Technical Disclosure Commons

Defensive Publications Series

August 2020

Optimized Upload of Telemetry Data

Robert Bruce Bahnsen

Yan Mayster

Optimized Upload of Telemetry Data

ABSTRACT

Modern telemetry entails the gathering of data (per location or time interval) in such large quantities that the data often exceeds the capacity of the communication channels to a server (or cloud). An example of such telemetry is the gathering of street-level imagery and video where, even when the data-collecting vehicle has a wireless 4G connection, a very large amount of data is gathered so quickly that it is written on storage media such as solid state, optical, magnetic storage, etc. and physically mailed to a processing center. Besides the delay attributable to physical mailing, manual processing of the storage media at the point of capture also causes a substantial delay.

This disclosure describes techniques to optimize the latency and bandwidth of sensor-acquired telemetry data by rapidly determining, e.g., at the point of capture, the value of specific pieces of data (image, sequence of video frames, etc.), and by utilizing the available communication and computational capabilities of the device to process data in the order of relative value.

KEYWORDS

- Telemetry
- Machine learning
- Street-image capture
- Street view
- Aerial video
- Drone video
- Data ranking

BACKGROUND

Modern telemetry entails the gathering of data (per location or time interval) in such large quantities that the data often exceeds the capacity of the communication channels to a server (or cloud). An example of such telemetry is the gathering of street-level imagery and video where, even when the data-collecting vehicle has a wireless 4G connection, a very large amount of data is gathered so quickly that it is written on such as solid state, optical, magnetic storage, etc. and physically mailed to a processing center.

Besides the delay attributable to physical mailing, manual processing of the storage media at the point of capture also causes a substantial delay. For example, the driver of a street-image capturing vehicle adds storage media (e.g., magnetic or optical disks, solid state drives, etc.) to a box and ships the box once it is filled up or upon a timeout. The manual processing and mailing of telemetry data results in delays that can be on the order of weeks between the capture of data and its public availability, such that the captured data can grow stale as the physical world changes.

Furthermore, due to the lack of an immediate feedback loop between the remote vehicle and the processing center, there is no clear understanding of the value of particular pieces of telemetry data, e.g., imagery, at the time of its capture. The value is only eventually (if ever) determined by additional analysis effort, after expending the full costs of ingesting all the captured imagery. These issues are compounded by the lack of compute capabilities on the remote capturing devices, which tend to be special-purpose, thin-client, or lightweight equipment with relatively low computational power.

Other examples of telemetry that have similar characteristics, e.g., large amounts of data collected in relatively short spans of time, data of varying value, multiple return channels, return channels of differing and time-varying latency and bandwidth, include: aerial image collection; automotive and flight data collection; remote healthcare; wildlife and forestry management; etc.

DESCRIPTION

This disclosure describes techniques to optimize the latency and bandwidth of sensor-acquired telemetry data by rapidly determining, e.g., at the point of capture, the value of specific pieces of data (e.g., images, sequence of video frames, etc.), and utilizing available communication and computational capabilities to process this data in order of estimated relative value. In the example of street-image capture, images of urban localities that have undergone large and recent change can be sent while images that show no change with reference to their last-captured values can be dropped.

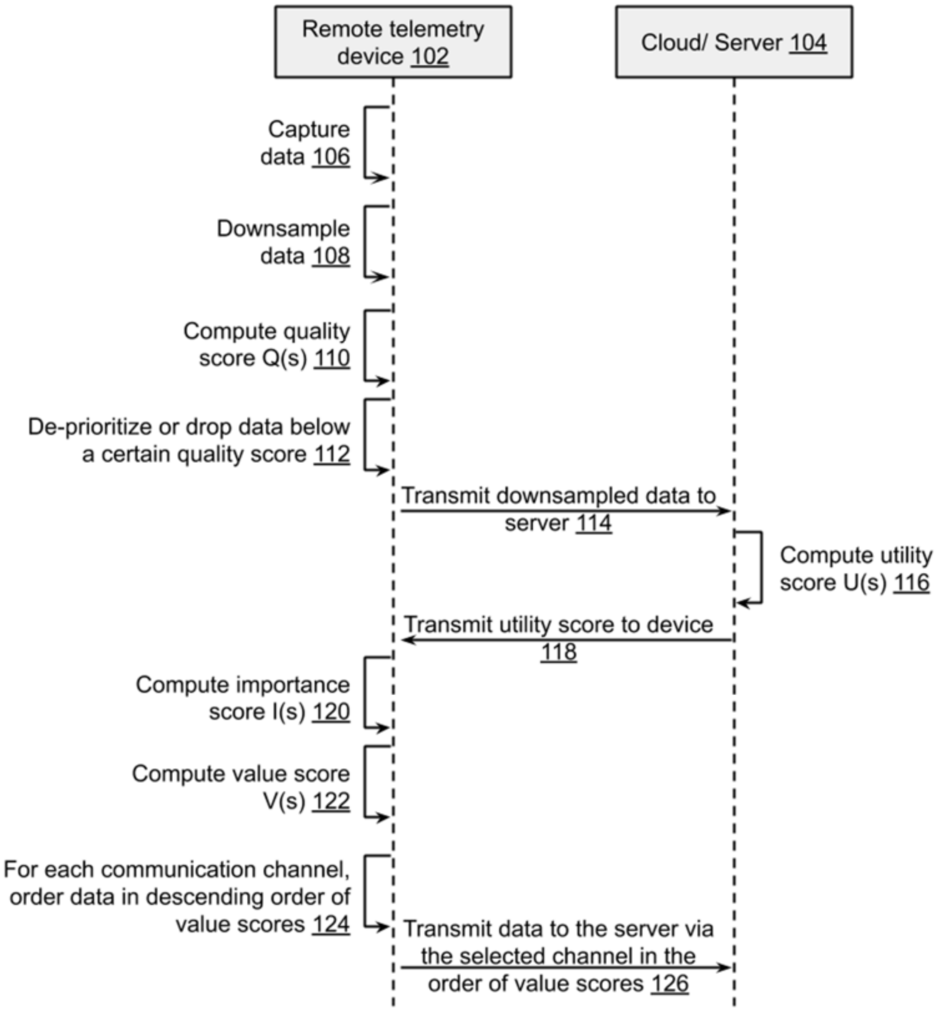

Fig. 1: Optimized upload of telemetry data

Fig. 1 illustrates optimized upload of telemetry data, per the techniques of this disclosure. A remote telemetry device (102) captures data (106). Some examples of telemetry data include street-image; aerial image; automotive and flight data; remote healthcare; wildlife and forestry management; etc. To transmit the data to the cloud (or server, 104), the device has multiple communication channels of varying bandwidth and latency, e.g., wireless 4G or 5G; single-disk overnight mail; disk batch mail, e.g., waiting for multiple storage media to be filled; or even the null channel, e.g., discarding data.

The captured data is downsampled (108), e.g., low-resolution images or low frame-rate videos obtained from the original data, such that data sent back for analysis over the live link (if any) is kept low. The downsampling is optional, e.g., no downsampling is performed if onboard compute capability permits computation of a quality score $Q(s)$ from the captured data. Depending on the compute capabilities available at the remote device, the data, whether in its downsampled version or its original high-resolution, full frame-rate version, is analyzed (in real time or near real time) to assign a non-contextual *quality score* (110), $Q(s)$, indexed by the serial number s of the data (or data-batch). The quality score is computed on-device, e.g., without upload to the cloud.

For example, in a street-image telemetry application, low-quality images include images that have a high degree of occlusion, e.g., storefronts that are blocked by trucks. As another example, in aerial-image telemetry, low-quality images include images where ground features are covered in clouds or smog. Conversely, in a weather-telemetry application, clouds or other weather conditions detected in the image are valuable examples of high-quality images. Images can also be assigned a low quality score due to technical attributes, e.g., poor light exposure, inappropriate shutter speed, incorrect ISO setting, etc. Data with $Q(s)$ below some threshold can then be safely dropped (112), e.g., sent to the null channel.

The downsampled data is sent for analysis to the server (114). The downsampling rate is selected to be such that the transmission takes up only a small fraction of the bandwidth of the live channel, thereby keeping the overhead cost for determining the priority of data transmission low. Based on the received, downsampled data, the server computes a context-based *utility score* (116), $U(s)$, which is a measure of its usefulness or importance.

For example, in a street-image telemetry application, an image-gathering vehicle may traverse arterial roads multiple times a day due to route-planning constraints. Street images obtained from vehicles driving over arterial roads multiple times a day may be duplicative and less useful, and receive a low utility score, if there is little substantive change across runs. Conversely, images received from a less frequently imaged area (e.g., a cornfield or a forest, imaged once in several months or years) may receive a relatively high utility score even if not displaying significant change. Indeed, for images from areas such as forests, it may be of importance to know that there has been little or no change.

As alluded to in the above examples, the utility score can be based not only on the data itself but on metadata as well, e.g., the location and time of capture. Also, the refresh cadence for the location may be taken into account, e.g., as explained earlier, a relatively low score in an infrequently visited area may be modulated to be higher than an otherwise similar score in a frequently visited area.

Additionally, the utility score can be based upon auxiliary data, e.g., historical data gathered for that location or neighborhood, or data gathered from other sources at similar times, etc. Differences between auxiliary data and the incoming data can be used to determine if the incoming data represents churn, or sufficiently new information. For example, in the street-image telemetry application, a new street-front, an on-going street-festival, a temporary spike in local population density, a new building, the removal of an old building, the change in name of a store, etc., are examples of information that can be determined as new based on the difference between auxiliary and incoming data.

The computation of the utility score can take the form of an iteration. For example, a low-resolution sample image may be sent to the cloud-based system to assign a score. If the score is relatively high, the device can send a single full-resolution image to the cloud, subject to the bandwidth availability at the particular time, to request a score with higher confidence. The logic for deciding how to use upload bandwidth can be managed by the cloud (pull model), by the device (push model), or a combination of the two.

An image can have a high quality score (e.g., it shows a clear, unobstructed street-image) but have a low utility score (e.g., it doesn't have new information). Likewise, an image can have a low-quality score (e.g., it is blurry), but have a high utility score (e.g., it shows important changes to the street that need to be communicated to the public).

The utility score is transmitted by the server to the remote telemetry device (118). At the remote device (or at the server), a total *importance score* $I(s)$ is derived (120) as a combination of the utility score $U(s)$ and the quality score $Q(s)$. For example, $I(s)$ can be a weighted linear combination of $Q(s)$ and $U(s)$, or can be a custom, application-dependent formula that establishes an appropriate balance between $Q(s)$ and $U(s)$.

The importance score of a piece of data is compared against the overall cost of utilizing each available communication channel, and is assigned a total *value score* $V(s)$ that represents the overall value of the data (122). The value score accounts for factors such as the amount of transfer time needed over a certain communication channel; the costs of using the communication channel; the availability of cloud-based and device-based computational capabilities; the amount of such computational resources (and the length of time) that would be consumed by the present piece of data (e.g., the opportunity cost); the expected inflow of more useful data in the near future that can more fruitfully occupy the communications channel; etc.

For example, a video that has a high importance score but very high cost of transmission over existing communication channels, or experiences a temporary non-availability of the most appropriate communications channel, may receive a low value score. As another example, GPS data, which is important metadata to most telemetry, can receive a high value score by virtue of both its criticality and its relatively small size. In each case, for each available communication channel, data is queued in descending order of its $V(s)$ score. When bandwidth becomes available, the data is uploaded (126), ordered by highest $V(s)$ score first.

Although the techniques do not entail any specific, absolute, latency deadline, the process works on a near real time basis. For example, the value of telemetry data being collected is assessed in a real time manner in relation to the current bandwidth and compute capability (including on-device GPU, ASIC, FPGA, or CPU) utilization. Given the current resources (bandwidth, local storage, and compute) and a pool of telemetry data, e.g., imagery, the use of available resources is optimized.

The techniques can be operative during telemetry data collection and after. For example, communications between the device and the server can continue after the data collection is stopped. In the example of street-image telemetry, it is typical to not collect images during the night; however, the remote telemetry device can continue to run and upload imagery data to the cloud during the night.

Local storage, which is a finite resource (similar to bandwidth and computational capability), can also be managed and accounted for during quality, utility, importance, and value scoring. For instance, if the imagery is of low interest, e.g., of an unchanging freeway, that imagery can be deleted by the device, saving storage space for more important data. Additionally, deleting relatively unimportant images saves upload bandwidth, since the transmission of full-resolution imagery to the cloud is skipped. For the lowest importance imagery that still passes the threshold for not discarding it, disks or other storage media can be eventually shipped in the mail so that full resolution imagery is made available eventually. In this manner, the described techniques enable the optimal usage of local storage.

When bandwidth availability is low, the data quality sent over the upload link can be further reduced (downsampled), with a lower confidence score being indicated in the associated utility score. Depending on the utility score, for highly important/urgent data, the server can proceed with the low-quality (e.g., low-confidence utility score) data in production services. If the bandwidth drops to the point that the network connection is dropped, the data is buffered until the network returns, utilizing available storage and ordering the obtained data by the on-device computed quality scores alone, queueing these for cloud-based contextual scoring when the connection becomes available.

Both on-device and cloud-based scores can be determined by the use of machine learning models to detect objects of interest (or lack thereof) in the data. In various applications, examples of detected objects indicative of high quality can include results of text recognition for certain patterns, street signs, generic storefronts, etc. Detected objects of low quality include artifacts and occlusions. As mentioned earlier, the quality score can be generated based on detected objects using on-device capabilities. At the server end, as mentioned earlier, tasks such as churn detection, comparison against the age of the most recently obtained data for the area, etc., can be performed to obtain the utility score.

The accuracy needed to post telemetry data, e.g., street imagery, in near real time can be achieved using a variety of techniques, e.g., GPS, inertial measurement units, dead reckoning, landmark recognition, road snapping, VPS, etc. On-device pose estimation can be improved at a later time when the full frame-rate, full-resolution data becomes available.

In this manner, the techniques of this disclosure enable determination of the most appropriate communication channel and processing type for captured telemetry data, based on the value of the data segment. The value tier that a given data segment falls in is based on metrics computed using on-device and/or server-side compute resources. Utilization of the uplink channel that transmits telemetry data to the server is optimized by sending data in descending order of value.

CONCLUSION

This disclosure describes techniques to optimize the latency and bandwidth of sensor-acquired telemetry data by rapidly determining, e.g., at the point of capture, the value of specific pieces of data (image, sequence of video frames, etc.), and by utilizing the available communication and computational capabilities of the local device to process data in the order of relative value.