Visage: Enabling Timely Analytics for Drone Imagery

Sagar Jha[*]
Youjie Li[*]
Microsoft

Shadi Noghabi
Vaishnavi Ranganathan
Microsoft

Peeyush Kumar
Andrew Nelson
Microsoft

Michael Toelle
Sudipta Sinha
Microsoft

Ranveer Chandra
Anirudh Badam
Microsoft

ABSTRACT

Analytics with three-dimensional imagery from drones are driving the next generation of remote monitoring applications. Today, there is an unmet need in providing such analytics in an interactive manner, especially over weak Internet connections, to quickly diagnose and solve problems in the commercial industry space of monitoring assets using drones in remote parts of the world. Existing mechanisms either compromise on the quality of insights by not building 3D images and analyze individual 2D images in isolation, or spend tens of minutes building a 3D image before obtaining and uploading insights. We present Visage, a system that accelerates 3D image analytics by identifying smaller parts of the data that can actually benefit from 3D analytics and prioritizing building, and uploading the localized 3D images for those parts. To achieve this, Visage uses a graph to represent raw 2D images and their relative content overlap, and then identifies the various subgraphs using application knowledge that are good candidates for localized 3D image based insights. We evaluate Visage using data from multiple real deployments and show that it can reduce analytics-latency by up to four orders of magnitude.

CCS CONCEPTS

• **Computing methodologies** → **3D imaging**; • **Information systems** → **Geographic information systems**; **Data streaming**;

KEYWORDS

drones, geospatial, 3D imagery, analytics, real-time

ACM Reference Format:
Sagar Jha, Youjie Li, Shadi Noghabi, Vaishnavi Ranganathan, Peeyush Kumar, Andrew Nelson, Michael Toelle, Sudipta Sinha, Ranveer Chandra, and Anirudh Badam. 2021. Visage: Enabling Timely Analytics for Drone Imagery. In *The 27th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '21), October 25–29, 2021, New Orleans, LA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3447993.3483273

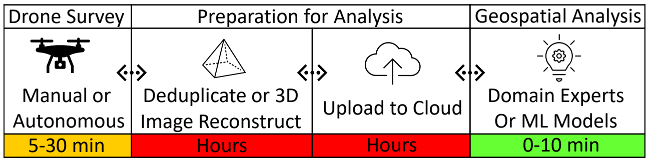


Figure 1: Time taken to create and transmit 3D images to the cloud is hindering interactive analytics with drone data.

1 INTRODUCTION

The next wave of remote monitoring is being enabled by on-demand three-dimensional (3D) imagery from manned or unmanned, aerial and terrestrial vehicles, drones, and robots [18, 107, 111]. Raw images captured in such settings are typically processed into a 3D image and uploaded as an aggregate to the cloud for analysis. Such analyses are increasingly used in commercial applications such as agriculture, mining, construction, energy, forestry, and disaster management to extract actionable insights [11, 21, 24, 35, 70].

Three-dimensional imagery is necessary to analyze structural, geometric, volumetric, and other holistic properties of real-world objects. For example, the structural integrity of a building, wind turbine, or a long stretch of railroad/pipeline requires 3D images to measure various geometric properties for calculating micro-stresses and changes [15, 97, 125]. However, there is an unmet need in running such analyses interactively. Currently, constructing a single 3D image of the entire surveyed region takes tens of minutes to hours for typical surveys [85, 126].

Data needed for 3D image construction is typically obtained as individual 2D image frames (or simply a frame) from cameras in motion that capture a very small part of the landscape. Hence, any information that is useful to gain a specific 3D insight is typically spread across multiple such frames. A simplistic approach of analyzing each frame individually, or in groups determined by temporal or spatial locality (GPS location of the camera), is often inadequate for obtaining accurate 3D insights. Temporal locality is not enough because frames needed for constructing a specific object of interest may be acquired over a period of time (such as images from different angles). Whereas, spatial locality is not sufficient because frames of interest may be captured from various distant locations.

Correlating across space and time quickly grows into a complex time-consuming task. Existing video-analytics techniques do not efficiently support cross-correlating information across 100-1000s of frames that are spatially and temporally distributed (typical for 3D imaging). State-of-the-art 3D solutions first consolidate the entire data by stitching the frames into a single large 3D image that virtually represents the landscape. This stitched image is then uploaded to

[*] Sagar Jha and Youjie Li contributed equally to the research as interns at Microsoft.

ACM MobiCom '21, October 25–29, 2021, New Orleans, LA, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8342-4/21/10…$15.00
https://doi.org/10.1145/3447993.3483273

cloud applications for generating insights. This process, however, takes several minutes and hinders interactive applications [85, 126].

The problem of timely 3D image analysis is most prominent in manual/automatic drone, robotic, and vehicle-based surveys, of the remote parts of the world, where edge deployments face both compute and network bottlenecks. We will focus the rest of this paper on such settings that we refer to collectively as drone surveys. Our goal is to help analyze 3D imagery from drone surveys interactively over weak network connections to cloud services.

Interactive drone data analytics is necessary for various real-world application scenarios: (1) Informed exploration: Drone operators need to analyze the data collected to quickly plan the next survey so that they can make the best use of time while in the field (drone-as-a-service operators need to immediately correct for erroneous data acquisitions, scan an anomaly at a slower speed or closer distance for clarity, to capture a different part of the spectrum with a specialized camera, etc.) [36, 48, 83], (2) Proactive maintenance: Help a maintenance engineer quickly operate the drone and act on insights immediately rather than having them revisit the remote location multiple times (e.g. monitoring/maintaining pipelines, power-lines, or solar panels in remote areas) [15, 38], (3) Disaster mitigation: To rapidly analyze drone images and provide timely insights to agencies that manage ongoing environmental disasters (e.g. wildfires and flooding) [59]. *However, two significant challenges, of limited compute and network capacity, hinder interactive drone data analytics, as illustrated in Figure 1.*

First, constructing accurate 3D images takes time as several pairs of non-local frames need to be inspected for content overlap, resulting in poor scalability. Stitching thousands of frames typically takes several minutes to complete [85, 126]. This seemingly inherent processing delay rules out any possibility of obtaining insights promptly for existing solutions. Powerful edge systems can provide real-time 3D aerial imagery [98] as well as analytics with 3D imagery [40]. However, providing such analytics interactively with weakly provisioned edge systems in a timely manner remains unsolved. Such powerful edge servers are difficult and expensive to carry or deploy in remote regions and ad-hoc situations where drones are often used.

Second, transferring either frames or the stitched 3D image or objects to the cloud is a bottleneck. While 3D images are typically 4-10x smaller in size compared to total frames' size, they are still often several GBs in size, normalized per twenty minutes of surveying by a typical drone[22]. Despite the promise of 5G networks in urban areas, many remote areas where drones are typically used in the commercial industry do not have broadband Internet, i.e. they have below 3Mbps *upload* speeds, while download speeds may be higher [19]. It is typical for the drone data generation rate to outpace the internet upload speeds by over an order of magnitude [27, 41]. The problem is more dire when using LIDARs, multispectral, and/or hyperspectral cameras that generate data of over 1Gbps [48, 83].

Furthermore, many applications require other information in the cloud for analytics including historic data, data from other edge locations, satellite imagery, weather station data, and sensors/camera/data streams which are hard to replicate at the edge [94, 108]. Therefore, a system is needed that provides network-based optimization as well as compute acceleration for interactive drone analytics.

In this paper, we present a new 3D image data transfer mechanism called Visage, that runs at the edge and enables interactive analytics on drone data. At a high level, Visage quickly converts raw frames coming in from drones to a stream of small and localized 3D images. Instead of constructing the full 3D image in one go, Visage incrementally constructs only portions that can help the application gain necessary insights quickly. Furthermore, it uses domain knowledge to prioritize compute and network resources towards the smaller 3D images, based on their predicted importance for the application.

Performing these operations online as well as in parallel is paramount for interactive analytics. If performed sequentially, the upload has to wait for the entire survey and the stitching to finish. However, performing the operations simultaneously is complicated as the drone is still surveying and adding more frames.

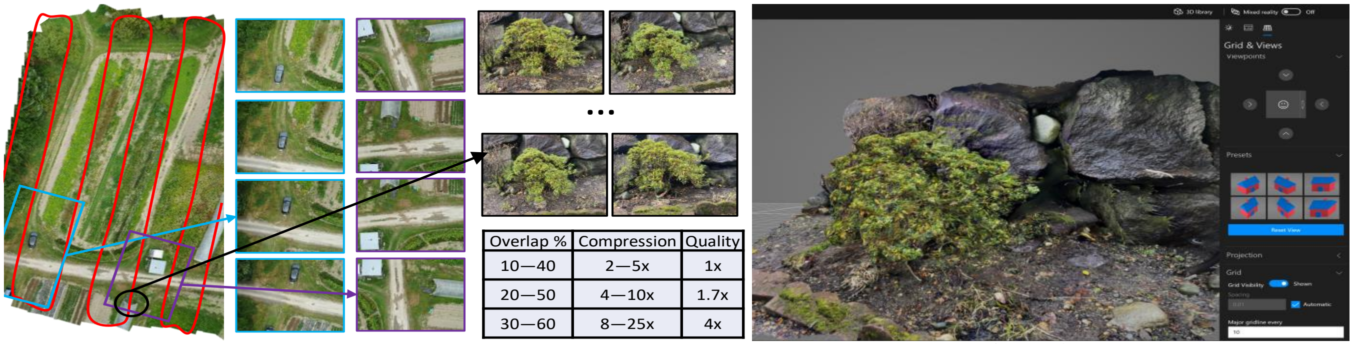
We propose using a graph data-structure as the right abstraction for interactive drone processing. The key insight behind choosing a graph is that drone frames are related to each other from a content overlap perspective, rather than from a purely temporal or spatial perspective. It is the content of the frames that allows the stitching of frames into a coherent 3D image. When considering more holistic content-oriented aspects to find related frames, a graph data structure emerges naturally where each frame has content that overlaps with many other frames that are neither spatially nor temporally local. Existing techniques condense such a graph into a single 3D image before performing analysis [32, 84] while we propose that the graph be processed by prioritizing focus on the parts of the graph that are more important for the interactive application. We achieve this by making the following contributions:

- Application-focused 3D image processing: To speed up the processing of frames, Visage not only leverages field-of-view overlap (modeled as a graph of frames) as the locality metric for converting raw-frames to 3D images but also focuses such efforts only on the *relevant* objects/features. Visage uses an application/domain knowledge-based model to identify such relevant objects/features and uses only the raw-frames containing information about them to prioritize building the associated local 3D image.
- Informed stream processing: The localized 3D images are built incrementally and online by carefully selecting from newly added raw-frames only those that are related by content to the 3D image being built. The application/domain-model is additionally used to identify the correct subset of the incoming frames to be streamed from the drone.
- Progressive transfer: Lastly, the 3D images are sent to the cloud by progressively enhancing their quality to further reduce the latency to insights. Furthermore, the important objects/features within the localized 3D images are enhanced in quality faster than the rest of the image to further improve the bang-for-buck on the network transfers.

Experimental results from simulated drone images as well as data from multiple deployments show that in Visage applications start receiving important information and insights within minutes rather than hours. Moreover, we improve latency by up to four orders of magnitude compared to existing methods, all while using only a commodity laptop as an edge and with a modest 3Mbps uplink capacity.

2 MOTIVATION & BACKGROUND

High computational overhead of 3D image stitching: Drone data is collected with high levels of overlap between images. The



Overlap %	Compression	Quality
10—40	2—5x	1x
20—50	4—10x	1.7x
30—60	8—25x	4x

Figure 2: Drone analytics are complex: Left most figure shows a typical aerial drone path (in red) with overlapping frames. Overlap is caused not only by adjacent frames (blue boxes) but also by frames captured from multiple angles and distances (purple boxes). Such overlap is needed for constructing high-quality 3D images which are used for holistic anomaly detection such as the plants with no fruit (black boxes and its 3D image on the right). To save time, Visage identifies and builds only smaller 3D images that contain anomalies as opposed to stitching a large 3D image of the entire surveyed region.

overlap helps to align the individual 2D images into a larger 3D image. Drones typically capture images such that there is a 40–90% overlap between adjacent frames [28, 33, 87] (as shown in Figure 2) to help obtain high quality 3D images. The quality metric shown in Figure 2 is that of coverage in terms of 3D pixels per square inch. Computer vision techniques, in the form of a "stitching" software, are then used to fit the 2D images into a virtual 3D space called a point cloud using techniques such as structure from motion [82], binocular/trinocular matching [7, 135], and feature map constraint satisfaction [23]. These are computationally intensive operations that are super linear in time complexity as they compare and match image feature content across several pairs of frames for fitting the 2D images successfully into the 3D point cloud (we describe this process in more detail in the next section).

The overlapping nature of the imagery results in massive amount of raw data that forces the construction of the 3D image to happen at the edge especially so for weak Internet connections that we are targeting. As shown in Figure 2, raw imagery can be up to 25x more than the size of the corresponding 3D image. This is an incredible acceleration opportunity to exploit.

Analyzing 3D images: 3D images are projected on to 2D surfaces to create "*unique-frames*" as they contain a unique pixel per area that they cover. These unique frames are then transferred to the cloud where they are analyzed in detail. For example, in an orchard, each unique frame containing a tree could be projected onto a 2D "cylindrical" image going around the tree. This cylindrical projection provides holistic (e.g. fruit count), volumetric (e.g. tree trunk and foliage volume), geometric (e.g. total branch length), and other structural analytics.

However, not all portions of the landscape are required to be analyzed in such detail and therefore the 3D images and projections must be considered only where necessary. Most "insights" derived from real-world scenarios tend to be sparse. For example, in 100s of acres large orchard, the insight of interest might be to identify a handful of under-performing trees (e.g. low fruit count). Therefore, the drone operator may upload only those images and seek advice from an expert interactively as to what more images (e.g. with a different camera type such as hyperspectral/multispectral or with higher resolution) of the specific trees they want while the operator is still in the orchard. In the absence of an interactive option, the upload and analysis would require the operator to come back another day for further data acquisition, thus wasting time and resources.

Existing drone systems combine all the frames into a single 3D image as they typically do not target wide area interactive applications such as the ones we want to focus on. In this work, we set out to only build and project the portions of the 3D image that can provide insights or at least prioritize and accelerate those portions. Thus, in the case of the orchard for example, we use only the frames of those trees of interest (by the help of traditional image inference techniques) and then prioritize the building of their 3D models alone instead of a 3D model of the entire orchard.

However, the information required to understand which projections to choose, which pixels to choose in each projection, and what frames to use to get those projections to an acceptable quality etc., are all application related choices and therefore, we must extract such knowledge to solve this problem efficiently. Visage leverages this knowledge available only at the application layer by incorporating them as domain-specific models used when building the 3D image.

Large data transfer to cloud: 3D stitching to create unique frames is typically done at an edge location near the area of interest, simply because the amount of raw data generated (greater than 32Mbps per camera is typical) is well beyond the network capacity available in these remote locations.

However, the analysis on the data is commonly performed at a remote location such as the cloud. There are several reasons for this including: (1) the human/machine consumer for the data is not present at the edge, (2) the area being studied is geographically large/spread-out such that no single edge can realistically cover it, and multiple drone pilots are simultaneously operating and aggregating the data in the cloud, (3) an edge that is powerful enough for the applications cannot be deployed in these remote locations, and (4) applications in this space often combine drone data with other datasets present in the cloud, e.g. older imagery, satellite imagery, weather data, and other application/domain data. Therefore, we see that this is primarily an edge-assisted data-transfer problem, rather than a pure edge-compute one.

Even with improved stitching, transferring the required unique-frames to applications running remotely over weak links in a timely manner is still challenging. Unique-frames (large panoramic stitched images) can each be 10s of MBs to a few GBs[22] in size and there can be 10s of unique-frames per survey. While existing image or video compressors can reduce the volume of the unique-frames to be transmitted, the achievable compression ratio to quality tradeoff is still not acceptable as these projections often contain mostly unqiue information not present in other unique-frames.

We propose exploiting application knowledge and looking into the semantics of each segment/feature/object in unique-frames and then sending those data fragments that are more important to the application at a higher priority/quality, while delaying the sending of or reducing the quality of or even artificially "synthesizing" less important fragments (such as background sky, soil, or lawn in unique frame of a tree in an orchard) to improve the bang-for-buck on the network.

3 VISAGE OVERVIEW

In this section, we present the goals of the design and the architecture of Visage, motivated by our customers' requirements.

Goal-1: Reducing Computation. Identifying which 3D images to prioritize to meet interactive needs is hard. Using an inference model on every frame to help identify the parts of the surveyed region that are of interest to the application can be expensive especially given the fact that frames have a lot of overlap with other frames. A balanced approach is needed to broadly look for interesting aspects without duplicating the inference efforts by avoiding frames that overlap excessively with frames that have already been searched. As soon as the regions of interest are found, inference must focus on frames that have overlapping content to help quickly identify the set of frames needed to construct the 3D image of the interesting aspect.

Goal-2: Efficient deduplication. Ideally, drone data streaming should aim to send no more than one pixel per representative unit of area in the region surveyed. This reduces the network and compute requirements for analytics. Building a 3D image from a collection of overlapping 2D images benefits not only holistic analytics but also eliminates redundancies from data. However, 3D image construction is a slow process and the downstream latency benefits must be balanced with the upstream processing latency. Hence, the 3D construction must be prioritized on parts that are important to the application. That way, the network usage can be optimized with more important bits taking preference over others.

Goal-3: Dynamically prioritized streaming. For applications running remotely over a weak link, Visage must start streaming the most important pixels of the most important unique-frames (e.g. a unique-frame of a cracked railroad segment is more important than the unique frame of a segment that is bent), to improve the bang-for-buck from the weak network. However, new raw frames can change the relative importance of the data to be uploaded as more important aspects may be discovered. Therefore, Visage needs to keeps track of unique frames (and various segments within them) and upload them in the order of their relative importance to applications.

3.1 System Architecture

Visage is designed for interactive sessions between a surveyor and an operator. The operator is a person or an autonomous agent present at the edge location where the drone survey needs to take place. The surveyor is a person or an autonomous agent which is remotely interacting with the operator via a cloud agent that analyzes 3D images. The surveyor provides the operator with a large region of interest to be surveyed and an image segmentation model, such as MobileNet V2[103], with segment types that the surveyor considers important (different importance level can be set for each segment type). Figure 3 shows this as the domain-spec inference model.

The operator then feeds the region of interest into a drone specific navigation planning software [29, 86] that helps chart a path that the drone follows for the fastest and most battery/energy optimal way to survey the region. The raw frames from the drone are then streamed to Visage that runs on a laptop or workstation(s) carried to the field by the operator represented as the "Edge" box in Figure 3. The streaming from the drone is performed on an ISM band frequency using a custom protocol between the drone and a joystick that is attached to the edge. The edge is implemented as a collection of Docker [30] containers in a Kubernetes[60] environment, where it can easily run on a single laptop or a cluster of ruggedized servers that can be mounted in a vehicle with the operator.

The raw frames are first absorbed into the graph data store. The nodes of this graph represent images and the edge, and their weights represent how much potentially overlapping content the images may have with respect to each other based on the camera viewing angle (described in Section 4).

The graph is then searched by the localized stitching agent using a random walk for frames that contain aspects that are important for the application using the domain-spec model. As soon as such aspects are found, the stitching agent aggressively searches connected frames to start building a localized 3D image and the corresponding unique frame as shown by the stitching component in Figure 3.

The progressive encoding agent then starts working on the unique frames that are generated to encode them into a number of layers and segments with varying priorities. The priority data uploader uses these encoded segments and starts transmitting them to the cloud in the descending order of priority. In mobile and remote settings, the edge is connected to the cloud in our setting typically using a 3/4/5G or LTE connection. In some settings, the edge is stationed permanently at a remote outpost (where the drones are also often stored) connected via broadband.

The cloud component receives the segments and progressively enhances the unique-frames by decoding them and feeds them to the surveyor. The surveyor analyzes the unique-frames as the segments are received and sends insights to the surveyor who then decides which parts of the region need attention or more detailed surveying and instructs the operator accordingly.

4 VISAGE DESIGN & IMPLEMENTATION

4.1 Frame-Graph Data Store

As data is ingested into Visage, either streamed live (or deferred) from the drones, it is stored in the form of a weighted graph data structure with frames as vertices. Two frames are connected via an edge if they have any potential overlap and the weight of the edge represents the amount of potential overlap. This graph is stored as a Python-based adjacency list and is used in next components to build 3D images and unique-frames.

Visage uses the camera metadata (location, direction, focus, field of view etc.) to help reduce the search space when looking for frames

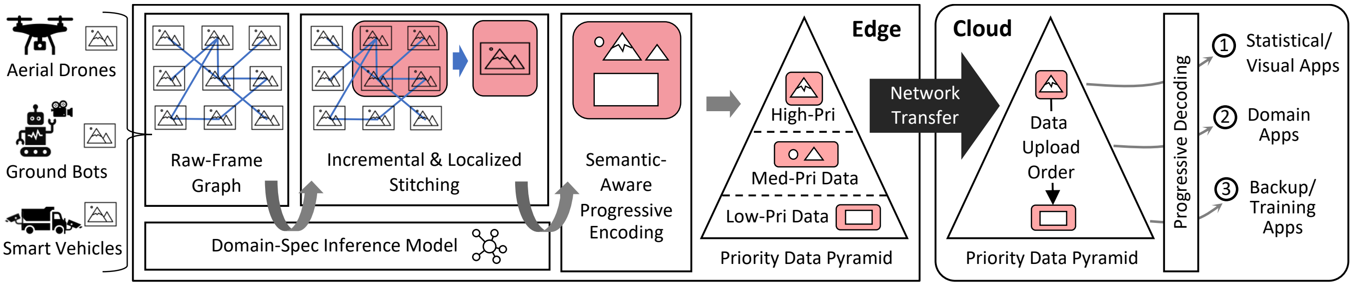


Figure 3: Visage's Architecture: Visage ingests live frames from drones. A domain inference model is used for continually identifying which new frames to stitch into 3D images, which the unique-frames are generated from. Using the domain models, semantic-aware progressive encoder, encodes segments in a fine-grained priority-aware manner. The uploader progressively transmits data to cloud following the order of a priority pyramid. Finally, the progressive decoder, decodes the data at the cloud.

that may potentially overlap with each other without even looking at the pixel data, as opposed to using an exhaustive and thus expensive computer vision based search of similar features to identify overlaps. Each frame is tagged by the drone with its location (GPS) as well as the direction (roll, pitch, yaw, and focus) in which the camera was pointing at the time of capturing the frame. This information is used to a build a three-dimensional cone with a cutoff-distance, beyond which the focus is considered moot, to represent the field of view of the frame. For each newly added frame, its cone's intersection with all other existing cones in a radial distance from the location of the camera is calculated. An edge is added for every cone intersection and the weight of the edge [0-1] is set in proportion to the relative size of the intersection.

We calculate these intersections using a PostGIS [89] geometric database instance. Each new cone is first added to the database, then a query is posed to search for all the intersecting cones within a radius bound which is equal to the cutoff-distance. Such 3D geometric queries, while sounding complex, are a well optimized query type in vector calculus based databases that are typically answered in under a millisecond even for 1,000 cones [91, 92] which is significantly more cones than what we find within typical cutoff-distances used in our settings.

4.2 Localized Stitching

Visage uses a novel local stitching approach where, instead of building a global 3D image with all the frames, it constructs a set of local 3D images of areas of interest to the application. Visage uses domain-knowledge based segmentation model (and other data sources described further) to identify sub-regions (henceforth referred to as segments) that are important for the application and stitches the corresponding frames into a localized 3D image. The challenge lies in identifying the areas of interest from a large set of frames without running computationally heavy inference models on every frame. Towards this, Visage uses a two phase approach. First, it discovers a frame from an unexplored area of interest using a random walk approach on an area-tracking data structure. Once such a frame is found, it uses a greedy approach on the graph data store to find all frames that overlap with the segment of interest and builds the 3D image with those frames. A detailed description each of these components follows.

Identifying segments of interest: Typically, Visage uses domain-specific segmentation and labeling models to identify important segments to prioritize. Such a domain-model can be supplied by the application developer. Alternatively, they can choose one from the model library that we have built for various scenarios in agriculture, energy, and other commercial industry verticals and simply adjust the importance of each segment type to their desire (e.g., the foliage is more important than the trunk). Additionally, Visage also accepts direct input from the operator or the surveyor as to which areas to prioritize over others (e.g. specific aisles of trees or specific trees in an orchard). Lastly, historic information or other imagery (such as satellite imagery that is significantly low in resolution and can also be out of date) can be used to supplement the domain-model.

Discovering unexplored segments of interest: The goal of this phase is to quickly and efficiently discover new segments that have not been stitched or are being stitched into a localized 3D image. Visage tracks the explored areas, i.e., areas for which the 3D model has been constructed, in an "area-tracking" data structure. For this, we use another instance of PostGIS [90].

In order to avoid searching all frames for new segments of interest, Visage uses a random walk to search for newly obtained frames for important segments while relying on the area-tracking data store to record old frames that have already been explored with the domain-model. Visage picks frames with the less amount (a configurable value) of intersection with the areas that are marked as already explored in the area-tracking database as new frames can overlap with old frames. This search completes when the surveyed region has been fully covered in the area-tracking database.

Another optimization we use when detecting important segments is to run the domain models on a lower resolution version of the frames. However, this can come at the cost of reduced accuracy. We explore this trade-off using profiling (shown in more detail in Section 5) and show that for typical high-resolution drone frames, segmentation is often just as accurate when the frames are reduced in resolution by up to 36%.

Creating localized 3D images: Visage uses a set of well known 3D mapping and panoramic image stitching techniques for creating 3D imagery with several sequential stages. The first stage involves laying out frames on a virtual canvass based on their geolocation and then extracting precise pixel correspondences in each overlapping pair of frames. We extract local visual features, specifically, scale invariant key-points such as DoG [69] and Daisy feature descriptors [124]. However, other alternatives (SIFT [69], Superpoint [26]) can be used as well. The key-points are compared by computing pairwise Euclidean distances between feature descriptor vectors at the matching stage. Feature matching is accomplished by solving approximate nearest neighbors in the feature descriptor space and the pixel matches are further refined using robust estimation techniques that leverages epipolar geometry constraints [46].

Given thousands of pixel correspondences over many pairs of images, we use an approach called structure-from-motion (SfM) to simultaneously reconstruct the 3D positions of all the triangulated 3D points and the cameras positions associated with each of the input images. At its core, SfM computation involves solving a very large nonlinear least squares optimization problem, which is also referred to as bundle adjustment [118]. Then, given the optimized camera position, orientations and the estimated 3D points, we estimate a coarse ground plane by robustly fitting a 3D plane to the reconstructed 3D points. The input images are then warped onto the ground plane by transforming each input frame via a 2D homography transformation that can be computed from the estimated camera pose parameters. Finally, all the overlapping warped frames obtained from the previous step are seamlessly merged by using image stitching algorithms that aim to compute good seams and adjust the pixel colors between overlapping image areas such that the seams are visually in-distinctive and gives the impression of a single seamless image.

The stitching of a localized 3D image using the above process can be started as soon as a sufficient number of frames have been found to cover the segment of interest in consideration. Once a frame from with a segment of interest is discovered, Visage uses the graph data store (Section 4.1) to find the most relevant other frames and constructs the 3D image. Each stitching job independently decides which frames to consider for the stitching. For the selection, those frames are considered that are most likely to contain the pixels corresponding to the important segment that triggered the current localized 3D image in the first place.

To identify such frames, using the graph data store, the frames with the most overlap (ones with the highest edge weights) in proximity to the geographic location of detected important segments are traversed. The stitching job, exhaustively runs the domain-model on each such frame (not a random walk) to decide whether to include that frame or not, as it knows that the important segment is highly likely to be found here. However, the stitching itself starts only when enough number of such frames from multiple directions have been identified in order to create a high quality 3D image of the important segment. In our implementation, this is simply a timeout of an alarm that is reset each time a new frame is added live by the drone that overlaps with any of the frames that are considered already a part of the stitching job. This is enforced at the frame ingestion time.

4.3 Semantic encoder & prioritized transfer

Localized 3D images are then projected onto 2D surfaces such as a plane (an aisle of trees) or a cylinder (around a tree) so that analysis with traditional computer vision pipelines becomes easier. Even with deduplication through 3D image reconstruction, the resulting unique-frame (or projections) sizes are huge, i.e., they can contain 10-100s million pixels with sizes of 10–100s MBs [85]. However, not all of the data in the frame is useful for the application. Most of the frame is often irrelevant and unimportant information, e.g., sky, lawn, background of the orchard for studying fruit trees.

Also, drones are configured to capture pessimistically high resolution images (4K is common with 32bit pixel depth) as a way to ensure that the acquired data can serve future needs of applications that may demand higher resolution. However, 1080P or lower resolutions are often enough for analysis in today's applications. Moreover, among the segment types output by the domain-model, not all are equally important. For example, the fruit on a tree could be more important than the tree itself for an orchard. The encoder's goal is is to tie the resolution and quality of each segment of unique-frames to application requirements, which we call "semantic-aware encoding". By doing so, we are able to compress further beyond any given image encoder. For example, we can use high quality JPEG (or any given codec) encoding for the segments containing fruit while a low quality JPEG encoding can be used for other segments. This enables a segment-differentiated encoding that leads to orthogonal savings. We take this to the extreme by having a progressively encoded option.

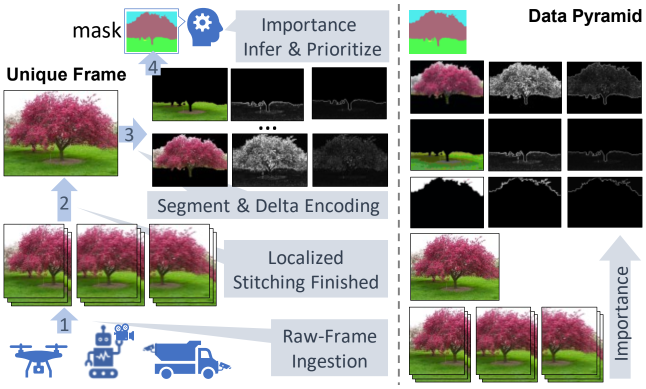


Figure 4: An example showing how a unique-frame is produced, encoded, segmented, and prioritized differentially to reduce analytics latency.

Figure 4 illustrates the process of semantic-aware progressive encoding. Once a unique-frame is generated, it is decomposed by the domain-model into multiple small segments with different *semantics* (e.g., tree, grass, sky). Then each segment is compressed with base-delta encoders (JPEG-XR [34] used in our implementation), i.e., a base file containing the minimal and most important data, followed by multiple delta files each enhancing the base.

By breaking down unique-frames into such fine-granularity data, Visage is able to aggressively shrink the transmission volume and the network latency to start analytics. It first only sends the minimal, yet the most important files (typically only 0.01% of the unique-frame size) which identify the segment types, boundaries and locations (segmentation masks). Then, it progressively sends the rest of the data and gradually enhances fidelity of data in a way that prioritizes the requirements of the application. Thus, by using a combination of progressive image encoding and prioritized transfer of segments, Visage is able to improve the perceived latency for remote applications.

4.4 Prioritized Data Transfer Engine

As shown in Figure 4, data transfer is done via a priority queue schema. This engine retains a priority index structure maintaining an ordered queue of items to be transferred (we use a PostgreSQL DB instance to track the priority of each file). Each item in this priority index is a pointer to a file to be transferred, either segmentation mask, base file, delta file, unique frame, or frame. The mapping of files to priority indexes is a configurable component in the data transfer

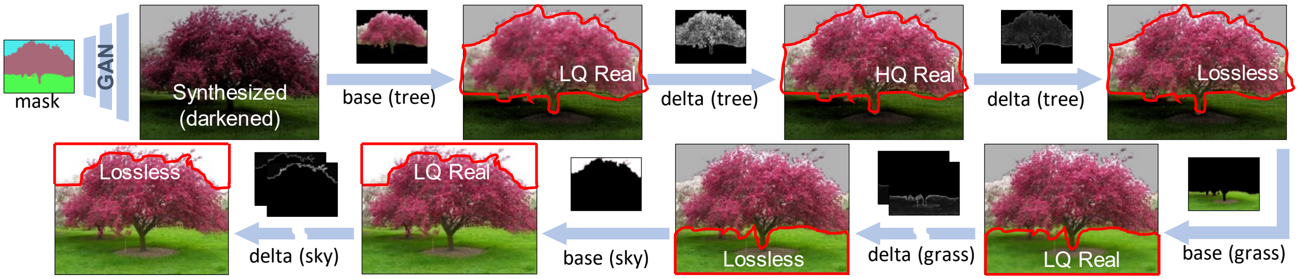


Figure 5: An example of how a unique-frame is progressively decoded in the cloud so that tasks can run as soon as their expected segments and quality have reached. Artificially "synthesized" segments are used when only the mask is received and they are are intentionally darkened in this illustration to make them easy to spot. The "synthesized" segments are, however, hard to spot for humans and certain ML models thereby offering instant visualization and reducing latency for applications.

engine. In our implementation, for each newly encoded segment, the engine uses the associated segment type to dynamically decide the priority of the base file, and uses this priority to relatively decide the priority of the delta files. For example, if the mask determines the priority of the "tree" segment to be '5', the delta files are given priority '4.5'. Raw data is given a low priority.

The data transfer engine uses a remote/cloud mounted folder shared with the application in the cloud. The engine (implemented in C++) runs as a container that monitors the folders being created by stitching containers, understands their importance level (using the priority index DB), and copies the data to the remote mounted folder in that order. The transfer engine also allows preemption, i.e., it can stop mid-file when needed and switch over to more important ones. In addition, it handles failures and reboots by tracking its own progress in the priority index DB. While priority-based data synchronization solutions exist in other settings (Section 6), this is a new synchronization mechanism for edge-cloud systems that supports priorities and hierarchies when sending *files* to a cloud storage service.

4.5 Decoder and artificial data generation

Unique-frames often contain a significant number/amount of segments not useful to remote applications and surveyors, e.g., the background elements when sending panoramic images of small objects. We use this as an opportunity to further optimize the latency of insights to the surveyor.

The masks of segments of unique-frames arrive first in the cloud, followed by the actual data of the unique-frames (base and delta files). These masks are typically 10,000× smaller in size compared to the size of unique-frames generated at the edge. By just using these masks, some statistics can be obtained in the cloud right away. The presence of certain segment types indicated by the importance level in masks can help applications perform certain tasks such as counting, sizing, and anomaly detection.

In parallel, Visage uses the mask to "synthesize" pixels in blank unimportant segments, therefore, offering instant visualization and reducing latency to certain application tasks that accept "synthesized" data (statistical ones) as described in the next section.

As shown in Figure 5, using the mask and a trained GAN, Visage quickly generates a synthetic frame. After a short time, the base file of the most important segment ("tree" in our case) shows up. Then, Visage offers a hybrid view of the images with real pixels for important segments and faked data for less important segments. This is especially useful for scenarios where the surveyor is a human who interactively works with the operator for informed exploration. It is also useful for DNNs that are overfit to high-quality training data and hence require the low-importance segments to be present.

As time passes, the refinement delta bits arrive and the rest of the data (for less important segments) follows later when there is more bandwidth available. In all cases, applications register for notifications from the filesystem, which automatically triggers functions and pipelines for each refinement stage (as a way of tying image quality with the task at hand).

4.6 Domain Inference & GAN Models

Both the inference model at edge and GAN model at cloud are developed specific for each domain, instead of each application, due to the common features and characteristics shared in the same domain.

For the inference model, we currently use MobileNetV2 [103] because of its low footprint and small size (3MB). Applications can also specify a model of their choice at deployment time. In addition to choosing or specifying a model, applications must indicate how importance is calculated for each segment. To facilitate complex applications with many segment types, we implemented the following automated approach to extract importance.

Developers can instead provide a black box (often containing an upstream application) that helps us transparently understand the importance order of segments. For each segment that Visage generates, we distort the pixels corresponding to that segment (*e.g.*, by adding noise) and feed the degraded image into the application black box. We expect the black box to output a certain score when fed with an image. By varying the degree of distortion on each segment individually and observing the delta of the output score, Visage can profile the relative importance of any segment. Visage then uses the importance metric to create a mask for a image. In the absence of such quantifiable insights, we work with a domain expert as well as the customer to set the importance values for each segment.

To train an inference model, either public data on internet or historical data collected can be used as the training sets. When labeled data is easily available, we paired the label masks with their true images. For settings without labeled data (the more usual case), we use virtual reality environments to generate labeled images. For example, raw drone frames are obtained from the Airsim [3] platform which is used for stress testing the system as well as generating synthetic/virtual labeled data for modelling. We hire video game content design companies[93] to develop these environments in Airsim. Then we operate virtual drones in these environments and collect frames which are labeled automatically by the virtual reality engine. Such generated image sets are for training not only domain-specific inference models but also for the GANs in Visage.

For GAN models, the choice is abundant thanks to the fast paced community of computer vision. In general, more complex GANs often offer more realistic synthesis, which might be favored considering that the cloud is no lack of powerful resources. In Visage, however, we adopt a simple and classic one "pix2pix" [55], which is able to translate a label mask to an image at a decent quality level without compromising the visualization nor the applications (see Figure 5, 9 and 10). The model size is only 200MB and can be trained with a single RTX GPU (11GB) [80]. During image generation, its throughput also shadows the network speedup.

5 EVALUATION

We present experimental results that validate our claims about the benefits of using Visage. Our goals are to quantify the overheads and benefits of individual components, and to also identify the end-to-end latency benefits of Visage compared to other techniques.

5.1 Experimental setup

We use a mix of real-world and synthetic datasets as the stream of data to process. In either case, we emulate a *virtual drone* by replaying the captured stream of an actual/synthetic drone from the dataset.

Real-world datasets: We have deployed Visage in 8 different real-world customer trials across various commercial industry sectors including agriculture, energy, and disaster management. For the experiment, we focus on 3 main scenarios: 1) an information exploration use-case surveying an agriculture farm, 2) a proactive maintenance use-case monitoring oil pipelines using a manned aircraft, and 3) a disaster study/mitigation use-case responding to hurricane damage with aerial drone imagery. We have studied a total 5TB of frames from more than 500 drone surveys among the three scenarios. Majority of the surveys had only one camera per drone, 15 of them had two cameras per drone, 7 of them had four cameras per drone, and lastly 1 of them had six cameras on the drone.

Synthetic datasets: In order to stress-test the system and also to obtain large quantity of images with ground-truth labels from the virtual reality environment (Section 4.6). The simulator is setup to continuously stream raw frames by operating a virtual aerial drone with a single downward facing camera in the environment generated at a rate of 20 frames per second each with a resolution of 1024x768 pixels.

Testbed: We use a Xeon [54] Desktop with NVIDIA GTX GPU [79] to help replay the real-world datasets as well as run the Airsim drone simulator, a Surface Book 2 [76] as the edge, and Azure Blob Storage [75] as the remote destination. We use a 10Gbps link between the drone emulator desktop and the edge laptop, essentially ensuring no bottlenecks on this link. We emulate an upload link between the edge and the cloud with different speeds between 3 and 15 Mbps, typical of remote areas [19] where drones are operated in the commercial industry. The network bandwidth is throttled by setting the maximum upload bandwidth for the file share mounted on the edge. For more realistic network setting, towards the end of this section, we also summarize the results from a real world deployment by analyzing the logs.

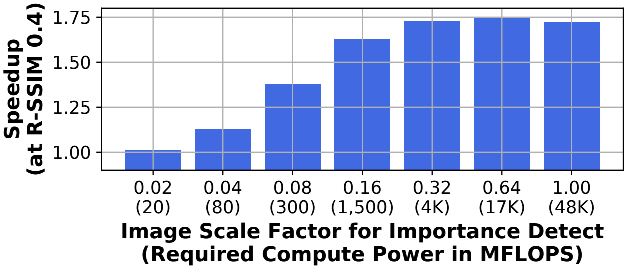


Figure 6: Edge resource proportionality of Visage. In general, the larger the scale factor is, the more accurate the importance detection will be. This means that less redundant data will be sent, enabling applications to achieve a target score faster. Visage can increase this speedup with more compute power at the edge to process higher resolution images.

Metrics: In addition to *visual evaluation* by domain experts and standard latency or resource utilization metrics, we also adopt the following metrics to compare Visage with other approaches.

- *Region-weighted Peak Signal to Noise Ratio (R-PSNR)* [123]: A classic metric based on PSNR that assesses the fidelity of reconstructed/decompressed images by measuring per-pixel distortion between the ground-true image and the reconstructed one. A higher value indicates a higher fidelity. Typical values in wireless transmission of compressed images are 20-25dB [63, 113]. We adopt a "region-weighted" version [123] of PSNR to focus more on regions that are more important to the application, instead of treating all pixels equally. For example, in the agriculture use-case, regions with plants are weighted higher. Weights are determined by the application and are consistent with the sensitivity scoring used in [127].
- *Region-weighted Structural Similarity Index (R-SSIM)* [123, 138]: Another classic metric for assessing the fidelity of reconstructed/decompressed images by measuring the structural similarity between the ground truth and the reconstruction. *R-SSIM* is calculated in a similar manner as *R-PSNR*.
- *Object Coverage Ratio* is a state-of-the-art metric to assess processed images by employing a deep-learning model as the evaluator [55, 81, 96, 102, 132]. Unlike classical PSNR/SSIM, the deep-learning based metric captures the features and semantics that are critical to applications. A standard semantic segmentation DNN (dilated ResNet) [136, 137] is adopted to obtain the pixel-count ratio of chosen objects for a batch of images, *e.g.*, *healthy* forest pixel coverage ratio. The better the image transmission approach, the better is the ratio, *i.e.*, the transmitted image is comparatively closer to the original image at any given time.
- *Object Count* is a similar metric that also employs a deep-learning based object detector [72] to count the number of chosen objects in a batch of images [122], e.g., the number of trees in a unique-frame.

5.2 Individual Component Analysis

Important Segment Detection: Visage is able to adjust the benefits it delivers to applications in proportion to the important segments detected. However, this depends on being able to use a GPU at the edge for detecting important segments in the data. A central premise of Visage is that powerful edge systems are not easily available in remote regions and therefore, we perform the importance detection at lower resolution - scaling down images based on a *scale factor* between 0 and 1. However, by scaling down, the chances of a segment *falsely* being marked as important may change and therefore the resulting unique frame might not be structurally the same as the one created when using high-resolution inference (*e.g.*, as determined by *R-SSIM*). For our workloads, we have found that there is a significant scaling opportunity as segmentation models are often optimized for significantly lower resolutions compared to our datasets.

Figure 6 shows the speedup vs. scale factor to reach a target structural similarity score of 0.4. In particular, we find that most of the flexibility is in scaling the images between $0.02^2 - 0.64^2$. Furthermore, we see that for a given *R-SSIM*, higher scale-factors have diminishing returns. This provides us with a way to profile and adjust the parameters in our system when considering the frame resolution to go with for a given drone data generation rate and GPU throughput.

As an example, Figure 6 shows that beyond a scale-factor of 0.32, a higher resolution importance-detection may not be needed for most applications that can work with an R-SSIM of 0.4 (something that was enough for our analytics). This helps us attain most of our goals with just one laptop-class GPU (Surface Book 2) designed for simple graphics when consecutive frames overlap by 40%. For more cameras and more overlap (therefore higher frame rate) needed for increasingly complex applications, we recommend that drone operators have a server-class GPU that is designed for DNNs.

Localized Stitching: We perform a stitching microbenchmark by running the stitching on the entire real-world data set for both global (de-facto approach) and local (Visage's approach) stitching. We measure various metrics, as discussed below.

Stitching latency: We observe that with local stitching, 95% of the runs complete stitching within 10 minutes while compressing the relevant frames to unique-frames by as much as 80% (on average). However, with global stitching the median takes ≈ 90 minutes, and the tail goes upward of 4 hours with the compression ratio compared to raw frames coming to an average of 1:8.7. The speed up resulting from having to stitch only the important parts with Visage makes it possible to run interactive surveys with feedback sent back to the surveyor within a few minutes. We have also compared our stitching speed against the software provided by a leading drone manufacturer. We report that our stitching software is up to 20% slower for raw frames with a high overlap (over 40%), whereas we are comparable for lower overlap. However, we can speed up our localized stitching also using this external module and therefore, the benefits are complementary.

Accuracy of Stitching: We measure the PSNR of both local and global stitching. We observe local stitching produces 4× reduction in number of pixels (on average) while achieving a global PSNR that is consistently over 40 dB (considered very high [13, 63, 113]) when compared to the corresponding regions carved out of a full 3D construction. This implies high quality of the constructed image and hence most applications will not be able to differentiate between unique-frames that are produced using our approach and those obtained using the global stitch approach.

Computational overhead: The ability to perform smaller local stitching jobs allowed us to run all benchmarks on a single laptop (four-core Intel i7, 16GB DRAM, NVIDIA 1060 GTX). We also evaluated the impact of adding more compute for stitching. Scaling out the cluster (more nodes) did not show huge benefits since the traditional stitching approaches are single machine (to the best of our knowledge). Visage's new local stitching approach still relies on these traditional techniques for the actual stitching. Instead, scaling up the compute (more cores on a single node) showed improvements up to a certain point. When scaling from 16 cores to 32 cores, we saw a 26% improvement in the stitching time on average. Unfortunately, further increasing the number of cores did not change the results much because of the non-parallelizable aspects that are inherent to stitching.

Approach	JPG	Visage-JPG	JXR	Visage-JXR
R-PSNR (dB) at each level				11.6 / 14.8
			14.1 / 17.3	14.1 / 17.4
		11.6 / 14.8	18.8 / 21.4	18.8 / 21.5
	mean / 99tile	41.0 / 42.6	33.9 / 36.6	33.9 / 36.7
	41.2 / 42.8	41.2 / 42.8	Inf / Inf	Inf / Inf
Compress Ratio at each level				13K / 38K
			126.3 / 141.3	157.8 / 392.9
		13K / 38K	14.3 / 18.1	18.6 / 70.3
		4.4 / 19.3	3.3 / 4.1	4.3 / 17.2
	3.1 / 4.3	9.2 / 31.5	2.4 / 3.6	2.0 / 2.6

Table 1: A summary of the benefits of the semantic-aware progressive encoding of Visage in terms of *R-PSNR* and compress ratio at each level, where the 1st level of Visage is the mask. We find that Visage improves their baseline encoders by reaching the same *R-PSNR* much earlier (*i.e.*, higher compress ratios).

Semantic Encoding & Prioritized Transfer: To evaluate benefits of the semantic-aware progressive encoding and prioritized transmission in Visage (for uploading unique-frames from the edge to cloud), we compare its performance with the industrial standard approaches: a) JPEG and PNG as traditional encoders (single layer), and b) JXR a progressive multi-layer encoder [34]. The domain-model in this evaluation is configured with only two importance levels for the sake of simplicity. *e.g.*, image segments with trees are important and those without trees are not important. In addition, for the importance ratio, i.e., the ratio of "important" pixels among the raw frames, we picked the worst-case of 69% from our dataset. We pick this value to show the benefits of Visage even when important segments are in the majority. In practice the benefits are significantly better, since the typical (95^{th} percentile) importance ratio of our dataset is less than 5%.

Across the experiments, we measure various metrics including: the compression ratio, the compression quality (*R-PSNR*, etc.), and the transfer latency (i.e., time to encode, upload the data to remote cloud, and then decode the data). We note that the encoding and decoding themselves are fairly fast and finish within 30 seconds. The upload on the other hand takes significantly longer.

Table 1 shows a quick summary of using both traditional (JPG) and progressive (JXR) encoders standalone vs. within Visage (with the added benefits of semantic encoding). Visage offers multiple layers of encoding: mask, base, delta(s), shown top to bottom in the table, regardless of employed encoders. As shown, Visage is able to achieve the same *R-PSNR* as the traditionals at each level, but at a substantial higher compression ratio (less transmission data). We note that the Visage-JPG option corresponds to state-of-the-art systems that perform importance based differential encoding in monolithic images [66]. Visage takes this idea to the extreme to perform progressive encoding and transfer.

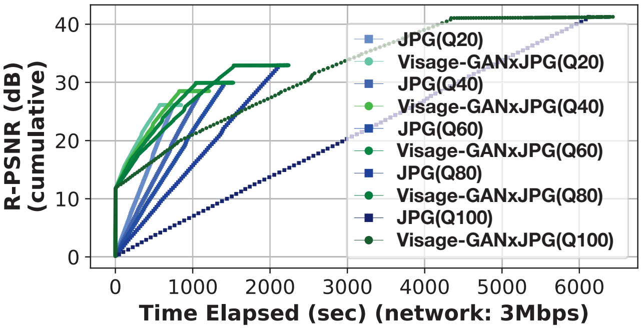


Figure 7: Comparison between Visage and baseline JPG in terms of *R-PSNR* v.s. elapsed time for transferring a batch of unique-frames to cloud. The higher the *R-PSNR* with shorter time is better. Visage constantly improves the performance of employed JPG, regardless of JPG qualities, via the GAN technique, auto-detection of important semantics, and prioritized transmission.[1]

Impact of the underlying encoder We measure the impact of the various configurations on Visage's encoder.
Impact of compression parameters: We evaluate the performance under different "quality" parameter ('Q') of the JPEG encoder and compare the performance with and without Visage, as shown in Figure 7. Visage reaches target *R-PSNR*s much quicker for all varying qualities. For example, on the highest quality of 100 (Q100), it reaches R-PSNR of 20 and 40, 70% and 30% faster respectively. In addition, at the highest quality level (Q100), we find that the benefits of Visage to scale almost linearly with the proportion of less-important data per unique frame. We also validated this by varying the importance ratio in our synthetic dataset but not shown here for the sake of brevity.
Impact of the adopted encoder: We evaluate Visage across various encoders. Figure 8 further validates the orthogonality of Visage's benefits of priority upload by showing that regardless of underlying encoding techniques, when used with Visage, the performance always improves. For instance, to reach a target *R-SSIM* of 0.8, Visage can reduce the latency by 40% over all baselines. *However, the priority uploader is only as good as the model.* We observe that our agriculture models developed with Airsim are 93% as accurate as models developed with real-world data when detecting low-yield areas in a corn-field and therefore, we encourage human inspection of data on the slow path when using models built purely based on Airsim. These comparisons correspond to systems that tune the compression parameters when uploading image from the edge to the cloud automatically according to DNN robustness in the cloud [68, 127].

Impact on application-defined metrics We measure how well Visage works for application-defined metrics, such as state-of-the-art DNN-based metrics. We evaluate this for the agriculture usecase with two user defined metrics: a) *Plant Count* counting the total number of plants in the drone survey, and b) *Plant Coverage Ratio* measuring the ratio of plant areas to other areas. For both cases, as shown in Figure 9 and Figure 10, Visage reduces the latency by over two orders of magnitude on average while reaching the same end result (less than 6% inaccuracy). The improvement varies across different datasets, with some cases reaching 13,000×. These comparisons correspond to the systems that optimize for application metrics when transferring images and videos to the cloud [66].

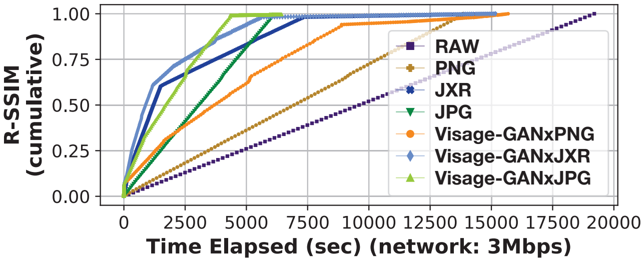


Figure 8: Comparison between Visage and various approaches in *R-SSIM* v.s. time. Visage accepts arbitrary encoders and improves their performances significantly.

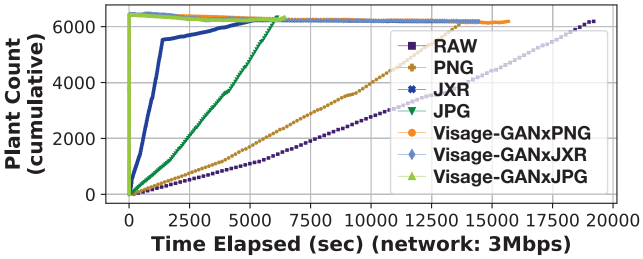


Figure 9: Comparison in *Plant Count* v.s. time among different approaches. The faster the ground-truth count (achieved by RAW) is approached, the better it is. Visage almost reach the ground-truth count instantly and improves the baseline performance by 13,000×. This can be attributed to realistic images generated by the GAN technique in Visage while only requires transferring a tiny label mask to cloud.

5.3 End-to-end Benefits & Analysis

Edge Benefits Analysis We compare deploying Visage at the edge vs. dirctly uploading all drone data to the cloud where stitching and analyzing is all done in the cloud. We compare these two options across runs from the synthetic dataset under varying edge to cloud network connections. In general, when the drone data generation rate is higher than the network bandwidth, using Visage at the edge pays off, unless the total generated data can be uploaded to the the cloud in a few minutes which is the goal for interactivity.

In particular, with just *one laptop's* worth of edge performance, Visage is typically significantly better. For the median (or 99th percentile) survey size, any uplink bandwidth lower than 15Mbps (or 80Mbps) is favourable toward using the edge. To put this in context, 40% of the United States has 3Mbps connections [19], and most data in agriculture, energy, construction, forests, and disaster management is uploaded with 3G only, as LTE and 5G coverage is limited.

End-to-end Benefits, Deployment Experiences, and Lessons Visage is deployed at six locations in North America where it has helped ingest data from over 500 drone surveys to date. Here, we describe how Visage benefits one of the larger deployments with over 2TB of data processed to date with drones covering 100s of acres of farm land on a weekly basis since Oct. 2019. We use Normalized Difference Vegetation Index (NDVI) to characterize anomalies as low NDVI typically represents areas of low agricultural yield.

[1] *Region weights* favor the plant region at a factor of 9 than others, which is determined by the end application. *Cumulative* denotes the metric is calculated based on the entire batch of transmitted frames.

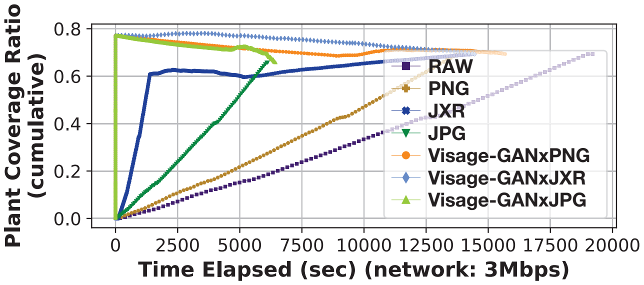


Figure 10: Comparison in *Plant Coverage Ratio* v.s. time. Visage outperforms all compression baselines by achieving a better ratio substantially faster. This improvement can be attributed to both GAN and auto-detection/prioritization in Visage which partially satisfies the evaluator at the start and then boosts up the fidelity at important regions first.

The end-to-end latency is measured as the time of ingestion of the last frame needed before the corresponding unique-frame can be stitched to the time the application generates the insight in the cloud. This is essentially representing how long does it take for the remote application to start seeing insights/inferences on the data, which is critical for interactive applications. We run this on a single-laptop edge with a 1Mbps uplink connection (rural broadband).

Across the surveys, the median and 95%ile times for the first insight for Visage are 18 and 33 minutes, respectively while they are 78 and 301 minutes, respectively when Visage is used without a domain model. Likewise, 50% of the insights are obtained by the median times of 35 and 278 minutes respectively for the two systems. Thus, Visage can interactively get feedback from the surveyor.

We also find that a majority ($> 62\%$) of important unique-frames wait less then 10% of the total survey time before the stitching starts, i.e., they are able to build good quality 3D images of important segments/objects using frames with the time between the first and last one needed for an acceptable quality 3D image being $< 10\%$ of the total survey time, therefore, showing that timeout is a good strategy for when to stop stitching.

As expected, the stitching and drone-survey latencies add to the inference latency of more complex segments/objects showing that *Visage is only as good as the domain model and can provide more benefits when the important areas are small.* At the extreme, we find in our data sets that about 10% of the localized 3D images finish only after more than 77% of the survey-time. Even for such unique-frames' inferences, Visage provides latency improvements because of its prioritized and differentiated encoding of segments. Not surprisingly, we also find that there are some drone surveys (about 5%) where our technique produces the same stitching latency as the traditional one because Visage was forced to create a large 3D image.

Our deployments have given us valuable lessons in terms of adapting the algorithm, changing the system design, and optimizing the implementation. We list a few important lessons that we learned along the way. Several design elements in Visage were motivated by these learnings and many of our current work streams and future backlogs are based on these learnings too.

- Acquisition quality matters more than analytics latency: The quality of analytics (and the latency) depends entirely on the quality of the data acquired. Very early on, we received feedback about the usage of drones and how often pilots had to go back to the field to correct for erroneous data and missing data to help obtain high-quality imagery. That motivated us to design an interactive system between the pilot and a domain expert.
- Actionable insights matter more than analytics: Statistics on collected data is important for drone imaging. However, the inference of actionable insights is even more important for a drone pilot who is also a subject matter expert capable of taking actions in the field. We had concrete feedback from one of the infrastructure usecases urging us to provide insights as the pilot also happens to be a certified technician for the domain. This motivated us to develop an interactive analytics method with a supervised model to detect anomalies.
- Live-streaming from a drone is harder than we imagined: Drones typically communicate via WiFi, Bluetooth, or ISM bands to joysticks that connect to the edge system running Visage. However, with ever-improving battery life, we realized that often the drones would travel very far quickly, and this affects the quality of live-stream data. Therefore we had to build a robust frame skipping mechanism that uses a batch of frames as a unit for figuring out what to skip and what to analyze live.
- Forget the edge, the cloud is also not ready for drones: Over the last several months, we had to build/prototype a number of machine learning and streaming services in our cloud to address high-resolution geospatial imagery. Much of the existing geospatial technology in the cloud tends to focus on satellite imagery that often has global coverage with low resolution whereas drones are optimized for local coverage with high resolution. Problems include lack of labeling tools for large and unwieldy TIFF images, lack of trained DNNs that operate directly on 3D imagery, and absence of proper inference tools that provide actionable insights with the least effort from the developers.

6 RELATED WORK

Video Analysis Systems: Many works propose video analytic pipelines for ingesting camera streams in scenarios of smart homes or smart cities. Those systems serve different purposes such as customizability of the pipeline [6, 47, 141], lower latency/cost of video processing [51, 58], better resource-quality trade-offs for analytics [57, 130, 131], or bandwidth savings by offloading preprocessing tasks to the edge [16, 17, 133]. However, they are not suitable for geospatial data. These systems perform frame-by-frame analysis on a per camera-stream basis, which is less efficient because of semantic redundancy and may also miss insights that can only be obtained from 3D models.

Several solutions have been proposed to accelerate video analytics by leveraging semantic similarity. Feature vector based systems [31, 42, 43, 128] extract a cache or a store of features found from a stream of frames to skip subsequent information with similar features. Motion vector and optical flow based systems track multiple moving cameras and objects through space and time to perform semantic deduplication, both for data representation as well as analytics [12, 56, 66, 67, 73, 128, 140]. Unlike these techniques, Visage uses field-of-view of cameras as a proxy for detecting potential overlap between frames without having to look at any data or pixels in the frames. Such a technique is handy to save resources when monitoring stationary scenes which is often the case in the commercial industry sector that we target.

Network Acceleration & Deduplication: Network acceleration and deduplication strive to remove redundancy in data transmission and storage, respectively. However, traditional techniques have focused on removing identical content[25, 37, 53, 61, 78, 100, 105, 114, 117, 139] or background elimination in stationary cameras. Our focus is on image data where there is semantic similarity that is hard to detect using hashing techniques. Instead, we use computer vision and machine learning techniques to detect duplication.

Prioritized Data Sync: While priority-based data transfer solutions exist in other settings [20, 104], traditional cloud backup solutions do not offer prioritization - being able to specify folder-level priorities remains a popular feature users request [5, 49, 106]. For instance, while Dropbox allows users to explicitly select "Sync Now" on a per-file basis, it does not support prioritizing folders or hierarchical priority levels [50]. The lack of understanding how data is generated makes it hard to automatically infer priorities in these generic solutions. In Visage, we extract application knowledge to help understand priorities. To the extent of our knowledge, the data-pyramid approach of Visage is the first priority-based file uploader.

Image & Video Encoding: Image [34, 88, 95, 120] and video compression [4, 44, 45] are widely studied techniques. While they help reduce redundant information within a frame or nearby frames, they do not eliminate redundancy from geospatial data with duplication spread across the entire dataset. Therefore, techniques such as stitching are needed. In this paper, we show how a previously unexplored tradeoff in stitching (accuracy vs. efficiency) reduces latency.

Recent research has proposed using deep learning models as the codecs [1, 10, 62, 74, 99, 112, 115, 116, 119] or artificially increasing the decoding resolution [64, 129] for visual consumption, or optimizing encoding for neural networks as opposed to humans [68, 127] for better compression. Our techniques are complementary to these. Visage builds upon progressive encoding techniques like JPEG-XR [34]/-2K [95] by exploiting semantic segmentation [8, 9, 121, 134]. However, segmentation itself is insufficient for prioritization due to the lack of knowledge of segments' relative importance. Visage builds a total order of importance for each segment type to accelerate transfers. It also helps human consumers of data by leveraging GAN techniques [55, 77] to skip transmission of unimportant segments.

ML-optimized streaming of video over unpredictable networks has been proposed [52, 71, 110, 129]. There are also techniques to give higher share of the bit rate to foreground elements that humans focus on [66, 101, 109]. However, Visage makes this more generic (not just background/foreground for human consumption).

Real-time SLAM systems: Visual simultaneous localization and mapping systems [2, 14, 39, 65] are used by autonomous vehicles and robots to navigate in the real world. These systems focus on helping an agent get from point A to point B in an unknown or a semi-known environment, from an obstacle avoidance or time/coverage optimization perspective. However, our focus is high-quality data acquisition in mostly unknown or new environments with drones, robots, vehicles that have limited on-board computation capabilities besides simple obstacle avoidance or ability to follow a preset path, and this is why Visage is complementary to SLAM optimizations.

Drone-sourced live video analytics: Several studies have been conducted towards real-time drone-video analysis [40, 73]. Common to these works is the use of GPS for fast localization of the anomaly before sending relevant frames for feature matching. [40] marks detected anomalies on virtual engineering drawings while [73] improves perceptual quality of image stitching by using optical flow. However, both of these apply only to 2D frame analytics; none builds a 3D model of the anomalies and accelerates the transmission of the 3D model's projections for helping the remote surveyor obtain real-time information.

7 DISCUSSION AND FUTURE WORK

Visage enables interactive 3D image based analytics for data from manned/unmanned, aerial/terrestrial, drone, robot, or industrial vehicles with cameras. The specific applications presented include monitoring of mostly static assets in the commercial industry (e.g. a farm, a pipeline, a wind-turbine, a railroad segment). However, the holistic benefits of 3D image analytics can apply to other scenarios as well.

An emerging geospatial scenario comes from satellite imaging which live-streams large amounts of data to data centers where it can be analyzed. An optimization system like Visage can help here with dynamic prioritized streaming and adapt the available communication bandwidth to focus on application-specific targets. The optimization of satellites and ground stations for data-transfer with prioritized streaming can enable real-time earth-scale sensing applications.

Another scenario is a static network of high-definition surveillance cameras which are fixed in location but have overlapping fields of view to holistically track and gather details from an object of interest moving in this area. In such an application, the general background and cameras are static but the object of interest, like a person in the field of vision, can move at arbitrary orientations and at fast speeds which enforces the need for multi-dimensional video analytics. A multi-dimensional (including time on top of spatial dimensions) tracking approach ensures capturing that critical information which would be neglected by two dimensional analysis.

8 CONCLUSION

Analyzing geospatial imagery generated by aerial drones, terrestrial robots, or cameras mounted on industrial vehicles in real-time at remote edge locations with weak connectivity is hard. Such raw geospatial imagery is huge in size because of redundancies and hard to analyze because geospatial applications require combining information from several frames. Today's systems operate at two extremes: a) Upload and analyze raw frames one by one in isolation which is not only slow but also misses holistic insights that can be obtained only by combining multiple frames and b) Compress and combine all the raw frames into a single 3D image to be uploaded and analyzed, which takes a substantial amount of time. We propose Visage, which prioritizes building, uploading, and analysis of only those portions of the 3D image that are deemed important for the application based on domain knowledge extracted by a segmentation model deployed at the edge. Our experiments with real deployment data shows that Visage reduces latency of analytics by up to four orders of magnitude for a wide range of geospatial-applications.

REFERENCES

[1] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. 2019. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*. 221–231.

[2] Fawad Ahmad, Hang Qiu, Ray Eells, Fan Bai, and Ramesh Govindan. 2020. CarMap: Fast 3D Feature Map Updates for Automobiles. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 1063–1081. https://www.usenix.org/conference/nsdi20/presentation/ahmad

[3] Microsoft AirSim. https://github.com/microsoft/AirSim.

[4] Alliance for Open Media Video 1 Codec. https://aomedia.org/av1-features/.

[5] Allow priority settings for files being uploaded. https://onedrive.uservoice.com/forums/913522-onedrive-on-windows/suggestions/6802608-allow-priority-settings-for-files-being-uploaded.

[6] Ganesh Ananthanarayanan, Yuanchao Shu, Landon Cox, and Victor Bahl. Microsoft-Rocket-Video-Analytics-Platform, https://github.com/microsoft/Microsoft-Rocket-Video-Analytics-Platform.

[7] N. Ayache and F. Lustman. 1991. Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 1 (Jan 1991), 73–85. https://doi.org/10.1109/34.67633

[8] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll. 2018. A Deep Convolutional Neural Network for Video Sequence Background Subtraction. *Pattern Recognition* 76 (2018), 635–649.

[9] Mohammed Chafik Bakkay, Hatem A Rashwan, Houssam Salmane, Louahdi Khoudour, D Puigtt, and Yassine Ruichek. 2018. BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 4018–4022.

[10] Johannes Ballé, Valero Laparra, and Eero Simoncelli. 2019. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017*.

[11] Drone Base. How Drones Are Supporting Renewable Energy. https://blog.dronebase.com/how-drones-are-supporting-renewable-energy.

[12] Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael Cafarella, Tim Kraska, and Sam Madden. 2020. MIRIS: Fast Object Track Queries in Video. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1907–1921. https://doi.org/10.1145/3318464.3389692

[13] A. Baviskar, S. Ashtekar, and A. Chintawar. 2014. Performance evaluation of high quality image compression techniques. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 1986–1990. https://doi.org/10.1109/ICACCI.2014.6968643

[14] Ali J. Ben Ali, Zakieh Sadat Hashemifar, and Karthik Dantu. 2020. Edge-SLAM: Edge-Assisted Visual Simultaneous Localization and Mapping. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (MobiSys '20)*. Association for Computing Machinery, New York, NY, USA, 325–337. https://doi.org/10.1145/3386901.3389033

[15] Bentley Systems for Maintenance Inspections. https://www.bentley.com/en/solutions/transportation-inspection.

[16] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G. Andersen, Michael Kaminsky, and Subramanya R. Dulloor. 2019. Scaling Video Analytics on Constrained Edge Nodes. In *arXiv, cs.CV, 1905.13536*. arXiv:cs.CV/1905.13536

[17] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. 2015. Glimpse: Continuous, Real-Time Object Recognition on Mobile Devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*. Association for Computing Machinery, New York, NY, USA, 155–168. https://doi.org/10.1145/2809695.2809711

[18] UAV Coach. Agricultural Drones: How Drones Are Revolutionizing Agriculture and How to Break into this Booming Market. https://uavcoach.com/agricultural-drones/.

[19] Federal Communications Commission. 2020 Broadband Deployment Report. https://docs.fcc.gov/public/attachments/FCC-20-50A1.pdf.

[20] Resilio Connect. Job Priority. https://connect.resilio.com/hc/en-us/articles/360000006084-Job-priority-.

[21] DroneDeploy For Construction. https://www.dronedeploy.com/solutions/construction/.

[22] CropQuest Precision Agriculture Data. https://www.cropquest.com/precision-ag-drone-images-big-data/.

[23] University of Washington CSE455. Features and Image Matching. https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect6.pdf.

[24] Ana I. De Castro, Jorge Torres-Sánchez, Jose M. Peña, Francisco M. Jiménez-Brenes, Ovidiu Csillik, and Francisca López-Granados. 2018. An Automatic Random Forest-OBIA Algorithm for Early Weed Mapping between and within Crop Rows Using UAV Imagery. *Remote Sensing* 10, 2 (2018). https://doi.org/10.3390/rs10020285

[25] Biplob Debnath, Sudipta Sengupta, and Jin Li. 2010. ChunkStash: Speeding up Inline Storage Deduplication using Flash Memory. In *2010 USENIX Annual Technical Conference (ATC)* (2010 usenix annual technical conference (atc) ed.). USENIX. https://www.microsoft.com/en-us/research/publication/chunkstash-speeding-up-inline-storage-deduplication-using-flash-memory/

[26] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[27] DJi Mavic 2 Mini Specs. https://www.dji.com/mavic-mini/specs.

[28] DJI Recommended Overlap for Drone Imagery. https://support.pix4d.com/hc/en-us/articles/203756125-How-to-verify-that-there-is-enough-overlap-between-the-images.

[29] DJI Terra. https://www.dji.com/dji-terra.

[30] Docker Container Environment. https://docs.docker.com/get-started/overview/.

[31] Utsav Drolia, Katherine Guo, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan. 2017. Cachier: Edge-Caching for Recognition Applications. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 276–286. https://doi.org/10.1109/ICDCS.2017.94

[32] DroneDeploy. Agremo Plant Count Health. https://www.dronedeploy.com/product/market/nyocdrbegyikewvm/#agremo-plant-count-health.

[33] DroneDeploy Recommended Overlap for Drone Imagery. https://support.dronedeploy.com/docs/making-successful-maps.

[34] F. Dufaux, G. J. Sullivan, and T. Ebrahimi. 2009. The JPEG XR Image Coding Standard. *IEEE Signal Processing Magazine* 26, 6 (November 2009), 195–204.

[35] Ramón A. Díaz-Varela, Raúl De la Rosa, Lorenzo León, and Pablo J. Zarco-Tejada. 2015. High-Resolution Airborne UAV Imagery to Assess Olive Tree Crown Parameters Using 3D Photo Reconstruction: Application in Breeding Trials. *Remote Sensing* 7, 4 (2015), 4213–4232. https://doi.org/10.3390/rs70404213

[36] EarthSense TerraSentia Terrestrial LiDAR Sensing. https://www.earthsense.co/.

[37] Ahmed El-Shimi, Ran Kalach, Ankit Kumar, Adi Ottean, Jin Li, and Sudipta Sengupta. 2012. Primary Data Deduplication—Large Scale Study and System Design. In *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*. USENIX, Boston, MA, 285–296. https://www.usenix.org/conference/atc12/technical-sessions/presentation/el-shimi

[38] F. Flammini, C. Pragliola, and G. Smarra. 2016. Railway infrastructure monitoring by drones. In *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)*. 1–6. https://doi.org/10.1109/ESARS-ITEC.2016.7841398

[39] Jorge Fuentes-Pacheco, Jose Ascencio, and J. Rendon-Mancha. 2015. Visual Simultaneous Localization and Mapping: A Survey. *Artificial Intelligence Review* 43 (11 2015). https://doi.org/10.1007/s10462-012-9365-8

[40] Shilpa George, Junjue Wang, Mihir Bala, Thomas Eiszler, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2019. Towards Drone-Sourced Live Video Analytics for the Construction Industry. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3301293.3302365

[41] Yan Li Hang Cui Min Xu Dan Wu Henrik Rydén Sakib Bin Redhwan Guang Yang, Xingqin Lin. 2018. A Telecom Perspective on the Internet of Drones: From LTE-Advanced to 5G. In *arXiv, cs.NI, 1803.11048*. arXiv:cs.NI/1803.11048

[42] Peizhen Guo, Bo Hu, Rui Li, and Wenjun Hu. 2018. FoggyCache: Cross-Device Approximate Computation Reuse. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 19–34. https://doi.org/10.1145/3241539.3241557

[43] Peizhen Guo and Wenjun Hu. 2018. Potluck: Cross-Application Approximate Deduplication for Computation-Intensive Mobile Applications. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '18)*. Association for Computing Machinery, New York, NY, USA, 271–284. https://doi.org/10.1145/3173162.3173185

[44] H.264/AVC1: Advanced Video Coding. https://www.itu.int/rec/T-REC-H.264.

[45] H.265/HEVC: High Efficiency Video Coding. https://www.itu.int/rec/T-REC-H.265.

[46] Richard Hartley and Andrew Zisserman. 2004. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511811685

[47] H. He, Z. Shao, and J. Tan. 2015. Recognition of Car Makes and Models From a Single Traffic-Camera Image. *IEEE Transactions on Intelligent Transportation Systems* 16, 6 (2015), 3182–3192.

[48] Headwall Hyperspectral Remote Sensing. https://www.headwallphotonics.com/hyperspectral-sensors.

[49] How to prioritize the upload of files. https://support.google.com/drive/forum/AAAAOxCWsToJvZX4JEObwg/?hl=fr.

[50] How to set file syncing priority. https://help.dropbox.com/installs-integrations/sync-uploads/prioritize-files-sync.

[51] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B. Gibbons, and Onur Mutlu. 2018. Focus: Querying Large Video Datasets with Low Latency and Low Cost. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, 269–286.

[52] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A Buffer-based Approach to Rate Adaptation: Evidence from A

Large Video Streaming Service]. In *Proceedings of the 2014 ACM conference on SIGCOMM*. 187–198.

[53] Sunghwan Ihm, KyoungSoo Park, and Vivek S. Pai. 2010. Wide-Area Network Acceleration for the Developing World. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference (USENIXATC'10)*. USENIX Association, USA, 18.

[54] Intel. Intel Xeon Processors, https://www.intel.com/content/www/us/en/products/details/processors/xeon.html.

[55] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[56] Samvit Jain, Xun Zhang, Yuhao Zhou, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, Paramvir Bahl, and Joseph Gonzalez. 2020. Spatula: Efficient cross-camera video analytics on large camera networks. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*. 110–124. https://doi.org/10.1109/SEC50012.2020.00016

[57] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: Scalable Adaptation of Video Analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 253–266. https://doi.org/10.1145/3230543.3230574

[58] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. Optimizing Deep CNN-Based Queries over Video Streams at Scale. *CoRR* abs/1703.02529 (2017). arXiv:1703.02529 http://arxiv.org/abs/1703.02529

[59] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev. 2019. Early Forest Fire Detection Using Drones and Artificial Intelligence. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 1060–1065. https://doi.org/10.23919/MIPRO.2019.8756696

[60] Kubernetes Cluster Manager. https://kubernetes.io/docs/concepts/.

[61] Fujitsu Laboratories. https://www.fujitsu.com/global/about/resources/news/press-releases/2014/0916-01.html.

[62] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang. 2018. Learning Convolutional Networks for Content-Weighted Image Compression. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3214–3223. https://doi.org/10.1109/CVPR.2018.00339

[63] X. Li and J. Cai. 2007. Robust Transmission of JPEG2000 Encoded Images Over Packet Loss Channels. In *2007 IEEE International Conference on Multimedia and Expo*. 947–950. https://doi.org/10.1109/ICME.2007.4284808

[64] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 136–144.

[65] H. Lim, J. Lim, and H. J. Kim. 2014. Real-time 6-DOF monocular visual SLAM in a large-scale environment. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 1532–1539. https://doi.org/10.1109/ICRA.2014.6907055

[66] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge Assisted Real-Time Object Detection for Mobile Augmented Reality. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 25, 16 pages. https://doi.org/10.1145/3300061.3300116

[67] Xiaochen Liu, Pradipta Ghosh, Oytun Ulutan, B. S. Manjunath, Kevin Chan, and Ramesh Govindan. 2019. Caesar: Cross-Camera Complex Activity Recognition. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems (SenSys '19)*. Association for Computing Machinery, New York, NY, USA, 232–244. https://doi.org/10.1145/3356250.3360041

[68] Zihao Liu, Xiaowei Xu, Tao Liu, Qi Liu, Yanzhi Wang, Yiyu Shi, Wujie Wen, Meiping Huang, Haiyun Yuan, and Jian Zhuang. Machine Vision Guided 3D Medical Image Compression for Efficient Transmission and Accurate Segmentation in the Clouds. arXiv:cs.CV/1904.08487

[69] David Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60 (11 2004), 91–. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[70] Arko Lucieer, Steven M. de Jong, and Darren Turner. 2014. Mapping landslide displacements using Structure from Motion (SfM) and image correlation of multi-temporal UAV photography. *Progress in Physical Geography: Earth and Environment* 38, 1 (2014), 97–116. https://doi.org/10.1177/0309133313515293 arXiv:https://doi.org/10.1177/0309133313515293

[71] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 197–210.

[72] Francisco Massa and Ross Girshick. 2018. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch.

[73] Xiangyun Meng, Wei Wang, and Ben Leong. 2015. SkyStitch: A Cooperative Multi-UAV-Based Real-Time Video Surveillance System with Stitching. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. Association for Computing Machinery, New York, NY, USA, 261–270. https://doi.org/10.1145/2733373.2806225

[74] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool. 2019. Practical Full Resolution Learned Lossless Image Compression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10621–10630. https://doi.org/10.1109/CVPR.2019.01088

[75] Microsoft. Azure Blob Storage https://azure.microsoft.com/en-us/services/storage/blobs/.

[76] Microsoft. Surface Book 2https://www.microsoft.com/en-us/p/surface-book-2/8mcpzjjcc98c?activetab=pivot%3aoverviewtab.

[77] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. In *arXiv, cs.LG, 1411.1784*. arXiv:cs.LG/1411.1784

[78] NetApp. NetApp Data Compression, Deduplication, and Data Compaction. https://www.netapp.com/us/media/tr-4476.pdf.

[79] NVIDIA. NVIDIA GeForce GTX graphics cards, https://www.nvidia.com/en-us/geforce/10-series/.

[80] NVIDIA Corporation. NVIDIA TITAN RTX, https://www.nvidia.com/en-us/titan/titan-rtx/.

[81] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. 2016. Visually Indicated Sounds. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2405–2413. https://doi.org/10.1109/CVPR.2016.264

[82] Onur Özyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. 2017. A Survey on Structure from Motion. *CoRR* abs/1701.08493 (2017). arXiv:1701.08493 http://arxiv.org/abs/1701.08493

[83] Parrot Sequoia+ Hyperspectral Drone Camera. https://www.sensefly.com/camera/parrot-sequoia/.

[84] Pix4D. Inspect, analyze and visualize your crop changes all year round. https://www.pix4d.com/product/pix4dfields.

[85] Pix4D. Processing Large Datasets. https://support.pix4d.com/hc/en-us/articles/202558579-Processing-Large-Datasets.

[86] Pix4D. https://www.pix4d.com/.

[87] Pix4D Recommended Overlap for Drone Imagery. https://support.pix4d.com/hc/en-us/articles/203756125-How-to-verify-that-there-is-enough-overlap-between-the-images.

[88] Portable Network Graphics (PNG). http://libpng.org/pub/png/libpng.html.

[89] PostGIS: A Geospatial Information Services Engine for PostgreSQL. https://postgis.net/.

[90] PostGIS: Geometries. http://postgis.net/workshops/postgis-intro/geometries.html.

[91] PostGIS Polygon Query Performance Profiling. https://carto.com/blog/postgis-performance/.

[92] PostGIS ST Intersection Optimization. https://www.r-spatial.org/r/2017/06/22/spatial-index.html.

[93] NORTH POWDER. https://northpowder.com/.

[94] PrecisionAg. Big Data, Big Crops? How Ag Can Harness the Power of Satellites and the Cloud. https://www.precisionag.com/digital-farming/big-data-big-crops-how-ag-can-harness-the-power-of-satellites-and-the-cloud/.

[95] Majid Rabbani and Rajan Joshi. 2002. An Overview of the JPEG 2000 Still Image Compression Standard. *Signal Processing: Image Communication* 17, 1 (2002), 3 – 48.

[96] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *arXiv, cs.LG, 1511.06434*. arXiv:cs.LG/1511.06434

[97] Railroads continue to tap drone technology to inspect track, bridges. https://www.progressiverailroading.com/mow/article/Railroads-continue-to-tap-drone-technology-to-inspect-track-bridges--57270.

[98] Real-time Aerial Mapping with a GPU Cluster.

[99] Oren Rippel and Lubomir Bourdev. 2017. Real-Time Adaptive Image Compression. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. International Convention Centre, Sydney, Australia, 2922–2930.

[100] riverbed. https://www.riverbed.com/products/steelhead/steelhead-sd-wan.html.

[101] Abdul H. Sadka. 2002. *Compressed Video Communications*. Wiley.

[102] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Red Hook, NY, USA, 2234–2242.

[103] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:cs.CV/1801.04381

[104] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. 1990. Coda: a highly available file system for a distributed workstation environment. *IEEE Trans. Comput.* 39, 4 (April 1990), 447–459. https://doi.org/10.1109/12.54838

[105] Citrix SD-WAN. https://www.citrix.com/products/citrix-sd-wan/.

[106] Set sync priority on a per-folder basis. https://forum.odrive.com/t/set-sync-priority-on-a-per-folder-basis/636.

[107] Sebastian Siebert and Jochen Teizer. 2014. Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Automation*

in Construction 41 (2014), 1–14. https://doi.org/10.1016/j.autcon.2014.01.004

[108] Rajendra P. Sishodia, Ram L. Ray, and Sudhir K. Singh. 2020. Applications of Remote Sensing in Precision Agriculture: A Review. *Remote Sensing* 12, 19 (2020). https://www.mdpi.com/2072-4292/12/19/3136

[109] Yu Sun, Ishfaq Ahmad, Dongdong Li, and Ya-Qin Zhang. 2006. Region-based Rate Control and Bit Allocation for Wireless Video Transmission. *IEEE Transactions on Multimedia* 8, 1 (2006), 1–10.

[110] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 272–285.

[111] Lina Tang and Guofan Shao. 2015. Drone remote sensing for forestry research and practices. *Journal of Forestry Research* 26 (06 2015), 791–797. https://doi.org/10.1007/s11676-015-0088-y

[112] L. Theis, W. Shi, A. Cunningham, and F. Huszár. 2017. Lossy Image Compression with Compressive Autoencoders. In *International Conference on Learning Representations*. https://openreview.net/pdf?id=rJiNwv9gg

[113] N. Thomos, N. V. Boulgouris, and M. G. Strintzis. 2006. Optimized transmission of JPEG2000 streams over wireless channels. *IEEE Transactions on Image Processing* 15, 1 (2006), 54–67. https://doi.org/10.1109/TIP.2005.860338

[114] Achieving Storage Efficiency through EMC Celerra Data Deduplication. https://www.dell.com/community/s/vjauj58549/attachments/vjauj58549/celerra/660/1/h6065-achieve-storage-effficiency-celerra-dedup-wp.pdf.

[115] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. 2015. Variable Rate Image Compression with Recurrent Neural Networks. In *arXiv, cs.CV, 1511.06085*. arXiv:cs.CV/1511.06085

[116] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. 2017. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5306–5314.

[117] DOT: Data-Oriented Transfer. http://www.cs.cmu.edu/~dot-project/.

[118] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. 2000. Bundle Adjustment — A Modern Synthesis. In *Vision Algorithms: Theory and Practice*, Bill Triggs, Andrew Zisserman, and Richard Szeliski (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 298–372.

[119] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*. 4790–4798.

[120] G. K. Wallace. 1992. The JPEG Still Picture Compression Standard. *IEEE Transactions on Consumer Electronics* 38, 1 (Feb 1992), xviii–xxxiv.

[121] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* (2020).

[122] Xiaolong Wang and Abhinav Gupta. 2016. Generative image modeling using style and structure adversarial networks. In *European conference on computer vision*. Springer, 318–335.

[123] Z. Wang and Q. Li. 2011. Information Content Weighting for Perceptual Image Quality Assessment. *IEEE Transactions on Image Processing* 20, 5 (2011), 1185–1198.

[124] Simon Winder, Gang Hua, and Matthew Brown. 2009. Picking the Best Daisy. In *Computer Vision and Pattern Recognition* (computer vision and pattern recognition ed.). IEEE Computer Society. https://www.microsoft.com/en-us/research/publication/picking-the-best-daisy/

[125] WJE. Utilizing Drones to Assess and Maintain Your Structure's Integrity. https://www.wje.com/knowledge/webinars/detail/utilizing-drones-to-assess-and-maintain-your-structures-integrity.

[126] Changchang Wu. 2013. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*. IEEE, 127–134.

[127] Xiufeng Xie and Kyu-Han Kim. 2019. Source Compression with Bounded DNN Perception Loss for IoT Edge Computer Vision. In *Mobicom 2019*. 1–16.

[128] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. 2018. DeepCache: Principled Cache for Mobile Deep Vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 129–144. https://doi.org/10.1145/3241539.3241563

[129] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural Adaptive Content-Aware Internet Video Delivery. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*. 645–661.

[130] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzynek, and Edward A. Lee. 2018. AWStream: Adaptive Wide-Area Streaming Analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 236–252. https://doi.org/10.1145/3230543.3230554

[131] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J. Freedman. 2017. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, Boston, MA, 377–392.

[132] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European conference on computer vision*. Springer, 649–666.

[133] Tan Zhang, Aakanksha Chowdhery, Paramvir (Victor) Bahl, Kyle Jamieson, and Suman Banerjee. 2015. The Design and Implementation of a Wireless Video Surveillance System. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom '15)*. Association for Computing Machinery, New York, NY, USA, 426–438. https://doi.org/10.1145/2789168.2790123

[134] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2881–2890.

[135] Zhichao Chen and S. T. Birchfield. 2007. Person following with a mobile robot using binocular feature-based tracking. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 815–820. https://doi.org/10.1109/IROS.2007.4399459

[136] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[137] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Segmentation on MIT ADE20K dataset in PyTorch. https://github.com/CSAILVision/semantic-segmentation-pytorch.

[138] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[139] Benjamin Zhu, Kai Li, and Hugo Patterson. 2008. Avoiding the Disk Bottleneck in the Data Domain Deduplication File System. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST'08)*. USENIX Association, USA, Article 18, 14 pages.

[140] Yuhao Zhu, Anand Samajdar, Matthew Mattina, and Paul Whatmough. 2018. Euphrates: Algorithm-SoC Co-Design for Low-Power Mobile Continuous Vision. In *Proceedings of the 45th Annual International Symposium on Computer Architecture (ISCA '18)*. IEEE Press, 547–560. https://doi.org/10.1109/ISCA.2018.00052

[141] ZoneMinder. https://zoneminder.com/features/.