Towards Drone-sourced Live Video Analytics for the Construction Industry

Shilpa George Carnegie Mellon University shilpag@cs.cmu.edu

Thomas Eiszler Carnegie Mellon University teiszler@cs.cmu.edu Junjue Wang Carnegie Mellon University junjuew@cs.cmu.edu

Padmanabhan Pillai Intel Labs padmanabhan.s.pillai@intel.com Mihir Bala University of Michigan mihirkb@umich.edu

Mahadev Satyanarayanan Carnegie Mellon University satya@cs.cmu.edu

Abstract

This paper investigates the use of drones for live inspection in the construction industry. The key technical challenge is the real-time registration of the drone video feed to the architectural plan. We present and evaluate three different approaches for registration and propose an edge-based prototype using visual features. Our evaluations show that GPS alone is not sufficient for accurate registration, but with visual features, accuracies within ten centimeters can be achieved.

ACM Reference Format:

Shilpa George, Junjue Wang, Mihir Bala, Thomas Eiszler, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2019. Towards Drone-sourced Live Video Analytics for the Construction Industry. In *The 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19), February 27–28, 2019, Santa Cruz, CA, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3301293.3302365

1 Introduction

The U.S. construction industry is estimated to be over \$1.8 trillion by 2022 [4]. A recent McKinsey report [1] estimates that productivity in this industry has only improved at about 1% per year over the past 20 years. The use of drones for inspection of progress at construction sites, with prompt detection and reporting of construction errors, has emerged as a promising approach to improving productivity [6].

The use of drones in construction faces two challenges. First, performing real-time analytics on drone-sourced videos requires more processing power than what is available on a small, lightweight and inexpensive drone. Second, drones typically run Linux or an embedded RTOS, while engineering and architectural software such as Autodesk Revit and Navisworks are only implemented for Windows. Direct, real-time registration of drone observations on engineering drawings offers the highest potential for productivity improvement, yet this is not feasible today. Even when mobile devices such as tablets are used at construction sites today, they are only used to view a PDF file of engineering drawings. This is only slightly better than viewing a hard-copy printout.

HotMobile '19, February 27-28, 2019, Santa Cruz, CA, USA

https://doi.org/10.1145/3301293.3302365

1

to enable drone-sourced video analytics for live building inspection. The substantial computing resources of a cloudlet are used to meet the high computational demands. At the same time, the virtualization capability of a cloudlet is used to run a Windows virtual machine (VM) that contains the original engineering and architectural drawings for the construction site as well as the Autodesk software that was used to create those drawings. Accurate real-time registration of drone observations on drawings emerges as a key challenge in this setting. We focus on addressing the following problem: When the analytics pipeline detects a construction defect in the drone-sourced video, where exactly on the corresponding drawing should the error be noted? While localization has been studied extensively [11] [14], our focus is not on absolute coordinates but on coordinates relative to an engineering drawing. One can view this problem as an inverted form of augmented reality (AR): rather than annotating the real world with virtual observations (classic AR), our goal is to annotate a virtual world (engineering drawing) from real-world observations. A person living in the virtual world (e.g., an engineer or architect at a remote Internet location), should be able to see annotations created in real-time by a drone flight.

In this paper, we describe our experience in using edge computing

The main goal of this paper is to illustrate the potential of using drones for live video analytics in construction industry by showcasing building inspection as a use case. We briefly introduce the current uses of drones in construction industry in Section 2 and describe the case study conducted and its setup in Section 3. We explore the performance of live registration and the accuracy obtained using various techniques from Section 4 to Section 6. In Section 6 we present our proposed solution of using feature-based registration. We describe related work in Section 8. In Section 9 we have a discussion and describe areas of future research and finally conclude in Section 10.

2 Background: Drones in Construction

The use of drones in the construction industry is still in its infancy. Drones have been used to inspect hard-to-reach parts of a constructions site, and to take ad-hoc aerial photographs of the site. Systematic use of drones to improve productivity of the construction process is just beginning.

One of the most sophisticated uses of drones in construction to date involves the production of a 360° time-lapse video of the construction progress. This is accomplished by periodic preciselycontrolled flights around a site to capture high-resolution imagery from all directions. After each flight, the captured data is sent to the cloud for offline processing. As the position of the drone in flight

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

^{© 2019} Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6273-3/19/02.



Figure 1: Completed Tepper Building (October 2018)



Figure 2: Revit 3D model of Tepper Building

can never be perfectly controlled, the offline processing will visually match the captured images to each other and to images from past flights. These are then stitched together and presented as an HTML5 visualization that lets one view the progress of the construction from any direction and over time. Although this is a valuable capability, it has two key limitations. First, the process is inherently offline. Second, manual effort is needed to register drone observations on engineering drawings.

Our goal is to overcome both these limitations through a a live, interactive drone-based inspection system. During a drone flight, captured video is streamed to a ground-based cloudlet. The system generates a synthetic view, registering and superimposing the live drone view on top of the corresponding portions of the engineering drawings. An important benefit of live video analytics over offline analytics is that it is possible to actuate the drone in real time. This allows zooming in/out the camera or changing the flight path of the drone based on current observations, thus enabling immediate followup of trouble spots, and re-imaging (typically closeups) of specific features. Accurate online positioning is critical for other construction applications such as live thermal mapping, where thermal imagery could be used to identify leaks, and inspect them up close within the same drone flight.

3 Case Study: CMU Tepper Building

The Tepper Building is a new 315,000 square feet building that was completed in September 2018 to house the Business School at Carnegie Mellon University. Fig. 1 shows this building shortly after completion of construction. Our use of drones in this construction project occurred close to its completion, in the late spring and summer of 2018.

We obtained detailed engineering plans of this building from the architectural firm that designed the building. These plans were provided as a set of files for the Revit software package. The plans include both 3D models of the building, as well as 2D elevations from different directions (e.g., Fig. 2 and Fig. 3). Industry standards specify the accuracy of these inputs to be within one-quarter inch (roughly 0.6 cm) of actual construction. This quantity serves to



Figure 3: Two Revit Elevation Views of Tepper Building

bound the drone positioning accuracy that we seek to achieve — there is no point in being more accurate than the plans themselves!

For drone flights, we used a DJI Phantom 4 Pro UAV that is equipped with a variety of sensors (GPS, altimeter, etc.) and a 4K steerable video camera. During each flight, we capture both the video and sensor data, and stream these to a ground-based cloudlet. The cloudlet has a copy of the building model, and attempts to register and overlay the drone video on top of the model view. The cloudlet we used to develop our prototype has an eight-core Intel[®] CoreTM i7-5960X processor (3.00GHz) with 32 GB of memory and an NVIDIA GeForce GTX 960 GPU.

Figure 4 illustrates our overall system architecture. The drone communicates over DJI's proprietary Lightbridge 2 wireless technology [5] with a controller held by a human operator. The wireless channel carries flight control signals from the controller to the drone. In the other direction, it carries video, GPS and other sensor data. The controller relays the video and other sensor streams over WiFi or 4G LTE to the ground-based cloudlet. Two VM instances are configured on the cloudlet. One VM instance runs the Gabriel platform that was originally developed for wearable cognitive assistance [10], but is repurposed here for drone video analytics. This VM also runs a SIFT-based localization pipeline (discussed in Section 6) on the arriving video frames. The extracted value is passed on to the Autodesk VM instance. In an ideal implementation, the latter VM instance would just run the Revit software. However, our initial experiments showed that the current Revit implementation is too slow in redrawing its model to reflect drone updates. We are tracking down the source of this performance slowdown. In the interim, to achieve real-time tracking, we feed the Revit model to a Unity engine that is able perform real-time redisplay in response to drone-sourced location updates to Revit files.

Using this experimental setup, we aim to answer the following questions:

- (1) When relying solely on GPS and position sensors, what is the accuracy of positioning? Is this sufficient to accurately register the drone camera view to the building model?
- (2) Assuming near-perfect computer vision, how accurately can we estimate the positions of certain building features relative to reference features?
- (3) Using real-world computer vision how close can we come to the above ideal?



Figure 4: System Architecture

4 Built-in Drone Position Sensors

If we can accurately determine the position and orientation of the drone, then we can calculate the view frustum of its camera. Assuming the building model has been calibrated to real-world coordinates, we can then determine precisely where the drone camera view intersects the model, and thus localize any visible issues onto the model. In this section we explore if the measurements from GPS and position sensors of the drone are accurate enough to satisfy our requirement.

The specifications of DJI Phantom 4 Pro drone gives Hover Accuracy Range, a loosely related metric to GPS positioning accuracy, to be 0.5 m in the vertical direction and 1.5 m in the horizontal direction. While this specification helps set the expectation for its GPS-based positioning accuracy, exact numbers need to be measured.

Horizontal Accuracy: To measure the horizontal accuracy of the drone's GPS receiver, we compared the distance of two known locations to the calculated distance from the drone's GPS readings using the Haversine formula. Fig. 5 shows the horizontal GPS accuracy results. The average distance measurement error is 1.50 m with a standard deviation of 0.21 m.

Altitude Accuracy: The DJI mobile SDK makes available only the relative altitude with respect to the take-off location rather than absolute altitude from the sea level. To evaluate the accuracy of such relative altitude, we tied ropes of 4 different lengths with a small weight to the drone. We recorded the altitude reading while taking-off by flying the drone straight up to a position in which the rope is tightened and vertical to the ground with the weight barely touching the ground. We then flew the drone briefly for a minute and took another measurement when landing. We took three experiments for each rope length. Fig. 6 shows the measured altitude results. The average take-off altitude errors ranges from 6 cm to 18 cm. The landing altitude errors, ranging from 17 cm to 48 cm, are significantly larger than take-off errors due to sensor drift.

Thus, GPS and barometers alone cannot provide the needed accuracy for our view registration task. Note that in this paper we have not measured orientation errors; these can only further degrade the registration accuracy.

Ground	GPS-based	Absolute
Truth	Measurement	Error
(m)	(m)	(m)
7.62	9.34	1.72
7.62	9.11	1.49
7.62	6.27	1.35
14.94	16.72	1.78
14.94	13.68	1.26
14.94	13.52	1.42

Average error is 1.50 m Standard deviation is 0.21 m

Figure 5: Measured GPS Horizontal Error

5 Near-Ideal Computer Vision

Instead of trying to accurately determine the drone's absolute position, we investigate whether visual features can be used to directly register the drone's views on the engineering drawings. To control variables and get an idea of the theoretical limits of this approach, we use artificial easily-detected features in this section. In particular, fiducial markers [7] [9] [17] have been widely used for positioning and pose estimation for many computer vision and augmented reality applications. These markers leverage judiciously designed patterns to achieve reliable detection of key points and represent the simplest, most accurate visual approach.

We place a few markers at known positions on the wall as reference markers. We use the detected coordinates of reference markers in a test image together with their wall position to calculate the transform matrix (a homography [19]) that relates the pixels in the test image to wall coordinates. Using the calculated homography, we are able to transform any image points to wall coordinates.

To evaluate the accuracy of our method, we placed 100 ArUco markers [9] on a building's sidewall, as shown in Fig. 7. These markers are printed on four pieces of tabloid size paper as 5x5 grids. Each marker is 4.6 cm by 4.6 cm in size and 0.8 cm away from adjacent markers in the same grid. The horizontal and vertical distance among grids are 61.0 cm. We took 54 test images from different viewpoints with our drone. We chose 8 corner markers as reference points and evaluated the view registration errors for the rest of detected markers in each test image. For each test marker, the calculated coordinates of its four corners are compared against their wall positions. Fig. 9 shows the cumulative distribution function (CDF) of the registration errors along the horizontal and vertical directions of the wall. Over 90% of the errors are less than 1 cm while the worst cases are less than 6 cm. These results come close to the allowed errors (0.6 cm) of the engineering drawings and demonstrate the potential of using visual features for precise view registration.

6 SIFT: Real-World Computer Vision

In the real world, it is not possible to have fiducial markers placed around the building to aid inspection. However, there are many naturally-occurring distinctive features such as corners of windows,

Ground	Take-Off Altitude	Landing Altitude
Truth (m)	Error (m)	Error (m)
0.63	0.13 ± 0.10	0.17 ± 0.17
1.27	0.06 ± 0.02	0.41 ± 0.39
1.85	0.18 ± 0.15	0.28 ± 0.15
2.49	0.17 ± 0.15	0.48 ± 0.12





Figure 6: Drone Altitude Error

Figure 7: Fiducial Markers Drone Image

Figure 8: Tepper Reference Images



Figure 9: Registration Error using Fiducial Markers

brick patterns and other architectural components which allow humans to easily find correspondences between the building image and the engineering drawing. A variety of computer vision techniques have been developed to find distinctive regions in an image. In particular, Scale Invariant Feature Transform (SIFT) features [15] describe distinctive regions that characterize an image. These features are robust to transformations such as scale, rotation, and illumination, which often occur in drone images. SIFT has been widely used in various computer vision tasks such as image stitching, image registration, object detection, tracking, and robot localization.

Matching SIFT features across two inputs requires them to have similar visual appearances; however, the engineering drawing does not have similar visual features as a camera image. This makes it impossible to directly match the drone's camera view to the virtual engineering drawing. To overcome this challenge we add a layer of indirection. We first capture a handful of references images — these are wide-angle images of the exterior of the building taken from a few different locations. Fig 8 shows an example reference image of the Tepper Quad building. These reference images are manually registered to the drawings using key features such as corners of walls and windows. During the drone flight, we can now use SIFT matching to find correspondences between the live camera view and the reference images, and then transitively, find the matching coordinates in the engineering drawing.

In our implementation, the cloudlet stores a database of reference images that span the exterior of Tepper building along with an offline, pre-computed set of SIFT features from these images. For an incoming frame or query image from the drone, we first use its GPS coordinates to perform coarse localization and narrow down the number of reference images against which the frame has to be matched. The cloudlet then extracts SIFT features of the query frame and performs approximate nearest neighbor search using the FLANN library [16] to match its features to the SIFT features of the reference images. Thus, for each of the typically three thousand or so features extracted from the query frame, we obtain a candidate match and reference image id. A simple voting scheme is used to retrieve the top 4 reference images having the largest number of candidate matches to the query image.

To register the drone view to the reference image we find the projective transform that maps feature correspondences between the two images. Projective transform or homography requires that the points chosen in a scene be coplanar; this assumption generally holds in our case as facades of a building are largely planar. To estimate the parameters of the homography transform we use the RANSAC algorithm [8]. RANSAC also helps us refine our feature matches as it removes spurious matches that do not fit our hypothesis. Fig. 11 shows an example SIFT feature match between a query image and a reference image. The red bounding box shows the region on the reference image to which the query image maps using the computed homography transform. A query image is considered to be correctly registered if the best homography returned by RANSAC has at least 20 inlier points.

Once we register the drone view to a reference image, we need to find its position on the engineering drawing. As we had already computed and saved the mapping from the reference image to the engineering drawing, we simply need to multiply the homography obtained above by this mapping to produce a single transform from query image pixels to coordinates of the engineering drawing. Fig. 10 gives the overall flow of our registration algorithm.

7 Evaluation

We evaluate the accuracy of our localization approach by calculating the error between the estimated position of particular points and the ground truth positions in the engineering drawings. For this, we selected distinct points such as corners of windows on the engineering drawing whose positions in the coordinate system are known. We manually find the pixel coordinates of these points in query images, compute their positions in the engineering drawing based on the output of our registration algorithm, and compare these values to the actual locations in drawings. Fig. 12 shows the CDF of positioning error for 548 points over 167 query frames. Most of the time, our system is able to localize the camera view to the engineering drawing to within 10 cm of accuracy, and within 20 cm over 95% of the time. Note that our system is not able to find a valid mapping for every frame. As mentioned above, if a sufficient number of inliers is not



(a) Step 1: Register query image on the reference image

(b) Step 2: Project the mapping on to engineering drawing

Figure 10: Registration pipeline



Figure 11: Example of SIFT feature matching between query image (left) to reference image (right).



Figure 12: Registration Error using SIFT matching

found, no output is produced. Approximately 20% of the frames tested resulted in no solution. Just as the accuracy of the approach is important, the decision on the placement of the compute needs consideration. Can the compute be done solely on the drone with the results of its position shipped back? Or is there a need to offload to a cloudlet?

One concern is whether the localization pipeline can be executed fast enough for use on the live drone feed. In particular, SIFT feature extraction is notoriously compute intensive, which impedes real-time use. In our prototype, we use a GPU implementation of the SIFT algorithm [2] that can process a 1280x720 image in 14-15 ms on our GPU. Approximate nearest neighbor matching takes about 75 ms. The complete pipeline, therefore, takes on average 97 ms for a 720p image when run on a cloudlet which has an eight-core, 3 GHz Intel[®] CoreTM i7-5960X processor with 32 GB of memory and an NVIDIA GeForce GTX 960 GPU. Thus, with the assistance of the ground-based cloudlet, we are able to accuately localize the drone over ten times per second, enabling real-time use at reasonable flight speeds.

In contrast, if we attempt to run the SIFT-based localization algorithm on drone hardware, performance drops by an order of magnitude. On an Intel[®] JouleTM 570x module, a 1.7 GHz quad-core embedded SoC comparable to processors used in high-end smartphones and more capable than typical drone platforms, our algorithms completes in 857 ms for a 720p image. The significant difference in drone-based and edge-based processing capability is corroborated in Wang et al. [20] in an object detection context.

Therefore, for this application, the ground-based edge computing infrastructure serves two improtant purposes. Not only do the cloudlets significantly speed up the localization by an order of magnitude, they also allow execution of Windows-based proprietary architectural software in VMs.

A video demo of our prototype can be found in https://youtu.be/odOXFBpMdG8.

8 Related Work

Image-based localization is a widely researched area in computer vision and robotics. Some of its applications include urban navigation, location recognition, and augmented reality on mobile devices. For localization, we have used a feature-based image retrieval approach taken by Irschara et al. [11] and Sattler et al. [18]. Other popular approaches include SLAM [12] and PTAM [13] where they simultaneously construct the 3D model of the environment and localize the camera pose within this map in real-time. Our work differs from them in that, while these works map to points in the physical world, we register the live view to the architectural model which is the golden standard in construction. Recent approaches to image-based localization that use deep-learning [3] are still in early stages and do not perform as well as feature based methods.

9 Discussion and Future Work

The construction industry has seen a more rapid adoption of drones than any other commercial sector. From site surveying to maintenance, drones are being utilized in all stages of a construction lifecycle. Most use-cases of the drone today are restricted to offline processing, where the data captured from the drone are uploaded to the cloud to create topographic maps, 3D mesh files or other formats for visualization. While offline analytics has its advantages, live-analytics using drones in construction is an area of topical interest. Live-analytics of drone feed enables the potential to actuate the drone in real time. Our work is a first step towards realizing that vision. A major impediment to drone-sourced live video analytics is the lack of processing power available on a lightweight and compact drone. In this work, we leverage edge computing and existing techniques in computer vision to realize live-registration of drone view onto virtual architectural drawings. Live-registration is a key step in realizing many applications such as live-mapping, monitoring, and real-time error detection. Decisions that used to take hours or days can be done in minutes using live video analytics.

Of course, our work represents just the initial forays into this application space. Many interesting near term and long term research questions are raised by our work. Our current prototype uses manually-registered reference images. It may be possible to automate this aspect, for example by using long-term mean GPS readings to accurately locate the reference images, or by using preciselyplaced fixed cameras. In addition our system uses homographic transforms, which make sense when dealing with planar surfaces; more sophisticated mappings may be needed to robustly handle partially constructed walls, or curved facades. Likewise, although SIFT features are robust to rotations in the camera plane, they are not effective with out of plane rotations. More robust features will be needed for proper matching when the drone is obliquely facing a wall. In addition, it may be possible to train a DNN to directly detect architecturally significant features that can be directly mapped to the engineering drawing without requiring intermediate reference images. Finally, the ultimate goal of inspection is to find any variance between the partially-constructed building and the plans; automating such anomaly detection will require further investigation.

10 Conclusion

This paper investigated the potential use of drones in the construction industry, and demonstrated a working prototype of a live inspection use case. Key to this is the use of live visual registration to the architectural plans made possible with support of ground-based cloudlet infrastructure. We believe the combination of semi-autonomous drones and live video analytics will be a source of significant productivity gains for the construction industry in the near future.

Acknowledgements

We thank our shepherd, Lin Zhong, and the anonymous reviewers for their guidance in improving the presentation of our work. We greatly appreciate the technical assistance of the following organizations: Hangar, Renaissance 3 Architects, PJ Dick, and Carnegie Mellon University Campus Design and Facility Development. This research was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0051 and by the National Science Foundation (NSF) under grant number CNS-1518865. Additional support was provided by Intel, Vodafone, Deutsche Telekom, Verizon, Crown Castle, NTT, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view(s) of their employers or the above-mentioned funding sources.

References

- [1] Filipe Barbosa, Jonathan Woetzel, Jan Mischke, Maria JoÃčo Ribeirinho, Mukund Sridhar, Matthew Parsons, Nick Bertram, and Stephanie Brown. 2017. *Reinventing Construction: A Route to Higher Productivity*. Technical Report. McKinsey Global Practices Institute.
- [2] Mårten Björkman, Niklas Bergström, and Danica Kragic. 2014. Detecting, segmenting and tracking unknown objects using multi-label MRF inference. *Computer Vision and Image Understanding* (2014).
- [3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. 2017. DSAC-differentiable RANSAC for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 3.

- [4] BusinessWire. 2018. United States \$ 1.77 Billion Construction Industry Analysis 2013-2018 with Forecasts to 2022. https: //www.businesswire.com/news/home/20180410006055/en/United-States-1. 77-Billion-Construction-Industry-Analysis. (Typo in title: should be \$1.77 Trillion).
- [5] DJI, Inc. 2017. Lightbridge 2 Professional Quality Live Streaming From the Sky. https://www.dji.com/lightbridge-2.
- [6] DroneDeploy. 2018. 2018 Commercial Drone Industry Trends. https://www. dronedeploy.com/resources/.
- [7] Mark Fiala. 2005. ARTag, a fiducial marker system using digital techniques. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 2. IEEE, 590–596.
- [8] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
- [9] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.
- [10] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2014. Towards Wearable Cognitive Assistance. In Proceedings of the Twelfth International Conference on Mobile Systems, Applications, and Services. Bretton Woods, NH.
- [11] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. 2009. From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on. IEEE, 2599–2606.
- [12] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. 2012. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31, 2 (2012), 216–235.
- [13] Georg Klein and David Murray. 2007. Parallel tracking and mapping for small AR workspaces. In Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 225–234.
- [14] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. 2007. Map-Based Precision Vehicle Localization in Urban Environments.. In *Robotics: Science and Systems*, Vol. 4. Citeseer, 1.
- [15] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision (2004).
- [16] Marius Muja and David G Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1) 2, 331-340 (2009), 2.
- [17] Alessandro Mulloni, Daniel Wagner, Istvan Barakonyi, and Dieter Schmalstieg. 2009. Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing* 2 (2009), 22–31.
- [18] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2011. Fast image-based localization using direct 2d-to-3d matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 667–674.
- [19] Richard Szeliski. 2010. Computer vision: algorithms and applications. Springer Science & Business Media.
- [20] Junjue Wang, Ziqiang Feng, Zhuo Chen, Shilpa George, Mihir Bala, Padmanabhan Pillai, Shao-Wen Yang, and Mahadev Satyanarayanan. 2018. Bandwidth-efficient Live Video Analytics for Drones via Edge Computing. In Proceedings of the Third IEEE/ACM Symposium on Edge Computing (SEC 2018). Bellevue, WA.