remote sensing

MDPI

Article

Real-Time Orthophoto Mosaicing on Mobile Devices for Sequential Aerial Images with Low Overlap

Yong Zhao [1], Yuqi Cheng [1], Xishan Zhang [2], Shibiao Xu [3,*], Shuhui Bu [1], Hongkai Jiang [1], Pengcheng Han [1], Ke Li [4] and Gang Wan [5]

1 Institute of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China; zdzhaoyong@mail.nwpu.edu.cn (Y.Z.); chengyuqi@mail.nwpu.edu.cn (Y.C.); bushuhui@nwpu.edu.cn (S.B.); jianghk@nwpu.edu.cn (H.J.); hanpc1125@mail.nwpu.edu.cn (P.H.)
2 Institute of Mechanical Technology, Xi'an 710043, China; xishan.zhang@outlook.com
3 Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
4 Zhengzhou Institute of Surveying and Mapping, Zhengzhou 450052, China; like19771223@163.com
5 Institute of Aeronautics, Aerospace Engineering University, Beijing 101416, China; casper_51@163.com
* Correspondence: shibiao.xu@nlpr.ia.ac.cn

Received: 21 October 2020; Accepted: 6 November 2020; Published: 13 November 2020

Abstract: Orthophoto generation is a popular topic in aerial photogrammetry and 3D reconstruction. It is generally computationally expensive with large memory consumption. Inspired by the simultaneous localization and mapping (SLAM) workflow, this paper presents an online sequential orthophoto mosaicing solution for large baseline high-resolution aerial images with high efficiency and novel precision. An appearance and spatial correlation-constrained fast low-overlap neighbor candidate query and matching strategy is used for efficient and robust global matching. Instead of estimating 3D positions of sparse mappoints, which is outlier sensitive, we propose to describe the ground reconstruction with multiple stitching planes, where parameters are reduced for fast nonconvex graph optimization. GPS information is also fused along with six degrees of freedom (6-DOF) pose estimation, which not only provides georeferenced coordinates, but also converges property and robustness. An incremental orthophoto is generated by fusing the latest images with adaptive weighted multiband algorithm, and all results are tiled with level of detail (LoD) support for efficient rendering and further disk cache for reducing memory usages. Public datasets are evaluated by comparing state-of-the-art software. Results show that our system outputs orthophoto with novel efficiency, quality, and robustness in real-time. An android commercial application is developed for online stitching with DJIdrones, considering the excellent performance of our algorithm.

Keywords: aerial images; DOM; low overlap; mosaicing; georeferenced; orthophoto

1. Introduction

In recent years, aerial image mosaicing has been used in many scenes, such as farmland mosaicing, forest fire detection, post-disaster relief, and military reconnaissance. Generally, the task of aerial image mosaicing can be implemented in two ways. First is offline mosaicing [1–3], where the mosaicing process is usually applied after obtaining all the image data of the target area with the unmanned aerial vehicles (UAVs). This approach could provide integrated information for image mapping. Thus, the mosaicing precision is generally accurate. On the basis of the estimated camera poses, the second method is online mosaicing, which stitches images in real-time [4–6]. This approach is necessary in some specific application scenarios such as live map visualization through virtual reality [7–9].

In general, the major difference between online mosaicing and offline mosaicing is the core technique in estimating the camera pose and the 3D point cloud from images. SLAM [10–13]

and structure from motion (SfM) [14,15] are separately used to obtain the pose and point cloud. Then, the homography transformation method is used to project the image to the correct position; finally, these images are fused. In some special scenarios, such as post-disaster rescue and military reconnaissance, we aim to mosaic the map in real time. Therefore, our purpose is to improve the accuracy of online mosaicing by designing a novel framework and algorithm.

SfM methods [14–17] are fundamentally offline in nature with expensive computation cost, and processing low overlap images using SLAM methods [4] is still very difficult. Thus, this study aims to develop an online incremental stitching method, which is extremely efficient and robust, for mobile devices. While most SfM and SLAM systems use bundle adjustment for camera poses and landmark position refinement, it is still computationally expensive for an online system and insufficiently robust against low overlap and poor matching results. To solve these problems, we proposed a planar restricted pose graph optimization method to estimate camera poses and fuse orthophotos simultaneously. This approach can work effectively in the low overlap scenario and accelerate the calculations to achieve real-time mosaicing in embedded devices as demonstrated in Figure 1. Therefore, the proposed graph optimization not only brings higher efficiency, but also converges property and robustness. In summary, the main contributions are as follows:

1. A novel online georeferenced orthophoto mosaicing framework with high efficiency and robustness: Compared with the existing commercial software and current state-of-the-art mosaicing systems, our method proposes a complete solution for real-time incremental stitching on mobile devices. Considerable improvements are considered for robustness and efficiency to adapt to the challenging requirements of high-quality orthoimage generation with relatively fast speed and less computation.
2. Planar restricted online pose graph optimization: A planar-restricted global pose graph optimization algorithm is proposed and compared with other 2D aerial image-mosaicing schemes and traditional SLAM or SfM systems. Instead of using sparse 3D map points which is outlier sensitive, keypoint matches are parameterized to planar restricted reprojection errors where parameters are effectively reduced. This method can achieve better robustness and efficiency, even if the overlap rate is low.
3. An adapted weighted multiband images fusion algorithm with LoD based tiling, caching and rendering Memory resources in mobile devices are often limited. Thus, retaining the entire mosaic for large-scale datasets is impossible. In addition, the display system demands a tiled DOM for efficient rendering. To solve these problems, the orthophoto consists of several image tiles, which are managed with a hashed least recently used (LRU) cache; the LoD technique is also used for quick rendering.
4. An android application demonstrating algorithm effectiveness on mobile devices: To show the realistic performance in cellphones, an android software is designed to upload flight mission for DJI drones. In addition, we integrate the presented algorithm by providing restful web service through C++.

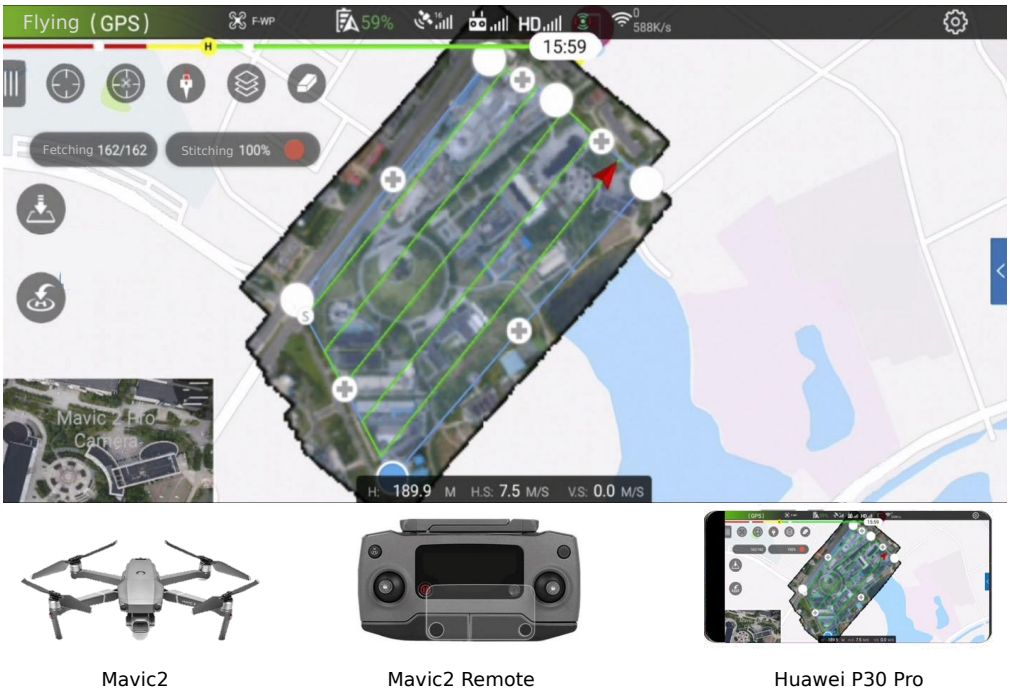

Figure 1. In order to demonstrate our proposed algorithm's effectiveness on mobile devices, we implemented an android application. This screenshot demonstrates a live map during the flight where both flight plan and online stitching are performed on a cellphone. Although Mavic2 and Huawei P30 Pro are used, our algorithm is not limited to these devices. As we known, this is the first android application that supports online georeferenced large-scale low-overlap aerial images stitching. A demonstration video can be found on YouTube website: https://youtu.be/BmjtEte8sgo.

2. Related Work

In recent years, a large number of methods have been proposed for image mosaicking [5,6,18,19], which takes advantage of SfM and SLAM technologies. The representative works are summarized as follows.

To achieve high-quality image mosaicing, the mature option is SfM, which is designed to explore the most information and reconstruct a metric model for offline orthoimage generation from unordered multiview images. Over the years, various SfM methods have been proposed, including incremental [14,15], global [16,17] and hierarchical approach [20]. Incremental SfM is the most popular strategy to reconstruct 3D images from unordered images; it is also used for the basic technology of orthophoto generation [1,3]. The typical work of SfM based offline mosaicing is [21]. This method optimizes the pose by generating a 3D point cloud, identifying the ground control points, and finally mosaicing all the prepared images simultaneously. SfM-based methods always take hours to generate the final orthoimage and all images required prior to computation. However, they are unsuitable for real-time and incremental usage, and they usually require at least trifocal overlap to capture cameras and reconstruct a scene [22]. In [2], a fast offline georeferenced aerial image stitching method is proposed with high efficiency, where a planar constrained global SfM is used for pose graph optimization. This method has great potential to achieve real-time ability on personal computers.

To estimate camera pose for image mosaicing in real-time, a SLAM-based method is a better choice. The typical work of SLAM-based online mosaicing is [5]. This method uses bag of words (BoW) to obtain images that are partially overlapped. Then, it calculates the perspective transformations between the images and eliminates the outliers using BaySAC [23] instead of RANSAC [24]. Finally, Dijk-stras algorithm [25] is adopted to to determine the seam, which represents the minimized difference between the two images. In the state-of-the-art Map2Dfusion [4], GPS information is integrated into SLAM by synchronizing GPS information with video streaming time, which conquers accumulated drift. Then an

adaptive weighted pyramid is used for image fusion to generate a mosaicing. Lati et al. [18] propose a novel technique based on fuzzy clustering to separate outliers, and a bilinear interpolation algorithm is used in the image fusion process. A multi-threaded architecture-based image mosaicing method [26] using incremental bags of binary words was proposed to speed up the mosaicing process. In [27], sparse BA was used for pose optimization to accelerate the SLAM algorithm. MGRAPH [28] proposes a multigraph homography method to generate incremental mosaics in real-time. Although they could easily further integrate the algorithm in the embedded devices, the accuracy is poor. In summary, these traditional SLAM-based methods should estimate camera pose and calculate the 3D point cloud simultaneously; this approach is computationally expensive. These methods use many tricks to improve the efficiency of the computation. However, they are still difficult to implement in embedded devices, which have low computation resources. GSLAM [29] provides a general SLAM framework and benchMark, which may be used to develop efficient SLAM implementation and publish it as a plugin of embedded devices.

UAVs generally fly at a relatively high position for capturing aerial images; thus, many scenes could be assumed as planes. Our goal is to design a planar optimization method to reduce the computation cost, thereby accelerating the estimation of camera pose and achieving real-time mosaicing in embedded devices.

3. Methodology

3.1. Proposed Framework

The final deployment of our system is an android application and the architectural is illustrated in Figure 2. To maintain the portability of our algorithm to other mobile operating systems, a restful HTTP API is designed instead of Java Native Interface (JNI) to minimize coupling between UI and stitching algorithm implemented with C++. Sequential images with GPS information are obtained through DJI mobile SDK (https://developer.dji.com/mobile-sdk) and posted to the algorithm during flight mission.

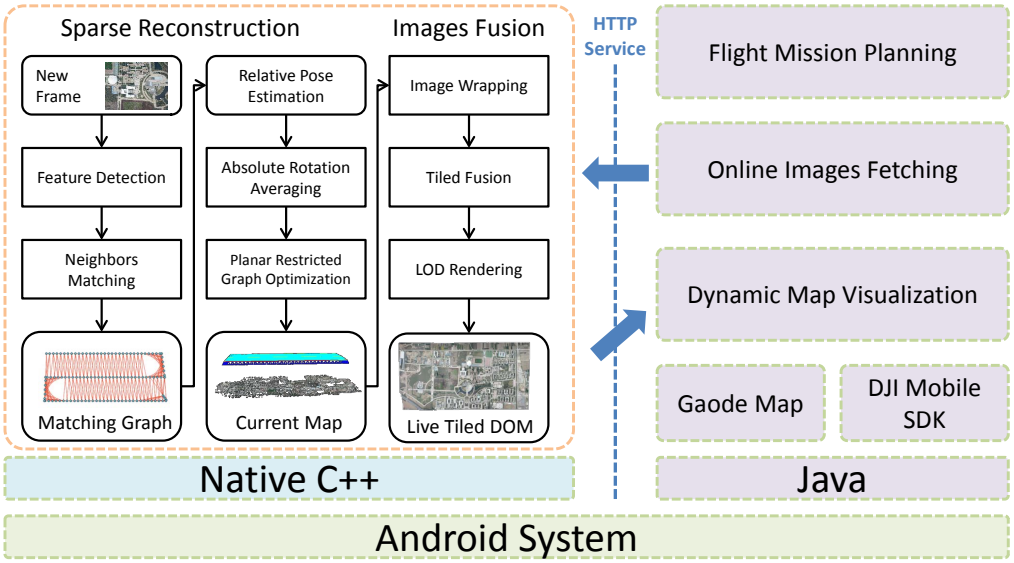

Figure 2. The framework of our system on android devices. All stitching algorithm pipelines are written by C++, which runs in the linux native layer, and the Java-based user interface (UI) calls live mosaic service through HTTP API. A way-point flight mission is planned and uploaded through DJI Mobile SDK, and images are fetched online with GPS information. The mosaic algorithm receives these images and stitches them incrementally during the flight. In addition, a tiled dynamic map layer is served though HTTP interface and visualized with GaoDe Map SDK.

When the stitching algorithm receives the online images, keypoint features are extracted and matched to update the matching graph with an appearance and spatial-based fast neighbor query and matching algorithm. Planes instead of map points are used to represent the reconstruction; thus, a PnP-based tracking strategy is unsuitable for initial pose estimation. An absolute rotation averaging algorithm is performed to obtain the initial pose information by jointly considering GPS and relative two-view pose constraints. The final reconstruction and camera poses are updated through our proposed local planar restricted pose graph optimization method, where high robustness and efficiency are achieved. The latest frame with pose and stitching plane information is then published for further orthophoto fusion.

Inspired by the novel SLAM-based aerial mapping system Map2DFusion [4], an adaptive weighted multiband rendering algorithm is used to fuse images incrementally. To further save memory budget and accelerate mosaic rendering, the final orthophoto is represented with an LRU cached image tiles segmented by Mercator projection [30], and LoD algorithm is considered to update the multiple-level tiles. When the real-time orthophoto layer is updated, a notification is sent to the UI to refresh the display.

3.2. Appearance and Spatial Based Fast Neighbors Query and Matching

Similar to traditional SfM and SLAM methods, the keypoint correspondences between images are the core inputs for sparse reconstruction in our method. A large number of feature descriptors are proposed for image matching, including Sift, SURF, BREIF, ORB [31], and AKAZE, where ORB is popularly used in SLAM systems due to its high efficiency, and SIFT is used by most commercial software and open-source SfM projects due to its good quality and robustness. ORB is unsuitable to process low overlap images; thus, we adopt the SIFT descriptor with a shader GPU-accelerated implementation, which is totally adequately fast for our task. To reduce the computation to only approximately 1000 keypoints, an appearance joint spatial-based fast neighbor query and matching strategy is used to explore most good matches. Through our matching strategy and reconstruction algorithm, these few keypoints are sufficient for most sequences and our evaluations show that increasing the keypoint number does not effectively enhance stitching quality.

For every image with GPS information, the keypoints and descriptors are extracted, and the frame is indexed by geometry position with k-nearest neighbor algorithm implemented by FLANN [32]. Some studies show that a square root (Hellinger) kernel, instead of the standard Euclidean distance, provides better performance for SIFT descriptor similarity measurement. A transformation is performed initially to map the descriptors from the original SIFT space to the RootSIFT space to improve retrieval and matching quality. Then, neighbor image query is performed using geometry information by the k-nearest algorithm. In addition, for every neighbor, we perform a fast global matching through the following pipeline:

1. **BoW-accelerated global correspondences with cross check.** A BoW vocabulary is pretrained with k-means algorithm because brute force matching is highly computationally expensive, and features are transformed to some small spaces to accelerate global searching by matching each space separately. This strategy is adopted to obtain an initial matching, and the accelerated version of DBoW implemented by GSLAM [29] is used in this work.
2. **Outliers filtering based on epipolar geometry and matching angle histogram.** The initial matching contains outliers. A simple histogram-based voting filtering is performed to increase inlier rate. In addition, a fundamental matrix is estimated with the random sample consensus [24] procedure for geometric outlier removal.
3. **Multi-homography-based rematching.** The previous procedures may ignore some good matches. Thus, we try to find them again with a multihomography assumption. The previous inlier matches are used to calculate multiple homography matrices. Then, window searching is performed for every feature without match to find the remaining potential matches. The distance between

matched keypoint and epipolar line is computed, and the match is only accepted when the distance is below the fixed threshold.

The proposed method is very efficient, and the experiments show that only approximately less than 5 milliseconds is required to match an image pair in one thread and can obtain numerous reliable matches. The matches between images form an incremental matching graph, which can be used for the following online sparse reconstruction.

3.3. Online Planar Restricted Sparse Reconstruction

PnP-based camera pose tracking is popularly used in traditional SLAM systems. However, it is unsuitable in our system because no map points are estimated. In addition, generally, a PnP constraint requires the landmarks of at least three images. This condition is not satisfied when the overlap is below 2/3. To obtain a good initial pose estimation for the following graph optimization, recent novel global SfM methods have explored relative pose graph technic, name rotation, and translation averaging. This algorithm is also used in SLAM system [33] for decoupling the rotation and translation estimation, showing high robustness. GPS information is always available for our system; thus, we further fuse the absolute geometry location information in the rotation averaging step to obtain better robustness. This algorithm is called the absolute rotation averaging algorithm.

For each frame, we compute the local Cartesian coordinates $\mathbf{g}_i$ from the GPS information. For the latest image i, two-view reconstructions are performed to obtain relative pose relationships against its neighbors, denoted as $\mathbf{t}_{ij}$ and $\mathbf{R}_{ij}$. The length of direction $\mathbf{t}_{ij}$, which is also called the scale s_{ij} , can be estimated with the absolute distance of GPS:

$$s_{ij} = ||\mathbf{g}_i - \mathbf{g}_j||. \tag{1}$$

The absolute offset between two images in the GPS coordinate is known and forms the following equation:

$$s_{ij}\mathbf{R}_i\mathbf{t}_{ij} = \mathbf{g}_{ij} = \mathbf{g}_j - \mathbf{g}_i. \tag{2}$$

where $\mathbf{R}_i$ is the absolute rotation of current frame i. When the DOF of $\mathbf{R}_i$ is 3 and this equation provides two constraints, $\mathbf{R}_i$ can be calculated by maximizing the following expression when more than two noncollinear connections exist:

$$\hat{\mathbf{R}}_i = \arg\max_{\mathbf{R}_i} \sum_j (s_{ij}\mathbf{R}_i\mathbf{t}_{ij}) \cdot \mathbf{g}_{ij} \tag{3}$$

Although orthonormal matrices with positive determinant are easy to understand and popular used for rotation representation, it is not suitable for estimation since the over parameterization. While unit quaternions constitute an elegant representation for rotation, to find the solution for $\mathbf{R}_i$, the expression can be rewritten in quaternion form as follows:

$$\hat{\mathbf{q}}_i = \arg\max_{\mathbf{q}_i} \sum_j (\mathbf{q}_i\mathbf{r}_{ij}\mathbf{q}_i^*) \cdot \hat{\mathbf{r}}_{ij} \tag{4}$$

where $\mathbf{q}_i$ is the unit quaternion form of $\mathbf{R}_i$, $\mathbf{r}_{ij}$ is the quaternion form of $s_{ij}\mathbf{t}_{ij}$, and $\hat{\mathbf{r}}_{ij}$ is the quaternion form of $\mathbf{g}_{ij}$. To solve this expression, ref. [34] provides a closed form solution by computing the eigenvector of a sum product matrix N. However, those estimations may still contain outliers, so that we further use the relative pose pairs of current frame for robust pose propagation. Traditional nonlinear optimization uses Gauss Newton or Levenberg-Marquarelt (LM) algorithm, which is sensitive to noise. Modern Global SfM systems use modern L1 optimizer to carry out robust averaging of relative rotations that is efficient, scalable and robust to outliers. We jointly considered all relative

constraints from two-view reconstructions and absolute prior to the above equation, and the L1-based rotation and translation averaging algorithm [35] is used to estimate the initial pose robustly.

Once the image initial pose is available, a local planar restricted pose graph optimization is considered to further adjust the submap consisting of the latest frame with its neighbors. Different from the traditional bundle adjustment, our graph optimization is triangulation-free, and reprojection errors are not projected from the map points but from original feature matches. GPS prior is also fused for convergence and robustness. For image i, we denote its pose and height with a seven-DOF parameter, $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i, h)$. The local optimization aims to optimize the pose and height information in the submap $\mathbf{S}$ by joint consideration of image matches and GPS priors, as follows:

$$\hat{\mathbf{T}} = \arg\min \sum_{i,j \in \mathbf{S}} \sum_{m} \xi_m(\mathbf{T}_i, \mathbf{T}_j) + \sum_{i \in \mathbf{S}} \tau(\mathbf{T}_i, \mathbf{g}_i), \tag{5}$$

where $\tau(\mathbf{T}_i, \mathbf{g}_i) = \mathbf{t}_i - \mathbf{g}_i$ is the GPS prior factor, while $\xi_m(\mathbf{T}_i, \mathbf{T}_j)$ is the reprojection error of match $m(\mathbf{u}_i, \mathbf{u}_j)$ between frames i and j, which is defined as:

$$\mathbf{a} = \mathbf{R}_i \cdot Proj^{-1}(\mathbf{u}_i), \tag{6}$$

$$\mathbf{p} = \mathbf{t}_i + \frac{(h_i + h_j)/2 - \mathbf{t}_i^z}{\alpha^z}\mathbf{a}, \tag{7}$$

$$\xi_m(\mathbf{T}_i, \mathbf{T}_j) = Proj(\mathbf{R}_j^{-1} \cdot (\mathbf{p} - \mathbf{t}_j)) - \mathbf{u}_j. \tag{8}$$

Here, $\mathbf{a}$ is the unit vector presents the view direction of keypoint $\mathbf{u}_i$ and p is the 3D location of this keypoint. h_i, h_j are the local ground height estimation of image i, j. And we denote $Proj$ as the pinhole projection in camera coordinates, while $Proj^{-1}$ is the inverse projection in $z = 1$ plane:

$$\mathbf{u} = Proj(\mathbf{p}) = \left(\frac{x * f_x}{z} + c_x, \frac{y * f_y}{z} + c_y\right)^T, \tag{9}$$

$$\hat{\mathbf{p}} = Proj^{-1}(\mathbf{u}) = \left(\frac{x - c_x}{f_x}, \frac{y - c_y}{f_y}, 1\right)^T. \tag{10}$$

To solve this graph optimization problem, the popular open-source Ceres Solver library (http://ceres-solver.org) is used in our implementation. To visualize the reconstruction, the keypoints can further project to the stitching plane and form tracks used in traditional SLAM and SfM methods.

3.4. Georeferenced Images Fusion with Tiling and LoD

The preview sparse reconstruction updates the current map and publishes the latest frame for incremental orthophoto stitching. Our online stitching and rendering pipeline encounter several challenges, as follows:

1. The fusing should be efficient for real-time processing. The computational expensive offline view selection and seam finding methods based on graph cut are unsuitable here.
2. The mosaic result should be rendered efficiently. Publishing the entire image frequently is impossible because we use network for the dynamic orthophoto publishing. The map rendering engines often require tiled images with LoD support, and only tiles that are visible and updated should be refreshed.
3. The processing should be memory efficient to run on mobile devices. After fusing hundreds of images, the final mosaic could be very large, and the memory resource in mobile devices is very limited. An efficient caching algorithm should be considered only to hold active data.
4. The final mosaic should be as ortho and smooth as possible. The stitching is not fully ortho because no 3D dense reconstruction procedure is considered in the entire pipeline for efficiency. However, we can still preserve the view, and the blending method should smooth the seam lines to obtain a natural result.

The estimated plane is used to wrap the original image to a local DOM using perspective reprojection. An adaptive weighted multiband algorithm [4] is used to fuse the aligned images. The final orthophoto consists of several Mercator projection tiles required by the map display engine, and low-level pyramid tiles with LoD support is managed for immediate rendering. To solve these problems, we hold the mosaic with a thread-safe hashed LRU cache to save memory budget and launched a thread pool for further speed acceleration.

3.5. Mobile Application with Flight Planning and Live Map

The DJI drones provide mobile Android and IOS SDK. Thus, controlling it through smartphones and tablets is possible. When the algorithm is tested in x86 system, we attempt to transplant the stitching implementation to mobile devices. A network API interface based on HTTP protocol is designed for better portability, and Android platform is currently selected. However, iPhone or iPad devices should be used for easy implementation. We use DJI UX SDK to visualize the first person view (FPV) video and drone status, and GaoDe SDK is used for satellite map visualization, way-point mission planning, dynamic orthophoto displaying, and interaction. When users select a survey area and flight height, a way-point mission is automatically generated considering fly velocity, forward, and sideward overlap rate. The mission is checked and uploaded to the connected drone for autonomous surveying. Meanwhile, images taken by the drone are fetched through SDK during the flight. Images are then posted to the stitching SDK through HTTP, and the incremental orthophoto is updated after the image is processed with simultaneous refreshing of display.

4. Experiments

In this section, we focus on the stitching algorithm evaluation with qualitative and quantitative analyses. First, we perform a full test on the public DroneMap2 dataset (http://zhaoyong.adv-ci.com/npu-dronemap2-dataset) with more than 20 different sequences captured in countryside and cities. Second, we compare our live orthophoto without post processing to the state-of-the-art software Pix4DMapper and DJITerra. Third, we perform quantitative precision evaluation on a dataset with GCP check points and illustrate the detailed absolute errors. Finally, we performed statistic computation of our system on x86 PC and mobile arm devices with comparison to show the efficiency. To ensure fairness, all x86 experiments are performed on a computer with Intel i7-6700 CPU, 16 GB RAM, and Nvidia GTX 1060 GPU. Furthermore, a cloud-based processing is demonstrated to show the future scalability and cooperativity of our system.

4.1. Results on DroneMap2 Dataset

To evaluate the adaptiveness of our algorithm to different scenarios, the public DroneMap2 dataset consisting of more than 20 different sequences is used for aerial mapping evaluation. Our algorithm can process all sequences with acceptable quality by using the easy-to-converge design. Some overview results with highlight of detailed screenshots are demonstrated in Figure 3. Under local planar assumption, our method still shows high robustness to buildings or even mountains in sequences, such as *mavic-campus* and *phantom4-mountain*.

However, the result is imperfect due to the incremental style. Few mismatches are also highlighted in the details and can be divided into three categories, as follows:

1. Seam-line cutting of moving objects: Traditional seam finding methods are unsuitable here due to the incremental stitching style. In addition, for efficiency, the seam-lines are automatically determined by the adaptive weighted blending. The stitching seam-line may cut the moving objects, such as cars. Thus, half of the cars are rendered, as illustrated in the highlights of Figure 3a,d,n.
2. Live reconstruction drift between airlines: The stitched result is difficult to adjust because the algorithm renders orthophoto lively. However, even with fused GPS information, the sparse

reconstruction and pose estimation contain small drift and are updated after more observations, which may cause mismatches, as shown in Figure 3 for sequence *mavic-factory* and *mavic-warriors*.

3. Homography mismatch caused by high buildings: The sparse reconstruction and fusion steps assume that the ground is a local planar. Thus, homography projection is used to wrap original image to the stitching plane. For high buildings, mismatches may be observed, as illustrated in Figure 3b.

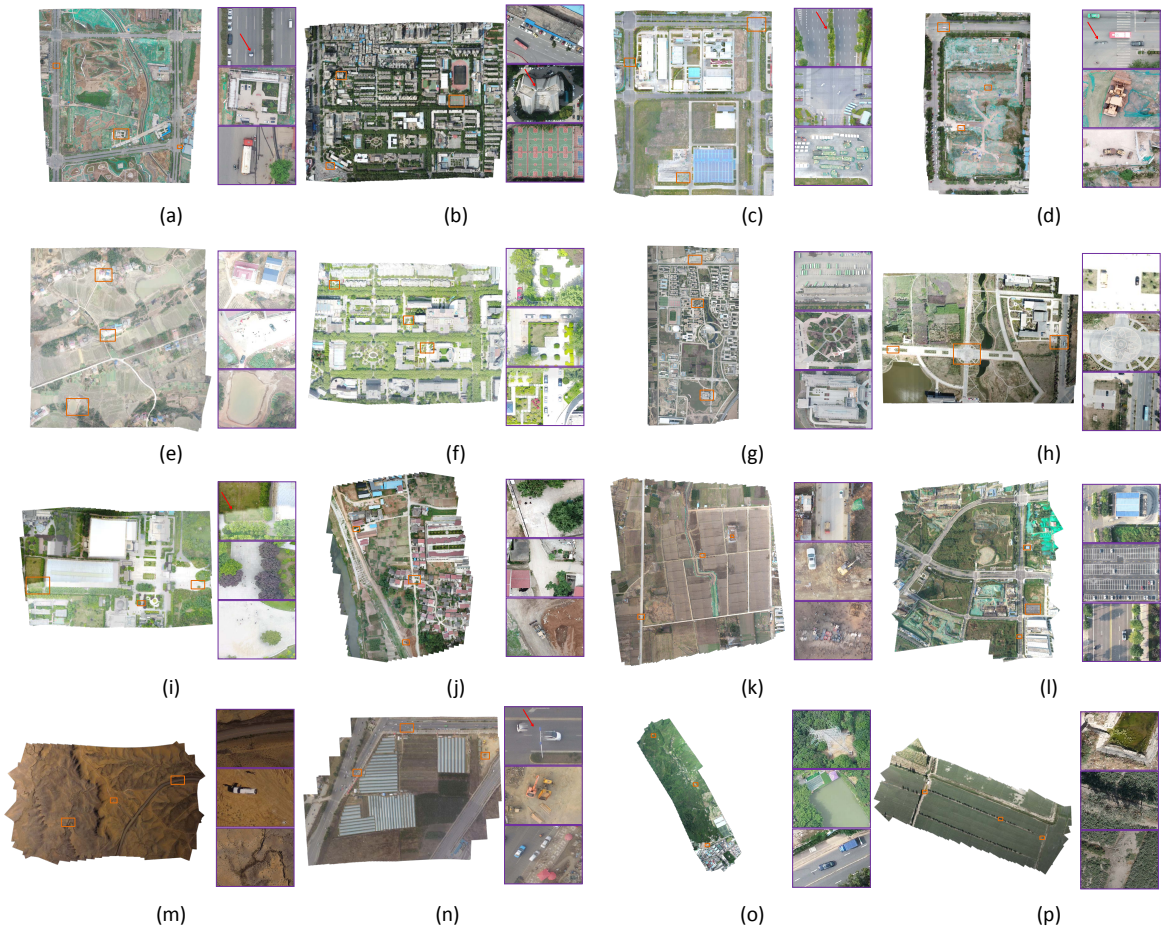

Figure 3. Mosaic results of the proposed method on public NPU DroneMap2 Dataset (**a**) *mavic2-road* (**b**) *mavic-campus* (**c**) *mavic-factory* (**d**) *mavic-fengniao*(**e**) *mavic-huangqi* (**f**) *mavic-library* (**g**) *mavic-npu* (**h**) *mavic-river*(**i**) *mavic-warriors* (**j**) *mavic-yangxian* (**k**) *p4r-field* (**l**) *p4r-roads2* (**m**) *phantom3-olathe* (**n**) *phantom3-strawberry* (**o**) *phantom4-mountain* (**p**) *xag-xinjiang*. Some screenshots are highlighted to demonstrate the mosaic details. Although few mismatches are caused by moving objects, reconstruction drift and high buildings, which are highlighted in red arrows, our stitching quality is generally high for different sequences including mountains and buildings.

Despite these slight mismatches, our system is able to output live orthophoto with novel quality and robustness. The live results are even comparable to the hour offline processing of state-of-the-art commercial software. A comparison with DJITerra and Pix4DMapper on some sequences is demonstrated in Figure 4. The three systems output high-quality results and some differences are highlighted. Planar assumption is used in our algorithm, and all images are stitched with homography projection. Thus, the lines in our results remain straight. By contrast, the SfM-based systems usually use 3D triangle mesh to render digital elevation model and orthophoto map, probably causing mismatch and deformation, as demonstrated in Figure 4.

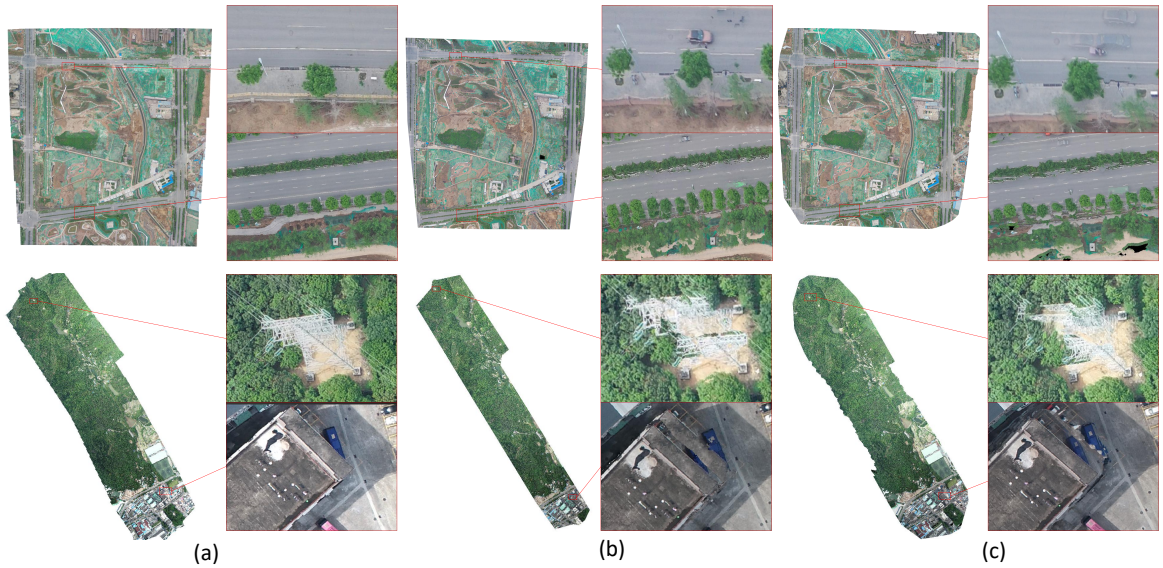

Figure 4. (**a**) Proposed method comparison with (**b**) DJITerra and (**c**) Pix4DMapper on sequence *mavic2-roads* and *mavic-factory*. An emphasis (e.g., red arrows) on the important differences between the results is provided to obtain clear comparisons. Homography projection is used; thus, the lines remain straight in our results, and the details show that our live mosaic looks even better than the offline results of state-of-the-art commercial software in some circumstances.

4.2. Live DOM Quality Comparison

The above dataset evaluation can only be compared with offline systems. Thus, we further evaluate our method by comparing with the live map function of software DJITerra. We use DJITerra to plan the flight mission and process the live downloaded preview images with our algorithm and Pix4DMapper. When using a gimbal for aerial photography, the camera is usually vertically downward. This assumption is easier for reconstruction 3D point cloud with dense stereo and stitching orthophoto with similarity assumption. However, the camera is not always vertical when a gimbal is not used. We reduce the overlap setting to 70% and allowed the camera to not be fully facing down because our system and DJITerra show good result in most traditional circumstances. The reconstruction becomes more challenging under this condition given that SLAM systems often require a higher overlap and the camera rotation reduces robustness. The mosaic result comparison of live map and offline is illustrated in Figure 5. We compare the live orthophoto of DJITerra by screenshots recorded given that it is covered by the postprocessing result. The comparison shows that our method is robust against low-overlap and camera rotations, where our algorithm outperforms even the existing novel offline methods. Moreover, our system only consumes approximately 1 min using x86 Linux computer to process this image sequence, and the flight time is nearly 20 min. This finding indicates that the algorithm is much faster than that used in real-time.

Here are some key ideas why our algorithm can be considerably fast while maintaining high quality:

1. Planes are used instead of map points for optimization. We do not rely on outlier sensitive map points. Thus, less keypoints are extracted, higher robustness is obtained, and less parameters are used for optimization. This feature dramatically decreases the optimization complexity and brings faster processing ability. We do not require to carefully handle outliers, and the optimization converges with less time.
2. GPS information is tightly used throughout the matching, reconstruction, and fusion procedures. We use the GPS information throughout the entire pipeline to reduce time budget because they are available for our georeferenced stitching. The GPS-aided absolute rotation averaging prevents poor pose estimation and accelerates the convergence of graph optimization.

3. Direct orthophoto blending without dense and mesh reconstruction. Most SfM systems perform dense reconstruction and mesh triangulation before DOM rendering. Our method directly fuses images to the final mosaic efficiently, and our reconstruction step uses planar assumption at the first stage. Thus, it provides better quality than map point-based SLAM front-end systems.

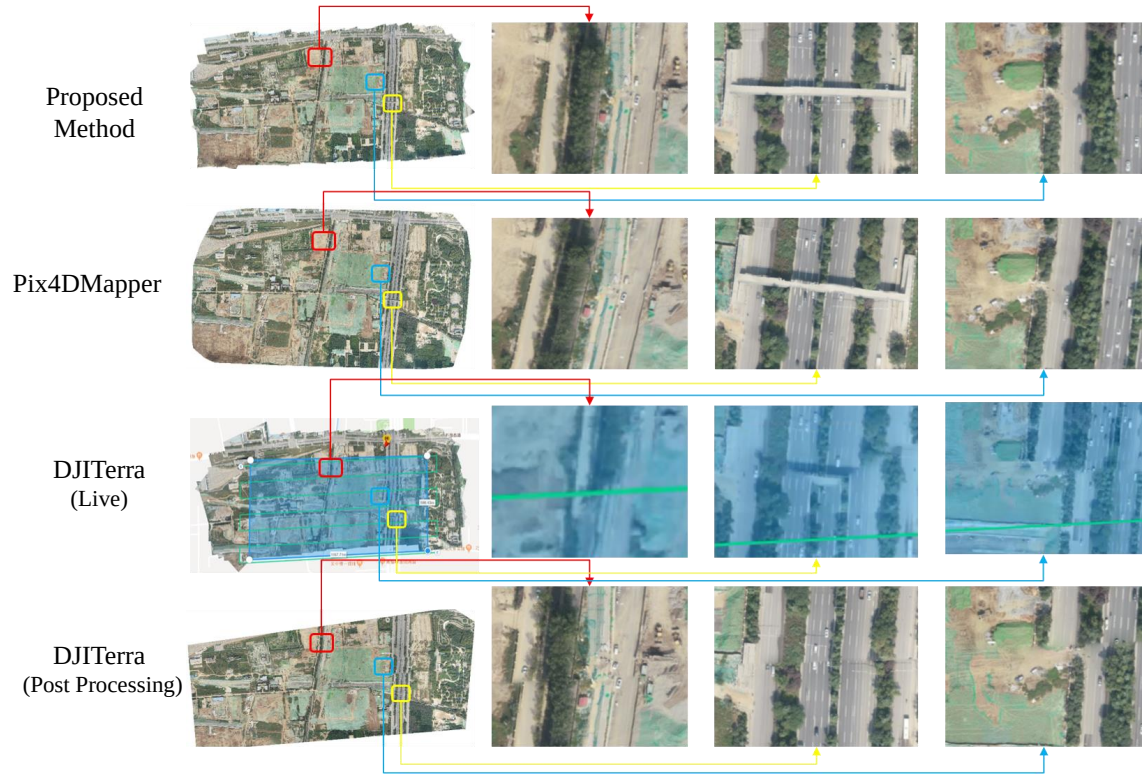

Figure 5. Stitching result comparison between the proposed method and state-of-the-art commercial software. DJITerra will launch a postprocessing step to refine the mosaic. Thus, we illustrate the result before and after postprocessing. DJITerra performs unsatisfactorily in this flight with evident mismatches in live map or postprocessing results due to the low flight overlap and random pitch. Our system is able to handle these challenges and output high-quality live orthophoto, which even looks better than Pix4DMapper, where the bridge and roads looks distorted.

4.3. Computation Performance Comparison

Efficiency is the most important feature and target of our system because of the real-time requirement on mobile phones. To evaluate the computational performance and compare with state-of-the-art commercial software DJITerra and Pix4DMapper, we process the public DroneMap2 dataset in the same computer with Intel i7-6700 CPU, 16 GB RAM, and Nvidia GTX 1060 GPU. As demonstrated in Table 1, all methods output original resolution results, and our method is much faster than the other software. Our system is able to process over 10 MB JPG-compressed full resolution images and over 10 preview images with 1080p resolution in one second. Processing multiple images in parallel is difficult given their incremental design, and our method even uses less CPU computation resources. For better portability on different platforms, most procedures only use CPU without intensive computational optimization, and other potentials can be explored by considering particular hardware platforms.

Table 1. Time usage statistics in seconds for processing sequences of DroneMap2 dataset. DJITerra failed to process sequence *xag-xinjiang* without any tips about the reason. The results show that our algorithm is much faster than other state-of-the-art commercial softwares.

Sequence	Location	Images	Resolution	Size (MB)	Ours		DJITerra (s)	Pix4DMapper (s)
					Time (s)	Average (MB/s)		
mavic2-road	Xi'an, Shaanxi	240	5472 × 3078	1895.4	119.7	15.8	583.0	4867.0
mavic-campus	Xi'an, Shaanxi	293	4000 × 3000	1479.4	64.4	22.9	568.0	4707.0
mavic-factory	Xi'an, Shaanxi	359	4000 × 3000	1924.6	148.7	12.9	666.0	4811.0
mavic-fengniao	Xi'an, Shaanxi	216	4000 × 3000	1102.5	87.6	12.6	434.0	3303.0
mavic-garden	Suzhou,Jiangsu	247	4000 × 3000	1241.0	89.1	13.8	550.0	4753.0
mavic-hongkong	Hong Kong	288	4000 × 3000	1439.5	130.2	11.1	575.0	4723.0
mavic-huangqi	Hengyang, Hunan	229	4000 × 3000	1156.5	98.2	11.8	454.0	4351.0
mavic-library	Xi'an, Shaanxi	205	4000 × 3000	997.3	69.2	14.4	365.0	3786.0
mavic-npu	Xi'an, Shaanxi	119	4000 × 3000	603.2	34.1	17.7	194.0	1834.0
mavic-river	Xi'an, Shaanxi	166	4000 × 3000	960.4	64.7	14.8	408.0	3746.0
mavic-warriors	Xi'an, Shaanxi	96	4000 × 3000	779.7	26.2	29.7	182.0	1623.0
mavic-yangxian	Xi'an, Shaanxi	165	4000 × 3000	840.2	75.4	11.1	392.0	2935.0
p4r-field	Xi'an, Shaanxi	683	5472 × 3648	5939.0	381.1	15.6	1837.0	23,941.0
p4r-roads	Xi'an, Shaanxi	138	5472 × 3648	1058.6	84.2	12.6	501.0	4135.0
p4r-roads2	Xi'an, Shaanxi	203	5472 × 3648	1556.1	125.6	12.4	556.0	6332.0
p4r-village	Xi'an, Shaanxi	136	5472 × 3648	1038.4	85.4	12.2	353.0	2762.0
phantom3-olathe	Olathe, USA	160	4000 × 3000	898.7	74.6	12.0	312.0	2514.0
phantom3-strawberry	Xi'an, Shaanxi	184	4000 × 3000	990.5	86.3	11.5	473.0	3591.0
phantom4-mountain	Shenzhen, Guangdong	81	4864 × 3648	627.7	51.9	12.1	252.0	1752.0
xag-xinjiang	Yuli, Xinjiang	303	4864 × 3648	2179.4	163.7	13.3	-	4362.0

4.4. Quantitative DOM Precision Evaluation

The target of this study is to stitch a live georeferenced orthophoto on mobile devices, where efficiency and robustness is the first priority. This finding indicates that we slightly sacrifice some precision. A quantitative precision evaluation is performed on sequence *p4r-field*, and the detailed errors of our result are illustrated in Figure 6. Although the RMSE of our system is slightly larger than SfM methods. Our result is globally consistent and good in appearance because the optimization error reduces the mismatch instead of reconstructing the precise locations. This precision is adequate for a large number of applications, such as emergency rescue searching, path planning, and measuring.

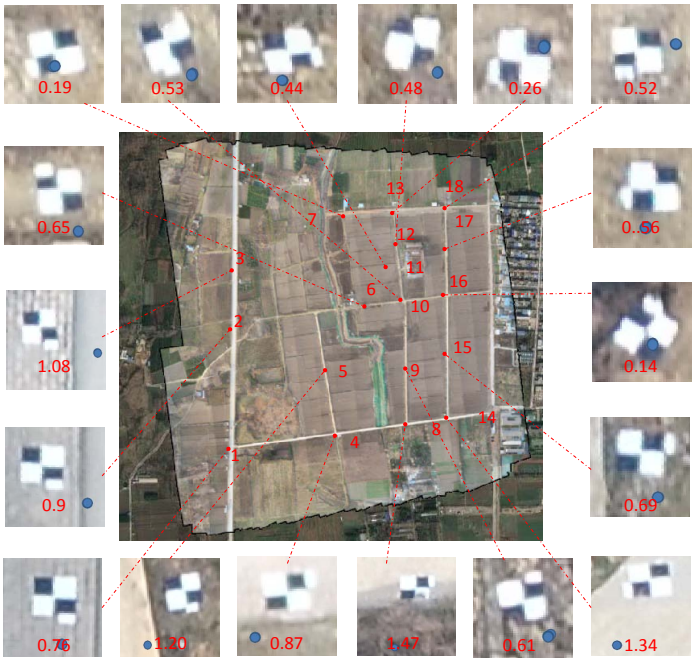

Figure 6. Detailed errors of GCPs in *p4r-field* dataset. The white–black marker size is 60 × 60 cm, and blue dot indicates the GCP location. The absolute error indicates the distance between the marker center and GCP position; it is measured in meters with red text. Since we organize the screenshots manually, so that the resolution may look not equal. Although the root mean square error (RMSE) of our method is slightly larger than Pix4DMapper and DJITerra, it is precise for most applications.

4.5. Web-Based Live Map Sharing

Our live map can not only be visualized by the host device, but can also be shared through local area network (LAN) or even worldwide network, owing to the restful API design. A web-based display implementation is developed, and all devices with browsers are able to visit the low-latency high-resolution live map. Figure 7 demonstrates the display screenshots on PC and smart phone, and the evaluation shows that only 100 KB/s bandwidth is required to visualize and update the live map.

Figure 7. The live map can be visualized in PC and mobile phone web browsers. To share the dynamic orthophoto through network efficiently, a restful API, and a corresponding rendering strategy is designed. Only tiles, which are visible and updated, are transferred, and a network with over 100 KB/s bandwidth is sufficient to render a low-latency high-resolution dynamic layer. This finding indicates that the real-time mapping service can be placed everywhere and shared through the Internet.

5. Conclusions

In order to realize an online georeferenced orthophoto mosaicing solution for mobile devices, we present a novel plane restrained visual SLAM method with high efficiency and robustness. Firstly, an appearance and spatial correlation-constrained fast low-overlap neighbor candidate query and matching strategy is used for efficient and robust global matching. Then, a novel graph-based pose optimization is applied for overlapping images. GPS information is also fused along with 6-DOF pose estimation, which not only provides georeferenced coordinates, but also converges property and robustness. Finally, an incremental orthophoto is generated with an adaptive weighted multiband algorithm for fast mosaicing, which is specific suitable for low-overlap images. In order to evaluate the effectiveness of the proposed method, we compared it with state-of-the-art software. Experimental results show that the proposed method is fast, accurate and robust. Furthermore, an android commercial application is developed for online stitching with DJI drones, considering the excellent performance of our algorithm. Due to the low computation requirements of cellphones, our system only outputs 2D orthophoto instead of 3D reconstruction, and the use of homography transformation and planar assumption requires more flight height. In the future, full 3D reconstructions with both 3D and ortho outputs are desired.

Author Contributions: Conceptualization, S.X. and S.B.; methodology, Y.Z. and Y.C.; writing–review and editing, P.H. and X.Z.; supervision, H.J.; resources, K.L. and G.W.; All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work was supported in part by the National Key R&D Program of China (Grant 2018YFB2100602), and in part by the National Natural Science Foundation of China (Nos. 61620106003, 91646207, 61971418, 61671451 and 61573284).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Verhoeven, G. Taking computer vision aloft–archaeological three-dimensional reconstructions from aerial photographs with photoscan. *Archaeol. Prospect.* **2011**, *18*, 67–73. [CrossRef]
2. Zhao, Y.; Liu, G.; Xu, S.; Bu, S.; Jiang, H.; Wan, G. Fast Georeferenced Aerial Image Stitching with Absolute Rotation Averaging and Planar-Restricted Pose Graph. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–16, doi:10.1109/TGRS.2020.3008517.
3. Vallet, J.; Panissod, F.; Strecha, C.; Tracol, M. Photogrammetric performance of an ultra light weight swinglet UAV. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *XXXVIII-1/C22*, 253–258, [CrossRef]
4. Bu, S.; Zhao, Y.; Wan, G.; Liu, Z. Map2dfusion: Real-time incremental UAV image mosaicing based on monocular slam. In Proceedings of the IEEE Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference, Daejeon, Korea, 9–14 October 2016; pp. 4564–4571.
5. Botterill, T.; Mills, S.; Green, R. Real-time aerial image mosaicing. In Proceedings of the IEEE Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference, Queenstown, New Zealand, 8–9 November 2010; pp. 1–8.
6. De Souza, R.H.C.; Okutomi, M.; Torii, A. Real-time image mosaicing using non-rigid registration. In *Pacific-Rim Symposium on Image and Video Technology*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 311–322.
7. Lütjens, M.; Kersten, T.; Dorschel, B.; Tschirschwitz, F. Virtual Reality in Cartography: Immersive 3D Visualization of the Arctic Clyde Inlet (Canada) Using Digital Elevation Models and Bathymetric Data. *Multimodal Technol. Interact.* **2019**, *3*, 9. [CrossRef]
8. Edler, D.; Keil, J.; Wiedenlübbert, T.; Sossna, M.; Dickmann, F. Immersive VR Experience of Redeveloped Post-industrial Sites: The Example of Z̈eche Hollandïn Bochum-Wattenscheid. *KN J. Cartogr. Geogr. Inf.* **2019**, *69*, 267–284.
9. Hruby, F.; Castellanos, I.; Ressl, R. Cartographic Scale in Immersive Virtual Environments. *KN J. Cartogr. Geogr. Inf.* **2020**, [CrossRef]
10. Mur-Artal, R.; Montiel, J.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *arXiv* **2015**, arXiv:1502.00956.
11. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
12. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
13. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [CrossRef]
14. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
15. Snavely, N. Bundler: Structure from Motion (SFM) for Unordered Image Collections. 2008. Available online: http://phototour.cs.washington.edu/bundler/ (accessed on 10 January 2020).
16. Moulon, P.; Monasse, P.; Marlet, R. Global fusion of relative motions for robust, accurate and scalable structure from motion. In Proceedings of the Computer Vision (ICCV), 2013 IEEE International Conferenc, Sydney, Australia, 1–8 December 2013; pp. 3248–3255.
17. Sweeney, C. Theia Multiview Geometry Library: Tutorial & Reference. Available online: http://theia-sfm.org (accessed on 1 October 2020).
18. Lati, A.; Belhocine, M.; Achour, N. Robust aerial image mosaicing algorithm based on fuzzy outliers rejection. *Evol. Syst.* **2019**, *11*, 717–729.
19. Warren, M.; McKinnon, D.; He, H.; Glover, A.; Shiel, M.; Upcroft, B. Large scale monocular vision-only mapping from a fixed-wing sUAS. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 495–509.

20. Gherardi, R.; Farenzena, M.; Fusiello, A. Improving the efficiency of hierarchical structure-and-motion. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1594–1600.
21. Turner, D.; Lucieer, A.; Watson, C. 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 Sens.* **2012**, *4*, 1392–1410. [CrossRef]
22. Salaün, Y.; Marlet, R.; Monasse, P. Robust SfM with Little Image Overlap. *arXiv* **2017**, arXiv:1703.07957.
23. Botterill, T.; Mills, S.; Green, R.D. New Conditional Sampling Strategies for Speeded-Up RANSAC. In Proceedings of the BMVC 2009, London, UK, 7–10 September 2009; pp. 1–11.
24. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
25. Davis, J. Mosaics of scenes with moving objects. In Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231), Santa Barbara, CA, USA, 25 June 1998; pp. 354–360.
26. Garcia-Fidalgo, E.; Ortiz, A.; Bonnin-Pascual, F.; Company, J.P. Fast image mosaicing using incremental bags of binary words. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1174–1180.
27. Guizilini, V.; Sales, D.; Lahoud, M.; Jorge, L. Embedded mosaic generation using aerial images. In Proceedings of the 2017 IEEE Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, Brazil, 8–11 November 2017; pp. 1–6.
28. Juan Jesus Ruiz, F.C.; Merino, L. MGRAPH: A Multigraph Homography Method to Generate Incremental Mosaics in Real-Time From UAV Swarms. *IEEE Robtics Autom. Lett.* **2018**, *3*, 2838–2845.
29. Zhao, Y.; Xu, S.; Bu, S.; Jiang, H.; Han, P. GSLAM: A General SLAM Framework and Benchmark. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
30. Snyder, J.P. *Map Projections—A Working Manual*; US Government Printing Office: Washington DC, USA, 1987; Volume 1395.
31. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the Computer Vision (ICCV), 2011 IEEE International Conference, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
32. Muja, M.; Lowe, D.G. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP* **2009**, *2*, 2.
33. Tang, C.; Wang, O.; Tan, P. Gslam: Initialization-robust monocular visual slam via global structure-from-motion. In Proceedings of the 2017 IEEE International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 155–164.
34. Horn, B.K. Closed-form solution of absolute orientation using unit quaternions. *JOSA A* **1987**, *4*, 629–642. [CrossRef]
35. Chatterjee, A.; Madhav Govindu, V. Efficient and robust large-scale rotation averaging. In Proceedings of the IEEE International Conference on Computer Vision, Sydney Australia, 1–8 December 2013; pp. 521–528.