3-22-2018

Assessment of Structure from Motion for Reconnaissance Augmentation and Bandwidth Usage Reduction

Jonathan B. Roeber

ASSESSMENT OF STRUCTURE FROM MOTION FOR RECONNAISSANCE AUGMENTATION AND BANDWIDTH USAGE REDUCTION

THESIS

Jonathan B. Roeber, Captain, USAF

AFIT-ENG-MS-18-M-055

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-18-M-055

ASSESSMENT OF STRUCTURE FROM MOTION FOR RECONNAISSANCE AUGMENTATION AND BANDWIDTH USAGE REDUCTION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Jonathan B. Roeber, BS

Captain, USAF

March 2018

AFIT-ENG-MS-18-M-055

ASSESSMENT OF STRUCTURE FROM MOTION FOR RECONNAISSANCE AUGMENTATION AND BANDWIDTH USAGE REDUCTION

THESIS

Jonathan B. Roeber, BS
Captain, USAF

Committee Membership:

Dr. Scott Nykl
Chair

Dr. Scott Graham
Member

Dr. Robert Leishman
Member

AFIT-ENG-MS-18-M-055

Abstract

Modern militaries rely upon remote image sensors for real-time intelligence. A typical remote system consists of an unmanned aerial vehicle, or UAV, with an attached camera. A video stream is sent from the UAV, through a bandwidth-constrained satellite connection, to an intelligence processing unit. In this research, an upgrade to this method of collection is proposed. A set of synthetic images of a scene captured by a UAV in a virtual environment is sent to a pipeline of computer vision algorithms, collectively known as Structure from Motion. The output of Structure from Motion, a three-dimensional model, is then assessed in a 3D virtual world as a possible replacement for the images from which it was created.

This study shows Structure from Motion results from a modifiable spiral flight path and compares the geoaccuracy of each result. A flattening of height is observed, and an automated compensation for this flattening is performed. Each reconstruction is also compressed, and the size of the compression is compared with the compressed size of the images from which it was created. A reduction of 49-60% of required space is shown.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

ASSESSMENT OF STRUCTURE FROM MOTION FOR RECONNAISSANCE AUGMENTATION AND BANDWIDTH USAGE REDUCTION

I. Introduction

1.1 Basic Intelligence Imagery Collection

Modern militaries rely upon remotely-transmitted imagery for intelligence, surveillance, and reconnaissance. This imagery is typically transmitted from a collection device, such as an unmanned aerial vehicle (UAV), to an analysis center through a bandwidth-constrained satellite connection.

As defense budgets are scrutinized more heavily now than in the past, it is becoming cost- and time-prohibitive to increase available bandwidth by developing and deploying new military satellites. The increasingly uncertain fiscal future suggests that a stronger emphasis on cost-effective technologies and techniques may be critical for continued United States military strength. Development and deployment of a new satellite can cost hundreds of millions of dollars; while this may have been a reasonable expense in the past, new budgets may be less accommodating and consider it an excessive cost instead.

Additionally, investments in new satellites become risky as nations increasingly contest the space environment. In 2007, the People's Republic of China tested an anti-satellite missile, destroying a defunct Chinese weather satellite and generating an estimated 35,000 pieces of debris [1]. Models have been created to show the hazards of increased space debris, and intentional military actions to destroy objects in space increase the threat of second-order collisions exponentially [2]. As time goes on, the

threat of intentional or accidental damage to satellites increases; new investments must weigh the fiscal cost of satellite production against the increasing risk of space collisions rendering the satellites inoperable.

An alternative approach to alleviate the bandwidth constriction is to decrease the amount of bandwidth required to send information from collection devices to intelligence analysts.

1.2 Reduced Bandwidth Requirement via Structure from Motion

One approach to reduce the required bandwidth is to preprocess imagery on the collection device and transmit only useful information back to the analysis center. As mobile computing power increases, this strategy becomes more feasible. Improvements in computer vision algorithms further support a preprocessing approach. UAVs which once operated alone and acted essentially as always-on security cameras may be augmented to operate in swarms which collect and intelligently process images in real-time. In such a setup, only information deemed relevant would be sent back to an analysis center, reducing the amount of bandwidth required, and the information could be augmented with additional geospatial information which previously required too much computation power for feasible calculation on a mobile platform such as a UAV.

In this study, a set of computer vision algorithms collectively known as Structure from Motion (SfM) are examined for use in adding geospatial information and reducing bandwidth required for transmission of imagery through a constrained link. A SfM pipeline takes as input a set of two-dimensional images and outputs a three-dimensional meshed and textured model, known as the reconstruction, based on information deduced from features of the input set of images. Ideally, the reconstruction exactly matches the original scene from which the input images were created, but re-

alistically some warping and aberration occurs in the process. This study assesses the geospatial accuracy of reconstructions from input sets under various reconnaissance conditions. Potential bandwidth savings are also investigated by comparing the sizes of compressed reconstructions to the sizes of their corresponding input image sets.

A three-dimensional virtual world is utilized to reduce overall cost of the study and ensure repeatability of results. Use of a virtual world prevents the need to purchase a UAV, camera, and navigation equipment, obtain operator licenses, plan and coordinate flights, etc. The virtual world also enables precise positioning and collection of images which would be impossible in real-world tests, and the parameters of this collection are modifiable in a simple configuration file. Because the virtual world includes the original scene and can load the reconstruction as an additional model in the scene, it is possible to overlay the reconstruction on the original to view the differences between the two. Small differences, e.g. in spatial characteristics of a reconstructed building compared to the corresponding characteristics of the original building, may make a drastic difference in an intelligence scenario. A quantification of error reveals differences among input sets, which aids determination of UAV flight paths and camera settings for the best reconstruction results.

This thesis contributes an assessment of a common SfM pipeline's geoaccuracy when given slightly different sets of images from a virtual UAV orbiting around a set of buildings. It shows, using georeference points, that the height (Z-scale) error is typically the greatest error component and proposes a method to automatically correct this error. Finally, reconstructions are compressed and compared with input set sizes for the purpose of reducing satellite bandwidth required in intelligence imagery applications, and recommendations are proposed to reduce reconstruction sizes.

II. Background

2.1 Technical Overview

This thesis describes the Structure from Motion (SfM) computer vision process. An SfM pipeline takes 2D images of an object or scene as input and outputs a 3D model based on the images. The 3D model ideally represents the correct dimensions and proportions of the real-world object in the 2D images. This chapter describes the basic foundations and stages of a common SfM pipeline and discusses some useful applications of SfM in areas such as geology, architecture, and movie production.

Human Vision Versus Computer Vision.

The process of capturing imagery, whether by a biological eye or a camera, involves conversion of a 3D scene into a 2D image, thus losing 3D spatial information in the process. Human vision accounts for some of this 3D loss by interpreting lighting conditions, such as shadows, and distance conditions, such as the size of a known object on a 2D plane. For instance, a shadow in a picture may give the human viewer a hint about the object's position, and a recognizable object may give the viewer an idea about the image's overall scale or about the 3D positioning of the object in that scene. Figure 1 shows an easily-recognizable scene: a closeup picture of a coin.

Computers have no such intuition; to a computer, an image is a long string of numbers which may be thought of as a two-dimensional matrix. Each position of the matrix represents a pixel–the smallest unit of visual information in the picture. Computer vision algorithms must perform operations on the pixels of an image to deduce information without the benefit of *understanding* its content.

Figure 1 helps to demonstrate this difficulty. To a human observer, it is obvious that the image is of a coin with a strong leftward light source casting a shadow

to the right. The edges are clearly defined, and the strong light does not impact understandability of the image. However, the lighting and shadows may wreak havoc on computer vision algorithms, especially those which rely on edge detection. The edges toward the left of the image are strongly white, while the edges toward the right are dark; a computer vision algorithm may incorrectly interpret the light (or dark) edges as non-edges, thus losing information that would otherwise be obvious to humans. This is one of many potential issues in automated analysis of images, and while much work has been completed in the computer vision field to alleviate these issues, the simplest way to work through them is often to focus on capturing images without such issues before attempting automated analysis.

Camera Properties.

A digital imaging device consists of several electronic components which work together to detect and record bundles of light as usable data. In general, the components include a lens, aperture, shutter, sensor chip, analog-to-digital converter, and some post-processing hardware. The lens serves to gather and angle light into the camera body, and the aperture controls how much light reaches the sensor chip. The shutter aids in gathering a precise amount of light in a single image capture by opening and closing at a set interval, thereby letting only a certain amount of light hit the sensor chip during image capture. The analog-to-digital converter then creates binary data based on sensor's detection of photons, which is then processed and utilized in any image-based computer application.

Variations on these components change the results of the final image output, and some tradeoffs must be made in real-world applications. For instance, the camera lens brings more light into the sensor than a lens-less alternative (known as a pinhole camera), but the lens itself introduces distortion which may interfere with computer

Figure 1. A quarter with strong light source from the left; edges on the left are bright white, but edges on the right are dark, causing potential confusion to automated analysis programs

vision-based applications which require a high degree of precision. Various techniques exist to reduce and correct this inherent distortion, from better manufacturing processes to algorithmic post-processing, but no solution can completely remove the distortion introduced by a lens.

The distance between the lens and the sensor chip also greatly affects the resulting image. This distance, known as the focal length, is a key parameter on camera lenses and determines the field of view and magnification of images. A longer focal length results in a narrower field of view and higher magnification, while a shorter focal length results in a wider field of view and lower magnification. Lenses are created for different situations and focal lengths, from telephoto lenses with long focal lengths to fisheye lenses with very short focal lengths. In computer vision applications, different focal lengths may produce different results even when the rest of the system is the same.

In computer vision, the terms *intrinsic properties* and *extrinsic properties* are used to refer to the camera's hardware and positioning in the world, respectively. Intrinsic properties include aspects such as focal length and corrections for lens and sensor distortion, while extrinsic properties relate to the camera's location (often specified in latitude/longitude/altitude coordinates) and orientation (often determined by accelerometer). These properties are associated with every image and are utilized in undistortion and feature-matching algorithms.

Due to imperfections and defects in the manufacturing process, commodity camera sensors may produce imagery with slight distortions from reality; this is most noticeable in cameras with fisheye lenses, which capture a greater field of view but tend to produce images with curved lines which would otherwise be straight. Some distortions are also caused by imprecise placement of the light sensor in the camera body or misalignment of the lens. Because some computer vision processes require

extremely high precision, these distortions can cause certain algorithms to fail or yield poor results. To counter the distortions, it is possible to perform a camera calibration process which results in a 4x4 matrix known as the calibration or intrinsics matrix. The matrix serves to reposition pixels from an uncorrected image to their correct position in a corrected image. The camera intrinsics matrix is described in detail in Kaehler and Bradski's *Learning OpenCV 3* book, and the OpenCV 3 library provides functions to calibrate a camera by holding up a known object (such as a chessboard) in varying positions in front of the camera [3].

In computer vision systems with multiple (typically three or more) views, such as in SfM, it is possible to obtain a camera calibration matrix without the chessboard-style routine. This process, known as self-calibration or auto-calibration, takes uncalibrated images and produces a calibration matrix for each image [4].

Structure from Motion Pipeline.

An SfM pipeline consists of several stages from initial collection of images to a simplified, textured mesh ready for use. This section provides an overview of each stage of a typical SfM pipeline.

Image Collection and Preprocessing.

SfM processes seek to determine 3D information when given only 2D images as input. The images must be collected carefully and with conditions as ideal as possible. A good input image contains plenty of bright light but few or no shadows. Glare from shiny surfaces must also be avoided, because the glare may appear differently from various perspectives and disrupt the feature detection and matching stages. Partially cloudy scenes are ideal for outdoor reconstruction, but the area must still be bright enough to capture images without grainy noise from the camera sensor. Figure 2

shows a set of input images taken from an indoor environment with little glare.

Sparse Reconstruction.

With a set of undistorted images, various algorithms and techniques from the computer vision field can 1) detect important points, known as features, in the images; 2) determine similarity of features and cross-correlate their locations across images; and 3) determine the 3D location of each feature point. This is known as sparse reconstruction, which is the first step of a SfM pipeline.

Feature detection algorithms have an extensive history within computer vision, beginning with edge detection, progressing through corner detection, and eventually resulting in modern algorithms to detect important points in an image. A widely-used feature detector is David Lowe's Scale-Invariant Feature Transform (SIFT), a patented algorithm which seeks to identify features without respect to an image's scale, rotation, or lighting conditions. The algorithm is freely available for research and academic use, and the pipeline used in this work utilizes SIFT in its sparse reconstruction step.

The SIFT algorithm is useful in SfM and other feature matching algorithms due to its assignment of a unique keypoint descriptor to each feature it detects. The keypoint descriptor is a set of vectors which may be used to compare with other keypoint descriptors: the more similar the magnitudes and directions of two descriptors' vectors, the more likely the features are to correspond to the same point in an image.

The sparse reconstruction process begins by detecting features in a set of input images, often using SIFT. Figure 3 shows an input image with feature points detected by SIFT. When all features are detected, they are then matched among the images. Figure 4 shows the matching detected between two images. This process involves $\binom{n}{2}$ matching operations, where n denotes the number of input images, because each image

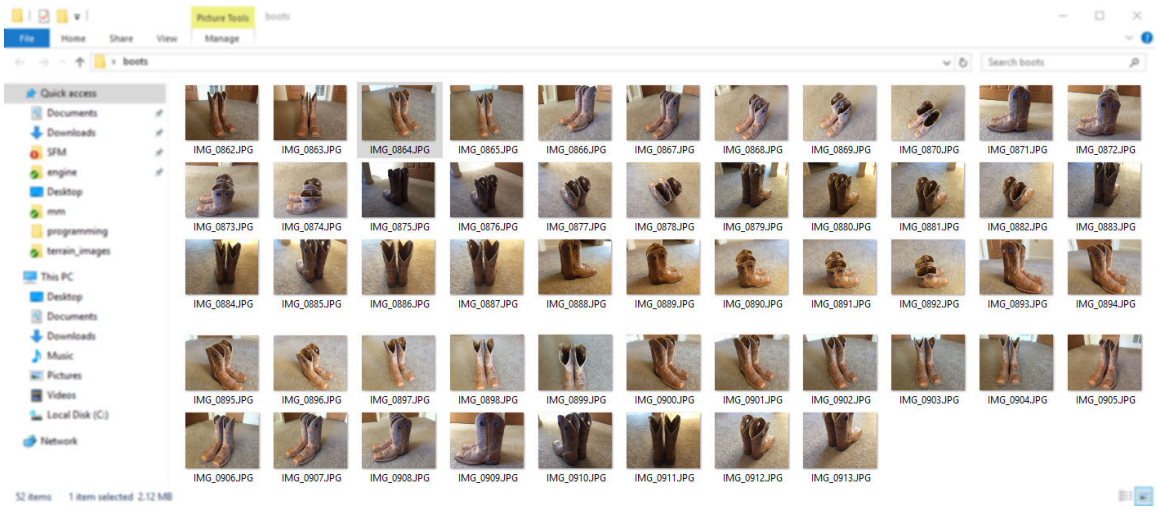

Figure 2. A set of images to be utilized in a SfM pipeline

must be compared with every other image. There are therefore $O(n^2)$ comparisons to make in this step, which necessitates careful selection of a number of input images: too few images will result in very little information for future SfM stages to work with, but too many images will result in unacceptably high processing requirements.

When feature matches are determined, epipolar geometry, a geometric system useful for relating two cameras facing a similar scene, is utilized to triangulate the relative locations of the cameras and the 3D locations of feature points in space. The feature points' locations are subject to reprojection error, a phenomenon that occurs when two camera angles do not agree on the precise positioning of a feature. By incorporating more camera angles for a specific feature, it is possible to statistically determine the most correct location of the feature in 3D space. A process known as bundle adjustment minimizes the reprojection error of all points simultaneously. Figure 5 shows the result of the sparse reconstruction step.

Dense Reconstruction.

Sparse reconstruction does not give a full appreciation of the original scene, so extra work, known as dense reconstruction must be accomplished to fill in the gaps. A

Figure 3. An input image with SIFT features shown

Figure 4. Two input images with feature matches shown

Figure 5. The output of the sparse reconstruction step, performed in the VisualSFM program

common method of dense reconstruction involves the Patch-based Multi-View Stereo (PMVS) framework [5]. PMVS takes as input a sparse point cloud and set of registered (i.e., correctly positioned and oriented in 3D space) camera views and outputs a *patch* of 3D points from each camera perspective by determining the depth of each pixel in the image and rejecting outliers. These patches are combined into a dense point cloud, which is a superset of the sparse cloud from the previous step [6].

This dense point cloud then serves as input to further algorithms for refinement. A zoomed-out view of a dense point cloud may give the appearance of a meshed model due to the quantity and density of points, but the SfM process is not entirely finished at this stage. Figures 6 and 7 show two angles of the dense point cloud resulting from dense reconstruction step.

Mesh Creation.

With the dense point cloud complete, it is possible to create a 3D model from the points. A 3D model imbues a topology upon a set of vertices, creating a watertight set of faces. In this application, the topology is a connected set of triangles, where each triangle's vertices correspond to three vertices in the dense point cloud. A 3D model is superior to a simple point cloud in this application, because it acts as a non-porous structure and can be further processed and tweaked for computer graphics applications. The creation of a 3D model from a set of points is known as mesh reconstruction, and there are several methods available to complete this step. In this study, Pierre Moulon's implementation of Michael Kazhdan's Poisson Reconstruction process is used for mesh reconstruction [28]. Figure 8 shows the result of the Poisson Reconstruction on the dense point cloud in Figures 6 and 7.

When mesh reconstruction is complete, the resulting model may have too many vertices to be readily usable in graphics rendering applications, even with powerful

Figure 6. The output of the dense reconstruction step, performed in the VisualSFM program

Figure 7. Another view of the output of the dense reconstruction step

video cards. To combat this, decimation techniques exist to reduce the number of vertices in 3D models. One such technique is Quadric Edge Collapse Decimation, which is employed in this work. Figure 9 shows the result of applying Quadric Edge Collapse Decimation to the reconstructed model; the simplified model requires less space and uses less graphical processing power to render. While the mesh in Figure 8 contains nearly 1.4 million faces, the decimated mesh in Figure 9 contains just under 50,000 faces–a reduction by a factor of 28.

Mesh Texturing.

Once a simplified, easily-renderable 3D model is created, it may be textured with pieces of the starting image set. This is accomplished by placing the images into 3D space according to where the camera was (the camera positions are derived during the SLAM process in the sparse reconstruction stage), which is a process called raster

Figure 8. A Poisson Reconstruction of the model, performed in Meshlab

Figure 9. The result of decimating the model to reduce the number of vertices

registration. With the images properly positioned in 3D space, each model face normal is compared with the normal direction of each image. In naive implementations, each model face simply uses the image with the smallest normal angle difference as its source for texturing information. However, newer algorithms add improvements to account for blur, bad lighting conditions, improperly-registered images, etc. These improvements are discussed extensively in [7]. When the 3D model is textured, the SfM process is complete, and the model may be imported into various 3D graphics applications. Figure 10 shows a final, textured model, and Figure 11 shows the source image of the model's textures. In this case, the combined space requirement of the output model and its texture file is 3.58MB, which is 3% of the original 116MB required for the pictures input into the pipeline.

Virtual Experiments.

Analysis of algorithm performance is typically subject to unwanted interference from unfortunate realities. In the case of computer vision, these unfortunate realities include camera sensor imperfections, non-ideal lighting conditions, and adverse weather. Additionally, real-world experiments often require extensive planning and expensive equipment, and the early stages of experimentation can reveal costly mistakes which require repetition of the process and purchase of new equipment.

These conditions are unavoidable for the development of real-world systems, but algorithmic analysis targets a much earlier phase of development and can afford to abstract many of the unfortunate realities away. In computer vision, this abstraction may involve the use of a 3D virtual environment rather than an experiment site in the real world. A virtual environment provides a myriad of benefits for the researcher, including very precise measuring ability, control over otherwise-uncontrollable parameters such as weather, lighting, and camera sensor conditions, and access to otherwise-

Figure 10. A final, textured model created by a SfM pipeline with some manual processing

Figure 11. The source image for texturing the model; automatically generated in Meshlab from pieces of the input images

unknowable truth data such as absolute 3D positioning of elements of the truth scene.

By using a virtual world, it is possible to define bright, shadowless lighting with perfect, particle-free visibility, and the camera sensor may be defined to be a perfect pinhole model with no lens. The perfect pinhole camera is modifiable per experiment, removing the need for costly camera equipment and allowing precise definition of the camera's intrinsic properies. Furthermore, the use of a virtual world ensures that the absolute position of specific features in the scene are known, rather than estimated, and this perfect knowledge enables noise-free quantification of error.

Virtual experiments also allow for perfect rebuilding of results from well-defined starting conditions, whereas real-world experiments are always subject to slight environmental differences which may alter results. Replayability helps researchers to reproduce and verify accuracy of results and build on previous work with more rigor and confidence.

Alignment.

A virtual world also enables *overlaying* a reconstruction on the original scene for comparison, such that a 3D model intersects with another 3D model in the world. However, the SfM process cannot infer scale and absolute world-position properties from images (though some experiments with GPS-tagged images may rectify this), so the reconstruction must be scaled and positioned to be as close to the original scene as possible [8]. In the case of outdoor reconstruction, the ground plane serves as a base for the scaling and positioning effort: by aligning the ground plane of the reconstruction with the ground plane of the original scene, all other elements (such as buildings or cars) are also scaled and positioned with respect to the ground. This will reveal any geometric or texturing flaws in the reconstruction which may have resulted from the SfM process.

Error Quantification.

In an ideal SfM implementation, the output model is perfectly indistinguishable from the original scene: each vertex or edge in the reconstruction perfectly represents a corresponding vertex or edge in the original, with no missing vertices or edges, and the textures are also perfectly matched. However, no such ideal implementation exists; the reconstructed model must always contain some difference from the original scene in the form of simplification (i.e. fewer vertices and edges) and/or deformation (i.e. inaccuracies in the reconstruction). This deviation from the original scene may be quantified to help determine the usefulness or accuracy of the SfM implementation.

There are two potential quantification methods: visual similarity and model similarity. In the visual similarity approach, an image is captured from a specific angle in the original scene, and a corresponding image is captured from the same perspective of the reconstruction. The two images are compared, and the difference between them is the quantification of error. The visual similarity approach is a difficult computer vision problem, because it requires human-like intuition to determine how similar two pictures are. The current technology for general image similarity quantification is not advanced enough for SfM analysis, so the more useful approach involves determining similarity of models. More research in visual similarity-based methods is recommended in [6].

The model similarity method is also not trivial. It is possible to compare the model of the reconstruction to the original scene, but significant differences prevent a direct geometric comparison. The reconstruction, while visually similar, has a different number of vertices and edges than the original model; this prevents a one-to-one comparison. Even if the reconstruction were decimated or interpolated to contain the same number of vertices and edges, the issue of vertex mapping remains: it is not possible to say which vertex of the reconstruction corresponds to a particular vertex

of the original.

Due to the inherent differences between the reconstruction model and the original model, a geopoint-based system may be a more manageable approach to model comparison. In this approach, a set of relevant points are chosen on the original model. When the recontruction is created, the same set of points are marked on the reconstruction. This provides two sets of points with a one-to-one mapping, and the 3D distances between the points can be determined and averaged for an overall error metric.

Error Correction.

In the case of outdoor reconstruction, images are generally captured from a downward angle, resulting in loss of information about the height of the scene [9]. The result is a slightly flattened reconstruction where all other aspects are relatively accurate. The degree of flattening depends on several factors, such as the flight path, focal length and downward angle of the camera, and geometry of the scene itself.

For this reason, it may be useful to conduct a second, error-correcting alignment step which involves scaling in the Z dimension only. Once a reconstruction is initially aligned with its original model and geopoints are chosen, it is possible to scale the reconstruction in the Z dimension in small increments, checking the error metric at each step. As the reconstruction changes scale, its geopoint markers will also change position and yield different distances to the corresponding truth points on the original model. The Z-scale which provides the smallest overall error is then accepted as the best compensation for the flattening in the Z dimension.

In a real-world scenario, it would not be possible to use this small-increment method to correct the Z-dimension error without manually measuring the positions of the geopoints in the real-world scene. If this measurement is not possible, e.g. in

an intelligence-gathering situation, it may be necessary to utilize a different method for Z-dimension correction. Assuming alignment of the reconstruction can be done sufficiently well and in real-time (by using GPS or other positioning information), the tallest point of the reconstruction could be determined. Then, the collection device (such as a UAV) could fly to the corresponding point in the real-world scene and utilize a range-finding sensor to determine the height of the point. The model could then be scaled accordingly to match the highest point of the reconstruction with the highest point of the real-world scene.

Compression and Space Comparison.

For this work, it is necessary to compare the bandwidth usage of the traditional video-streaming approach to the usage of the model-streaming approach. If the model-streaming approach offers great accuracy but requires excessive bandwidth, it may not be a useful improvement in the great intelligence-gathering situation. Because specifications for military UAVs and satellites are generally not available for public use, this work focuses primarily on the size of the reconstructed model in comparison to the size of the images from which the reconstruction was created. The use of original images offers a useful frame of reference for comparison, because a real-world video-streaming approach will typically require more bandwidth than sending still images; therefore, comparison of a reconstructed model to still images will offer a more conservative, pessimistic improvement metric than comparison to the more realistic, higher-bandwidth video stream.

To accomplish the comparison, one may use a freely-available compression utility to simulate the compression which would take place in a real-world scenario. The compressed size of the images may be compared to the compressed size of all necessary vertices, edges, textures, and texture-mapping indices of the model. Because the

model contains repetitive information in its vertex and texture-mapping coordinates, it may have a much greater compression ratio, i.e. the ratio between compressed and uncompressed size, than the set of images. Additionally, the image set may already utilize a compressed format such as Portable Network Graphics, in which case further compression will have little or no effect on the image set size.

2.2 Related work

SfM algorithms find use in a wide array of computer vision fields. In this section, a few applications of SfM are discussed, particularly those which utilize unmanned aerial vehicles (UAV) and/or virtual environments to capture or synthesize imagery. We emphasize applications and methods which may assist the goals of reduced satellite bandwidth usage and increased geospatial awareness for military intelligence purposes.

Structure from Motion Applications.

A review of UAV technology was conducted by Nex and Remondino, including a brief history of UAV usage, modern capabilities, and applications for imagery capture [10]. The review describes uses of UAVs for 3D imaging and reconstruction, including agriculture, forestry, archeology, architecture, environmental monitoring, emergency management, and traffic analysis. Difficulties with accurately reconstructing large-scale, non-flat scenes such as buildings are discussed, and additional research is suggested [10]. The advent of inexpensive UAVs with high-quality visual sensors has simplified work which was traditionally far more expensive and dangerous, such as tasks requiring flight of a manned helicopter with Light Detection and Ranging (LiDAR) equipment into remote areas. This decrease in cost may also benefit small-scale and/or swarm-based intelligence applications.

A study of UAV usage in wilderness vegetation monitoring was conducted, showing that UAVs are generally superior to satellites for this purpose due to higher resolution of imagery and no limit on revisit times. The study shows the use of physical geo-reference points to assess and improve the accuracy of post-analysis and combination of individual images. An improvement of the geopoint-based technique is shown over the typical sole reliance on GPS truth data from the sensor platform [11].

Another study captured topographic data of an outdoor area with both a traditional LiDAR setup and an experimental UAV setup with a consumer grade camera to create a SfM reconstruction. The analysis showed that the SfM approach performed as well as the more-expensive LiDAR approach. The application of measuring soil erosion is briefly discussed [12]. Other studies also assess and praise the accuracy of SfM, compared with LiDAR, for geoscience applications. Accuracies within one decimeter are reported [13][14]. The relative accuracy of SfM approaches may assist UAV-based military intelligence applications by enabling robust 3D reconstructions without the need for heavy, expensive LiDAR sensors. Additionally, SfM may benefit stealth intelligence missions due to its passive nature, whereas LiDAR requires actively sending an electromagnetic signal to the target area of interest, potentially revealing the intent and/or location of the UAV.

Koutsoudis et al. assessed the accuracy of a commercial SfM software package as compared with results from a 3D range scanner for the purpose of heritage building reconstruction. The accuracy of reconstruction was also assessed by measuring specific line segments on the physical building and comparing them with the mathematically similar measurements of the reconstruction. Strong emphasis was placed on the importance of good lighting conditions and powerful PC hardware for a useful SfM experience [15].

Specific difficulties of outdoor scene reconstruction, such as in the case of intelli-

gence gathering and movie production, were described and partially mitigated by Kim et al. The difficulties involved poor lighting conditions (too bright, causing strong shadows, or too dark) and background scenes which may interfere with accuracy of the SfM process. A background separation scheme was proposed, where the foreground and background are reconstructed in separate pipelines and recombined at the end of the process [16]. In a real-world intelligence application, one must mitigate these compounding factors; however, a virtual world may omit the sky or background details and thus avoid the issues of foreground and background separation. Additionally, a virtual world offers greater lighting flexibility for experimentation, such as toggling of shadows, fog, and/or glare.

The use of synthetic environments, such as 3D virtual worlds, to measure SfM accuracy has been demonstrated in other work. Nilosek, Walvoord, and Salvaggio utilized synthetic imagery and its associated truth data to measure the accuracy of a 3D reconstruction from imagery taken at nadir angle. They found that GPS and orientation data are sensitive to noise inherent in the SfM process itself and attributed this sensitivity to errors in the image-to-image correspondence step. Inaccuracies of reconstructed building height were also discussed and attributed to the nadir angle of capture. A process of georegistration is discussed in which the reconstructed points are transformed from an arbitrary coordinate frame to an earth-centric frame [9]. These findings may benefit satellite-based intelligence activities, where the sensor is often nadir to its target; however, for applications involving small, agile UAVs, it is possible to capture images from a wider variety of angles and distances. Further research may reveal flight paths which can counteract or overcome the poor image-to-image correspondence found in this study.

A demonstration and assessment of a SfM-based visual navigation system was conducted by Alix. The experimental setup utilized the simultaneous localization

and mapping (SLAM) technique to determine the image sensor's position, and the SLAM technique's estimation of the sensor's position was compared with its actual position in the virtual environment. Similar to [9], a method to transform from arbitrary to world space was proposed, and synthetic imagery was utilized to carry out the experiment [17].

An experiment conducted by Ekholm showed the ideal capture angle of a UAV-based sensor for SfM accuracy in a particular scenario. The experiment involved a virtual scene of an untextured cityscape over which a virtual UAV flew at various angles and flight patterns. An image capture angle of around 45 degrees was found to be ideal for reconstruction accuracy in the given scenario. Accuracy was determined by comparison of truth to reconstruction model vertices in the world rather than by depthmap or specific geopoint comparison. The experiment was created and run in the Blender 3D modeling application [18]. This experiment served as a starting point for finding a useful flight path in this paper's experiment, but Ekholm's flight path styles were designed for larger UAVs flying in relatively straight lines, as opposed to the finer, more circular orbits employed in this study, and thus a constant 45-degree angle was not ideal in this case.

SfM and other computer vision techniques also find use in other novel defense applications. Colson demonstrated a SfM-based approach for the application of automated aerial refueling, which is a necessary technique to refuel UAVs due to latency issues between the UAV and UAV pilot. He utilized a scaled-down fighter jet model, suspended on cables and approaching stereo cameras, within a high-fidelity motion capture chamber to simulate a fighter aircraft approaching a fuel tanker aircraft. He estimated the accuracy of SfM based on the truth data from the chamber and assessed that the SfM setup was *not* sufficiently accurate for this purpose [19]. Other vision-based techniques, such as shelled point cloud registration, have proven successful for

automated aerial refueling in virtual environments and are now under assessment in real-world experiments [20]. With sufficient refinement, these techniques may benefit intelligence applications by enabling 3D model-based recognition of objects of interest after reconstruction.

While time and space efficiency are not specifically studied in this work, they are critical for the intended application of onboard reconstruction. An embedded processor on a UAV is tightly constrained in processing ability and electrical power availability, and it relies on efficient algorithms to perform any complex task satisfactorily. A voxel-hashing technique is proposed in [21], which would drastically reduce graphical processing unit (GPU) memory required to represent 3D point clouds, and [22] builds upon this work by processing small batches of imagery in a stream rather than individually, further reducing the amount of GPU memory required for reconstruction.

Structure from Motion Tools.

A typical SfM workflow consists of sparse reconstruction, dense reconstruction, mesh creation, and texturing. A detailed explanation of each stage is not summarized here, but a selection of software capable of performing these processes is discussed.

An increasing number of free, open-source SfM tools are available for research. Each tool has its own abilities and nominal application sets, and most tool developers focus on performing a specific part of the reconstruction well rather than spreading effort over the entire process.

The graphical user interface-based VisualSFM was developed by C. Wu and utilizes the Clustering Views for Multi-view Stereo (CMVS) and Patch-based Multi-view Stereo (PMVS) tools to construct a dense point cloud from a set of pictures [23][24][5]. A command-line based alternative to VisualSFM is Noah Snavely's Bundler, which

performs a sparse reconstruction suitable for input to CMVS and PMVS [25][26].

A newer framework, the Multi-view Environment (MVE), is capable of performing all steps of the reconstruction except texturing, for which the Multi-view Stereo Texturing (MVS-Texturing) tool is built [27][7]. An alternate mesh creation process may be substituted by utilizing Pierre Moulon's PoissonRecon tool, which offers empirically better results for different scene types [28].

The model visualization program Meshlab can perform mesh creation (via the aforementioned PoissonRecon tool) and texturing as well as mesh simplification and manual trimming of undesired points and mesh planes [29].

Other software programs are also under active development or available for experimentation on niche SfM situations. OpenMVG is designed to be a simple pipeline for SfM and is intended for small-scale reconstruction [30][31]. The Colmap program utilizes Nvidia's Compute Unified Device Architecture (CUDA) framework to enable GPU-accelerated dense cloud and surface reconstruction [32][33] but may need further development to handle a wider array of use cases efficiently. The OpenCV framework may be compiled with a third-party module to construct a sparse 3D cloud and perform SLAM, but it is not developed and maintained as heavily as the other software discussed [34]. Theia is a newer program capable of sparse reconstruction, and its primary goals are "usability, extendibility, and scalability." It states adherence to rigorous, modern coding standards and includes unit tests to ensure correctness of its algorithms [31].

III. Methodology

3.1 Methodology

To examine the feasibility of SfM-augmented intelligence collection with real-world devices would be expensive, time-consuming, and error-prone, especially for such an early assessment. For this reason, a virtual world is utilized, from which a virtual UAV captures imagery from a sample scene of interest under varying conditions. To conduct a real-world test, just a few requirements are UAV equipment with high-precision navigation and camera apparatus, a license to operate the UAV, consistent and ideal weather conditions throughout the study, and a relevant, unchanging scene of interest. In real-world tests, errors in navigation measurements and aberrations in the camera lens and sensor introduce noise into the SfM process, and the result of SfM with these undesired inputs contains a mix of errors from sensors and from the SfM process itself. These errors are difficult to distinguish from one another. Thus, a virtual experiment was favored over a real-world experiment, though a real-world experiment may be conducted in the future if merited by the results of the virtual experiment.

Virtual World.

A virtual representation of a relevant scene was created using the AfterBurner engine [35]. The engine builds upon the OpenGL graphics rendering library by including resource management, input/output support for software and hardware devices, and hierarchical transformations in the virtual world.

The scene includes a set of adjacent buildings representing a cityscape, clustered together on a block. The particular structures in the scene vary in height, layout, and texture, while still maintaining simple box-like structure; this enables easier visual

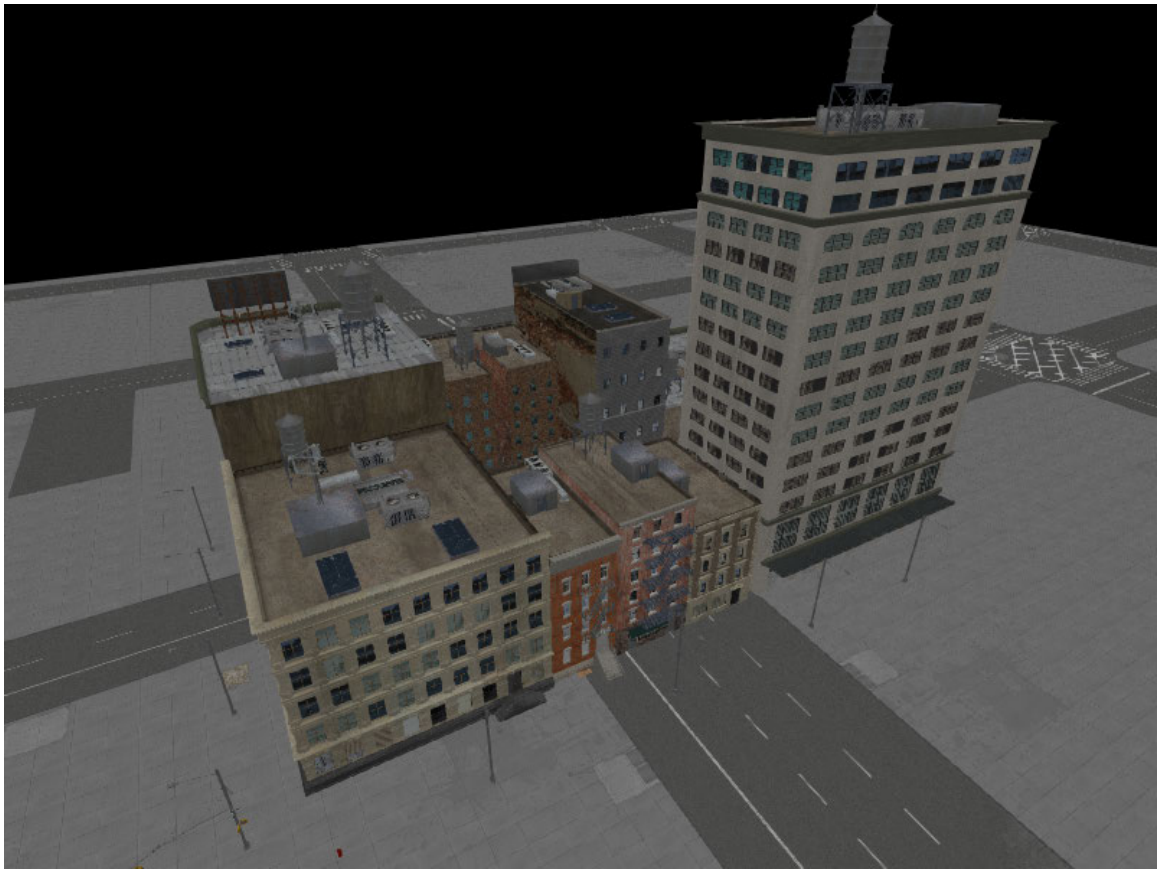

Figure 12. Original scene, rendered in AfterBurner

inspection for discrepancies between the reconstruction and the original structure. Figure 12 shows a visualization of the scene. The placement of buildings in the middle of a street intersection is an artifact of modifying the original model for use in the experiment; only cosmetic appearance is affected. Within the virtual world, the height of the building is set to 65.858 meters, and the length and width of the block are both 58.528 meters.

The scene also includes a small unmanned aerial vehicle with an attached camera. The camera is capable of capturing images within the world for later processing through a reconstruction pipeline, and its intrinsic (such as field of view) and extrinsic (such as position and orientation) properties can be modified dynamically to produce different types of imagery.

Structure from Motion Pipeline.

A set of open-source programs were compiled and combined to create an automated structure from motion pipeline which takes as input a set of images and produces as output a three-dimensional, textured mesh which can be loaded into the virtual world and overlaid on the original scene for comparison. Figure 17 shows an example of a reconstruction overlaid on the original scene.

Three tools were utilized: 1) TU Darmstadt's Multi-View Environment (MVE), 2) Michael Kazhdan et al.'s Screened Poisson Surface Reconstruction tool, and 3) TU Darmstadt's MVS-Texturing tool[27][28][7]. The process involves several discrete steps; each step, and the software used to complete it, is listed in Table 1.

The default MVE settings were used for sparse and dense reconstruction. For the mesh/surface generation step, a point weight of 0 and depth of 10 were given to the PoissonRecon tool in order to disable surface screening, resulting in smoother surfaces. The screened variant of Poisson reconstruction was developed in response to

Table 1. SfM steps and software utilized for each step

SfM Step	Software
Imagery Collection	AfterBurner engine
Sparse Reconstruction	MVE
Dense Reconstruction	MVE
Mesh/Surface Generation	PoissonRecon
Mesh/Surface Texturing	MVS-Texturing

observations that Poisson reconstruction tends to oversmooth data[36][37]. However, because the scene in this work consisted primarily of buildings with flat planes, the unscreened setting produced better results than the screened alternative, which would produce jagged surfaces more useful in reconstructing, for example, sculptures with fine details. The default MVS-texturing settings were used for the texturing step.

UAV Flight Paths.

Each experiment consisted of a UAV with a virtual camera which took pictures at predefined positions along a flight path which varied by experiment. The positions were determined based on the number of pictures to be taken and the path of the UAV. The UAV paths were determined by potential real-world scenarios and previous work by Ekholm, who experimentally determined an optimal downward camera angle of around 45 degrees in a similar scenario[18].

The main path consisted of three ascending, spiraled orbits around the set of buildings with a particular radial distance ρ, initial height z_0, and height between orbits z_b. Figure 13 shows the flight path variables.

Each flight path was specified in the virtual world as a parametric function $f(t)$, $0 \leq t < 1$. With this construction, any value of t between 0 and 1 will correspond to a three-dimensional point (x, y, z) in the virtual world:

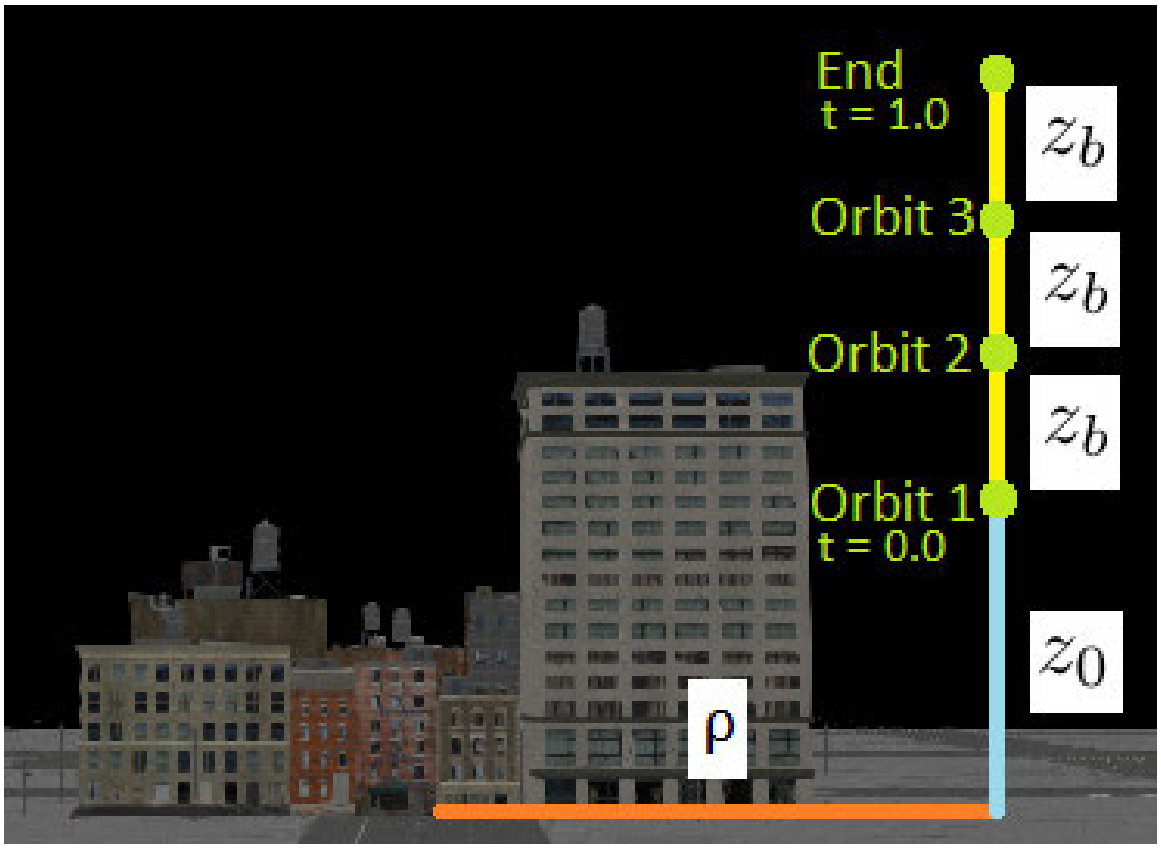

Figure 13. Flight path variables, front view

$$f(t) = (x, y, z) \tag{1}$$

The x and y positions are defined by first determining the current azimuth ϕ in radians:

$$\phi = 2\pi((t \bmod \frac{1}{3}) * 3) \tag{2}$$

The modulus serves to segment t into three orbits, ensuring each value of ϕ will be reached three times. The value of ϕ is then used to determine the x and y coordinates:

$$x = \rho \sin \phi \tag{3}$$

$$y = \rho \cos \phi \tag{4}$$

The value of t also determines the z value as follows:

$$z = z_0 + 3t * z_b \tag{5}$$

Each flight captured 90 evenly-spaced images, which offered a subjectively favorable tradeoff between reconstruction fidelity and computation time required. The camera viewpoint was always fixed on the center of the scene (as opposed to the center of the tall building, on which the error metric was evaluated; see Section 3.1 below). This approach led to increased distortion on the top of the large building when viewed from the front but enabled less distortion in the overall scene, making the manual alignment process less error-prone.

An example of the flight path, along with camera frusta depicting each of the 90 positions and orientations of images captured, is shown in Figure 14.

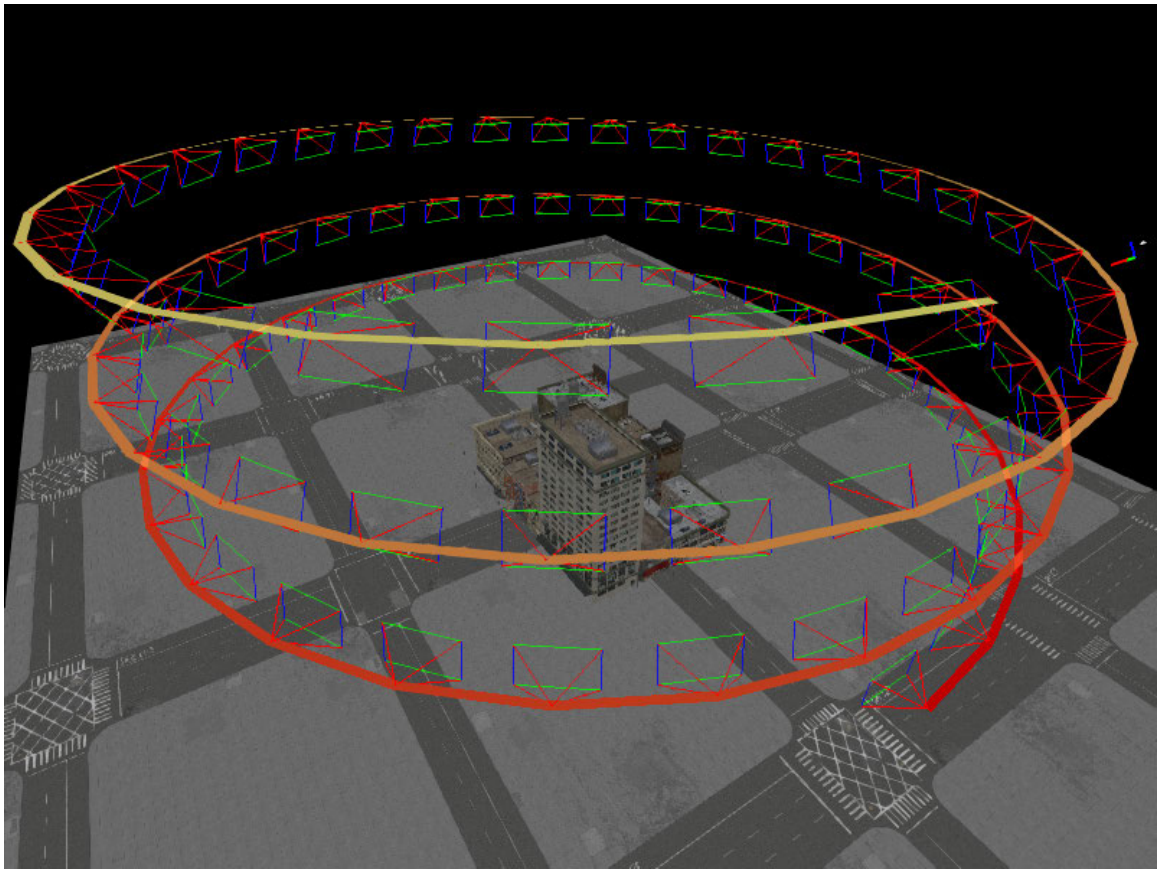

Figure 14. Spiral flight path example with 90 images

Virtual Camera Intrinsic Parameters.

Utilizing a virtual world for imagery collection enables simple camera creation with user-defined properties. Furthermore, the parameters of the camera are guaranteed, provided the code is correct, to be free of manufacturing defects seen in real-world cameras. It also becomes unnecessary to utilize a camera lens as required in the real world, because a virtual sensor can capture 100% of the light that reaches it without any loss by an analog image sensor. This enables creation of a perfect pinhole camera and removes the possibility of lens distortion.

This leaves two main parameters still configurable: image dimensions (width and height) and field of view. In the experiments, image dimensions were held constant at width 1920 pixels and height 1080 pixels, and the horizontal field of view varied based on each dataset.

Alignment.

Prior to evaluation, it was necessary to align the ground plane of the truth data with the ground plane of each reconstruction, because the SfM process cannot compute an absolute scale, translation and rotation without some other information, e.g. GPS or inertial measurements. Even with this information available, the reconstruction may not benefit from it due to noise inherent in the SfM process[9]. Therefore, the alignment was performed manually by ensuring the road planes and smaller building sides, especially at the bases, were aligned as closely as possible. This resulted in a scene where the truth and reconstruction models were overlaid, making distortions and inaccuracies readily apparent. Figure 15 shows an example original scene. Figure 16 shows the reconstruction of this scene, from the same perspective, after alignment with the original. Figure 17 shows the reconstruction overlaid on the original, revealing significant warping in the Z dimension.

To counteract the warping effect, a second scale was applied in the Z dimension only. This Z-scale was optimized for each dataset by iteratively attempting scale factors in increments of .01, starting at 1.00, until the minimum XYZ error was achieved. Figure 18 shows the reconstruction overlaid on the original with the optimal Z-scale factor applied.

Evaluation Criteria.

Two evaluation criteria were chosen and assessed: spatial accuracy and model storage size. For each criterion, a truth value was compared to the reconstruction value. The criteria of time and power were not studied explicitly due to the constantly-improving nature of embedded platforms with graphical processing units; any results would be relevant only until a newer, faster embedded platform was released. This study instead focuses on performance of the algorithms and software under different conditions.

Spatial accuracy was assessed by using a virtual version of georeference points utilized in other work[11][15]. The georeference points chosen for this set consisted of the lower-left window corners of the tallest building in the set; see Figure 19 for exact positioning of georeference point placement in this scene. The process of identifying corners was manually performed for the truth set and for each reconstruction, resulting in a one-to-one mapping of 84 XYZ truth points to reconstruction points. With this information, a sum-of-squared-differences approaches was utilized to find an error metric, where a lower number represents a better reconstruction in terms of spatial accuracy. Two spatial error metrics were utilized: one for Z error only and one for XYZ error. For a set T of truth points and R of reconstruction points, each of length n and containing x, y, and z components, the Z-only error is the arithmetic mean of absolute Z distances between corresponding points:

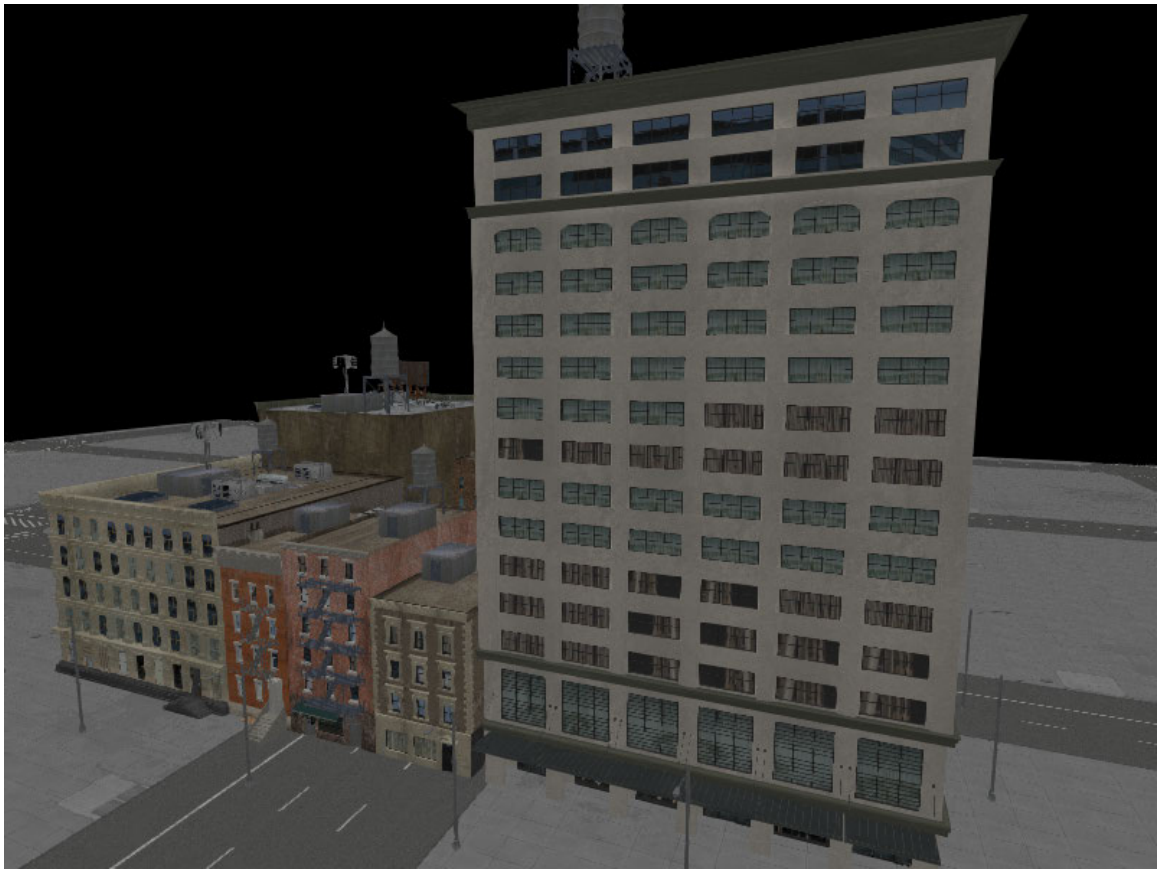

Figure 15. Original scene

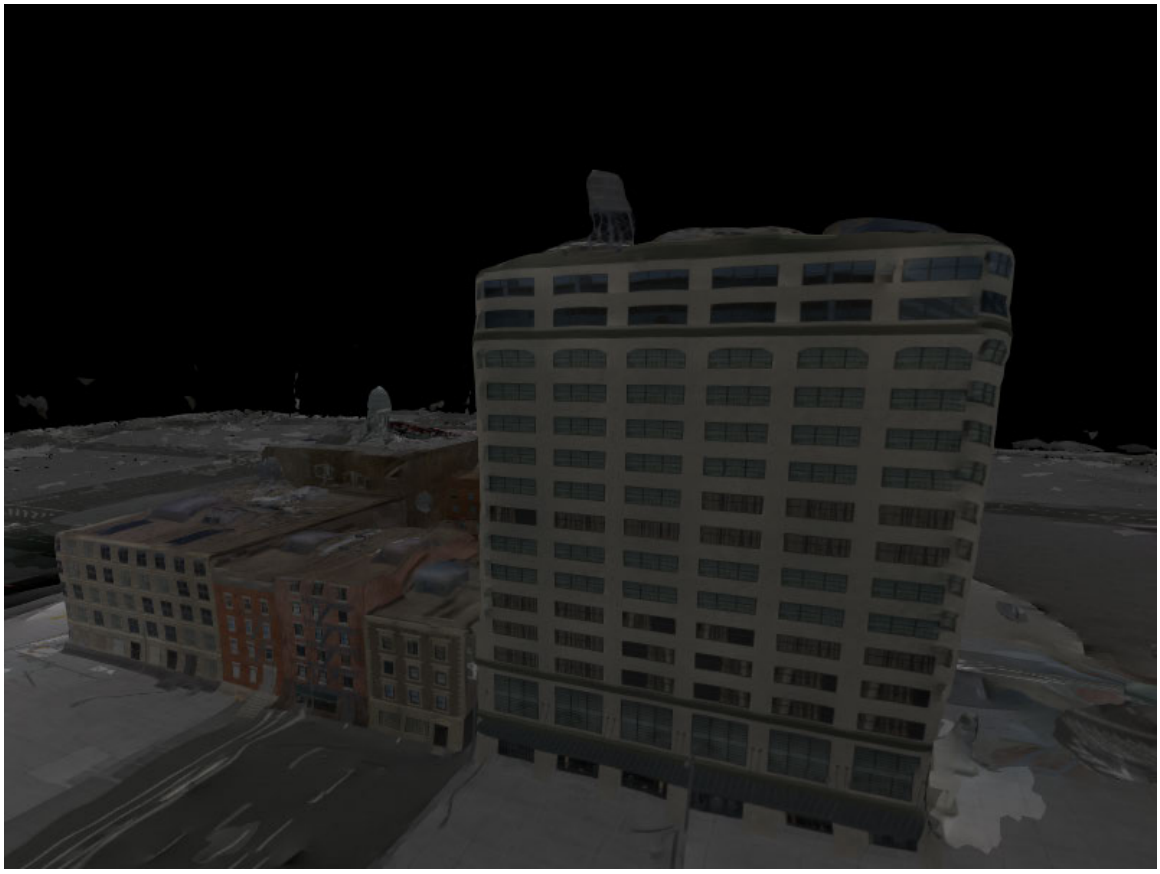

Figure 16. Reconstruction from same perspective as Figure 15, after manual alignment

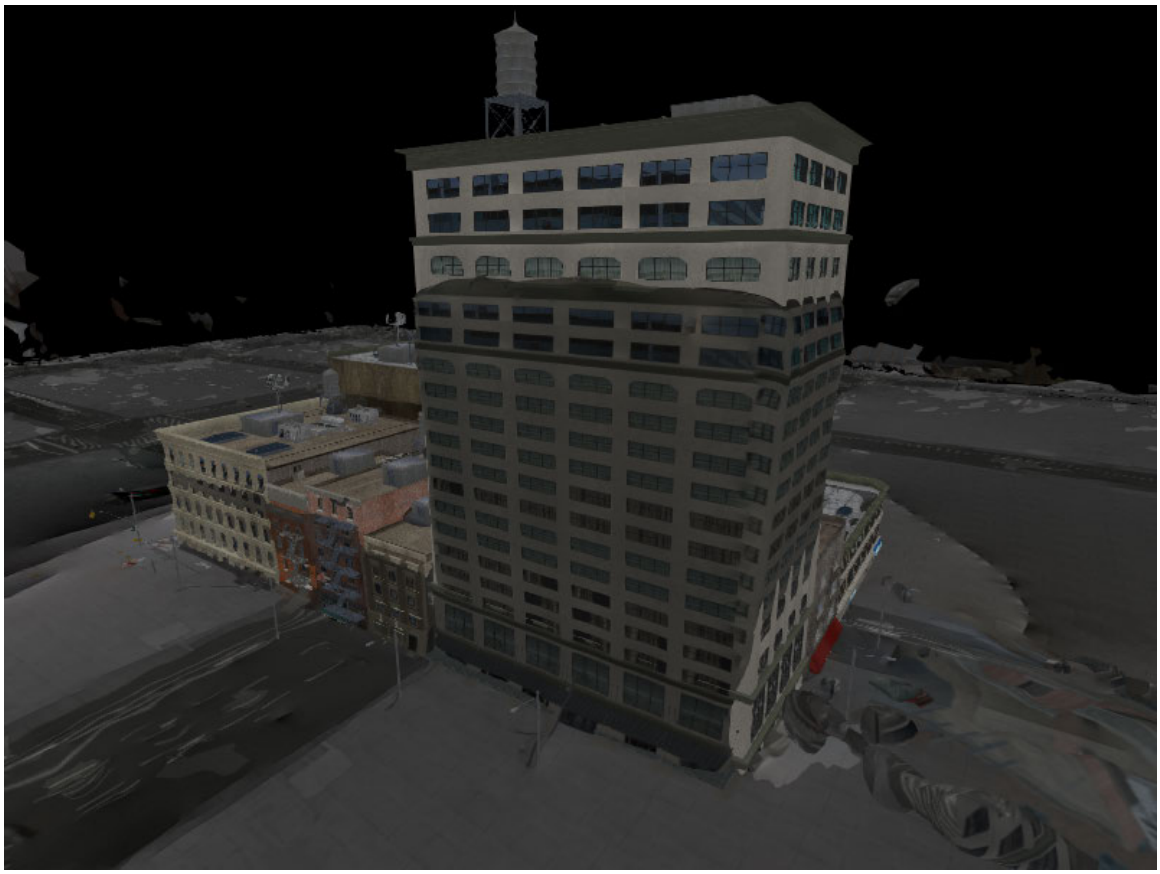

Figure 17. Reconstruction overlaid on original, showing significant warping in Z dimension

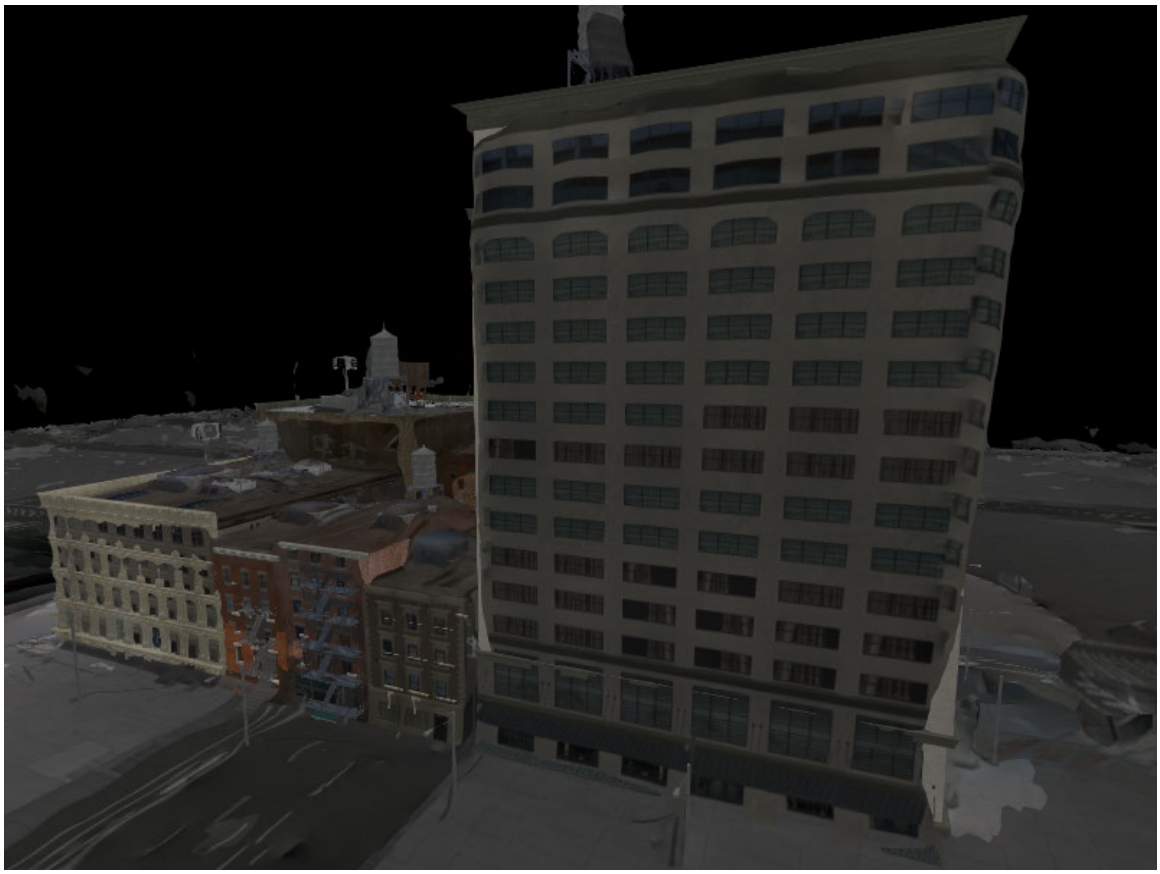

Figure 18. Reconstruction overlaid on original, with optimal Z-scale applied; compare with the inferior accuracy of Figure 17, where the optimal z-scale is not applied.

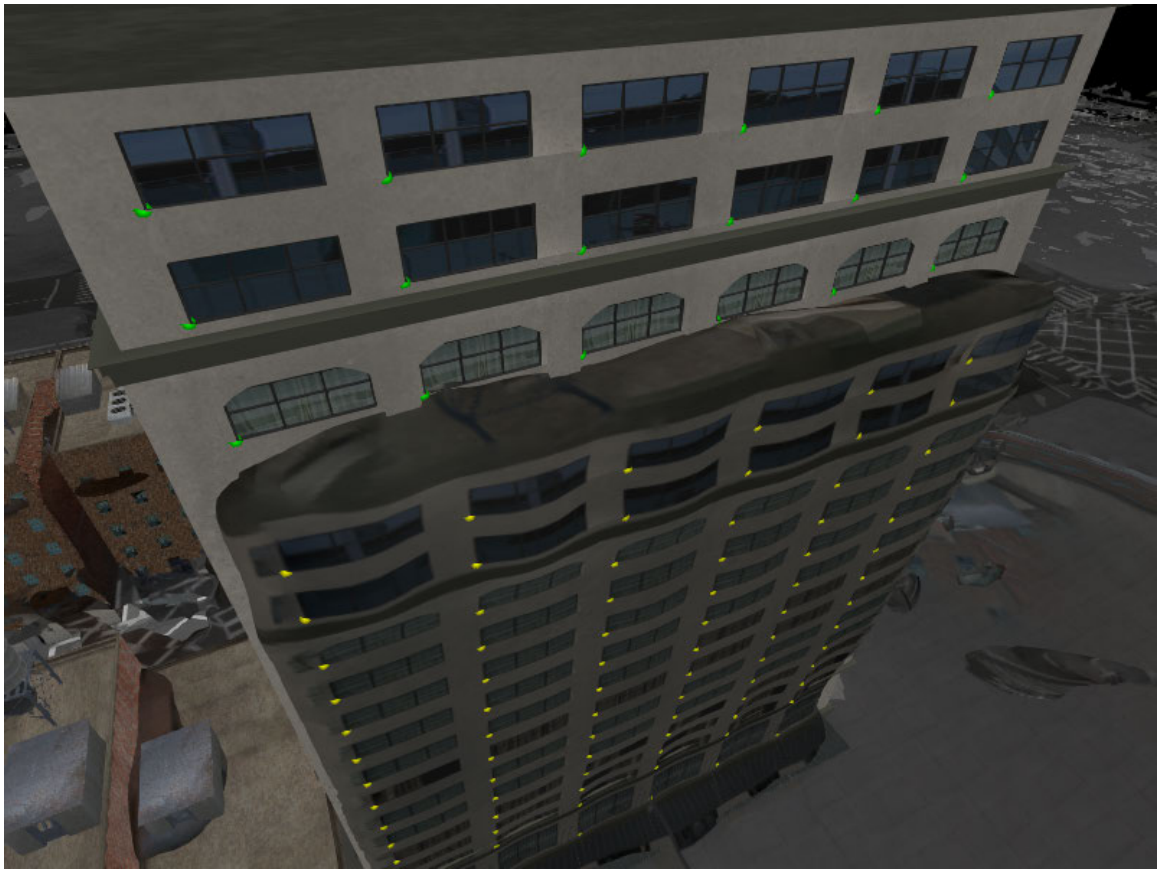

Figure 19. Visualization of geopoints; green dots represent truth data, and yellow dots represent reconstruction data. The spatial difference between each corresponding pair quantifies the distortion of the reconstruction.

$$\frac{\sum |R_z - T_z|}{n} \tag{6}$$

The XYZ error is the arithmetic mean of three-dimensional distances between corresponding points:

$$\frac{\sum \sqrt{(R_x - T_x)^2 + (R_y - T_y)^2 + (R_z - T_z)^2}}{n} \tag{7}$$

For this study, $n = 84$, which corresponds to the number of windows on the large building. The ground floor of windows was not included due to inconsistencies in the SfM process; in some cases, particularly those with higher-altitude flight paths, the ground floor was not reconstructed clearly enough to distinguish windows from the ground.

Storage size was assessed by comparing the total reconstruction size, including model and texture images, to the total size of the original images used in the SfM process. This comparison allows assessment of potential bandwidth savings via compression for the purpose of reconnaissance, i.e. whether transmitting the reconstruction requires more or less bandwidth than transmitting the source images. Because the output of the SfM pipeline includes a model file in the uncompressed Wavefront OBJ format, Igor Pavlov's 7-Zip utility was utilized to simulate compression prior to transmission on a bandwidth-constrained link. The default settings of 7-Zip were applied; these settings are listed in Table 2.

Table 2. 7-Zip settings utilized in compression study

Setting Name	Value
Archive format	7z
Compression level	Normal
Compression method	LZMA2
Dictionary size	16 MB
Word size	32
Solid Block size	2 GB

IV. Results and Discussion

4.1 Results

Hardware/Software Environment.

The experiments were conducted on a Thinkpad P50 laptop with an Intel Xeon E3-1505M v5 processor clocked at 2.80 GHz, which provided 8 threads to the operating system, and 16 GB RAM. The MutliView Environment (MVE) does not support GPU processing at the time of writing, so all SfM reconstruction was performed on the CPU. MVE was compiled and run within Ubuntu 16.04 with the default Linux kernel.

Though the time required to process each dataset was not rigorously measured, each set required around 15-25 minutes to complete the entire SfM process. Sets which took longer generally produced more subjectively pleasing results with fewer artifacts and more accurate texturing, though the geoaccuracy was not necessarily better.

Datasets.

Several datasets were created to quantify SfM process accuracy under varying conditions. The dataset parameters are listed in Table 3. A graphical depiction of the parameters is shown in Figure 13. The final, optimized error values for each dataset are summarized in Table 4. The uncorrected mean XYZ error is given. Next, the optimal Z-scale, determined by incremental search, is shown. This optimal Z-scale is applied, and the new XYZ error is listed. Finally, the percentage improvement from application of the Z-scale is listed.

The datasets mainly varied based on field of view, ground radius, and starting height. For datasets with greater ground radius ρ, the distance between orbits z_b was

increased to compensate for the overall increased distance from the scene; without this change, the images from each ring of the spiral would not have differed much and would have failed to provide additional features to assist the SfM process. For datasets with greater ground radius, the field of view was decreased to compensate for the increased distance. This resulted in a narrower view of the scene and enabled capture of more fine textural details rather than a wider view of mostly empty space.

Dataset 7 produced both the best improvement from Z-scaling and the best overall corrected results; it also had the longest ground radius and narrowest field of view settings. Figure 20 shows an overview of the scene prior to reconstruction, and Figure 21 shows the initial reconstruction. Figures 22 and 23 show drift from truth position to reconstruction position of each geopoint for the uncorrected and corrected reconstruction, respectively, from the perspective of building front. Figure 24 shows the error values at various levels of Z-scaling, from 1.00 to 1.70 in increments of .01. Finally, Figure 25 shows the reconstruction with its Z-scale set to minimize XYZ error, and Figure 26 gives another perspective of the final result. The application of a corrective Z-scale decreased the mean XYZ error from 7.416 meters to 1.347 meters, or by 81.8%.

Dataset 7 was based on Dataset 4, which used similar parameters but a shorter ground radius and resulted in a slightly worse final XYZ error of 1.960. However,

Table 3. Dataset parameters

Dataset #	H-FOV (deg)	ρ (m)	z_0 (m)	z_b (m)
1	80	90	73	20
2	80	90	33	20
3	100	90	33	20
4	65	140	33	35
5	65	140	133	40
6	65	140	3	35
7	40	250	33	60

Table 4. Dataset geoaccuracy results

#	Old XYZ Error (m)	Best Z-Scale	New XYZ Error (m)	Improvement
1	10.488	1.47	4.464	57.4%
2	6.866	1.20	4.625	32.6%
3	8.616	1.33	4.071	52.8%
4	5.441	1.22	1.960	64.0%
5	11.646	1.62	2.581	77.8%
6	7.967	1.34	2.078	73.9%
7	7.416	1.33	1.347	81.8%

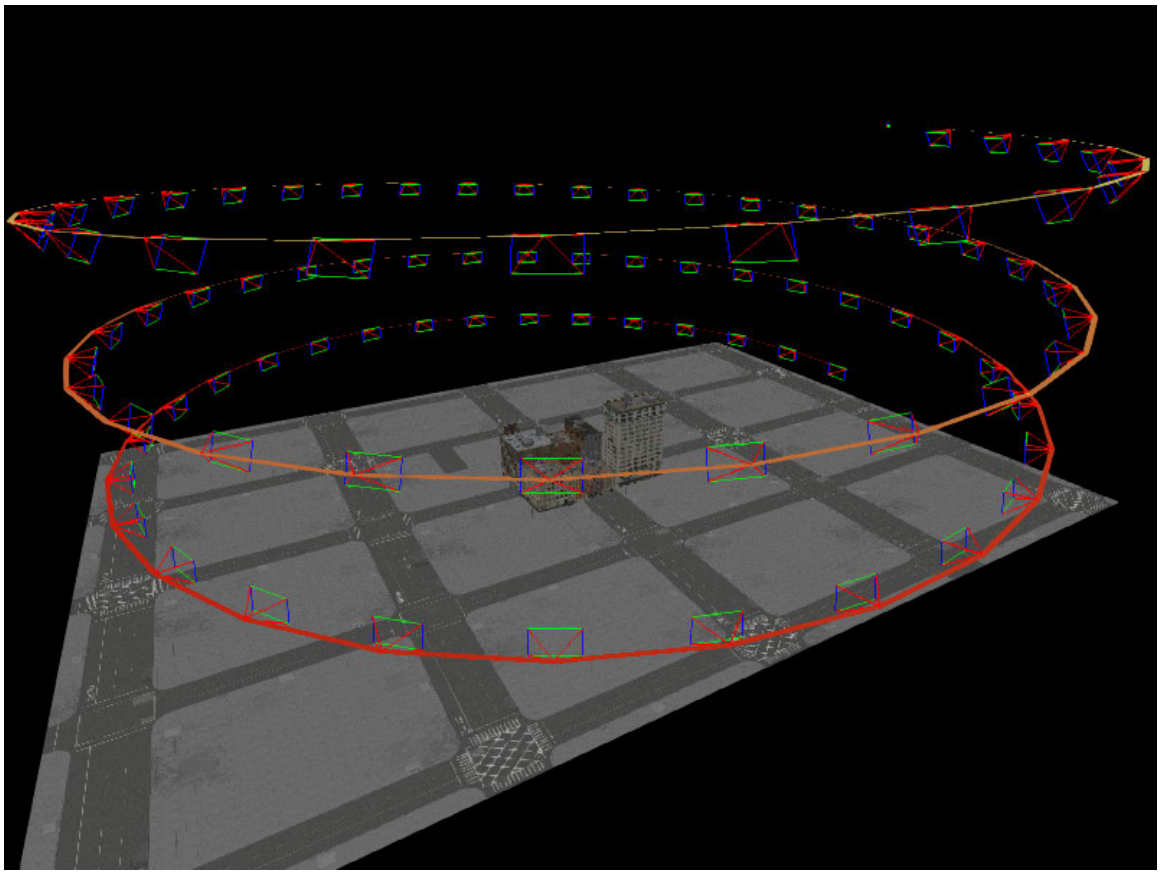

Figure 20. Overview of Dataset 7

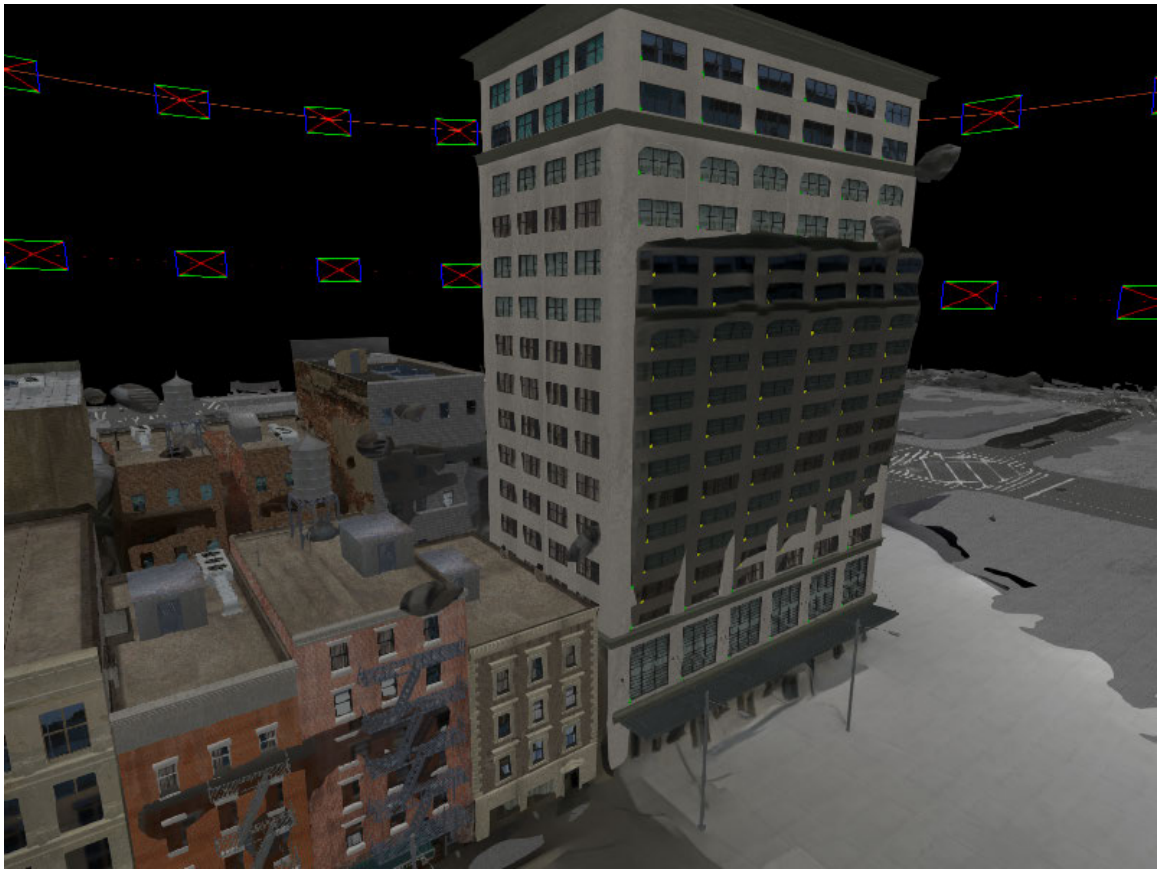

Figure 21. Uncorrected reconstruction of Dataset 7

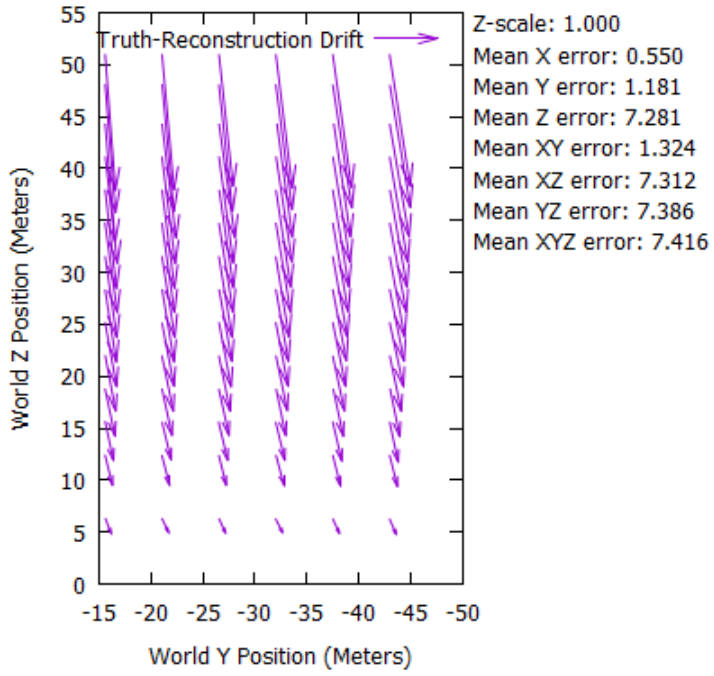

Figure 22. Uncorrected geopoint drift of Dataset 7 (error values given in meters)

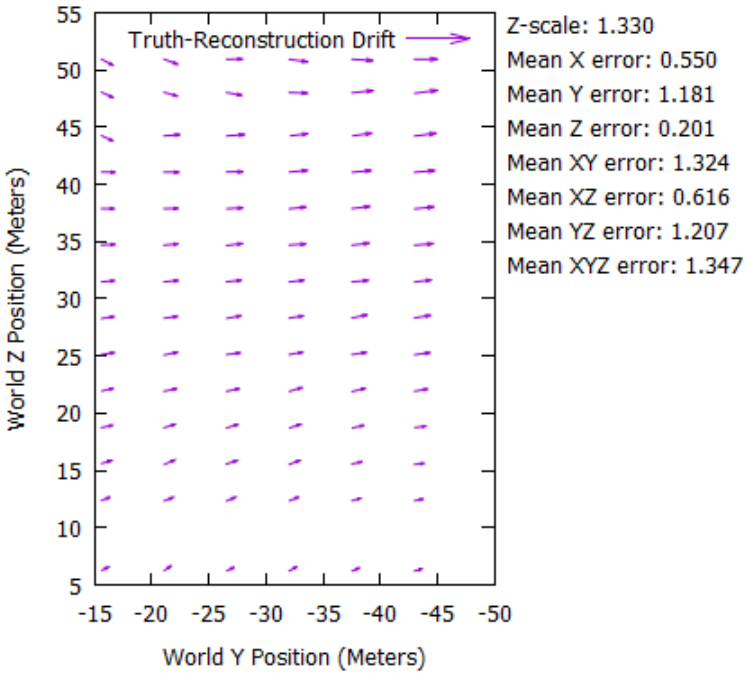

Figure 23. Corrected geopoint drift of Dataset 7 (error values given in meters)

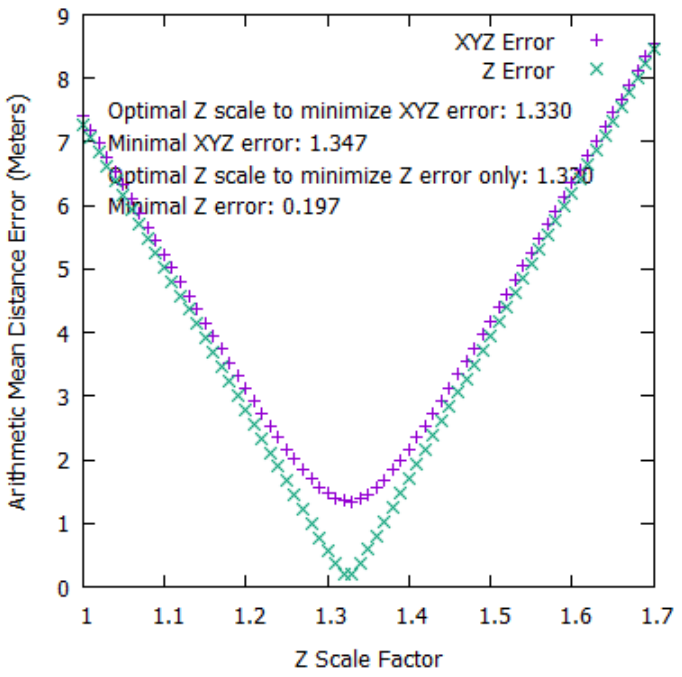

Figure 24. Error values at various Z-scale adjustments for Dataset 7 (error values given in meters)

Dataset 4 required less Z-scaling—1.22, compared with Dataset 7's optimal value of 1.33—to correct the original reconstruction. Figure 27 shows an uncorrected view of Dataset 4, while Figure 28 shows a corrected view with the optimal Z-scale.

Both Datasets 7 and 4 yielded superior final accuracy results to the others. This may support a determination that flight paths with longer distances, narrower fields of view, and initial starting positions at half the height of the scene are superior in SfM applications involving large building reconstruction, provided an appropriate Z-scale factor can be determined. In a real-world application, where truth geopoints may not be known, determination of a scale factor may be aided by GPS altitude data and cheap, lightweight range-finding equipment such as ultrasonic sensors. The UAV would fly over the highest point of the scene, which is determined by the highest point in the initial reconstruction after ground alignment, find the true height of the scene, and scale up the reconstruction to match.

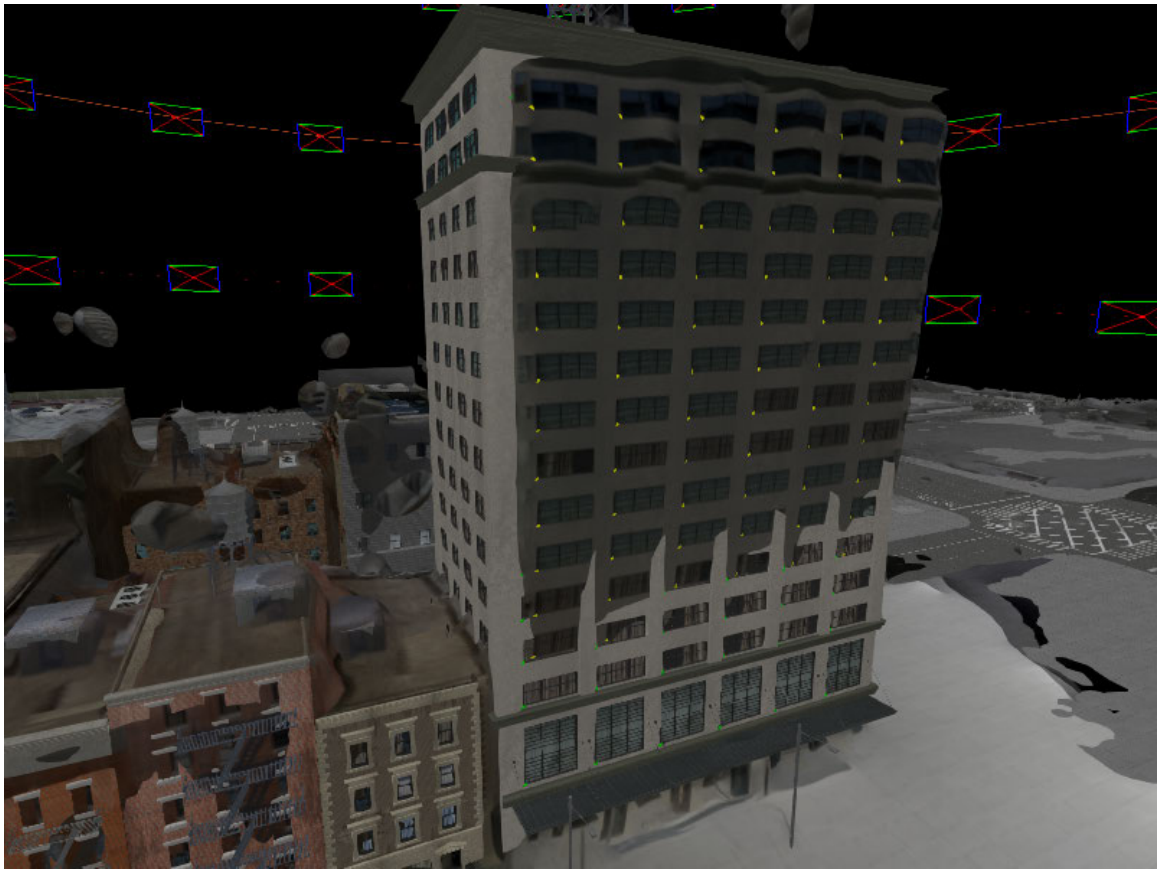

Figure 25. Corrected reconstruction of Dataset 7

Table 5. Dataset storage size comparison

#	Image Set Size (Compressed)	Reconstr. Model Size (Compressed)	Size Reduction
1	96.1MB	39.3MB	59.1%
2	89.7MB	36.6MB	59.1%
3	72.1MB	33.7MB	53.3%
4	81.5MB	41.3MB	49.3%
5	96.4MB	38.8MB	59.8%
6	73.0MB	29.6MB	59.5%
7	83.3MB	37.4MB	55.1%

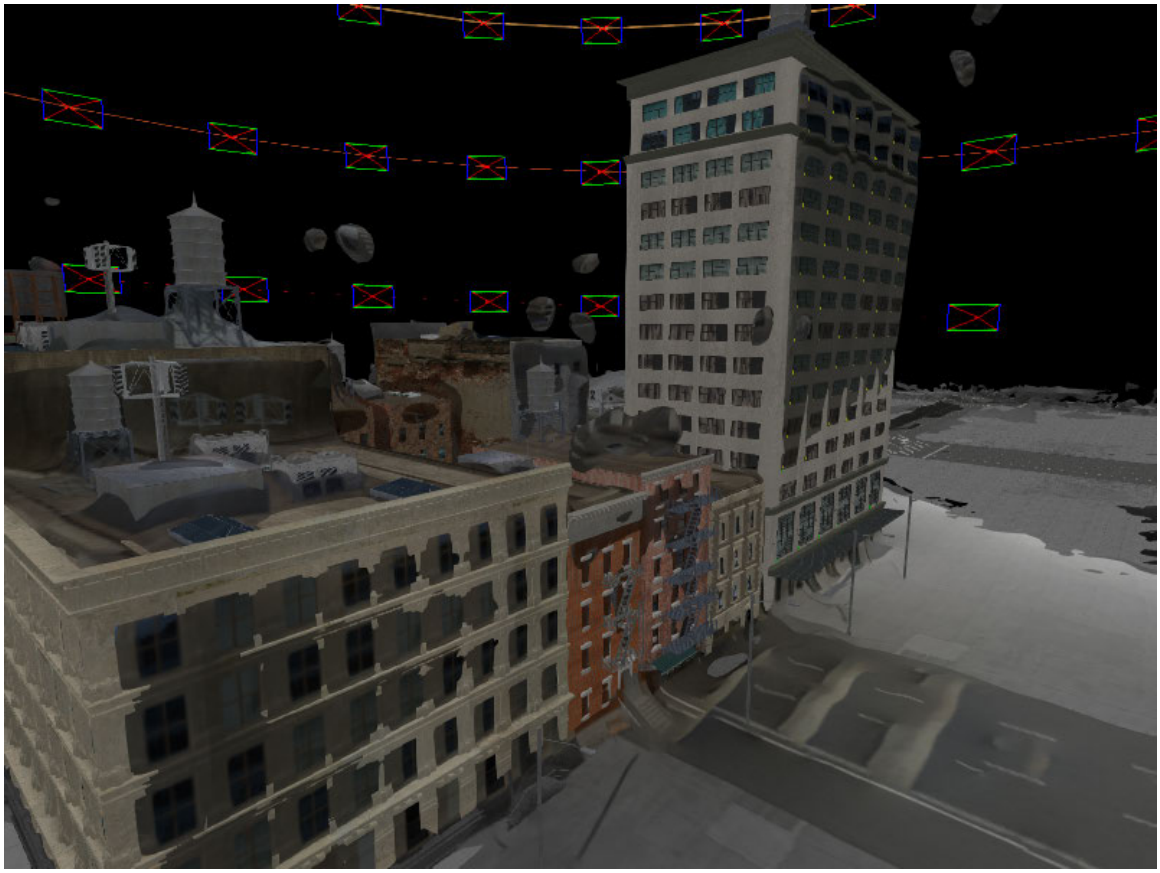

Figure 26. Corrected reconstruction of Dataset 7 (another viewpoint)

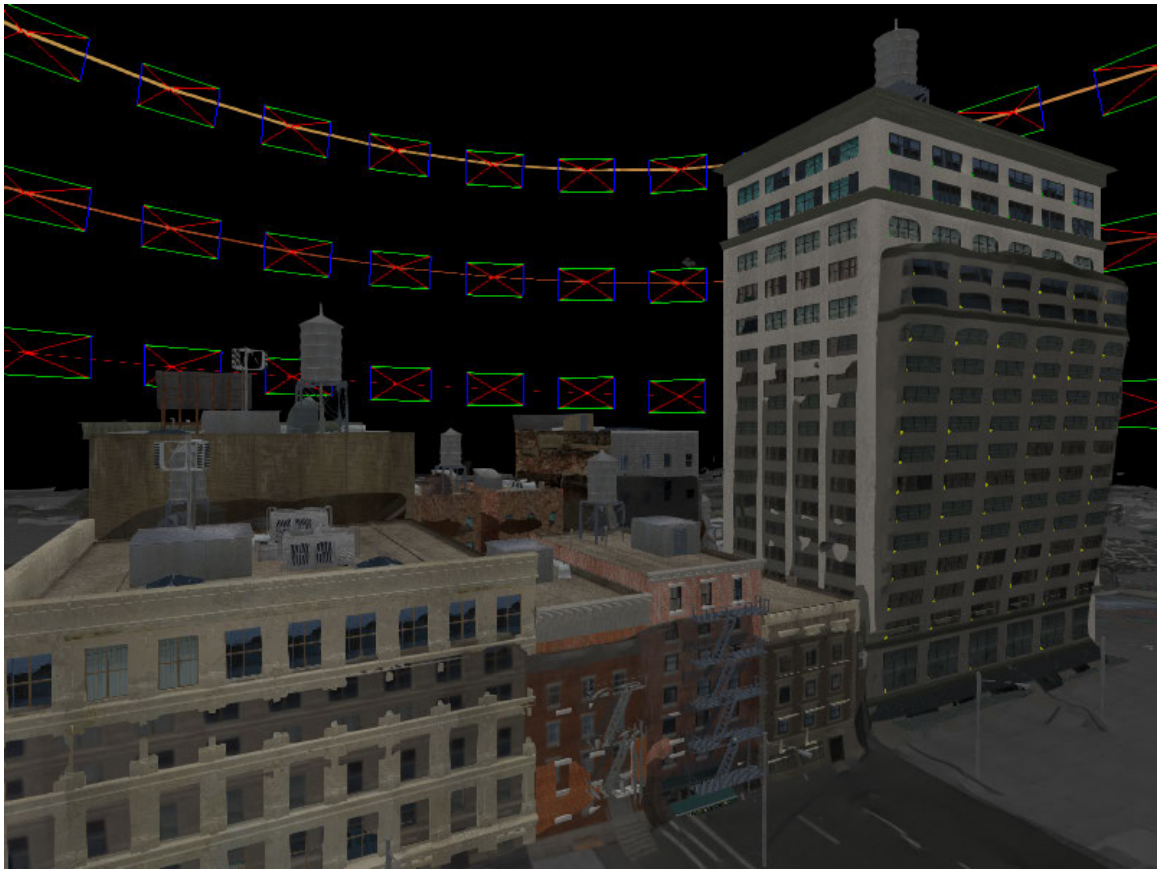

Figure 27. Uncorrected reconstruction of Dataset 4

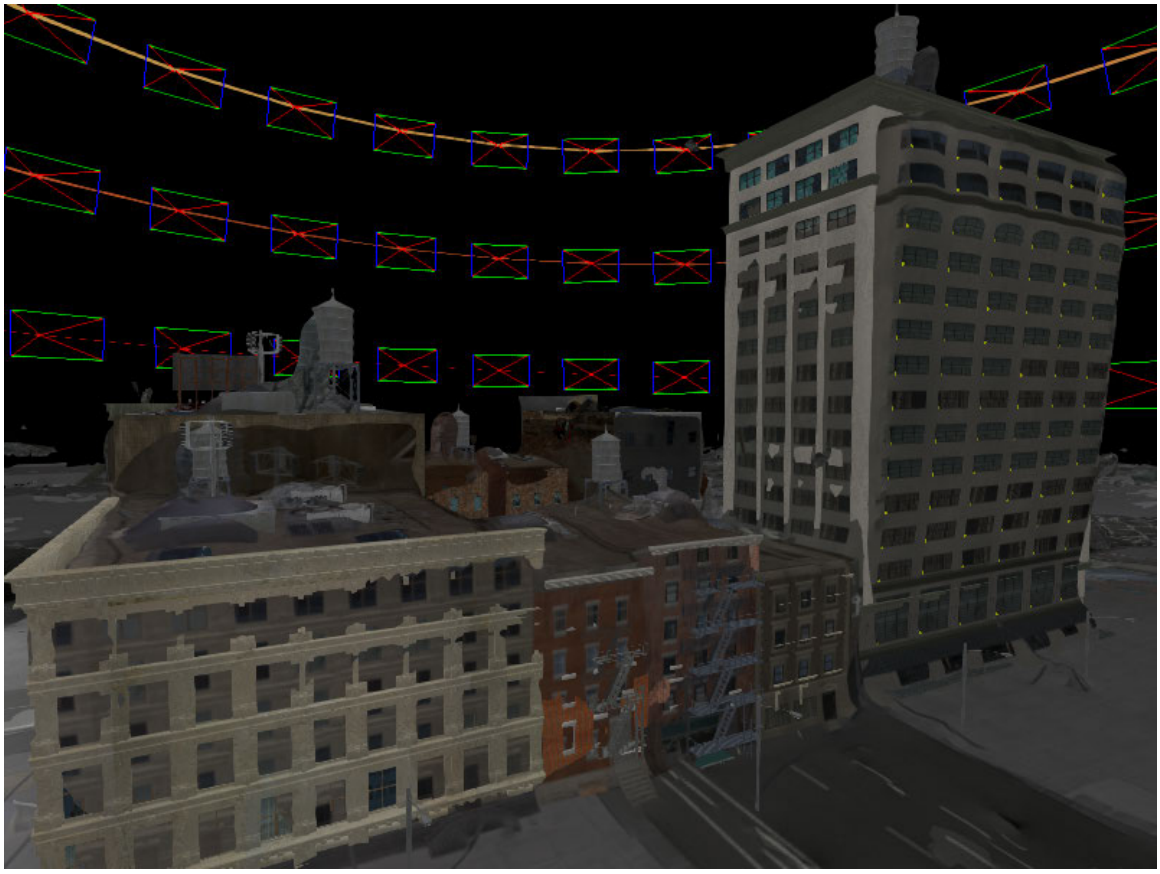

Figure 28. Corrected reconstruction of Dataset 4

Table 5 compares the compressed storage size of each dataset's set of 90 images with the size of its corresponding compressed reconstruction. All models were 49-60% smaller than the set of images from which they were created. All compression was performed with Igor Pavlov's 7-Zip utility. This reduction in size is very favorable for any application which requires imagery to be transmitted through a bandwidth-constrained link, and it may be especially beneficial to intelligence applications which require not only imagery but also geospatial awareness of targets of interest. A reconstructed 3D model, with proper alignment and low geospatial error, can precisely determine the latitude, longitude, and altitude of a specific feature, such as a window or door; this information is far more difficult to determine from only 2D images.

4.2 Discussion

Main Findings.

The warped Z-scale effect discussed in [9] was confirmed, with some differences and expansions upon the work. In the previous work, the warped Z-scale was attributed to imagery taken from nadir angle; however, in this work, a similar effect was also observed in reconstructions obtained from non-nadir images and a different SfM software suite. It was also shown that this warping effect can be compensated by applying a Z-scale typically between 1.2 and 1.6 to obtain greater geoaccuracy. With the cityscape dataset, the best results were obtained from images taken at further distances with narrower fields of view, especially when flight paths started at half the height of the target of interest.

It was also shown that models from SfM can be compressed to require less space than the images from which they were created, with the added benefit of geospatial information. This provides a compelling case for use of SfM in intelligence applications or any similar application which relies on low-bandwidth data connections.

Additionally, the work was performed in a virtual world[35], allowing for fine-tuned control, deterministic repetition of results, and cost and schedule savings; this approach demonstrated a low-risk method to inform and guide future investment and research.

Shortcomings.

The use of a virtual world, along with manual intervention in the alignment process, created some necessary inaccuracies and intentional lack of realism in the final results. The alignment process for each reconstruction was performed manually, leading to potential inconsistencies between datasets. This process could be automated by methods discussed in other work [9][17], but the resulting alignment may be less accurate with respect to the ground plane than manual alignment, causing even greater error. This issue must be addressed to fully automate and obtain acceptable results from, e.g., a UAV-based intelligence gathering process where human intervention is not possible.

The virtual world setting also introduced some unrealistic artifacts. The scene included many repeating textures, potentially confusing the feature matching steps and degrading the final result. Weather and lighting conditions were theoretically perfect, and the virtual UAV camera had no radial distortion or flaws naturally occurring in manufacturing processes; these perfections yielded clarity in the feature detection steps which is not possible in real-world scenarios. These issues with the virtual world could be mitigated by utilizing a scene with less textural repetition, including photorealistic weather and lighting conditions, and adding simulated inaccuracies in the camera construction.

V. Conclusions and Future Work

5.1 Conclusions

This research utilized a virtual world to quantify Structure from Motion (SfM) software performance for potential use in military intelligence applications. The virtual world approach enabled assessment of ideas in a perfect, controllable environment with rapid reconfiguration for different datasets. The properties of the flight path and camera were modifiable between test runs with simple tweaks to a small configuration file, enabling quick collection of varying types of synthetic imagery. The virtual world also enabled low-cost testing by preventing the need to purchase quadrotor aircraft, cameras, and other accessories which would be necessary for a real-world flight test. The virtual experiment also reduced necessary training by eliminating the need to register an aircraft and earn the license to fly, and real-world weather conditions did not impact test scheduling when they otherwise would during a real-world test.

A concept was presented to transmit reduced-size 3D models, rather than image or video streams, over bandwidth-constrained links, with the added benefit of geospatial information in the models. The virtual aircraft collected images in a precisely-defined spiral flight path, and the images were input to a Structure from Motion (SfM) pipeline. The SfM pipeline returned a 3D model for each dataset, enabling analysis and comparison to their corresponding input sets. The 3D models contained geospatial information not originally included in the 2D images, which may be beneficial to intelligence entities.

The accuracy of the geospatial information was assessed by comparing geopoint locations on the original scene to their corresponding geopoint locations in the reconstructed scene after manual ground alignment. Most datasets showed 3D accuracy to within 10 meters before correction, with higher error values in the Z dimension than

in the X and Y dimensions. This bias toward Z error confirms similar findings in [9].

Mitigating factors were proposed to compensate for distortion in the Z dimension. The reconstructed model was scaled upward in the Z dimension at discrete scale values, and at each scale value, the overall error was calculated. For each dataset, the Z scale value which corresponded to the lowest overall error was considered optimal. The optimal Z scale values ranged from 1.20 to 1.62, and the corrected models had error values from 1 to 5 meters.

The 3D models and their corresponding image sets were also compressed through the free 7-Zip compression utility, and the size of each compressed model was compared with the size of its corresponding compressed image set. A reduction of 49.3% to 59.8% of required space was shown. If employed on a large scale, the SfM method of imagery transmission may reduce the overall amount of satellite bandwidth required to support remote intelligence collection and reduce the need for expensive new space assets.

With further work to automate the process and improve its accuracy, timeliness, and energy consumption, the SfM-based approach may prove superior to traditional video- and image-based approaches to unmanned aircraft-based intelligence collection.

5.2 Future Work

This work is only a small portion of research necessary to enable SfM-based intelligence collection. Further work is required to determine accuracy of reconstruction under realistic, less-than-ideal conditions.

An SfM pipeline must be assessed on relevant hardware, as it may not be realistic to assume that a high-powered desktop or server processor will be available in a real-world scenario. A starting place for this assessment is the Nvidia Jetson series of embedded processors. The Jetson boards feature Nvidia Tegra graphics processing

units (GPU), allowing for energy-efficient GPU operations to take place. To make this approach work, a suitable software solution must be found and adapted to the ARM processor architecture present in Jetson boards, and the solution must support GPU acceleration to take full advantage of the Jetson's processing capabilities. The Multi-View Environment, Poisson Reconstruction, and Multi-View Stereo-Texturing tools utilized in the initial research effort do not support GPU acceleration at the time of writing, but open-source tools are rapidly being developed which may support GPU acceleration throughout all stages of the SfM pipeline.

A more rigorous quantification of time and energy cost is also required. The SfM approach in the initial research was performed without close monitoring of required time and without any monitoring of required energy. These factors are of critical importance in a real-world scenario, where there is a direct correlation between energy required and fuel spent, and intelligence analysts need real-time data without excessive waiting for SfM processing to complete. An initial assessment of required time and energy would require a representative system to work with. A measurement of time would be most useful if both CPU time and wall clock time were measured, to allow comparison and possible identification and removal of unneeded processes on the platform. An energy quantification could start with usage of a simple power usage tracker such as the Kill-a-Watt, and further analysis could then convert the used energy into actual fuel requirements based on the intended real-world platform.

Some important factors in the initial research were abstracted away for simplicity, such as the background scenery. The sky and landscape were removed from the virtual world, but this is not an option for real-world research. Work has been conducted to remove the background for outdoor SfM work, but more research is needed to understand how such an approach may work with aerial imagery [16]. Image segmentation algorithms may help with this effort, but the algorithm parameters would

require robust values to ensure as much background is removed as possible without removing the important parts of the scene.

A simple spiral flight path may be realistic for small, quadrotor-style aircraft, but such a path is not as likely to be utilized for larger, fixed-wing aircraft. A useful extension to the spiral work would include other types of flight paths. Some research in this area was conducted by Ekholm, but revalidation and definition of further flight path styles may reveal ideal paths to use for SfM imagery capture [18].

Additionally, it may be useful to research the performance of SfM with many different sources of imagery, as shown in [38] for crowd-sourced images of the Colosseum in Rome. There are many different intelligence platforms, and image sensor specifications may vary greatly among them; research to determine how different images may be and still work acceptably well with SfM pipelines may help to determine which sources of imagery can work well together for this purpose.

In addition to further flight path research, more study is needed to determine the feasibility of using long-range imagery for SfM applications. In this case, *long range* may refer to high-altitude aircraft or even satellite imagery. In a real-world intelligence scenario, it may be unlikely to obtain close-range imagery of a target, necessitating a greater effort to work with longer-range imagery to accurately reconstruct 3D structures. Some work has been done in a synthetic environment to determine performance of near-nadir imagery in SfM applications and may serve as a starting point to evaluate longer-distance, lower-resolution pictures for their potential usefulness [9].

Future work may also replace the manual alignment process described in this paper with an automated approach. The automated approach would utilize positioning data from the original images, e.g. in the form of embedded geotags with latitude, longitude, and altitude information, and position the reconstructed model in the

virtual world based on this information. This would also enable a proof of concept for real-time model transfer to an accurate location and scale in a second instance of the virtual world, which may represent an intelligence operator's view. The OpenMVG library provides rudimentary support for geotagged source images and may be a useful starting point to replace the manual alignment process with an automatic process.

In the initial research effort, the reconstructed models showed a significant shortening of height. This shortening was corrected with the use of geopoints on the original scene's model and the reconstruction. However, in a real-world intelligence scenario, the geopoints on the original scene would not be available for reference. An automated alternative to the geopoint-based approach may be to perform the reconstruction, determine the highest point on the reconstruction, fly to the corresponding point in the original scene, utilize a range-finding sensor to determine the height of the scene at that point, and scale the reconstruction to match the newly-found height. Further research could simulate and validate or invalidate this approach, providing a more realistic method to account for Z-scaling error in the reconstruction.

The shortening of height may also be corrected at the algorithmic level. A detailed analysis of state-of-the-art algorithms may reveal opportunities for optimization for the case of downward-facing imagery. A compensation or correction for Z-scale error, perhaps based on downward tilt of the camera, may be inserted directly into algorithms for sparse and dense reconstruction. This would eliminate the need to consider Z-scale error after the model is created and would increase accuracy earlier in the SfM pipeline, reducing cascading errors in the subsequent steps.

Finally, real-world flight tests must be conducted to determine the performance of SfM pipelines under non-ideal conditions. A sufficiently refined SfM implementation should be able to handle compounding factors such as glare and shadows, which may appear on any sunny day. With sufficient preparation and planning, real-world flight

tests would further assist leaders in determining the feasibility of the SfM imagery solution and make decisions about continued research and acquisition.

Bibliography

1. T.S. Kelso. Analysis of the 2007 Chinese ASAT Test and the Impact of its Debris on the Space Environment. *Advanced Maui Optical and Space Surveillance Technologies*, pages 321–330, 2007.

2. J. C. Liou. Collision activities in the future orbital debris environment. *Advances in Space Research*, 38(9):2102–2106, 2006.

3. Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library.* " O'Reilly Media, Inc.", 2016.

4. Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

5. Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

6. Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

7. Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision*, pages 836–850. Springer, 2014.

8. Richard Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

9. David Nilosek, Derek J. Walvoord, and Carl Salvaggio. Assessing geoaccuracy of structure from motion point clouds from long-range image collections. *Optical Engineering*, 53(11):113112, 2014.

10. Francesco Nex and Fabio Remondino. UAV for 3D mapping applications: a review. *Applied Geomatics*, 6(1):1–15, 2013.

11. Darren Turner, Arko Lucieer, and Christopher Watson. An automated technique for generating georectified mosaics from ultra-high resolution Unmanned Aerial Vehicle (UAV) imagery, based on Structure from Motion (SFM) point clouds. *Remote Sensing*, 4(5):1392–1410, 2012.

12. Mark A. Fonstad, James T. Dietrich, Brittany C. Courville, Jennifer L. Jensen, and Patrice E. Carbonneau. Topographic Structure from Motion: a new development in photogrammetric measurement. *Earth Surface Processes and Landforms*, 38(4):421–430, 2013.

13. L. Javernick, J. Brasington, and B. Caruso. Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry. *Geomorphology*, 213:166–182, 2014.

14. M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

15. Anestis Koutsoudis, Blaž Vidmar, George Ioannakis, Fotis Arnaoutoglou, George Pavlidis, and Christodoulos Chamzas. Multi-image 3D reconstruction data evaluation. *Journal of Cultural Heritage*, 15(1):73–79, 2014.

16. Hansung Kim, Jean-Yves Guillemaut, Takeshi Takai, Muhammad Sarim, and Adrian Hilton. Outdoor Dynamic 3-D Scene Reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(11):1611–1622, 2012.

17. Daniel Alix. Error Characterization of Flight Trajectories Reconstructed Using Structure from Motion, 2015.

18. Jared Ekholm. 3-D Scene Reconstruction from Aerial Imagery, 2012.

19. Kevin Colson. Toward Automated Aerial Refueling: Relative Navigation with Structure From Motion, 2016.

20. Christopher Parsons and Scott Nykl. Real-time automated aerial refueling using stereo vision. In *International Symposium on Visual Computing*, pages 605–615. Springer, 2016.

21. Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Trans. Graph.*, 32(6):169:1—-169:11, 2013.

22. F. Reichl, J. Weiss, and R. Westermann. Memory-Efficient Interactive Online Reconstruction From Depth Image Streams. *Computer Graphics Forum*, 35(8):108–119, 2016.

23. Changchang Wu. Towards linear-time incremental structure from motion. *Proceedings - 2013 International Conference on 3D Vision, 3DV 2013*, pages 127–134, 2013.

24. Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore Bundle Adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, number 1, pages 3057–3064, 2011.

25. Noah Snavely, Steven Seitz, and Richard Szeliski. Photo Tourism: Exploring Photo Collections in 3D. *SIGGRAPH Conference Proceedings*, pages 835—-846, 2006.

26. Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

27. Simon Fuhrmann, Fabian Langguth, and Michael Goesele. MVE-A Multi-View Reconstruction Environment. *Eurographics Workshop on . . .*, pages 11–18, 2014.

28. Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.

29. Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, volume 2008, pages 129–136, 2008.

30. Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

31. C. Sweeney, T. Höllerer, and M. Turk. Theia: A fast and scalable structure-from-motion library. *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, pages 693–696, 2015.

32. Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

33. Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

34. Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

35. Scott Nykl, Chad Mourning, Mitchell Leitch, David Chelberg, Teresa Franklin, and Chang Liu. An overview of the steamie educational game engine. In *Frontiers*

in Education Conference, 2008. FIE 2008. 38th Annual, pages F3B–21. IEEE, 2008.

36. M. Alliez, P. and Cohen-Steiner, D. and Tong, Y. and Desbrun. Voronoi-based variational reconstruction of unoriented point sets. *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 39 – 48, 2007.

37. Josiah Manson, G. Petrova, and Scott Schaefer. Streaming Surface Reconstruction Using Wavelets. *Computer Graphics Forum*, 27(5):1411–1420, 2008.

38. Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing Rome. *Computer*, 43(6):40–47, 2010.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
03–22–2018	Master's Thesis	Sep 2016 — Mar 2018

4. TITLE AND SUBTITLE

Assessment of Structure from Motion for Reconnaissance Augmentation and Bandwidth Usage Reduction

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Roeber, Jonathan B., Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-18-M-055

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

intentionally left blank

10. SPONSOR/MONITOR'S ACRONYM(S)

intentionally left blank

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Modern militaries rely upon remote image sensors for real-time intelligence. A typical remote system consists of an unmanned aerial vehicle, or UAV, with an attached camera. A video stream is sent from the UAV, through a bandwidth-constrained satellite connection, to an intelligence processing unit. In this research, an upgrade to this method of collection is proposed. A set of synthetic images of a scene captured by a UAV in a virtual environment is sent to a pipeline of computer vision algorithms, collectively known as Structure from Motion. The output of Structure from Motion, a three-dimensional model, is then assessed in a 3D virtual world as a possible replacement for the images from which it was created. This study shows Structure from Motion results from a modifiable spiral flight path and compares the geoaccuracy of each result. A flattening of height is observed, and an automated compensation for this flattening is performed. Each reconstruction is also compressed, and the size of the compression is compared with the compressed size of the images from which it was created. A reduction of 49-60% of required space is shown.

15. SUBJECT TERMS

Structure from Motion; Modeling and Simulation; Geoaccuracy; Intelligence, Surveillance, and Reconnaissance; Unmanned Aerial Vehicle

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Scott Nykl, PhD, AFIT/ENG
U	U	U	UU	80	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4395; scott.nykl@afit.edu