We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

5,500
Open access books available

136,000
International authors and editors

170M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

3D Reconstruction through Fusion of Cross-View Images

Rongjun Qin, Shuang Song, Xiao Ling and Mostafa Elhashash

Abstract

3D recovery from multi-stereo and stereo images, as an important application of the image-based perspective geometry, serves many applications in computer vision, remote sensing, and Geomatics. In this chapter, the authors utilize the imaging geometry and present approaches that perform 3D reconstruction from cross-view images that are drastically different in their viewpoints. We introduce our project work that takes ground-view images and satellite images for full 3D recovery, which includes necessary methods in satellite and ground-based point cloud generation from images, 3D data co-registration, fusion, and mesh generation. We demonstrate our proposed framework on a dataset consisting of twelve satellite images and 150 k video frames acquired through a vehicle-mounted Go-pro camera and demonstrate the reconstruction results. We have also compared our results with results generated from an intuitive processing pipeline that involves typical geo-registration and meshing methods.

Keywords: cross-view 3D fusion, photogrammetry, remote sensing, mesh reconstruction, 3D modeling

1. Introduction

3D data generation often requires expensive data collection such as aerial photogrammetric or LiDAR flight [1, 2]. Depending on the required accuracy, resolution and other specs of the final products, the efforts in data collection and processing can exponentially grow. Alternative and low-cost data sources are of particular interest for wide-area 3D modeling [3]. Satellite sensors running 24/7 offer overview images covering large regions with single scans, which comparatively come with lower cost than aerial flights and do not require physical access to the area of interest [4]. On the other hand, there exist many ground-view images coming either from crowdsourcing platforms or collected using relatively low-cost equipment (e.g., video frames from low-cost cameras) that provides high-resolution information of objects. Both the overview and the ground-view data are complementary to each other and their view differences being approximately 90° forms cross-view dataset, a fusion of which may yield a low-cost solution for city-scale 3D modeling. This chapter describes our ongoing work (an earlier work is described in [5]) in an attempt to address this challenging task by proposing an integrated framework to fuse the 3D results of satellite overview and ground-view video frames to generate 3D textured mesh models presenting both top and side view features.

The available commercial satellite images often have 0.3–0.5 m GSD (ground sampling distance) and ground-view images can easily reach a GSD of a few millimeters. With significantly different resolution, the resulting 3D geometry may be associated with different uncertainties, which adds additional challenges for the fusion task of these two types of data, which include:

1. The quality of 3D output separately generated from satellite images and ground-view images are scene-specific and may differ in terms of completeness and accuracy. Algorithms and basic principles for addressing image-based 3D modeling are relative standard, thus the image quality and their respective characteristics play a major role in the reconstruction results, such as the photo-consistency/temporal differences/illumination among images, their geometric setup, completeness in terms of coverage, intersection angles, etc.

2. Due to the large view differences, the overview and ground-view dataset may share very limited region in common, and additionally the 3D output from the ground-view dataset may come with no geo-referencing information and may contain non-rigid topographic distortions (e.g., trajectories drift or distortions due to inaccurate interior/exterior orientation estimation), which further add challenges in 3D geo-registration of the dataset.

3. The combined 3D point clouds are from two sources with different resolution, uncertainty, and radiometric properties of textures, which present difficulties in both the geometric reconstruction of meshes and the texture mappings. Thus, obtaining visually consistent textured meshes the preserve information to the maximal extent is extremely challenging.

We introduce in our proposed method major contributions to address the above-mentioned challenges to form a complete fusion pipeline. These contributions are: (1) we introduce a monocular video-frame-based 3D reconstruction pipeline to achieve the minimal geometric distortion by leveraging the speed and accuracy in a photogrammetric reconstruction pipeline called MetricSFM. (2) We introduce a cross-view geo-registration and fusion algorithm that takes point clouds generated from satellite multi-view stereo (MVS) images and from ground-view videos, to co-register the ground-view point clouds to the overview point clouds; (3) we extend a view-based meshing approach to accommodate point clouds with images coming from different cameras. The rest of this chapter is organized as follows: Section 2 introduces related works and the overview of the proposed pipeline; Section 3 introduces our methodologies of the components of the pipeline in details, Section 4 describes the experiment dataset and the results of the 3D reconstruction; and Section 5 concludes this chapter by discussing potential works moving forward.

2. Related works and an overview of the proposed pipeline

The uses of multi-source 3D data have been attempted for different purposes, such as for localization, geo-registration, image synthesis, cartographical model generation [6–9], and planetary applications using different types of sensors [10–14]. For example, [8] utilized a combination of UAV (Unmanned Aerial Vehicles) images and mobile LiDAR (Light Detection and Ranging) for 3D model generation, where the geo-registrations are performed using manually measured ground control points (GCP) from the LiDAR data, followed by a Bundle

Adjustment [15] of the UAV images. All were performed following a surveying-grade processes, thus minimal topographical distortions needed to be addressed in critical or non-optimally collected data (e.g., monocular video collection with a single trajectory).

Correlating the satellite overview and ground view images is extremely challenging because the areas in common can sometimes be barely the ground or even less (due to vegetations and moving objects). There are two types of approaches to address relevant tasks, such as (1) cross-view images localization [9, 16, 17] and (2) cross-view image synthesis [6, 7]. Since the traditional feature-based matching methods fail in cross-view data, the major technical approaches for cross-view data instead are to learn deep representations between cross-view data, with various strategies for learning scene-level descriptors used to match cross-view data, combing learned semantics and geometric transformation. A few early works also explored the use of manually crafted features for such a task [16, 18]. Most of the existing methods exploring 3D data co-registration require a certain common regions, and the transformation is often assumed to be simple models such as similarity or rigid transformations [19, 20]. Thus, exploring methods for registering wide-area, cross-view dataset potentially with complex geometric distortions are particularly of interest and can offer tangible solutions for low-cost 3D data generation.

Meshing point clouds seems to be a standard practice with many applicable algorithms available [21]. However, for image-based point clouds, meshing requires the use of the visibility information between the view and each point [22, 23] which sometimes are not easily available for multi-source data as first of all, they may share different camera model, and second of all, standard software packages generating point clouds from images do not offer such visibility information. As a result, a standard practice of using multi-source image-based point clouds only takes point-cloud-based meshing methods [21], which are designed for very dense point clouds and do not necessarily work well for point clouds with the level of uncertainty and complexity as the image-based point clouds.

Despite these challenges, we consider the problem of turning the MVS satellite images and ground-view Go-pro data to be approachable, if scenario-specific information and intermediate results of the stereo reconstruction pipeline are available. To achieve, we have the following three considerations:

1. Monocular ground-view video frames taking alongside the street do not offer an optimal camera network, thus it is possible that the results of the 3D reconstruction contain geometric distortion, for example, trajectory drifts, or topographic distortion due to the incorrectly estimated interior/exterior orientations [24], which will further add challenges to the geo-registration, we therefore consider to optimize our photogrammetric reconstruction workflow by considering self-calibration for each incremental reconstruction to minimize the potential trajectory drift.

2. We observed that in an urban environment, the boundary of objects from the satellite point clouds, for example, buildings, might coincide well with the boundary produced by projecting the façade point clouds to the ground; therefore it can be seen as a view-invariant feature for co-registering the satellite point clouds and ground-view point clouds.

3. Meshing methods will unlikely to work well on the combined point clouds (from satellite and ground-view point clouds) without the use of visibility information. Although theoretically possible, re-implementing a meshing
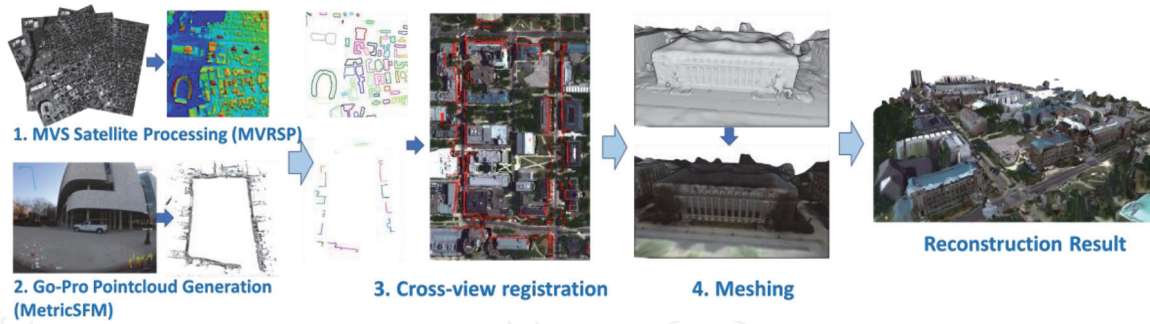
Figure 1.
The general workflow of our processing pipeline.

algorithm considering different camera models can be painstakingly trivial. We consider the satellite point clouds to be associated with an orthophoto under a parallel projection, thus the visibility can be easily computed and incorporated into an image-based meshing [23] and texture mapping pipeline [25].

To sum, our proposed data generation pipeline considers three major components. As shown in **Figure 1**, which includes separate 3D data generation (for MVS satellite images and ground-level video frames), geo-registration, and meshing.

The MVRSP (based on [4, 26, 27]) and MetricSFM are, respectively, our developed system for processing the satellite data and ground-level video frames. A cross-view registration method is performed for overview and ground-view point cloud registration, which utilize the boundary information derived from both types of point clouds. Finally, the co-registered point clouds are processed by a modified meshing and texture mapping algorithm that innovatively consider both perspective and parallelly projected image (satellite orthophoto) in an integrated optimization framework.

3. Methodology

3.1 Multi-view (MVS) satellite image processing

The MVS satellite processing follows methods in [4, 26], which takes a pair-wise reconstruction followed by a DSM (Digital Surface Model) fusion as shown in **Figure 2**. Given a set of images, we will first apply an analysis algorithm presented
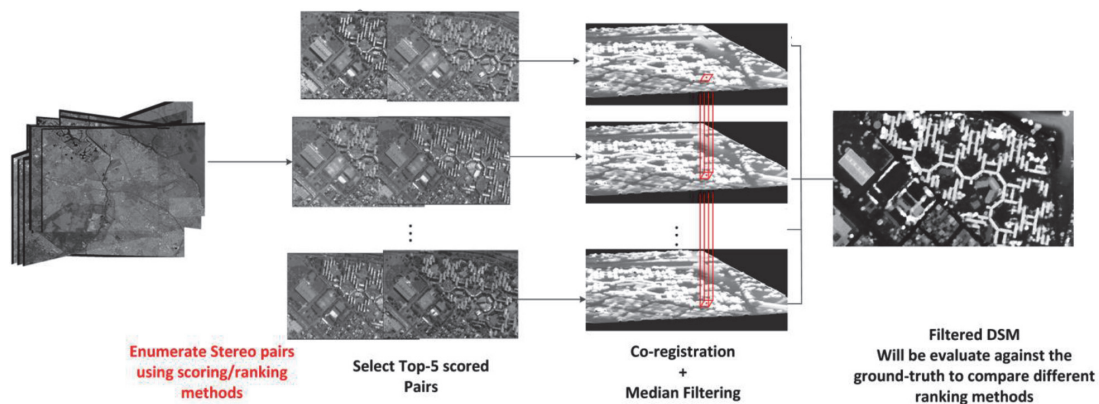

Figure 2.
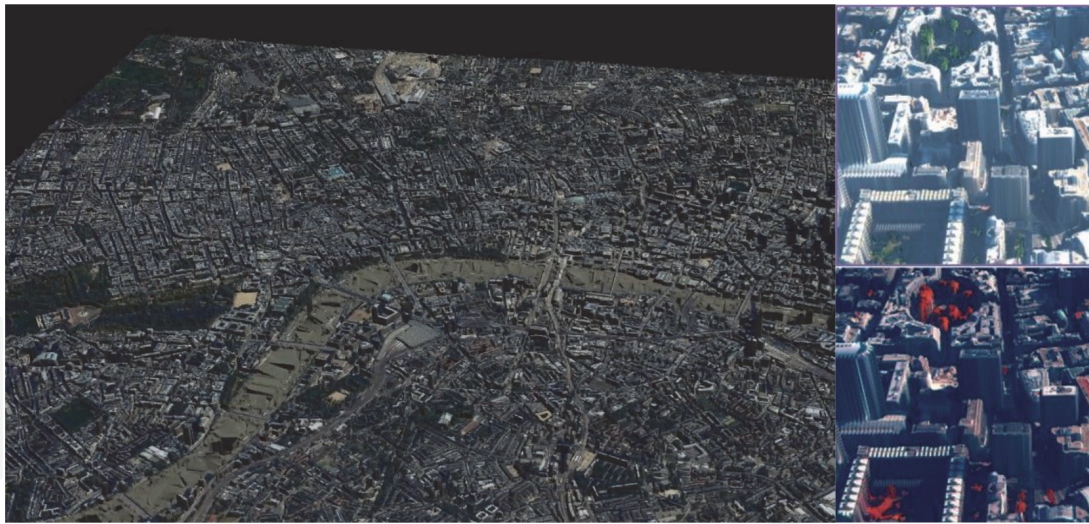*A workflow of the multi-view satellite image processing [28].*

Figure 3.
3D reconstruction of the central area in London (ca. 50 km²). Left: overview of part of the area; right: top, enlarged RGB (red, green, blue) color image, bottom, pseudo color image (near infrared, red, green).

in [28] to rank the matchability of the satellite stereo pairs (enumerated from the existing images), and then we take the top five stereo pairs to perform relative orientation and stereo dense matching using a software called RPC stereo processor [4, 27]. The core matching algorithm uses a hierarchical semi-global matching [29] with modifications to accommodate large-format images [30]. The use of multiple stereo pairs enables sufficient redundancies for high-quality 3D reconstruction, and the images consist of both Worldview I/II images (data will be introduced in Section 4). The produced individual DSMs resulting from different stereo pairs are co-registered with a shift-based registration which search for translation parameters in reference to one of the pairs (which is used to be the first pair in the pair ranking), and the co-registered DSMs are fused following an adaptive depth-fusion method [26] that utilizes the color information of the orthophoto, which were shown to achieve better accuracy than a simple median depth filtering. The readers may refer to specific details of the reconstruction in [4, 26, 28].

A typical digital surface model generated using our pipeline is shown in **Figure 3**, which indicates a 3D reconstruction result of the central area of the city of London. Worldview-III images with a 0.3-m resolution are used, thus the resulting surface models are with the same resolution.

3.2 3D reconstruction from ground-view monocular image sequences

Monocular 3D reconstruction refers to the process of recovering shape of objects using images taken from a single video camera. As compared to typical stereo/multi-stereo images captured from well-distributed angles, such video sequences present sub-optimal camera network in which the pose estimation is often inaccurate for metrically correct 3D reconstruction. Oftentimes, the structure from motion and SLAM (simultaneous localization and mapping) approaches are used to compute the camera poses and generate 3D semi-dense or dense point clouds. These methods although provide visually pleasant trajectories and point clouds, they may often be metrically incorrect and present drifting problems. In this section, we introduce a monocular 3D reconstruction system that leverages the speed of a typical SLAM system and rigorous photogrammetric optimization. We first present typical components for 3D reconstruction and then briefly introduce the processing workflow of the system.

3.2.1 A 3D reconstruction pipeline

Figure 4 presents a typical image-based 3D reconstruction pipeline. Raw images or undistorted images (through pre-calibrated parameters) are taken as the input and follow a series steps named feature extraction and matching, relative orientation, bundle adjustment and dense image matching, and output intrinsic and extrinsic orientation parameters and dense point clouds. Among these steps, the GPS (global positioning system) or IMU (inertial measurement unit) can be optionally taken as observations to bring global datum. Below we briefly introduce these components and their specifics in a ground-view image sequence scenario:

Camera intrinsic and extrinsic parameters: the camera intrinsic parameters refer to the internal geometry of the camera and often considered as focal length, principal points, and lens distortions. The extrinsic parameters refer to the poses (position and facing) of each image, normally represented by six parameters: three for a point in Euclidean coordinate (camera perspective center) and three rotation angles (sometimes are represented directly as rotation matrix).

Pair selection: pair selection tells the system what are the images that are likely to observe the same object, such that a connected graph can be built [31, 32] to formulate observations to recover 3D geometry. In the ground-view scenario, this can be simply formulated using the timestamp of the frames.

Feature extraction and matching: features represent areas or points of interest in images and denote special pieces of information. In 3D reconstruction, points are the most popular feature representations due to their simplicity and flexibility. Point features can be understood as corners or spots that are distinctive and easily identifiable across different images with various levels of perspective differences and typical features are SIFT (Scale-Invariant Feature Transform) [33], SURF (Speeded up robust features) [34], ORB (Oriented FAST and Rotated BRIEF) [35], etc. Once these points are extracted, feature matching aims to associate identical points across different images, which essentially represents corresponding rays from different images. Typically done with an exhaustive search, feature matching in a ground-view video frame scenario can be speed up by considering the fact of horizontal moving thus to reduce the search space [36].

Incremental relative orientation/pose estimation: the incremental relative orientation refers to the process starting with a two-view relative orientation, followed by sequentially orienting the rest of the images given the feature point correspondences. Often the estimation process needs to address blunders in the observations and the state-of-the-art procedure takes RANSAC (random sampling consensus) [37] for robust and automated relative pose estimation. RANSAC used a random sampling strategy that starts with randomly sampled feature matches (observations) instead of all the observations for relative orientation (model estimation), runs the same process for multiple times, and selects the model (estimated orientation parameters) accounting for most of the observations with reasonable residual. This has dramatically improved the automation in relative orientation and subsequently the incremental procedure, as it theoretically only requires the error
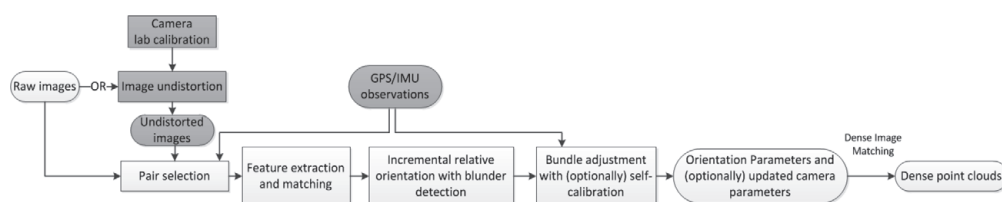

Figure 4.
A typical 3D reconstruction pipeline, dark-gray blocks indicate optional steps.

rate of the matches be larger than 50%, while apparently the state-of-the-art feature extractors and matchers do much better with images in most of the applications.

Bundle adjustment: is a refinement process for the intrinsic and extrinsic camera parameters simultaneously with the 3D coordinated of the scene points since the measurements are prone to errors [38]. It involves a global minimization scheme using robust nonlinear least-squares algorithm such as Levenberg-Marquardt [39]. This often comes with a procedure called self-calibration [40] that simultaneously estimates the lens distortions of the camera. In a ground-view video frame scenario, because the bundle adjustment is particularly time consuming, it may sometimes be simplified to only perform local bundle adjustment instead of considering all available images.

3.2.2 3D reconstruction using ground-view image sequences

Ground-view image sequences formulate a specific scenario in which a typical 3D reconstruction pipeline can be customized to accommodate the need for speed and accuracy. Our general workflow is presented in **Figure 5**. It is similar to a SLAM pipeline [36] with the differences that the local and full bundle adjustment considers the estimation of camera lens distortion parameters. Typically, the system starts with an initialization module that aims at estimating the camera pose for the two images used in the initialization by utilizing the matched features between them, this is in line with the first half of incremental relative orientation as mentioned above. Moreover, this module generates initial 3D points of the scene by triangulating the matched feature points from the two images. After generating the initial reconstruction, the tracking module (in dashed box) starts to localize every image by finding its pose, which is similar to the second half of the relative orientation which sequentially add image frames to the system. In this module, the temporal relation between the images is used by assuming a constant velocity motion model so that we can get an initial estimate of the current image pose. Thus, using the estimated pose, we can directly project the 3D points into the current image and perform window-based search for the potential feature matches with the projected points. Consequently, we can save computations by searching correspondences only inside this window instead of searching in the whole image. Then, using these correspondences, the current camera pose can be estimated. It should be noted that the concept of keyframes are used to identify important frames in which the poses will be optimized through bundle adjustment, because frames that are estimated through a constant velocity are considered to close enough to interpolate. For images that fail the constant velocity motion model, the tracking module performs full feature matches to find feature in previous frames that have an associated map point using a spatial resection (i.e., a Perspective-n-Point (PnP) algorithm) [41] by taking existing 3D points and 2D correspondences to compute their pose, and such images are then taken as the new keyframes, in the meantime features with no 3D correspondences will be triangulated as candidates of 3D map points.
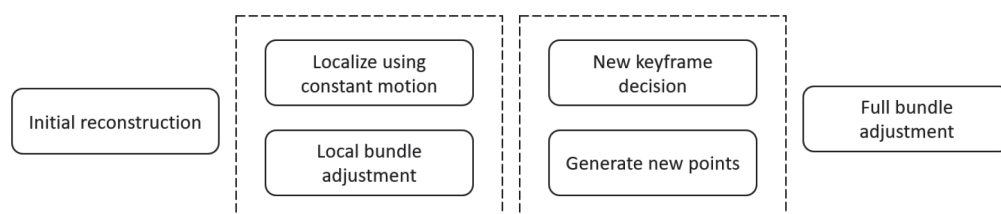

Figure 5.
A 3D reconstruction pipeline using ground-view video frames.
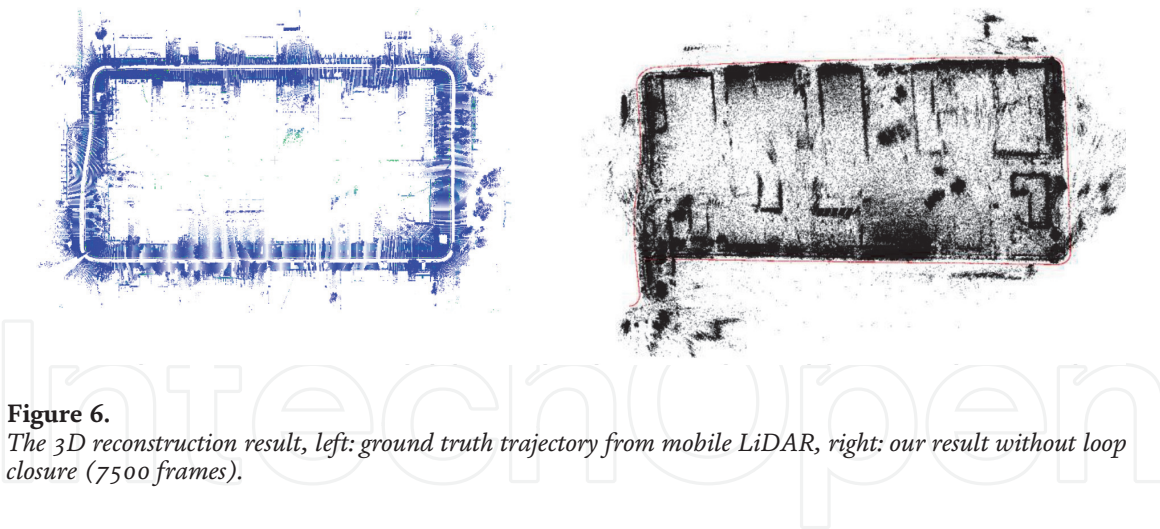
Figure 6.
The 3D reconstruction result, left: ground truth trajectory from mobile LiDAR, right: our result without loop closure (7500 frames).

Once the tracking module accumulates frames to a pre-defined number, a full bundle adjustment is used interchangeably with local bundle adjustment to refine the estimated measurements. These aforementioned processes are implemented in an in-house software package called MetricSFM. A sample from the 3D reconstruction results is shown in **Figure 6**.

3.3 Cross-view 3D point co-registration and fusion

Non-rigid distortion of the ground-view data (e.g., trajectory drift) and very limited overlapping region among cross-view data make them difficult to be registered without significant manual effort. Based on the assumption that the object boundaries (e.g., buildings) from the over-view data should coincide with footprints of façade points from ground-view, we tackle these problems by proposing a fully automated geo-registration method for cross-view data, which utilizes semantically segmented object boundaries as view-invariant features under a global optimization framework. Taking the over-view point clouds generated from satellite stereo/multi-stereo images and the ground-view point clouds from monocular video frames as the input, the proposed method takes a "two-step" strategy to solve the non-rigid cross-view registration problem using object boundaries, which is further optimized through a constrained bundle adjustment to keep 2D-3D consistencies.

3.3.1 Building boundary extraction from ground-view and over-view point clouds

The building extraction on the over-view point cloud is achieved by converting the point cloud into a digital surface model (DSM), on which the well-developed morphological top-hat [42, 43] can be used to extract a binary mask for all the high objects like tree and building. For satellite orthophoto containing multi-spectral information, the NDVI (Normalized Difference Vegetation Index) [44] can be extracted to further remove the trees from the binary masks. The ground-view building detection is based on the observation that the building façade points are usually vertical to the horizontal ground plane. We therefore determine the vertical direction by calculating the normal vector for all the points and then selecting the direction with the largest number of normal vectors pointing to the vertical directions. Once the vertical direction is obtained, all the ground-view points are projected onto the horizontal plane, which is followed by a classical region growing method [45] to extract point cloud segments. Finally, those segments with the number of points greater than a threshold are kept as the extracted ground-view buildings. The results of building boundary extraction from both over-view and ground-view data can be seen in **Figure 7**.

3.3.2 Individual building segment matching

In order to efficiently search for accurate registration parameters locally to address potential topographical errors of the point clouds (e.g., drifted trajectory resulting metrically incorrect point clouds), we developed a simple 2D point cloud registration algorithm that performs sampled exhaustive search. Given the over-view point set P_d of size n_d as the reference point cloud, and the ground-view point set P_s of size n_s as the matching point cloud, with the scale difference s between two point sets. Firstly, the distance map (as **Figure 8(b)** shows) for P_d is calculated using distance transformation [46, 47], in which the distance of each pixel (colored in gray-level, darkest referring to the closest distance) to the region of interest (in our scenario this refers to the boundary from the overview data). P_s is centralized by subtracting the central point for each point from P_s. Assuming a fixed scale determined by sparse known observations such as GPS positions, we perform an exhaustive-search through the rotation and translation space to find the optimal parameters. The final rotation parameter and translation parameter were found as ones that minimize the co-registration error in the distance map, and an example result is shown in **Figure 8(c)**.
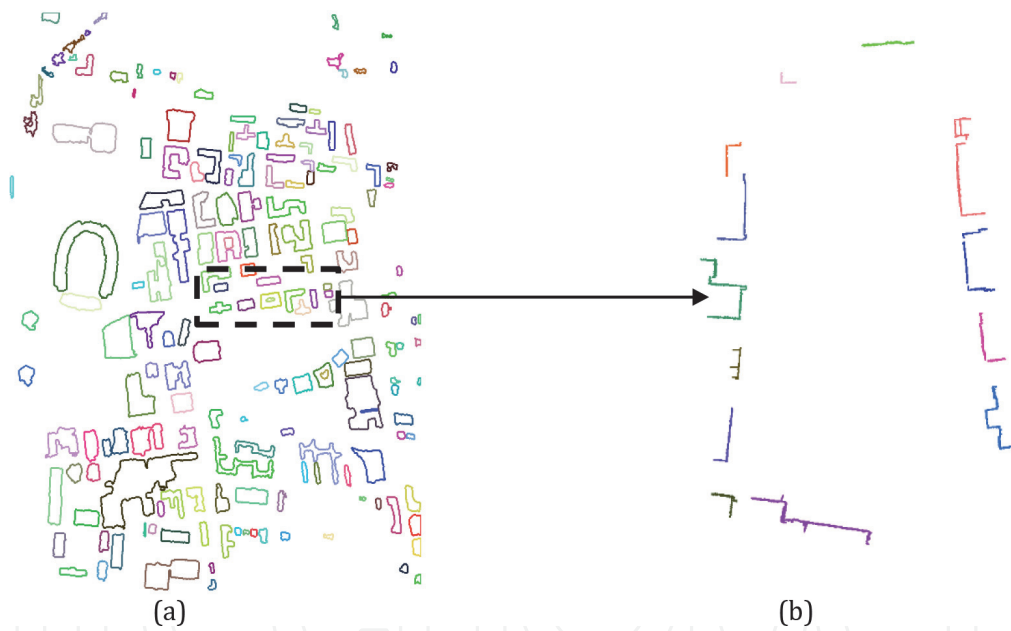

(a) (b)

Figure 7.
Illustration of building boundary extraction results from (a) over-view and (b) ground-view data.
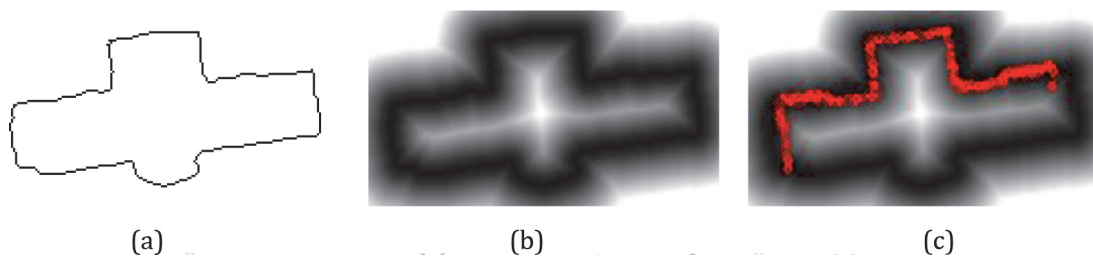

(a) (b) (c)

Figure 8.
Exhaustive search-based local matching algorithm. Given the over-view building boundary points P_d as destination in (a), the distance map in (b) is calculated where the intensity of pixel denotes the closest distance to P_d, then the global solution in (c) is obtained by our proposed method. Red points represent the ground-view point P_s.

3.3.3 Global optimization for consistent building segment matching using graph-cut

In the previous building segment matching step, a list of transformations $\mathcal{T} = \{T_i, i = 1, 2, \dots\}$ is generated, which constitutes the final hypotheses for each building segments. We consider that the transformation hypothesis for neighboring building segments to be similar, therefore, we consider formulating this constrain in an energy minimization problem (Eq. (1)):

$$E(\mathcal{T}) = \sum_{B} D(B, T) + \sum_{B_i, B_j} V_{B_i, B_j}\left(T_{B_i}, T_{B_j}\right), \tag{1}$$

where $D(B, T)$ is the data term for teach building segment B with a transformation T in $\mathcal{T}$, and $V_{B_i, B_j}\left(T_{B_i}, T_{B_j}\right)$ is the smooth term that penalizes differences of two transformations T_{B_i} and T_{B_j} of the building segments B_i and B_j.

3.3.3.1 Data term

Given a building B and a transformation T, we first collect its k-adjacent buildings (including B), measured using distance between barycentric coordinates. These segments after transformation are used to verify how close they are to the over-view building segments. To robustify the evaluation, we consider counting the number of points that are close enough to the overview building segments, as follows (Eq. (2)):

$$D(B, T) = \sum_{p \in B} c(p, p') = \begin{cases} 0, \text{ if } d(p, p') < d_{th} \\ 1, \text{ otherwise} \end{cases} \tag{2}$$

where $c(p, p')$ is the cost of a point p that belongs to the building B, which equals to 0 if the distance $d(p, p')$ between p and its closest point p' in the over-view building boundaries is smaller than d_{th}, and equals to 1 otherwise. This formulation can effectively keep the value range of the data term limited. For example, the value of $d(p, p')$ can be very large if an incorrect transformation converts the point p far away from p'; however, $c(p, p')$ can eliminate the influence of this mistake to generate more reasonable cost value.

3.3.3.2 Smooth term

The smooth term $V_{B_i, B_j}\left(T_{B_i}, T_{B_j}\right)$ penalizes the transformation associate with two neighboring buildings being too different, shown in Eq. (3):

$$V_{B_i, B_j}\left(T_{B_i}, T_{B_j}\right) = \begin{cases} p1, \text{ if } \left\|\theta_{B_i} - \theta_{B_j}\right\| < \theta_{th} \text{ and } \left\|t_{B_i} - t_{B_j}\right\| < t_{th} \\ p2, \text{ otherwise} \end{cases} \tag{3}$$

where θ is the rotation angle in 2D and t is translation, and we assign a small penalty $p1$ to neighboring segments with transformation different smaller than a given threshold, otherwise we assign a larger penalty. The weights and thresholds can be determined based on the noise level of data. The solution Eq. (1) can be achieved efficiently through graph-cut algorithm [48].

3.3.4 Bundle adjustment for pose refinement

The co-registration is further performed in the vertical direction using the overlapping ground points, and this is followed by a bundle adjustment of all image poses such that they are consistent with the registered ground-view point clouds. This is achieved by weighting the unknown poses to be close to the poses after the transformation. An additional bundle adjustment benefits the poses to be strictly following the epipolar constraints thus offers consistent 2D-3D relationship for further processing such as texture mapping.

Both the overview and ground-view point clouds are then combined, and their overlapping point clouds were fused as follows: for areas where both satellite point clouds and ground-view point clouds exist, we take the ground-view point clouds as it with a resolution presents higher accuracy and certainty. An example of co-registered cross-view point clouds is shown in **Figure 9**.

3.4 Meshing and texture mapping of cross-view fused point clouds

3.4.1 Mesh reconstruction of cross-view fused point cloud

As mentioned in Section I (Introduction), a point cloud-based meshing method [21] is unlikely to yield visually consistent meshes (an example is shown in **Figure 14**). Therefore, our solution considers the use of image information for mesh reconstruction. The base method [23] takes the constructed Delaunay tetrahedra of the point clouds as the input to extract the surface. These tetrahedra can be viewed as a connected graph, in which the tetrahedra are the nodes and shared/common faces are edges. **Figure 10** shows the procedure: black triangles denote cameras, dash arrows denote visual rays, each point in 3D space can be determined by at least two rays, which connect the object points and camera centers, here we call it ray visibility. Based on ray visibility, tetrahedra intersected with rays are evaluated by their probability to be in a free space (outer space), and tetrahedra behind the ray endpoint are evaluated by their probability belonging to the full space (inner space). Such a graph labeling can be casted to a s-t minimal cut problem and solved with maxflow algorithms [49]. The final surfaces are the common faces of the tetrahedra labeled as free and full spaces (**Figure 10**).

Our pipeline extends from this base algorithm by incorporating point clouds generated from the satellite images. The following steps give streamline from source points to surface mesh model.

Delaunay 3D triangulation: 3D triangulation or tetrahedralization is extended from 2D triangulation, which partitions a polyhedron into non-overlapping basic 3D elements, where the vertices of tetrahedra take the vertices of the original
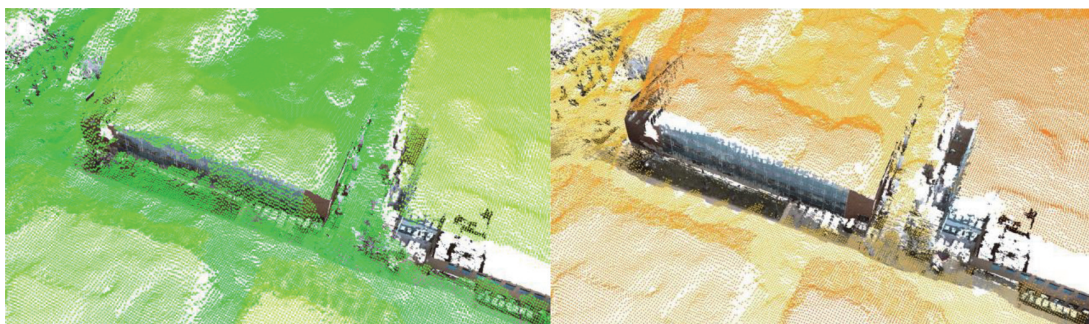

Figure 9.
Co-registered cross-view point clouds are fused (left: before; right: after) by only keeping the high resolution results. Non-textured points are over-view satellite point clouds.
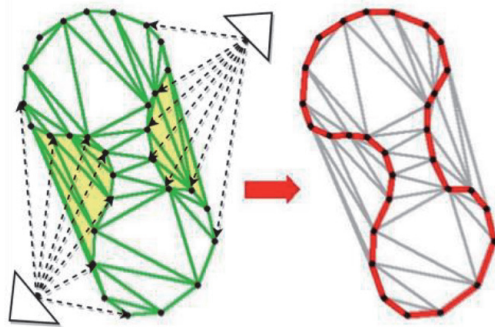
Figure 10.
Left: Green network is Delaunay triangulation, yellow region (free space) is tetrahedra which intersected with rays (dash arrows), and white region is tetrahedra labeled as full space. Right: red lines are surfaces between full and free space, which are common faces shared by those tetrahedra (artwork from [50] with minor edit).

polyhedron. Delaunay tetrahedron reconstruction [51] divides the convex hull of points into compact simplices, where neither extremely long edge nor extremely sharp angle is included. Many well-known commercial packages and open source projects have implemented the algorithm that creates Delaunay tetrahedron from point set, here we use CGAL [52] an open source computational geometric algorithm library to construct tetrahedra.

Visibility: each ray will propagate its confidence to intersected nodes (tetrahedra) and edges (triangle faces) of the tetrahedra graph. The algorithm was implemented by an open-sourced project OpenMVS [53]. Dense points and their associated images with poses are the most common source of visibility in our framework, often under a perspective geometry. However, the geometric model of satellite camera sensors is different (e.g., rational polynomial coefficients) [4]. By considering that the point clouds can be associated with the orthophoto through a parallel projection, we proposed a two-step method: (1) project satellite point on to grid, only the highest point is recorded in each cell. (2) Create vertical visual rays from those points.

Assigning weights for the graph: our method follows a so-called soft visibility weighting model that was used by the base algorithm. The readers may refer to the original paper [23] for more detail.

Solving min-cut problem: once weighting procedure for the edges is done, we use IBFS (incremental breadth first search) [54] maximum flow algorithm to solve minimum s-t cut problem. And finally, the common faces between source and sink tetrahedra are extracted to build up optimum surface model.

3.4.2 Texture mapping of cross-view fused point cloud

Our texture mapping framework is based on Waechter's work [25] which has been well practiced and widely used by rather popular open source projects, for example, OpenMVS [53]. Texturing a 3D model from multiple registered images is typically performed in a two steps approach: (1) select view(s) should be used to texture each face yielding a preliminary texture and (2) optimize the texture to avoid seams between adjacent texture patches.

Best view selection: the base method [25] determines face visibility (distinct from ray visibility) for all combinations of views and faces by first performing back face and view frustum culling, then renders faces onto images, using depth buffer to determine the nearest faces. Lempitsky et al. [51, 52, 55] compute a labeling that assign a view to be used as texture for each mesh face using a pairwise Markov random field energy formulation. We consider the ground-view images are perspective, and the satellite orthophotos are in parallel projection. Our texture

mapping considers the orthophoto as one of the images with only few simple modifications: we balanced data term of ortho images to compensate resolution gap and make ortho images as the default sources for texturing.

Seamless texture fusion: in Waechter et al.'s method [25], they proposed a global and local color adjustment method to blur the seams, which extended Lempitsky and Ivanov's [55] color adjustment approach. The original approach only accounts for color difference on vertices to measure color difference along the seam line, called global adjustment. The extended method added a local adjustment with Poisson editing [56] affect border strip of image patches. In our case, since the resolution of orthophoto is way lower than the ground-view images, prior to applying the fusion of image patches, we up-sampled orthophoto to the same resolution as that of the ground-view images. After color balancing and Poisson editing, color differences can be well-adjusted and seams are successfully been blurred.

4. Data description

We take the Ohio State University (OSU) Columbus Campus as our test site, of which we have collected 12 overlapping satellite images consisting of WorldView-I and WorldView-II images (information shown in **Table 1**). These images selectively form 31 pairs used for the reconstruction based on the method of [28], and many of these images are not from the same year thus creating challenges for the reconstruction. **Table 2** provides an overview of the first 10 pairs used from the acquired images: not all of these pairs form in-track stereo, while the large redundancy does provide the advantage in producing more accurate surface model. **Figure 11** shows the generated digital surface model. The achieved RMSE (root-mean-squared-error) is 1.26 m evaluated through LiDAR point clouds, and the RMSE reached 0.60 m by excluding changed buildings, rivers, and trees.

We have also collected approximately 300 GB of Go-pro videos covering a trajectory equivalent to 33 km, and the reconstruction for the ground-view images take 150 k frames (with a resolution of 1500 $\times$ 2000 pixels per frame) out of these videos. **Figure 12** shows the reconstructed point clouds of approximately two thirds

	Acquisition time	Sensor	Off nadir (degree)	Sun elevation angle (degree)	Resolution (meter)	Cloud cover percentage (%)
1	2009-04-01	WorldView-01	1.80	52.40	0.50	0.00
2	2010-04-15	WorldView-01	15.40	58.20	0.52	0.00
3	2010-09-25	WorldView-02	13.00	48.30	0.49	0.04
4	2010-09-25	WorldView-02	19.20	48.30	0.52	0.01
5	2011-10-08	WorldView-02	4.30	43.80	0.47	0.00
6	2012-01-09	WorldView-01	20.00	26.10	0.55	0.00
7	2012-01-09	WorldView-01	32.70	26.20	0.67	0.00
8	2013-08-06	WorldView-02	15.80	64.20	0.50	0.00
9	2013-12-28	WorldView-01	22.90	24.50	0.57	0.00
10	2014-06-06	WorldView-02	23.50	70.80	0.54	0.00
11	2015-04-17	WorldView-02	25.60	56.80	0.56	0.00
12	2019-01-05	WorldView-02	19.90	26.60	0.52	0.00

Table 1.
Twelve overlapping satellite images used for satellite-based 3D reconstruction.

Pair	Intersection angle (degree)	Sun difference angle (degree)	Time difference (days)	Left image ID	Right image ID
1	6.20	0.00	0	3	4
2	12.70	0.10	0	6	7
3	13.60	5.80	379	1	2
4	2.90	1.60	719	6	9
5	9.80	1.70	719	7	9
6	8.70	4.50	378	3	5
7	14.90	4.50	378	4	5
8	7.70	6.60	304	8	10
9	2.10	14.00	315	10	11
10	5.70	30.20	1359	11	12

Table 2.
*Examples of metadata of pairs used for satellite-based 3D reconstruction. These data come in level 1. The image ID refers to those in **Table 1**.*
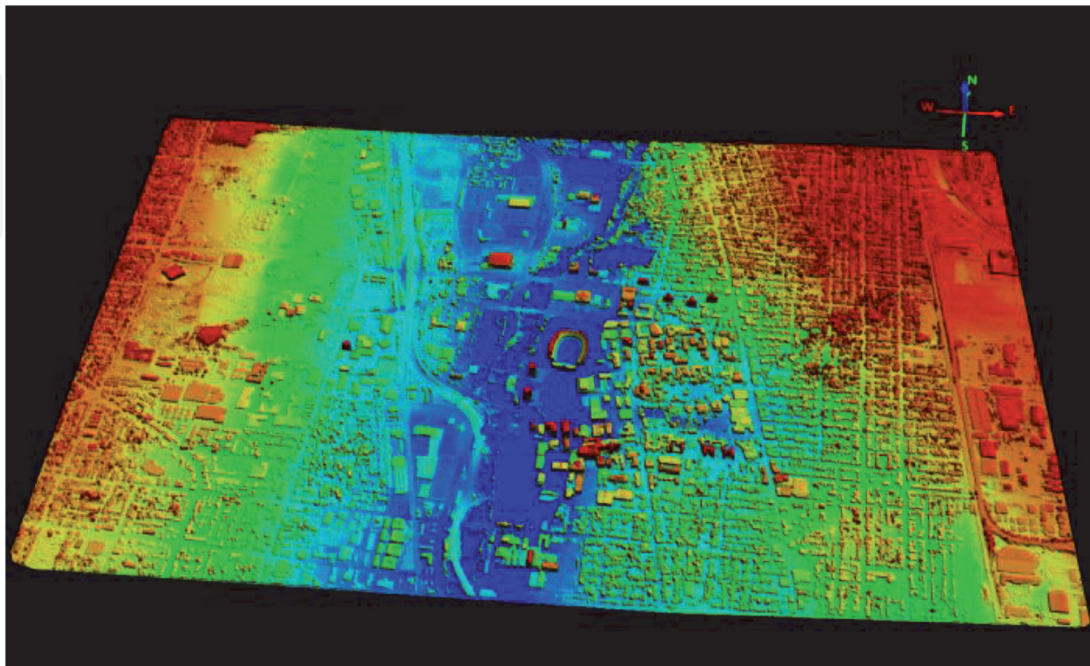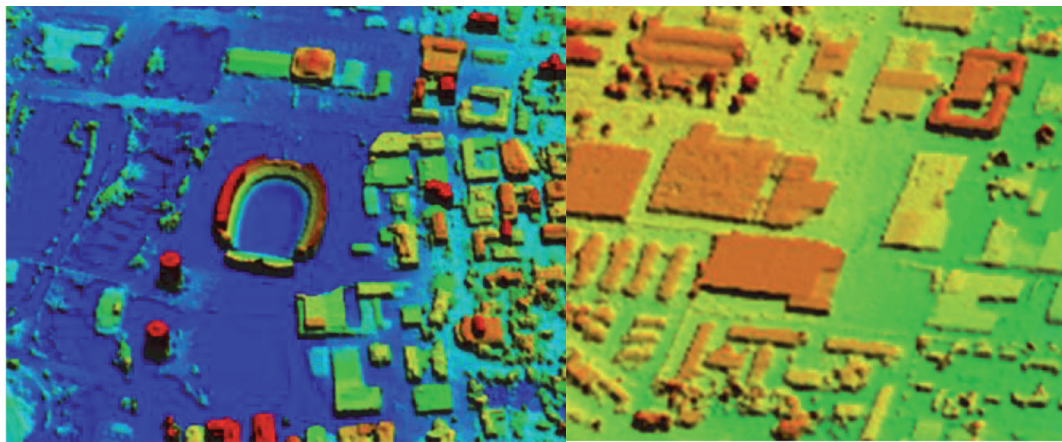

Figure 11.
The generated digital surface models of the OSU campus using our satellite data processing pipeline. The top-row shows enlarged views.

Figure 12.
Dense reconstruction using our processing pipeline for two thirds of the campus region, totaling 7 billion color points.

of the region. The pose estimation time takes approximately 20 hours and dense matching takes 4 h in a normal i-7 desktop computer.

5. Experiment results

We demonstrate that the resulting geometry shows completeness in terms of the rooftop and façade information (for places where ground-view images are available). **Figure 13** provides an overview of the registered point clouds and a comparison showing the mis-registration using a typical point cloud based algorithm [20].

With the registered point clouds, we can generate the meshes using our proposed meshing pipeline introduced in Section 3.4. **Figure 14** shows the reconstructed meshes (shaded and textured) using our pipeline, and we have also included the results from a pure point cloud-based meshing method, which visually demonstrates much worse results. In **Figure 15**, we have also included the reconstruction results of a relatively larger region using our reconstructed pipeline.

5.1 Accuracy evaluation

We have compared the resulting combined model with the ground truth Airborne LiDAR data as shown in **Figure 16**, in which we include two sample areas (top and bottom row of **Figure 16**). Since the airborne LiDAR does not cover the façade information, we evaluate the accuracy of the results using resampled DSM to the same grid. It is expected that the combined model with the incorporated street-view point clouds should have better accuracy given the more accurate point clouds of the (partial) ground and building boundaries. From **Figure 16**, we can observe that the satellite DSM (left column), due to the lower resolution, shows blurred object boundaries, as compared to the combined model (middle column). **Figure 17** plots the error distributions, and it evidences our observations in **Figure 16**: the object boundaries in the satellite DSM show larger errors than the combined model, and it can be also seen in some regions of the ground that the combined model presents less error due to the captured fine ground structures (marked in red circle of **Figures 16** and **17**, bottom row). **Table 3** calculates the RMSE (root mean squared error) of these two areas, and it shows that the combined model improves at 0.20 m in accuracy for

Figure 13.
Registration result of ICP (a) and our method (b) on the distorted ground-view trajectory. (c) Part of the registered ground-view point clouds generated on 150 k Go-Pro images.

area 1 and 1 m for area 2. This shows significant improvement in terms of data accuracy, and we should note that this evaluate is only on the DSM and it is expected that if the façade data evaluation is considered (if ground truth of the façade geometry is available), the accuracy improvement can be significantly more.

6. Conclusion

In this chapter, we propose a framework for fusing results from cross-view images for 3D mesh reconstruction. We present our processing framework (**Figure 1**) that
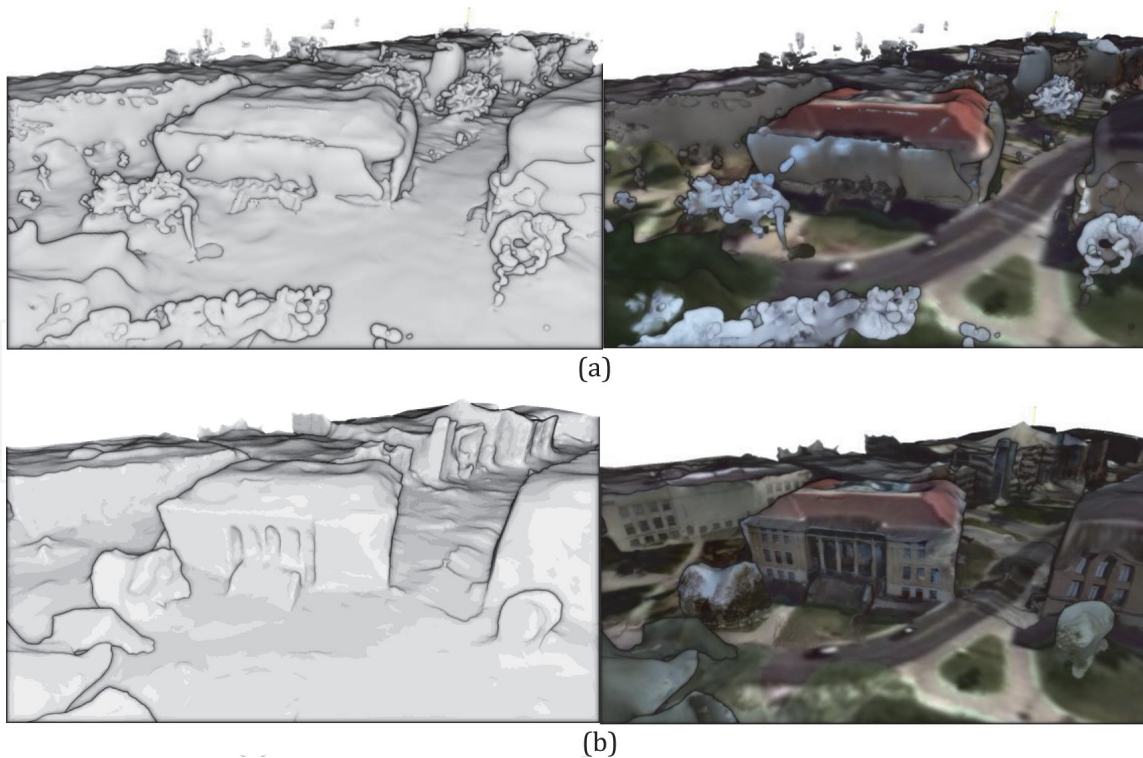
Figure 14.
Left: shaded mesh model. Right: textured mesh model. (a) Reconstructed mesh using Poisson reconstruction.
(b) Reconstructed mesh using our reconstruction method.

Figure 15.
A screenshot of the generated textured mesh of the OSU campus area using our proposed pipeline, which includes information from the top-view and details on the facades.

consists of three major components: (1) 3D reconstruction separately from the top-view satellite images and ground-level images; (2) cross-view geo-registration between the satellite point clouds and ground-view point clouds; 3) meshing reconstruction based on the combined satellite and ground point clouds. In each of these components, we present our developed systems and on-going research efforts in addressing the potential challenges (introduced in Section 1.1) and the in-progress results. We demonstrate that our proposed pipeline can achieve visually more consistent textured meshes, in comparison to a standard and intuitive processing method. The proposed framework and the attempts for integrating satellite and ground-view images and
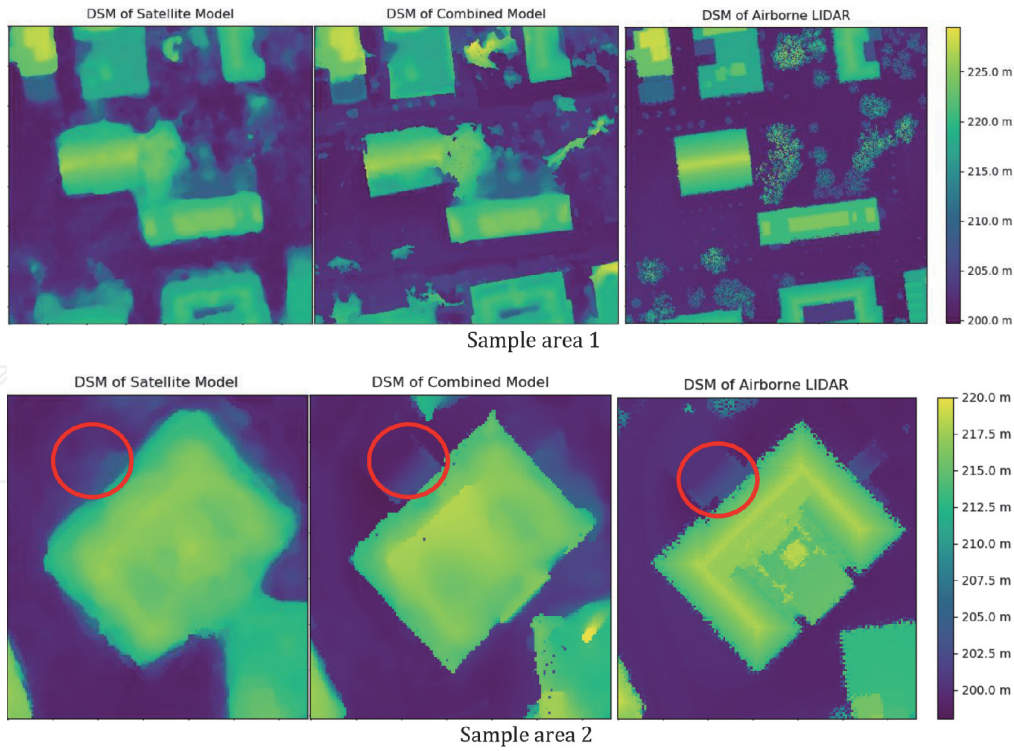
Figure 16.
*DSM from satellite stereo (left column)/combined model (middle column)/airborne LIDAR (right column).
Top and bottom row indicates two difference samples (sample area 1 and sample area 2). The red-circled
region shows that a ground structure is well compared in the combined model, as compared to the satellite DSM.*
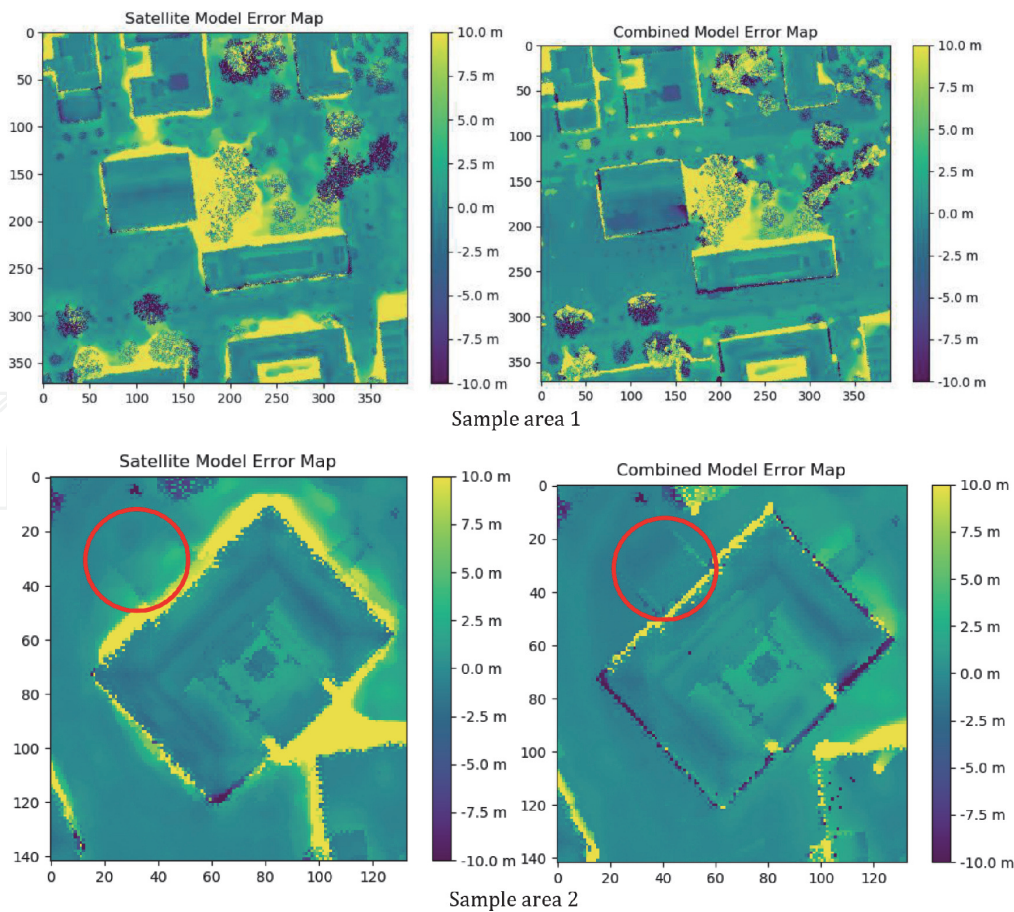

Figure 17.
*Error maps of satellite model (left column) and combined model (right column) evaluated against the LiDAR
DSM. Top and bottom row indicates two difference samples (sample area 1 and sample area 2). The red circled
region shows smaller errors in the combined model due to that the ground structure is well captured.*

	RMSE (m) – Area 1	RMSE (m) – Area 2
Satellite model	4.315	3.505
Combined model	4.138	2.532

Table 3.
Error evaluation.

converting them to textured models can be of particular interest for data collection in areas where standard datasets such as aerial/UAV (unmanned aerial vehicle) photogrammetric/LiDAR flights. We have demonstrated that DSM generated from the combined model using our workflow can be 1-m more accurate than the satellite DSM and is expected to be much more accurate if the evaluation on the façade is considered (as the satellite DSM does not have façade information at all). Our future works include further optimizing individual modules of our processing pipeline and part of these modules will be made available once they are optimized for practical uses.

Acknowledgements

Author details

Rongjun Qin[1,2]*, Shuang Song[1], Xiao Ling[1] and Mostafa Elhashash[2]

1 Department of Civil, Environmental and Geodetic Engineering, The Ohio State University, Columbus, Ohio, USA

2 Department of Electrical and Computer Engineering, The Ohio State University, USA

*Address all correspondence to: qin.324@osu.edu

IntechOpen

References

[1] Haala N, Cavegn S. High density aerial image matching: State-of-the-art and future prospects. In: International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences. Vol. 41. Netherlands: Copernicus Publications; 2016

[2] Schwarz B. LIDAR: Mapping the world in 3D. Nature Photonics. 2010;**4**:429

[3] Bosch M, Kurtz Z, Hagstrom S, Brown M. A multiple view stereo benchmark for satellite imagery. In: Presented at the Proceedings of the IEEE Applied Imagery Pattern Recognition (AIPR) Workshop, October 2016. 2016

[4] Qin R. RPC stereo processor (RSP) –a software package for digital surface model and orthophoto generation from satellite stereo imagery. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. III. Netherlands: Copernicus Publications; 2016. pp. 77-82

[5] Qin R, Song S, Huang X. 3D data generation using low-cost cross-view images. In: Presented at the the International Archives of Photogrammetry and Remote Sensing. ISPRS Congress 2020 (Delayed to 2021 Due to Coronavirus), Nice, France. 2020

[6] Regmi K, Borji A. Cross-view image synthesis using conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. pp. 3501-3510

[7] Lu X, Li Z, Cui Z, Oswald MR, Pollefeys M, Qin R. Geometry-aware satellite-to-ground image synthesis for urban areas. In: Presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2020

[8] Gruen A, Huang X, Qin R, Du T, Fang W, Boavida J, et al. Joint processing of Uav imagery and terrestrial Mobile mapping system data for very high Resolution City Modeling. In: ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 1. Netherlands: Copernicus Publications; 2013. pp. 175-182

[9] Lin T-Y, Cui Y, Belongie S, Hays J. Learning deep representations for ground-to-aerial geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 5007-5015

[10] Kwan C, Chou B, Ayhan B. Enhancing stereo image formation and depth map estimation for Mastcam images. In: 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). 2018. pp. 566-572

[11] Qin R, Kwan C, Ayhan B. Generation of stereo images for Mastcam imagers. In: Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXVI. Bellingham, Washington, USA: SPIE; 2020. p. 1139207

[12] Ayhan B, Kwan C. Mastcam image resolution enhancement with application to disparity map generation for stereo images with different resolutions. Sensors. 2019;**19**:3526

[13] Boyle R. NASA Uses Microsoft's HoloLens and ProtoSpace to Build its Next Mars Rover in Augmented Reality. Seattle, Washington, USA: GeekWire; 2018. Available from: https://www.geekwire.com/2016/nasa-uses-microsoft-hololens-build-mars-rover-augmented-reality/

[14] Kwan C, Chou B, Ayhan B. Stereo image and depth map generation for images with different views and resolutions. In: 2018 9th IEEE Annual

Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). 2018. pp. 573-579

[15] Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW. Bundle adjustment—A modern synthesis. In: Vision Algorithms: Theory and Practice. Springer; 2000. pp. 298-372

[16] Lin T-Y, Belongie S, Hays J. Cross-view image geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013. pp. 891-898

[17] Tian Y, Chen C, Shah M. Cross-view image matching for geo-localization in urban environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 3608-3616

[18] Castaldo F, Zamir A, Angst R, Palmieri F, Savarese S. Semantic cross-view matching. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015. pp. 9-17

[19] Gruen A, Akca D. Least squares 3D surface and curve matching. ISPRS Journal of Photogrammetry and Remote Sensing. 2005;**59**:151-174

[20] Rusinkiewicz S, Levoy M. "efficient variants of the ICP algorithm," in 3-D digital imaging and Modeling. In: Proceedings. Third International Conference on, 2001. 2001. pp. 145-152

[21] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. 2006

[22] Tran S, Davis L. 3D surface reconstruction using graph cuts with surface constraints. In: European Conference on Computer Vision. 2006. pp. 219-231

[23] Labatut P, Pons JP, Keriven R. Robust and efficient surface reconstruction from range data. In: Computer Graphics Forum. Hoboken, New Jersey, US: Wiley; 2009. pp. 2275-2290

[24] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. New York, US: IEEE; 2013. pp. 2100-2106

[25] Waechter M, Moehrle N, Goesele M. Let there be color! Large-scale texturing of 3D reconstructions. In: European Conference on Computer Vision. 2014. pp. 836-850

[26] Qin R. Automated 3D recovery from very high resolution multi-view satellite images. In: ASPRS (IGTF) Annual Conference, March 12–16, Baltimore, Maryland, USA. 2017. p. 10

[27] Qin R. RPC stereo processor (RSP) – a software package for digital surface model and orthophoto generation from satellite stereo imagery. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. (to Appear in ISPRS Congress July 2016). 2016

[28] Qin R. A critical analysis of satellite stereo pairs for digital surface model generation and a matching quality prediction model. ISPRS Journal of Photogrammetry and Remote Sensing. 2019;**154**:139-150

[29] Hirschmüller H. Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008;**30**:328-341

[30] Qin R. Change detection on LOD 2 building models with very high resolution spaceborne stereo imagery. ISPRS Journal of Photogrammetry and Remote Sensing. 2014;**96**:179-192

[31] N. Snavely, "Bundler: Structure from Motion (SFM) for Unordered Image Collections," Available online: phototour. cs. washington. edu/bundler/ (accessed on 12 July 2013), 2010

[32] Snavely N, Seitz SM, Szeliski R. Skeletal graphs for efficient structure from motion. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. 2008. pp. 1-8

[33] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004;**60**:91-110

[34] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. In: Computer Vision–ECCV 2006. New York, US: Springer; 2006. pp. 404-417

[35] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision. 2011. pp. 2564-2571

[36] Mur-Artal R, Tardós JD. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics. 2017;**33**: 1255-1262

[37] Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. 1981;**24**: 381-395

[38] Förstner W, Wrobel BP. Photogrammetric Computer Vision. 1st ed. New York, US: Springer International Publishing; 2016

[39] Nocedal J, Wright S. Numerical Optimization. New York, US: Springer Science & Business Media; 2006

[40] Gruen A, Beyer HA. System calibration through self-calibration. In: Gruen TSHA, editor. Calibration and

Orientation of Cameras in Computer Vision. Vol. 34. New York, US: Springer; 2001. -163, 193

[41] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem. International Journal of Computer Vision. 2009;**81**:155

[42] Qin R, Fang W. A hierarchical building detection method for very high resolution remotely sensed images combined with DSM using graph cut optimization. Photogrammetry Engineering and Remote Sensing. 2014; **80**:37-48

[43] Vincent L. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing. 1993;**2**:176-201

[44] Carlson TN, Ripley DA. On the relation between NDVI, fractional vegetation cover, and leaf area index. Remote Sensing of Environment. 1997; **62**:241-252

[45] Tremeau A, Borel N. A region growing and merging algorithm to color segmentation. Pattern Recognition. 1997;**30**:1191-1203

[46] Fabbri R, Costa LDF, Torelli JC, Bruno OM. 2D Euclidean distance transform algorithms: A comparative survey. ACM Computing Surveys (CSUR). 2008;**40**:1-44

[47] Meijster A, Roerdink JB, Hesselink WH. A general algorithm for computing distance transforms in linear time. In: Mathematical Morphology and its Applications to Image and Signal Processing. New York, US: Springer; 2002. pp. 331-340

[48] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001;**23**:1222-1239

[49] Orlin JB. Max flows in O (nm) time, or better. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. 2013. pp. 765-774

[50] S. Clark. (2020. The Surface Grower Algorithm. Available from: http://www.cs.carleton.edu/cs_comps/0405/shape/surface_grower.html

[51] Van Kreveld M, Schwarzkopf O, de Berg M, Overmars M. Computational Geometry Algorithms and Applications. New York, US: Springer; 2000

[52] Fabri A, Pion S. CGAL: The computational geometry algorithms library. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2009. pp. 538-539

[53] D. Cernea, OpenMVS: Open Multiple View Stereovision, 2015. Available from: https://openmvg.readthedocs.io/en/latest/software/MVS/OpenMVS/

[54] Goldberg AV, Hed S, Kaplan H, Tarjan RE, Werneck RF. Maximum flows by incremental breadth-first search. In: European Symposium on Algorithms. 2011. pp. 457-468

[55] Lempitsky V, Boykov Y, Ivanov D. Oriented visibility for multiview reconstruction. In: European Conference on Computer Vision. 2006. pp. 226-238

[56] Pérez P, Gangnet M, Blake A. Poisson image editing. In: ACM Transactions on Graphics (TOG). Vol. 22. New York, US: ACM Publications; 2003. pp. 313-318