Prioritized Data Transmission in Airborne Camera Networks for Wide Area Surveillance and Image Mosaicking

Daniel Wischounig-Strucl Markus Quartisch
Bernhard Rinner
Institute of Networked and Embedded Systems
Alpen-Adria-Universitaet Klagenfurt, Austria
<firstname.lastname>@uni-klu.ac.at

Abstract

Unmanned aerial vehicles (UAVs) are an emerging research area and we equip these with high resolution cameras and build a wireless network for wide area surveillance. The sensed telemetry data and images are processed on-board in a distributed manner to generate an orthographic mosaick and augment with sensed data.

In this work we present a prioritized data transmission scheme for a wireless network of mobile aerial camera nodes for wide area surveillance. The goal of this protocol is to transfer the telemetry data, mosaicking data and images efficiently over the limited wireless network such that an overview image can be generated incrementally. Our experiments with up to four UAVs demonstrate very short delays for the final mosaick, due to the prioritization by the network protocol. Low resolution image data and meta data for mosaicking is prioritized over the full sized image data.

1. Introduction

Various applications require wide area surveillance systems to be deployed for frequent monitoring of an inaccessible area. Such examples include the response management after severe disasters, the monitoring of large construction sites, and agricultural mapping, among others. In some use cases a very short launch phase of the system is of outermost importance. Taking into account the limited resources, especially human resources.

Considering given requirements, our camera network for wide area surveillance is built from mobile camera nodes that are able to place themselves autonomously. A wireless network is used to transmit sensed data, i.e., high resolution images, position and orientation data, and data from additional environmental sensors independent of existing infrastructures. In areas of bad accessibility mobile cameras bound to the ground may fail due to a very limited range of view compared to aerial cameras. As a consequence, we use highly mobile camera nodes which are basically small-scale unmanned aerial vehicles (UAVs) equipped with cameras.

In this work we present a mobile camera network deployed on aerial vehicles with embedded processing capabilities. The goal of the presented system is to provide an online overview image and augment it with additional information about the scene, gained from all sensors on the UAV. The processing and analysis of the sensed data, such as captured high resolution images, is executed distributely on the camera nodes, while intermediate results are delivered to the ground station.

We propose a custom network protocol that deals with the scheduling of prioritized images and meta data to utilize the limited communication channel efficiently. During mission execution, processing results are used to provide feedback to the UAV's control to allow active interventions on certain events, such as, the absence of area coverage or bad image quality.

The remainder of this paper is organized as follows: Section 2 gives a short overview on related work. In Section 3 the use case and system architecture is introduced, while Section 4 elaborates the distributed mosaicking and presents our new image delivery protocol. Section 5 presents evaluation results and finally Section 6 concludes the paper and gives some outlook on future work.

2. Related Work

Wireless camera networks on mobile platforms introduce additional challenges in terms of bandwidth scheduling and resource planning. In [1] wireless sensor networks built from off-the-shelf cameras are able to ubiquitously retrieve video and still images, among other sensor data, from the environment. When avoiding visual sensors among the deployed sensors the bandwidth requirements are lesser, while the coverage constrains may get more complex due to the limited sensing range. The Airshield project [4] presents a system utilizing different wireless networking technologies, such as WiMAX and wireless LAN 802.11g, for environmental monitoring.

The objective of resource optimization is to adapt the available resources, e.g., energy and communication bandwidth, such that the data is transmitted according to best effort principle, i.e., a fair approach where all packets are treated equally [13]. Shiang *et al*. [12] compare a centralized approach for surveillance networks, a congestion game approach and a distributed greedy approach to efficiently share the available wireless network resources and transmit captured information from cameras to a central ground station. The project AggieAir [3] covers a control system with human operators steering UAVs that use two separated network architectures for control data and sensed data.

In the work of He and Wu [7] wireless video sensor networks are explored according to their resource utilization. The behavior of a wireless video sensor node and its performance under resource constraints are analyzed. The examined resource constraints include limitations with respect to energy supply, on-board computational capabilities and transmission bandwidth.

Aerial camera networks have gained importance over the past years, though they require more complex and active communication links to transmit telemetry data in addition to images. Communication over of cellular networks for establishing reliable air-to-ground links is one approach, besides building a wireless infrastructure. In [5] the authors present a coverage analysis for cellular networks for low altitude operational areas based on aerial RSSI measurements.

In our system the dynamic placement, wide area coverage and the network communication play central roles. The processing is executed distributed among the camera nodes and a heterogeneous network as proposed in [16] is used for the control path and scheduled data transfer.

3. System Description

In our use case of wide area surveillance the available human resources are very limited. Hence, our proposed wide area surveillance system works as unsupervised and autonomous as possible. Autonomous operation allows the handling of multiple UAVs, from take-off to landing, with a minimum of expertise and training of operators required.

Definition of the surveillance area. The operator initiates a mission by just drawing the observation area borders and marking known obstacles on an arbitrary map, cf. Figure 1. Different observation properties can be defined for subareas, e.g., the update frequency or the target resolution. The mission planning and execution is accomplished completely autonomously according to the methods by Quaritsch *et al*. in [11].

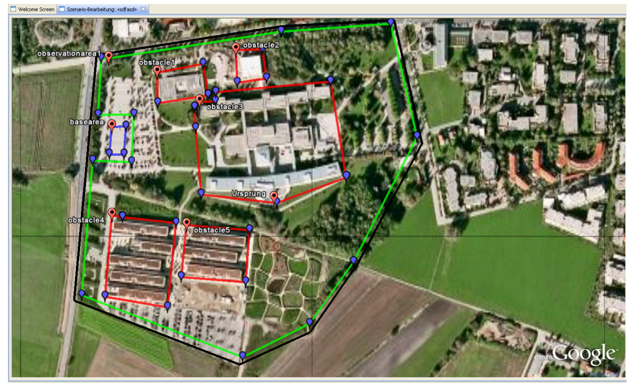


Figure 1. Example of a sketched scenario in the scenario editor by the operator. On an arbitrary map the operator marks the observation area green and marks obstacles in red.

Picture point planning. In the first planning step picture points, i.e., GPS coordinates where the cameras are actively positioned to capture images, are computed to cover the observation area efficiently. Heterogeneous camera properties, such as the allowed altitude or field of view, are considered.

Route planning. In the second step the route planning is executed for multiple vehicles, considering physical limits of single UAVs, such as battery capacity, flight speed, among others.

Finding the optimal solution for the route planning is computationally very expensive because the modeled *capacitated vehicle routing problem* is an NP-hard problem [20]. An initial estimation based on a clustered Christofides heuristic, presented in Figure 2, is sent to the camera nodes for an immediate mission start. This plan is refined during flight by Variable Neighborhood Search as proposed in [8].

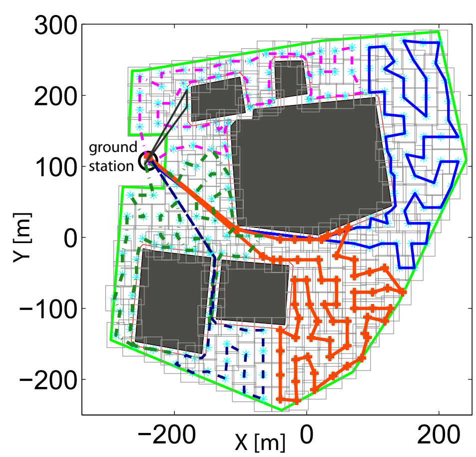


Figure 2. The mission plan of the sketched scenario results in six routes for camera nodes.

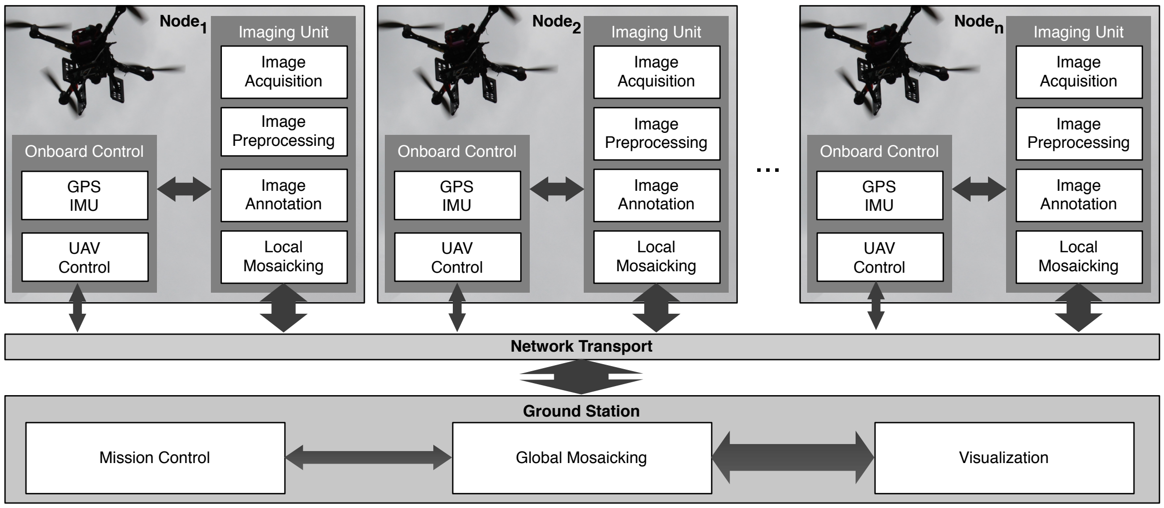


Figure 3. System overview presenting multiple camera nodes with their processing components and the common communication layer.

3.1. System Architecture

Our proposed system for monitoring unknown wide areas compromises three main components presented in Figure 3:

Mobile Camera Nodes: Each of the aerial camera nodes consist of the UAV itself, sensors such as a high resolution camera and two processing units.

Network: A wireless network infrastructure is deployed for transmitting captured images and sensed meta data. Our proposed custom application layer protocol schedules the transmission of captured data whereas it does not rely on any specific transport protocol.

Ground Station: Single images are merged to an overview image according to the preprocessed mosaicking data. The result is presented to the operator.

The *Onboard Control* of each camera node contains the *GPS and IMU* sensors and the *UAV Control* unit for flight navigation. It interacts with the *Imaging Unit* to provide meta data for image annotation and responds on quality feedback from the image processing. While control data is exchanged with the *Mission Control* at the *Ground Station* the image and mosaicking data is passed to the *Global Mosaicking*.

4. Distributed Image Aquisition and Processing in the Camera Network

For an efficient mosaicking the processing on the camera node is split into acquisition, preprocessing and local mosaicking. The global mosaicking is executed on the ground station.

Due to limits of the available network bandwidth and wireless network connectivity in the whole area, a complete transmission of the high resolution images may typically not be completed during flight time. Consequently, the mosaicking is started with a rough placement of low resolution images by meta data only and enhanced by higher resolution image data and structure data later. At the ground station we employ such a hybrid mosaicking approach incorporating multiple mobile camera nodes as proposed in our previous work [19].

4.1. Image Processing Pipeline

Images are captured and annotated with meta data from the onboard control. In the preprocessing the image quality is analyzed, images are compressed and feature keypoints are extracted. To generate a rough mosaick the ground station collects the incoming data from multiple camera nodes and incrementally refines the orthographic overview image when more precise image transformations and higher resolution images are available.

4.1.1 Image Acquisition and Image Processing

When the camera node approaches the coordinates, where to capture an image, it decreases its speed to reduce the likeliness of motion blur. The imaging unit assesses the quality of the image in terms of motion blur, determined as proposed in [2] and the view angle of the camera from the IMU data. If the view angle significantly deviates from the nadir view, the expected perspective distortion is too high to gain good results when orthorectifying. This is used as feedback for the onboard control which decides on further actions. Images of a low quality are scheduled with a very low priority, cf. [19].

4.1.2 Multi-Resolution Images

Full sized images from the high resolution camera are too large to be processed or transmitted directly. Hence, each image is converted to progressive JPEG2000. This encoding gains different resolution and quality representations within a file stream without additional overhead.

The progressive JPEG2000 compression is applied with a defined number of layers in resolution-layer-component-position (RLCP) mode which allows the data accumulation from layer to layer. The boundaries of the layer chunks in the image stream and the compression rate for each layer in bits per pixel are accomplished by the compression process. This allows us to split the image into smaller data chunks, cf. Figure 4, which are annotated with different priorities and enqueued for transmission accordingly. The lowest quality and resolution chunk is enqueued with a high priority for transmission.

For the local mosaicking on the camera node also one chunk of lower resolution is necessary. At the ground station the high resolution image is built incrementally, by concatenating the byte stream layer by layer.

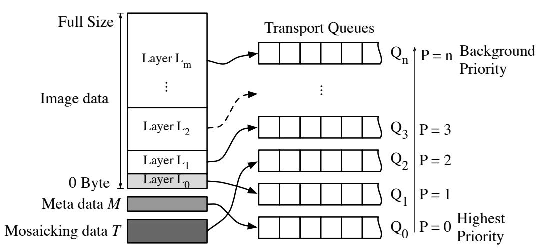


Figure 4. JPEG2000 layers of one full resolution image are presented in the byte stream view. The annotated meta data M, image layer L_0 and mosaicking data T is enqueued to the transportation queues with the highest priorities.

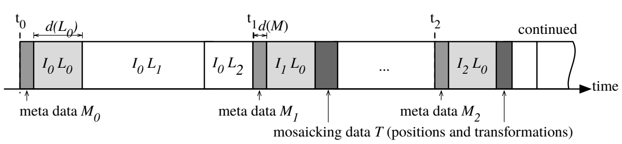
4.2. Prioritized Data Scheduling

For an immediate presentation to the operator an efficient data scheduling and prioritization is necessary. To achieve this scheduled transfers efficiently, we propose a new application layer protocol for prioritized meta data and progressive image data. Our network protocol conducts $n = m + 2$ transmission queues with different priorities, sketched in Figure 4, whereas m is the number of image layers, that can be different for heterogeneous camera nodes. The scheduling is managed on the camera node in the manner of transmitting higher prioritized queues, i.e., Q_0 to Q_2, before sending the remaining image layers.

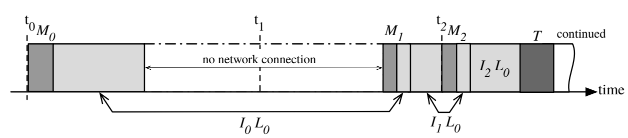
For every new image the meta data M is put to the highest priority queue Q_0, to transmit it to the ground station immediately. The image layers are enqueued for transmission according to their resolution and size to incrementally refine the global mosaick, as proposed in[19]. The interruptible stream representation of JPEG2000, discussed in [15] and [9], allows the ground station to gain additional information from any received data chunk. The mosaicking data T containing image transformations and feature keypoints is transmitted after the lowest image layer L_0.

The ideal case of data scheduling on one camera node is sketched in Figure 5(a), where a certain bandwidth is always available. On one camera node, for example, the first image is taken at t_0 and preprocessed. Its meta data, contained in separate files, is transmitted with the highest priority $P = 0$ to the ground station followed by the lowest resolution and quality layer with a lower priority $P = 1$. At t_1 the next image is captured. Again, with the highest priority its meta data is transmitted to the ground station followed by its lowest resolution layer L_0. The remaining resources $r = \Delta t \cdot bw(\Delta t, p)$ between both captures are available for transmission of additional quality layers $L_i, i \geq 1$ with priority $P = i + 2$ in the background. Where $\Delta t = t_i - t_{i-1} - d(M, L_0)$ is the remaining time between two image captures, d denotes the transmission time of the meta data and the first image layer, and *bw* denotes the available bandwidth during Δt and along the path p_i.

After the processing of at least two images on the imaging unit the computed mosaicking data, i.e., image transformations, camera poses, and structure data, are transmitted with the priority level $P = 2$ subsequently after the image layer L_0 and followed by the remaining layers according to their priorities.



(a) The ideal case, if sufficient bandwidth is available.



(b) In situations of bad network connectivity the data transfer gets stalled, as presented in this example.

Figure 5. The meta data packages and image data is scheduled when sending from the camera node to the ground station.

Our protocol is designed to be robust against link failures due to limited communication range caused by obstacles, long distances or the number of concurrently active camera nodes, among others. Missing packets of one data chunk are optionally and collectively requested after each complete chunk, only if this data is mandatory for the ground station, i.e., typically necessary for Q_0 to Q_2.

Figure 5(b) depicts the transmission of the individual data chunks according to our proposed scheduling scheme in case of missing network connectivity. Meta data and the low-resolution image layer have higher priorities and are thus sent as soon as a connection to the base station is re-established. To overcome the incomplete data transmission, in periods when no new images are taken, such as the landing phase for a battery refill and the take off phase to the first picture point, the remaining higher resolution layers from the background queue are transmitted. Typically closer to the ground station a higher bandwidth is available.

4.3. Incremental Image Mosaicking

Based on available low resolution images the local mosaicking is executed iteratively. Intermediate results are transmitted to the ground station for further processing, because the processing capabilities on the camera nodes are limited. Furthermore, low resolution images are insufficient for feature extraction on the ground station, but satisfying to be mosaicked by intermediate results from different camera nodes and to be visualized. This leads to an incremental, distributed, and resource aware mosaicking method that is presented in the following.

4.3.1 Local Mosaicking

An initial set of at least two images is processed by bundle adjustment on keypoints from low resolution representations, immediately after they are captured from the camera. Overlapping images are determined based on a rough projection estimated from GPS and IMU data. The extracted keypoints within these overlapping areas are the input for a pairwise matching among all overlapping image pairs. The matched keypoints act as input for the bundle adjustment, cf. [10] to compute the scene structure and camera extrinsics, that is incrementally refined when more images are considered.

With the camera extrinsics, the images are perspectively orthorectified, i.e., the projection on the ground plane is applied. Furthermore, the relative camera positions and orientations are decomposed from the camera extrinsics and fused with the GPS and IMU data to provide feedback to the planning.

For an accurate wide area mosaicking it is important to avoid perspective distortions when transforming images. To solve this, our previous work [17] presents an approach that considers only keypoints on a common ground plane. We apply this approach and enhance it by using iterative sparse bundle adjustment instead of pairwise image processing by structure from motion. If the number of matching keypoints is not sufficient in one plane, we try to fit parallel planes at other levels.

As a result, for each image a perspective transformation, i.e., homography, to project the image onto the horizontal ground plane and one similarity transformation for the global mosaicking are computed.

4.3.2 Global Mosaicking, Ground Station Fusion and Presentation

The global mosaicking is executed according to the same method as the local mosaicking. However, for global mosaicking the mosaicking data from all camera nodes is accumulated and used as initial estimation for the bundle adjustment. Additional keypoints that are not considered by the local mosaicking, i.e., keypoints within non-overlapping areas, are matched with keypoints of images from neighboring camera nodes and added to the bundle adjustment.

In some applications it is required to present a wide area overview image quickly by accepting quality trade-offs. To achieve this, initial image projections gathered from the local mosaicking data are applied to low resolution images before high resolution images are available. In Figure 6 such a resulting mosaick is presented immediately to the operators. This refinement process is iterative and increasing the resolution and quality with every additional image according to the following procedure.

1. A rough placement of images is executed, just by the initial camera projection obtained from the annotated meta data.
2. If additional resolution levels are received for already mosaicked images the transformations are adapted.
3. The output of the local mosaicking and the global bundle adjustment deliver enhanced image transformations. Such transformations consist of an orthorectifying projective component and similarity transformation to place the image.
4. If additional images with mosaicking data are received the global fusion is re-executed.

After updating images in the available resolution they are blended with subjacent images according to the methods presented in [6, 18]. Nearby images are only re-placed if their transformations have been changed due to mosaicking in their neighborhood.

4.4. Feedback to Planning

The *Mission Control* module at the ground station (cf. Figure 3) monitors and adapts the ongoing mission by exchanging data with the *UAV control* of individual camera nodes. Furthermore, the *Mission Control* is informed on the analysis results of the mosaicking to adapt flight plans, e.g., to redistribute cameras to uncovered areas. The following information is provided as feedback to the *Mission Control*:

Figure 6. Presentation view of the high resolution mosaick and current mission state (red trajectory). The most recent images are immediately placed on top, while outdated mosaick is faded in the background.

Image Quality: The blurring and perspective distortion of images are detected in the preprocessing and verfied if these parameters are acceptable to improve the overview image from these image's orthorectified projections.

Coverage: The coverage contribution of single images is evaluated in terms of area and ground resolution and afflects the global mission planning if subareas are covered insufficiently.

5. Evaluation & Results

The evaluation of our proposed camera network focuses on the image transmission and network performance. High resolution images are taken during real flights at the planned picture coordinates and processed onboard, as described earlier. Each camera node is equipped with an embedded processing unit operated by Linux and various sensors, such as low cost GPS and IMU sensors, besides a modified Canon PowerShot S80 high resolution camera attached via USB.

Images are transferred to the ground station via our network protocol from four concurrently active nodes in the test setup. A communication interface on each camera node based on wireless LAN 802.11g is used as control path and for sensed data transmission.

Implementations are realized in Java to integrate well into the Java based framework developed for operator interaction, planning and presentation among other components.

Image Compression

Images are compressed on the camera node by using the Kakadu library [14] for image processing that is integrated via Java Native Interface. The Kakadu library allows a detailed parameterization of JPEG2000 when encoding or decoding. Among the adjusted parameters, the compression

Layers	Resolution [px]	Rate [bpp]	Total [B]	Chunk [B]
4	3264×2448	2.85	2 853 015	1 580 345
3	1632×1224	1.17	1 272 670	563 849
2	816×612	0.48	708 821	587 167
1	408×306	0.10	121 654	121 654
Meta Data M			867	867
Mosaicking Data T			22 136	22 136

Table 1. Example data chunk sizes, in Bytes, of a JPEG2000 compressed image with the meta data and mosaicking data. The given compression bit rate of each layer is related to the full resolution

order is set to resolution-layer-component-position (RLCP) progressive to achieve scalable images in terms of resolution. The encoding ends up in a rate of 2.8 bits per pixel (bpp) on average for the full-sized image, which yields to good results. The number of compression layers are set to four, what results in the resolution and quality levels presented in Table 1. With these parameters the average encoding time on the target platform is 1.8 seconds where the target file size is around 2.8 MB.

Network Performance

In the test setup one AP Linksys WRT54GL was deployed at the ground station in 1 m height and configured with 20 dBm output power in the 2.4 GHz band. To stress our network protocol we intentionally limit the coverage of the wireless network. The protocol is implemented in Java and is typically independent of the underlying transport layer. In our tests the transport protocol UDP is employed underneath. The performance of our network protocol implementation was tested with one, two, and four concurrent camera nodes with route lengths of 829 s, 923 s, 938 s, and 1011 s. These representative routes are selected from the six planned routes, cf. Figure 2. During this mission, the longest line-of-sight distance of 496.45 m is too far for a reliable wireless communication with the required bandwidth for image transfers. Furthermore, the average flight time between two consecutive picture points ranges from 9.76 s to 12.1 s, meanwhile the sensed data is processed and enqueued for transmission.

Based on simulations, the estimated reception power within the observation area is presented in Figure 7. Obstacles, marked as dark grey areas, reduce the available network coverage. For a single camera node the concentric circles mark the simulated link rate boundaries, whereby the green border represents the observation area. It can be clearly seen, that we have to assume an insufficient network connectivity in the outer areas of the observation area.

In the main test one mission is executed where the camera nodes are flying along their routes at an altitude of 40 m simultaneously. In Figure 8 the results of the network tests are shown separately for meta data M, mosaicking data T and for each of the four image layers L_0 to L_3. It can be

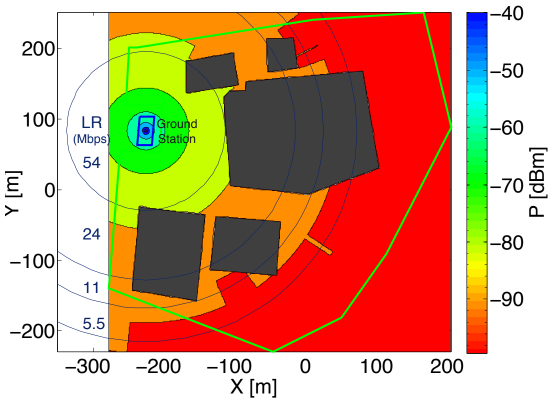


Figure 7. The available reception power at a specific scenario considering path loss due to obstacles (dark grey) at a transmit power is 20 dBm at 2.4 GHz. The dark blue concentric circles around the ground station sketch the optimum free space link rate limits.

seen, if the camera node is out of network range the data transmission is stalled (cf. horizontal lines in the graph). Furthermore, the gradient depends on the available bandwidth. The grey lines mark the generated data on the camera nodes, while the colored lines are the currently transferred and accumulated data. The three most important transport queues Q_0 to Q_2 contain data chunks of approximately 1 kilobyte for meta data, up to 140 kilobytes for the lowest image resolution and 24 kilobytes for mosaicking data.

It can be explored that the meta data of node$_1$ and node$_3$ was delivered with a significant delay, but still before the mission was completed. Whereas the high resolution image data of those nodes was not able to be delivered during the active phase, i.e., while new images are captured frequently. Hence, all lowest resolution layers L_0 and the meta data M from all participating camera nodes is transmitted completely during the mission. Higher resolution image layers are not completely transmitted, cf. layer L_1 to L_3 for node$_1$ and layer L_3 for camera node$_3$.

In Figure 9 we compare our network scheduling protocol with the continuous image transmission over TCP. The information gain at the ground station, i.e., the visual information the operator can receive from the presented mosaick, is computed as ratio of the received mosaicking data and images to the generated data on the camera node.

A global mosaick can already be generated from the meta data, low resolution images and preprocessed mosaicking data, thus we weight this data with 50 % of the total information. When using continuous image transmission via TCP the global mosaicking does not receive sufficiently image data immediately, to improve the total information presented to the operator. The mean benefit of our scheduling protocol is 22.1 % with a standard deviation of 6.93 % in the tested scenario. In the first few seconds no advantage

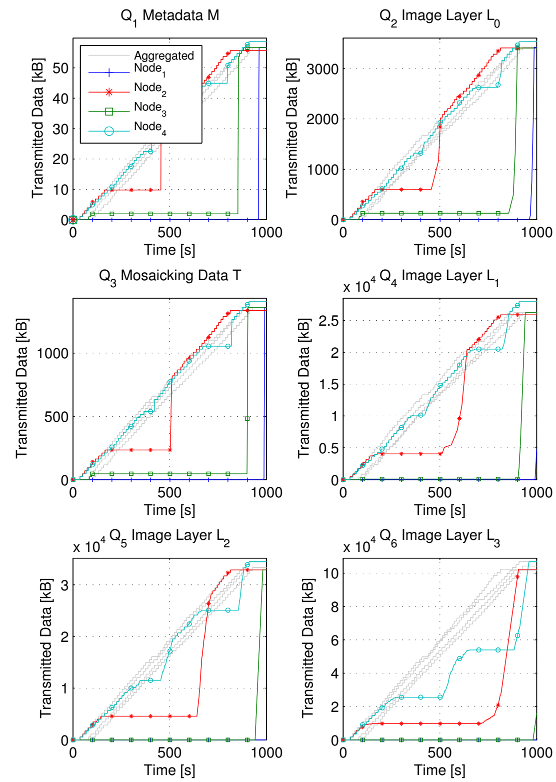


Figure 8. The accumulated transmission data of four concurrently active camera nodes is presented here for all six prioritization queues in this test setup. The grey graphs represent the data that is generated at individual imaging units.

can be explored, because the available bandwith can handle all aggregated data. As soon as some nodes are out of range or within areas of low bandwidth our proposed scheduling shows its benefits of up to 40.87 %.

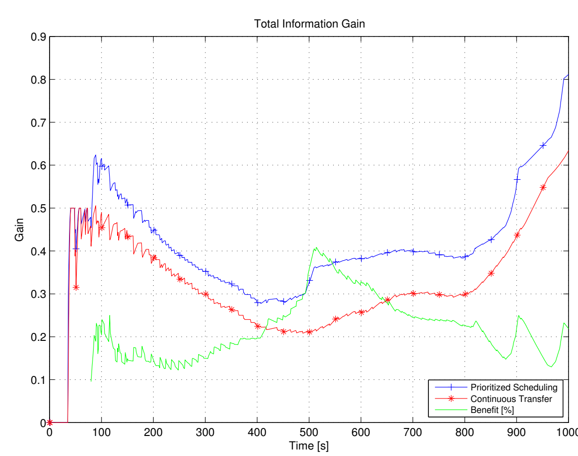


Figure 9. The total information gain at the ground station for the prioritized and continuous transmission. An information gain of 1.0 is reached when all generated data is transmitted.

6. Conclusion

In this work we have shown that our image transfer protocol outperforms the continuous transmission over TCP over the wireless network in a wide area within the given constraints. The available bandwidth is optimally utilized with the prioritization of data. By exploiting JPEG2000 and its incremental stream mode the mosaicking could be executed even though high resolution layers arrived very late. Low resolution representations are transmitted with an acceptable delay and presented immediately using a hybrid mosaicking approach. We discovered that a global prioritization among all participating camera nodes could further improve the response time for the mosaicking. The distributed mosaicking process could further be improved by dynamically arranging tasks and resources among the camera nodes to exchange mosaicking data and enhance the local mosaicking.

Acknowledgment

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] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Journal of Computer Networks*, 51(4):921–960, 2007.

[2] X. Chen, J. Yang, Q. Wu, and J. Zhao. Motion blur detection based on lowest directional high-frequency energy. In *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, pages 2533–2536, September 2010.

[3] C. Coopmans and Y. Han. 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)*, pages 1–7, 2009.

[4] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld. AirShield: A system-of-systems MUAV remote sensing architecture for disaster response. In *Proceedings of the 3rd Annual IEEE Systems Conference*, pages 196–200, March 2009.

[5] N. Goddemeier, K. Daniel, and C. Wietfeld. Coverage evaluation of wireless networks for Unmanned Aerial Systems. In *Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps)*, pages 1760–1765, December 2010.

[6] N. Gracias, M. Mahoor, S. Negahdaripour, and A. Gleason. Fast image blending using watersheds and graph cuts. *Image and Vision Computing*, 27(5):597–607, April 2009.

[7] Z. He and D. Wu. Resource allocation and performance analysis of wireless video sensors. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(5):590–599, May 2006.

[8] V. Hemmelmayr and K. Doerner. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802, 2009.

[9] X. Li and J. Cai. Robust Transmission of JPEG2000 Encoded Images Over Packet Loss Channels. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 947–950, July 2007.

[10] M. Lourakis and A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):2:1–2:30, March 2009.

[11] M. Quaritsch, R. Kuschnig, H. Hellwagner, and B. Rinner. Fast Aerial Image Acquisition and Mosaicking for Emergency Response Operations by Collaborative UAVs. In *Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, May 2011.

[12] H.-P. Shiang and M. v. d. Schaar. Information-Constrained Resource Allocation in Multicamera Wireless Surveillance Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(4):505–517, April 2010.

[13] S. Soro and W. Heinzelman. A Survey of Visual Sensor Networks. *Advances in Multimedia*, 2009(2009):21, January 2009.

[14] D. Taubman. Software architectures for JPEG2000. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, volume 1, pages 197 – 200 vol.1, 2002.

[15] N. Thomos, N. Boulgouris, and M. Strintzis. Optimized transmission of JPEG2000 streams over wireless channels. *IEEE Transactions on Image Processing*, 15(1):54–67, January 2006.

[16] T. Winkler and B. Rinner. Pervasive smart camera networks exploiting heterogeneous wireless channels. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, pages 1–4, March 2009.

[17] D. Wischounig-Strucl, M. Quaritsch, and B. Rinner. A Structure Based Mosaicking Approach for Aerial Images from Low Altitude of Non-Planar Scenes. In A. Wendel, S. Sternig, and M. Godec, editors, *Proceedings of the 16th Computer Vision Winter Workshop*, pages 51–58. Graz University of Technology, February 2011.

[18] Y. Xiong and K. Pulli. Fast image stitching and editing for panorama painting on mobile phones. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–52, June 2010.

[19] S. Yahyanejad, D. Wischounig-Strucl, M. Quaritsch, and B. Rinner. Incremental Mosaicking of Images from Autonomous, Small-Scale UAVs. In *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, September 2010.

[20] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.