Machine Vision and Applications
DOI 10.1007/s00138-015-0699-5

ORIGINAL PAPER

Resource aware and incremental mosaics of wide areas from small-scale UAVs

Daniel Wischounig-Strucl[1] · Bernhard Rinner[1]

Received: 7 December 2013 / Revised: 13 January 2015 / Accepted: 18 June 2015

Abstract Small-scale unmanned aerial vehicles (UAVs) are an emerging research area and have been recently demonstrated in many applications including disaster response management, construction site monitoring and wide area surveillance where multiple UAVs impose various benefits. In this work we present a system composed of multiple networked UAVs for autonomously monitoring a wide area scenario. Each UAV is able to follow waypoints and capture high-resolution images. In order to overcome the strong resource limitations we implement an incremental approach for generating an orthographic mosaic from the individual images. Captured images are pre-processed on-board, annotated with other sensor data and transferred by a prioritized transmission scheme. The ultimate goal of our approach is to generate an overview mosaic as fast as possible and to improve its quality over time. The mosaicking exploits position and orientation data of the UAV to compute rough image projections which are incrementally refined by scene structure analysis when more image data is available. We evaluate our incremental mosaicking in the strongly resource limited UAV network composed of up to three concurrently flying UAVs. Our results are compared to state-of-the-art mosaicking methods and show a unique performance in our dedicated application scenarios.

Keywords Resource-aware · Mosaics · Orthophoto · UAV · Drone · Network · Scheduling · 3D reconstruction · Bundle adjustment · Structure-from-motion · Unmanned aerial vehicle

1 Introduction

There are many applications which require an overview in our public life. In some aspects aerial photography from airplanes satisfies such requests, but often more details, more recent views or different perspectives of scenes are requested.

Among such ambitious scenarios we find the monitoring of large construction sites where frequent flights and many different view points allow a detailed progress monitoring. In sensitive situations during response management after severe disasters we cannot resort to existing cameras or (communication) infrastructures. Areas may be restricted or inaccessible for manned vehicles. But an overview mosaic generated from individual images is required virtually in an instant to successfully complete required missions.

In our main use case of emergency and disaster response the overall goal is to build overview mosaics from large unknown areas quickly. We employ multiple small-scale quad-rotor unmanned aerial vehicles (UAVs) that are able to vertically take-off and land (VTOL) concurrently to improve the time of coverage.

The usage of such easy to operate autonomous aerial sensing platforms since human resources are limited in emergency situations. The coordination of the UAVs, the data transmission and the quick and efficient overview mosaic generation is still an open topic in many state-of-the-art image processing methods. To successfully transmit, mosaic and merge high resolution images, we propose an incremen-

✉ Daniel Wischounig-Strucl
dws@strucl.com

Bernhard Rinner
bernhard.rinner@aau.at

[1] Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria

Published online: 11 July 2015

tal image processing and mosaicking where we are able to increase the mosaic quality over time.

1.1 UAV system

Small-scale UAV platforms have been introduced in the past by companies such as Ascending Technologies[1] or Microdrones.[2] They offer battery powered devices with different control and sensing capabilities. With a take-off weight between one and five kilograms the UAVs can still carry sufficient payloads such as high resolution cameras and operate for 12–45 min. The presented system is designed to be deployed on any kind of small-scale UAV platform considering adaptations to UAV specific commands. Our UAVs incorporate two processing units with various sensors, such as inertial measurement unit or global navigation systems, and at least one camera.

1.2 Contribution

The contribution of this work covers the resource aware and incremental improvement of an overview mosaic from high resolution images transmitted via an aerial network from small scale UAVs. This approach fills the gap between panorama mosaics and expensive full 3D reconstructions by delivering results in an increasing quality in a very short time. Therefore the flight routes are optimized to reduce redundant data, such as overlap between images, which ends up in a challenge for the mosaicking approach, that is addressed here.

1.3 Outline

The remainder of this article provides details on the deployed UAV system in Sect. 3 after discussing related work in Sect. 2. First in Sect. 4.1 we present the prioritized image transfer and second the incremental registration in Sect. 4.2. Section 5 evaluates real world case studies and Sect. 6 concludes this work.

2 Related work

In recent times, aerial photogrammetry technologies expanded to higher altitudes such as satellites that provide already high resolution photos of the earth surface, unfortunately in no-real-time. Typically, individual images are taken from a single airplane and processed offline after landing. We studied related works of (aerial) camera networks and online mosaicking for wide area application which has been rarely investigated before. In the work of Akyildiz et al. [2] wireless sensor networks built from off-the-shelf cameras are able to ubiquitously retrieve video and still images from the environment. A clear trend is to use state-of-the-art communication interfaces within camera networks with all their advantages and drawbacks. The evolution reaches from single cameras on UAVs flying at high altitude (e.g, [1,9]) to networks of cameras also deployed at low altitudes (e.g, [12,23]). When live data streaming is necessary UAV camera networks require more complex and active communication links to transmit the sensed data. To achieve live updated images after severe disasters the work of Pratt et al. [21] presents individually and manually steered UAVs to transmit live images from the scene.

Moreover, in the project AggieAir [11] two separated network architectures are employed. One link is used for control data such as manual steering and the other one is utilized for the transmission of sensed data with higher bandwidth. Besides the networking challenges, most of the aerial mosaicking approaches, rely on orthogonal images from high altitudes where the structure of the scene is negligible. On the contrary, in the online aerial mosaic generation proposed by Turkbeyler et al. [28] images from low altitude are treated as orthogonal and the image transformations are estimated by employing the homography. This results in distortions when the covered area contains a structured scene, as explained and compared in our previous work [30].

The recent review [10] presents an detailed overview of UAV technologies, the available payload and consequent fields of applications. In other domains, such as underwater mapping, unmanned vehicles are well established [13]. Recently multiple vehicles are combined to increase their resource efficiency and operating range. In this work we are addressing the combination of efficient data transmission and mosaicking from low altitudes.

3 Overview

Our system mainly deals with the response management after severe disasters where mosaics should be generated as fast as possible from images captured by small-scale, low-altitude UAVs. Figure 1 shows the individual components are presented where weak points arise in a typical image processing chain.

We investigate why an incremental strategy that considers the scene structure is potentially more successful than traditional offline mosaicking methods.

Hence, we divide the mosaicking process into multiple stages that reuse already processed data and present intermediate output mosaics after each step (Fig. 2).

[1] http://www.asctec.de/ last visited November 2013.

[2] http://microdrones.com/ last visited April 2013.

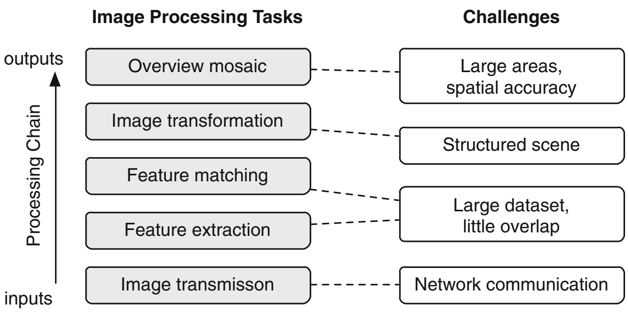

Fig. 1 Key processing steps and their associated challenges

We have designed a system that covers three main components: First the highly mobile small-scale UAVs equipped with various sensors and processing units; second the wireless communication network for control and data transmission; and third, the ground station where routes are planned, the data is collected, processed and visualized. In Fig. 3 the light grey arrow in the background sketches the data flow of our process through the blocks marked in green which are covered by this work on the application layer. On the left side multiple user interfaces are shown. One interface allows the operators to input their requirements. Another interface is available for observing the results.

3.1 Mission

The main requirements for a *mission* are a bundle of available resources, i.e, multiple participating heterogeneous UAVs, the area to cover and constraints required by the application.

The user specifies the time to complete the mission and the temporal and spatial target resolution. From these inputs, predefined routes are generated and included as mission plan into the mission. Hence, the route planning itself is an emerging research topic covered by other works such as Mersheeva et al. [20]. In this work we accept already generated route plans with dedicated picture points, which are 3D locations in the world coordinate system, where images shall be taken. The planned routes are sent to the UAVs via the communication network.

Figure 3 depicts two scenarios for our mosaicking system. In our first scenario, a fire fighter practice scenario we planned five consecutive missions by a single UAV to update the final overview image frequently by more recent data. The flight time of each route was about 620 s over the area of about 12,000 m^2. In a second scenario three UAVs complete one mission concurrently. The whole area of more than 45,000 m^2 is covered in less than 500 s. Since, a maximum flight time of any of our UAVs in one mission is not allowed to exceed 840 s. After completing a flight each UAV autonomously lands and stays active to complete unfinished tasks.

3.2 Networking

In wide area scenarios we cannot rely on an existing communication infrastructure with sufficient bandwidth available for transmitting high resolution images. The typical period between capturing images is about 10–15 s on an individual UAV. But the amount of raw data exceeds more than 10 MB per image and we have multiple UAVs in operation concurrently. Because of the limited resources on the UAV and requirement of online mosaicking an efficient strategy needs to be realized to process and transmit the captured data.

We utilize a wireless LAN infrastructure based on IEEE 802.11a at 5 GHz with three antennas attached in multiple input and multiple output mode (MIMO). The antenna setup is optimized to emphasize on best connectivity, even during motion and tilt of the UAV [31]. Extensive tests have shown that the deployment of elected off-the-shelf wireless LAN components deliver a stable communication within the proposed scenario, but with limited rates. There the max-

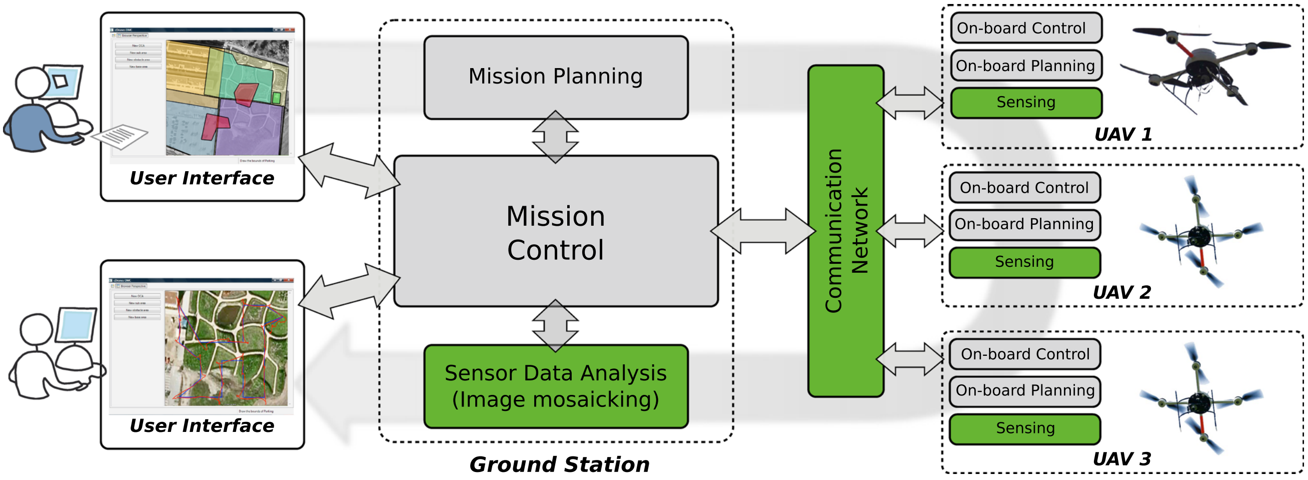

Fig. 2 Our system is composed by the small scale UAVs, the communication network and the ground station with its GUI

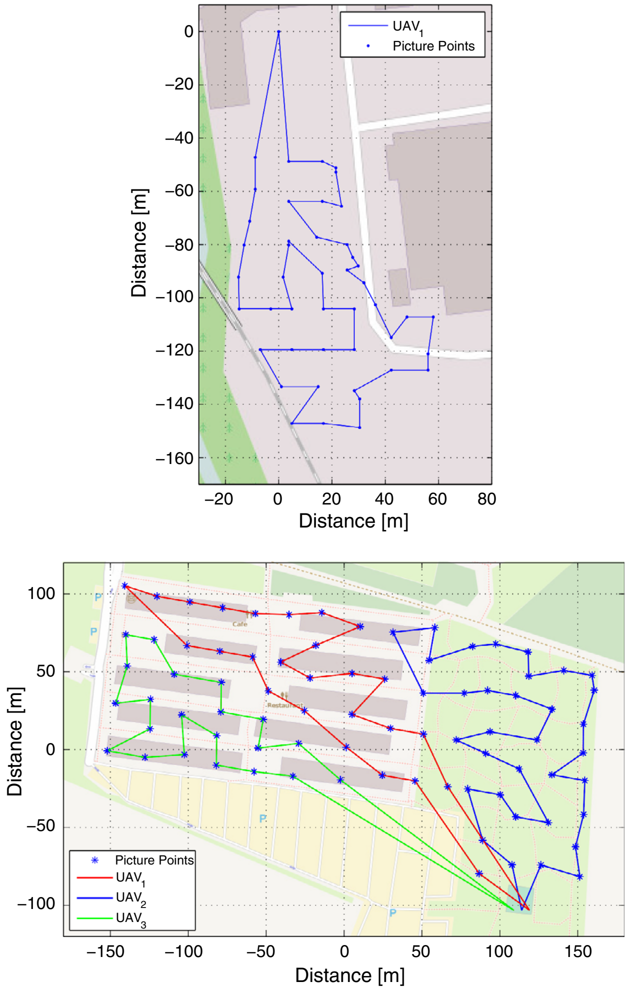

Fig. 3 The planned routes for the single UAV scenario (*above*) and the multi-UAV scenario (*below*)

imum achievable data rate is 54 Mbit/s in close regions. For independence of any existing wireless infrastructure, we deploy our own wireless base station and mount it at an exposed position to increase network coverage and reliability.

Due to bandwidth limitations and variations the transmission of complete high-resolution images is time consuming. Limitations emerge from the scenario requirements of covering a wide area in a very short time. Bandwidth limitations result from non-existing infrastructures, long distances or occlusions due to the scene structure. On the other hand, variations in the available bandwidth also occur due to the dynamics in the mission. If UAVs fly along their routes they interfere each other depending on their distance and location to the ground station.

3.3 Imaging

On our small-scale UAVs we utilize high resolution cameras for RGB still images, such as lightweight off-the-shelf digital consumer cameras with resolutions up to 12 megapixels and remote control capabilities. The cameras are mounted on our adaptive camera mount which is stabilized and able to adjust its view angle. This camera is directly connected via USB to the onboard processing unit, an Intel® Atom™ embedded unit for controlling of the capture settings and triggering the shutter.

The image processing on-board the UAV is implemented using shell and Python scripts and the Kakadu JPEG2000 library[3] while mathematical operations are implemented utilizing different libraries, such as approximate nearest neighbors (ANN) [4], basic linear algebra subprograms (BLAS) [5] and the linear algebra package (LAPACK) [3]. During our demonstrations we have deployed one Pentax A40 RGB compact camera with 12 megapixel resolution and modified Canon PowerShot S80 compact cameras with 8 megapixel.

4 Efficient overview image generation

For efficient and resource aware mosaicking we pre-process and annotate images with meta-data already on the UAV. This data is transmitted for immediate presentation to the ground station where it is merged to an initial mosaic. The transmission is based on data prioritization to transfer important data first in limited bandwidth scenarios. Already processed results from previous mosaics are integrated to reduce overall processing effort only to fresh data. According to our assumptions to have further image resolutions or structure data available we aim for improving the quality over time. Fresh image data is defined to be either images of uncovered areas that are not transmitted before or higher resolution representations of already covered areas. Basically three individual steps in our incremental mosaicking are considered for quality improvement:

A Meta-data based mosaic is generated by simple image placement using only meta-data, such as UAV positions $\mathbf{t}_c$, orientations $\mathbf{R}_c$ and intrinsic camera parameters $\mathbf{K}_c$.

Feature based mosaicking is executed with feature extraction and registration of images by a similarity transformation if an immediate improvement is required by the application. We are reducing the overlapping image areas already during planning to reduce redundant data during transmission. This reduces the quality but is efficient as first part of the image-data based mosaicking.

Structure based mosaicking is the last leap of improvement when employing the structure analysis of the scene. A sparse 3D model is constructed from the extracted key-

[3] Kakadu JPEG2000 Encoder, http://www.kakadusoftware.com; last visited on April 2nd 2013.

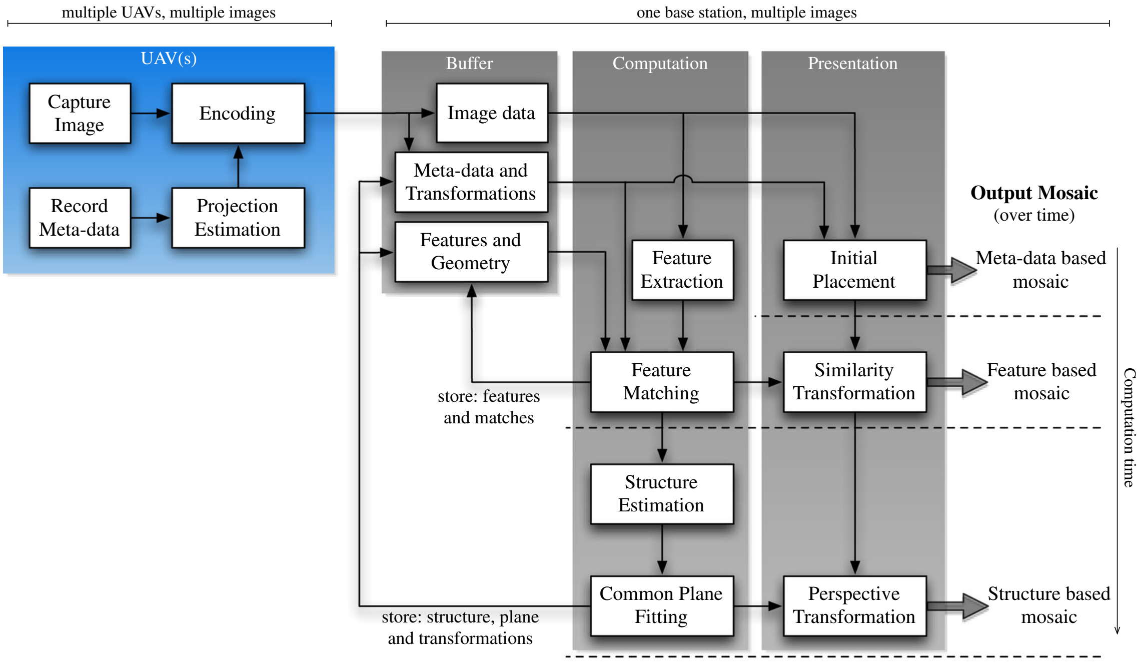

Fig. 4 This block chart sketches the incremental mosaicking process at the base station and the principle blocks on each UAV

points X_i and we estimate a common projection plane within this 3D structure. Images are mapped on to this common ground plane, which concludes the image-data based mosaicking.

On the ground station the pre-processed data from the UAVs is collected and images are placed in their available resolution representation immediately on the mosaic.

$$\mathbf{T}_i = \begin{cases} \mathbf{T}_{\text{cam}_i} = f(\mathbf{t}_c, \mathbf{R}_c, \mathbf{K}_c) & \text{if } q_{\text{image}} < \gamma, \\ \mathbf{T}_{\text{image}_i} = f(\mathbf{T}_{\text{cam}_i}, X_i) & \text{if } q_{\text{image}} \geq \gamma. \end{cases} \tag{1}$$

Common ways to compute image transformations $\mathbf{T}_i$ in Eq. 1 for an individual image $\mathbf{I}_i$ are (a) the direct geometric projection from the camera extrinsic and intrinsic parameters $\mathbf{T}_{\text{cam}_i}$, or (b) the more time consuming estimation based on the image data known as image registration. The transformation function f from described in Sect. 4.2. The image-data based mosaicking $\mathbf{T}_{\text{image}_i}$ is applied if sufficient image data is available or the individual images are available with a better quality measured by q_{image}, later elaborated in Eq. 5. Typically, we define the threshold $\gamma = 0.5$ but this parameter is a function of the application constraints. γ should be increased if a higher quality is preferred.

In Fig. 4 the basic processing blocks and layers of our mosaicking are presented. On the left the internal sensing units on the UAV platform encode data and transmit it to the base station. Data is collected and organized with already processed data in the *buffer*. In the *computation* layer the different processing stages are covered while the *presentation* layer executes the image transformations and provides different quality levels of mosaics. Apparently, these outputs depend on the achieved computation results.

Immediately, after receiving image data it is directly presented as meta-data based mosaic through the *presentation* layer. Based on these transformations neighboring and overlapping images are determined for further processing steps. An improved mosaic can be generated by registering images if the available resolution of the received image is sufficient. In the last improvement stage the extracted features are input to the structure estimation and can provide a final mosaic which considers image transformations computed from the 3D scene structure.

4.1 Prioritized image transfer

In wide area scenarios, such as disaster scenarios we cannot rely on an existing infrastructure with sufficient bandwidth available for transmitting high resolution images. Throughout the mission multiple UAVs capture more and more images. The typical period between capturing images is about 10–15 s. But the amount of raw data exceeds more than 10 MB per image which is challenging to transmit. The limited resources on the UAV require an efficient strategy to process and transmit the captured data.

On the UAV images or fragments of images are prioritized before transmission according to their forecast benefit to the whole overview mosaic. Basically, data from newly explored areas, which can be a complete image or a rectangular fragment of one image, has the highest priority and should be

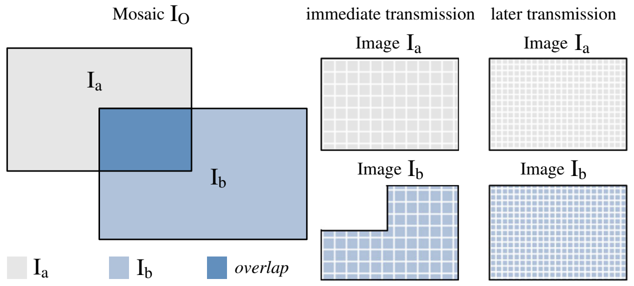

Fig. 5 Data separation of two overlapping images. The highest priority is assigned to new data with low resolution. The complete images are scheduled in higher resolutions later for transmission

transferred immediately. The route and coverage planning divides the total area into a grid of elements G_k of a square of the size g^2. For scenarios of an area to cover of dozens of hectares the planning component prefer grid elements of a dimension of $g = 5\,m$. This dimension is the smallest unit and is application specific because it has an impact on the computational effort [22].

For efficient scheduling we split the image to patches according to the grid size. By the meta-data of the UAV we are able to determine if one grid element is visible in the image completely or which portions of the image share the covered area. The highest priorities for new images are assigned according to the number of grid elements N_G newly covered and not contained by any other image on this UAV. Individual fragments G_k of the size g^2 are scheduled by the priority p_{k_i} for each image $\mathbf{I}_i$. The priorities are related to the number of participating images covering the same area beforehand and the total number of images N, Eq. 2. If more than 50 % of the whole image area $A(\mathbf{I})$ is newly covered, the whole image is considered. This is defined by $p_{\mathbf{I}}$ in Eq. 3.

$$p_{k_i} = N - |\{\mathbf{I}_i | G_k \in A(\mathbf{I}_i)\}| \tag{2}$$

$$p_{\mathbf{I}} = \sum_{i=0}^{\frac{A(\mathbf{I}_i)}{g^2}} p_{k_i} \tag{3}$$

We have developed a scheduling scheme that orders the priorities of image data and utilizes region of interest (ROI) encoding before transmission. On each UAV it considers the UAVs meta-data and already transmitted image data. In Fig. 5 an example of two overlapping images is presented where image $\mathbf{I}_a$ is captured first. It covers only new areas it is completely scheduled with a high priority. The image area is not cropped but scheduled at a low resolution, sketched by the larger grid. For image registration on the ground station higher resolutions are required. Thus, the remaining image data in higher resolutions is scheduled for transmission with a lower priority later.

The second image $\mathbf{I}_b$ is captured a few seconds later. The data of the newly covered area is determined by the meta-data projection and is scheduled with the highest priority again. Redundant areas are mainly required for the image registration but are not essential for the quick output generation.

The described assignment of different priorities a-priori can be efficiently mixed in one image. For example, important regions within one image are encoded with a higher bit rate than already covered areas from other images [6].

4.1.1 Progressive image encoding

In our approach we exploit JPEG2000 motivated by the work of Frescura et al. [15] who presented a wireless network with JPEG2000 image transmission. JPEG2000 encoding serves various features such as ROI encoding, resolution or layer progressive encoding, and tiling. Images are split into lower resolution representations during the encoding of high resolution still images with JPEG2000. In progressive image encoding the number of intermediate resolutions or quality levels is defined by the applied method and the size of each image. In JPEG2000 encoding we can arbitrarily define the number of quality layers. The lowest resolution representation in the lowest quality is allocated at the beginning of the JPEG2000 stream followed by the remaining resolutions. In general the packets of one JPEG2000 image can be interleaved by four different ways, i.e, ordered by resolution (R), position (P), component (C), or in layers (L). The primary selector employed in this work is the resolution resulting in a progression order denoted as RLCP.

Each image is received at the ground station at a resolution κ_I and validated against the resolution requests for the image-data based mosaicking. If κ_I is within the interval $[\kappa_L, \kappa_U]$ it is included to the current patch for image registration. This interval is application specific and depends on the timing and resolution constraints of the applications. We allow to constrain two parameters: (a) the target resolution which directly represents the value κ_U and (b) the penalty from $pe = [0.0, 1.0]$ of not achieving the target resolution. This penalty defines the relation of κ_L to κ_U as $\kappa_L = pe \cdot \kappa_U$. The maximum resolution of the full sized image is denoted as κ_F. In typical disaster response scenarios we define κ_U to be 800 px. On the base station the received resolutions are evaluated to a resolution quality, to decide if one image is included into the image-data based mosaicking or just placed by its meta-data.

$$q_{res} = \begin{cases} 0 & \text{if } \kappa_I < \kappa_L, \\ \frac{1}{2}\,\frac{\kappa_I - \kappa_L}{\kappa_U - \kappa_L} & \text{if } \kappa_I <= \kappa_U, \\ \frac{1}{2} + \frac{\kappa_I - \kappa_U}{2(\kappa_F - \kappa_U)} & \text{if } \kappa_I > \kappa_U. \end{cases} \tag{4}$$

When receiving the JPEG2000 bit stream, image fragments are concatenated to decode the data. The image

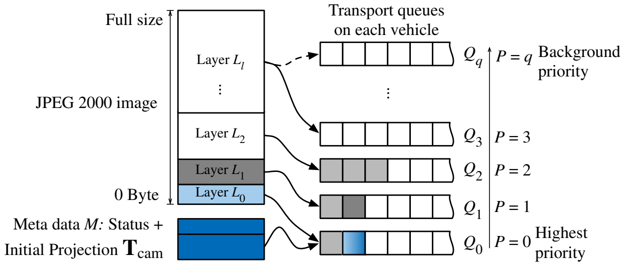

Fig. 6 One single image split into layers containing different resolutions of one image that are en-queued in prioritized queues for transmission

resolution layers are decomposed to determine the maximum image resolution for the mosaicking and visualization. The benefit of this resolution is represented by resolution quality q_{res} in Eq. 4.

If the current resolution is less than κ_L the quality representative $q_{res} = 0$. Such images are not eligible for the image-data based registration and are only placed by their meta-data transformation $\mathbf{T}_{\text{cam}}$.

4.1.2 Application layer scheduling

Basically, images are split into fragments and prioritized according to their resolution and qualities. In our previous work [29] we evaluated the fair use of the available bandwidth. On each UAV the images are progressively encoded by JPEG2000 and buffered. The initial priorities for the whole image $p_{\mathbf{I}}$ and for regions p_{g^2} in the image are processed by the JPEG2000 resolution layer encoding. Thus, each encoded resolution layer represents the resolution related to the assigned priority $p'_{\mathbf{I}} = p_{\mathbf{I}} \frac{\kappa_I}{\kappa_F}$ and is enqueued for transmission.

The highest priority is assigned to image fragments of the lowest resolutions and uncovered regions, determined earlier in Eq. 2. Image fragments of the next higher resolution and lower priority are attached to the second highest priority queue as demonstrated in Fig. 6. The first packet is denoted as layer L_0 and includes the basic image data combined with all JPEG2000 headers and the meta-data M. The meta-data contains the rough transformation estimation $\mathbf{T}_{\text{cam}}$ besides some status information. This data is required for continuous monitoring and immediate feedback to the global planning at the ground station. For example, to observe uncovered areas by inspecting the meta-data projections and to trigger a re-planning. Since our scheduling is assigned to the application layer it can be built on top of any transport protocol. In our approach we are using UDP as transmission protocol because it performs well in the utilized wireless LAN infrastructure and is well integrated in the hardware and system components of our UAV systems [19].

4.1.3 Prioritized transmission queues

Our transmission scheme conducts q queues with different priorities, sketched in Fig. 6 where we distinguish between the important data for the first mosaic and quality improvements. The number of image layers l can be different for heterogeneous UAVs and different images. The scheduling is managed on the UAV in the manner of transmitting and emptying higher prioritized queues, i.e, Q_0 and Q_1, before transmitting the remaining image layer data from lower prioritized queues. If the image layer L_2 already contains the minimum resolution κ_L the image-data based mosaicking can be integrated. For every new image the meta-data M and lowest resolution L_0 is put to the highest priority queue Q_0. The data transmission of one queue element should not be interrupted by any other transmission. However, an image layer of a large size should not be enqueued entirely, but divided into smaller chunks added to the same queue. By this approach we can interrupt the transmission of lower prioritized queues at any point (depending on the chunk size, e.g, 50 kB) if a higher prioritized queue is filled meanwhile. Otherwise, in low bandwidth scenarios the transmission of a larger image layer would block more important data if the transmission is stalled.

Our scheme is designed to be robust against link failures due to limited communication range caused by obstacles, long distances or the number of concurrently active UAVs. Missing packets of one data chunk are optionally requested after the transmission of the corresponding chunk. This consolidated kind of packet re-transmissions is efficient in wireless networks since it does not require a transmission control in the underlying network layer and still goes well along the JPEG2000 decompression, if single packets are mission of one chunk.

4.2 Incremental registration

The core of the incremental mosaicking is the iterative refinement of the output image by considering different kinds of data incrementally received via our network scheduling and by applying different mosaicking methods. This method is sketched in Algorithm 1 which is triggered on any newly arriving image-data. The simplest method to generate a mosaic from this data is to compute the transformation from averaged UAV meta-data and use a direct geometric projection. This method is initially applied to every newly incoming image extremely quick. If the current image $\mathbf{I}_i$ is available in a sufficient quality for further processing it is mosaicked by well-known image registration methods (feature extraction and feature matching). After feature extraction on individual images, we improve the processing time by determining neighboring images before matching images to reduce the computational effort. The resulting image transformations replace the previously estimated geometric projections and can update the mosaic by new image transformations $\mathbf{I}_i$. In real non-planar scenes such image registrations result in high distortions, already discussed in Sect. 1. Furthermore, by re-using results from the meta-data and feature based mosaicking such as the initial transformation and the matching keypoints of all available images we are able to estimate a rough 3D structure for further analysis. The computed 3D structure is the base of finding a common projection plane to further improve the image transformation. If sufficient image data is available we end up in the estimation of improved image transformations by limiting the considerable feature keypoints from previous results.

Algorithm 1 Incremental mosaicking code

```
while images $\mathbf{I}_i$ arrive do
    if $q_{image}(\mathbf{I}_i) \geq \gamma$ then
        extract features of $\mathbf{I}_i$
        determine neighbors by the meta-data $M$
        match features among neighbors
        if operator requests mosaic then
            find similarity and apply $\mathbf{T}_{\text{image}_i}$
        end if
        estimate or update structure
        find or update the common plane
        compute image transformation on plane $\mathbf{T}_{\text{image}_i}$
        if operator requests mosaic then
            update mosaic with $\mathbf{T}_{\text{image}_i}$ for all images
        end if
    else
        place image $\mathbf{I}_i$ by $\mathbf{T}_{\text{cam}_i}$
    end if
end while
```

We evaluate the projective quality q_{proj} that is representing the spatial accuracy of the image transformation and the image quality q_{res} for each image.

$$q_{\text{image}}(\mathbf{I}_i) = f(q_{proj}(\mathbf{I}_i), q_{res}(\mathbf{I}_i)) \tag{5}$$

The emphasis of each component in this evaluation depends on the application constraints and defines where individual images are used. Bad images are neglected later since they will not improve the output mosaic due to high distortions. On the other hand, images that are only available in a low resolution but with good projection quality can easily and quickly be included into the output mosaic. In our settings we have been using a distribution of $\frac{2}{3}norm(q_{proj}) + \frac{1}{3}norm(q_{res})$.

The quality q_{proj} is computed from the UAV's meta-data and camera parameters over the photo release interval and elaborated in Eq. 15. The other quality component q_{res} represents the image quality in terms of spatial resolution was earlier elaborated in Eq. 4.

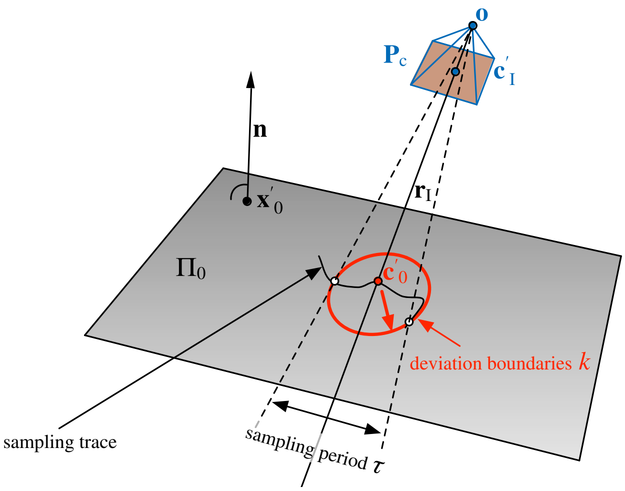

Fig. 7 Projection of the camera extrinsics to an arbitrary ground plane

4.2.1 Meta-data based mosaic

The first transformation estimation is executed on the UAV platform utilizing all available sensor data from the UAV's meta-data as the extrinsic parameters of the cameras. The image pixel coordinates are mapped into the world coordinate system by the intrinsic camera matrix. Since we know an estimate position and orientation of the camera in the world coordinates we are able to project the image based on the intrinsic camera model to the mosaic. In ideal systems, these transformations are directly computed from the camera calibration and UAV's sensor data, Eq. 6. Our camera extrinsic parameters are afflicted by sensing errors on the UAV and the intrinsic parameters from the camera calibration are also inaccurate because of different conditions between indoor calibration and outdoor use (such as focal distance and aperture). We employ simple statistical filters to achieve a first approximation and have noticed that the mean of the position and view angles are sufficient as initialization, elaborated in Eq. 13. The most severe inaccuracies arises due to the poor synchronization between the flight controller (GPS and IMU sensors) and the camera. The point of time when taking an image cannot be determined exactly because of the design of the camera and unknown internal delays. During image capturing the UAV encounters environmental influences that disturb the position and attitude angles heavily. Roll and pitch angles are typically more afflicted by the system dynamics and cause arbitrary perspective views. The horizontal yaw angle is exposed to a steady drift and offset.

Figure 7 depicts the estimation of the extrinsic parameters on the vehicles attitude data over a certain sampling period τ. Within this period τ the projection of each image center $\mathbf{c}'_I$ on an arbitrary ground plane Π_0 is estimated from theintrinsic and extrinsic camera parameters $\mathbf{P}_c$. This allows a rough estimation of the projection quality q_{proj} for each captured image defined in Eq. 15.

$$\mathbf{P}_c = \left[\mathbf{I}\ \mathbf{t}_c\right]\ \mathbf{R}_c\ \mathbf{K}_c \tag{6}$$

$$\mathbf{t}_c = (x_c, y_c, z_c)^T = f(O_{GPS}, X_{GPS}) \tag{7}$$

In Eq. 6 the extrinsic camera parameters are derived from the 3D coordinate translation $\mathbf{t}_c$ related to the GPS position in Eq. 7, the 3D rotation matrix $\mathbf{R}_c = R(\Psi)\ R(\Phi)\ R(\Theta)$ determined from the IMU angles for yaw, pitch and roll $\{\Psi, \Phi, \Theta\}$ and the camera intrinsic parameters $\mathbf{K}_c$.

$$\Pi_0 := \{\mathbf{n}, \mathbf{x}'_0\} \tag{8}$$

The horizontal ground plane Π_0 (Eq. 8) is defined by the normal vector $\mathbf{n} = (0, 0, 1)^T$ through the origin $\mathbf{x}'_0 = (0, 0, 0)^T$. If we transform images to this ground plane Π_0 by the approximated view from $\mathbf{P}_c$ we still explore perspective distortions in the transformed images. These distortions cannot be compensated by the meta-data only, instead the image-data itself has to be considered to estimate more accurate camera projections.

These parameters are converted from the UAV specific format to a unified format and coordinate system. The preferred orientation of the camera is the nadir view (vertically downwards) to a horizontal scene. The camera translation is a function of the scenario origin O and the current position within the sampling period τ which is the time from the trigger signal t_{0_i} of the camera to the first data received from the camera at the imaging unit. The scenario origin O is equal to the origin $\mathbf{x}'_0$ mapped to GPS coordinates O_{GPS}. The current position of the UAV is determined from the GPS data X_{GPS} in relation to O_{GPS}. The projected principle point $\mathbf{c}'_0$ in Eq. 11 on the plane Π_0 is defined as intersection of the principle axis $\mathbf{r}_I$ through the camera origin $\mathbf{o}$ and the transformed image center $\mathbf{c}'_I$ in Eq. 10. During the sampling period τ the camera origin $\mathbf{o} = f(\mathbf{t}_c)$ and orientation $\mathbf{R}_c$ varies. This variation causes projections to the ground plane different from the true projection.

$$\mathbf{c}'_I = \mathbf{P}_c\,\mathbf{c}_I \tag{9}$$

$$\mathbf{r}_I = \frac{\mathbf{o} - \mathbf{c}'_I}{norm(\mathbf{o} - \mathbf{c}'_I)} \tag{10}$$

$$\mathbf{c}'_0 = \mathbf{o} - \frac{\mathbf{n}\cdot(\mathbf{o} - \mathbf{x}'_0)}{\mathbf{n}\cdot\mathbf{r}_I}\mathbf{r}_I\ \Big|\ \mathbf{n}\cdot\mathbf{r}_I \neq 0 \tag{11}$$

The individual projected image centers during the period τ are denoted as $\mathbf{c}_0(t)$ for each individual sample. The deviation from the true projection is presented in Fig. 7 as circle k around all points $\mathbf{c}_0(t)$. Given a set of all points $\mathbf{c}_0(t)$ we find the smallest bounding circle by the smallest enclosing circle method [26], defined by the center $\mathbf{m}'$ and the radius r. The radius r of the smallest circle k around all points during τ_i is an estimate of the projective quality q_{proj} for this image in Eq. 12.

$$k = \{\mathbf{c_0}(t), t \in \{t_0, t_0 + \tau\}\ \big|\ norm(\mathbf{m}' - \mathbf{c}_0(t))) \leq r\}$$

$$\mathbf{m}' = (x'_m, y'_m, z'_m)^T \tag{12}$$

The number of discrete samples w_i captured from the UAVs sensors during τ depend on the update capabilities of UAV flight controller, the GPS update rate and IMU sampling rate.

$$\mathbf{c}'_0 = \frac{1}{w}\sum_{t=t_0}^{\tau}\mathbf{c}_0(t) \tag{13}$$

Hence, the rough image transformation $\mathbf{T}_{\text{cam}}$ is estimated from the average camera projection for $\mathbf{c}'_0$. The quality function in Eq. 15 for each initial image projection is defined by the average angle deviation α_p from the nadir view $\mathbf{n}$ during the period τ. It is represented by the deviation boundaries around the projected centers. The radius is normalized to the maximum allowed deviation from the physical properties of pitch and roll angles of up to 15° restricted by the vendor of the used UAVs for controlled waypoint flights.

$$\cos(\alpha_p) = \frac{(\mathbf{o} - \mathbf{c}'_0)\cdot\mathbf{n}}{norm(\mathbf{o} - \mathbf{c}'_0)\,norm(\mathbf{n})} \tag{14}$$

$$q_{proj} = 1 - \cos(\alpha_p)\,norm(r, r_{15^\circ}) \tag{15}$$

Finally, the transformation is simply computed between the projection on the ground plane (Eq. 11) and the native nadir projection of the image. We compute the resulting homography transformation $\mathbf{T}_{\text{cam}}$ in Eq. 16 from the camera projection $\mathbf{P}_c$ and projected points $\mathbf{x}'$ as proposed in [18].

Algorithm 2 Meta-data based transformation

```
Input:   intrinsic camera matrix $\mathbf{K}_c$
         camera position $\mathbf{t}_c(t)$
         camera orientation $\mathbf{R}_c(t)$
Output:  transformation $\mathbf{T}_{\text{cam}}$
         quality estimation $q_{proj}$
for $t = t_0$ to $t_0 + \tau$ do
    compute camera matrix $\mathbf{P}_c(t)$
    compute ground projection $\mathbf{c}_0(t)$
end for
estimate mean projection $\mathbf{c}'_0$
estimate projection quality $q_{proj}$
compute $\mathbf{T}_{\text{cam}}$ directly from projection $\mathbf{c}'_0$
return $\mathbf{T}_{\text{cam}}$, $q_{proj}$
```

$$\tilde{\mathbf{x}} = \mathbf{T}_{\text{cam}}\cdot\mathbf{x} \Rightarrow \tilde{\mathbf{x}} \times \mathbf{T}_{\text{cam}}\cdot\mathbf{x} = 0 \tag{16}$$

This major pre-estimation step is executed on the limited resources on the UAV according to Algorithm 2. The resulting homography transformation is included in the meta-data to the image and transmitted to the ground station.

4.2.2 Feature based mosaic

Panorama mosaics in typical image registration applications are generated from a single view point only by rotating a camera around its axis. In Eq. 17 this homography transformation is only related to the rotation $\mathbf{R}$ and the intrinsic camera parameters $\mathbf{K}$.

$$\mathbf{H} = \mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1} \tag{17}$$

An image created by this method is an extension of the FOV but in our case images are taken from multiple UAVs from different view points. To improve image transformations when the extrinsics are insufficient the exploration of the image data in Eq. 1 is required. This process is computationally expensive because keypoint extraction and matching has to be executed on the image data. After evaluating speeded up robust features (SURF) and scale invariant feature transform (SIFT) we decided to employ SIFT for feature extraction teamed with k-d-tree keypoint matching [16]. The outputs of the feature extraction and matching are immediately used for the estimation of image-data based transformation. According to our studies [30] we apply a pair-wise matching function for finding the similarity transformation among selected images to minimize the distortion error propagation during the mosaicking. Initially, an image with a minimum perspective distortion from the available images at the ground station is selected, i.e, $\min(q_{proj})$. For the pair-wise estimation we build and maintain an ordered set P of image pairs $p(a, b)$ from the total number of N images (Eq. 18).

$$P := \{p(a, b) \mid a \leq N, b \leq N\} \tag{18}$$

$$\mathrm{order}(P)\mathrm{by}(|A_O(\mathbf{I}_a, \mathbf{I}_b)|, |K_m(a, b)|, q_{proj}(a)+q_{proj}(b)) \tag{19}$$

To avoid global optimizations at this point, the ordering within this set of image pairs is based on the parameters of the ordering function in Eq. 19. First, the size of the overlapping area $A_O(\mathbf{I}_a, \mathbf{I}_b)$ is investigated as the primary ordering criteria, followed by the number of matched features $|K_m|$ between two images a and b denoted with function $(\equiv)$ in Eq. 21. At this stage we assume that sufficient features are matched and equally distributed over the overlapping area A_O.

$$\mathbf{x}_i = (x, y) \in \mathbf{I}_a \quad \mathbf{x}_j = (x, y) \in \mathbf{I}_b \tag{20}$$

$$K_m = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \equiv \mathbf{x}_j, \mathbf{x}_i \in A_O(\mathbf{I}_i), \mathbf{x}_j \in A_O(\mathbf{I}_j)\} \tag{21}$$

Distortions in the mosaic are reduced if the quality estimation q_{proj} in Eq. 15 indicates a minimum deviation from the nadir view for both images $q_{proj}(a)$ and $q_{proj}(b)$. Even when employing the similarity transformation the quality of the initial image is important. The smaller the approximated angle deviation α_p, the less distortion errors are propagated to the following images. A good estimation is found by random sample consensus [14] from the highest ranked pair in the set P where both images $\mathbf{I}_a$ and $\mathbf{I}_b$. In Eq. 22 this transformation is defined by a scaled 2D rotation and translation for all two-dimensional pixels of both images where $\mathbf{x}$ denotes the original pixel and $\tilde{\mathbf{x}}$ the transformed one.

$$\tilde{\mathbf{x}} = \mathbf{T}_{\mathrm{image}_{a,b}}\mathbf{x} = \begin{bmatrix} s\,\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \tag{22}$$

Images in the set P with a very low rank are kept back. Reasons for poor ranks are small overlapping areas $A_O(\mathbf{I}_a, \mathbf{I}_b) \leq \delta$ or large projection angles α_p. The overlapping constraint δ is defined by the application requirements and later defined in Eq. 23. These images keep their meta-data based placement unless additional images arrive that could improve their rank in the set, e.g, by increased overlap.

Algorithm 3 Image-data based transformations

```
Input:    transformation $\mathbf{T}_{\mathrm{cam}}$ and quality $q_{proj}$
          image data $\mathbf{I}$
Output:   transformation $\mathbf{T}_{\mathrm{image}}$
extract features from $\mathbf{I}$
compute overlapping area to all other received images
match features with overlapping images
rank image $\mathbf{I}$ into previous set $(P)$
compute similarity transformation $\mathbf{T}_{\mathrm{image}}$
return image transformation $\mathbf{T}_{\mathrm{image}}$
```

The method elaborated in Algorithm 3 is executed on the ground station. Intermediate steps of the processing are presented in Fig. 8a, b where individual images are stitched based on image data. In contrast to traditional image registration we can significantly speed up the feature matching process by pre-selection of possible image pairs and reducing the number of features to match. This optimization utilizes the meta-data and is presented in the next section.

4.2.3 Efficient keypoint extraction and matching

Adding new images to large image sets results in a high computational effort when employing standard methods (global optimizations) for image registration. Without knowledge of possible overlapping images every new image forms $N - 1$ possible pairs where N is the total number of images. In similar images, e.g, grassland, water surfaces, forests, among others, standard methods will result in various false positive keypoint matches. To find relations between images prior

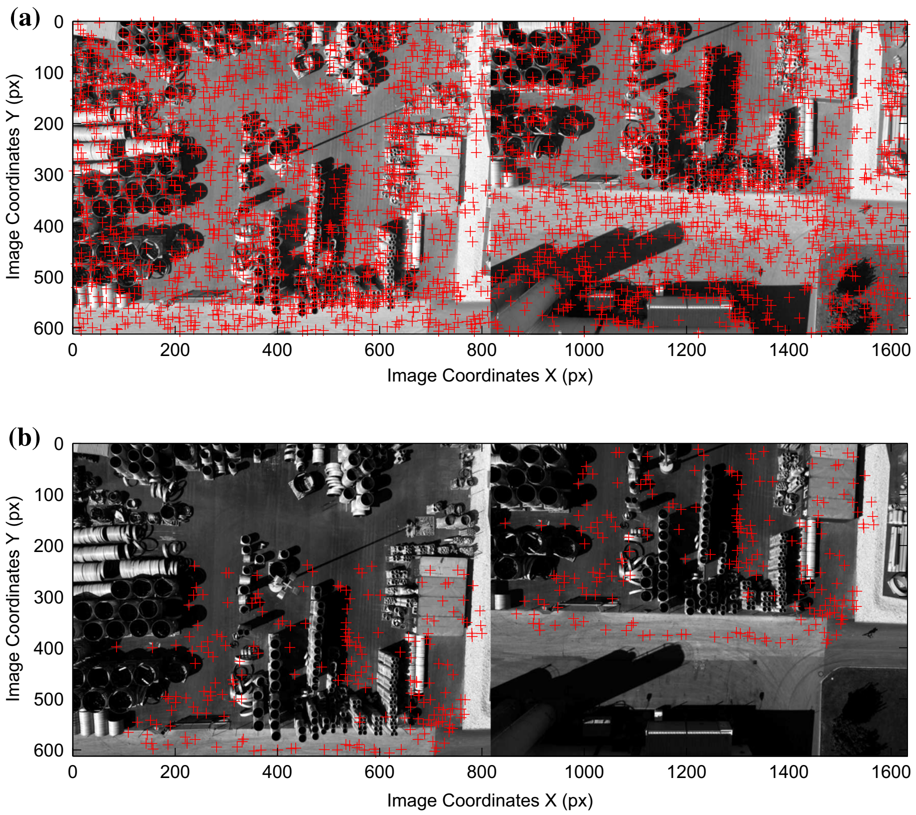

Fig. 8 Image registration steps presenting extracted and matched keypoints

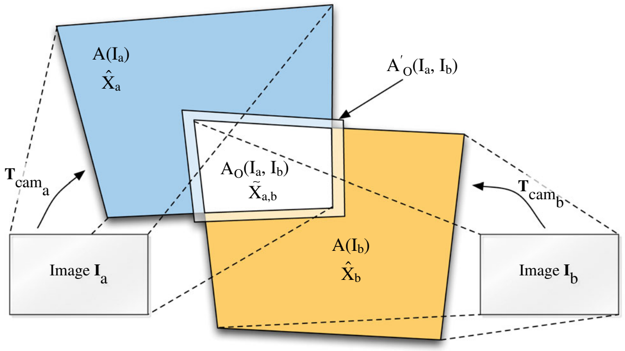

Fig. 9 The intersection of two images is defined as overlapping area

we consider the local neighborhood property, if two images share an area overlap of a certain amount.

To reduce the number of possible matches within the total set P of all possible image pairs we discard images with no or just little overlap in border regions. By reducing outliers and false positives we gain more robust spatial image transformations. The set of pairs $\tilde{P}$ is reduced only to pairs of images $p(a, b)$ with significant overlap $A_O(\mathbf{I}_a, \mathbf{I}_b)$ in the overlapping region between image $\mathbf{I}_a$ and image $\mathbf{I}_b$ in Fig. 9.

$$\begin{aligned}\tilde{P} &:= \{p(a,b) \in P \,\big|\, |A_O(\mathbf{I}_a, \mathbf{I}_b)| \\ &\geq \delta \cdot \max\left(|A(\mathbf{I}_a)|, |A(\mathbf{I}_b)|\right), a < b\}\end{aligned} \tag{23}$$

In Eq. 23 the overlapping area is defined to be larger than a fraction δ of the area of the larger image. This is computed efficiently by intersecting convex four-point-polygons of image areas. An image area $A(\mathbf{I}_i)$ on the projection plane is defined in Eq. 24 after applying the initial transformation $\mathbf{T}_{\mathrm{cam}_i}$ to each pixel. Each pixel $\mathbf{x} \in \mathbf{I}_i$ is transformed into the image polygon to a pixel $\tilde{\mathbf{x}}$. This transformed pixel is specified by the two-dimensional coordinates $(\tilde{x}, \tilde{y})^T$. The overlapping area of a pair of images is defined by their area set intersection (Eq. 25).

$$A(\mathbf{I}_i) = \{\tilde{\mathbf{x}} | \tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^T, \mathbf{x} \in \mathbf{I}_i, \tilde{\mathbf{x}} = \mathbf{T}_{\mathrm{cam}_i}\mathbf{x}\} \tag{24}$$

$$\begin{aligned}A_O(\mathbf{I}_a, \mathbf{I}_b) &= A(\mathbf{I}_a) \cap A(\mathbf{I}_b) \\ &= \{\mathbf{x} = (x, y)^T \mid (x, y) \in \mathbf{I}_a, \mathbf{I}_b\}\end{aligned} \tag{25}$$

The keypoint matching is now efficiently executed on selected pairs of images $p(a, b) \in \tilde{P}$ where the overlapping area is sufficient. This number of selected pairs $|\tilde{P}|$ is significantly reduced in comparison with considering all keypoints in image pairs ($k = 2$) in the set of all possible matches P for an incrementally growing set of N images in Eq. 26.

$$|P| = \frac{1}{2} \cdot (N - 1) \cdot N = \left.\frac{N!}{k! \cdot (N - k)!}\right|_{k=2} \tag{26}$$

Experiments have shown that the generic overlapping constraint δ is not sufficient in dynamic situations. To achieve mosaics of a comparable quality to full 3D reconstructions, the influences of harsh environments and different scene structures have to be considered into the overlapping constraint. State-of-the-art mosaicking methods, even of planar scenes, require an overlap of $\delta > 0.9$ as presented in Caballero et al. [8]. In dense structured scenarios a higher overlap is often required, to compensate environmental influences such as wind gusts and irregular UAV movement we introduce an additional overlapping margin η. It expands the overlapping area to A'_O (Eq. 29) to search for correspondences. While δ is constrained by the application requirements and considered during the planning, η is adapted for each vehicle and during flight separately as function of q_{proj} of the last five images (Eq. 27).

$$\eta = \frac{1}{N-m} \sum_{i=m}^{N} 1 - q_{proj}(i) \quad | \, m = \max(0, N-5) \qquad (27)$$

In our approach we extract SIFT keypoint features in one image $\mathbf{I}_i$. A subset $\tilde{X}_i$ of all available keypoints $\hat{X}_i$ for this image is exploit for matching in the overlapping area. The subset for two overlapping images $\{\tilde{X}_a, \tilde{X}_b\}$ is defined in Eq. 28 for those points $\mathbf{x}$ that reside inside this expanded overlapping area polygon are considered for matching.

$$\begin{aligned} \tilde{X}_{a,b} &:= \{\tilde{X}_a, \tilde{X}_b\} \\ &\subset \{\mathbf{x}_a, \mathbf{x}_b \,|\, \mathbf{x}_a \in \hat{X}_a, \mathbf{x}_b \in \hat{X}_b : \mathbf{x}_i \in A'_O(\mathbf{I}_a, \mathbf{I}_b)\} \qquad (28) \\ A'_O(\mathbf{I}_a, \mathbf{I}_b) &= \text{expand}\,(A_O(\mathbf{I}_a, \mathbf{I}_b), \eta) \qquad (29) \end{aligned}$$

In Fig. 10 the left image $\mathbf{I}_a$ is marked with a blue border and has an overlapping polygon A'_O (cyan) with the right image $\mathbf{I}_b$ surrounded by a red border. All keypoints $\hat{X}_a$ and $\hat{X}_b$ are shown in gray in the background while the selected keypoints for matching are colored. The sets X_a and X_b are the extended sets of the overlapping area $\tilde{X}_a$ and $\tilde{X}_b$ including the margin η and considered for the feature matching between image $\mathbf{I}_a$ and image $\mathbf{I}_b$. Thereafter, RANSAC is applied on this set to determine the similarity image transformation $\mathbf{T}_{\text{image}_i}$. All points from $X_{a,b}$ that are located with in the approximation threshold ϵ are called inliers to the estimated similarity transformation target function. The estimated transformation $\mathbf{T}_{\text{image}_i}$ replaces the initial transformation $\mathbf{T}_{\text{cam}_i}$ of each image.

$$\begin{aligned} X_{a,b} = &\{(\mathbf{x}_a, \mathbf{x}_b) \big| \mathbf{x}_a \in X_a, \mathbf{x}_b \in X_b, \\ &|desc(\mathbf{x}_a) - desc(\mathbf{x}_b)| < \epsilon\} \qquad (30) \end{aligned}$$

4.2.4 Structure based keypoint selection

The next step in the incremental approach is the reduction of inaccuracies and distortions caused by the scene structure. We profit from different view points of our UAVs flying over the observation area to gain structure information of the monitored scene. Hence, for the reconstruction of a scene we require at least two different views to compute depth information—which we gain due to our multi-view setup where every image taken at a different pre-planned location. For full feature-based 3D reconstructions our resources are insufficient [17]. However, we are able to exploit sufficient data for a rough 3D reconstruction. Results from previous feature extraction and feature matching are reused in the computation of a sparse 3D point cloud.

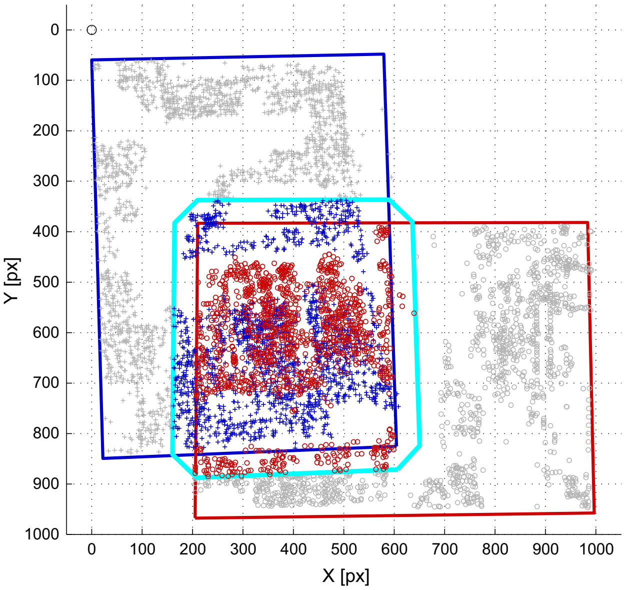

Fig. 10 An image pair projected by the camera extrinsics with relevant keypoints in the overlapping area of at least $\delta = 0.3$ of each image size expanded by $\eta = 0.1$

Since we are not interested in a nice 3D model, as presented in Fig. 11, rather than a quick planar overview mosaic. We introduce an initial projection plane Π_0 parallel to the camera plane supported by our knowledge about the planned routes. We compute the epipolar geometry relation of matching keypoint pairs by initially employing the structure-from-motion approach [24,25] that is defined by the fundamental matrix $\mathbf{F}$. One initial image pair $p(a, b)$ is selected from the pairs of images $\tilde{P}$ by its lowest perspective distortion error q_{proj} and the smallest re-projection error in the fundamental matrix estimation from the keypoint matches. Therefore, the first step is the estimation of the epipolar geometry for each pair that results in the fundamental matrix [18] in Eq. 31.

$$\mathbf{F} = \mathbf{K}^{-T}\,\mathbf{T}\,\mathbf{R}\,\mathbf{K}^{-1} \qquad (31)$$

Thereafter, the bundle adjustment [27] is executed incrementally by adding additional images. The 3D structure is composed from the set of points X' and cameras C' that are

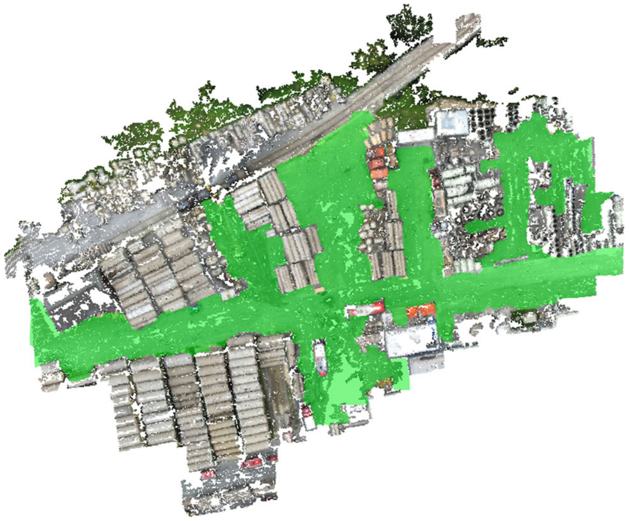

Fig. 11 In this dense point cloud (enhanced by patch-based multi-view stereo method (PMVS) [17]) the observed area is presented from nadir view and the ground plane is manually annotated in *green*. This plane is our request projection plane (color figure online)

also adjusted to minimize the re-projection error. In each iteration a sparse point cloud is enlarged and increased in accuracy by additionally matching keypoints. The majority of matched keypoints remain on surfaces seen from above because those are visible in multiple images due to the nadir perspective. On homogeneous surfaces, we explored at least feature keypoints along edges between objects and the ground plane.

4.2.5 Common projection plane

Finding the optimum projection plane that keeps geospatial relations [30] is now an incremental process, explained in Algorithm 4, since the 3D structure model also incrementally grows and improves. We define the optimum plane as the ground plane that is most perpendicular to the principal axis of the cameras admitting with the assumption that all camera orientations are nadir. In non-horizontal scenes this target function has to be reconfigured.

```
Algorithm 4 Plane fitting
  Input:   matched keypoints $X$ and pairs $\tilde{P}$
  Output:  common projection plane $\Pi_0$
           sparse 3D structure $X'$
           image transformations $\mathbf{T}_{\mathrm{image}_i}$
  find epipolar geometry for all pairs $\tilde{P}$ and keypoints $X$
  add matches to bundle adjustment
  (re)-adjust the bundle
  update the common ground plane $\Pi_0$
  select keypoints within $|d(\mathbf{x}', \Pi_0)| \leq \epsilon$
  update image transformations on plane $\Pi_0$
  return common projection plane $\Pi_0$ and structure $X'$
```

The estimation of the ground plane is executed on the Euclidean coordinate output of the incremental bundle adjustment and updated when incrementally adding more images considering previous results as initialization. The general plane function in Eq. 32 is used in its definition by the non-zero normal vector $\mathbf{n}$ through the point on the plane $\mathbf{x}'_0$ in Eq. 33 where a point is defined as $\mathbf{x}' = (x', y', z')^T$. Each keypoint $\mathbf{x}$ in the two-dimensional image space is related to its three-dimensional point $\mathbf{x}'$ by its projection.

$$d = -a\,x'_0 - b\,y'_0 - c\,z'_0 \tag{32}$$

$$\Pi := \mathbf{n}\,(\mathbf{x}' - \mathbf{x}'_0) = 0 \tag{33}$$

According the definition in Eq. 34 three points $(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3)$ are required to determine the normal vector of the plane. Any of the three points can be chosen as $\mathbf{x}'_0 = (x'_0, y'_0, z'_0)^T$ in the plane definition.

$$\mathbf{n} = (\mathbf{x}'_2 - \mathbf{x}'_1) \times (\mathbf{x}'_3 - \mathbf{x}'_1) \tag{34}$$

The approximation of a plane by RANSAC can end up with any arbitrary plane. However, when defining the plane constraints, i.e, the normal vector in vertical direction $\mathbf{n_z} = (0, 0, 1)^T$, the result is imprecise because our 3D model and the initial camera extrinsic parameters are afflicted by deviations. In Fig. 12 the incremental growing and improvement is demonstrated.

Due to the constraint flight height to the same altitude during the mission we achieve a well defined initialization for the horizontal plane as the plane through the camera centers in Eq. 35.

$$\Pi_c := \mathbf{n_c}(\mathbf{o}_i - \mathbf{o}) \quad \forall \mathbf{o}_i \in C' \tag{35}$$

If at least three camera positions $\mathbf{o}$ are available, a plane can be fit through these camera origins with orthogonal regression and RANSAC. The angle θ between the resulting camera plane normal vector $\mathbf{n_c}$ and the projection plane normal vector $\mathbf{n}$ in Eq. 36 has to be minimized.

$$\cos(\theta) = \mathbf{n_c} \cdot \mathbf{n} \quad \text{where} \quad |\mathbf{n_c}| = |\mathbf{n}| = 1 \tag{36}$$

With a small number of images $n < 10$ a very sparse point cloud results, hence the camera plane normal vector $\mathbf{n_c}$ is necessary to achieve a plausible plane fitting results. When adding more images, the bundle adjustment appends additional keypoints $\mathbf{x}'$ to the set of 3D points X' and also increases the accuracy of existing points. Within this updated point cloud the plane fitting is executed repeatedly with the previously estimated ground plane serving as initialization.

A plane is the ground plane of a scene if it is the lowest horizontal plane with the largest number of inliers defined

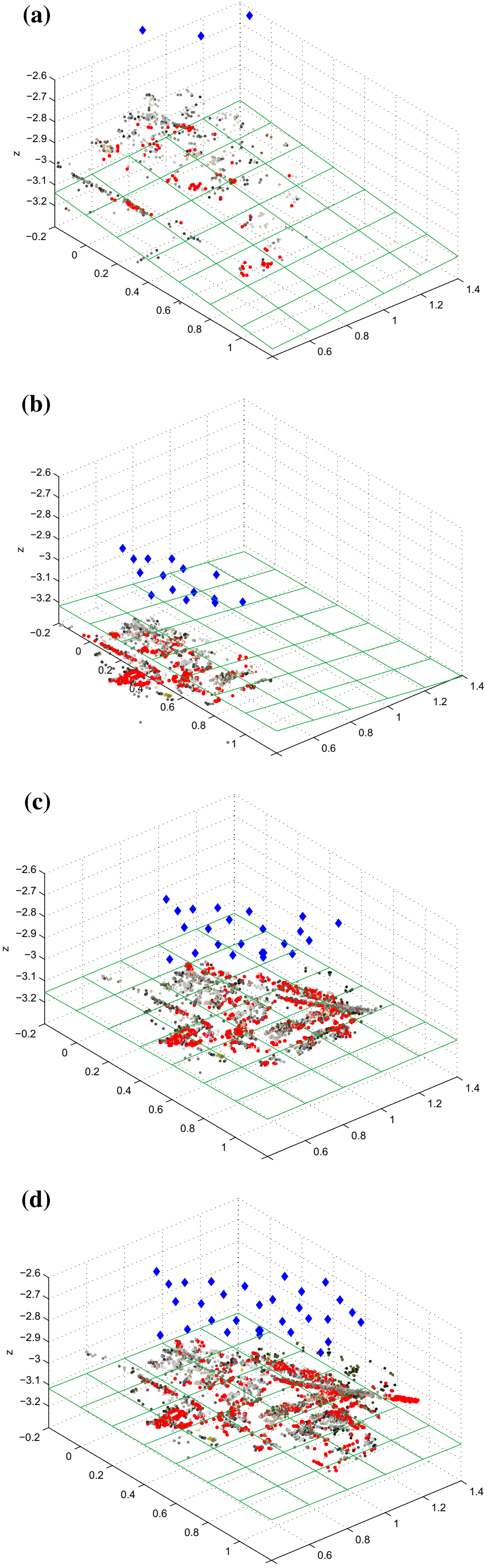

Fig. 12 Incremental refinement of the 3D structure and the ground plane. The *diamond* mark the approximated UAV positions. Points marked in *red* lay on the ground plane, sketched as *green grid* (color figure online)

in Eq. 37. During the RANSAC method all points are evaluated to be within or outside a certain distance $|d| \le \epsilon$ to the selected sample set. These inliers on the ground plane compose the image transformation later.

$$X'_{\Pi} := \{\mathbf{x}' \in X' \mid |d(\mathbf{x}', \Pi_0)| \le \epsilon\} \tag{37}$$

$$d(\mathbf{x}', \Pi_0) = \mathbf{n} \cdot \mathbf{x}' + \mathbf{x}'_0 \tag{38}$$

Points from the current result set X' with a positive point-to-plane distance $d(\mathbf{x}', \Pi_0) > 2\,\epsilon$ above the plane are marked as outliers and discarded. This reduces the search space in the set of points for further iterations.

Any update of the ground plane adds additional inliers within the distance of ϵ. Other points are removed when dropping out of the margin ϵ due to a slight tilt of the plane. The final ground plane Π_0 is defined in Eq. 39 and its inliers in Eq. 37 that are considered for computing the image transformation. After adding a number of images the mosaicking process according to Fig. 4 is executed again and the resulting plane Π_0 converges to a stable solution.

$$\Pi_0 := \mathbf{n}_0(\mathbf{x}' - \mathbf{x}'_0) \tag{39}$$

In the examples in Fig. 13 the inlying keypoints $\mathbf{x}' \in X'_{\Pi}$ on the resulting ground plane Π_0 are marked.

The image transformations computed from this ground plane inliers $\mathbf{x}'$ are denoted as the image transformation $\mathbf{T}_{\text{image}}$ and employed for the final transformation $\mathbf{T}$.

4.2.6 Mosaic generation: visualization

Figure 14a shows individual images $\mathbf{I}_i$ processed according to their assigned image transformation $\mathbf{T}_i$ independent of their currently received resolution. The same transformation is also employed to generate a mask image which is required for blending individual images to one mosaic, presented in Fig. 14b. Each color value of pixel $\mathbf{x} = (x, y)^T \in \mathbf{I}$ is copied to the destination image with the perspective transformation which is derived from the homography definition in Eq. 40 and pixel mapping in Eq. 41.

$$\mathbf{x}' = \mathbf{T}\,\mathbf{x} \quad \mathbf{T} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \tag{40}$$

$$x' = \frac{h_{00}\,x + h_{01}\,y + h_{02}}{h_{20}\,x + h_{21}\,y + h_{22}}, \quad y' = \frac{h_{10}\,x + h_{11}\,y + h_{22}}{h_{20}\,x + h_{21}\,y + h_{22}} \tag{41}$$

In Fig. 14c, d the displacement vectors in the meta-data based mosaic are sketched and Fig. 14e presents the images registered according to these displacement vectors in the image-data based approach. Depending on the available computing power the visual optimization by seamless image

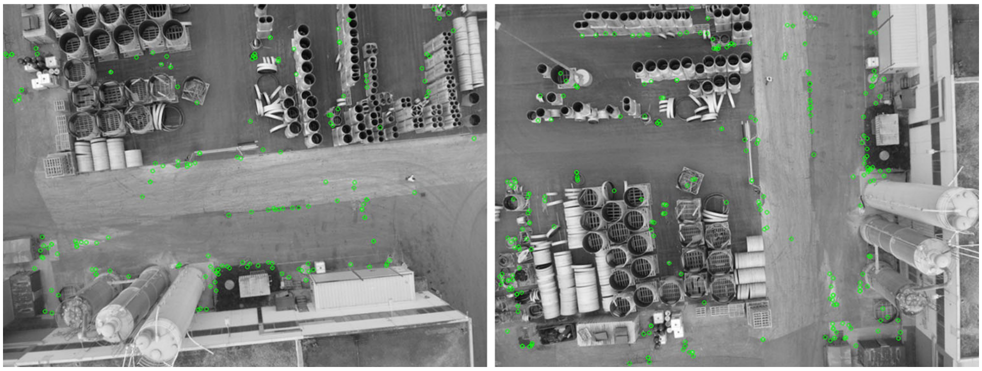

Fig. 13 *Green* markers show the selected keypoints on the ground plane in these example images (color figure online)

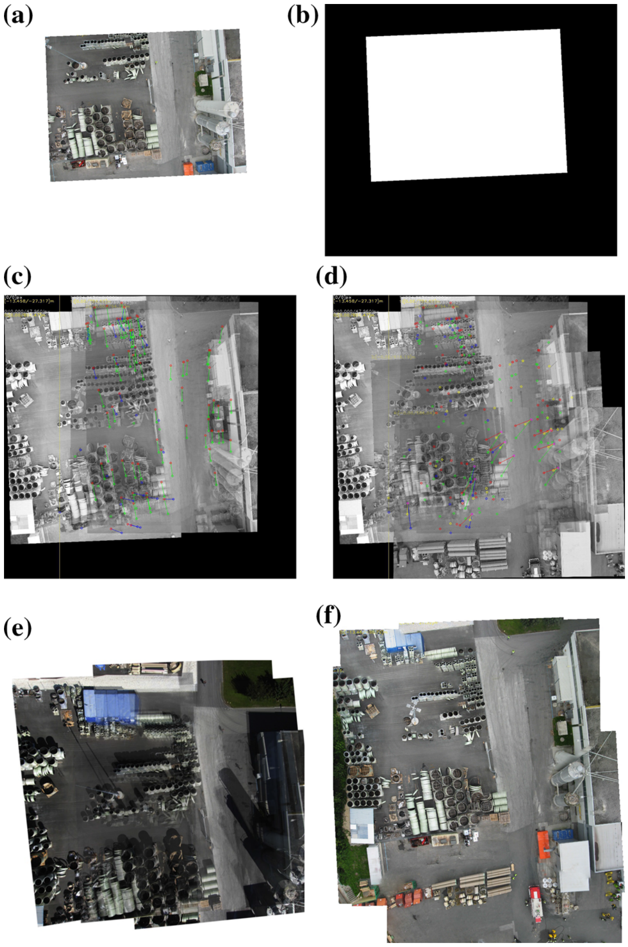

Fig. 14 These figures sketch the final mosaicking steps. In **a** one image is transformed and with its mask added to the previous mosaic in **c**. Finally single images are merged with blending in **f**

blending can be executed and blends the transformed images over each other. This step is important for adjusting exposure, saturation and de-ghosting objects. The open source software enblend[4] implements the state-of-the-art algorithm using multi-resolution splines [7] for combining images.

5 Case studies and evaluation

We successfully deployed our approach in two scenarios where we evaluated our approach against quality criteria defined in Eq. 42. This quality benchmark Ω is applied to the final mosaic as well as any intermediate stages.

$$\Omega(\mathbf{I}_O) = \nu\, \Omega_{corr}(S_I, \mathbf{I}_O, \Pi_0) + \lambda\, \Omega_{net}(S_I) + \rho\, \Omega_A(\mathbf{I}_O) \tag{42}$$

The individual quality components are evaluated throughout the whole process to provide feedback to the incremental processing. Depending on the mission preferences the components weights are adjusted. The overall sum of weights ν, λ, and ρ in Ω is 1.

The first quality component refers to the cross correlation $CC()$ Ω_{corr} of a set of currently mosaicked images $S_I = \{\mathbf{I}_i : 1 < n \leq N\}$ only within the ground plane area. The area on the ground plane is defined as intersection between the image pixels and the plane Π_0 including the variance ϵ in Eq. 43.

$$A(\mathbf{I}_i \cap \Pi_0) := \{\mathbf{x}'_i \in \mathbf{I}_i \mid |d(\mathbf{x}'_i, \Pi_0)| \leq \epsilon\} \tag{43}$$

$$\Omega_{corr}(S_I, \mathbf{I}_O, \Pi_0) = \frac{1}{n} \sum_{i=1}^{n} \frac{1 + CC(A(\mathbf{I}_O \cap \Pi_0), A(\mathbf{I}_i \cap \Pi_0))}{2} \tag{44}$$

The second component presents the transmitted data in Eq. 45. With the weight $\varphi = 0.5$ the contribution to the final mosaic is balanced between the number n of images and resolution layers $res(\mathbf{I}_i)$.

$$\Omega_{net}(S_I) = \varphi \frac{n}{N} + (1 - \varphi) \frac{1}{n} \sum_{i=0}^{n} res(\mathbf{I}_i) \quad | \mathbf{I}_i \in S_I \tag{45}$$

The third component $\Omega_A(\mathbf{I}_O)$ of Eq. 42 evaluates the covered area in the scene according to the requirements. In some sce-

[4] http://enblend.sourceforge.net/ last visited on November 11th 2013.

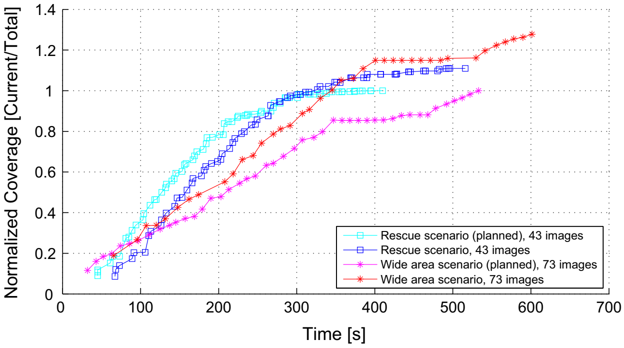

Fig. 15 This graph shows the covered area over time normalized to the requested area from the planning

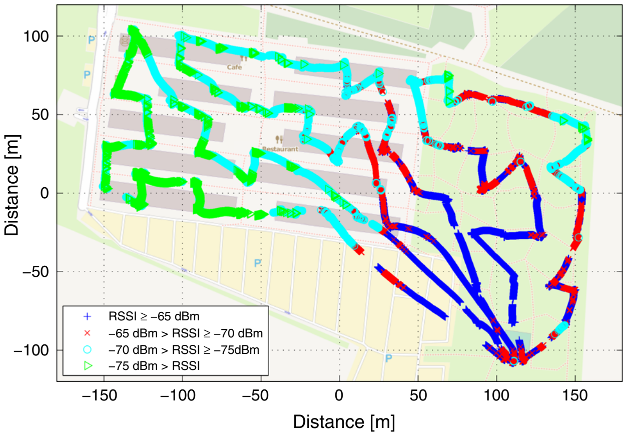

Fig. 16 Recorded received signal strength (RSSI) during the mission

narios, such as rescue scenarios, the covered area over time is of outermost importance. On the other hand for other applications, the same amount of data can be utilized with less but higher resolution images. During planning the whole scenario area is divided into a number $u = |G|$ of sub areas $G = g^2$ of different required resolutions κ_G Each subset $G_i \subseteq A(\mathbf{I}_O)$ has an individual requested resolution and the function $A()$ defines the area in world coordinates and $|A()|$ is the size of this covered area. The evaluation of the overview mosaic $\mathbf{I}_O$ is examined with these requirements against the planned coverage area. The size of the covered area $|A(G_i)|$ and the resolution $res(G_i)$ are evaluated in Eq. 46 for each subset.

$$\Omega_A(\mathbf{I}_O) = \frac{1}{u}\sum_{i=0}^{u} |A(G_i)| \cdot res(G_i) \tag{46}$$

In both scenarios we evaluate the contribution of individual images to the whole mission. In the fire fighter practice scenario the UAV routes concentrate around the center of the observation area which is the center of action and we achieve more overlap. Here the growth of the newly covered area is smaller than in the wide area monitoring scenario where each individual image covered a large new area. In Fig. 15 these evaluations are presented over time. The graphs are normalized to the planned area to cover. The finally covered area was larger than the planned area in both scenarios because of position inaccuracies.

Furthermore, we evaluated the duration for transmitting individual resolutions from all UAVs in the wide area scenario. For detailed analysis on the network we have been recording the data from network and link layers as well as from the application layer scheduling. The results are presented in Fig. 16 for the wide area scenario along the georeferenced routes. Each data point in the graph represents one measurement sample synchronized to the UAV position.

The received signal strength measurements are discussed in more detail in the work of Yanmaz et al. [31]. They exe-

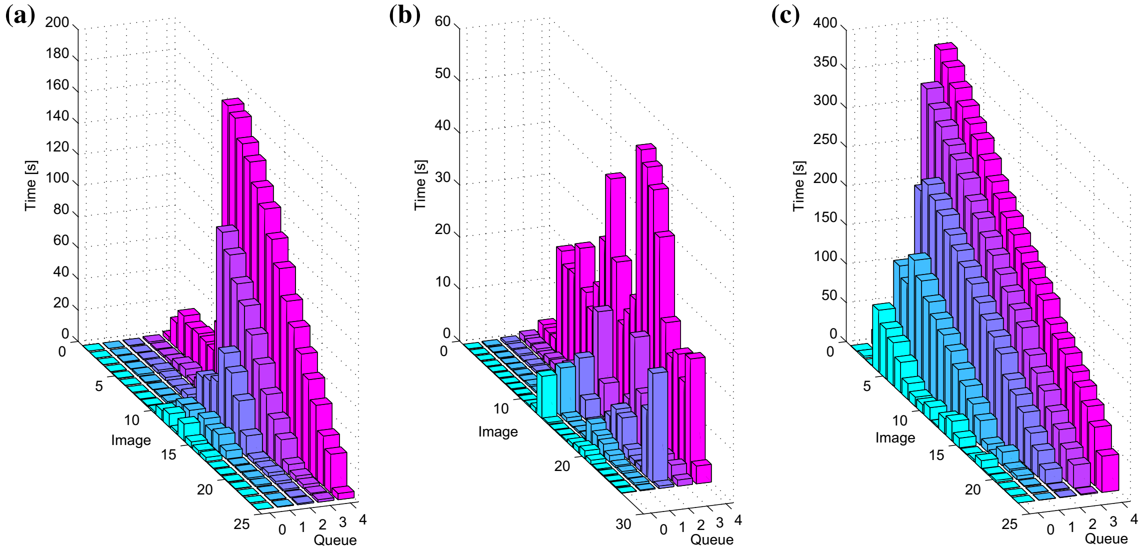

Fig. 17 Delay times between capturing images on the UAV and the reception of each specific image layer at the ground station

Table 1 Delay times of individual image layers from capturing to the ground station

Queue	Maximum delay [s]			
	UAV$_1$	UAV$_2$	UAV$_3$	
Q_0	10.7974	8.0504	79.2130	
Q_1	12.5058	9.1944	157.7492	
Q_2	52.8634	20.9972	242.4114	
Q_3	115.6260	20.8368	335.7916	
Q_4	186.5008	54.2094	373.4210	
Queue	**Mean delay [s]**			
	UAV$_1$	UAV$_2$	UAV$_3$	Total
Q_0	1.6827	0.5942	16.3619	5.9200
Q_1	2.9642	1.0695	53.5570	18.2296
Q_2	10.4332	3.2804	111.1155	39.6074
Q_3	27.2692	4.5191	167.8719	63.4659
Q_4	76.0867	23.4496	214.1930	100.8522

cuted and verified different tests on orientation and power levels by employing the free-space path loss model. We have been evaluating the transmission of single image layers and the prioritization of the data among all three UAVs. In Fig. 17 the individual delays of each resolution layer from each UAVs is measured, from the transmission start to the complete transfer of the respective layer. This evaluation shows that one UAV went into an area of reduced signal strength and was not able to transmit higher resolution data. Thereafter, only the lowest resolution representations of succeeding images could have been transmitted. Immediately when an UAVs went back into an area of better signal quality the remaining and stalled data could have been transmitted. More detailed, Table 1 shows the maximum and average delays from capturing one image until its layers are received at the ground station.

The image quality analysis in our selected scenarios delivers promising results, although these images contain highly structured scenes with an obvious but fractional ground plane. In Fig. 11 this structure of the fire fighter practice scenario is demonstrated where the ground plane is annotated manually for evaluations. The incrementally generated overview mosaic is analyzed according to the quality definition $\Omega(\mathbf{I}_O)$ defined in Eq. 42. The structure based mosaicking covers the 3D structure estimation, plane fitting and estimation of the image transformation of the ground plane that is executed on incoming image data. We analyze re-projection errors of the estimated 3D structure and the evolution of the ground plane. Example results are already presented in Figs. 12 and 13.

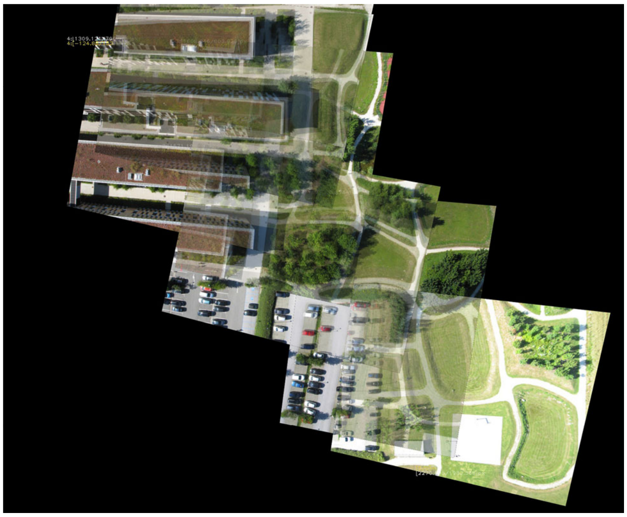

Fig. 18 Meta-data based intermediate mosaic including seven received images

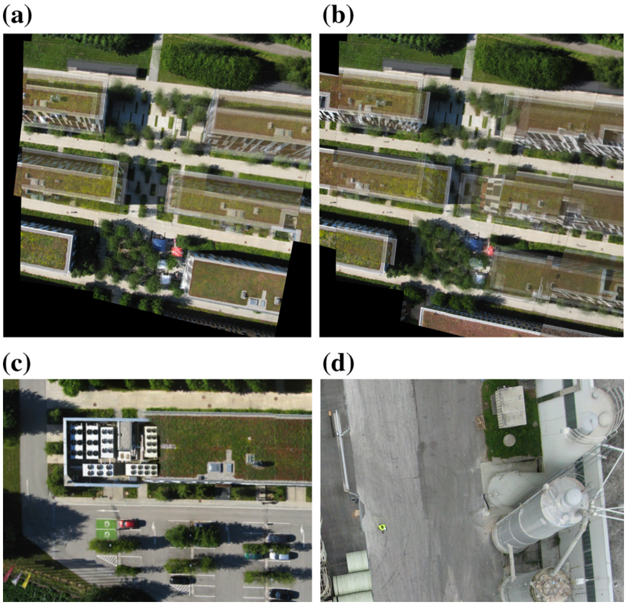

Fig. 19 Incremental mosaicking in detail. The ground plane is accurately mosaicked, while objects, i.e, buildings, occlude regions of the ground plane area

The first stage of the incremental mosaic is the meta-data based mosaic. This obviously imposes high deviations from the true scene. Environmental conditions and sensor uncertainties directly influence the image transformation quality. In Fig. 18 one intermediate result is presented of the meta-data based mosaic where all images are directly placed according to their annotated meta-data.

Any correlation evaluation executed on this meta-data based mosaic would present very bad results. Nevertheless, the immediate presentation of this mosaic to operators is important and valuable for a quick manual inspection. In the overall quality evaluation this poor visual quality is compensated by a very quick response time.

After the structure analysis and plane fitting, the common plane regions are successfully determined and the

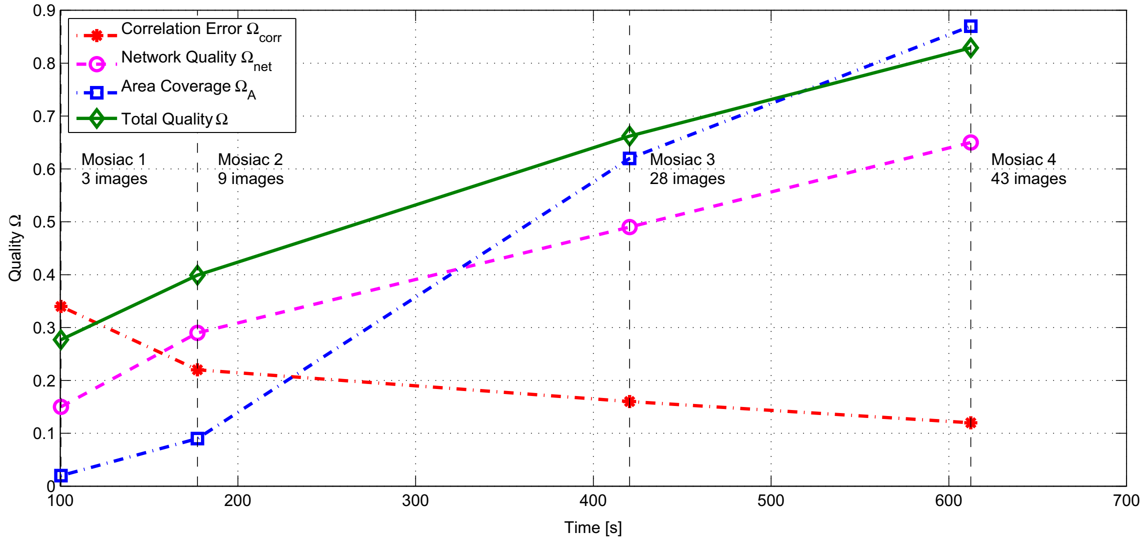

Fig. 20 Quality improvement over mission time by our incremental approach

mosaic is build on these regions. In Fig. 19a a few images are mosaicked by the image-data based approach. And in Fig. 19b more regions are occluded by buildings around the ground region because additional images from different views are incorporated to the mosaic. Occlusions of the ground plane and resulting multiple ghost objects are a challenge for the blending algorithm. The remaining images in Fig. 19 demonstrate the magnified blending outputs from both scenarios. In Fig. 19d we notice this effect at the silos where the ground plane area is emphasized because it is seen in more images from the same perspective.

To determine the overall quality throughout the mosaicking process, the correlation of single images stitched to the overview mosaic and the covered area are computed according to Eq. 42 and presented in Fig. 20 at different stages of the mission when the evaluation measurement was executed.

Furthermore, we compare our incremental approach to state-of-the-art mosaicking methods in terms of computational effort, presented in Table 2. The reference methods are AutoPano[5] which is the most prominent commercial software for mosaicking images and Pix4D[6] which does excellent 3D reconstructions, where AutoPano fails in dense structured scenes.

The most important evaluation is the measurement of dedicated landmarks for determining the spatial relative accuracy. The ratio of distances on the ground plane represents a measurement for the overall distortion of the image. We determined some landmarks and measured true distances and compared them to our final overview mosaic and the mosaic output of the reference methods.

Table 2 The final offline computation results compared to state-of-the-art mosaicking by time when all images are available

Test set	Method	Processing time (s)
Fire fighter practice	AutoPano	103
	Pix4D	1814
	Our Method	423
Wide area monitoring	AutoPano	217
	Pix4D	(only one UAV) 1378
	Out Method	504

The test system employs an Intel Core Duo processor with 2.4 GHz and 4 GB memory

Table 3 Detailed distance measurements of all three methods in cm

Id	True length	AutoPano	Pix4D	Our method
a	4.03	4.23		4.02
b	4.00	3.75	4.04	3.95
c	4.00	3.20		3.87
d	4.02	3.94	4.02	4.03
e	4.03	4.11	4.01	4.06
f	4.00	4.75	4.04	3.96
g	6.06	5.93	6.15	6.12
h	6.06	5.94	6.19	6.15
i	72.08	68.02	72.11	72.57
j	72.12	65.32	72.14	73.11
k	20.03	14.63	20.34	20.97

The deviation from the ground truth of our approach is less than 13 cm over the whole mosaic in north to south direction (Table 3). The reference from Pix4D delivers slightly more robust spatial relations of less than 5 cm while the 2D mosaic

[5] http://www.kolor.com/ visited on December 5th 2013.

[6] http://pix4d.com/ visited on December 5th 2013.

from AutoPano shows poor spatial relations with an average deviation of 30 cm. In average our results can be compared to the output of Pix4D by showing an equal deviation of 5 cm which is satisfying since Pix4D was not able to mosaic the whole area.

6 Conclusion

In this work we presented an incremental mosaicking approach of aerial images from multiple small-scale UAVs flying at low altitudes. Our approach was driven by dedicated applications such as disaster response management over wide areas where the mosaic should be generated as quick as possible. We faced the challenges of mosaicking dense structured scenes and managed limited communication capabilities. Our achieved mosaics preserve spatial relations and show reduced distortions compared to traditional mosaicking methods. Due to the prioritized data transmission and the incremental processing of data teamed with structure based mosaicking we successfully generate a growing mosaic by utilizing limited resources efficiently. In terms of the visual quality of the mosaic we cannot compete with expensive and optimized 3D reconstruction methods consuming unlimited resources. But we deliver an overview mosaic in less than a tenth of the processing time when we consider the whole set of images. Moreover, we observed that our network scheduling together with the progressive image encoding is crucial for the incremental mosaicking. Preselecting important data on the UAVs lead to the optimal utilization of available communication channels.

Finally, operators, e.g, during the fire fighter practice scenario, are satisfied with the intermediate semi-transparent mosaicking results where they are able to manually and quickly identify objects. Our mosaicking approach is robust if a ground plane exists in the scenario and it is clearly visible in many images but might will deliver a wrong projection plane if no ground plane can be determined. These issues are already considered in our future work of allowing more degrees of freedom for the projection plane.

Acknowledgments This work was performed in the project *Collaborative Microdrones* (*cDrones*) of the research cluster Lakeside Labs and was partly funded by the European Regional Development Fund, the Carinthian Economic Promotion Fund (KWF), and the state of Austria under grant KWF-20214/17095/24772.

References

1. Ahmad, A., Samad, A.: Aerial mapping using high resolution digital camera and unmanned aerial vehicle for Geographical Information System. In: Proceedings of the 6th International Colloquium on Signal Processing and Its Applications (CSPA), pp. 1–6 (2010)
2. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A survey on wireless multimedia sensor networks. J. Comput. Netw. **51**(4), 921–960 (2007)
3. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. In: Society for Industrial and Applied Mathematics. Philadelphia (1999)
4. Arya, S., Mount, D.M.: Approximate nearest neighbor queries in fixed dimensions. In: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, pp. 271–280. Society for Industrial and Applied Mathematics (1993)
5. Blackford, L.S., Petitet, A., Pozo, R., Remington, K., Whaley, R.C., Demmel, J., Dongarra, J., Duff, I., Hammarling, S., Henry, G.: An updated set of basic linear algebra subprograms (BLAS). ACM Trans. Math. Softw. **28**(2), 135–151 (2002)
6. Bradley, A.P., Stentiford, F.W.: JPEG 2000 and region of interest coding. In: Proceedings of Digital Image Computing Techniques and Applications (DICTA2002), pp. 1–6. Melbourne (2002)
7. Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Trans. Graph. **2**(4), 217–236 (1983)
8. Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Unmanned aerial vehicle localization based on monocular vision and online mosaicking. J. Intell. Robotic Syst. **55**(4), 323–343 (2009)
9. Cheng, Y., Xue, D., Li, Y.: A fast mosaic approach for remote sensing images. In: Proceedings of the International Conference on Mechatronics and Automation (ICMA), pp. 2009–2013 (2007)
10. Colomina, I., Molina, P.: Unmanned aerial systems for photogrammetry and remote sensing: A review. ISPRS J. Photogramm. Remote Sens. **92**, 79–97 (2014)
11. Coopmans, C., Han, Y.: AggieAir: an integrated and effective small multi-UAV command, control and data collection architecture. In: Proceedings of the 5th ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications (MESA09), pp. 1–7 (2009)
12. Daniel, K., Dusza, B., Lewandowski, A., Wietfeld, C.: AirShield: a system-of-systems MUAV remote sensing architecture for disaster response. In: Proceedings of the 3rd Annual IEEE Systems Conference, pp. 196–200 (2009)
13. Elibol, A., Kim, J., Gracias, N., Garcia, R.: Efficient image mosaicing for multi-robot visual underwater mapping. Pattern Recognit. Lett. **46**, 20–26 (2014)
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
15. Frescura, F., Giorni, M., Feci, C., Cacopardi, S.: JPEG2000 and MJPEG2000 transmission in 802.11 wireless local area networks. IEEE Trans. Consum. Electron. **49**(4), 861–871 (2003)
16. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. **3**(3), 209–226 (1977)
17. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
18. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN:0521540518
19. Hui, J.W., Culler, D.E.: IP is dead, long live IP for wireless sensor networks. In: SenSys'08: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. ACM Request Permissions (2008)
20. Mersheeva, V., Friedrich, G.: Routing for continuous monitoring by multiple micro UAVs in disaster scenarios. In: Proceedings of the European Conference on Artificial Intelligence, pp. 588–593 (2012)
21. Pratt, K.S., Murphy, R., Stover, S., Griffin, C.: CONOPS and autonomy recommendations for VTOL small unmanned aerial system

based on Hurricane Katrina operations. J. Field Robotics **26**(8), 636–650 (2009)

22. Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., Rinner, B.: Networked UAVs as aerial sensor network for disaster management applications. e & i Elektrotechnik und Informationstechnik **127**(3), 56–63 (2010)
23. Roßmann, J., Rast, M.: High-detail local aerial imaging using autonomous drones. In: Proceedings of 12th AGILE International Conference on Geographic Information Science: Advances in GIScience, pp. 1–8. Hannover (2009)
24. Sturm, P.: Multi-view geometry for general camera models. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005 (CVPR'05), pp. 206–212 (2005)
25. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Computer Vision—ECCV'96 1065(Chapter 61), 709–720 (1996)
26. Thite, S.: Smallest enclosing circle of a set of points in the plane. Website. http://www.win.tue.nl/-sthite/mincircle/ (2000) last visited on July 3rd 2013
27. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment–a modern synthesis. Vis. Algorithms Theory Pract. **1883**, 298–372 (2000)
28. Turkbeyler, E., Harris, C., Evans, R.: Building aerial mosaics for visual MTI. In: Proceedings of the 5th EMRS DTC Technical Conference. Roke Manor Research, Romsey, Hampshire SO51 0ZN, Edinburgh (2008)
29. Wischounig-Strucl, D., Quartisch, M., Rinner, B.: Prioritized data transmission in airborne camera networks for wide area surveillance and image mosaicking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 17–24 (2011)
30. Yahyanejad, S., Wischounig-Strucl, D., Quaritsch, M., Rinner, B.: Incremental mosaicking of images from autonomous, small-scale UAVs. In: Proceedings of the 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance (2010)
31. Yanmaz, E., Kuschnig, R., Bettstetter, C.: Channel measurements over 802.11a-based UAV-to-ground links. In: Proceedings of the IEEE GLOBECOM Workshops, pp. 1280–1284 (2011)

Daniel Wischounig-Strucl received the M.Sc. degree in Telematics from Graz University of Technology, Austria in 2006. He received his Ph.D. degree in Computer Science from Alpen-Adria-Universtät Klagenfurt, Austria in 2013. After four years in industry in the field of signal processing at CISC Semiconductor GmbH, Austria he held research positions from 2009 to 2013 with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt. His current activities include teaching and research in embedded computing, signal processing, 3D reconstruction, modelling and computer vision.

Bernhard Rinner (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in Telematics from Graz University of Technology, Austria in 1993 and 1996, respectively. He is full professor and chair of pervasive computing at Klagenfurt University. He held research positions with Graz University of Technology from 1993 to 2007 and with the Department of Computer Science, University of Texas at Austin, from 1998 to 1999. His current research interests include embedded computing, embedded video and computer vision, sensor networks and pervasive computing. Prof. Rinner has been co-founder and general chair of the ACM/IEEE International Conference on Distributed Smart Cameras and has served as chief editor of a special issue on this topic in the Proceedings of the IEEE and IEEE Computer.