Cloud-based collaborative 3D reconstruction using smartphones

Fabio Poiesi
Technologies of Vision
Fondazione Bruno Kessler
Trento, IT
poiesi@fbk.eu

Alex Locher
Computer Vision Laboratory
ETH
Zurich, CH
alocher@vision.ee.ethz.ch

Paul Chippendale
Technologies of Vision
Fondazione Bruno Kessler
Trento, IT
chippendale@fbk.eu

Erica Nocerino
3D Optical Metrology
Fondazione Bruno Kessler
Trento, IT
nocerino@fbk.eu

Fabio Remondino
3D Optical Metrology
Fondazione Bruno Kessler
Trento, IT
remondino@fbk.eu

Luc Van Gool
Computer Vision Laboratory
ETH
Zurich, CH
vangool@vision.ee.ethz.ch

ABSTRACT

This article presents a pipeline that enables multiple users to collaboratively acquire images with monocular smartphones and derive a 3D point cloud using a remote reconstruction server. A set of key images are automatically selected from each smartphone's camera video feed as multiple users record different viewpoints of an object, concurrently or at different time instants. Selected images are automatically processed and registered with an incremental Structure from Motion (SfM) algorithm in order to create a 3D model. Our incremental SfM approach enables on-the-fly feedback to the user to be generated about current reconstruction progress. Feedback is provided in the form of a preview window showing the current 3D point cloud, enabling users to see if parts of a surveyed scene need further attention/coverage whilst they are still in situ. We evaluate our 3D reconstruction pipeline by performing experiments in uncontrolled and unconstrained real-world scenarios. Datasets are publicly available.

CCS CONCEPTS

• **Computer vision** → **Computer vision problems**; • **Computer vision problems** → Reconstruction; • **Image and video acquisition** → 3D imaging;

KEYWORDS

Collaborative 3D Reconstruction, Structure from Motion, Mobile Device.

ACM Reference Format:
Fabio Poiesi, Alex Locher, Paul Chippendale, Erica Nocerino, Fabio Remondino, and Luc Van Gool. 2017. Cloud-based collaborative 3D reconstruction using smartphones. In *CVMP 2017: 14th European Conference on Visual Media Production (CVMP 2017), December 11–13, 2017, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3150165.3150166

CVMP 2017, December 11–13, 2017, London, United Kingdom
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5329-8/17/12…$15.00
https://doi.org/10.1145/3150165.3150166

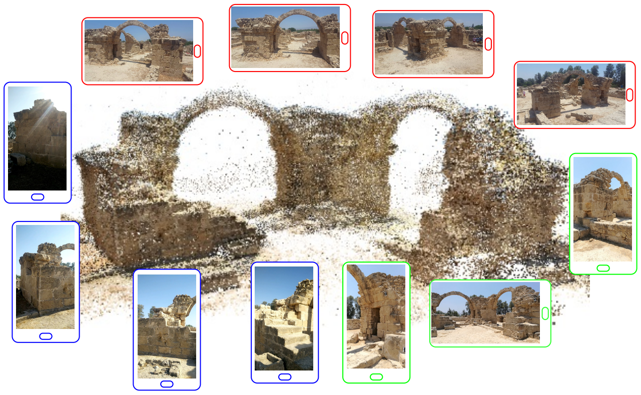


Figure 1: 3D point cloud of the archaeological site of Saranta Kolones (Cyprus) obtained collaboratively using three smartphones. Images are colour-coded according to the smartphone they were taken from.

1 INTRODUCTION

After nearly a decade of mobile image-based 3D reconstruction research [15, 28], advanced prototypes have started to reach app-stores. These apps (e.g. [10, 30]) enable users to become 3D creators by digitising real-world objects around them. Such apps share similar 3D reconstruction pipelines that are based on Structure from Motion (SfM) techniques. In SfM, distinctive points of a captured scene are triangulated from independent images taken from different viewpoints to create a sparse 3D point cloud.

Computer vision algorithms embedded in these apps generally target a specific object size (from small desktop objects to people [11, 15]), although some work has also been done to reconstruct larger scale objects using geo-referencing cues [33]. Irrespective of what a target object is, or its size, a user must systematically gather images from all around it to create a complete 3D point cloud. Objects which exhibit a geometrically simple form (e.g. a column) require fewer images compared to more complex structures (e.g. a church), hence planning, experience and feedback are crucial to assist non-expert operators to ensure the quality and efficiency of the process. State-of-the-art solutions for SfM on smartphones only offer feedback to single users during acquisitions [15, 19], and collaborative approaches that potentially offer concurrent feedback to multiple users currently do not exist.

Our work was stimulated by collaborative and scalability issues with non-expert users in mind, which lead us to create a pipeline that could enable multiple users to succesfully work together. We acheived this by implementing an incremental and joint SfM pipeline (Fig. 1). In practice, the development of such a scalable and collaborative architecture introduces several challenges, such as: how to coordinate multiple users to guarantee complete object coverage, how to cope with acquisition streams coming from different devices taken at different times of the day (i.e. illumination variations) and how to cope with the lack of guaranteed overlaps across images from different acquisition sessions.

In this paper, we present a collaborative image-based 3D reconstruction pipeline to perform image acquisition with a smartphone and geometric 3D reconstruction on a server. Images are selected from a smartphone's camera feed based on both their quality and on their novelty [18]. The smartphone app provides on-the-fly reconstruction feedback in a preview window to all users that are co-involved in an acquisition. The server is composed of an incremental SfM algorithm that processes received images and then seamlessly merges them into a single point cloud [18]. 3D-point triangulations and estimations of camera parameters are computed using Bundle Adjustment [29]. Additionally, a Multi View Stereo (MVS) algorithm can also be selected to derive denser point clouds. Differently from [31] and [23], our pipeline updates and augments a global 3D point cloud with each new image received, regardless of which user has acquired it, instead of just building separate sub-models and then fusing them later. It is this concurrent reconstruction that enables our system to offer up-to-date visual feedback to users during acquisition. Our server also offers a web-based visualisation service where users can preview 3D scenes, estimated reconstruction parameters and display dense point clouds. We evaluate our proposed pipeline with experiments carried out in real-world scenarios, such as cultural heritage sites and city monuments. We quantify performance by analysing the completeness of reconstructions as a function of time and of the number of users involved in an acquisition.

2 RELATED WORK

The proposed system uses Structure from Motion (SfM) to automatically determine camera parameters and 3D cloud points. In general, SfM methods can be categorised into three different approaches: global [27], hierarchical [6, 8] and incremental [24, 26, 32]. *Global* methods achieve 3D reconstruction using sets of sequential pre-collected images that are usually processed in batches. Camera rotations and translations are globally estimated using pairwise geometries. 3D points are triangulated from matched features across images and adjusted via global optimisations such as Bundle Adjustment (BA). *Hierarchical* methods group pre-collected images into clusters based on a spatial distribution of keypoints. Reconstructions are augmented by triangulating points from the same clusters and also by iteratively merging them across different clusters. *Incremental* methods build 3D point clouds by either triangulating images one-by-one [31], or by processing them in mini-batches (groups of N, e.g. 15, images) [9] as they are added to an already-initiated reconstruction. Point cloud initialisation is usually carried out by triangulating points from the first two images only. Incremental methods are more computational expensive than global methods, but enable online SfM processing. We based our pipeline on an incremental SfM approach to process images as they are uploaded to a reconstruction server.

SfM-based methods produce only sparse point clouds, as they solely triangulate feature points extracted from images. When SfM is coupled with Multi View Stereo approaches, dense point clouds can also be produced [24]. However, there is a marked increment in computational complexity between the two, making real-time performance with off-the-shelf smartphones resource prohibitive. High-end smartphones have recently been shown to produce sparse 3D volumetric models at ~11Hz [21], dense models at ~0.4Hz [28], and textured mesh models at ~0.02Hz [15]. Reconstruction methods that run solely on smartphones have the advantage of providing users with instant feedback about acquisitions and can also function without an internet connection. However, current fully-on-device implementations inhibit collaborative strategies [31], consume all available device resources, and prohibit the integration of other demanding applications such as object classification [2]. SfM has also been split across smartphone and servers to distribute computational loading [12, 31]. In this case, a smartphone is usually in charge of capturing images, and a server performs the computationally intensive tasks of estimating camera orientations and point triangulation. However, this can often lead to short delays (e.g. 40s) between the transmission of images to a server and the updating of 3D reconstructions [31].

In the case of real-time constrained applications, Visual Simultaneous Localization And Mapping (VSLAM) methods [22] have demonstrated promising results. VSLAM methods can either be feature-based [22] and produce sparse 3D reconstructions, or can be direct [4] and produce reconstructions with denser point clouds. The latter category is computationally more expensive than the former as it performs camera pose estimation and mapping directly on pixel values, rather than on feature points. VSLAM can also be performed collaboratively to aid multi-robot navigation by improving trajectory estimation and accuracy of a global map [5, 14, 23]. Because VSLAM is aimed at real-time applications (e.g. robotic navigation), point clouds are generated at 30Hz from low-quality images, typically 320x240 [4], resulting in low-quality point clouds.

In our work, we use a cloud-based SfM strategy with high quality images transmitted to a remote server to build high-quality reconstructions, for use in Augmented Reality and Virtual Reality applications. Our pipeline is designed to provide users with rapid feedback about the status of a reconstruction, and to enable the combining of images from multiple devices/users from concurrent and disjoint sessions. Reconstruction feedback is key to help users collaborate and understand which portions of an object need further attention. Differently from other smartphone-based 3D-reconstruction frameworks (e.g. [30]), we use a subset of images that are automatically selected from a smartphone's camera video feed, thus optimising bandwidth. In [30], a maximum of 70 images

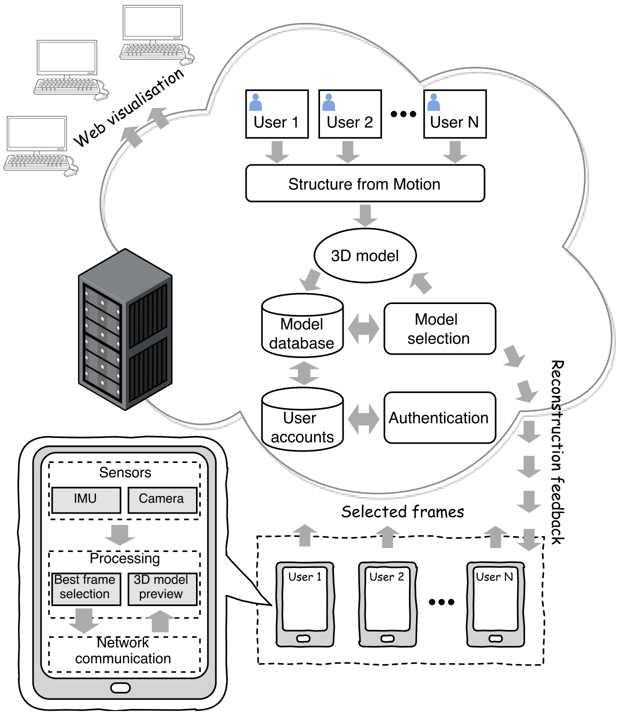


Figure 2: Proposed collaborative 3D reconstruction pipeline.

can be taken using the camera's burst mode, thus restricting acquisitions to small objects, whereas in our system we can transmit images from several smartphones indefinitely. In contrast to [10], our pipeline is also designed to work with off-the-shelf smartphones that do not use active sensors (e.g. laser projectors). In [12], the authors highlight the benefits of having 3D reconstructions that can be jointly performed on a smartphone and on a server, and they provide an analysis of workload balancing. A similar type of architecture has also been adopted in our collaborative framework.

In the following sections, we introduce the approach that we have taken to identify related groups of images from multiple users and acquisition sessions, explaining how we can estimate camera parameters from different devices, how we can update portions of a global model, and how we can provide on-the-fly 3D model reconstruction feedback to all collaborators.

3 OVERVIEW

The proposed pipeline facilitates the automatic and collaborative 3D reconstruction of objects from images acquired by a smartphone app and concurrently geometrically processed on a cloud server. Collaborative, i.e. multi-user, acquisitions lend themselves to improved object reconstruction quality, better object coverage, and faster object acquisition campaigns. Fig. 2 summarises the implemented pipeline.

The pipeline consists of a *smartphone app* and a *3D reconstruction server* [18]. Each user's smartphone running the app must first be authenticated by our cloud service. Unique smartphone identifiers (ID) are assigned based on a user's account credentials, a device's manufacturer, its model and operating system. The smartphone app runs an algorithm that automatically selects images for transmission based on their deemed relevance for the reconstruction engine, associating them to corresponding users based on ID. Accelerometer measurements from a device's Inertial Measurement Unit (IMU) are transmitted alongside the images to aid pose estimation and object reconstruction. Device haptic feedback (i.e. phone vibration) also provides users with an intuitive means of understanding whether they are moving the device in a system-acceptable way. Network communication between the reconstruction server and device is bidirectional and asynchronous. The app offers a user the option to start a new reconstruction session, or, to update past sessions with new images. To generate up-to-date and meaningful feedback, the smartphone sends periodic requests to the server for new updated point cloud models. The server responds to these requests by providing the most recent (up-to-date) reconstructed version of the object being surveyed. The remote server handles user authentication as well as generating 3D reconstructions and previews for app and web-based visualisations. This web-based visualisation enables users to interact with their reconstructions, e.g. see estimated camera positions and interact with the dense point cloud. Users can choose with whom to share reconstructions by adding them as contributors via an email option. Thus, reconstructed objects are associated to owners and shared contributors. The 3D reconstruction server is powered by an incremental SfM algorithm (similar to [27]) that implements a 3D point triangulation method tailored for collaborative 3D reconstruction.

4 ON-DEVICE IMAGE SELECTION

The mobile app automatically selects a subset of images from the camera's video feed (@30Hz) to avoid the transmission of poor or near-identical images. In cases where the spatial overlap between images is large, SfM algorithms can sometimes produce inaccurate 3D triangulations, hence it should be avoided. Moreover, by transmitting only 'selected' images we can drastically reduce the bandwidth usage between smartphone and server.

The image selection algorithm is composed of two-stages. Firstly, the quality of each image is assessed by looking at its global sharpness [25], as sharp images significantly improve SfM outputs because good features can be extracted and compared reliably. Secondly, recent sharp images are compared based on their visual information to previously acquired images. Images should have some degree of overlapping visual content whilst still being sufficiently spatially separated. Experiments have shown that content overlaps between 20% and 80% contain significantly new content to make their triangulation worthwhile [3]. In our method, the overlap between images is quantified using feature points. Overlap is scored by comparing the ratio between newly calculated features (from a new image) to older features from previously uploaded images. We use an ORB-based feature extractor [22] because of its real-time performance on off-the-shelf smartphones. Old features are stored in a circular buffer that is periodically updated when new images are selected. In practice, K (=500) ORB feature keypoints are extracted from each new image available from the smartphone's camera video feed. An new image is selected if at least 20%, but not more than 80%, of the new features match with those in the circular buffer. When an image is selected, the app retrieves the value of the gravity vector from the accelerometers in the IMU. The gravity vector helps the SfM algorithm to correctly orient the reconstruction. Selected images along with their gravity vectors are placed in a queue for uploading to the 3D reconstruction server. A dedicated thread within the app manages this queue and the uploads to the server. To help the user understand the frequency of images being selected, the smartphone app also generates haptic feedback each time an new image is selected for transmission.

5 COLLABORATIVE 3D RECONSTRUCTION

The SfM algorithm running on the server uses multiple threads to process independent and asynchronous uploads of images from different users (Fig. 3).

For each newly received image, the algorithm matches news computed features to those from a subset of N_s (e.g. 20) images taken from the same session. This subset is composed of images already stored in the database that have a high visual similarity with the new one, as images with high similarity are most likely to overlap and include the same parts of the scene to reconstruct. Each image is indexed in a database using a 1M word vocabulary based on RootSIFT features [1]. The vocabulary is trained on the Oxford5k dataset [20] and visual words are created using hierarchical k-means [17]. Visual similarity is computed using the Term Frequency-Inverse Document Frequency (TF-IDF) scoring method [34].

In a collaborative scenario, images of the same scene can be acquired by different smartphones from different viewpoints, leading to a diminished likelihood of overlapping content (compared to those acquired by the same smartphone that are temporally ordered). For images taken from the same device, N_l (e.g. 3), out of the last N_s images are directly added to the subset of images to match. This saves computation time as we can bypass the retrieval operation for this subset of images and the probability of irrelevant matching candidates is reduced. The remaining $N_s - N_l$ images are retrieved from the database of indexed images using TF-IDF. To estimate image orientation parameters, a standard method based on a ratio test and geometric verification (i.e. fundamental/essential matrix computation) [7, 13] is applied.

Relative image orientations are encoded in fundamental and essential matrices and are initially estimated using 2D-3D correspondences from estimated image feature points regardless of which smartphone they were captured from. If a nominal focal length is available from an image's EXIF metadata, or from former acquisitions by the same device, we estimate an essential matrix using a five-point algorithm based on RANSAC [16]. However, if the focal length is not available, then we estimate the fundamental matrix using an eight-point algorithm (also based on RANSAC) and then infer the essential matrix [7].

We refine camera (intrinsic) parameters and orientations using two iterations of Bundle Adjustment (BA) in order to handle images acquired from heterogeneous cameras. We assume that a smartphone's camera configuration remains fixed during an acquisition, as images acquired by the same smartphone should share the same intrinsic parameters. We create *intrinsic groups*, where each group contains images acquired by the same smartphone. A locally bounded BA refines only the most recent N_{bal} cameras and associated points. Note that there is still the chance that some images may be oriented unsuccessfully due to a lack of feature matches during the reconstruction process. Once the reconstruction has grown by more than η_{glb} , a full BA over all cameras and points, taking into consideration intrinsic parameters along with their affiliations to intrinsic groups, is performed. In this way, the intrinsic parameters of each smartphone's camera can be estimated. This two-stage BA saves computation time and increases the stability of the BA optimisation.

6 3D RECONSTRUCTION PREVIEW & VISUALISATION

All users involved in a collaborative effort have the option to visualise their joint progress via a dedicated preview window in the smartphone app (Fig. 4), or to interact with the reconstruction session via a web page.

The preview model in the app shows the user the global 3D reconstruction of all concurrent users in the form of a point cloud. The preview window runs on a separate thread that periodically (e.g. 3 seconds) sends requests to the server to see if there are any new models to display. When an acquisition is ended by a user, and the 3D reconstruction process is completed on the server, requested updates for new models is increased to every 10 seconds. These intervals are flexible and can be set by the user in the app settings.

Fig. 4 shows an example of a collaborative 3D reconstruction with the preview window activated. The upper images are samples of selected images. Fig. 4b shows two screenshots that were captured from the live view of two smartphones during an acquisition. The left-hand smartphone (Sony Z5 - blue) was in acquisition mode (see rec button is ON). The right-hand smartphone (LG Nexus 5X - red) also had the preview window activated to visualise reconstruction feedback. This preview window shows the global point cloud of the reconstructed object in addition to the positions of the cameras (green points). Note: model preview had been deactivated on the blue smartphone to better display the results in this screenshot. The right-hand figure of Fig. 4b shows the 3D point cloud with the colour-coded cameras representing each smartphone's pose during acquisition. The 3D points triangulated from the blue smartphone's images are coloured cyan, those triangulated from the red smartphone's are coloured orange, and those triangulated by both are coloured magenta. The total number of 3D points in this example is 71816, of which 39768 (55%) were triangulated from solely the blue smartphone, 25398 (35%) were triangulated from solely the red smartphone and 6650 (9%) were triangulated from images from both devices. This exemplifies the collaborative nature of the proposed system and illustrates the system's potential for creating large models with multiple smartphones in a short period of time.

In addition to the app preview window, users can also access additional features via a web browser: to visualise the position of the oriented images, change cloud points sizes, download estimated camera parameters and access all intermediate reconstructions.

Fig. 5 shows two screenshots of our web-based visualisation for the same reconstruction session as Fig. 4. In the top-left corner of

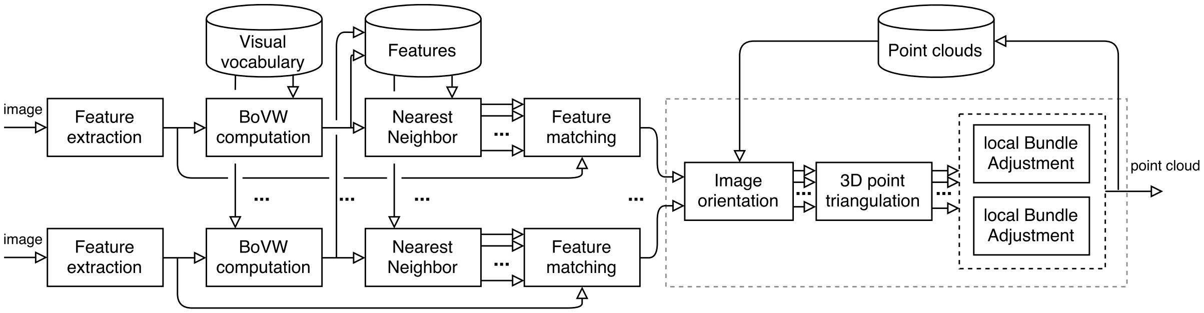


Figure 3: Incremental Structure from Motion pipeline used to enable collaborative 3d reconstruction.

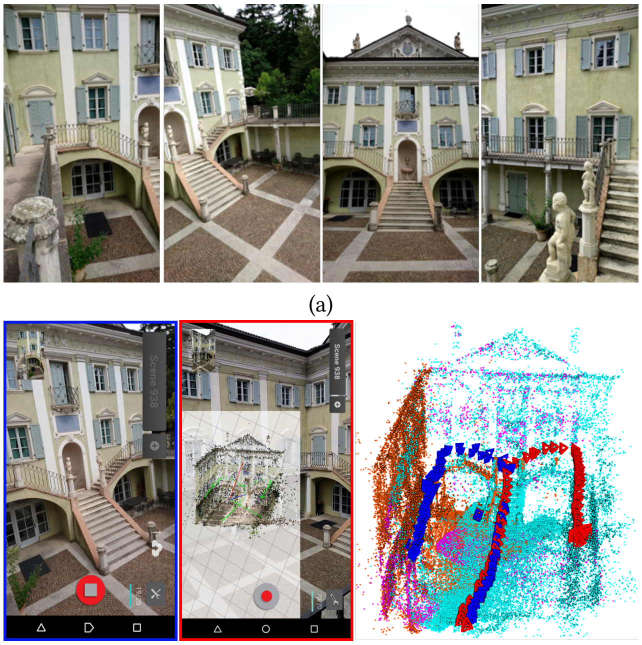
Fig. 5a we can see the user owner and the option to add users to the same reconstruction session. The top-right corner contains a list of intermediate reconstructions, where we can download the point cloud (ply) and the parameters estimated during SfM (nvm). The bottom part of the page shows the first eight selected images that are flagged as done, which means that they have been triangulated. In Fig. 5b we can see the point cloud of the reconstructed building and the oriented images. The 'Visualisation Settings window' shown on the right-hand side, can be used to modify the appearance of the point cloud and to enable different features such as the visualisation of the dense point cloud.

7 EXPERIMENTS AND RESULTS

The following section describes the experiments we performed to demonstrate the capabilities of the introduced pipeline. We will report datasets collected, collaborative reconstruction results, and system responsiveness to the user during acquisition. All of the experiments reported were conducted on an Intel Xeon 2.30GHz machine with 128GB of RAM and 20 cores. If not stated otherwise, the following parameters were used: $N_l = 5$, $N_l = 20$, $N_{\text{bal}} = 20$, $\eta_{\text{glb}} = 15\%$.

7.1 Dataset

Our experiments depict real-world scenarios and the images we collected were captured using six different, off-the-shelf Android smartphones. During the acquisition sessions, users were advised to hold their smartphones naturally in their hands, and to slowly walk around the object they wanted to scan pointing their camera towards it; haptic feedback would let them know if they were doing a good job. To the authors best knowledge, there are currently no datasets available that were captured using multiple and different smartphones that depict buildings and objects from different viewpoints. Consequently, we have made our four datasets publicly available for future researchers[1].



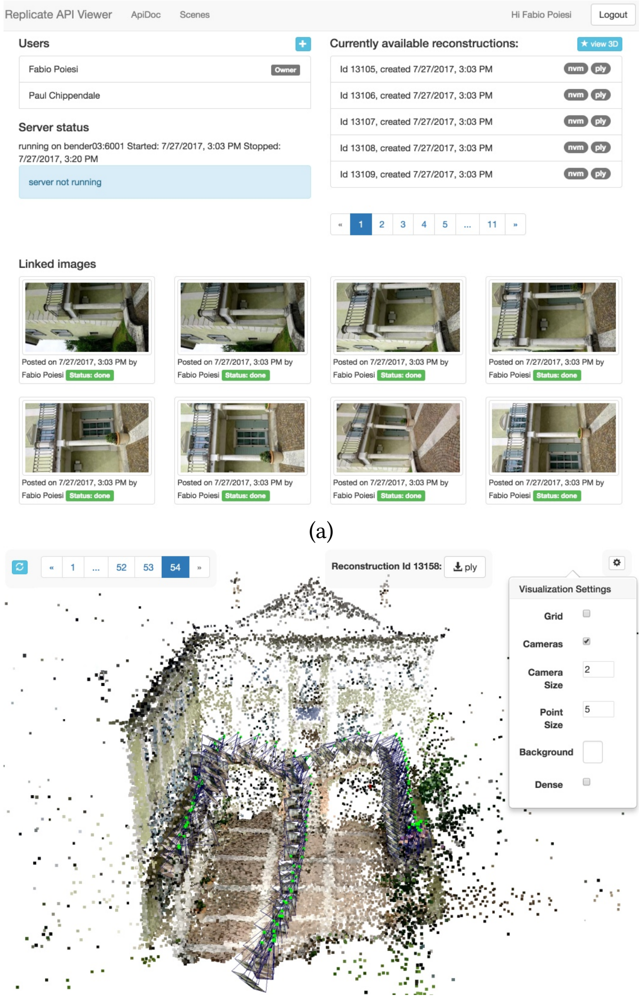
(a)

(b)

Figure 4: Example of collaborative 3D reconstruction. (a) Selected images from the VillaTambosi dataset. (b) Screenshots from two smartphones during the acquisition and reconstructed object. The preview window shows the current point cloud plus the positions of the cameras as green points. The colour-coded point cloud shows the global 3D reconstruction with the position and orientation of the smartphones. The 3D points triangulated by the blue smartphone are coloured cyan, those triangulated by the red smartphone in orange and those triangulated by both in magenta. See video at https://youtu.be/bobWgdLtzIg for more details.

The first dataset (*SarantaKolones*) features an archeological area in Cyprus called 'Saranta Kolones' (i.e. forty columns), which is part of the Pafos archaeological site, listed as a UNESCO World Heritage Site. The size of the archaeological area scanned is approximately 16m×16m×5m. Ten videos (at 30Hz) were recorded using 3 different smartphones held in both landscape and portrait. Due to on-site Internet connection difficulties, the dataset was recorded using the video mode of the smartphones and post-processed later by the same image selection algorithm. We emulated a collaborative scenario, by transmitting the selected images of multiple sequences interleaved to the reconstruction service and evaluated the point clouds as if they were from a live acquisition session.

[1] Dataset webpage: http://tev.fbk.eu/collaborative3D.



Users

Fabio Poiesi

Paul Chippendale

Server status

running on bender03:6001 Started: 7/27/2017, 3:03 PM Stopped: 7/27/2017, 3:20 PM

server not running

Currently available reconstructions:

- Id 13105, created 7/27/2017, 3:03 PM
- Id 13106, created 7/27/2017, 3:03 PM
- Id 13107, created 7/27/2017, 3:03 PM
- Id 13108, created 7/27/2017, 3:03 PM
- Id 13109, created 7/27/2017, 3:03 PM

Linked images

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

Posted on 7/27/2017, 3:03 PM by Fabio Poiesi Status: done

(a)

Reconstruction Id 13158:

Visualization Settings

Grid □

Cameras ☑

Camera Size 2

Point Size 5

Background □

Dense □

(b)

Figure 5: Web-based visualisation. (a) Page of VillaTambosi's reconstruction session. (b) Point cloud displayed along with the available settings.

The other 3 datasets were acquired in Trento, Italy. The second dataset (*PiazzaDuomo*) features the north facing façade of the cathedral in Piazza Duomo. The façade is 100m wide and 30m tall. The third dataset (*CaffeItalia*) features the south facing façade of a painted building in Piazza Duomo. The façade is 30m wide/long and 15m tall. The fourth dataset (*VillaTambosi*) features the courtyard of the historical Villa Tambosi, roughly 14m×10m. These datasets were acquired live using the introduced smartphone application (i.e. the images have passed through the image selector algorithm and were directly uploaded to the reconstruction server for real-time modelling).

Table 1 and 2 summarise the organisation and present details of these datasets.

Table 1: Dataset SarantaKolones description. Key: NoSF: Number of Selected Frames; TNoF: Total Number of Frames; L: Landscape; P: Portrait.

Seq.	Device	Resolution	NoSF/TNoF	Orientation
1	Samsung S6	3840x2160	84/2942	L
2			54/1682	L
3			56/1968	P
4	Huawei P9	1920x1080	152/4884	L/P
5			154/5073	L/P
6			210/7035	L/P
7	OnePlus One		117/4083	P
8			105/4034	P
9			59/2021	P
10			44/1503	P

Table 2: Trento's datasets (PiazzaDuomo, CaffeItalia and VillaTambosi) description. Key: NoSF: Number of Selected Frames; L: Landscape; P: Portrait.

Dataset	Seq.	Device	Resolution	NoSF	Orientation
PiazzaDuomo	1	Samsung Galaxy Alpha	640x480	91	L/P
	2	LG Nexus 5X	1920x1080	64	L/P
	3	Sony Z5		74	L
CaffeItalia	4	Samsung Galaxy Alpha	640x480	218	L/P
	5	LG Nexus 5X	1920x1080	175	L/P
	6	Sony Z5		107	L/P
VillaTambosi	7	LG Nexus 5X	1920x1080	70	P
	8	Sony Z5		85	P

7.2 Evaluation

The datasets were processed with the introduced pipeline to demonstrate the collaborative capabilities of the system. We have quantitatively evaluated the VillaTambosi and SarantaKolones results by analysing (i) the cumulative acquisition time, (ii) reconstructed cloud point evolution, and (iii) a completeness measure.

The *completeness measure* is based on the assumption that the final point cloud is the desired achievement for the reconstruction session. Taking this as a reference, we wanted to observe how quickly it can be completed in percentage terms. Therefore, we computed the completeness $C(t)$ of intermediate point clouds $\mathbf{P}^t$ by analysing the point cloud of the latest reconstruction available $\mathbf{P}^l$ as a reference. We then voxelised the point cloud with a voxel size of $1/2^{11}$ of the reference bounding box. We choose this value as an indicative size for each voxel to understand when the point cloud had grown over the whole space. Each voxel v contains a set of points $\mathbf{P}_v$. We consider a voxel v as being occupied, if v contains at least 5 cloud points (i.e. $|\mathbf{P}_v| \geq 5$). We mathematically define completeness as

$$C(t) = \frac{\sum_v f(\mathbf{P}^t, v) \cdot f(\mathbf{P}^l, v)}{\sum_v f(\mathbf{P}^f, v)} \quad \text{, where} \quad f(\mathbf{P}, v) = \begin{cases} 1 & |\mathbf{P}_v| \geq 5 \\ 0 & \text{otherwise.} \end{cases}$$

7.3 Collaborative Reconstruction

Fig. 6 qualitatively illustrates results pertaining to the reconstructions of CaffeItalia, PiazzaDuomo and SarantaKolones. In these

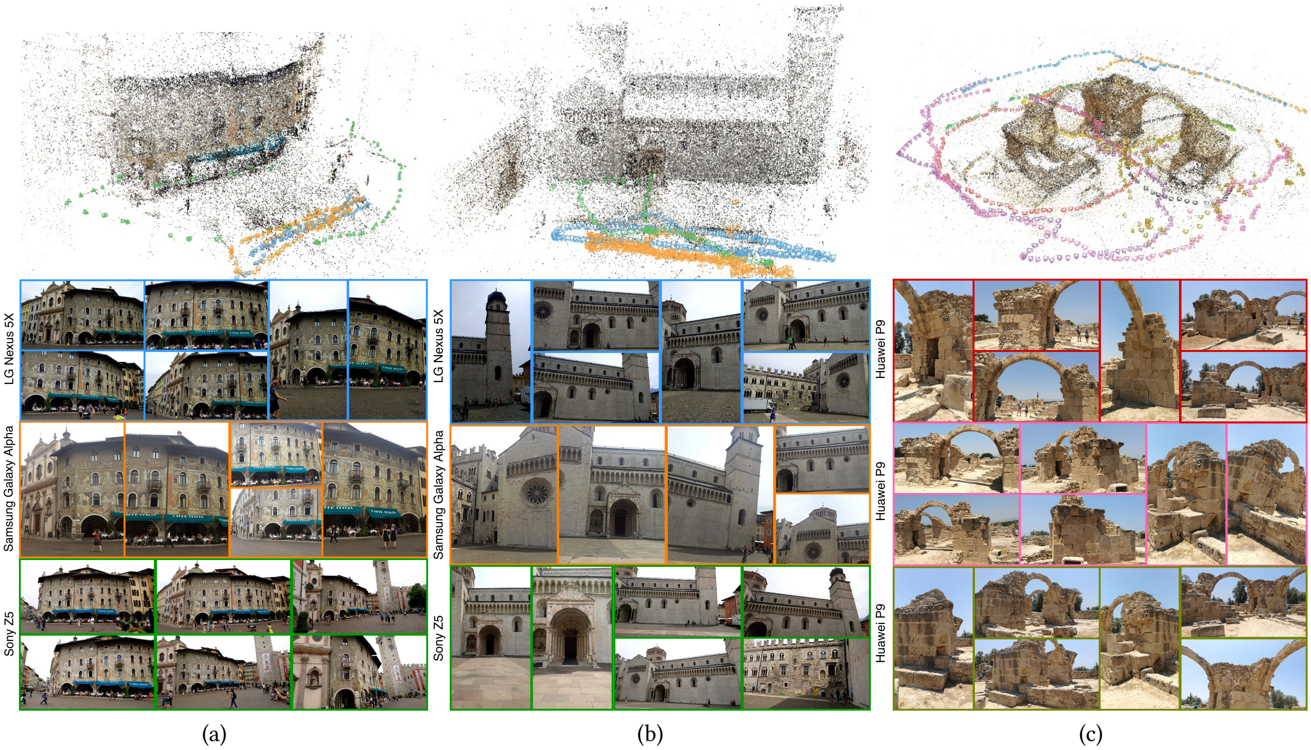


Figure 6: Collaborative reconstructions of buildings: (a) CaffeItalia; (b) PiazzaDuomo; (c) SarantaKolones. The colour coded cameras in each point cloud show the estimated positions and orientations of the selected images transmitted during acquisition. Examples of these transmitted images are displayed under each point cloud. Image boundary colours have been associated to the colour coded cameras in the respective point clouds.

results, we can also observe the trajectories of the different users, whose joint acquisitions led to the three point clouds displayed.

We numerically evaluated the benefits of the collaborative reconstructions for VillaTambosi and SarantaKolones. We numbered the users participating incrementally based on when they started the acquisition.

Fig. 7 shows the evolution of the number of triangulated points and completeness measure over time for the VillaTambosi dataset. From the figure we can see that that user #1 started scanning the scene and user #2 joined after ~20s. User #1 finished their acquisition after ~3 minutes (the number of points triangulated by user #1 remained constant until the end of the reconstruction), user #2 paused their acquisition at ~320s and continued to acquire images from 510 seconds onwards. Thanks to their collaborative effort, the point cloud reached 90% completeness after 300 seconds with around $4.5{\cdot}10^3$ points. With only user #1, the total number of points at the same time would only have been $2.5{\cdot}10^3$.

Fig. 8 shows the order of the submitted sequences in the SarantaKolones dataset. With the introduced pipeline and submission order, the acquisition would have taken around 10 minutes, whereas for a single user, it would have taken 23 minutes. The completeness of the SarantaKolones dataset reached 80% at about 4 minutes and using data from three different sequences. As can be seen, completeness is not monotonically increasing. This is due to the fact that sometimes by adding more images the system can completely change its camera configuration estimate leading to results that might be less complete than previous models. As soon as the connectivity of the view graph is sufficiently high, camera poses get more stable and completeness saturates. In Fig. 9 we can observe the evolution of the point cloud with time.

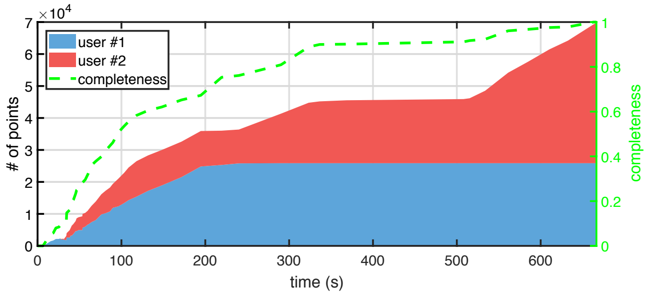


Figure 7: Temporal evolution of the number of points in the VillaTambosi cloud that each user contributed to. The colour of each user is associated to the colours of the smartphones in Fig. 4.

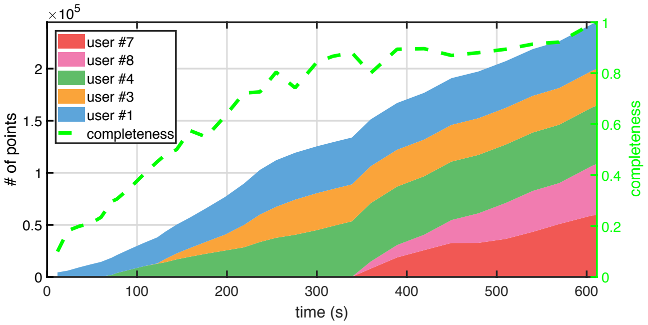


Figure 8: Temporal evolution of the number of points in the SarantaKolones cloud that each user contributed to.

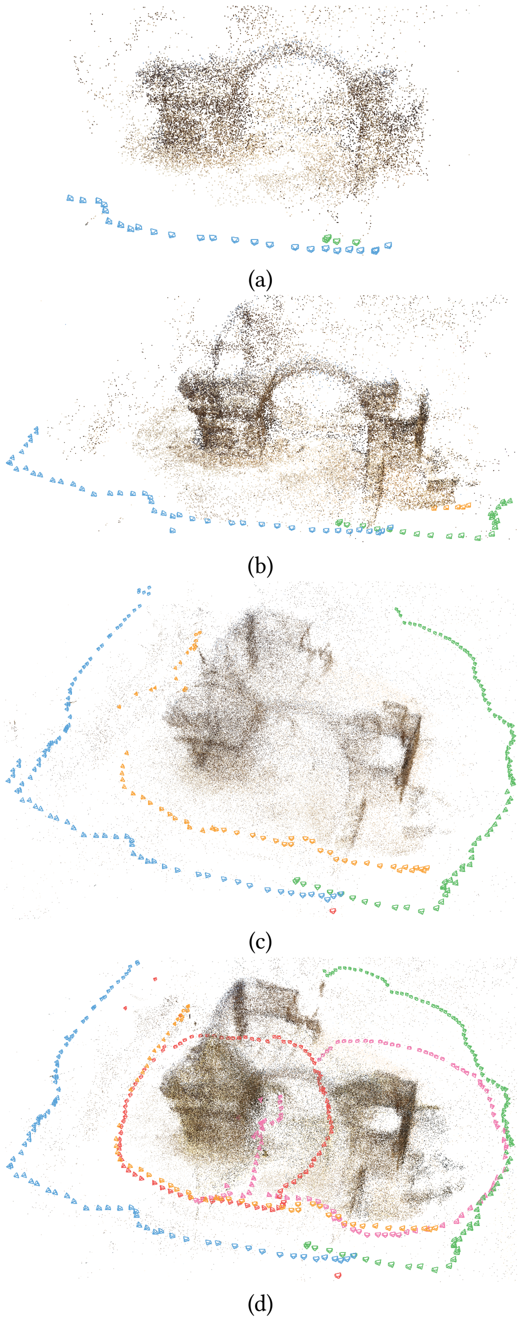
7.4 Status Feedback Responsiveness

We evaluated system responsiveness by comparing the time it took for users to receive point cloud previews after their images had been uploaded to the server; naturally this is also linked to network bandwidth as well as server processing time. For example, the point cloud of the final reconstruction of VillaTambosi is 3MBytes, therefore there will be a variable latency depending on network connection type (3G, 4G or WiFi).

Fig. 10 shows a graph of image upload time vs. the availability of 3D reconstructions, measured during the VillaTambosi experiment (Fig. 4). We considered all of the images, irrespectively of the user who uploaded them, to focus our analysis on the reconstruction time with respect to the image to process. Time zero corresponds to the time instant when the first image was uploaded. From the graph we can see how intermediate reconstructions are quickly computed after new images are uploaded to the server. Before acquisition interruption at time instant 400s, the time interval between the last uploaded image and the last reconstruction is 20s. At that time, the server did not generate other reconstructions, which means that all of the images had been processed up to that time. The gap between 400s and 500s was intentionally created during our experiment to illustrate the collaborative capability of our pipeline. During this experiment, the red user in Fig. 4b communicated to the other user to acquire more images of one part of the building as it appeared sparser than others on their app previe window. The effect of this continuation led to an increased number of points in the cloud of the right-hand side of the building. This can also be seen in Fig. 7 after time instant 500s, where the point cloud generated by the acquisition of user blue increases.

8 CONCLUSIONS

This article presented a 3D reconstruction pipeline that enables multiple users to collaboratively acquire images of an object resulting in a single 3D point cloud. Smartphones can be used to acquisition images and an app automatically select the best of them to transmit to a reconstruction server. The server processes these images with an incremental Structure from Motion algorithm to generate a 3D point cloud. The design of our pipeline enables users to coordinate a reconstruction session as they can concurrently visualise the joint 3D point cloud via a previewer on their smartphones. Image uploads from multiple smartphones are handled by the server with a multi-threaded design that performs local Bundle Adjustments


(a)

(b)

(c)

(d)

Figure 9: Collaborative reconstruction of SarantaKolones at intermediate stages. Reconstruction progress after (a) 72s, (b) 131s, (c) 318s and (d) 611s. The colour-coded smartphone cameras in each reconstruction show the positions and orientations of selected images during acquisition, and are associated to the colours of the users in Fig. 8.

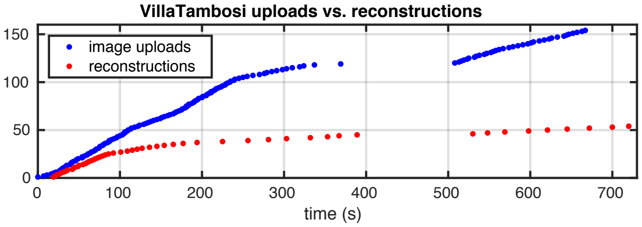


Figure 10: Image upload and reconstruction time. As the number of images uploaded to the server increases, the reconstruction grows and new models are generated.

based on intrinsic groups. We showed how the proposed pipeline can effectively manage multiple users and speed up the acquisition process via experiments carried out in real-world scenarios.

Future research directions will involve the implementation of a lightweight 6DoF pose tracking and 3D reconstruction algorithm running on the smartphone, similar to [19], to create an Augmented Reality-based guidance for the user during image acquisition. We are working towards the full integration of a progressive Multi View Stereo method inside our pipeline to provide users with dense point clouds as reconstruction previews. Moreover, we will develop a semi-automatic editing tool to remove noisy cloud points prior to the generation of object meshes.

ACKNOWLEDGEMENTS

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 687757.

REFERENCES

[1] R. Arandjelovic and A. Zisserman. 2012. Three things everyone should know to improve object retrieval. In *Proc. of Computer Vision and Pattern Recognition.* Providence, US.

[2] S.-Y. Bao and S. Savarese. 2011. Semantic structure from motion. In *Proc. of Computer Vision and Pattern Recognition.* Colorado Springs, US.

[3] P.E. Carbonneau and J.T. Dietrich. 2017. Cost-effective non-metric photogrammetry from consumer-grade sUAS: implications for direct georeferencing of structure from motion photogrammetry. *Earth Surface Processes and Landforms* 42, 3 (Mar. 2017), 473–486.

[4] J. Engel, T. Schops, and D. Cremers. 2014. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Proc. of European Conference on Computer Vision.* Zurich, CH.

[5] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. 2013. Collaborative monocular SLAM with multiple Micro Aerial Vehicles. In *Proc. of Intelligent Robots and Systems.* Tokyo, JP.

[6] R. Gherardi, M. Farenzena, and A. Fusiello. 2010. Improving the efficiency of hierarchical Structure-and-Motion. In *Proc. of Computer Vision and Pattern Recognition.* Colorado Springs, US.

[7] R.I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision.* Cambridge University Press.

[8] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. 2009. Turning mobile phones into 3D scanners. In *Workshop in Computer Vision and Pattern Recognition.* Miami, US.

[9] A. Irschara, C. Zach, and H. Bischof. 2007. Towards wiki-based dense city modelling. In *Proc. of International Conference on Computer Vision.* Rio de Janeiro, BR.

[10] ItSeez3D. 2017. (Aug. 2017). http://www.itseez3d.com

[11] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. 2014. Turning mobile phones into 3D scanners. In *Proc. of Computer Vision and Pattern Recognition.* Columbus, US.

[12] A. Locher, M. Perdoch, H. Riemenschneider, and L. Van Gool. 2016. Mobile Phone and Cloud - a Dream Team for 3D Reconstruction. In *Proc. of Winter Conference on Applications of Computer Vision.* Lake Placid, US.

[13] D.G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 2, 60 (Nov. 2004), 91–110.

[14] J.G. Morrison, D. Galvez-López, and G. Sibley. 2016. MOARSLAM: Multiple Operator Augmented RSLAM. In *Proc. of Distributed Autonomous Robotic Systems.* London, UK.

[15] O. Muratov, Y. Slynko, V. Chernov, M. Lyubimtseva, A. Shamsuarov, and Victor Bucha. 2016. 3DCapture: 3D Reconstruction for a Smartphone. In *Computer Vision and Pattern Recognition Workshops.* Las Vegas, US.

[16] D. Nister. 2004. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26, 6 (Apr. 2004), 756–770.

[17] D. Nister and H. Stewenius. 2006. Scalable Recognition with a Vocabulary Tree. In *Proc. of Computer Vision and Pattern Recognition.* New York, US.

[18] E. Nocerino, F. Lago, D. Morabito, and F. Remondino et al. 2017. A Smartphone-based pipeline for the creative industry - The REPLICATE project. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* Nafplio, GR.

[19] P. Ondruska, P. Kohli, and S. Izadi. 2015. MobileFusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. *IEEE Trans. on Visualization and Computer Graphics* 21, 11 (Nov. 2015), 1251–1258.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Proc. of Computer Vision and Pattern Recognition.* Minneapolis, US.

[21] V.A. Prisacariu, O. Kahler, D.W. Murray, and I.D. Reid. 2015. IEEE Trans. on Visualization and Computer Graphics. *Real-time 3D tracking and reconstruction on mobile phones* 5, 21 (May 2015), 557–570.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. 2011. ORB: An efficient alternative to SIFT and SURF. In *Proc. of International Conference on Computer Vision.* Sydney, AU.

[23] P. Schmuck and M. Chli. 2017. Multi-UAV collaborative monocular SLAM. In *Proc. of International Conference on Robotics and Automation.* Singapore.

[24] J.L. Schonberger and J.-M. Frahm. 2016. Structure-from-Motion Revisited. In *Proc. of Computer Vision and Pattern Recognition.* Las Vegas, US.

[25] T. Sieberth, R. Wackrow, and J.H. Chandler. 2016. Automatic detection of blurred images in UAV image sets. *Journal of Archaeological Science* 122, 12 (Dec. 2016), 1–16.

[26] N. Snavely, S.M. Seitz, and R. Szeliski. 2008. Modeling the world from Internet photo collections. *International Journal on Computer Vision* 80, 2 (Dec. 2008), 189–210.

[27] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys. 2015. Optimizing the viewing graph for Structure-from-Motion. In *Proc. of Computer vision and Pattern Recognition.* Boston, US.

[28] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. 2013. Live metric 3D reconstruction on mobile phones. In *Proc. of International Conference on Computer Vision.* Sydney, AU.

[29] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. 2000. *Bundle adjustment - a modern synthesis.* Vision Algorithms: Theory and Practice, Springer-Verlag, Berlin, GE.

[30] TRNIO. 2017. (Aug. 2017). http://www.trnio.com

[31] O. Untzelmann, T. Sattler, S. Middelberg, and L. Kobbelt. 2013. A scalable collaborative online system for city reconstruction. In *Workshop in International Conference on Computer Vision.* Sydney, AU.

[32] C. Wu. 2013. Turning mobile phones into 3D scanners. In *Proc. of 3D Vision.* Tokyo, JP.

[33] S. Zhang, J. Shan, Z. Zhang, J. Yan, and Y. Hou. 2016. Integrating smartphone images and airborne LIDAR data for complete urban building modelling. In *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* Prague, CZ.

[34] J. Sivic; A. Zisserman. 2003. Video Google: a text retrieval approach to object matching in videos. In *Proc. of International Conference on Computer Vision.* Nice, FR.