preliminary data. A journal paper publication is still expected when more data is collected and processed.

- **Task 5.** Field Validation, Performance Evaluation, Closeout Subtask
  - 5.1. Conduct a full-scale field validation and performance evaluation of the developed modules in Tasks 2, 3, and 4. A Subject Matter Expert and partners from Black Hills, Kiewit, and/or METEC will be included to provide input. Subtask
  - 5.2. Final report.
- ✓ **Originally proposed deliverables:** (1) Final report; (2) Commercialization Plan.
- ✓ **Actual deliverables:** (1) the Final Report is completed, which includes the field validate tests conducted at PRCI at Houston. PRCI worked as a Subject Matter Expert as suggested by PHMSA. Please note: the field test in PRCI did not include a UAS-based gas leakage test due to the PRCI facility constraints (PRCI did not have a gas leakage facility at the time of the field tests). (2) More investigations are needed to determine which part of the research outcomes can be feasibly commercialized if any.

## 1.2 Education Accomplishments (student engagement)

- Undergraduate researcher
  - (1). Gabriel Clark, software engineering major, NASA and Microsoft interns, had been working on the project from September 2019 to July 2023. Gabriel established his own start-up company that works on UAS and image processing, and 3D reconstruction of civil infrastructure inspections.
- PhDs in Construction Engineering
  - (1) Houman Kosarirad, PhD student at University of Nebraska-Lincoln, worked on this project in Spring 2022.
  - (2) Chongsheng Cheng, Associate Professor, Chongqing Jiaotong University, had worked on this project from September 2019 to June 2020
  - (3) Zhexiong Shang, Associate Professor, Hainan University, had worked on this project from January 2019 to December 2019
- Postdoc fellows
  - (1) Chongsheng Cheng, Associate Professor, Chongqing Jiaotong University, had

---

worked on this project in from July to December of 2020

(2) Zhexiong Shang, Associate Professor, Hainan University, had worked on this project from January 2020 to December 2022

### 1.3 Publications

- Journal/conference publications.
  1. Shang, Zhexiong, and Zhigang Shen. 2022. "Flight Planning for Survey-Grade 3D Reconstruction of Truss Bridges" *Remote Sensing* 14, no. 13: 3200.  
<https://doi.org/10.3390/rs14133200>
  2. Shang, Z., Shen, Z. 2023. "Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures". *Complex Intell. Syst.* 9, 909–926.  
<https://doi.org/10.1007/s40747-022-00831-5>
  
- Conference presentations/posters/invited talks, etc.
  3. Shang, Z., Shen, Z. Webinar to ASCE T&DI UAS Committee on April 6, 2023, "Topology-based UAV path planning for automated multi-view stereo 3D reconstruction of complex structures"
  4. Shang, Z., Cheng, C., Clark, G., Shen, Z. (2020) "Detecting, locating, and mapping internal gas pipeline corrosion using thermography and photogrammetry" poster at the 2020 PHMSA Pipeline R&D Forum, February 19-20, Arlington, VA

---

## 2 INTRODUCTION

### 2.1 Problem Statement

Maintaining the safe and efficient operation of 2,734,580 miles of gas pipelines (including 1,356,358 Distribution Main, 965,167 Distribution Service, 112,120 Gas Gathering, and 300,936 Gas Transmission) in the US is a daunting task, especially when faced with aging gas transportation infrastructure and increased demands on gas usage. Unmanned Aerial Systems (UAS) mounted with various types of sensors and cameras can substantially extend pipeline operators' view ranges, especially in locations that are normally difficult to access. As a result, UAS-based pipeline/tank inspections have attracted many attentions with anticipations of improving safety and efficiency in inspecting hazardous areas, reducing maintenance costs, reaching difficult and/or remote locations[1], and dispatching emergency responses [2]. Equipped with 4K/8K visual cameras and high-resolution infrared cameras, UAS provides opportunities for highly efficient and frequent asset surveying without endangering inspection staff or suspending operations. Not only has UAS demonstrated its value in inspections under API-570/653 in detecting faults such as hairline cracks, corrosions, gas leakage, and others, but UAS has also proven to be an efficient data collection tool for risk-based inspections of oil/gas assets under API 579/580 [3]

Despite these clear advantages, current UAS inspections are constrained by line-of-sight and complex path specification issues when inspecting complex 3D objects [4]. Current 2D/2.5D waypoint paths planning, which allows UAS to fly beyond line-of-sight autonomously, has difficulties handling the detailed inspection tasks of complex 3D structures [4] in which UAS needs to closely follow the surface shapes of each individual structure.

Numerous 3D coverage path planning (CPP) methods were proposed [5]to automatically generate inspection paths to cover all or most surfaces of the inspected 3D structures, and the applications of these CPP can be found in ship hull inspection [6], powerline inspection [7], wind turbine inspection [8], and other areas [9]. Typical UAS pipeline/tank inspection procedures include multiple iterative steps (Fig. 1) such as mission planning (what to inspect), path planning (how to inspect), image capturing, defect detection and localization, and inspection data storage and management. While there has been some siloed progress in autonomous pipeline inspection (e.g., automatically identifying right-of-way encroachment), a systematic autonomous UAS pipeline/tank inspection platform that addresses the full-loop pipeline/tank inspection procedure as shown in Figure 1 has yet to be seen.



**Figure 1 Iterative UAS pipeline facility inspection procedures**

The lack of such an end-to-end autonomous UAS solution has hindered the oil/gas pipeline

---

industry from realizing its full potential in enhancing pipeline safety and reducing methane emissions due to leakage. When a pipeline leak has exuded gas with infrared absorption rates different from ambient air, thermographic patterns of gas leaks can be tracked by infrared images [10].

Many existing studies have established the validity of optical gas imaging (OGI) methods in detecting leaks from valves, pipes, and other equipment in both above ground [11] and buried conditions [12]. Other studies have suggested that thermographic imaging can further be used to evaluate oil spills [13]. Despite the progress in OGI detections, most OGI leak detection studies were based on stationary or handheld camera-base, the effectiveness and performance of UAS-based OGI in gas leak detection have yet to be systematically investigated. Studies indicated that the standoff distance and shot angle between the infrared camera and the damaged pipe or tank spots significantly affects the performance of OGI leak detection measured by the minimum detectable leak rate (MDLR) [14]; however, even for experienced UAS operators it is a demanding task to control UAS at consistently close distances along the complex 3D surfaces of the inspected pipelines or tanks, either manually or using 2/2.5 D waypoints. Clearly, to achieve consistent, high-quality thermographic inspection outcomes, high-quality 3D path planning tools are a must.

#### *Challenges associated with UAS-based multi-view stereo (MVS) 3D reconstruction.*

Significant growth of unmanned aerial vehicle (UAV) applications has been witnessed in land surveying [15], urban surveillance [16] post-disaster evaluation [17], and structural inspections [18]. High-accuracy 3D reconstruction is often required for many UAV applications. Meanwhile, multi-view stereo (MVS) [19] was recognized as a viable tool for reconstructing high-quality three-dimensional (3D) models utilizing images from different viewpoints. However, using aerial MVS to reconstruct 3D models of large complex structures remains a challenge primarily due to the lack of appropriate path planning tools to determine the optimal UAV cameras' poses to acquire MVS images of the surveyed structures.

In this project, a topology-based UAV path planning method is presented and tested to improve both the scalability and reconstruction quality of the existing 3D path planning methods. The proposed method outperforms existing methods in the presented cases in terms of the computational cost, the reconstruction quality, and the camera initialization. Unlike the prior works that recursively select/adjust the cameras from a user-defined discrete candidate set, the proposed method continuously optimizes a set of randomly sampled cameras in the continuous space.

The core of the technique lies in introducing the topology parameters to guide the maximization of the two-view quality metric at each model surface solely based on the closely located cameras. The method designs one admissible camera to cover each model surface and, at the same time, cluster its neighbor cameras based on the topology of the surface connectivity. The global view planning problem is then decomposed into a collection of local optimization problems to find the optimal matched camera positions/orientations within each cluster such that the MVS quality is maximized. The reformulated optimization is computationally tractable and inherently parallel. Therefore, an adapted particle swarm optimization (PSO) framework is used to seek optimal solutions (i.e., camera views) efficiently. The result is an optimized camera trajectory that a

---

rotorcraft UAV can fly for high-quality 3D reconstructions of the inspected structure.

### *Challenges associated with UAS-based truss-structure 3D reconstruction.*

UAV inspections, either through remote control or simple automated waypoint flight paths (i.e., orbit, lawn mowing, etc.), are challenging to achieve the desired quality and completeness of the 3D reconstruction of truss structures [20]. The primary challenge is how to handle the complexity and self-occlusion problem of the truss geometry under the constraints of camera field-of-view (FOV), safety clearance, and flight duration.

A new UAV flight planning method is proposed to overcome these challenges by finding the optimal trajectories that maximize the reconstruction quality at truss surfaces. Unlike the NBV methods [21][22][23] where the vantage viewpoints were selected from a pre-determined discrete candidate set, the proposed method iteratively resamples the whole candidate set in the continuous space, increasing the searchability of finding the optimal viewpoints subset. Additionally, the method computes the shortest flight paths subject to the UAV capacity constraints (i.e., battery capability, autopilots limitation), enabling the more automated truss bridge reconstruction by single/multiple UAVs. Evaluation of the proposed method includes both a synthetic and a real-world truss bridge. The results showed that the proposed method outperforms both the recent sweep-based method [24] and the state-of-the-art NBV [23] in terms of the higher model quality with the increased automation and the fewer images/distance traveled in the air.

## **2.2 Related Studies**

In MVS, images must satisfy quality criteria [25] to obtain high-quality models. These criteria may vary slightly in different MVS algorithms, but there are some standard criteria such as the coverage/visibility, resolution, incidence angle, baseline, and parallax [26][27][28]. When a 3D model of the surveyed structure is available (i.e., 3D Computer-aided Design (CAD), building information modeling (BIM), 2.5D digital elevation model, rough photometric reconstruction), the UAV's camera views/paths can be designed in a model-based fashion where the optimal trajectories can be computed by maximizing the MVS quality at each observed surface of the 3D structures [29].

Among the model-based path planning methods, Next-Best-View (NBV) was recognized as the state-of-the-art [30][31][32][33]. NBV is a recursive view selection process: at each step, NBV selects the best view from an ensemble of candidate camera views based on a quality criterion. This approach has been used in several existing studies: from the early works where the focus was the indoor active range imaging in a controlled environment [34][35], to the more recent applications of the outdoor aerial MVS reconstruction [36][37][38][39][40]. Roberts et al. [38] modeled the MVS quality of a UAV trajectory as to how well the cameras cover the structure's surface light field. The method formulated the coverage function as submodular and found the near-optimal trajectories by (1) finding the best orientation at each candidate camera and (2) selecting the best subset of the camera positions based on the UAV constraints. Hepp et al. [39] also employed the sub modularity and used the information gain (IG) to quantize the goodness of

---

each camera viewpoint. Based on the volumetric representation of the structure's geometry, the method combined the optimization of camera positions and orientations into an integrated view selection process.

One issue in these NBV methods is that the greedily selected viewpoints do not guarantee the optimal quality. In addition, these methods assumed that a single camera view could determine a point in 3D space, which contradicts the basic MVS quality assurance principle that at least two different views are required. Smith et al. [41] studied a reconstruction heuristic that measured the MVS quality based on the two-view geometric analysis [26] [42]. The two-view quality heuristic measures not only the visibility/coverage of the cameras to the surface points but also the geometries concerning other views. Based on the heuristic, the method adaptively adjusted the cameras from a user-defined flight path for high-quality reconstruction of urban scenes.

Zhou et al.'s [43] study started with a dense camera set sampled along the normal of the urban structures. Based on the quality heuristic, the method greedily selected and adjusted vantage cameras in sequence to reduce the image redundancy while increasing the MVS quality. One major limitation of these studies is that the final reconstruction heavily relies on the quality of the initialized cameras because the methods only adjusted the cameras locally in space.

Therefore, the camera initialization can be a challenge when dealing with more complex geometries (such as residential buildings, chemical plants, etc.) where sharp corners, concave regions, and uneven surfaces exist across the model, making the initialization of good camera viewpoints difficult. Furthermore, the quality heuristic was evaluated based on all available cameras in the existing methods, which makes it computationally expensive since it requires quadratic operations to find the best quality matches at each point. The path optimization based on this heuristic requires re-evaluating the model surface quality at each iteration. These optimization methods often include infeasible view candidates per multi-view stereo criteria, where computation time can grow significantly as the size of the surveyed objects increases.

Over the past years, advanced flight planning solutions have been proposed for the automated inspection and 3D reconstruction of bridges. Most of the works configured the camera viewpoints to back-and-forth sweep the structural surfaces efficiently [27][28][29][30][31]. For example, Morgenthal et al. [16] densely reconstructed bridge piers in three steps: First, the method sliced each pier structure vertically at given intervals. Then, a dense set of horizontal camera views was sampled at each slice. Finally, the camera views across the sliced structures were connected vertically by a spiral path. Phung et al. [29] configured the orthogonal viewpoints along bridge surfaces based on the required ground sampling distance (GSD) and image overlapping. Discrete particle swarm optimization (DPSO) was employed to find the shortest path to connect these viewpoints. Bolourian and Hammad [31] scanned the bridge deck with varied densities of the camera views based on the critical level of defects at the deck surface. The method used the ray-tracing algorithm to avoid the occlusion caused by the on-site obstacles, which guarantees the quality of the collected images. A limitation of these sweep-based techniques is that the methods assume the structures majorly consist of planar surfaces (for viewpoints to scan along). This assumption does not hold for the truss structures due to the geometrical variances and the complexity of the truss components. In addition, sweeping the image views along the structural

---

surface often encourages collecting overly redundant images, which reduces the efficiency during both the inflight image acquisition and the post-flight image processing without increasing the reconstruction quality.

A notable limitation of these NBV techniques is that the methods relied on user-defined discrete candidate viewpoints. Unlike the 3D reconstruction of buildings where the candidate set can be defined as Overhead views naively surrounding the building geometries, trusses are composed of many slim, self-occluded, and non-planar components (i.e., beams, girders, connectors, etc.). Thus, it is difficult to determine a suitable-sized candidate set while ensuring complete coverage at every truss side.

### **2.3 The Project Objectives**

In this project, an autonomous UAS-based pipeline inspection and gas-leak detection platform is proposed:

- (1) to substantially enhance the quality and efficiency of UAS pipeline/tank inspections so that it can be widely adopted in a large amount of routine inspection tasks.
- (2) to evaluate the performance of UAS-based passive optical gas imaging (OGI) in gas leak detections.
- (3) to enhance the performance of UAS-based OGI through quality-based 3D path planning. The developed UAS-OGI platform is also expected to be able to conduct autonomous leak-detections of buried gas pipelines.

The first objective was achieved through an integration of a Google-Map-based 3D pipeline/route inspection database, a quality-based 3D UAS path-planning module, and a 3D photogrammetry model-based profile-change analysis module.

The second objective was achieved through experiments in outdoor thermographic leak detection using a high-performance thermal camera (FLIR A8303sc).

The third objective was achieved by using the path planning algorithms developed in objective 2 with the outcome of an outdoor gas leakage detection experiment.

The intended main target is in-service above-ground transmission or gathering gas pipelines/tanks and their routes. Using the proposed autonomous UAS inspection platform, not only are leaks and locations expected to be detected and mapped, but the platform itself can also be used in numerous areas of interest to the Office of Pipeline Safety such as identifying and quantifying geological impacts, right-of-way encroachments, river-crossing changes, land movements, third-party excavations, and pipeline mapping verification. The outcomes of the proposed autonomous UAS inspection, detection, and localization platform are anticipated to significantly enhance pipeline safety and transform pipeline inspection and maintenance operations through autonomous UAS

---

inspection and leak detections. In reference to the identified pipeline technology and general knowledge gaps in the Event Summary Report of the PHMSA 2018 Pipeline R&D Forum [44], the proposed approach is expected to address all three (technology and general knowledge) gaps identified by Work Group 2 – Remote Sensing/Leak Detection-Mitigation:

Gap 1 – Remote Sensing and Leak Detection Platforms that can deploy multiple sensor types,

Gap 2 – Further Validation of Remote Sensing and Leak Detection Technologies under realistic and differing operating conditions, and

Gap 3 – Improving Data Collection, Normalization, and Integration methods to enhance risk assessment tools for decision-making.

---

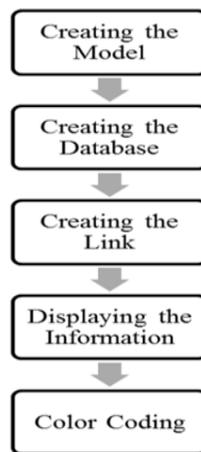
## 3 EXPERIMENTAL PROGRAM

### 3.1 Experimental Design and Test Procedure

The primary experimental design in this project is to use a combination of numerical simulations, lab tests and field validations, depending on the nature of the specific tasks.

#### 3.1.1 Task 1

In Task 1, the main task is to develop a prototype application to show the concept of integrating inspection data with 3D models of the inspected pipeline systems. As such, the experiments are focused on coding and debugging of the application and the user interface of the application. The focus of this task is more on the functionality of the application, and not on how user-friendly the interface is. The experimental design of the system is shown in the following figure.



**Figure 2:** The system design of the developed application in Task 1.

We developed the pipeline asset management system by using 3D pipeline models to manage pipeline inspection data. Figure 3 shows the general framework of the implemented system. Sketchup was used as the 3D graphic interface to interact (input, output, selection, visualization, and geo-referencing) with pipeline inspection data. This type of technology is called digital twin by some groups of researchers in the civil engineering field. The identified pipeline defects, such as dents, deformation, leakage can be geo-referenced to the 3D model using Google Map API, so that a more informed and more accurate evaluation can be conducted.

The system is developed with three major segments, they are (1) database; (2) 3D graphical user interface; and (3) system control unit. 3D graphical user interface is the major component to visualize the 3D model and provide user intersection. SketchUp is selected as the base engine for 3D model representation. The system control unit is the core component of this system. It is developed as a set of callback functions such that the interaction between data management stack and system user interface can be achieved. A screenshot of the system's UI is shown in Fig. 3 of the proposal. System control unit is a web-based API. The API files are developed with Ruby

language where the major functions to query the database, interact with web interface, and manipulate the 3D model are developed.

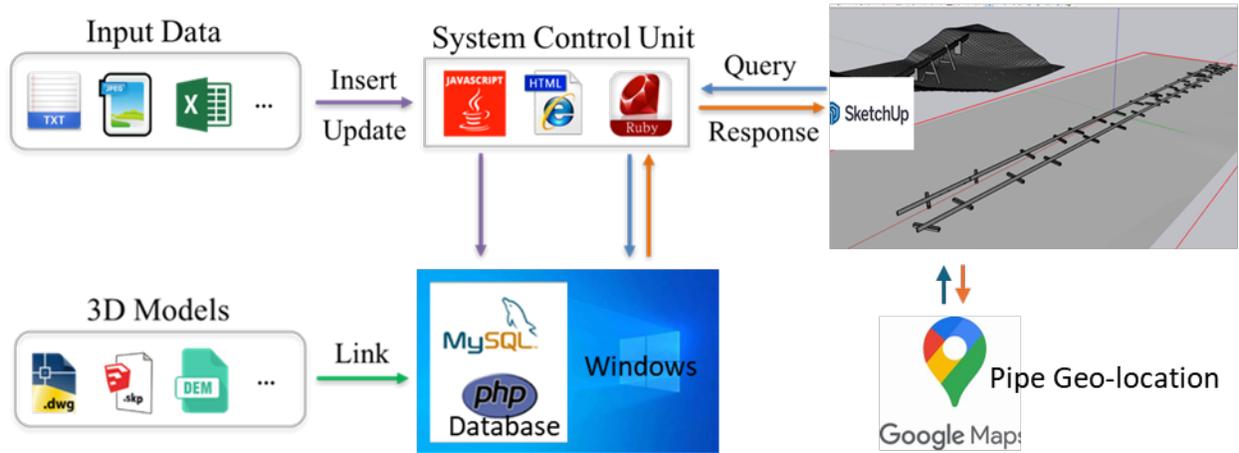


Figure 3 The system overview of the implemented 3D pipeline inspection data management system.

### 3.1.2 Task 2

Task 2 was focused on developing 3D coverage path planning (CPP) algorithms, which can be used to control the UAS to conduct autonomous pipeline inspections while maximizing the inspection quality. The experimental design of this task is through 1) numerical simulations using digital 3D environment and 3D object models; and then 2) through field tests to compare the field results to the ground truth if available.

#### 3.1.2.1 Task 2.1 Topology-based Path Planning Method for pipeline structure, gathering facility, and buildings [127]

The building models used in this section are just examples of 3D solid geometric objects with complex topographical structures (in Figure 4, Figure 5, and Figure 12 etc). The CCP algorithms developed and tested in this project are not expected to perform differently when applied to pipeline facilities, which are typically geometrically simpler than the tested building examples, except in the case of truss structures that will be covered separately in Section 3.1.2.2. As such, we expected same or better 3D reconstruction results when applied to pipeline facilities in terms of accuracy, consistency, and robustness performance. Another important reason to use these published building models in the projects is for benchmarking our developed algorithms against the existing CCP algorithms to demonstrate the quality improvement of our algorithms compared to the existing algorithms. So, apple-to-apple-comparison can be conducted between algorithms.

Figure 4 is an overview of the proposed method. Given a rough model of the target structure and the selected UAV, the method starts by processing the input model into a triangular surface mesh. The processed mesh contains the desired number/size of the surfaces, enables the per surface coverage planning. A topology-based coverage algorithm is then proposed to design one admissible camera to cover each model surface while clustering the candidate matching cameras to choose from. Based on the presented coverage algorithm, optimal camera views are found by

simultaneously maximizing the two-view quality metric within each cluster. The proposed optimization method integrates an approximated camera model and a two-view reconstruction metric into a parallelizable PSO, enables a randomly initialized camera set been optimized in an efficient and continuous way. After the optimization, a fast algorithm is presented to assess the MVS performance of the optimal cameras and adaptively generate complementary cameras, guaranteeing reconstruction quality even at the challenging regions. Finally, the cameras are converted into a shortest flight trajectory, which is then uploaded into the ground control stations for the automated UAV inflight mission execution and the postflight multi-view stereo reconstruction. In the following, the detail at each step is discussed.

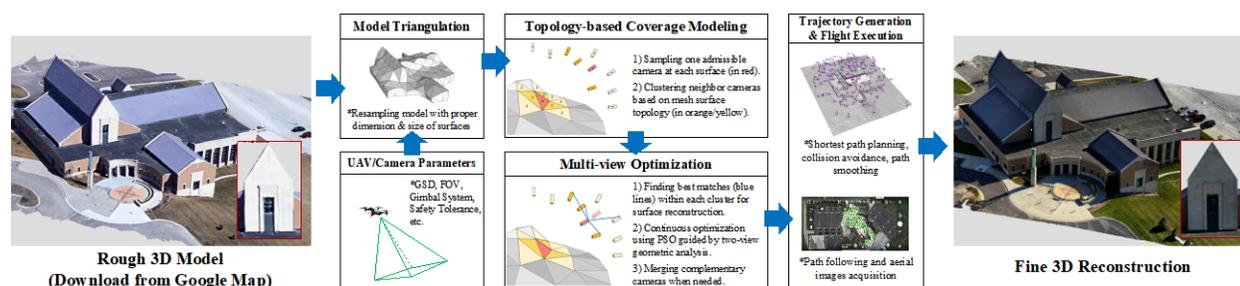


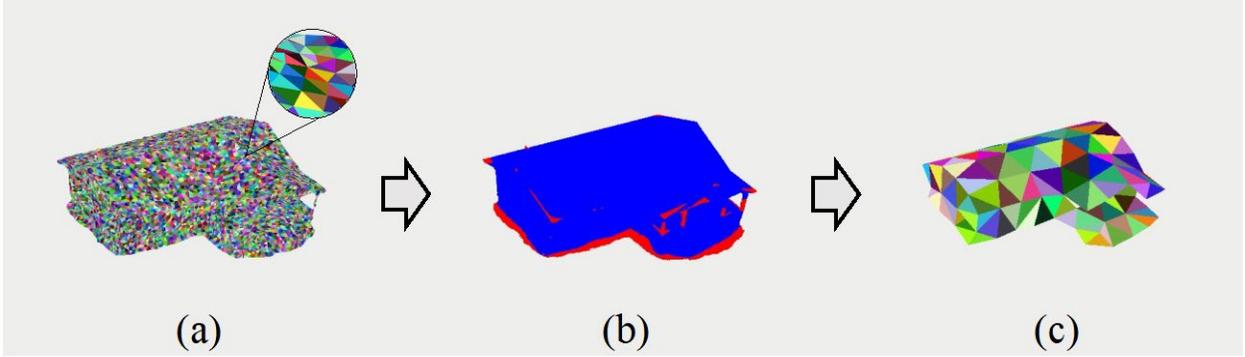
Figure 4. Overview of the proposed method for aerial 3D reconstruction

### *Input model Triangulation.*

A rough 3D model is needed as an input model for the coverage planning. The rough model can be downloaded from Google Earth/Map, or can be reconstructed from a simple nadir (2D) flight if a Google Earth/Map model is not available. Then, the raw input model needs to be triangulated such that each model surface can be covered by one viewpoint with the specific sensor model [45][46]. Thus, preprocessing of the input model is needed to convert the raw model (in point-cloud format or 3D CAD/BIM format) into a triangulated surface mesh, and to make the sizes of triangulated surfaces more evenly to be covered by the selected cameras. The proposed triangulation method is an iterative mesh resampling algorithm and is developed based on two criteria: (1) the triangulated model must be geometrically consistent with the raw input model; (2) each triangulated surface must fit into the camera’s FOV at the required distances (i.e., GSD); The first criteria guarantees the high fidelity of the processed model compared to the inputs while the second criteria ensures the baseline quality of the collected images.

Starting with a baseline surface size, the method incrementally resamples the input model into a triangular mesh (with the reduced surface sizes) until the mesh reaches the required fidelity. In this study, model fidelity is computed as an error metric that measures the nearest neighbor distance (NND) of the point correspondences between the resampled mesh and the input model. Based on this metric, a residual map is generated, and the ratio of outliers is computed based on a distance measurement  $\omega$ . If the ratio is less than a pre-defined threshold  $\tau$ , the mesh is considered as of high-fidelity. Otherwise, the surface size is reduced (at certain step  $\lambda$ ) and the above steps are repeated until the stopping criteria is satisfied. In this study, the Approximated Centroidal Voronoi Diagrams (ACVD) [47] is employed for the model surfaces resampling at each iteration, and the Poisson disk sampling [48] is used to represent the point set (for fidelity measurement). The baseline surface size is estimated based on the covered area of the selected camera FOV and the

baseline GSD [49]. Based on the experiments on various input geometries, we set  $\omega$  as 1 meter and  $\tau$  equals to 15% which preserves the major geometric components (e.g., façade, roof, etc.) of building-scale structures (normally at scale  $100\text{ m}^2 \sim 5000\text{ m}^2$ ). Noted that the algorithm outputs a mesh with a relatively small number of surfaces (i.e., larger surfaces size), thus encourage to generate less camera views which increases the efficiency of the inflight images acquisition and the post-flight 3D reconstruction. Figure 4 presents an example of the proposed method on triangulating an orbit flight reconstructed residual building. The result model contains significantly less surfaces while guaranteeing the baseline GSD at each image.



**Figure 5** Workflow of the model preprocessing algorithm on a residential building

- (a) The input model contains over 50,000 surfaces (resampled for visualization purposes, each triangle surface is color-coded); (2) The residual map between a resampled model and the input when the stopping criterion is met (points in blue denotes the inliers; points in red denotes the outliers); (c) Resampled model only contains 122 triangles. Noted the major structural components of the buildings are preserved.

### ***Topology-based Coverage Modeling***

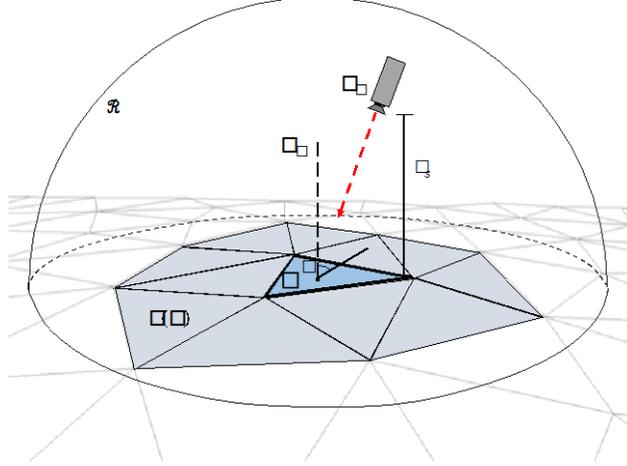
The proposed coverage algorithm defines the MVS camera search spaces based on the topology of the model surfaces. The algorithm includes two major steps: (1) sampling a set of overlapped cameras with each camera covering an unique set of mesh surfaces; and (2) clustering the neighborly located cameras as the candidate matches for the 3D triangulation at each surface.

#### *Sampling Overlapped Cameras*

For each mesh surface  $s$  ( $s \in \mathcal{S}$ ), an admissible camera view  $\nu_s$  must be within a hemisphere centered at the centroid of the mesh surface ( $c_s$ ) with the camera FOV covers the adjacent triangles of  $s$  (as in Figure 6). It is also required that each admissible view satisfies the camera gimbal constraints (e.g., camera pitch rotation constraint) and the UAV safety clearance from the surveyed object [29]. Eq.1 presented the constraints that formulates the camera search space:

$$\begin{aligned}
 d(\nu_s, c_s) &\leq \mathcal{R} \\
 \dot{d}(\nu_s, s) &\geq d_s \\
 \theta(\nu, n_s) &\leq \theta_{max} \\
 \pi(\nu_s, \gamma_{\#}(s)) &= 1
 \end{aligned} \tag{1}$$

Where  $d$  measures the Euclidean distance between  $v_s$  and  $c_s$ ,  $\mathcal{R}$  is the radius of the hemisphere which is defined by the baseline GSD given the camera focal length.  $\hat{d}$  computes the perpendicular distance between the camera and the surface, and  $d_s$  is the safety tolerance for the aircraft.  $\theta$  measures the observation between the camera ray and the surface normal  $n_s$ , and  $\theta_{max}$  indicates the maximal acceptable observation angle.  $\pi$  is the Boolean function that denotes the complete coverage of a camera view to the mesh surfaces.  $\gamma_{\mathcal{k}}(s)$  is a collection of the neighborly connected mesh surfaces centered at  $s$ . We set  $\mathcal{k}$  equals to 1 to ensure the minimal overlapping between the sampled cameras. The definition of  $\gamma_{\mathcal{k}}(s)$  and the selection of  $\mathcal{k}$  are detailly discussed in the next section.



**Figure 6** Each admissible camera view  $v_s$  must cover a neighborhood of the surfaces  $\gamma(s)$  (colored in gray) centered at  $s$  (colored in blue)

### Clustering Candidate Matchings

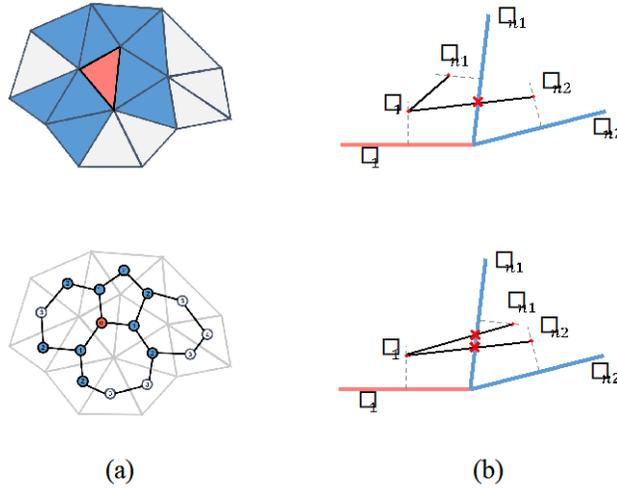
In MVS, triangulating 3D points from the closely located images can reduce the cost of the redundant image selection while filters out the erroneous points [25][26][27]. In this study, the topology of the mesh surface connections is used to guide the clustering of the neighbor cameras for the per surface MVS quality maximization. Given the mesh, we define a graph  $\mathcal{G}(\mathcal{S}, \mathcal{E})$  with each node denoted as a triangle surface and each edge as the edge connection between the surfaces. By defining every edge as a unit distance, the proximity of the surfaces can be computed by counting the minimal number of edges traversed between the surfaces. As shown in Eq. 2, the neighborhood of  $s$  is defined as the collection of the mesh surfaces with the proximity distance less than  $\mathcal{k}$ .

$$\gamma_{\mathcal{k}}(s) = \bigcup_{s' \in \mathcal{S}} s', \text{ if } \hat{d}(s, s') \leq \mathcal{k}, \mathcal{k} \in \mathbb{Z}^+ \quad (2)$$

Where  $\hat{d}$  is the distance of the shortest path between  $s$  and  $s'$ , and  $\mathcal{k}$  is a coefficient that controls the number of the neighbor surfaces. Due to one camera view is defined at each surface, thus the topological structure (i.e., neighborhood) of the surfaces (i.e.,  $\mathcal{G}$ ) can be used to map the closeness of the sampled cameras (as in Eq. 3).

$$\gamma_{\mathcal{k}}(\mathcal{v}_s) \leftrightarrow \gamma_{\mathcal{k}}(\mathcal{s}_{\mathcal{v}}) \quad (3)$$

Where  $\mathcal{s}_{\mathcal{v}}$  and  $\mathcal{v}_s$  respectively denotes the correspondent mesh surface and the camera.  $\gamma_{\mathcal{k}}(\mathcal{v}_s)$  as a cluster of closely located cameras within each one covers a surface in  $\gamma_{\mathcal{k}}(\mathcal{s}_{\mathcal{v}})$ . Because the influence of a camera is spatially limited (i.e., FOV, GSD), there is a large chance that the best camera views for triangulating  $\mathcal{s}_{\mathcal{v}}$  belong to  $\gamma_{\mathcal{k}}(\mathcal{v}_s)$ . With greater  $\mathcal{k}$  means more cameras are clustered, which increases the possibility of covering the real best matches. The extreme case is that all cameras are included in each cluster which is identical to perform the stereo-matching globally. In this study,  $\mathcal{k} = 3$  is used empirically to ensure the good matching results for complex scenes and, at the same time, keeps the computational load small (i.e., search the best matches in each cluster). Figure 6 (a) shows an example of the clustered neighbor surfaces with  $\mathcal{k} = 2$  for demonstration purposes.



**Figure 7 (a)** Top: the current model surface (in red) and the clustered neighbor surfaces (in blue); Bottom: the topology graph, denoting the current view (in red) and the clustered candidate matches (in blue); **(b)** surface selection when more than one surface is connected by an edge. Top: the condition where  $\mathcal{s}_{n1}$  is considered as the neighbor, while  $\mathcal{s}_{n2}$  is neglected due to the intersection between the line  $\mathbf{p}_1$  and  $\mathbf{p}_{n2}$ ; Bottom: both  $\mathcal{s}_{n1}$  and  $\mathcal{s}_{n2}$  are neglected due to the existence of intersections in both conditions.

In most cases, the growth of cluster size follows the properties of the triangular mesh (i.e., each mesh surface connects at most three other surfaces). However, there are cases when the surface edge is connected through more than one surface plane, such as at the concave area. Under this condition, a simple mechanism is defined to select the neighbor surfaces. As shown in Figure 6 (b), the current surface ( $\mathcal{s}_1$ ) is edge connected with two candidate surfaces ( $\mathcal{s}_{n1}$  and  $\mathcal{s}_{n2}$ ). By connecting the points along the normal of the current ( $\mathbf{p}_1$ ) and the candidate surface planes ( $\mathbf{p}_{n1}$  and  $\mathbf{p}_{n2}$ ), the surface with no intersection within the connection is selected as the neighborhood of  $\mathcal{s}_1$ . Noted due to the clusters are only determined based on the model topology, which is given in advance, thus we can cluster the surfaces/cameras ahead of the planning.

### **Multi-view Optimization** *Problem Formation*

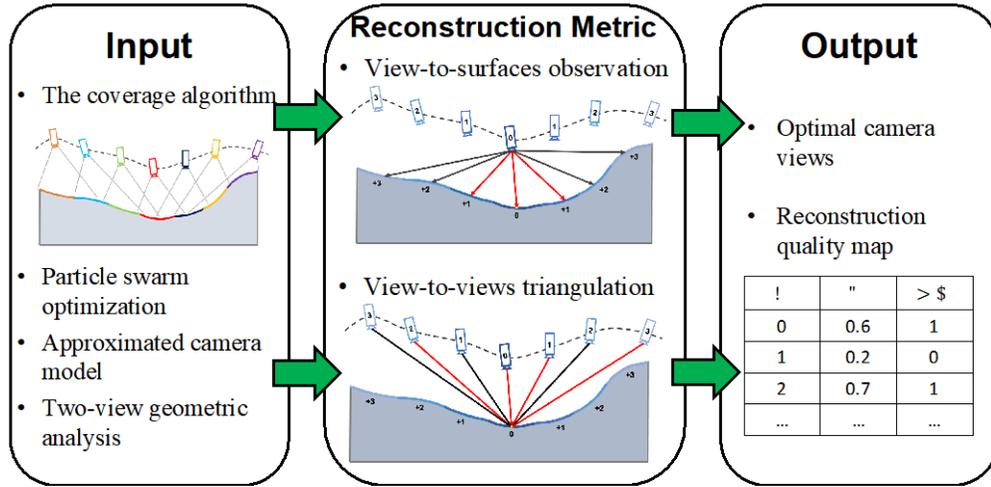
Given a model (i.e., triangular mesh) and a designed number of camera views, the objective is to find the optimal camera views ( $\mathcal{V}^*$ ) in terms of the positions and the orientations where the MVS quality at each surface of the model is maximized (Eq. 4):

$$\mathcal{V}^* = \operatorname{argmax}_{\mathcal{V}} \sum_{s \in \mathcal{S}} \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{V}} h(v, u, s) \quad (4)$$

Where  $v, u$  ( $u \neq v$ ) consist of the potentially matched cameras and  $h$  is the metric that models the MVS quality based the matched views. The computational complexity of finding the optimal solution is  $O(|\mathcal{S}||\mathcal{V}|^2)$  since we need to traverse through every camera combination to find the best one [22]. This formation, based on the proposed coverage algorithm (Section 2.2), can be reformulated as (in Eq. 5):

$$\mathcal{V}^* = \operatorname{argmax}_{\mathcal{V}} \sum_{s \in \mathcal{S}} h(v_s, \gamma_h(v_s), s) \quad (5)$$

Noted due to  $s$  and  $v_s$  are exclusively correlated, and  $h$  is a constant, the complexity of the problem is reduced to linear  $O(|\mathcal{S}|)$ . This strategy regularizes the search space needed to find the best matches, makes the multi-view optimization problem computational tractable even in continuous space.



**Figure 8** Overview of the multi-view optimization (The camera rays in red conceptually denotes the best matches selected from the candidates (camera rays in black) at each iteration)

### Multi-view Optimization

Figure 8 shows the overview of the optimization. In this study, the optimization is built on a priorly developed PSO framework [46]. The framework is adapted in this method for efficient and continuous MVS cameras optimization. The main steps of the optimization can be divided into six steps:

- Step 1: Initialize a population of particles with each particle denotes a set of randomly sampled admissible camera views.

- Step 2: Evaluate the cost of each particle through the fitness function defined by a reconstruction metric ( $\mathbf{h}$ )
- Step 3: Record the particle with the maximal fitness as the best particle.
- Step 4: Sequentially update the cameras within the best particle by switching the cameras belong to other particles in population [46]. Update the global best if the particle shows a better result.
- Step 5: Update the camera positions in each particle using the PSO mechanism. Recompute the camera view based on the coverage constraints. The camera is rejected if the recomputed positions/orientations do not lie in the admissible spaces.
- Step 6: Repeating steps 2-5 a number of iterations, or the global best record does not increase for at least 3 iterations (i.e., convergence).

The output is the camera views with the highest MVS quality. Noted that while Step 4 avoids the optimization been trapped at the local optima, this step is expensive because the cameras need to be sequentially rendered/evaluated, resulting in the long processing time. In the following sections, we firstly discuss how the reconstruction metric is calculated and then present an approximated camera model that significantly accelerates the optimization (especially at Step 4).

### Reconstruction Metric

In this subsection, we quantify the reconstruction quality  $\mathbf{h}$  at each surface given a set of cameras. The metric is an estimation of the MVS quality based on the two-view geometric analysis [41]. Technically, the metric is composed of two terms: the *view-to-surfaces observation* and the *view-to-views triangulation*. We discuss each term in detail as follows.

*View-to-surfaces observation* ( $\mathbf{h}_\sigma$ ): The view-to-surface observation measures the distance and the incidence angle between each camera view to the covered mesh surfaces (as presented in Eq. 6).

$$\mathbf{h}_\sigma(\mathbf{v}, \mathcal{S}) = \sum_{\mathcal{S}^+ \in \gamma_1(\mathcal{S})} \mathbf{h}_{res}(\mathbf{v}_s, \mathcal{S}^+) \mathbf{h}_{ang}(\mathbf{v}_s, \mathcal{S}^+)$$

$$\mathbf{h}_{res}(\mathbf{v}, \mathcal{S}) = \max \left[ 1 - \frac{d(\mathbf{v}, \mathbf{c}_s)}{\mathcal{R}}, 0 \right] \quad (6)$$

$$\mathbf{h}_{ang}(\mathbf{v}, \mathcal{S}) = \max \left[ 1 - \frac{\theta(\mathbf{v}, \mathbf{n}_s)}{\theta_{max}}, 0 \right]$$

Where  $\mathbf{h}_{res}$  is the image resolution factor that measures the camera to surface distance, and  $\mathbf{h}_{ang}$  is the image distortion factor that indicates the incidence angle between the camera direction and the surface normal. We constrain both factors within the range of  $[0, 1]$ . Noted we do not provide the visibility check in Eq. 6 because the sampled cameras must already satisfy the coverage constraint (Eq. 1).

*View-to-views triangulation* ( $\mathbf{h}_t$ ): For each camera view, the view-to-view triangulation factor seeks the  $m$  best matches from the candidate matching cameras in terms of the sufficient overlapping, the appropriate parallax, and the small baseline (as in Eq. 7):

---


$$\begin{aligned}
\mathbf{h}_t(\mathbf{v}, \mathbf{s}, m) &= \max_{\substack{\hat{\gamma}(\mathbf{v}) \subset \gamma(\mathbf{v}) \\ |\hat{\gamma}(\mathbf{v})|=m}} \sum_{\mathbf{v}^+ \in \hat{\gamma}(\mathbf{v})} \boldsymbol{\pi}(\mathbf{v}^+, \mathbf{s}) \mathbf{h}_{bas}(\mathbf{v}^+, \mathbf{s}) \mathbf{h}_{par}(\mathbf{v}, \mathbf{v}^+, \mathbf{s}) \\
\mathbf{h}_{bas}(\mathbf{u}, \mathbf{s}) &= \max \left[ 1 - \frac{d(\mathbf{u}, \mathbf{c}_s)}{\mathcal{R}}, 0 \right] \\
\mathbf{h}_{par}(\mathbf{v}, \mathbf{u}, \mathbf{s}) &= \exp \left( \frac{(\theta_{\angle \mathbf{v} \mathbf{c}_s \mathbf{u}} - \rho)^2}{2\sigma^2} \right)
\end{aligned} \tag{7}$$

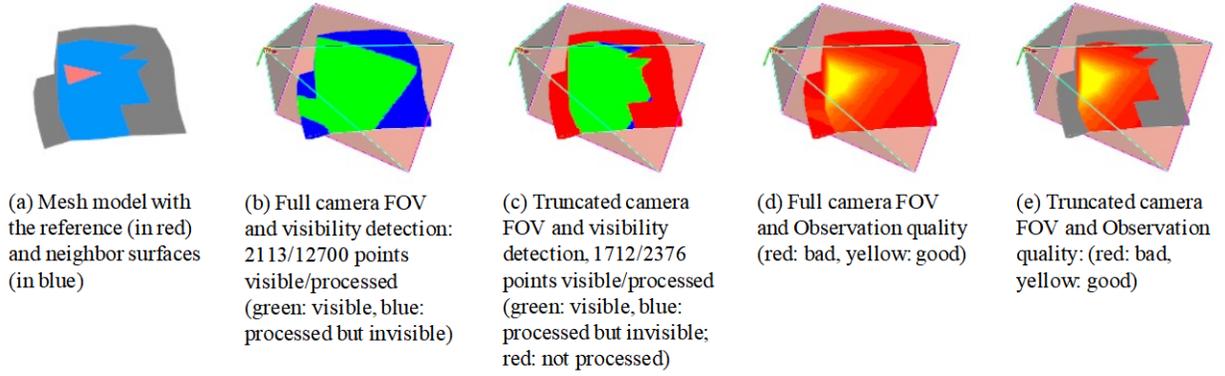
Where  $\mathbf{h}_{bas}$  evaluates the baseline effects of a camera pair and  $\mathbf{h}_{par}$  models the parallax dependency on the depth error. The  $\mathbf{h}_{par}$  is measured as a Gaussian function defined based on. We empirically set  $\rho$  as  $28^\circ$  for additional tolerance at large angles and let  $m$  equal to 4 as suggested in [25].

Finally, we concatenate the defined terms (Eq. 6 and Eq. 7), and the reconstruction metric at each camera view/model surface is obtained as in Eq. 8.

$$\mathbf{h}(\mathbf{v}, \mathbf{s}) = \mathbf{h}_v(\mathbf{v}, \mathbf{s}) \mathbf{h}_t(\mathbf{v}, \mathbf{s}, m) \tag{8}$$

### *Approximated Camera Model*

In general, the camera model that renders the visibility detection has been considered as the most expensive step for multi-view optimization [39][41]. Given a set of cameras and the surface points, the visibility of each camera is computed through the ray casting operation at every surface point in two steps: (1) measuring if the point is located within the camera FOV; (2) checking whether there is an obstacle between the camera and the surface point. The time complexity of this operation is quadratic in terms of the number of the cameras and the surface points. The proposed camera model reduces the computation cost while assures the visibility detection quality by actively restricting the camera visible points within  $\gamma(\mathbf{s}_v)$ . To ensure the approximated camera does not exclude the good quality points, a photometric metric [51] is used for the evaluation. As shown in Figure 9, comparing to the original camera, the approximated camera only takes 16% of the computation, but covers more than 80% of the regions with the best observation quality. Noted that this strategy would perform even better for large-scale structure because the visibility calculation does not grow as the model increases.



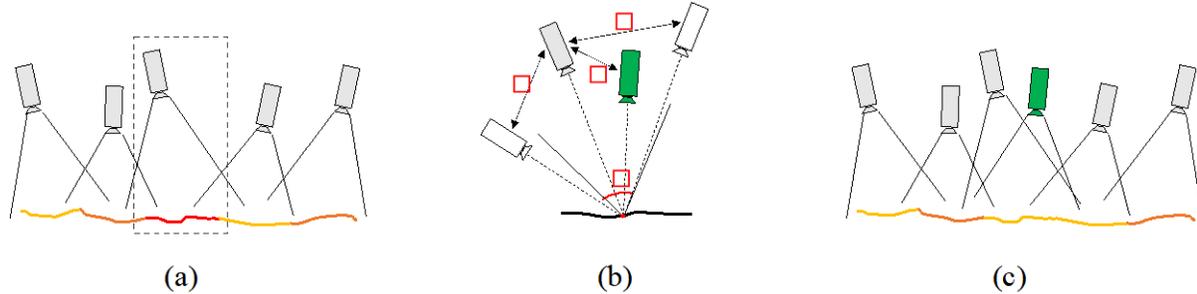
**Figure 9** Evaluation of the approximated camera model: (a) the mesh geometry at a selected (in pink) and its neighbors (in blue and  $\mathcal{h} = 3$ ); (b) visibility detection using the original camera model and the computation load; (c) visibility detection using the approximated camera and the computation load; (d) the observation quality of the original camera model; (e) the observation quality of the approximated camera model.

Based on the approximated camera model, the optimization can be facilitated by simultaneously evaluating the non-overlapped cameras. Given the graph  $\mathcal{G}$ , we construct a new graph  $\mathcal{G}'$  where each node denotes a cluster of surfaces/cameras in  $\mathcal{G}$ . Then, the edges in  $\mathcal{G}'$  are denoted as the connections between the clusters. Based on the approximated camera model, the non-overlapped cameras can be incrementally identified by breadth-first searching (BFS) through the disconnected nodes in  $\mathcal{G}'$ . To avoid the order dependency of the selected cameras, a random seed is used as the starting node at each iteration. After each search, a set of non-overlapped cameras is obtained. Then, we subtract the correspondent nodes in  $\mathcal{G}'$ , and use the same strategy to recursively select the non-overlapped cameras from the rest nodes. The recursion terminates until the number of unselected cameras is less than  $\varepsilon$ . For a mesh model containing 100 triangular surfaces, normally this procedure reduces 6-9 times of the optimization when compared to the existing framework where the camera views are evaluated/updated sequentially. Based on this strategy, our method can compute the optimal cameras in minutes even on a personal PC with limited computational resources (e.g., memory, GPU, etc.).

### *Sampling Complementary Cameras*

In most cases, we obtain high-quality dense reconstruction from the optimized results. However, for complexity scenes where there are locally sharp corners or deeply concave regions, complementary camera views may be needed for the sufficient coverage and depth estimation [39]. To account for this issue, a heuristic is proposed to adaptively sample complementary cameras at those regions. A workflow of the proposed heuristic is shown in Figure 9. First, the reconstruction quality at each mesh surface, given a set of optimal camera views, is evaluated. The model surfaces with insufficient reconstruction quality (i.e.,  $\mathcal{h} < 0.15$ ) are identified. For each extracted surface, we add a complementary camera which forms the good geometric reasoning with the existing camera. To achieve that, we sample several admissible cameras at the surface through the coverage model with the constraints of the parallax angle  $\varphi$  and baseline distance  $d$  between each newly sampled view and the existing camera for the good images matching. Next, we evaluate the quality of each admissible camera based on the observation quality  $\mathcal{h}_\sigma$ . The camera view with the best quality at the mesh surfaces is selected as the complementary camera. We iterate this procedure

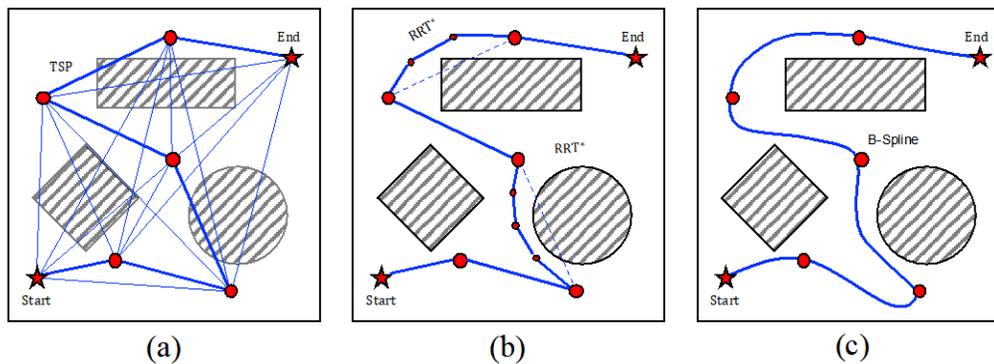
until complementary camera views are provided at every extracted surface. Finally, we merge the complementary cameras into the optimized camera set and output the merged cameras. In theory, we can repeat this process several times to enhance the reconstruction quality at those complex regions. However, adding redundant cameras often results in the diminished returns. Thus, we perform this process only once for all the experiments in this study.



**Figure 10** Workflow of sampling complementary cameras for global reconstruction: (a) identify the model surface with insufficient reconstruction (in dashed box); (b) sample admissible cameras (in light green) based on multi-view stereo geometric criterion, and the camera view with the best observation quality is selected (in green); (c) merge the newly sampled camera into the optimal camera set

### Trajectory Generation

For automated aerial reconstruction of real-world structures, the computed camera viewpoints need to be converted into a safe and efficient trajectory for aerial images collection. The process of the trajectory generation can be divided into three steps (as in Figure 11). First, we construct a complete graph with each node representing a camera position, and each edge indicating the Euclidian distance between the cameras. The start and end positions are determined based on the UAV take-off/landing spots. The shortest path travels through all the camera positions without considering the existence of the site obstacles is computed through the LKH-TSP solver [52]. Then, the collision detection is performed at each path segment using the hierarchy OBBTree [53]. If the collision is found in a path segment, the rapidly exploring random tree star (RRT\*) [54] [55] employed to reroute each segment. We define the search space using the ellipsoidal heuristic as in [35] to speed up the path finding process. Finally, we employ the B-spline Curve interpolation [56] to convert the collision-free path into a trajectory that can be safely followed by a rotary UAV.



**Figure 11** Workflow of the trajectory generation: (a) Find the shortest TSP path without considering the on-site obstacles; (2) Find the path segments collided with the obstacles and reroute the segments using RRT\*; (3) Smooth the path into the flight trajectory using B-Spline curve interpolation.

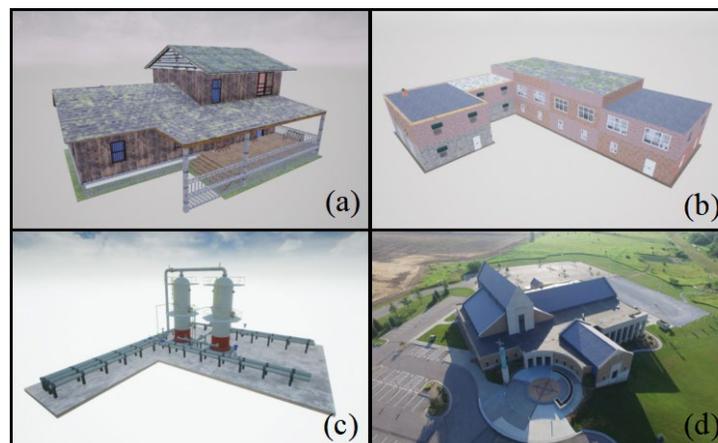
---

## **Experimental Setup**

### *The Selected Scenes*

As shown in Figure 11, three synthetic scenes and one real-world structure are selected for the evaluation of the proposed method. Using the synthetic scenes for evaluation has two advantages: 1) the ground truth models of the synthetic scenes are noise-free as opposed to other measurement tools (e.g., ground control points, GNSS-based measurement or laser scanning), enables the precise evaluation of the reconstruction quality (i.e., accuracy and completeness). 2) the synthetic environment is controlled so ensures the image quality is consistent and not affected by the random real-world environment factors (e.g., sunlight direction, shadows, wind, and moving objects in the field). In this way, the performance evaluation of different path planning algorithms can be compared on the algorithms themselves without concerning random impact from environmental factors. Due to the two advantages synthetic models are widely adopted in performance evaluation in MVS reconstruction [38][39][41][43]. To evaluate the robustness of our algorithm a real-world building case is also used.

The selected synthetic scenes are from free online sources and rendered in Unreal Engine 4: a 3D game engine with rich support of photo-realistic scenes. It is noted that these scenes were not originally designed for multi-view reconstruction, thus we adjust the original modular template such that sufficient salient features can be extracted from the rendered cameras. The three synthetic scenes are objects of (a) *Res.* [57], (b) *Com.* [58], and (c) *Ind.* [59] (as in Figure 9 (a) – (c)), each with unique geometrical properties. Specifically, (a) is a two-story residential house. It is a small building structure contains many geometrical complex regions, such as the gable roof, the dormers, as well as the balcony, that are hard to be fully covered by the aerial images; (b) is a commercial building with flat roof and rich surface textures. The model contains a concave area with sharp corners that is difficult to triangulate due to the large parallax. The last scene, (c) is an industrial oil/natural gas facility. It contains both the texture-less twin tank structures and the surrounding slim pipelines that are considered challenging geometries for multi-view reconstruction. To test the applicability of the proposed method on real-world applications, a catholic church located at the south of Lincoln, Nebraska (as in Figure 9 (d)) is used. The building is large scale and contains rich geometrical details and surface variations across the building façades/roofs, makes it a challenging task for high-quality reconstruction using UAV.



**Figure 12** The selected case studies in this study: (a) *Res.*, (b) *Com.*, (c) *Ind.*, and (d) *Church*.

---

### Selected Input Models

In this study, different types of input models are selected to evaluate the proposed method. For the *Res.* and *Church*, image reconstruction from an orbit flight path is used to obtain the rough geometries of the structures. However, we found the orbit paths fail to generate complete models for *Com.* and *Ind* (i.e., missing walls at the concave regions and the low coverage of the self-occluded pipe structures). Thus, we manually create the 2.5D models of these structures based on a rough estimation of the occupied area and the height of the model. Noted these input models only provide the coarse estimations of the ground truths, which needs to be further refined using the presented methods.

### Images Acquisition

For the synthetic scenes, we render images at the designed cameras through UnrealCV[60]. We set the camera at the resolution of 4000x3000 with the horizontal FOV as 90°. The sunlight shadows, the light reflection, and the camera exposure are disabled through the images collection.

For the real-world application, the first step is to convert the computed trajectories from the local coordinates to the World Geodetic System (WGS-84). Next, the converted geographic coordinates are imported into Ground Station Software. In this study, we select UgCS [61] due to its flexibility to import a sequence of waypoints with designated camera motions and the support of various types of drones. DJI Mavic 2 Zoom is selected as the aerial platform for the automated waypoints following and images acquisition. Due to the limited battery capacity as opposed of the designed flights, multiple flights are needed to travel through every designed waypoint. We manually set the drone return to home when the battery level is less than 30%. After the battery replacement, we fly the drone to the last visited waypoint and continue the mission until every waypoint is visited.

### 3D Reconstruction

After the images acquisition, the collected images are imported into Agisoft Metashape [62] for the 3D dense reconstruction. The positions of the cameras (GPS for the real-world environment) are used to facilitate the photo alignment steps. Noted the repeated textures in the synthetic scenes can cause false alignments between the images. Such alignments need to be reset and re-aligned to avoid the mis-aligned models. We set the quality as *high* both in the image alignment and dense point cloud reconstruction steps. For the 3D surface reconstruction, the interpolation mode and hole fillings are disabled to preserve the geometric details. Noted we only output the dense point cloud for the *Ind.* scenes since it is very costly to convert the slim structures (i.e., railings, pipelines, joints, ladders, etc.) into surface meshes without producing additional artifacts. Table 1 presents the general information and statistics of the selected structures, the type of the input geometry, and the reconstruction output.

**Table 1** Statistics of the selected structures: (from left to right) the areas, the height, the input type, the triangular surface number of the resampled mesh, as well as the reconstruction pipelines.

Scene	Type	Area (m <sup>2</sup> )	Height (m)	Input Geometry	Reconstruction Output
<i>Res.</i>	Synthetic	296.22	9.87	Orbit Flight Reconstruction	Triangular Mesh
<i>Com.</i>		1,344.29	9.12	2.5D	Triangular Mesh
<i>Ind.</i>		1,422.41	15.11	2.5D	Dense Point Cloud

---

## Comparisons

To evaluate the performance of our results, two methods are selected for the comparison: (1) the baseline Overhead and (2) the state-of-the-art NBV. Overhead is a commonly used path planning method for UAV photogrammetry and is widely used for benchmarking path planning algorithms [63]. Same as the prior work [38], the Overhead flight is designed as a lawnmower flight at a safety height above the scene followed by an orbit path to capture the oblique images. We uniformly distributed the camera views on both the lawnmower and the orbit path with camera views oriented towards the center of the scene. The 80/80 image overlapping between the adjacent views is required to ensure the complete reconstruction.

In this study, we implement a naive NBV method as presented in [38][39][63] which greedily select the vantage camera viewpoints from an ensemble of candidate cameras. The two-view quality evaluation [41] is used to guide the view selection process such that the compared results are majorly affected by the optimization methods instead of the quality criterion been used. To make a fair comparison and avoid the unbounded images collected in NBV, the number of camera views estimated in our method is used as an additional termination condition in NBV optimization.

## Evaluation Methods

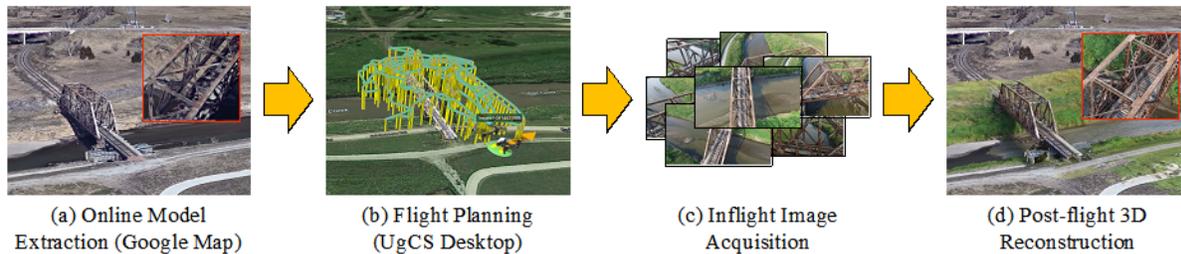
To quantitatively evaluate the reconstruction quality, the 3D models need to be initially transformed to the same coordinates of the ground truth. In this study, such transformation computed in a coarse-to-fine fashion: First, we roughly align the two models based on several well distributed points (e.g., corners) both from the reconstructed model and the ground truth. Next, we refine this rough alignment through the iterative closest point (ICP) registration [64] provide a robust registration result, we embed the RANSAC [65] into the ICP to remove the outliers as well as guarantee the registration convergence. Based on the aligned models, we compare the quality of the reconstruction models based on two indicators: *accuracy* and *completeness* as in the previous works [41]. Accuracy (also denoted as precision) measures the maximal geometric distance between the reconstruction to the ground truth, given a percentage of the points. To measure the *accuracy*, we compute the Euclidean distance between each point on the surface of the reconstructed model to the closest point on the ground truth. Then, we collect the distances from the whole point set, and compute the histogram map at 90%, 95% and 99% of the point set. The *completeness* (also denoted as recall) measures the coverage ratio of the reconstruction to the ground truth, given a distance threshold. To calculate the *completeness*, we use the same strategy to compute closest-point distance between the point sets, but count the percentage of the points within the pre-defined distance. We empirically set the thresholds as: 0.02m, 0.05m, and 0.2m, to evaluate the *completeness* at different levels of details (LoDs).

### 3.1.2.2 Task 2.2 Path Planning Method for Truss Structures [66]

#### Method Overview

Figure 13 shows the overview of the truss bridge reconstruction using the proposed flight planning method. The method assumes an existing rough geometrical model of the bridge, which can be extracted from the web (Google Map in our case) using third-party tools (e.g., OpenStreetMap).

The extracted model is an unstructured triangular mesh (KMZ format) containing both the bridge and the surroundings. The model is further processed to explore the camera/UAV search space around the bridge and the observable truss surface points for the subsequent view and path planning. The proposed method first computes the optimal viewpoints that maximize the reconstruction quality at each observable surface point by selecting the best subset from an iteratively resampled candidate set. The candidate is a set of densely sampled oblique viewpoints (i.e., multiple orientations at each position) initialized within the UAV free space while considering the camera/inspection parameters. After the optimization, the method converts the discrete viewpoints into single or multiple smooth flight trajectories subject to UAV constraints (i.e., aerodynamics, battery capacity, memory usage, and safety distance to the on-site objects). These trajectories are then transformed into the world coordinates (e.g., WGS84) and uploaded to the onboard autopilot system for the automated inflight image acquisition using single or multiple UAVs. A photogrammetric reconstruction software will use the acquired high-quality images to generate a truss bridge's geo-referenced, high-fidelity 3D model.



**Figure 13.** Overview of truss bridge reconstruction based on the proposed method.

### ***Initialization***

#### *Input Parameters*

Several important parameters must be defined as the inputs of the proposed method. In this study, we classify these parameters into four categories: (1) the UAV parameters, (2) the camera parameters, (3) the inspection requirements, and (4) the safety concerns. The UAV parameters include the physical properties of the selected UAV, such as the overall flight duration, the designed inflight speed, and the maximal number of executable waypoints in each flight. The camera parameters describe the properties of the onboard camera system, including the horizontal angle-of-view (AOV), the resolution of the onboard camera, and the gimbal pitch rotation limits. Due to the in-plane rotations do not change the image contents, we lock the gimbal roll angle as  $0^\circ$  and align the gimbal yaw with the UAV orientation. The inspection requirements are factors that control the quality of the collected images: they are the maximal/saturated ground sampling distance (GSD) and the incidence angle. The safety concerns are parameters that define the UAV flyable space. It includes the safe clearance to site objects, the minimum height above ground level (AGL), and whether the UAV is enabled to fly through the truss. Flying-through-truss is a binary coefficient that defines whether the spaces within the truss structure are available for UAVs to pass through. These spaces enable the UAVs to inspect the truss bridge's interior surfaces better. However, most consumer-grade UAVs cannot fly closely around metal structures (e.g., steel truss) because electromagnetic disturbances can affect the onboard sensing system (i.e., compass) and corrupt the GPS positioning capabilities. Therefore, for safety, we enable the flying-through-truss

option only when the UAV onboard navigation system can handle the signal interference. The symbols, descriptions, and the default values of the parameters are listed in Table 2 below.

**Table 2. Summary of the input parameters, the symbols, the descriptions, and the default values**

Categories	Parameters (Symbols)	Description	Value
UAV Parameters	Flight speed (A)	The designed travel speed between waypoints	1.5 m/s
	Flight duration (B)	The maximum duration of each flight constrained by the onboard battery capacity (Data from DJI Inspire 1).	15 mins
	Number of waypoints (W)	The maximum number of waypoints can be uploaded to the autopilot system at once (Data from DJI Inspire 1).	99
Camera Parameters	Horizontal angle of view (HAOV)	The horizontal angle of view of the onboard camera	90°
	Image resolution (R)	The resolution of the camera captured images	[4000, 3000]
	Gimbal pitch limits ( $[\varphi_-, \varphi_+]$ )	Lower and upper bounds of the gimbal pitch rotation	$[-90^\circ, 30^\circ]$
Inspection Requirements	Maximal ground sampling distance ( $GSD_{max}$ )	The largest acceptable size of the scene represented by each pixel for feature extraction and matching	8 cm/pixel
	Saturated ground sampling distance ( $GSD_{sat}$ )	The satisfied size of the scene represented by each pixel for feature extraction and matching	1 cm/pixel
	Maximal incidence angle ( $\theta_{max}$ )	The largest acceptable angle between the camera ray and the normal of the scene for feature extraction and matching	65°
	Saturated incidence angle ( $\theta_{sat}$ )	The satisfied angle between the camera ray and the normal of the scene for feature extraction and matching	15°
Safety Concerns	Safe distance tolerance ( $D_s$ )	The minimal distance between the center of the UAV and on-site object considering the GPS positioning error and signal interference	5 m
	Minimal height AGL ( $H_{min}$ )	The minimal height above ground to avoid the ground effect and the potential site objects (e.g., trees, vehicles)	7 m

---

Flying-through-truss ( $\rho$ )	A user-controlled parameter defines whether the selected UAV can fly at the interior of the truss bridge	0
---------------------------------	--	---

---

### *Preprocessing*

Based on the input parameters, the initial model extracted from Google Map is pre-processed to define the UAV configuration space, the search space of the admissible viewpoints, and the truss surface points for visibility/quality evaluation.

### *UAV Configuration Space*

UAV configuration space is the free space accessible by a UAV. Due to the external noise (e.g., GPS errors, wind, signal interference, etc.), a safety tolerance (Table 2) between the model and the selected UAV must be maintained. Since the input model format is a triangular mesh, the space inaccessible by a UAV can be defined by extruding the safety tolerance along the normal at every surface of the mesh. New mesh surfaces (highlighted as orange in Figure 13 (a)) that cover every side of the bridge with the defined tolerance are then constructed by connecting the adjacent extruded points. Positions located within this mesh or intersected with the mesh surfaces are considered collisions.

To define the free space within the truss, another surface mesh that covers the interiors of the truss structure (highlighted as yellow in Figure 13 (a)) is developed. This mesh can be manually created or downscaled from the convex hull. It is worth noting that this mesh is created only when the fly-through-truss option is disabled.

### *Viewpoints Search Space*

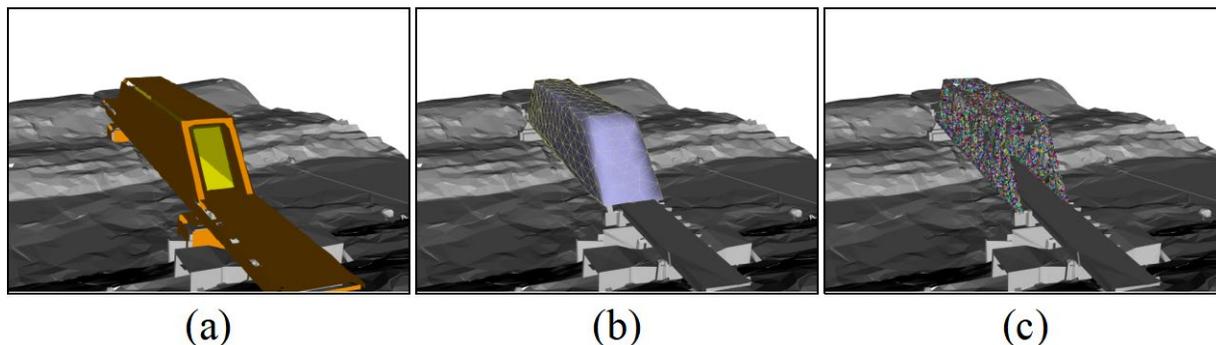
The viewpoints search space is a subset of the UAV configuration space where the baseline observation quality of the collected images is guaranteed. Thus, only the free spaces surrounding the truss surfaces within certain distances should be considered. To achieve that, we perform the Quickhull algorithm [67] to generate a watertight convex hull that tightly covers the input truss. The convex hull is then resampled into a uniformly distributed triangular mesh using ACVD [68]. Figure 13 (b) shows the triangle mesh for generating the candidate viewpoints set. For each triangular surface, a candidate viewpoint can be generated using the sampling-based coverage algorithm [69]. This strategy encourages the uniform sampling of the candidate set, which provides a good initialization for the subsequent optimization.

Please note that the candidate set only covers the exterior of the truss. To also sample the candidate viewpoints at the interior of the truss (when the fly-through-truss mode is activated), we reverse the normal of the convex hull and resample the surface mesh. The result is a double-sided triangular mesh where viewpoints can be sampled within the free space at both sides of the truss. In this study, we set the interior/exterior convex hull to contain 100/500 triangle surfaces respectively for the candidate viewpoints sampling.

### *Truss Surface Points*

The surface points are visible points located at the surface of the bridge truss structure. These points are utilized to measure each viewpoint's visibility and quality. Given the input model of the truss structure, Poisson disk sampling [70] is employed to sample the surface points at the model

surfaces evenly. The normal of each point is computed as the average of the surface normal at each local Poisson disk. Figure 13 (c) shows the sampled surface points at the truss surface.

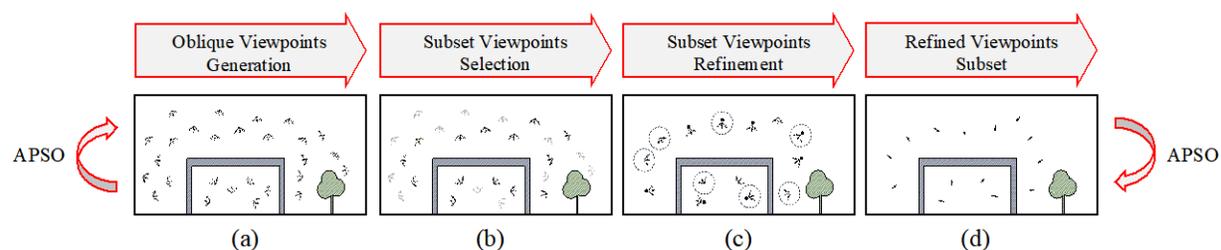


**Figure 14.** Model preprocessing of truss structure

- (a) the surface meshes that define the UAV configuration space (colored in orange) and whether UAVs are allowed to fly within the truss (colored in yellow); (b) the resampled convex hull for candidate viewpoints sampling (only the exterior mesh is shown); (c) the truss surface points (randomly colored) that guides the subsequent viewpoints planning.

### View Planning

The proposed view planning method finds the optimal viewpoints in a sampling-based iterative optimization schema: Initialized with a quasi-randomly sampled candidate set of oblique viewpoints, the method iteratively resamples the candidate set until the termination condition is met (e.g., number of iterations). At each iteration, the method finds a viewpoints subset from the candidate set based on the quality-efficiency metric, an efficient view selection algorithm, and viewpoints refinement process. The selected subset is utilized to guide the resampling of the candidate set in the next iteration. In this study, we wrap the candidate resampling process into the accelerated particle swarm optimization (APSO) [72] due to its lightweight and fast convergence speed, enables the selected subsets efficiently converge to the global optimal. Figure 15 shows the workflow of the proposed view planning method. The details of each step of the proposed method are discussed in the following paragraphs.



**Figure 15** Workflow of the proposed view planning method.

- (a) Sampling a set of oblique viewpoints (i.e., multiple orientations at each view position); (b) Selecting the subset of the viewpoint from the oblique set (non-selected viewpoints are gray out); (c) Refining the selected subset based on resampling operation (highlighted in black) and the mutation operation (highlighted in dashed circle); (d) The refined viewpoints subset can be used to guide the resampling of the oblique viewpoints in the next iteration or outputted when termination condition is met.

### Oblique Viewpoints Generation

---

The process of generating the set of oblique viewpoints is divided into two steps: 1) sample one admissible viewpoint at each surface of the mesh (i.e., subsampled convex hull); 2) spawn oblique orientations at each sampled viewpoint to enhance the searchability at each position.

#### *Sampling Admissible Viewpoints*

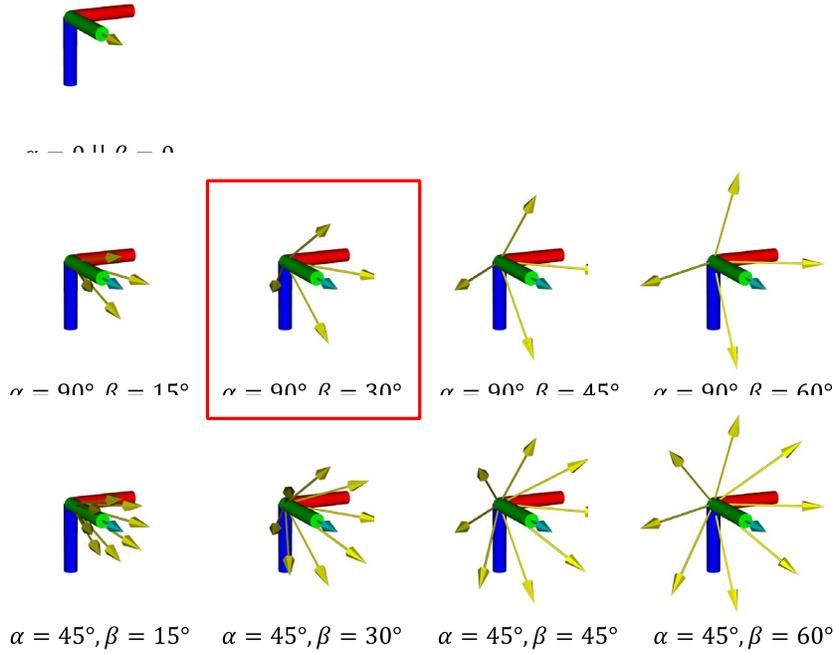
Mathematically, let  $\mathbf{m}$  ( $\mathbf{m} \in \mathcal{M}$ ) be a surface of the mesh, we define a viewpoint  $\mathbf{v}$  ( $\mathbf{v} \in \mathcal{V}$ ), is admissible if the following constraints are satisfied (Eq. 9):

$$\begin{aligned}
 \lambda(\mathbf{v}, \mathbf{m}) &\leq \text{GSD}_{max} \\
 \theta(\mathbf{v}, \mathbf{m}) &\leq \theta_{max} \\
 \varphi(\mathbf{v}, \mathbf{m}) &\in [\varphi_-, \varphi_+] \\
 \pi(\mathbf{v}, \rho) &= 1 \\
 h(\mathbf{v}) &\geq H_{min}
 \end{aligned} \tag{9}$$

Where  $\lambda$  computes the GSD of a viewpoint to the surface given the camera FOV and image resolution,  $\theta$  measures the incidence angle between the viewpoint and the normal of surface plane, and  $\varphi$  is the gimbal rotation angle. We set the initial orientation of each viewpoint as a ray casting from the viewpoint to the center of the mesh surface.  $\pi$  is a binary function that measures if the designed viewpoint is located within the UAV configuration space (i.e., no collision).  $h$  ensures the altitude of the viewpoint is AGL. These constraints formulate the viewpoint search space at each mesh surface.

#### *Spawning Oblique Orientations*

Due to the limited camera FOV and the complex truss geometry, the sampled viewpoint may be insufficient to cover every truss surface. Thus, we spawn extra orientations at each sampled position to increase the searchability. Given the initial orientation of the admissible viewpoint, the oblique orientations are symmetrically generated based on two parameters:  $\alpha$  and  $\beta$ .  $\alpha$  measures the adjacent angles between the extra orientations. The smaller of  $\alpha$ , the more oblique orientations are generated.  $\beta$  denotes the angle between the original and the oblique orientation. The larger  $\beta$  indicates the increased exploration ability of the oblique orientation. Noted the oblique orientations must also follow the gimbal constraints, the orientations with the pitch angle located outside of the gimbal limits must be rejected. Fig. 15 illustrates the oblique orientations (arrows in yellow) generated under the selection of different  $\alpha$  and  $\beta$ . In this study, we set  $\alpha = 90^\circ$  and  $\beta = 30^\circ$  based on the experiments.



**Figure 16** The collection of oblique orientations (arrows in yellow) from the initial one (arrow in cyan) based on the combination of different  $\alpha$  and  $\beta$ . Highlighted is the selected combination based on the experiments.

### *Viewpoints Subset Selection*

For dense modeling in MVS, only the geometric consistent images contribute to the final reconstruction [73]. Thus, in this section, we describe how to incorporate the geometric priors in MVS and select the viewpoints subset from a candidate set (i.e., oblique viewpoints). In the following paragraphs, we first present a quality-efficiency metric that measures the multi-view reconstruction quality given a set of viewpoints. The metric also identifies/ranks the contributed quality from each viewpoint. Second, we propose a greedy view selection algorithm to efficiently select the best viewpoint subset from a set of oblique viewpoints based on the presented quality-efficiency metric.

### *Quality-efficiency metric*

The quality-efficiency ( $F_{QE}$ ) metric is formulated as the weighted sum of the reconstruction quality ( $F_Q$ ) and the reconstruction efficiency ( $F_E$ ) as Eq.10 below:

$$F_{QE}(\mathbf{P}, \mathbf{V}) = \sigma F_Q(\mathbf{P}, \mathbf{V}) + (1 - \sigma) F_E(\mathbf{V}) \quad (10)$$

Where  $\sigma \in [0,1]$  is a constant coefficient that balances these two terms. In this study, we set  $\sigma = 0.8$  based on a thorough experiment. The presented metric encourages to obtain high-quality reconstruction from a small set of viewpoints. In the following, we discuss the computation of each term in detail.

---

*Reconstruction Quality:* The quality term predicts the reconstruction quality at each surface point on the truss, given a set of viewpoints. Due to the absence of the pixel-level contents at the planning phase, the metric is computed only based on the geometric priors at the image level [73] where the following principles are considered:

*Principle 1:* Each surface point must be covered by at least two high-quality images in terms of the sufficient GSD and the incidence angles for features matching.

*Principle 2:* Small baseline between the matched images can cause large triangulation errors for the depth interpretation.

*Principle 3:* Redundant images are uninformative views that do not reduce the depth uncertainty while can increase the computation workload.

Based on the above-mentioned principles, we formulate the quality metric as Eq. 11 below:

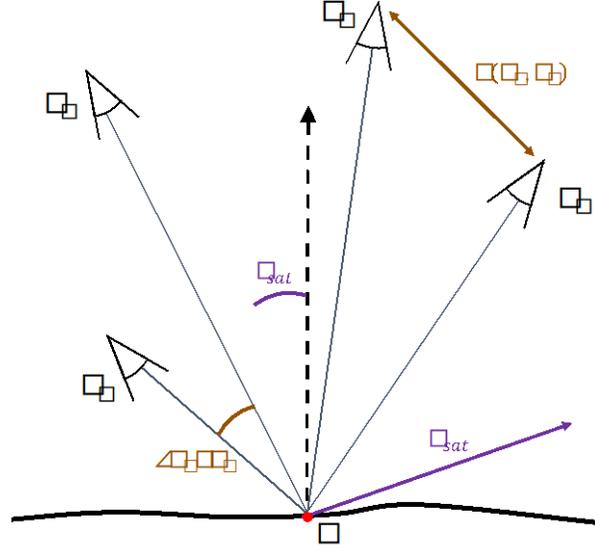
$$F_Q(\mathbf{P}, \mathbf{V}) = \frac{\sum_{\mathbf{p} \in \mathbf{P}} Q(\mathbf{p}, \mathbf{V})}{|\mathbf{P}|} \quad (11)$$

$$Q(\mathbf{p}, \mathbf{V}) = \max_k \sum_{\mathbf{v} \in \mathbf{V}_p} \tau(\mathbf{p}, \mathbf{v}) \cdot q(\mathbf{p}, \mathbf{v})$$

$$\mathbf{V}_p = \{ \mathbf{v} \mid \angle \mathbf{v} \mathbf{p} \hat{\mathbf{v}} \geq \mu; \mathbf{p} \in \mathbf{P}; \mathbf{v}, \hat{\mathbf{v}} \in \mathbf{V}; \mathbf{v} \neq \hat{\mathbf{v}} \}$$

$$q(\mathbf{p}, \mathbf{v}) = (1 - \omega) \left( 1 - \frac{|d(\mathbf{p}, \mathbf{v}) - d_{sat}|}{d_{max}} \right) + \omega \left( 1 - \frac{|\theta(\mathbf{p}, \mathbf{v}) - \theta_{sat}|}{\theta_{max}} \right)$$

Where  $Q$  measures the quality of a truss surface point  $\mathbf{p}$  ( $\mathbf{p} \subseteq \mathbf{P}$ ) as the sum of the  $k$  best observations (Principle 3). We set  $k$  equals to 3 due to the increased robustness of the three-view reconstructions at texture-less surfaces [107].  $\tau$  is a binary function that detects if the point  $\mathbf{p}$  is visible from  $\mathbf{v}$ .  $q$  measures the observation quality of each viewpoint which is computed as the average ( $\omega = 0.5$ ) of two factors: 1) the view-to-point distance; and 2) the view-to-point incidence angle (Principle 1). These two factors are normalized and saturated based on the input parameters.  $\mathbf{V}_p$  denotes the subset of viewpoints  $\mathbf{V}$  ( $\mathbf{V}_p \subseteq \mathbf{V}$ ) where the baselines at  $\mathbf{p}$  follows the stereo-matching constraints (Principle 2). Based on [108], we empirically set  $\mu = 15^\circ$  in this study. Fig.16 illustrates the geometries of the viewpoints to a surface point.



**Figure 17** The geometry between a set of viewpoints  $v_i$  and a surface point  $p$  where  $v_i \in V_p$ ,  $i = 1, 2, 3, 4$ .

*Reconstruction Efficiency:* The efficiency term measures the ratio of the non-selected viewpoints over the complete viewpoints set  $V$  (as in Eq. 12). This metric encourages to reduce the redundant images for the efficient aerial reconstruction.

$$F_E(V) = 1 - \frac{|V^*|}{|V|} \quad (12)$$

$$V^* = \{ v \mid F_Q(P, v) > 0; v \in V \}$$

$$F_Q(P, v) = F_Q(P, V) - F_Q(P, V - v)$$

Where  $F_Q(P, v)$  is the quality of the viewpoint  $v$  to every surface point  $P$ .  $V^*$  is the subset of  $V$  that contributes to the reconstruction quality  $F_Q$  (with  $F_Q(P, v) > 0$ ).

#### *Greedy views selection*

Selecting the viewpoints subset from an oblique set involves the selection of the viewpoint positions as well as best orientation at each position. Clearly, enumerating every possible combination is expensive. To make the problem tractable, we propose a greedy algorithm which includes three steps as follows:

*Step 1:* Measuring  $F_{QE}$  of the oblique viewpoints with the initial orientation at each viewpoint position. The output subset  $V^*$  is considered as the baseline for the view selection in the next step.

*Step 2:* Selecting one viewpoint in the baseline and substitute the current orientation to one oblique orientation. The current orientation of the viewpoint is updated if the overall  $F_Q$  is increased. Iterative this process to all oblique orientations at the position.

---

*Step 3:* Repeating the step 2 at every viewpoint in the  $\mathbf{V}^*$ . Stop the operation until every viewpoint has been visited or the overall quality does not increase a certain number of times (i.e., 5).

Clearly, the sequence of the viewpoint being selected (as in Step 2) significantly affects the final selection. To avoid the results being biased to a bad sequence, we perform multiple runs of step 2 in parallel with the viewpoints selected in a random order at each run. Among the different runs, the viewpoints subset  $\mathbf{V}^*$  with the maximal  $F_Q$  is chosen as the output of the algorithm.

### *Viewpoints Subset Refinement*

The selected viewpoints subset can be further refined to exploit the better solution in the search space. The main idea is to adjust the viewpoint  $\mathbf{v}^*$  in  $\mathbf{V}^*$  where  $F_Q(\mathbf{P}, \mathbf{v}^*)$  is low. The proposed viewpoints refinement is performed in two operations in sequence (as in Fig. 3 (c)): In the first operation, we resample the viewpoint with quality  $F_Q(\mathbf{P}, \mathbf{v}^*)$  less than a pre-defined threshold (i.e., 0.2). The viewpoints are updated if  $F_Q(\mathbf{P}, \mathbf{V}^*)$  is increased. In the second operation, a ratio (i.e., 25%) of the viewpoints with the lowest quality in  $\mathbf{V}^*$  are selected and incrementally mutated at positions within a defined radius (i.e., 5m). We update the mutated viewpoint if the  $F_Q(\mathbf{P}, \mathbf{V}^*)$  is increased. Preliminary results showed that this refinement step can improve the  $F_Q(\mathbf{P}, \mathbf{V}^*)$  at an average 6-8% in each iteration, while without increasing the size of  $\mathbf{V}^*$ .

### *Oblique Viewpoints Resampling*

In this subsection, we discuss the process of wrapping the oblique viewpoints sampling processed into APSO [104], based on the selected subset at each iteration. Compared to the conventional PSO, APSO is selected due to the increased convergence speed in solving multimodal optimization problems. Specifically, we define each set of oblique viewpoints as a particle and the presented quality-efficiency metric of the refined viewpoint subset as the fitness function. Eq. 13 shows the viewpoints update function adapted from the APSO.

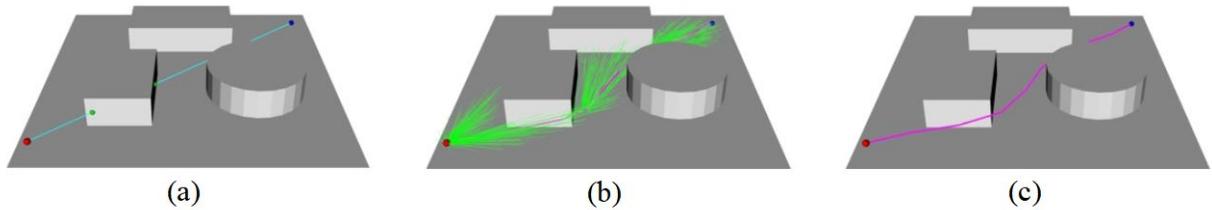
$$\begin{aligned} \mathbf{v}_{pos}^{t+1} &\leftarrow \mathbf{v}_{pos}^t + \mathbf{u}^{t+1} \\ \mathbf{u}^{t+1} &\leftarrow \mathbf{u}^t + \delta_1 \boldsymbol{\chi} + \delta_2 \nabla(\mathbf{v}_{pos}^g, \mathbf{v}_{pos}^t) \\ \nabla(\mathbf{v}_{pos}^g, \mathbf{v}_{pos}^t) &= \begin{cases} \mathbf{v}_{pos}^g - \mathbf{v}_{pos}^t, & \text{if } \mathbf{v}^g \in \mathbf{V}^{*g} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (13)$$

Where  $\mathbf{v}_{pos}$  is the position of a viewpoint, and  $\mathbf{u}$  is the particle velocity at the  $\mathbf{v}$ .  $t$  denotes the number of iterations, and  $g$  is the global best particle (i.e., viewpoint subset  $\mathbf{V}^*$  with maximal  $F_Q(\mathbf{P}, \mathbf{V}^*)$ ).  $\delta_1$  and  $\delta_2$  ( $\delta_1 = 0.8, \delta_2 = 0.5$ ) are coefficients that controls the update behavior at each viewpoint.  $\boldsymbol{\chi}$  is a standard normal distribution  $N(0,1)$  along each axis of the Euclidean space.  $\nabla(\cdot)$  is the function that measures the difference between the position of a viewpoint in a particle and the correspondent position in the global best (i.e., viewpoints correspondent to same mesh surface). We set the function to return 0 if the viewpoint is not belonging to  $\mathbf{V}^*$ , thus

encourages to resampling the viewpoint in the next iteration. Noted the presented equation (5) only updates/resamples the positions of the viewpoints. The initial and oblique orientations at each updated position need to be recomputed afterwards with the same strategy as discussed above.

### ***Trajectories Planning***

In this section, we connect the output viewpoints into the UAV executable trajectories. Our method begins with constructing a complete, undirect graph with each node indicating the position of a viewpoint and each edge as the distance of the collision-free path between every pair of the view positions. As shown in Fig. 17, the collision-free path between each pair of the viewpoints is computed in three steps: 1) we connect the viewpoints with a straight line and check if this line is collided with the on-site obstacles; 2) If a collision is found, the informed *RRT\** [109] is employed to efficiently reroute the path. If the path does not converge a given number of iterations, we recognize the path segment as not accessible, and a large penalty is assigned to the edge. 3) For each rerouted path, B-spline curve interpolation [[110] is implemented to further smooth the path segment for UAV following at the desired speed. The distance of the smoothed path is then measured as the cost of the edge between the viewpoints.



**Figure 18** Collision-free path planning procedure: (a) connect start (dot in red) and end (dot in blue) waypoints with straight line, check if collisions exist (dot in green); (b) Perform informed *RRT\** (in green) to reroute the path (in purple); (c) smooth the collision-free path using B-spline curve interpolation

Based on the constructed graph, the trajectory planning problem is then formulated as a capacitated vehicle routing problem (CVRP) [111]. To simplify the problem, we set the vehicles type and capacity as homogeneous, and let the routes start and end at the same spot (i.e., drone departure/landing). Two factors are considered as the major constraints of the problem. The first is the UAV battery capacity, which is a determinant on how long the UAV can stay in the air. The second is memory storage which limits the number of waypoints to be uploaded per flight. In this study, we employ the Lin-Kernighan-Helsgaun (LKH-3)[112] as the problem solver. The solver utilizes the improved symmetric transformation and 5-opt move generator to efficiently compute the paths while handling the battery/memory constraints. The output is a single or multiple routes, each route starts/ends at the same spot and travels through a partial of viewpoints under the imposed constraints. It is possible that the output paths still contain sharp corners that may not be tightly followed by the UAV with the desired speed. Under such conditions, the path can be either re-smoothed using the B-spline algorithm or manually checked/adjusted by the operator at the pre-flight stage. Noted while the presented method is initially developed for a single UAV to sequentially fly the paths (with replaced batteries). The method can be easily extended for multiple UAVs to fly in parallel by adjusting the flight speed of the UAVs at regions where the paths are intersected [113].

---

## Implementation Details

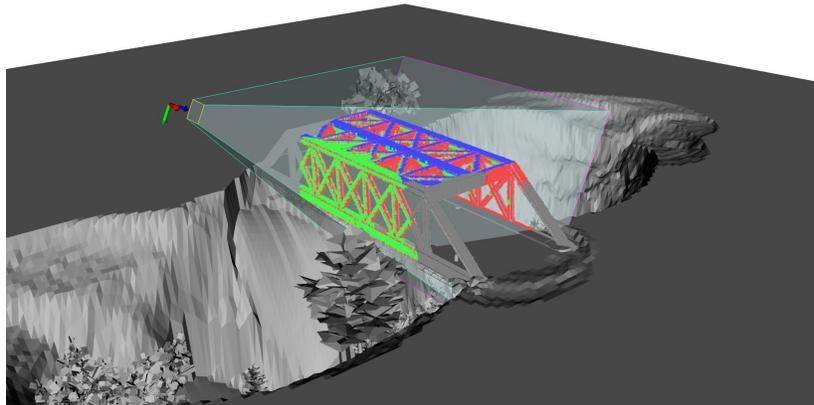
In this section, we future discuss the implementation details of the proposed method, including the visibility detection for quality evaluation, the procedures of automated flight execution, and the 3D reconstruction pipeline.

### Visibility Detection

The presented visibility detection not only considers the occlusions, but also the image triangulation properties. Given the onboard camera parameters (as in Table 2), we construct a viewing frustum to simulate the camera FOV at each viewpoint. The visibility detection is then performed in three steps:

- Step 1. We examine every surface point by checking whether the point is located within the frustum.
- Step 2. We cast a ray from the viewpoint to each point within the frustum and check whether the ray is intersected with any truss members. The surface points without intersections are considered to be visible by the viewpoint.
- Step 3. For each visible point, we measure the incidence angle between the point normal and the camera ray. Only the points with the incidence angle small than a predefined angular threshold ( $\theta_{max}$ ) are considered to be triangulable by the viewpoint.

In this study, the visibility detection algorithm is implemented using VTK [114]. Fig. 18 illustrates the visibility detection process of a single viewpoints on a synthetic truss bridge.



**Figure 19** visibility detection of a viewpoint to a truss bridge: The surface points in red are covered by the camera FOV (i.e., viewing frustum) but not visible (i.e., occluded); The surface points in blue are visible by the camera view, but the incidence angles are too large which often result in poor image triangulation; The points in green are visible and triangulable that passes the visibility detection.

### Flight Execution

Due to the flight trajectories are originally computed in the local coordinates, they need to be transformed to the World Geodetic System (WGS84) in order to be executable by a UAV. To

---

achieve that, several ground control points close to the truss bridge to be inspected are manually surveyed using a GPS receiver. Then, these GPS positions are correlated to the correspondent points in the local coordinates. Due to the relatively small scale of most bridge, the linear transformation (i.e., assume the surface is flat) is used to map the local coordinates to the WGS84 system.

After the transformation, the viewpoints are uploaded into UgCS [115], which is a ground station software, for the automated mission execution. The software contains a hardware-in-the-loop simulator that can perform the pre-flight check before the field deployment. In this study, we use DJI Inspire 1 as the flight platform to execute the missions, and DJI Zenmuse X3 for the aerial images collection. Inspire 1 can fly around 15 minutes when the wind speed is moderate. We restrict each flight to only using at most 80% of the battery capacity (i.e., 12 minutes) for the safety concerns. Noted DJI drones have a limit of at most 99 waypoints to be uploaded per flight, which also need to be considered in the trajectory planning.

### *3D Reconstruction*

After the flight executions, the collected aerial images are imported into a 3D reconstruction software. In this study, Agisoft Metashape [116] is selected since it has been previously used for the 3D reconstruction of bridge structures [117]. When the GPS of each image is available, *reference matching* is enabled to accelerate the image alignment process. To obtain the detailed reconstruction, we set the quality of both the image alignment and dense point cloud as *high* with the depth map as *aggressive* to actively filter out the noises in the final reconstruction.

## ***Evaluation***

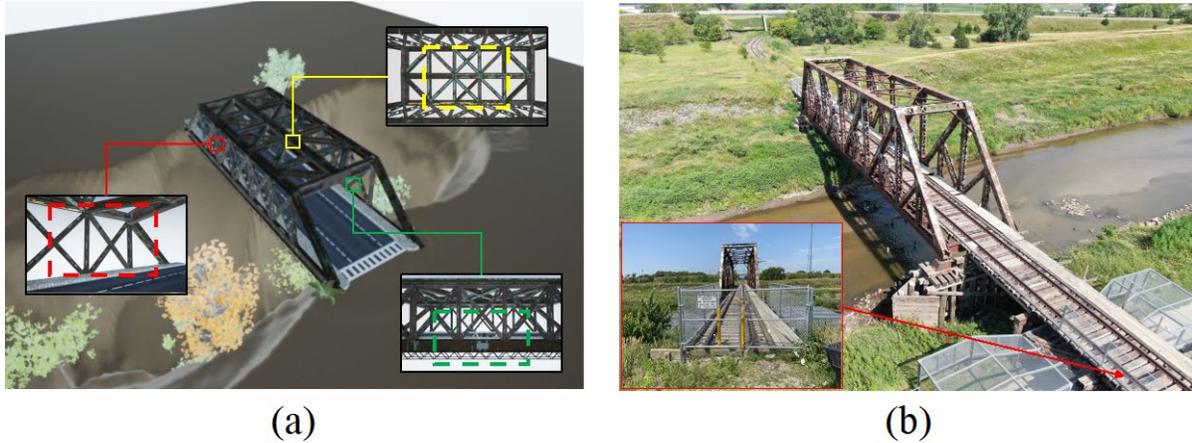
### ***Experimental Setup***

#### *Bridges Selection*

In this study, the performance of the proposed method is evaluated based on both a synthetic and a real-world truss bridge. Evaluation using the synthetic bridge has the advantage of controlling the environmental factors which are often considered as noises in image-based reconstruction (e.g., reflection, illumination change, shadows, moving objects, etc.). In this study, Unreal Engine 4 (UE4) is selected to render the synthetic environment due to the ability of providing the photo realistic scenes at high levels of details (LoDs). UnrealCV [118], an open-source computer vision SDK, is employed to render the image at each camera footprint. The selected synthetic truss bridge (as shown in Fig. 8 (a)) is a highway bridge across a valley. The bridge truss structure, which is downloaded from the Unreal Marketplace [119], has the dimension of  $66.7m \times 16.1m \times 12.0m$ . Noted the original package only contains the bridge superstructures, we import the deck surface and the surrounding environment to simulate the real-world condition. Due to the synthetic environment is noise-free, the *fly-through-truss* option is enabled ( $\rho = 1$ ) in the evaluation.

The selected truss bridge for the real-world experiment is an abandoned railway bridge located near the City Campus of the University of Nebraska-Lincoln (UNL) (as shown in Fig. 19 (b)). The bridge has the dimension of  $40m \times 4.7m \times 8.2m$  which is not accessible by the human at the time of the inspection due to the safety concerns. Noted the inner space of bridge is insufficient for the

UAV to pass through even without considering the magnetic interference. Thus, the *fly-through-truss* ( $\rho = 0$ ) is disabled in the real-world environment.



**Figure 20** (a) synthetic truss bridge and the detailed views of the selected regions; (b) the abandoned truss bridge to be surveyed, which is restricted to human access due to safety concerns (the detailed view)

### *Evaluation Methods*

Based on the authors' knowledge, there is no reported flight planning method specifically designed for the 3D reconstruction of truss bridges. Thus, a baseline and a state-of-the-art method for building reconstruction are selected to evaluate the performance of the proposed method in the simulated environment. In this study, the Overhead flight, which is composed of an orbit path with the camera surrounding the center of the scene followed by lawnmower path providing the bird views, is selected as the baseline approach. This path can be easily reproduced using the commercial flight planning software [115][120][121]. The overlap between the adjacent viewpoints is set at 80% to ensure the dense images registration. For the state-of-the-art approach, the NBV method presented in [94] is employed. The method incrementally adds the viewpoints with the largest marginal reward from a graph of the candidate cameras. The orbit path obtained from the Overhead flight is used to initialize the optimization. Noted compared to our method, the original implementation of [94] used different strategies for the space representation (i.e., voxel), the collision detection/avoidance, and the visibility detection that might affect the result. To avoid the confusion, we implement the NBV with same strategy as in our work such that the results are only affected by the optimization algorithms. Noted that the method [94] limits the viewpoints planning in a single flight (i.e., battery constraints) that might result in the incomplete reconstruction. Thus, in the simulation, we set UAV flight time as unlimited, and leave the evaluation on the trajectory planning in the field experiment.

For the field experiment, not only the reconstruction quality, but also the efficiency of the inflight images acquisition and the post-flight image processing are discussed. Thus, we select a sweep-based, multi-UAV supported route planning method [122] as the previous state-of-the-art. The method designed three routes to tightly cover the structure from different perspectives while considering the photogrammetric constraints (i.e., GSD, camera angles, overlapping, and etc.). Due to only one UAV is available (i.e., DJI Inspire 1) in the field experiment, the planning adjustment step (for multi-UAV operation) as in the existing work is skipped. Due to the method

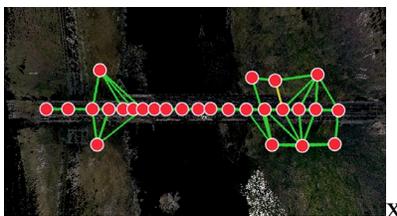
---

does not including the collision avoidance algorithm, manual check is needed to guarantee the safety of the mission.

### *Quality Evaluation*

Evaluation on the reconstruction quality includes the visual and the quantitative comparison. The visual comparison focuses on the observations of the texture smoothness and the artifacts in each model, especially at the geometric complex regions (e.g., truss interiors, connections, and slim beams). The quantitative evaluation aims to measure the geometric fidelity between the reconstruction and the ground truth. The evaluation includes three major steps: First, the reconstruction model is cropped and filtered to only contain the regions covered in the ground truth (i.e., truss bridge). Second, a coarse-to-fine alignment is used to transform the coordinate of the reconstruction to the ground truth. The coarse alignment is performed by the rigid transformation from a set of correspondence points. Based on the coarse alignment, the fine transformation is computed using iterative closest point (ICP) registration [123][124]. It is noted that the reconstructed models might contain outliers, thus RANSAC [124] is employed such that the refined transformations are robust to such outliers. Third, the F-Score as presented in [125] is used to measure the fidelity of the finely aligned model to the ground truth. F-Score measures the harmonic mean of two indicators: Precision ( $\mathcal{P}$ ) and Recall ( $\mathcal{R}$ ), given a distance threshold  $\chi$ . In general, a high F-Score indicators a reconstruction that is both accurate and complete. We refer the readers to [125] for the details of the computation of the metric. Due to each F-Score is computed with a distance threshold, thus we report the quantitative evaluation based on the F-Scores across a range of distance thresholds.

For the synthetic bridge, the ground truth model is known. Thus, the F-score can be directly computed by comparing the reconstruction to the ground truth. For the field experiment, terrestrial laser scanning (TLS) is used to obtain the ground truth model of the bridge. In this study, the Leica BLK360 laser scanner was selected. The scanner can obtain millimeter accuracy at a distance less than 10 meters which is sufficient to obtain a high-fidelity 3D model of the truss bridge. Fig. 9 shows the 2D view of TLS scanned truss bridge and the on-site scanning spots. The 3D model is registered from 27 scans with most of the scan are conducted on the bridge. The entire survey took around four hours for on-site data collection, and another five hours for the offsite data transmission and point cloud registration.



**Figure 21** 2D view of the terrestrial laser scanned truss bridge. The 27 red dots denote the positions at each scan, and the links indicate the registration between the scans. The colors of the links denote the overlapping quality between the scans (Green: at least 75% overlapping; Yellow: at least 50% overlapping).

### **3.1.3 Task 3**

Task 3 is focused on developing or identifying an application, which has the functionality of

---

comparing 3D volume changes or shifting of the pipeline route profiles from time to time, especially comparing the volume changes before and after some major hazardous events happened. The experiment design of this task is 1) to identify existing open-source applications; if an open-source application is found, the application will be evaluated through some test cases relevant to the pipeline infrastructure and the corresponding right-of-way. 2) if no available open-source application is found, the PI will develop one.

In our literature search for identifying 3D volume change calculation several applications were considered, including Autodesk Recap, MeshLab, and Cloud Compare. Although Autodesk Recap is free to academic users and we are more familiar, it is a proprietary software, not an opensource one. It was ruled out.

Between the two open-source application MeshLab and Cloud Compare, Cloud Compare appeared more user friendly and was more focused on functions of comparing changes, instead of on meshing functions as MeshLab did. The 3D cloud change comparison functions were evaluated and used in the two journal papers described in the experiments in Task 2 to compare deviations of laser-scanned cloud models to the UAS-based photogrammetry models[66][127]. Cloud Compare application was also evaluated in the PI's inline pipeline inspection project for identifying dents. It proved to be able to handle both large and small point cloud models from different data sources.

#### **3.1.4 Task 4**

The experimental design for Task 4 is to test the hypothesis that if it is possible use high-performance (in terms of image spatial resolution and temperature sensitivity) infrared camera, such as FLIR A8303sc or better, to identify the gas leakage. Two outdoor experiments were conducted at the UNL city campus. Both experiments were designed to use a household propane tank as a gas leaking source. In Experiment 1 two sets of infrared image data were gathered under different *in-situ* wind speed using the camera on a stationary tripod: 1) one image set of relatively small gas leakage and 2) another image set of a relatively larger leaking case. The relatively small or large leakage is a qualitative description and not a quantitative description.

After the experiment of using the stationary infrared camera, in Experiment 2, the FLIR A8303sc camera was mounted on a UAS of DJI-M600 Pro. Image data from two sources of gas leakage devices were captured from 16 feet above ground height while the UAS hovered vertically above the two leaking sources. Experiment 1 was conducted in summertime and Experiment 2 was conducted in wintertime.

Section 4.4 describes the details of the two experiments conducted on UNL's city campus.

#### **3.1.5 Task 5**

The experimental design of Task 5 is focused on the field validation of the developed autonomous UAS inspection methods on both gas pipelines, and relevant structures. The 3D models created through the image data obtained by the autonomous UAS system were compared against the laser scanned models to evaluate their geometric accuracy. The accuracy level was to evaluate the developed UAS control CPP algorithms.

---

## 4 RESULTS AND DISCUSSIONS

In this section the results of the experiments in each task were discussed.

### 4.1 Task 1: Testing and the results of the prototype of pipeline inspection data management system

The proposed 3D pipeline inspection data management system was implemented and tested using the partial PRCI field data to demonstrate the feasibility and the intended functions of the system. The test results indicated that the proposed system and the system components worked as expected. Figure 22 shows the graphical user interface of the implemented software application. There are 5 tabs under the dashboard, each corresponding to a specific function as the tab name suggested. The Inspection tab will show the overall defect list of the implemented PRCI pipelines. Clicking on the 3D point cloud tab will load the 3D point cloud models of the inspected pipelines. This tab can also load 3D point cloud models of the before and after-event to show the volume/distance changes of the objects (such as pipeline terrain ROW changes). Clicking on condition-based color-coding tab will bring up a color code 3D models with defects coded according to the predefined colors (Figure 26). The condition data is organized by the condition years. The dynamic defects selection tab allows users to sort and select by the defects' predefined categories. Finally, the Geo-location tab allows users to load Google Map and show the pipeline according to the geo-locations of the pipelines on Google map using Google map as a background (Figure 28).

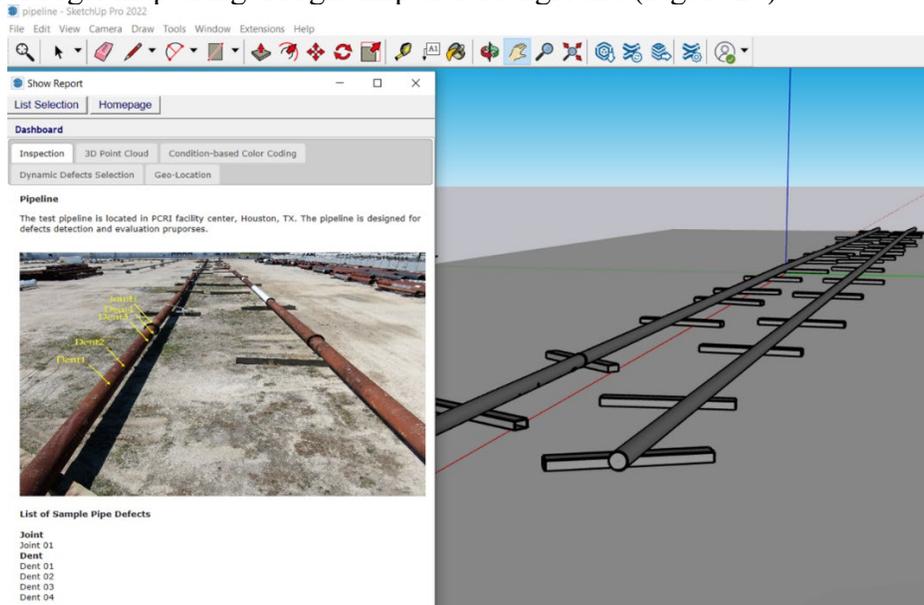
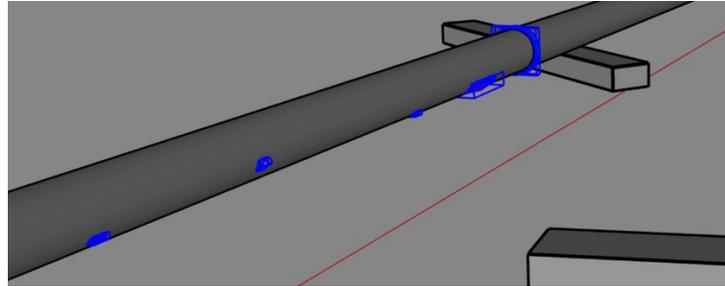


Figure 22. A demonstrative field inspection case using PRCI inspection data.

The Sketchup interface shows the documented Sketchup models of the two PRCI pipelines based on the 3D point cloud reconstructed using the developed drone flight path algorithm. Five illustrative defects were implemented in this model (shown as the blue clickable objects in Figure 23). The demonstrative defects (4 dents and 1 joint) were listed in the graphical interface. Clicking on the name of the defects will pop up the inspection report (text description and image

---

of the defects), as well as showing the location of the defects in the pipes. Please note although actual gas leakage detection was not conducted at PRCI, gas leaking can be documented the same way as other defects, such as dents, with locations on the pipeline with color codes complying with the industry practices.



**Figure 23. The implemented hypothetical defects**

The reconstructed 3D point cloud model can be loaded and viewed (Figure 24) in a web browser by clicking on tab named “3D Point Cloud”. Under this tab there are more sub-tabs for loading different 3D point cloud models, such as before-after and changes (Figure 25). The flood impact evaluation function under this tab used hypothetical data to demonstrate the intended functionality of the implemented system.

Figure 26 shows the color-coded defects map on the 3D models. Clicking on the defect will pop up a detailed inspection data (image and text (as an example in Figure 27), and can also be a 3D point cloud model).

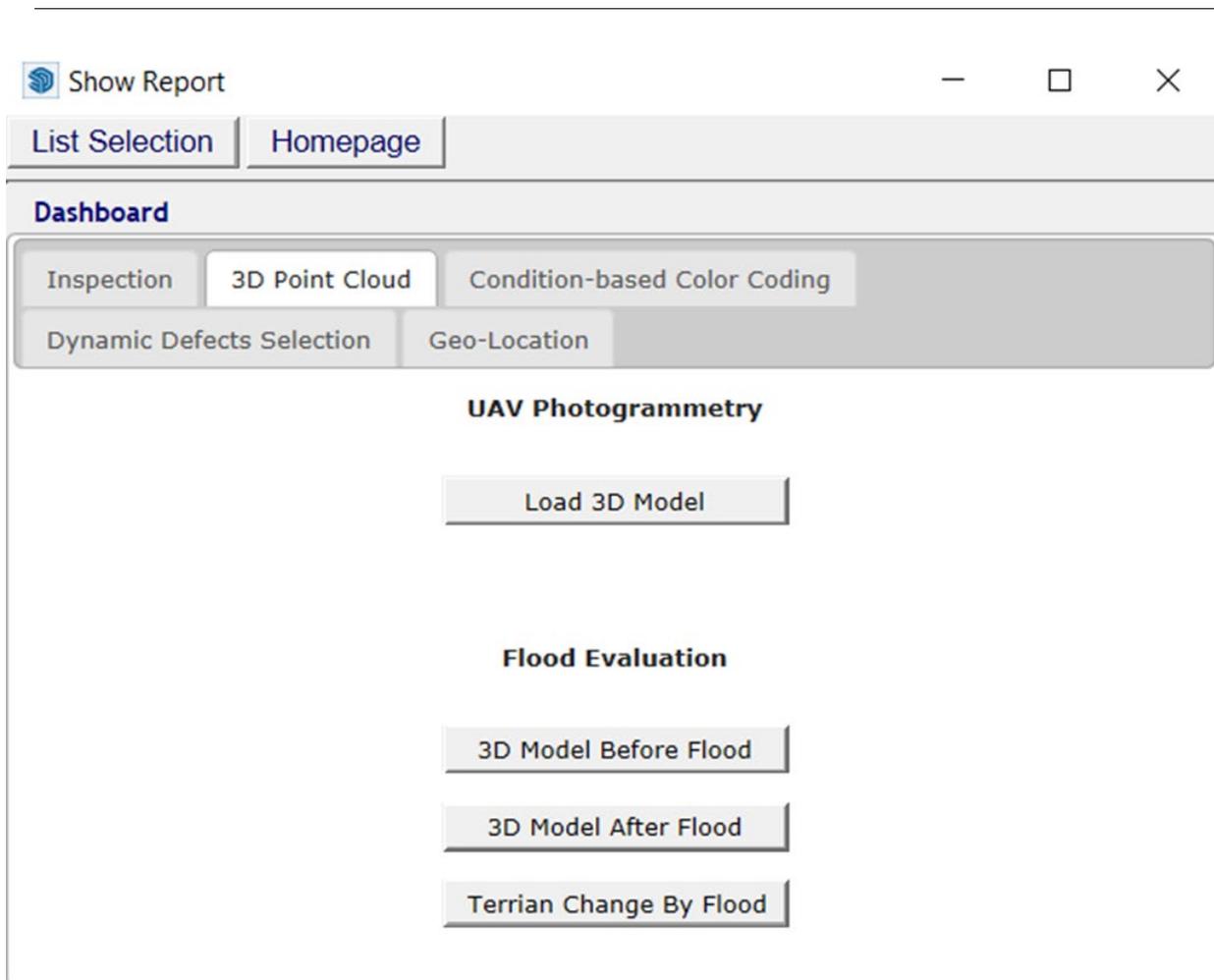
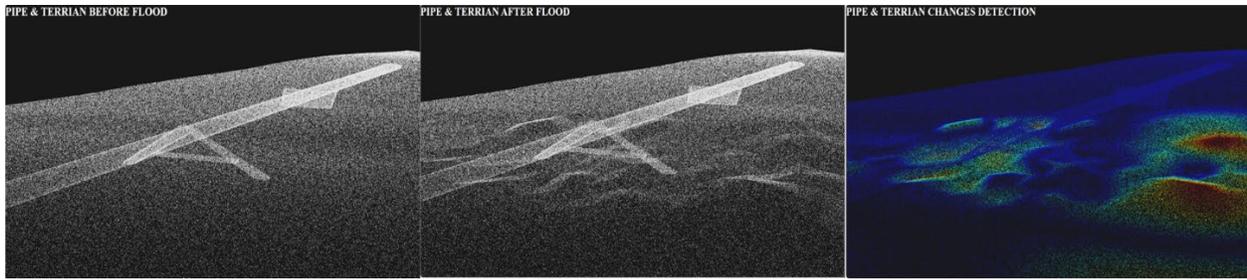
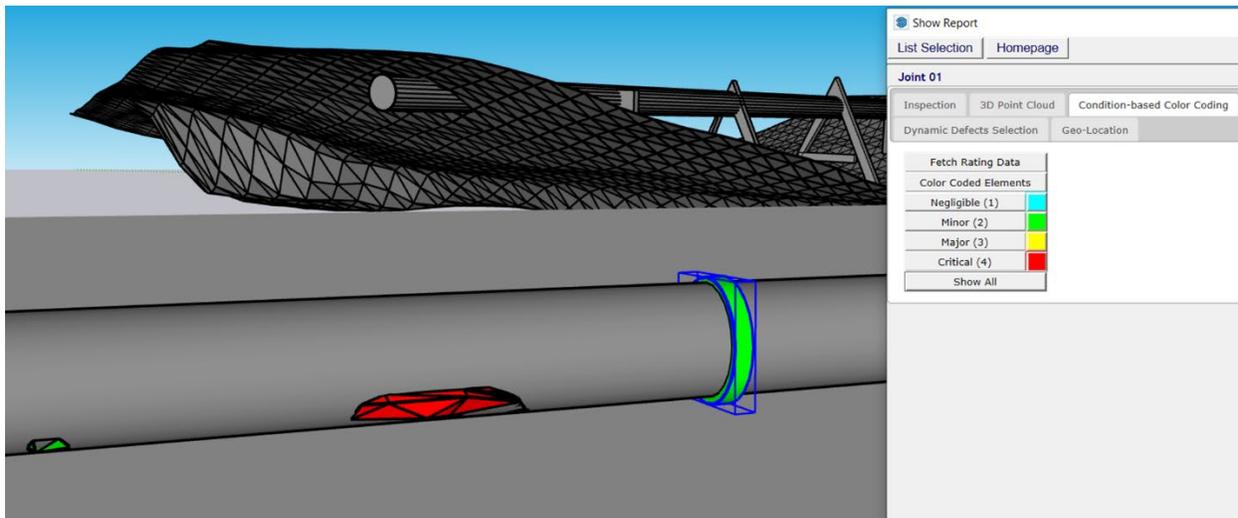


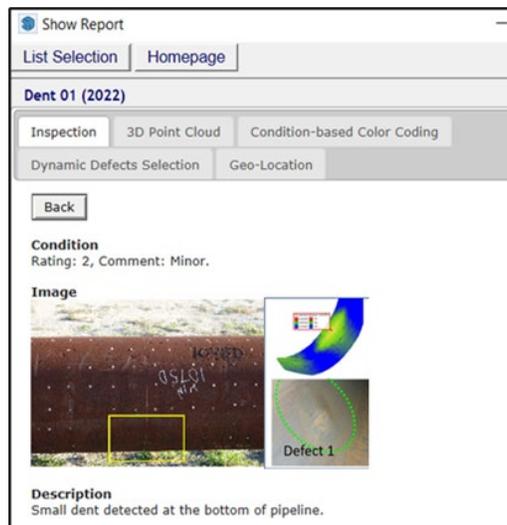
Figure 24. The point cloud model interface of PRCI field pipelines using UAS based photogrammetry.



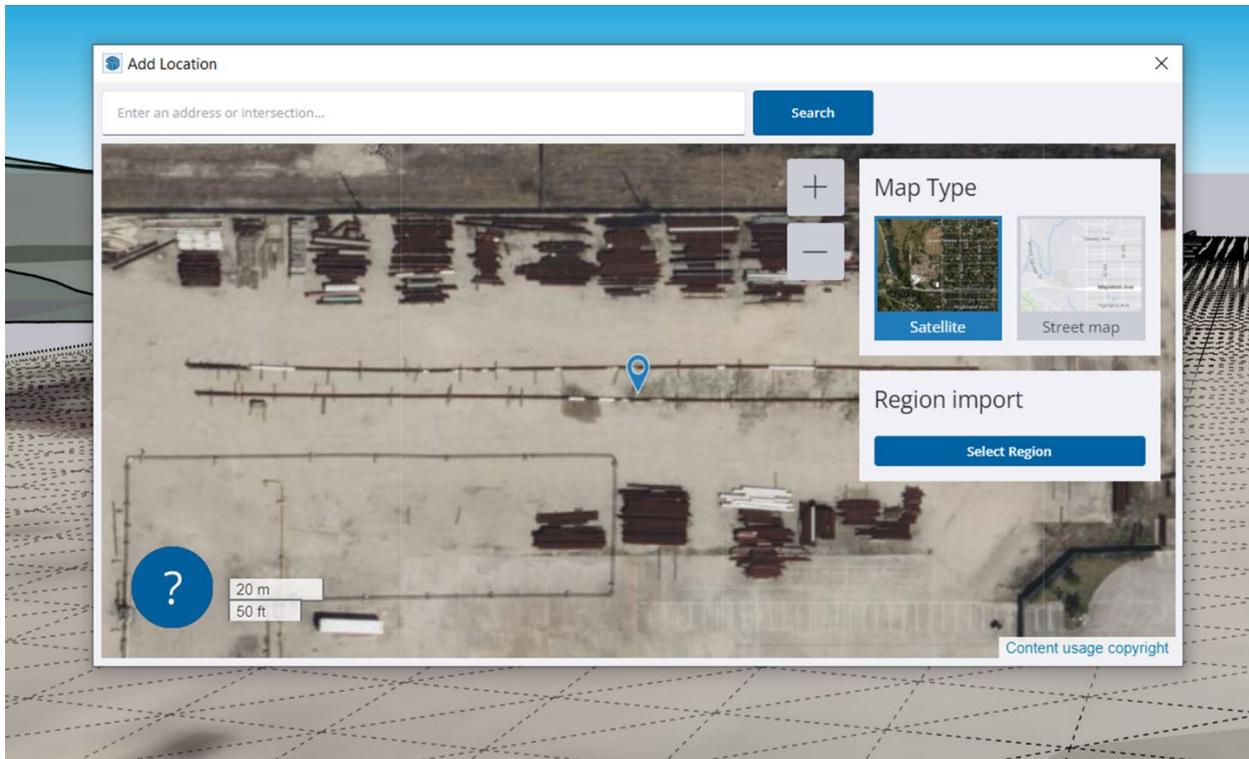
**Figure 25. Point cloud models of before (left), after (middle), and 3D changes (right) of a hypothetical flood event.**



**Figure 26. Color-coded defect conditions by year (demonstration purpose only)**



**Figure 27. Detailed data attached to the 3D clickable object**



**Figure 28. The Geo-referencing function of the system using Google Map**

We used a small sample database to implement this 3D data management system, and the results of the implementation proved the proposed system works as anticipated. The PI would like to point out that there are some restrictions or constraints to limit the open-source distribution of the implemented system. The first one is the license requirement from Sketchup, which is no longer free software, and users of this implemented system need to buy a software license from Trimble who now own the platform. Second, Google Map geo-referencing from the web browser now requires a developer authentication using Google Map security key. So, if a user would like to deploy the implemented system, the user needs to register with Google Map to become a developer. Third, the current databased system was built on MySQL free version, which might have some limitations or restrictions on the performance or data size. Any future updates either from Sketchup, Google Map or MySQL might affect the deployment of the implemented system. This is a software maintenance requirement to make the implemented system work for a longer term in the future.

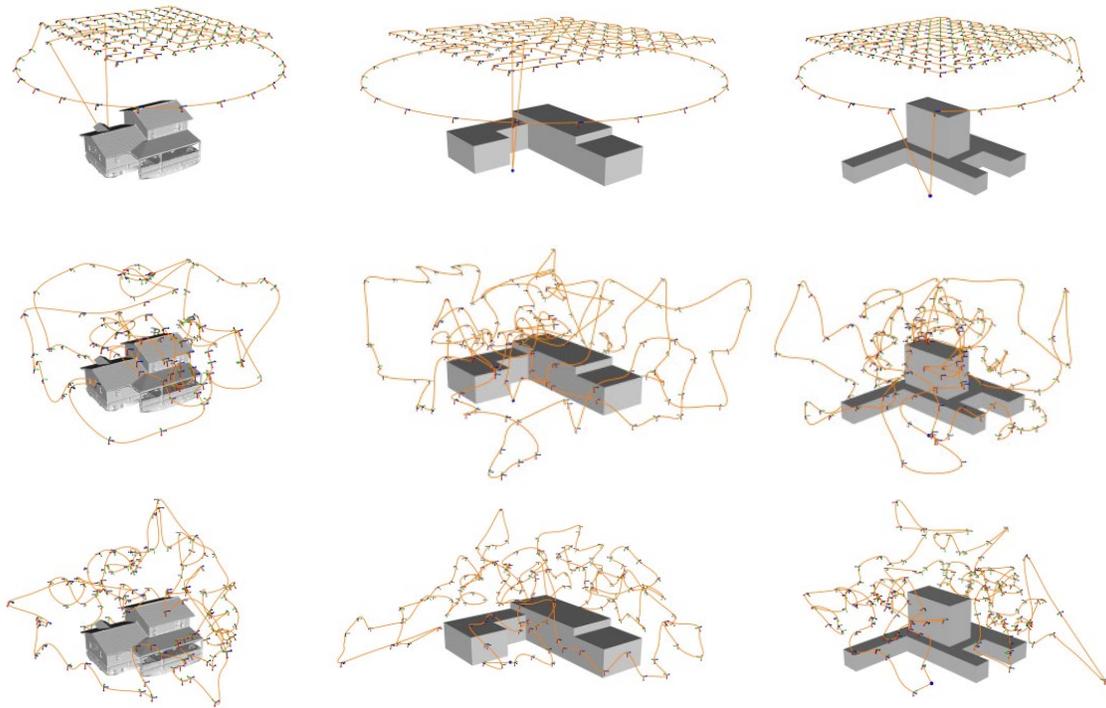
---

## 4.2 Task 2: Develop the quality-based 3D CPP algorithm(s).

### 4.2.1 Experimental Results of Task 2.1: Topology-based 3D CPP Algorithm

*Using Synthetic Scenes / Digital simulated Model*

In Fig 23, the input geometry, and the camera trajectories of the three synthetic scenes computed based on our method and the two comparison methods are presented. The exact UAV departure and landing spots are used such that the distances of the trajectories computed using different methods are comparable. Table 3 presents the statistics of the collected images and the distance of the flight trajectories on different scenes. The results show that Overhead produces relatively more images due to the required image overlay. NBV is clamped at the designated cameras before the convergence, which validates that our method keeps the number of the collected images small. Among the three methods, Overhead generates the shortest trajectory due to the closeness of the arranged cameras. Compared to the NBV, the trajectories generated from our method are slightly shorter even though we do not constrain the travel budget in either implementation.

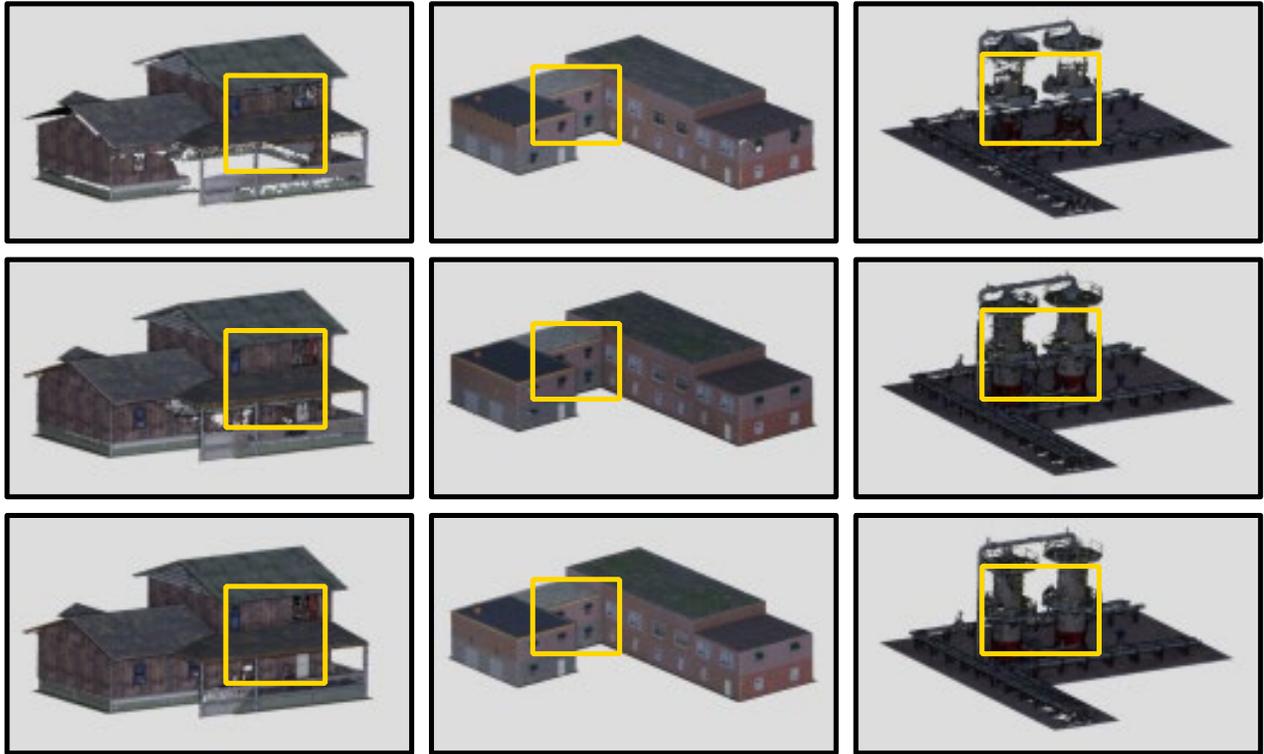


**Figure 29** The input models (including the reconstruction from orbit flights and the manually designed 2.5D) and the computed camera trajectories (in orange) of the three structures (in columns) using different methods: *Top: Overhead, Middle: NBV, Bottom: Ours*

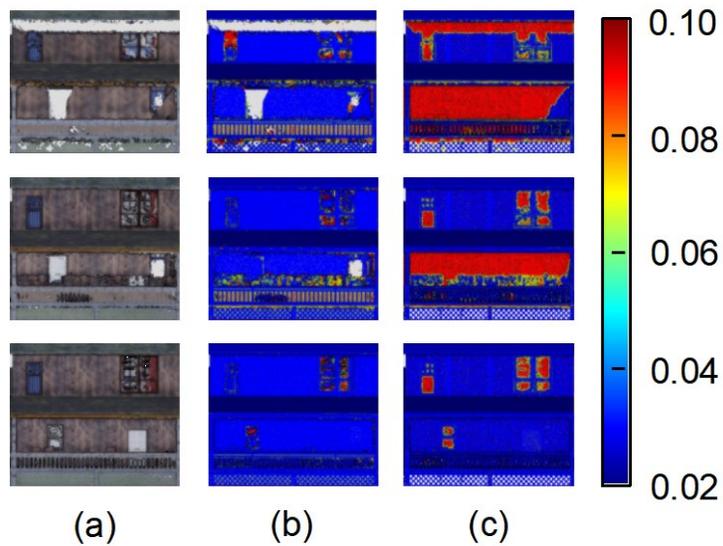
**Table 3** Statistic of the computed image views and the flight distance of the synthetic structures using the different methods.

Overhead			NBV			Ours		
Scene	#Images	Distance (m)	Scene	#Images	Distance (m)	Scene	#Images	Distance (m)
<i>Res.</i>	125	582.3	<i>Res.</i>	123	700.73	<i>Res.</i>	123	712.1
<i>Com.</i>	169	1099.1	<i>Com.</i>	163	1358.3	<i>Com.</i>	163	1080.6
<i>Ind.</i>	169	782.4	<i>Ind.</i>	164	1003.9	<i>Ind.</i>	164	816.6

Fig 24 illustrates the 3D reconstructions of the synthetic scenes. One challenging region for 3D reconstruction (the yellow box) at each scene is picked to exemplify the details of the evaluations. Specifically, for *Res.*, the selected region (as in Fig 25) is the front wall and balcony. It is observed that Overhead fails to reconstruct the walls beneath the roof and the balcony due to these regions being obstructed from the overhead flight; NBV presents better overall results than Overhead, especially for covering the wall sections. However, the balcony is still poorly reconstructed. In contrast, our method is capable of fully recovering these regions and shows the best result at both the ErrorAcc and ErrorComp maps. For *Com.*, a concave area is chosen (as in Fig 26). Both NBV and our method present fewer errors at the building corners than Overhead. Compared to NBV, our method shows slightly better results in the model accuracy, especially in reconstructing the slope window awnings. For *Ind.*, the gas tank (in Fig 27) is selected due to the geometric complexity and the texture-less at the structure surface. The slim structures (e.g., railing, ladders, pipelines) located around the tanks make it even harder for photogrammetric reconstruction. Compared to Overhead and NBV, our method shows the best results, especially at the model completeness. However, the method still fails to fully recover the detailed structures (e.g., ladders, railings). This might be caused by the relatively low detailed levels of the input geometries. Table 4 summarizes the quantitative evaluations of the reconstructions as opposed to the ground truth. The results further validate the observations: Our method shows superior results at both Acc. and Comp., especially for the *Res.* and *Com.* scenes. We observe that the NBV shows slightly better Acc. (i.e., 95%, 99%) in the *Ind.* scene. It might be partially caused by the fact that more structures are recovered by our method (i.e., higher Comp.). In addition, both methods are primarily affected by the fidelity of the input geometry. The relatively lower Acc., especially at 95% marks compared to the other scenes, shows the potential improvements of the methods. Overall, the proposed method presents superiority on the all-around reconstruction quality in terms of the more visual details and the fewer artifacts. This can be attributed to the proposed view planning strategy that cohesively optimizes the surface coverage and the stereo-matching quality at each model surface.

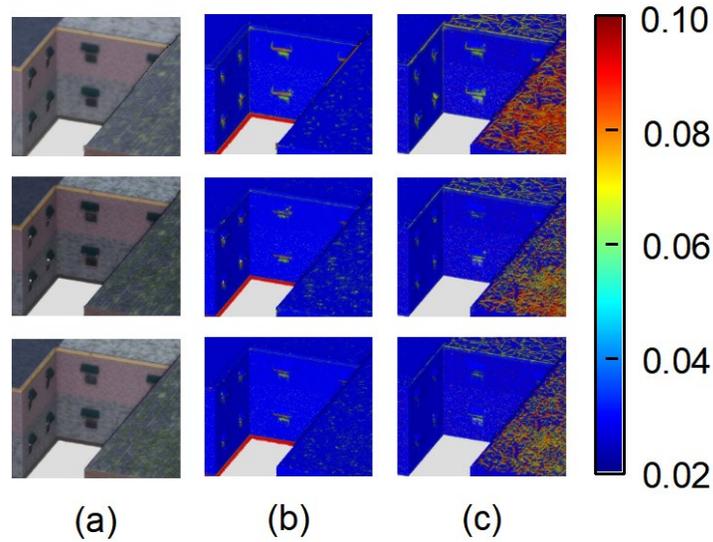


**Figure 30** 3D reconstruction of the three scenes (in columns) using different methods: Top: Overhead, Middle: NBV, Bottom: Ours. A challenging region for aerial reconstruction is chosen in each scene for the detailed comparison.

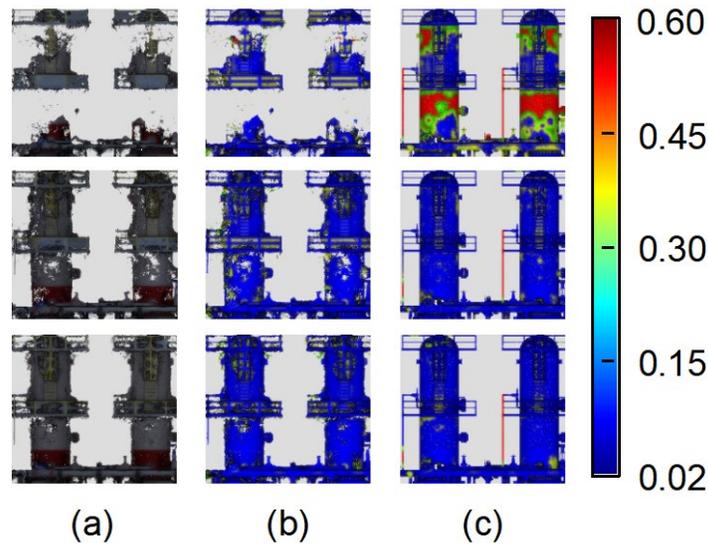


**Figure 31** Detailed comparison at the highlighted region in *Res*. Top: Overhead, Middle: NBV, Bottom: Ours. (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-coded  $Error_{Acc}$  map;(c)

enlarged view of the color-coded  $Error_{Comp}$  map. The unit of the color maps legend is meter.



**Figure 32** Detailed comparison at the highlighted region in *Com.* *Top:* Overhead, *Middle:* NBV, *Bottom:* Ours. (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-coded  $Error_{Acc}$  map; (c) enlarged view of the color-coded  $Error_{Comp}$  map. The unit of the color maps legend is meter.



**Figure 33** Detailed comparison at the highlighted region in *Ind.*: *Top:* Overhead, *Middle:* NBV, *Bottom:* Ours. (a) enlarged view of the 3D reconstruction; (b) enlarged view of the colored-coded  $Error_{Acc}$  map; (d) enlarged view of the color-coded  $Error_{Comp}$  map. The unit of the color maps legend is meter.

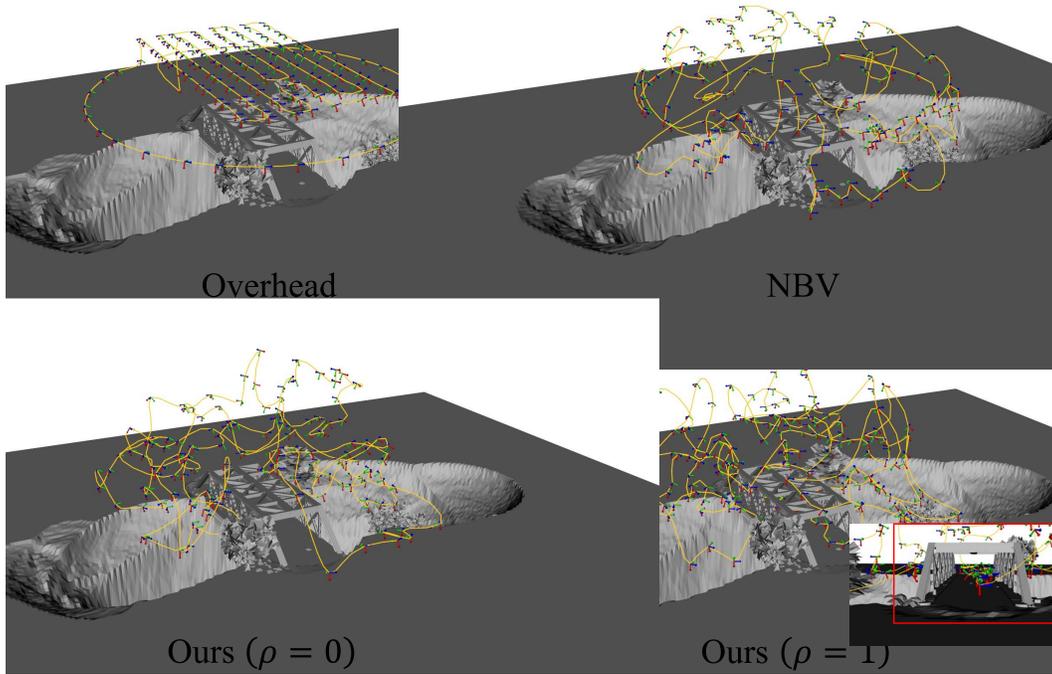
**Table 4.** Quantitative comparison of the reconstruction in terms of the model accuracy (Acc.) and the completeness (Comp.)

Scene	Method	Comp. 0.02m	Comp. 0.05m	Comp. 0.2m	Acc. 90% (m)	Acc. 95% (m)	Acc. 99% (m)
<i>Res.</i>	Overhead	34%	40%	64%	0.041	0.099	0.392
	NBV	42%	50%	73%	0.029	0.064	0.283
	Ours	<b>50%</b>	<b>57%</b>	<b>77%</b>	<b>0.020</b>	<b>0.042</b>	<b>0.268</b>
<i>Com.</i>	Overhead	45%	54%	81%	0.028	0.078	0.421
	NBV	47%	56%	<b>84%</b>	0.024	0.041	0.409
	Ours	<b>48%</b>	<b>58%</b>	83%	<b>0.023</b>	<b>0.036</b>	<b>0.384</b>
<i>Ind.</i>	Overhead	27%	38%	61%	0.120	0.175	0.361
	NBV	31%	50%	<b>68%</b>	0.104	<b>0.151</b>	<b>0.297</b>
	Ours	<b>38%</b>	<b>54%</b>	<b>68%</b>	<b>0.097</b>	0.153	0.319

#### 4.2.2 Experimental Results of Task 2.2: 3D CPP Algorithm for Truss Structure

##### *Digital truss bridge*

Figure 34 shows the trajectories computed using the Overhead, NBV and our method ( $\rho = 0,1$ ) on the digital bridge. To make fair comparison, we restrict the upper bound of NBV as the number of images generated with our method, such that the methods compute the same number of images. It observed that Overhead and NBV only generate the viewpoints surrounding the truss geometry. Instead, our method ( $\rho = 1$ ) enables the trajectories to pass through the truss (detailed view in Figure 34), provides the better observations at the truss interiors. Table 5 summarizes the statistic of the number of images and the flight distance computed with each method. Clearly, for the sake of the better surface coverage, setting  $\rho$  equal to 1 increases the overall number of images used. Figure 35 shows the visual comparison of the reconstructions between different methods. including the detailed views of two challenging areas as highlighted. The results showed that our method ( $\rho = 0$ ) can generate more visual appealing results at truss connections and beams when compared to both the Overhead and the NBV. In addition, the textures at the interiors of the top chords are mostly recovered by our method, especially when  $\rho$  equals to 1. Table 6 presents the measured F-score with varied distance thresholds ( $\chi = 0.05, 0.1, 0.2$ ). The results validate the observations that our method outperforms the other two at every distance threshold. Enabling the *fly-through-truss* option shows the best result which indicates that collecting the images from inside of the truss can indeed improve the overall reconstruction quality.



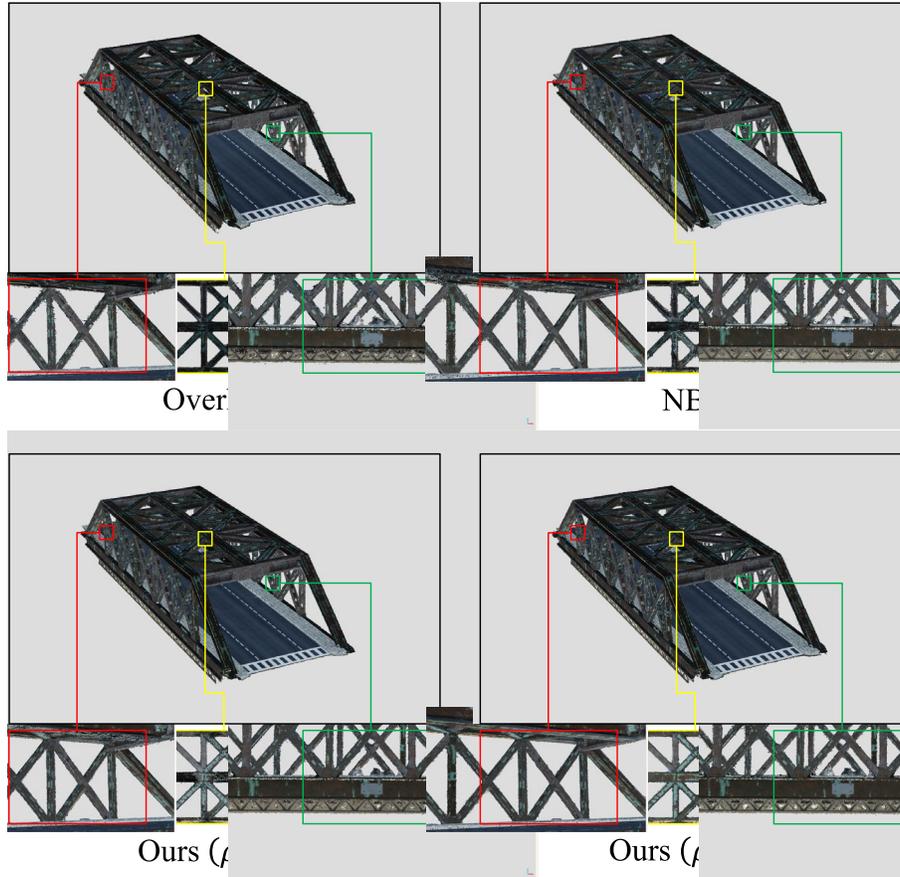
**Figure 34** The viewpoints and the trajectories generated by different methods.

**Table 5.** The number of images and the flight distance on the synthetic bridge

Methods	Number of Images	Flight Distance (meter)
Overhead	142	1039
NBV	152	1212
Ours ( $\rho = 0$ )	152	1346
Ours ( $\rho = 1$ )	181	1522

**Table 6** Quantitative comparison of the synthetic truss reconstruction between different methods.

Method	F-Score ( $\mathcal{P}, \mathcal{R}$ )		
	$\chi = 0.05$	$\chi = 0.1$	$\chi = 0.2$
Overhead	55.11 (62.07, 49.55)	74.95 (89.99, 64.21)	88.60 (96.56, 81.86)
NBV	57.51 (61.08, 54.34)	78.33 (90.10, 69.28)	90.93 (96.74, 85.79)
Ours ( $\rho = 0$ )	62.25 (63.68, 60.88)	82.45 (91.85, 74.80)	92.86 (97.35, 88.77)
Ours ( $\rho = 1$ )	<b>67.53 (66.74, 68.33)</b>	<b>85.49 (93.62, 78.65)</b>	<b>93.78 (97.81, 90.07)</b>

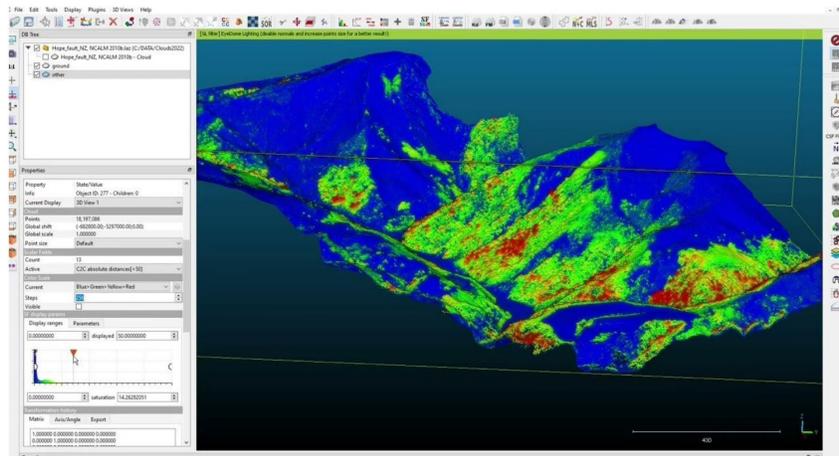


**Figure 35** Qualitative comparison on the 3D reconstruction of the synthetic bridge between different methods.

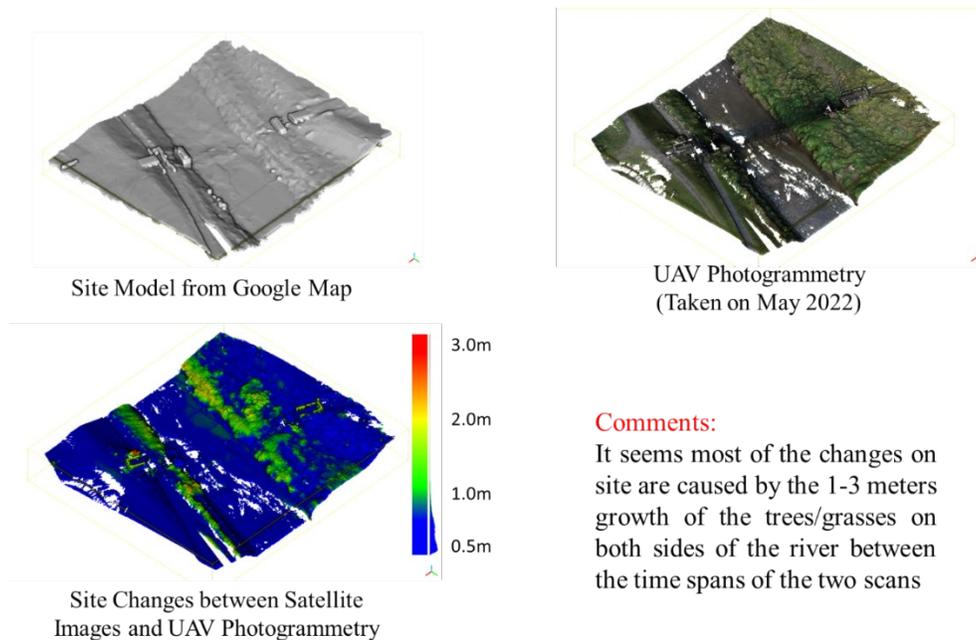
### 4.3 Task 3: Develop a 3D profile change identification and quantification module

Two sub tasks were included in this part: 1) Identify and evaluate the out-of-the-box photogrammetry software in terms of its capability and accuracy in processing a large inspection pipeline dataset and create demonstrative before-and-after 3D pipeline/route models; and 2) implement the module if necessary.

We identified a free opensource software Cloud Compare [126] to be used to identify 3D profile changes between and after potentially hazardous events to pipelines and the associated right-of-way. The quantitative volume changes can be used to evaluate the seriousness of pipeline, supporting structure, and earth moving, scour etc. which often cause unfavorable conditions to pipeline safety and operations. A threshold of volume change or relocation can be set to provide an automated alarm/alert to the pipeline operators. Figure 36 is an illustration of the comparison results of a hypothetical example of the generated 3D profile / volume change after a hazardous natural event. Figure 37 and Figure 25 show two examples using Cloud Compare software to identify 3D profile changes. The identified free application has been widely used as a freeware in the computer visions community.



**Figure 36.** An example of quantifying and locating large scale volume change along the pipeline route



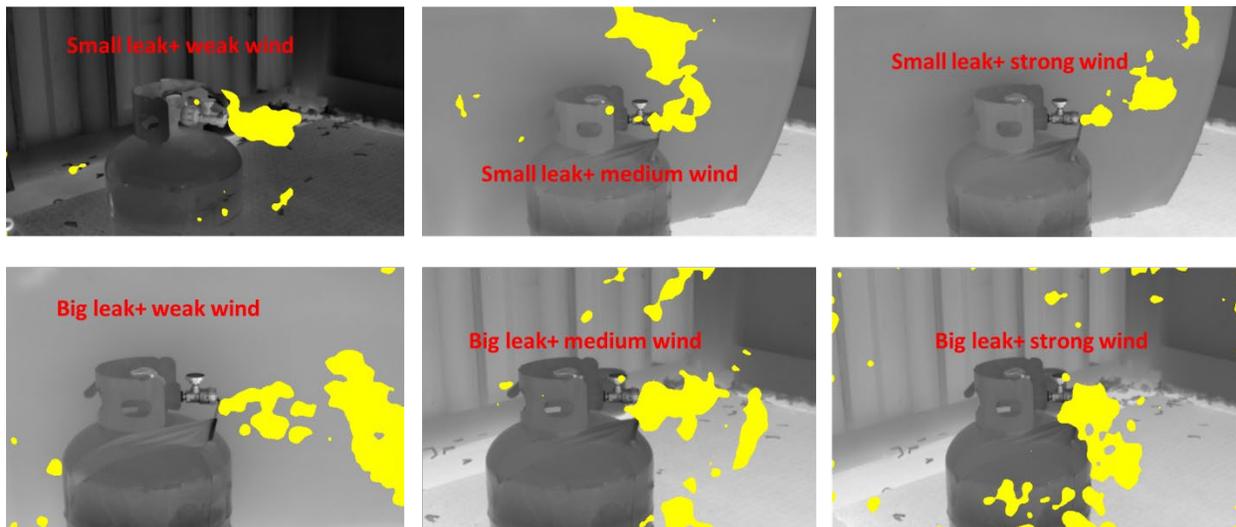
**Figure 37.** Results of the Cloud Compare using the 3D profile changes of the riverbanks underneath the truss bridge described in Section 4.5.2

#### 4.4 Task 4: Develop an infrared image processing module for leak detection

Given the success in the development of UAS close-range (15-30 feet) path planning algorithm, we can expect a UAS can take a close-range infrared video of the pipeline and potentially detect the gas leaking spots, either in the real-time or post-processing fashion. Real-time performance of gas leak detection capability will depend on the UAS on-board image processing hardware. To prove the idea, we conducted a field test to do a close-range infrared video recording of a leaking propane tank. As we expected we successfully observed the gas leaking pattern after processing the video clips. The following images (Figure 38) show the preliminary results of the test.

---

The infrared images used in the report were captured by the FLIR A8303sc camera (sensitivity 0.02 Kelvin; full-frame resolution 1280x720; maximum full-frame rate 60 HZ).



**Figure 38.** Left: outdoor propane tank gas leakage detected under different environmental conditions.

Two experiments were conducted to understand the feasibility and potential of using infrared images to detect gas leakage, which were expected to provide a fast and economical way to use UAS-based infrared camera to detect gas leakage. Experiment 1 was a ground-based experiment to use a stationary infrared camera to capture the infrared image. The stationary camera results would provide a good insight into the feasibility of the proposed method. Experiment 2 used a UAS-mounted infrared camera to capture the infrared images from 16-feet above ground to further evaluate the potential of this new approach in detecting gas leakage. The following are descriptions of the two experiments and their results.

Experiment 1 was conducted on April 27<sup>th</sup>, 2019, around 11:30 AM. The distance of the stationary camera was about 10-feet from the gas tank. The weather condition of the date was shown in Figure 40. At that time the average wind speed was about 28 mph NNW with a gust wind speed of about 43 mph. The ambient temperature was about 47 Fahrenheit degrees. On site weather station recording was not used in this experiment. Instead, local weather reporting data was used due to the qualitative evaluation nature of this study. *In-situ* wind speed was estimated between 25 to 43 mph. So, the weak wind was estimated at around 25-28 mph, the medium wind speed was estimated at around 30-34 mph, and the strong wind speed was estimated at 40-43 mph. The propane tank used in the experiment is a Blue Rhino 20-lb Steel liquid propane tank (Figure 39) with Blue Rhino Brass Propane Fuel Level-Gauge regulator (Type 1 connection) that can maintain a 0.5-2.0 psi pressure at the outlet. The small leak rate was estimated roughly at 0.5 -0.8 psi at the outlet; the big leak rate was estimated at 1.5-2.0 psi. The estimated rates were based on the observed valve knob positions. Even though the external parameters of experiment are qualitative, the results can still lead us to draw a conclusion that the infrared imaging detection method can be used to detect gas leakages based on the temperature difference between the leaked gas and the background ambient air temperature, despite adversary environmental conditions, such as strong

gust wind.



**Figure 39. The Propane Tank Used in the Experiments**



**Figure 40. Local weather conditions of the experiment**

The image processing employed a short time-series image processing algorithm by conducting pixel-to-pixel change comparisons between two raw infrared images captured at a 0.03-second time difference (at 30 fps). The pixel-by-pixel changes between the two frames (0.03 second apart) can clearly reveal the plume and the gas dispersion pattern even under the strong blowing wind (at a speed of 10-20 mph). The results of this experiment proved that the method worked theoretically with a satisfactory performance (clearly revealed the leakage and even the leaked gas profiles in the air) despite the adversary environmental conditions (strong wind).

In Experiment Two, two sizes of leakage sources were used simultaneously to evaluate the leakage detectability from the UAS platform. In addition to the 20-lb Blue Rhino tank used in Experiment One, a much smaller Coleman portable propane bottle was added to the experiment to further evaluate the detectability performance of this approach.



**Figure 41. The Coleman propane bottle used in Experiment 2**

The original plan is to use a DJI gimble mounted camera system for data acquisition, so that both UAS and camera angle/pose can be controlled simultaneously like we did in the Task 2 cases using the path planning algorithm to automatically acquire the image data. We originally planned to use our largest UAS (DJI M600 PRO) to carry the FLIR camera and all its accessories as shown.



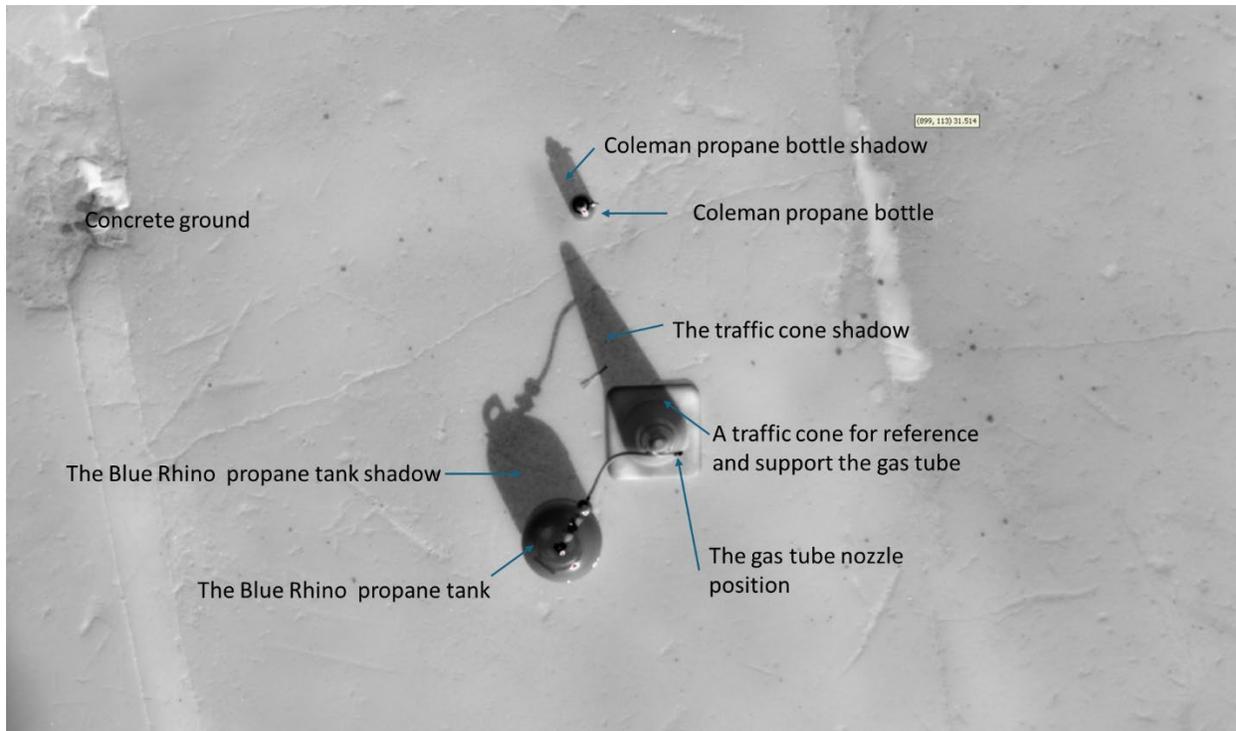
**Figure 42. Originally planned UAS system**

However, since FLIR data acquisition system requires a computer connection while capture the image. The UAS needs to carry FLIR camera (10 lbs.), two batteries and a computer (6 lbs.), plus the DJI gimble system (5 lbs.). As such the total weight of the data acquisition system way exceed the safe payload of the DJI UAS (the known largest UAS on market during the project time). As such, the research team needs to figure out an alternative way to carry the experiment. Eventually, a basketball net was used to replace the DJI gimble (Figure 43), so the total payload would not exceed the maximum payload of the M600. The price to pay is that we can only capture bird-view image with the FLIR camera axis perpendicular to the ground. Even though this case was not the ideal case as originally planned, we expect it would still provide insights for us to evaluate the UAS-based leakage detectability since it was a tougher case that just relies on the bird-view images.



**Figure 43. Basketball net mounted UAS data acquisition system**

Experiment 2 was carried out on May 2<sup>nd</sup>, 2019, at approximately 4:40 PM. The ambient temperature is approximately 60F, with a wind speed between 5-10 mph. The infrared videos were recorded from about 16 feet above ground.

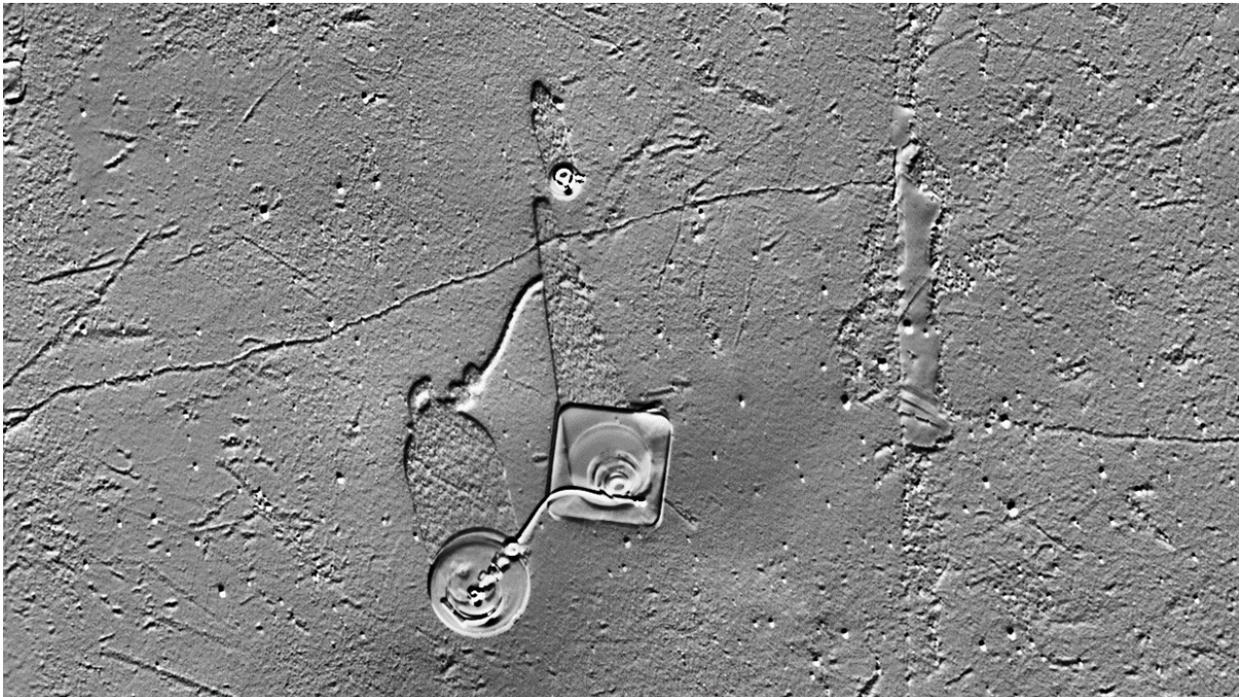


**Figure 44. Aerial view of the experiment layout (from 16 feet above ground)**

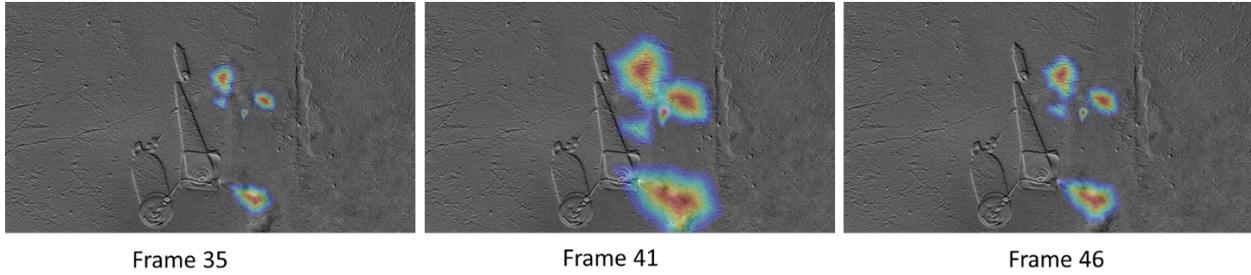
---

Compared to Experiment, the extra challenge for the image processing is the shaking and vibration of the UAS during the data acquisition process. Unlike the image processing in Experiment 1, where pixel to pixel comparison between frames can provide a good profile of the leaked gas, the same method could provide false detection result because there was no longer a true pixel to pixel correspondence between frames caused by the UAS moving and vibrations. In order to use the same image processing method, which was proved to be effective, an image pre-processing was adopted to first register/align frames based on the image features. This part was done by algorithms to automate the pixel/frame alignments. After the registration, the same method as the one used in Experiment 1 was used to extract the leaked gas profile after denoising to remove false pixels based on an estimated size threshold.

Figure 45 shows the baseline frame for the pixel-to-pixel comparison. Figure 46 shows the detection outcomes using deferent times/frames. Even though the in-situ wind speed at the nozzle was not recorded using devices, the in-situ wind speed was suspected to be the likely primary factor that affected the profiles of the dispersed leaking gas (under stronger winds the leakage profiles was smaller in size (e.g. frame 35), while under milder wind speed the profile tended to be larger in size (Frame 41). Heatmap was used to enhance the visual effects of the outcomes.



**Figure 45. Baseline frame**



**Figure 46. Detected gas leakage profiles from different times/frames**

Even though the research team was not able to use the gimble mounted UAS camera system to acquire the higher-quality infrared image data, the experiment still provided good insights into the feasibility and effectiveness of the proposed image processing methods and the developed UAS-based image acquisition system. It is anticipated that infrared images from gimble mounted infrared camera can improve the leakage detection performance since the camera can be positioned at the optimal perspective angle and distance. Different or additional image processing methods need to be adopted to address the UAS systems' movement and vibration issues, which posed extra challenges to the image processing method.

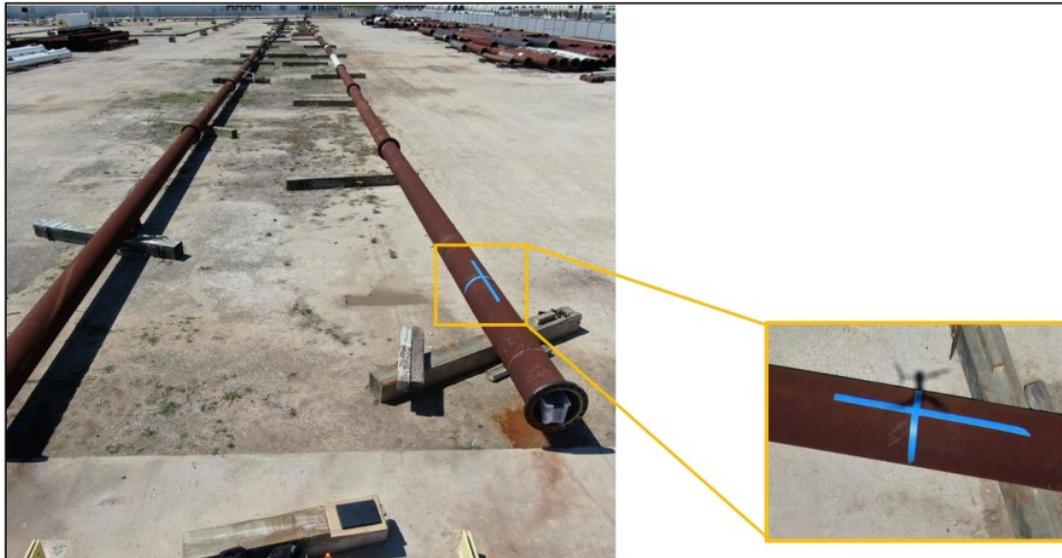
The PI can reasonably expect similar leakage detection outcomes when using our developed path planning tool to conduct a close-range UAS pipeline inspection in real-world scenarios, since both experiments were carried out outdoors with significant environmental impact

## **4.5 Task 5: Field Validation**

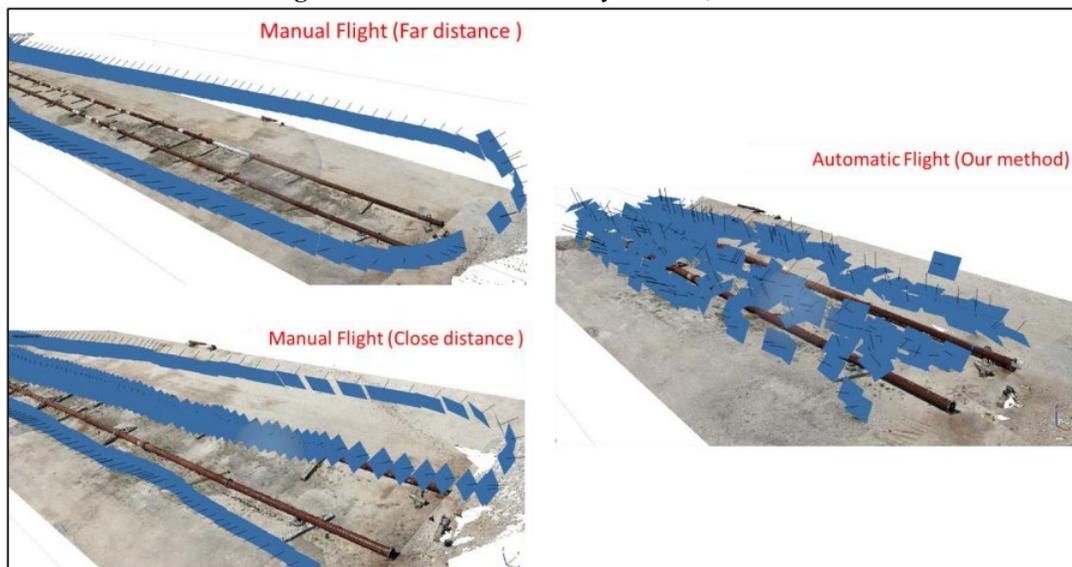
### **4.5.1 Results - Pipelines**

A 3-day field test was conducted in PRCI's Houston facility to assess the field performance of the developed path planning algorithm. The developed quality based UAS path planning algorithm (described in **Section 3.1.2.1**) was tested on some real-world pipelines with defects. The preliminary assessment indicated improved 3D modeling quality compared to manually operated UAS flights. The image data were acquired by a DJI Mavic 2 Pro rotary UAS.

Results from the preliminary field assessment indicated that the developed path planning method provide more reliable and higher quality image data for 3D reconstruction of pipelines, as Figure 47 (experimental pipeline layout) & Figure 48 (UAS/Camera Paths) show.



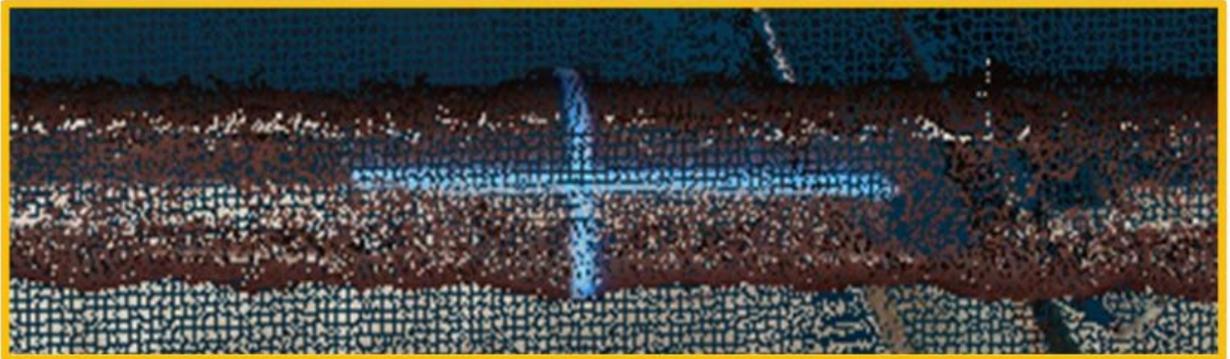
**Figure 47.** The field test facility at PRCI, Houston



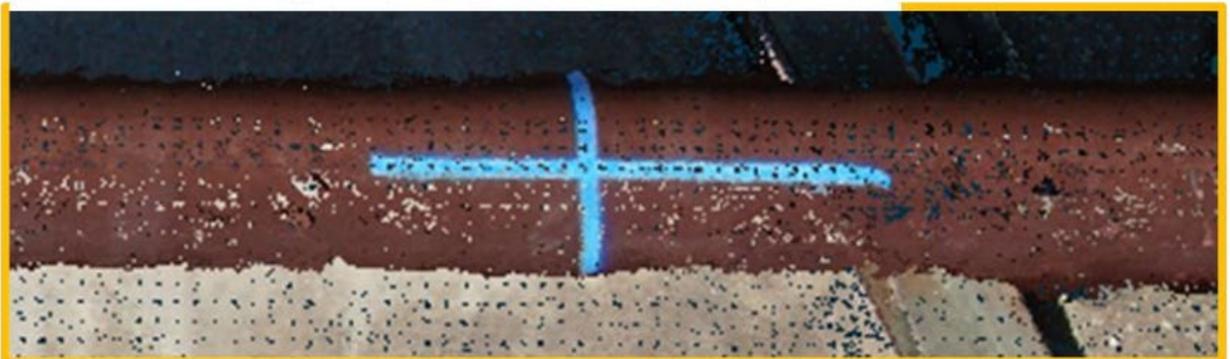
**Figure 48.** UAS flight paths by three different methods (the automatic method is based on the path planning algorithm described in Section 3.1.2.1)

---

## Manual Flight (Far distance )



## Manual Flight (Close distance )



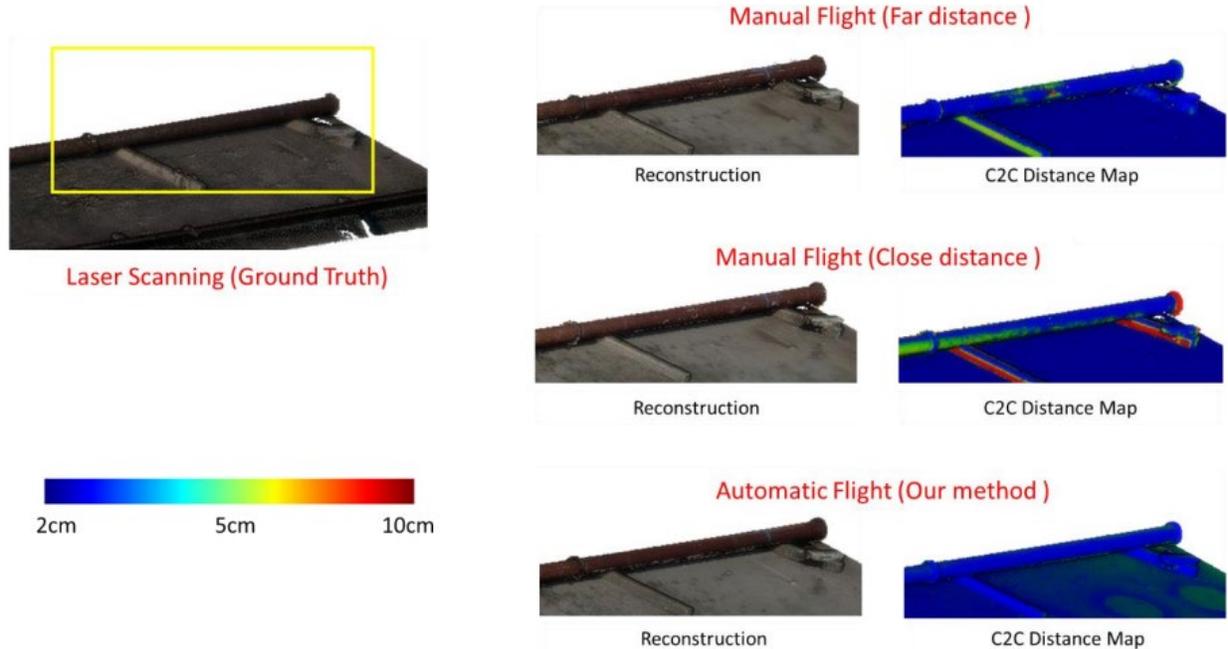
## Automatic Flight (Our method )



**Figure 49.** Prelim Visual quality comparison of photogrammetry-based 3D reconstruction using images from different path methods.

Further comparisons of the 3D pipeline reconstruction qualities of different paths (Figure 49) demonstrated the quality superiority of our developed autonomous CCP approach compared to manual UAS operations. The reconstructed three 3D pipeline models were also compared against

the laser scanned model. The results (Figure 50) also demonstrated the accuracy superiority of the developed method compared to the manual operations of UAS.



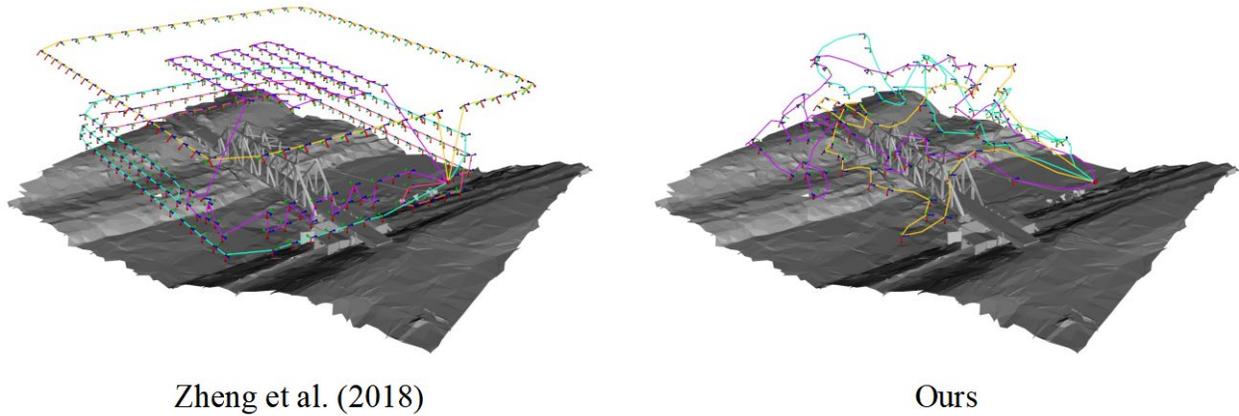
**Figure 50.** Prelim geometry quality comparison of photogrammetry-based 3D reconstruction using images from different path methods when compared to the laser-scanning results.

**Notes:**

The field test did not include a UAS-based high-performance infrared camera leakage detection experiment that was originally proposed in the project due to two reasons: (1) Technology Development Center (TDC) of Pipeline Research Council International (PRCI) did not have an outdoor leaking pipeline in their Houston facility. (2) the high-performance thermal camera system (FLIR A8303sc) owned by the PI was heavier than the UAS systems’ payload. And buying a more powerful UAS by DJI was banned by the new federal grant rule. So, instead, only the ground-based leakage detection was conducted at the UNL campus, and the results are reported in this final report.

**4.5.2 Results - Truss Structure**

Figure 51 compares the flight trajectories computed with Zheng et al. [24] and our method. It is evident that the paths generated with our method is much closer to bridge as opposed to Zheng et al. [24]. Table 7 Comparison of the statistics on flight execution and 3D reconstruction using different shows the statistics of both the inflight inspection and the post-flight reconstruction. Clearly, our method takes less images and spend number of flights/shorter time both during on-site and the offsite.

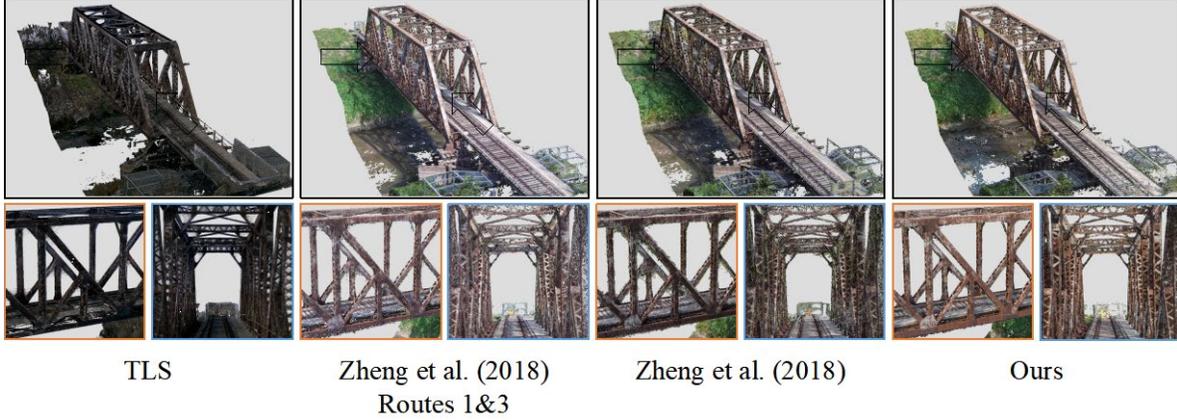


**Figure 51** The flight trajectories between the Zheng et al. [24] and our method. The colors denote the trajectories of different UAVs/flights for the mission execution.

In Figure 52, a detailed comparison between the reconstructed bridges and laser scanned model is presented. Due to the TLS model is obtained by scanning at the bridge interiors, it shows different color intensity when compared to the photogrammetry. To make a fair comparison, it is required the models to be reconstructed from a similar number of images. Thus, we also select the reconstruction only from the images taken at route 1&3 in Zheng et al.’s method Figure 52. The selected routes is an efficient setup in the existing work, which generates a similar number of images as ours. It is evident that although both methods recover the major truss structures. The model obtained from routes 1&3 only shows the worst result in terms of recovering the model details (e.g., slim beams and truss interiors.). The low-density of the point cloud shows insufficient coverage at the truss surface. In contrast to the Zheng et al [24], our reconstruction preserves more structural details with much less noise. For example, the holes in the top chord and the boundaries of the diagonal beams are much better recovered by our method (detailed views in Fig. 13). Table 8 presents the F-score of the truss reconstructions as opposed to the TLS. The results showed that our method slightly outperformed the Zheng et al. [24] in terms of both the Precision and Recall with less than half of the images been used, which validates both the efficiency and effectiveness of the proposed method.

**Table 7** Comparison of the statistics on flight execution and 3D reconstruction using different methods.

Methods	Flights	Number of Images	Flight Duration (minutes)	3D Reconstruction Duration (minutes)
Zheng et al., [24]	1	85	11.3	156.2
	2	71	10.2	
	3	77	10.4	
	4	64	9.6	
	Total	297	42.5	
Ours	1	48	10.6	63.4
	2	39	8.4	
	3	43	9.3	
	Total	<b>120</b>	<b>28.3</b>	



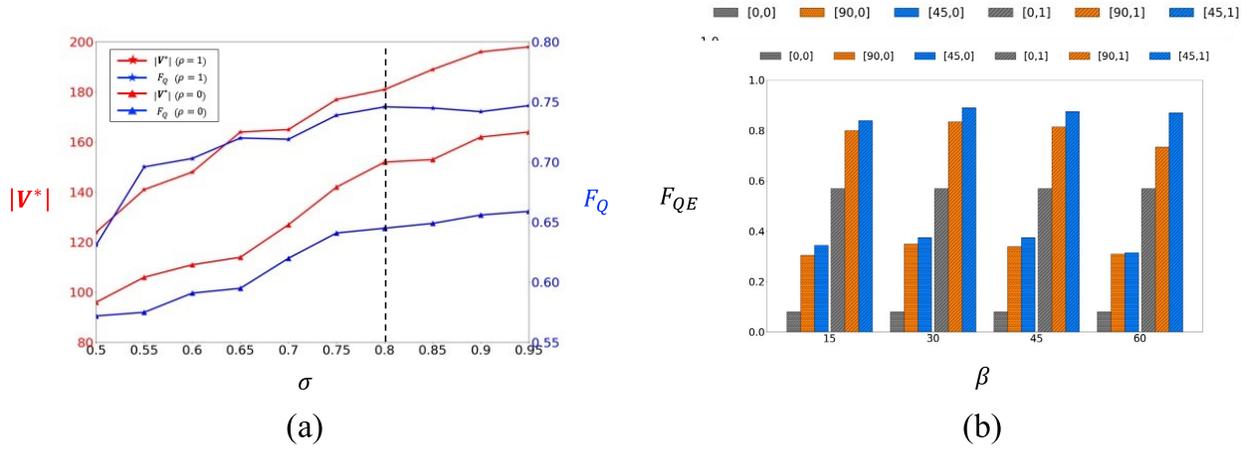
**Figure 52** The full bridge view and the detailed views of the final reconstruction of the real-world truss bridge generated using Zheng et al., (ref) and our method.

**Table 8** Quantitative comparison of the real-world truss reconstruction

Method	F-Score ( $\mathcal{P}, \mathcal{R}$ )		
	$\chi = 0.05$	$\chi = 0.1$	$\chi = 0.2$
Zheng et al. [14] (Route1&3)	45.21 (35.66, 61.75)	55.89 (41.44, 85.81)	64.87 (48.90, 96.31)
Zheng et al. [14] (Routes 1&2&3)	56.07 ( <b>43.41</b> , 79.14)	63.89 (49.12, 91.37)	71.23 (56.12, 97.47)
Ours ( $\rho = 0$ )	<b>56.31</b> (41.33, <b>80.41</b> )	<b>64.55</b> ( <b>49.83</b> , <b>91.61</b> )	<b>71.79</b> ( <b>56.75</b> , <b>97.70</b> )

### Computational Performance

In this subsection, we evaluate the performance of several parameters based on the results from both the synthetic and the real-world bridges. First, the effects of the weight coefficients  $\sigma$  on the optimization performance is evaluated. Figure 53(a) shows the quality metric  $F_Q$  and the number of collected images  $|V^*|$  under different  $\sigma$ . The figure showed that in contrast to  $|V^*|$  which increases monotonically as  $\sigma$ ,  $F_Q$  gradually decreases when  $\sigma$  close to 1. Such results might be affected by the fact that the over-redundant images cause diminished return. Due to a smaller number of images is preferred for the efficient reconstruction, thus we set the weight coefficient as 0.8 as a good trade-off between the reconstruction quality and the efficiency. Next, we evaluate the effects of the oblique orientations ( $\alpha, \beta$ ) to the quality-efficiency metric  $F_{QE}$ . As shown in Figure 53(b), compared to the conventional viewpoints ( $\alpha = 0 \parallel \beta = 0$ ), the oblique viewpoints significantly improve  $F_{QE}$  for all test cases (for both  $\rho = 0$  or  $\rho = 1$ ). The result indicates that the oblique orientations indeed increase the reconstruction quality. Among the different combinations of  $\alpha$  and  $\beta$ , we found the combination of  $\alpha = 90^\circ$  and  $\beta = 30^\circ$  shows the best result. Thus, we select it as the defaults in the experiments.



**Figure 53** (a) The values of  $F_Q$  and  $|V^*|$  when different  $\sigma$  are selected; (b) The values of  $F_{QE}$  under different combinations of  $(\alpha, \beta)$ . Noted the legend denotes  $[\alpha, \rho]$ .

---

## 5 CONCLUSIONS

The primary goal of this project is to develop an autonomous UAS pipeline inspection technological framework to enable detection of pipeline gas leakage, potential damages of pipeline and the supporting structures, soil movement along the pipeline route. The secondary goal is to develop an easy-to-use inspection data management framework to enable users to manage the inspection data through 3D model user interface. The applicable pipelines are above ground facilities.

There are two core parts in this project: 1) UAS-based accurate 3D structure reconstruction through photogrammetry. 2) UAS-based leakage detections method through infrared thermography (IRT).

Our proposed approach is through novel UAS CPP algorithms to allow accurate, low-cost, and efficient photogrammetry 3D reconstruction of complex structures of pipelines and the associated structures. The developed CCP algorithms also allow close-range UAS IRT imaging of the pipelines. The capability of the close-range IRT detection of pipeline gas leakage can only be achieved through the developed 3D CPP that controls the distance of the IRT camera to the inspected pipelines and detect the leaking spots within the IRT cameras' best effective range.

To this end, the PI developed two 3D autonomous UAS CPP algorithms to enable accurate 3D reconstruction of complex 3D pipeline structures including truss structure, a challenging structure for traditional UAS-based 3D modeling and reconstruction methods. The test results from both lab and field experiments indicated that accurate 3D structures in pipeline facilities can be reconstructed using the developed algorithms. The performance of the 3D reconstruction accuracy is comparable to traditional laser scanning method, but takes much less time compared to the time-consuming stationary terrestrial laser scanning methods.

IRT-based gas leakage detection tests were conducted in outdoor settings with different wind speed and gas leaking rates. The detecting results demonstrated that IRT video clips from high-performance IRT cameras can be used to detect gas leakages under even adversary environmental conditions such as strong wind condition, when the video clips were recorded within cameras' effective range, which can only be achieved through autonomous approach when scanning long distance pipelines in relatively short time.

The results from both experiments allowed us to conclude that the proposed data acquisition system (UAS and high-performance infrared camera system), and the proposed infrared image processing method worked, as anticipated, in a UAS-based infrared gas leakage detection approach. Experiment 1, though a stationary camera, allows us to conclude that the side-view images work just as well as the top view images in Experiment 2.

The deep learning method originally planned appears not necessarily a must-have method based on the results of both Experiment 1 and Experiment 2 with regard to different infrared image processing approach. Based on the results from both 3D autonomous path planning methods and the gas leakage detection methods, the PI can draw a conclusion that the developed 3D CPP algorithms and the developed IRT based gas leakage detection approach can work together to allow detection of pipeline hazardous defects, changes, and gas leakages with accurate locations.

One limitation in this research is that only propane type of gas leakage was experimented and evaluated. We are not able to conclude for sure that the same effectiveness of the detection can be extended to other type of gas leakages.

---

However, given the developed method is based on the temperature change and contrast against ambient environment, it should be able to apply to any type of gas leakages involving temperature contrast. Since FLIR A8303sc camera can record full-frame images (1280x720) at 60 fps, it is a good fit for using this detection method from a fast-moving platform like a rotary UAS as we did in Experiment 2 in Task 4, even with just the top view of the pipelines if future heavier payload UAS is not available.,

In related to API 653, since the developed inspection method is based on the accuracy of the reconstructed 3D models with texture, it can be used to conduct tank settlement evaluations, such as factors of out-of-plane, body-tilting, edge settlement, and other body deformations when the body changes of geometry dimensions greater than 2 centimeters. This conclusion is based on the 3D reconstruction results from both simulated (Figure 33) and real-world field cases (Figure 50). For larger complex structures such as cases involved truss structures (Figure 54), the deformation, displacement, and tilting of the inspected structures can be detected if the variances are greater than 5 centimeters based on the field test results of the truss bridge (Figure 52 and Table 8). The above-mentioned detection method is automated based on the 3D reconstructed models. This automated approach cannot be used to detect small size defects such hairline cracks due to geometry accuracy limitation. With the UAS mounted infrared camera (such the one used in this research) the developed infrared-based leakage detection method is well-suited for detecting gas leakage and local dispersion from the tank given the experiment results shown in Figure 38. Although our method can be used to help identify visible corrosion and cracks, it can only be used to detect this type of defects manually. However, image-based automated detection of such texture-based defects can be further developed based on our developed 3D reconstruction platform.



**Figure 54 An example of complex tank structure.**

In relation to API 570, we can draw the same conclusion since our developed method is based on

---

the accuracy level of the reconstructed visible 3D models. It does not matter if they are pipes or tanks or truss structures. The achieved level of geometry accuracy determines what kind of defects can be detected.

In terms of identifying Right of Way Encroachment, based on the field test result of the truss-bridge case (Figure 51, Figure 52, and Table 8), encroachment or river scour sediment or soil movement can be detected if the distance (change) value is greater than 5 centimeters.

---

## 6 FUTURE WORK

At this moment, the gas leakage detection results were generated through a post-processing image processing approach, which can be further improved in the future to allow real-time detections during the UAS autonomous flight mission. Achieving real-time gas leakage detection is critical to avoid catastrophic gas leaking accidents and save more lives.

Another area for future development is the autonomous defect detection of pipeline and its associate assets through the newly emerged deep learning methods such as the transformer methods and large language model. UAS-based pipeline inspections, while efficient and low-cost, will generate a large amount of image/video data, which are very difficult to process by human's traditional manual approach. Given the rapid advancement of AI in image processing and the breakthrough in Large Language Model in the image areas, we can reasonably expect the autonomous AI based defect and leakage detections can be achieved soon based on the current progress made in the LLM fields. .

Given the positive results from the research project, a full-scale inspection experiment on a real-world operating pipeline facility can be helpful in drawing more solid conclusions.

---

## 7 LESSONS LEARNED

There were some challenges encountered during the project's duration.

For example, in the original Task 4, the PI planned to develop a deep learning-based gas leakage detection method to detect gas leakages. However, as the project moved forward, we found there was no public validated gas leakage data in the form of infrared images. As a matter of fact, there was not many completed research on using infrared images to detect gas leakage, let alone the validated image data. Without a large quantity of validated infrared image data of gas leakage cases, it is very difficult, if possible, to develop a deep learning-based detection method.

Fortunately, the research team found a video processing approach, which can provide a good detection result based on our preliminary study. As a result, the developed gas leakage detection method was based on a traditional image processing approach instead of the originally planned deep learning approach.

The second challenge we encountered was how to mount the high-performance IRT camera and all its accessories on a market-available UAS with a gimble powerful enough to operate the heavy camera and the accessories. The payload of the PI possessed UAS was less than the weight of the camera plus all the camera controls. In this case, the risk of crashing the UAS and the IRT camera is very high while operating the gimble. However, the research team were able to find an alternative solution and carried out a key experiment to move the project forward. The results from the solution proved to be insightful.

---

## 8 REFERENCES

- [1] William Greenwood, Jerome Lynch, and Dimitrios Zekkos. 2019. "Applications of UAVs in Civil Infrastructure." *Journal of Infrastructure System*. 25: 2.
- [2] Laird Ferguson, Lindon Miller, Mike Wolfe, Jared McGinn, Lloyd Waugh. "Use of an unmanned aerial vehicle (UAV) to access transportation infrastructure, immediately after catastrophic storm event." *Resilient Infrastructure*, London, 2016.
- [3] Vidya Sudevan, Amit Shukla, and Hamad Karki. 2018. "Current and Future Research Focus on Inspection of Vertical Structures in Oil and Gas Industry." 18th International Conference on Control, Automation and Systems. PyeongChang.
- [4] Liang Yang, Juntong Qi, Jizhong Xiao, Xia Yong. 2014. "A literature review of UAV 3D path planning." The 11th World Congress on Intelligent Control and Automation. Shenyang.
- [5] Tauã Cabreira, Lisane Brisolaro, Paulo Ferreira. 2018. "Survey on Coverage Path Planning with Unmanned." *Drones* 3 (4): 38.
- [6] Brendan Englot, Franz Hover. 2012. "Sampling-Based Coverage Path Planning for Inspection of Complex Structures." Twenty-Second International Conference on Automated Planning and Scheduling. Sao Paulo.
- [7] Jiang Bian, Xiaolong Hui, Yongjia Yu, Xiaoguang Zhao, Min Tan. 2018. "A Robust Vanishing Point Detection Method for UAV Autonomous Power Line Inspection." *IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. Macau.
- [8] Björn Schäfer, Davide Picchi, Thomas Engelhardt, Dirk Abel. 2016. "Multicopter unmanned aerial vehicle for automated inspection of wind turbines." *24<sup>th</sup> Mediterranean Conference on Control and Automation*. Athens.
- [9] Amit Shukla, Hamad Karki. 2016. "Application of robotics in onshore oil and gas industry - A review Part I." *Robotics and Autonomous Systems* 75: 490-507.
- [10] Michael Vollmer, Klaus-Peter Möllmann. 2018. *Infrared Thermal Imaging: Fundamentals, Research and Applications*. Wiley-VCH.
- [11] Arvind Ravikumar, Jingfan Wang, Mike McGuire, Clay Bell, Daniel Zimmerle. 2018. "Good versus Good Enough?" Empirical Tests of Methane Leak Detection Sensitivity of a Commercial Infrared Camera." *Environmental Science & Technology* 52: 7.
- [12] Oleg Ershov, Alexey Klimov, Vladimir Vavilov. 2006. "Airborne laser IR thermographic system for detecting gas leaks from underground pipelines." *Quantitative InfraRed Thermography Journal* 3 (1): 11.
- [13] Yang Zhou, Lu Jiang, Yingcheng Lu, Wenfeng Zhan, Zhihua Mao, Weixian Qian, and Yongxue Liu. 2017. "Thermal Infrared Contrast Between Different Types of Oil Slicks on Top of Water Bodies." *IEEE Geoscience and Remote Sensing Letters*. 14 (7): 4.

- 
- [14]Arvind P. Ravikumar, Jingfan Wang, and Adam R. Brandt. 2017. "Are Optical Gas Imaging Technologies Effective For Methane Leak?" *Environmental Science & Technology* 51: 718–724.
- [15]R. el Meouche, I. Hijazi, P. A. Poncet, M. Abunemeh, and M. Rezoug, "UAV photogrammetry implementation to enhance land surveying, comparisons and possibilities," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2016, vol. 42, no. 2W2. doi: 10.5194/isprs-archives-XLII-2-W2-107-2016.
- [16]D. Bulatov, G. Häufel, J. Meidow, M. Pohl, P. Solbrig, and P. Wernerus, "Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, 2014, doi: 10.1016/j.isprsjprs.2014.02.016.
- [17]S. Valkaniotis, G. Papathanassiou, and A. Ganas, "Mapping an earthquake-induced landslide based on UAV imagery; case study of the 2015 Okeanos landslide, Lefkada, Greece," *Engineering Geology*, vol. 245, 2018, doi: 10.1016/j.enggeo.2018.08.010.
- [18]T. Rakha and A. Gorodetsky, "Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones," *Automation in Construction*, vol. 93. 2018. doi: 10.1016/j.autcon.2018.05.002.
- [19]S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1. doi: 10.1109/CVPR.2006.19.
- [20]A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell'Andrea, and M. Riley, "Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling," *Structure and Infrastructure Engineering*, vol. 14, no. 1, 2018, doi: 10.1080/15732479.2017.1330891.
- [21]K. Schmid, H. Hirschmüller, A. Dömel, I. Grixia, M. Suppa, and G. Hirzinger, "View planning for multi-view stereo 3D Reconstruction using an autonomous multicopter," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65, no. 1–4, 2012, doi: 10.1007/s10846-011-9576-2.
- [22]M. Roberts et al., "Submodular Trajectory Optimization for Aerial 3D Scanning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017-October. doi: 10.1109/ICCV.2017.569.
- [23]B. Hepp, M. Niebner, and O. Hilliges, "Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Transactions on Graphics*, vol. 38, no. 1, 2018, doi: 10.1145/3233794.
- [24]X. Zheng, F. Wang, and Z. Li, "A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction," *ISPRS Journal of Photogrammetry*

---

and Remote Sensing, vol. 146, 2018, doi: 10.1016/j.isprsjprs.2018.11.004.

- [25] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, “Multi-view stereo for community photo collections,” 2007. doi: 10.1109/ICCV.2007.4408933.
- [26] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, “Towards internet-scale multi-view stereo,” 2010. doi: 10.1109/CVPR.2010.5539802.
- [27] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys, “Pixelwise view selection for unstructured multi-view stereo,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9907 LNCS. doi: 10.1007/978-3-319-46487-9\_31.
- [28] C. Mostegel, F. Fraundorfer, and H. Bischof, “Prioritized multi-view stereo depth map generation using confidence prediction,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 143, 2018, doi: 10.1016/j.isprsjprs.2018.03.022.
- [29] C. Munkelt, A. Breitbarth, G. Notni, and J. Denzler, “Multi-view planning for simultaneous coverage and accuracy optimisation,” 2010. doi: 10.5244/C.24.118.
- [30] J. E. Banta, L. M. Wong, C. Dumont, and M. A. Abidi, “A next-best-view system for autonomous 3-D object reconstruction,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, vol. 30, no. 5, 2000, doi: 10.1109/3468.867866.
- [31] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann, “Active visual object reconstruction using D-, E-, and T-optimal next best views,” 2007. doi: 10.1109/CVPR.2007.383363.
- [32] E. Dunn and J. M. Frahm, “Next best view planning for active model improvement,” in *British Machine Vision Conference, 2009*, pp. 1–11. doi: 10.5244/C.23.53.
- [33] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, “View planning for 3D object reconstruction with a mobile manipulator robot,” 2014. doi: 10.1109/IROS.2014.6943158.
- [34] F. A. R. Martins, J. G. García-Bermejo, E. Z. Casanova, and J. R. Perán González, “Automated 3D surface scanning based on CAD model,” *Mechatronics*, vol. 15, no. 7, 2005, doi: 10.1016/j.mechatronics.2005.01.004.
- [35] X. Fan, L. Zhang, B. Brown, and S. Rusinkiewicz, “Automated view and path planning for scalable multi-object 3D scanning,” *ACM Transactions on Graphics*, vol. 35, no. 6, 2016, doi: 10.1145/2980179.2980225.
- [36] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixia, M. Suppa, and G. Hirzinger, “View planning for multi-view stereo 3D Reconstruction using an autonomous multicopter,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65, no. 1–4, 2012, doi: 10.1007/s10846-011-9576-2.
- [37] C. Hoppe et al., “Photogrammetric Camera Network Design for Micro Aerial Vehicles,”

---

in Computer Vision Winter Workshop, 2012, pp. 1–3.

- [38]M. Roberts et al., “Submodular Trajectory Optimization for Aerial 3D Scanning,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, vol. 2017-October. doi: 10.1109/ICCV.2017.569.
- [39]B. Hepp, M. Niebner, and O. Hilliges, “Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction,” ACM Transactions on Graphics, vol. 38, no. 1, 2018, doi: 10.1145/3233794.
- [40]T. Koch, M. Körner, and F. Fraundorfer, “Automatic and semantically-aware 3D UAV flight planning for image-based 3D reconstruction,” Remote Sensing, vol. 11, no. 13, 2019, doi: 10.3390/rs11131550.
- [41]N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, “Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark,” 2018. doi: 10.1145/3272127.3275010.
- [42]D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International Journal of Computer Vision, vol. 47, pp. 7–42, 2002, doi: <https://doi.org/10.1023/A:1014573219977>.
- [43]X. Zhou et al., “Offsite Aerial Path Planning for Efficient Urban Scene Reconstruction,” ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA 2020), vol. 39, no. 6, pp. 192:1--192.16, 2020.
- [44]Pipeline Research and Development (R&D) Forum 2018 Event Summary Report. Baltimore: US DOT Pipeline and Hazardous Materials Safety Administration.
- [45]A. Bircher et al., “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics,” in Proceedings - IEEE International Conference on Robotics and Automation, 2015, vol. 2015-June, no. June. doi: 10.1109/ICRA.2015.7140101.
- [46]Z. Shang, J. Bradley, and Z. Shen, “A co-optimal coverage path planning method for aerial scanning of complex structures,” Expert Systems with Applications, vol. 158, p. 113535, May 2020, doi: 10.1016/j.eswa.2020.113535.
- [47]S. Valette and J. M. Chassery, “Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening,” in Computer Graphics Forum, 2004, vol. 23, no. 3 SPEC. ISS. doi: 10.1111/j.1467-8659.2004.00769.x.
- [48]R. Bridson, “Fast poisson disk sampling in arbitrary dimensions,” in ACM SIGGRAPH Sketches, 2007, p. 1. doi: 10.1145/1278780.1278807.
- [49]O. Tziavou, S. Pytharouli, and J. Souter, “Unmanned Aerial Vehicle (UAV) based mapping in engineering geological surveys: Considerations for optimum results,” Engineering Geology, vol. 232, 2018, doi: 10.1016/j.enggeo.2017.11.004.
- [50]J. Kennedy and R. Eberhart, “Particle swarm optimization,” in IEEE International

---

Conference on Neural Networks - Conference Proceedings, 1995, vol. 4. doi: 10.4018/ijmfmp.2015010104.

- [51] C. Wang, F. Qi, and G. Shi, "Observation quality guaranteed layout of camera networks via sparse representation," 2011. doi: 10.1109/VCIP.2011.6116043.
- [52] K. Helsgaun, "An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems," Roskilde, 2017. [Online]. Available: [http://akira.ruc.dk/~keld/research/LKH/LKH-3\\_REPORT.pdf](http://akira.ruc.dk/~keld/research/LKH/LKH-3_REPORT.pdf)
- [53] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996, pp. 171–180.
- [54] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," in International Journal of Robotics Research, 2011, vol. 30, no. 7. doi: 10.1177/0278364911406761.
- [55] D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," 2014. doi: 10.1109/IROS.2014.6942976.
- [56] C. de Boor, "On calculating with B-splines," Journal of Approximation Theory, vol. 6, no. 1, 1972, doi: 10.1016/0021-9045(72)90080-9.
- [57] Dokyoo, "Modular Neighborhood Pack," Unreal Engine Marketplace, 2016. <https://www.unrealengine.com/marketplace/en-US/product/modular-neighborhood-pack>
- [58] PolyPixel, "Urban City," Unreal Engine Marketplace, 2014. <https://www.unrealengine.com/marketplace/en-US/product/urban-city>
- [59] B. Yuriy, "Industrial Vertical Vessel," Unreal Engine Marketplace, 2017. <https://www.unrealengine.com/marketplace/en-US/product/industrial-vertical-vessel>
- [60] W. Qiu et al., "UnrealCV: Virtual worlds for computer vision," 2017. doi: 10.1145/3123266.3129396.
- [61] SPH Engineering, "UgCs." 2021.
- [62] Agisoft LLC, "Agisoft Metashape User Manual: Professional Edition, Version 1.6," Agisoft LLC. 2020. [Online]. Available: [https://www.agisoft.com/pdf/metashape-pro\\_1\\_6\\_en.pdf](https://www.agisoft.com/pdf/metashape-pro_1_6_en.pdf)
- [63] T. M. Cabreira, L. B. Brisolara, and R. Ferreira Paulo, "Survey on coverage path planning with unmanned aerial vehicles," Drones, vol. 3, no. 1, 2019, doi: 10.3390/drones3010004.
- [64] C. Peng and V. Isler, "Adaptive view planning for aerial 3D reconstruction," in Proceedings - IEEE International Conference on Robotics and Automation, 2019, vol. 2019-May. doi: 10.1109/ICRA.2019.8793532.
- [65] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," IEEE

---

Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992, doi: 10.1109/34.121791.

- [66] Shang, Zhexiong, and Zhigang Shen. 2022. "Flight Planning for Survey-Grade 3D Reconstruction of Truss Bridges" *Remote Sensing* 14, no. 13: 3200.  
<https://doi.org/10.3390/rs14133200>
- [67] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, 1982, doi: 10.1016/0146-664X(82)90104-6.
- [68] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, 1996, doi: 10.1145/235815.235821.
- [69] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on BIM and UAV," *Automation in Construction*, vol. 131, 2021, doi: 10.1016/j.autcon.2021.103881
- [70] S. Valette, J. M. Chassery, and R. Prost, "Generic remeshing of 3D triangular meshes with metric-dependent discrete voronoi diagrams," in *IEEE Transactions on Visualization and Computer Graphics*, 2008, vol. 14, no. 2. doi: 10.1109/TVCG.2007.70430.
- [71] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions," 2007. doi: 10.1145/1278780.1278807.
- [72] X.-S. S. Yang and M. Karamanoglu, *Nature-Inspired Metaheuristic Algorithms Second Edition*, vol. 4, no. C. 2013.
- [73] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9907 LNCS. doi: 10.1007/978-3-319-46487-9\_31.
- [74] V. Barrile, G. Candela, A. Fotia, and E. Bernardo, "UAV Survey of Bridges and Viaduct: Workflow and Application," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11622 LNCS, 2019, doi: 10.1007/978-3-030-24305-0\_21.
- [75] G. Morgenthal et al., "Framework for automated UAS-based structural condition assessment of bridges," *Automation in Construction*, vol. 97, 2019, doi:

---

10.1016/j.autcon.2018.10.006.

- [76]X. Yiye and T. Yelda, “BrIM and UAS for bridge inspections and management,” *Engineering, Construction and Architectural Management*, vol. 27, no. 3, pp. 785–807, Jan. 2019, doi: 10.1108/ECAM-12-2018-0556.
- [77]S. Chen, D. F. Laefer, E. Mangina, S. M. I. Zolanvari, and J. Byrne, “UAV Bridge Inspection through Evaluated 3D Reconstructions,” *Journal of Bridge Engineering*, vol. 24, no. 4, 2019, doi: 10.1061/(asce)be.1943-5592.0001343.
- [78]Y. Pan, Y. Dong, D. Wang, A. Chen, and Z. Ye, “Three-dimensional reconstruction of structural surface model of heritage bridges using UAV-based photogrammetric point clouds,” *Remote Sensing*, vol. 11, no. 10, May 2019, doi: 10.3390/rs11101204.
- [79]Y. F. Liu, X. Nie, J. S. Fan, and X. G. Liu, “Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 5, 2020, doi: 10.1111/mice.12501.
- [80]A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell’Andrea, and M. Riley, “Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling,” *Structure and Infrastructure Engineering*, vol. 14, no. 1, 2018, doi: 10.1080/15732479.2017.1330891.
- [81]M. Sakuma, Y. Kobayashi, T. Emaru, and A. A. Ravankar, “Mapping of pier substructure using UAV,” 2017. doi: 10.1109/SII.2016.7844025.
- [82]J. J. Lin, A. Ibrahim, S. Sarwade, and M. Golparvar-Fard, “Bridge Inspection with Aerial Robots: Automating the Entire Pipeline of Visual Data Capture, 3D Mapping, Defect Detection, Analysis, and Reporting,” *Journal of Computing in Civil Engineering*, vol. 35, no. 2, 2021, doi: 10.1061/(asce)cp.1943-5487.0000954.
- [83]M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, “Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection,” *Automation in Construction*, vol. 81, 2017, doi: 10.1016/j.autcon.2017.04.013.
- [84]C. Eschmann and T. Wundsam, “Web-Based Georeferenced 3D Inspection and Monitoring of Bridges with Unmanned Aircraft Systems,” *Journal of Surveying Engineering*, vol. 143, no. 3, 2017, doi: 10.1061/(asce)su.1943-5428.0000221.
- [85]N. Bolourian and A. Hammad, “LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection,” *Automation in Construction*, vol. 117, 2020, doi: 10.1016/j.autcon.2020.103250.
- [86]C. Peng and V. Isler, “Adaptive view planning for aerial 3D reconstruction,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2019, vol. 2019-May. doi: 10.1109/ICRA.2019.8793532.
- [87]X. Zheng, F. Wang, and Z. Li, “A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction,” *ISPRS Journal of Photogrammetry*

---

and Remote Sensing, vol. 146, 2018, doi: 10.1016/j.isprsjprs.2018.11.004.

- [88] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on BIM and UAV," *Automation in Construction*, vol. 131, 2021, doi: 10.1016/j.autcon.2021.103881.
- [89] C. Eschmann, C. M. Kuo, C. H. Kuo, and C. Boller, "Unmanned aircraft systems for remote building inspection and monitoring," in *Proceedings of the 6th European Workshop - Structural Health Monitoring 2012, EWSHM 2012*, 2012, vol. 2.
- [90] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixia, M. Suppa, and G. Hirzinger, "View planning for multi-view stereo 3D Reconstruction using an autonomous multicopter," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65, no. 1–4, 2012, doi: 10.1007/s10846-011-9576-2.
- [91] C. Hoppe et al., "Photogrammetric Camera Network Design for Micro Aerial Vehicles," 2012.
- [92] A. Bircher et al., "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2015, vol. 2015-June, no. June. doi: 10.1109/ICRA.2015.7140101.
- [93] M. Roberts et al., "Submodular Trajectory Optimization for Aerial 3D Scanning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017-October. doi: 10.1109/ICCV.2017.569.
- [94] B. Hepp, M. Niebner, and O. Hilliges, "Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Transactions on Graphics*, vol. 38, no. 1, 2018, doi: 10.1145/3233794.
- [95] T. Koch, M. Körner, and F. Fraundorfer, "Automatic and semantically-aware 3D UAV flight planning for image-based 3D reconstruction," *Remote Sensing*, vol. 11, no. 13, 2019, doi: 10.3390/rs11131550.
- [96] Z. Shang, J. Bradley, and Z. Shen, "A co-optimal coverage path planning method for aerial scanning of complex structures," *Expert Systems with Applications*, vol. 158, p. 113535, May 2020, doi: 10.1016/j.eswa.2020.113535.
- [97] E. Dunn and J. M. Frahm, "Next best view planning for active model improvement," 2009. doi: 10.5244/C.23.53.
- [98] E. Banta, L. M. Wong, C. Dumont, and M. A. Abidi, "A next-best-view system for autonomous 3-D object reconstruction," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 30, no. 5, 2000, doi: 10.1109/3468.867866.
- [99] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, 1999, doi: 10.1109/34.799908.

- 
- [100] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, 1982, doi: 10.1016/0146-664X(82)90104-6.
- [101] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, 1996, doi: 10.1145/235815.235821.
- [102] S. Valette, J. M. Chassery, and R. Prost, "Generic remeshing of 3D triangular meshes with metric-dependent discrete voronoi diagrams," in *IEEE Transactions on Visualization and Computer Graphics*, 2008, vol. 14, no. 2. doi: 10.1109/TVCG.2007.70430.
- [103] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions," 2007. doi: 10.1145/1278780.1278807.
- [104] X.-S. S. Yang and M. Karamanoglu, *Nature-Inspired Metaheuristic Algorithms Second Edition*, vol. 4, no. C. 2013.
- [105] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9907 LNCS. doi: 10.1007/978-3-319-46487-9\_31.
- [106] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," 2007. doi: 10.1109/ICCV.2007.4408933.
- [107] M. Rumpler, A. Irschara, and H. Bischof, "Multi-view stereo: Redundancy benefits for 3d reconstruction," ... of the 35th Workshop of the ..., no. May, 2011.
- [108] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," 2010. doi: 10.1109/CVPR.2010.5539802.
- [109] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," 2014. doi: 10.1109/IROS.2014.6942976.
- [110] C. de Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, no. 1, 1972, doi: 10.1016/0021-9045(72)90080-9.
- [111] M. M. Solomon, "ALGORITHMS FOR THE VEHICLE ROUTING AND SCHEDULING PROBLEMS WITH TIME WINDOW CONSTRAINTS.," *Operations Research*, vol. 35, no. 2, 1987, doi: 10.1287/opre.35.2.254.
- [112] K. Helsgaun, "An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems," 2017. [Online]. Available: [http://akira.ruc.dk/~keld/research/LKH/LKH-3\\_REPORT.pdf](http://akira.ruc.dk/~keld/research/LKH/LKH-3_REPORT.pdf)
- [113] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, 2018, doi: 10.1016/j.conengprac.2018.03.002.

- 
- [114] W. Schroeder, K. Martin, and B. Lorensen, “The Visualization Toolkit (VTK),” Open Source, no. July, 2018.
- [115] SPH Engineering, “UgCs.” 2021.
- [116] Agisoft LLC, “Agisoft Metashape User Manual: Professional Edition, Version 1.6,” Agisoft LLC. 2020. [Online]. Available: [https://www.agisoft.com/pdf/metashape-pro\\_1\\_6\\_en.pdf](https://www.agisoft.com/pdf/metashape-pro_1_6_en.pdf)
- [117] V. Barrile, G. Candela, A. Fotia, and E. Bernardo, “UAV Survey of Bridges and Viaduct: Workflow and Application,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11622 LNCS, 2019, doi: 10.1007/978-3-030-24305-0\_21.
- [118] W. Qiu et al., “UnrealCV: Virtual worlds for computer vision,” 2017. doi: 10.1145/3123266.3129396.
- [119] WeekendWarriors, “Rusty Beams,” Unreal Engine Marketplace, 2017. <https://www.unrealengine.com/marketplace/en-US/product/rusty-beams>
- [120] S. Pix4D, “Pix4Dmapper 4.1 User Manual.” Lausanne, Switzerland, 2017. [Online]. Available: [www.pix4d.com](http://www.pix4d.com)
- [121] DJI, “Ground Station.” 2018. [Online]. Available: <https://www.dji.com/ground-station-pro>
- [122] X. Zheng, F. Wang, and Z. Li, “A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 146, 2018, doi: 10.1016/j.isprsjprs.2018.11.004.
- [123] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, 1992, doi: 10.1109/34.121791.
- [124] M. A. Fischler and R. C. Bolles, “Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.
- [125] A. Knapitsch, J. Park, Q. Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” in ACM Transactions on Graphics, 2017, vol. 36, no. 4. doi: 10.1145/3072959.3073599.
- [126] CloudCompare (version 2.12.4) [GPL software]. (2022). Retrieved from <http://www.cloudcompare.org/>
- [127] Shang, Z., & Shen, Z. (2023). Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures. *Complex & Intelligent Systems*, 9(1), 909-926.