Verification of Video Frame Latency Telemetry for UAV Systems Using a Secondary Optical Method

Sam Siewert[1] Muhammad Ahmad[2]
Trellis-Logic LLC, Boulder, Colorado, 80302, USA

Kevin Yao[3]
AAI Corporation, Hunt Valley, Maryland, 21030, USA

This paper presents preliminary work and a prototype computer vision optical method for latency measurement for an UAS (Uninhabited Aerial System) digital video capture, encode, transport, decode, and presentation subsystem. Challenges in this type of latency measurement include a no-touch policy for the camera and encoder as well as the decoder and player because the methods developed must not interfere with the system under test. The goal is to measure the true latency of displayed frames compared to observed scenes (and targets in those scenes) and provide an indication of latency to operators that can be verified and compared to true optical latency from scene to display. Latency measurement using this optical computer vision method was prototyped using both flight side cameras and H.264 encoding using off-the-shelf equivalent equipment to the actual UAS and off-the-shelf ground systems running the Linux operating system and employing a Graphics Processor Unit to accelerate video decode. The key transport latency indicator to be verified on the real UAS is the KLV (Key Length Value) time-stamp which is an air-to-ground transport latency that measures transmission time between the UAS encoder elementary video stream encapsulation and transmission interface to the ground receiver and ground network analyzer interface. The KLV time-stamp is GPS (Global Positioning System) synchronized and employs serial or UDP (User Datagram Protocol) injection of that GPS clock time into the H.264 transport stream at the encoder, prior to transport over an RF (Radio Frequency) or laboratory RF-emulated transmission path on coaxial cable. The hypothesis of this testing is that the majority of capture-to-display latency comes from transport due to satellite relay as well as lower latency line-of-sight transmission. The encoder likewise must set PTS/DTS (Presentation Time Stamp / Decode Time Stamp) to estimate bandwidth-delay in transmission and in some cases may either over or underestimate this time resulting in either undue added display latency or frame drop-out in the latter case. Preliminary analysis using a typical off-the-shelf encoder showed that a majority of observed frame latency is not due to path latency, but rather due to encoder PTS/DTS settings that are overly pessimistic. The method and preliminary results will be presented along with concepts for future work to better tune PTS/DTS in UAS H.264 video transport streams.

Nomenclature

B-frame = bi-directional inter-frame compressed frame using previous I-frame and future P-frame
CCTV = Closed Circuit Television
DCT = Discrete Cosine Transform
DTS = Decode Time Stamp
D_{system} = system delay from scene capture to display
$D_{capture}$ = camera delay for image capture and readout
$D^{e}_{reorder}$ = encoder delay for H.264 frame reordering for compression

---

[1] Senior Partner, Trellis-Logic LLC, 1942 Broadway, Suite 314, Boulder CO 80302; Assistant Professor, University of Alaska Anchorage, 3211 Providence Drive, Anchorage AK 99508, AIAA Senior member.
[2] Senior Consultant, Trellis-Logic LLC, 1942 Broadway, Suite 314, Boulder CO 80302.
[3] Video Engineer IV, Unmanned Aircraft Systems, AAI Corporation, Hunt Valley Maryland 21030.

$D^{e}_{process}$ = encoder delay for MPEG4 compression of digital video and H.264 packet processing
D^{e}_{buffer} = encoder buffer delay prior to unicast or multicast transport
$D_{network}$ = network transport delay for a digital video frame packet
D^{d}_{buffer} = decoder buffer delay prior to start of decoding
$D^{d}_{process}$ = decoder delay to decompress a frame
$D_{present}$ = player and display frame presentation delay
GoP = Group of Pictures, the I-frame and subsequent P-frames and B-frames in H.264
GPS = Global Positioning System
GPU = Graphics Processing Unit
H.264 = digital video compression and transport format and standard
HDMI = High Definition Media Interface
I-frame = intra-frame compressed frame using DCT and quantization of macroblocks
I/O = Input and Output
IP = Internet Protocol
KLV = Key, Length, Value
LAN = Local Area Network
LCD = Liquid Crystal Display
MPEG = Motion Picture Experts Group, a standards organization for digital video encoding
MPEG4 = the fourth revision of the MPEG standard for digital video encoding (compression)
NAL = Network Abstraction Layer, H.264 High Efficiency Video Coding standard packet format
NMEA = National Marine Electronics Association, typically used for serial GPS information
no-touch = uses event trace or I/O analysis and does not statically or dynamically link or modify application
NTSC = National Television Systems Council
OV = Optical Verification, the secondary analog optical method of verification
PCR = Program Clock Reference, sent from the encoder to the decoder to keep both synchronized over time
P-frame = inter-frame compressed frame using previous I-frame of P-frame and motion vector quantization
PPS = Pulses per Second for GPS time
PTS = Presentation Time Stamp
RF = Radio Rrequency
SDI = Synchronous Digital Interface
SLR = Single Lens Reflex camera
SMPTE = Society of Motion Picture and Television Engineers
UAS = Uninhabited Aerial System, including flight and ground equipment
UAV = Uninhabited Aerial Vehicle, a key component in the UAS
UDP = User Datagram Protocol, a connectionless Internet transport layer protocol

Introduction

Trellis-Logic LLC completed an experiment to better understand impact of using H.264 encoding on UAS, focused on latency measurement for digital video encode, transport, decode, and presentation frame-by-frame. Challenges in this type of latency measurement include a no-touch policy for the camera and encoder as well as the decoder and player. The goal of the work is to measure the true latency of displayed frames compared to observed scenes (and targets in those scenes) and provide an indication of latency to operators that can be verified and compared to true optical latency from scene to display. Latency measurement was completed using off-the-shelf H.264 encoding equipment, SDI camera, and an off-the-shelf Linux personal computer with PCI-e expansion slots and off-the-shelf graphics processing unit for H.264 decoding. The key transport latency indicator used in UAS systems is the KLV time-stamp[1], an air-to-ground transport latency time-stamp that measures transmission time between the UAS encoder elementary video stream encapsulation and transmission interface to the ground receiver and video presentation systems. The KLV time-stamp in the experiment described herein is GPS synchronized and employs serial or UDP injection of that GPS clock time into the H.264 transport stream at the encoder, prior to transport over a laboratory RF-emulated transmission path.

The original hypothesis for the experiment described in this paper was that the majority of capture-to-display latency most likely comes from transport delay due to bent pipe satellite relay as well as lower latency line-of-sight transmission. What was observed was that while this is true, the encoder likewise must set PTS/DTS to estimate bandwidth-delay in transmission and to buffer an entire GoP for encoding; if PTS/DTS is not properly tuned the encoder often over estimates transmission latency time and may use a large GoP to reduce bandwidth, either or both of which result in undue added display latency. Ideally the PTS/DTS should cause minimal delay in the decoder and presentation buffers of the ground system. To measure true frame delay and to verify the correctness of injected GPS time-stamp KLV data, Trellis-Logic LLC developed optical methods to determine true capture-to-display latency using an out-of-band analog video channel (run over RF coaxial) to compare to KLV injection indicators and encoder PTS/DTS. The basic finding was that extra undue latency due to conservative settings for PTS/DTS could be avoided by designing the decoder/player to tune PTS/DTS settings to be no larger than a small GoP and expected transmission delay, but that runs the risk of possible decode errors and partial frame update for worst case transmission delay; so perhaps a better approach is to use KLV time-stamps and optical verification methods to tune the encoder for the transmission path based on field tests and operational history.

This optical method is referred to herein as the OV (optical verifier) is compared to the KLV time-stamp indicators of transport latency and frame rate estimation. In laboratory testing with the off-the-shelf H.264 encoder, it’s clear that this encoder has conservative default settings for PTS/DTS often combined with large default GoPs, and if not tuned, the default introduces more capture-to-display latency than would be required based on actual transport latency. Based on this work, the authors recommend that KLV time-stamp and secondary optical latency verification methods be used for actual UAS/UAV camera-encoders and ground receivers to evaluate and tune both air-based encoders and ground-based decoders to assist with optimization of true observation latency (using the optical verifier) and to monitor it during operations (using the KLV time-stamp indicators).

Digital Video Latency Measurement Goals and Prior Work

The goal for the work presented in this paper was to provide a no-touch approach that requires no modification to off-the-shelf digital cameras and encoders (that are being analyzed for use in UAS or are already in use) and no modification to off-the-shelf digital video decoders and players. The reason for this is that it allows for performance comparison of commercial off-the-shelf solutions that can save cost and potentially provide the same or better performance than custom built solutions. However, the evaluation of and verification of these potentially lower-cost solutions is not simple since the system integrator may not have ready access to instrument key segments of the overall solution. For example, an encoder that has many of the desired features and low cost for H.264 encoding (power consumption, packaging, SDI camera interface, competitive cost) might not have a simple feature to time stamp the H.264 video frames to record time of acquisition from an off-the-shelf SDI camera. At the same time, H.264 does specify a packet injection standard for KLV and most encoders that would be considered allow for KLV injection with time-stamps[2]. To summarize, the goals for the project included:

1. No direct modification to encoders or decoders is allowed and the measurement system must work with any H.264 compliant UAS encoder and off-the-shelf commercial decoder.
2. Encoder and decoder software must be used as-is to allow off-the-shelf hardware and software solutions to be compared using the latency measurement method.
3. Direct comparison of performance must be possible by simply substituting hardware and software elements in a solution including the UAS camera and encoder as well as the ground decoder and presentation player.
4. The latency analysis should provide scene capture to display latency in frame periods latent (e.g. 16.67 millisecond periods for 60Hz digital video), but time-stamps used should be at least GPS accurate.
5. Analysis should log frame latency between the UAS camera capture and the presentation player as well as provide a display overlay.

The ability to change in and out various segments of the overall UAS digital video camera capture, encode, transport, decode and presentation system allows for comparison of competing off-the-shelf solutions in each segment of the UAS so that the best, lowest-cost and highest performance system can be integrated. The no-touch requirement allows for use of segment solutions as-is and without custom requirements for segment solution providers, keeping costs lower.

The initial concepts considered were simple observation of a digital clock that was envisioned to also display GPS time (therefore synchronized to the KLV injection) to millisecond accuracy, but as anyone knows who has attempted this, the authors also found that clocks captured this way on video that are not fully camera capture synchronized will produce blurry non-legible clock images. It would also be possible to use a standard such as the SMPTE time code, but it was not obvious how to integrate this directly into off-the-shelf H.264 encoders that might not already provide SMPTE time code as a feature[3]. So, the most obvious replacement for an external clock or an integrated time code feature in a video frame is a pattern generated from a well-timed graphics generator that can be observed and captured for manual analysis or machine vision analysis. This was the approach taken and the revised approach was first outlined in detail during the early investigation phase of the project where standards for time code insertion in both the video data and the ancillary data were reviewed – the key finding in this investigation was that not all encoders support some of these newly emergent standards and often while the time codes may be used in specific digital transmission environments for broadcast, they may not be available for more purpose-built UAS digital video links[4]. The goal was not to limit encoders that could be tested based on standards support, but rather to open up the potential to consider a wider range of H.264 encoders. The pattern generation and observation technique was found to work well and has been used by prior researchers with similar goals. This might be of interest to future analysis and research in this area, so the next section provides a quick survey of methods and past results.

Related Prior Work

The key challenge of the no-touch latency measurement by segment is really based on the limitation of not being able to simply modify and instrument segments like the encoder. If this was allowed, then the encoder could simply add GPS time-stamp data logging or in-band data to do so. The KLV metadata standard does require UAS H.264 equipment to allow for time-stamp injection, which the authors used, but by the standard, it's not totally clear when the KLV H.264 packets are added to the transport stream relative to digital video frame acquisition from the camera, compression encoding with MPEG4, or transmission over the IP network. In some sense, the goal of this work was not only to provide measurements, but to verify the standardized method of KLV time-stamp injection to ensure that it is a valid indicator of frame latency if used as a real-time overlay for player frame presentation. The overlay is envisioned to show frame display latency with red (significantly more frames than expected latency), yellow (one or fewer frames more latent than expected), or green (with expected latency due to necessary encoding and transport delays only). The overlay might have +1, +2, …, +n frames latent indication or might even show milliseconds of latency greater than expected. The point of this is a trust indicator for the user, so they would know if they were seeing frames more latent than say a half second of unavoidable latency.

The goal to verify KLV injection latency indication and overlay and the measure the true scene capture to display latency is a goal that has been shared by digital video security[5]. Work at the University of Adelaide shared the goal to measure frame capture to display latency using digital IP cameras compared to older CCTV (Closed Circuit Television) analog cameras. In this work, the researchers used a display to produce an indicator in time and to measure the latency until this indicator would also simultaneously appear on an IP security camera display system being analyzed along with an analog CCTV system as recorded by a digital SLR camera taking snapshots of both. From this method, they were able to compare the scene observation latencies of each digital IP security camera compared to CCTV and the actual time the indicator was changed to millisecond accuracy.

The work presented here took a similar approach, but had the added goal to measure latency of frames continuously and to also use the KLV metadata injection standards for H.264 to provide a log of frame latency over time and a real-time overlay showing current scene to display latency quality metrics (e.g. red, yellow, or green based on comparisons of actual to expected latency). The assumption for the Adelaide work is that any given IP security system will have stable latency over time (due to the encoding and network transport). In the UAS work, the goal was to design a method that can assess latency over time since UAS may use more than one downlink transport path (satellite, line of site, and various ground-based networks) and since the high compression requirements and features of H.264 can lead to more variation in encoder latency. The potential for more latency variation using H.264 and object-based temporal differences over large groups of pictures between I-Frames is a latency issue likewise noted by the Adelaide researchers and more recently by researchers working to minimize the encoder contributions to H.264 latency[6]. The overall latency of a frame is a simple latency sum as follows:

$$D_{system} = D_{capture} + D^{e}_{reorder} + D^{e}_{process} + D^{e}_{buffer} + D_{network} + D^{d}_{buffer} + D^{d}_{process} + D_{present}$$

This sum has been well noted in prior research on latency contributions end-to-end for H.264 networked digital video systems. Overall, the latency contributions can be categorized by the device that adds the latency, starting with the SDI digital camera (in our experiment), followed by the stages of H.264 encoding including encoder frame reordering, compression and packet processing, and temporary buffering prior to transmission over a network; followed by network transport delay; followed by decoder buffering (based on DTS, the decode time-stamp), decoder parsing and decompression processing; followed finally by the display driver presentation (based on PTS).

Stated simply, the goal of this work was to determine the contributions to latency between capture and presentation and to ideally determine the major contributions. As a secondary goal, the focus was the computation of a latency indicator that can be overlaid on the presented images so a viewer has a good indication of how latent the images are compared to capture time (a continuous latency quality indicator). As will be explained in key findings, the D^{d}_{buffer} was by far the largest contributor to latency that we observed and likely could be significantly reduced with better PTS/DTS computations and/or tuning of the encoder.

Analysis Method, Test Configuration and Preliminary Results

Based on the goals to create a no-touch latency analysis system, Trellis-Logic LLC built an experiment to make use of both KLV injection and a secondary optical frame latency analysis method to evaluate the latency added by the encoder, the network and the decoder as diagrammed in Figure 1, with the idea that this allows for quick swap-out and compare analysis of competing encoders and decoders, but also to provide a method for indication of frame latency in operation when the network is likely to be the main contributor to variation in frame latency. In the rest of this section the details of the elements used in this experiment and the software built are described along with preliminary results and key findings using the software system and off-the-shelf hardware configuration developed

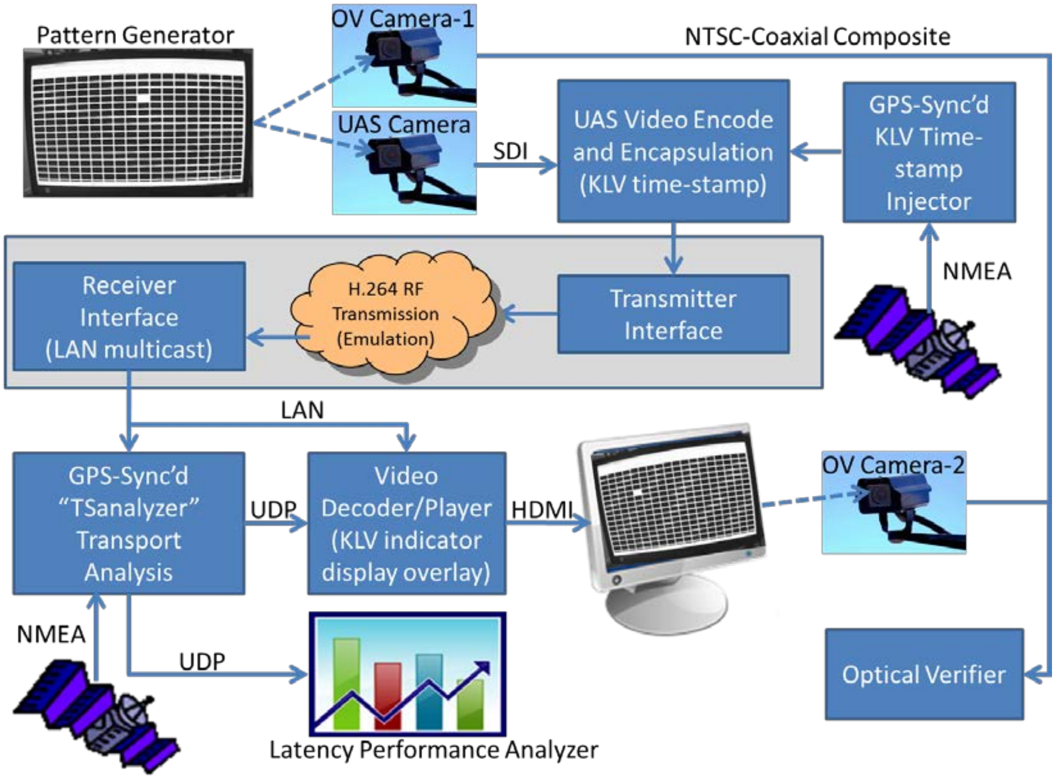

Figure 1. Encode and Transmission Latency Measurement. *A machine generated pattern at frame rate is observed by the in-band SDI camera with off-the-shelf encoder and transported through an RF emulator, buffered according to PTS/DTS and displayed. The out-of-band secondary optical method uses an analog link with minimal latency to compare the pattern displayed to the pattern observed (the secondary analog method has no observable frame-to-frame latency over the coaxial analog link). The latency in the encoded digital transmission path is the summation of the transmission latency and the buffer delay on the decoder.*

for the investigation using SD-SDI cameras[7] and newer HD-SDI cameras[8].

Optical Verification and Transport Analysis Configuration

The processing and transport segments from the camera capture interface down to the video presentation including both optical and transport latency measurement methods are depicted in Figure 1. The system in Figure 1 includes an out-of-band analog camera system which is used to observe the pattern generated for the UAS digital SDI camera and encoder. The two analog cameras in Figure 1 are linked via an RF coaxial cable to run NTSC analog video directly to two frame grabbers on the same computer for both the UAS scene verification camera and the decoder display verification camera. When both of these cameras were tested by observing the same pattern generator, there was no frame difference measureable between them because there were synchronized by the test system microcode and the latency of the continuous analog transmission is far less than a frame. As expected, the same pattern generation viewed by the SDI digital camera and displayed remotely produced a constant frame offset that could be explained by the transmission and encode latency plus buffer and hold time by the decoder according to the H.264 PTS/DTS set by the encoder. Likewise, the GPS time used to fill in KLV time-stamps in-band indicated the H.264 encapsulation and transmission latency.

Key Findings

The system as described herein was used to compare actual capture-to-display frame latency using the pattern generator and the secondary optical visual analyzer. Overall, in the off-the-shelf camera and encoder test configuration, the capture-to-display latency seen was far higher than the transport latency alone – the most probable conclusion is that the encoder PTS/DTS is set very conservatively by off-the-shelf encoders like the encoder tested, typically for 12 to 21 frames of latency, such that there is plenty of time to deliver H.264 packet data and buffer it on the decoder side – so most of the latency is due to D^{d}_{buff} based on the experiment completed and reported here. The test system allowed for additional transport latency introduction on the digital path and most encoders allow for tuning of key encoding parameters, however it was not obvious how to adjust PTS/DTS on the encoder, which would however be useful for minimizing any unnecessary decode and presentation delay.

Overall, based on the preliminary results from simple cases analyzed here, it is believed therefore that the value of these methods will be for tuning of encoder settings and for indication and continuous measurement of the transport latency (and contribution to potential frame loss or decode/display latency) given current encoder settings and actual air-to-ground transmission delays. To emulate transmission delays typical of flight environments, a layer 2 switch with delay capability can be used – in our case we used an Internet protocol forwarding machine that added a configurable delay to the video transport unicast or multi-cast UDP packets. The remainder of this paper presents the details of the secondary optical latency measurement system design and the preliminary latency findings for the off-the-shelf SDI camera, encoder and decoder/player system tested.

Transport Stream Analyzer

The latency analysis system shown in Figure 1 includes a custom built software application called *TSanalyer* which can parse the data in each 188 byte H.264 transport stream packet including digital video and metadata[9] to track the time of arrival for each packet type with GPS time-stamp accuracy and to determine which MPEG4 I-frame, P-frame, or B-frame the packet belongs to if it is frame data or which type of metadata packet it is if it is not video data. This simple real-time parser can intercept multicast H.264 packets emitted by the encoder on the same network that are also consumed by the decoder. By using multicast, the *TSanalyzer* essentially receives the H.264 packets at the same time as the decoder (when connected to a common IP switch). The TSanalyzer can also work unicast and parse and analyze H.264 encapsulated in UDP IP packets that are in turn forwarded to the decoder with minimal added latency. Both methods were used in the analysis presented here and it was found that neither LAN approach (multicast or unicast with forwarding) added significant delay compared to a frame period, so multicast was used on all encoders that supported this feature. The TSanalyzer and KLV GPS time-stamp injection is intended to be the operational solution for transport and scene observation latency indication with real-time overlays, but the goal was also to determine how accurate this type of indicator is and if the encoder or camera itself contributes significant latency to the overall end-to-end latency between scene observation and display.

Pattern Generator Design

The pattern generator used a simple grid that was configured to be 18x16 rows and columns with a filled in grid location that tracks from the top left corner to the bottom right and wraps back at a configurable delay. The pattern generator grid display provides a positive frame-by-frame latency indicator which does not blur or have issues with positive readout, and worst case, if the indicator image is captured in transition, some double illuminations were observed, but this was a minor issue since the goal was only to know the full frames of latency rather than the exact time (this was known by GPS and comparatively frame latency always exceeded the GPS time-stamp encapsulation and transport alone). Hardware methods to generate a gridded pattern were explored along with software that drives an LCD. Overall, an off-the-shelf 60Hz refresh, 2 millisecond latency pixel response monitor was found to be sufficient for displaying this pattern generated by a simple Python script running on an under loaded dedicated windows computer. The occasional pattern markers observed in two cells at the same time, when the indicator was transitioning from illumination of one cell to the next, introduces no more than 16 milliseconds of uncertainty. All testing was completed with the pattern generator set to 0.01666667 seconds between cell illuminations. The OV NTSC cameras used were only accurate to 30Hz, so reducing the grid pattern generation rate down to 0.03333333 seconds reduces the number of double illumination transitions observed, but likewise results in reduced accuracy of one 30Hz frame period. Faster shutter OV cameras and frame grabbers are a potential improvement, but the low-cost NTSC analog equipment used was sufficient to prove the OV concept and to determine that transport latency was less than the latency introduced by the encoder settings for PTS/DTS. Off-the-shelf *CameraLink* cameras have frame rates of 100Hz or more, so the error in OV could be significantly reduced with better equipment.

In-band GPS Time-stamp Injection

The off-the-shelf H.264 encoder and SDI camera used allowed for UDP or serial injection of KLV data into the H.264 transport packet stream. The experiment setup used two GPS clocks and GPS time-stamps for KLV packet injection at the encoder side that were compared with GPS time-stamps captured at the decoder interface to compute transport latency alone and the frame rate possible on the digital link between the encoder and the decoder. Fundamental to this approach is GPS time, which is synchronized to millisecond accuracy or better throughout the test system via sampling of NEMA PPS (Pulses Per Second) over serial interfaces to Garmin GPS receivers – for GPS the ultimate resolution is 14 nanoseconds[10], but the ability to compute time on any node in the test system is

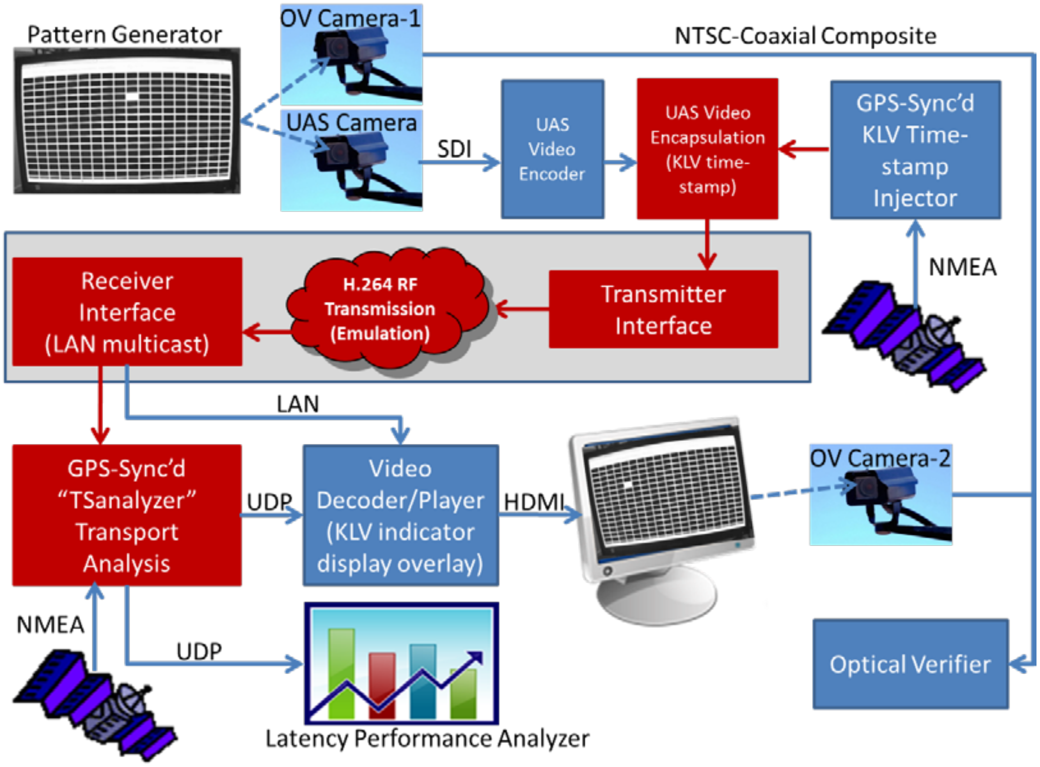

Figure 2. H.264 Packet Transport Latency. *The H.264 packet latency starts at the encoder injection (after video frames are encoded into MPEG4) and up until the decoder parser receives each KLV injection packet, but before the video packet MPEG4 is fully decoded and long before it is presented based on PTS. This path is shown in red in the diagram above.*

based upon the signal processing in the receiver and the Linux NTPD (Network Time Protocol Daemon), which reduced global time knowledge in the system to a millisecond. Millisecond resolution was more than sufficient for the proposed latency measurements for maximum frame rates of 60Hz (16.67 milliseconds).

Difference between Encoder to Decoder Transport Latency and Scene Capture to Display Latency

The difference between the latency observed for transport between the encoder and decoder and the pattern generator indicator latency measurement would by design include any buffer-and-hold time on the decoder due to PTS/DTS and any UAS camera capture and encoder latency. This is shown in Figure 2 by noting the KLV time-stamp packet path shown in red between the KLV injection and the decoder packet parser compared to the overall path.

Overall, in lab testing, the PTS/DTS provided by the encoder was the main determinant of the minimum capture-to-display latency (typically 12 to 21 30Hz frames with minimal jitter – within 400 to 700 milliseconds) rather than the transport time or estimated I-frame rate (note that the rate of B-frames or P-frames is much higher than I-frames by nature of the compression in MPEG4). Adjustment of PTS/DTS and the size of the GoP used by the encoder was not considered or tuned on the off-the-shelf H.264 encoder – default settings were used.

Latency of H.264 Packets Alone

The GPS PPS derived time-stamps injected at the encoder, after MPEG4 encoding, prior to transport, and at the decoder interface, prior to decode and presentation, showed latency consistently lower than the pattern generator latency measured using the OV. As shown in Figure 3, the latency of the KLV injected H.264 packets varied from 10 to just over 80 milliseconds.

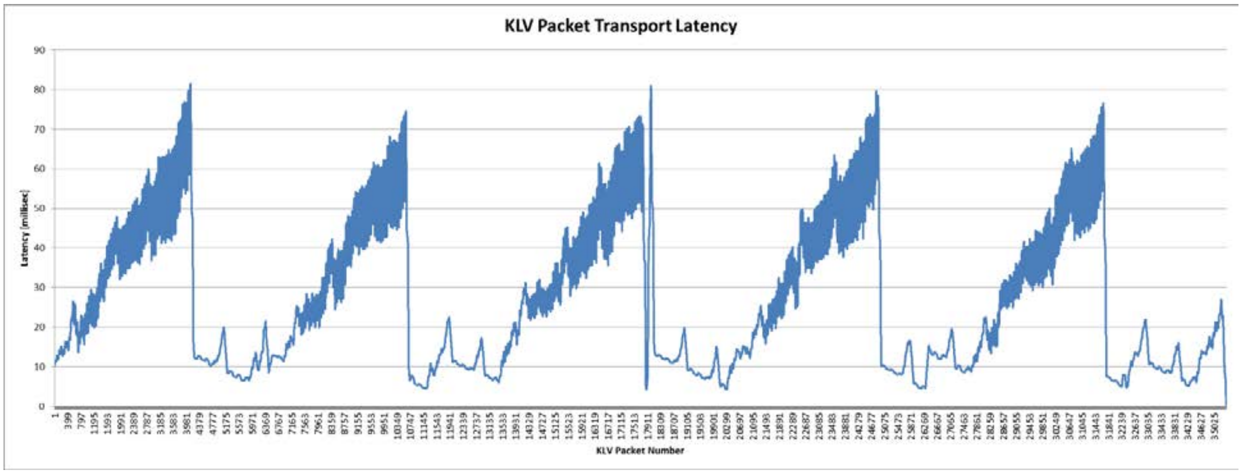

Figure 3. H.264 Packet Transport Latency Alone. *The H.264 packet latency was no more than 80 milliseconds (or at least 2 to 3 30Hz frames) compared to 12 to 21 frames of OV observed scene capture to display latency – so presumably at least 8 to 17 frames of latency are due to the encoder and PTS/DTS buffering.*

In general, even worst case, the H.264 packet latency was less than the total pattern generator OV measured latency by 620 milliseconds maximum to at least 280 milliseconds minimum. While it was not possible to measure the off-the-shelf H.264 encoder internal MPEG4 encoding delay, it is believed that the majority of the delay can be attributed to PTS/DTS which causes the decoder to buffer and hold frames longer than is really required for smooth video, but is required if B-frames or P-frames are re-ordered in a GoP. For a UAS digital video system it might be more important to have lower latency observation (within 4 frames) rather than guaranteed smooth video presentation and higher compression. With the capture to display latency approaching almost half of a second or more (in the case of 8 to 17 additional frames of latency observed in our testing) the impact of fully buffering a whole GoP (observed GoP was normally 12 frames) is a substantial issue with H.264 when added to the unavoidable transport delays.

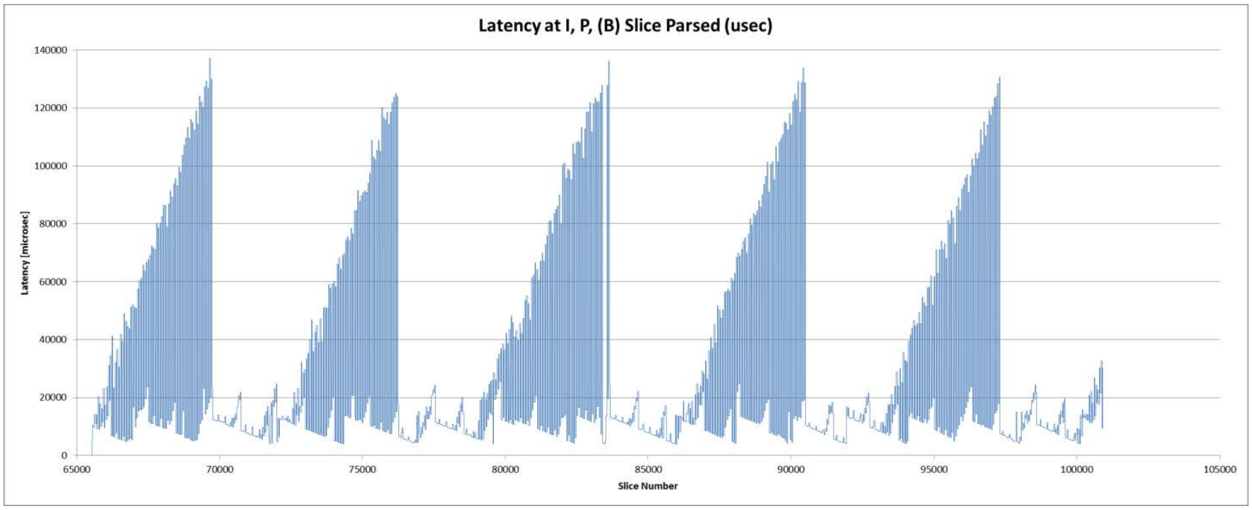

Figure 4. I-Frame and P-Frame First Packet Transport Latency Alone. *The latency between I-frame or P-frame first packets in each frame (note that in our experiment the encoder did not produce B-frames, or bi-directionally encoded frames in GoP sequences) is a good indicator of frame latency between the encoder and decoder. Our analysis showed no more than 140 milliseconds of latency between frames, so not significantly more than 4 frames of true latency in the worst case.*

Verification of H.264 Packet Latency with I-Frame Latency Measurements

To verify packet latency and the frame rates between the encoder and the decoder interface, the time delay between I-frames (the first frame in a Group of Pictures) was measured. Frame rate was found to match the 30Hz expected, but with jitter and lag as shown in Figure 4.

Without modifying the GoP size or default off-the-shelf H.264 encoder settings, a maximum of 140 milliseconds was observed in one test, but this is still no more than about 4 frames of latency for 30Hz SD-SDI video (133.33 milliseconds) from the SDI camera and H.264 encoder used – so it would appear that 4 frames of latency was actually due to the time required to encode a frame in the GoP and transport of the packets to the decoder. The remaining minimum of 8 more frames of latency therefore was assumed to come from buffer-and-hold by the decoder due to the set PTS/DTS by the encoder, which was at least three times what was needed based on transport bandwidth delay alone. The PTS observed is plotted in Figure 5 and DTS was normally not present (zero), which means that DTS should be assumed equal to PTS in H.264. No attempt was made to compute the Time to Present

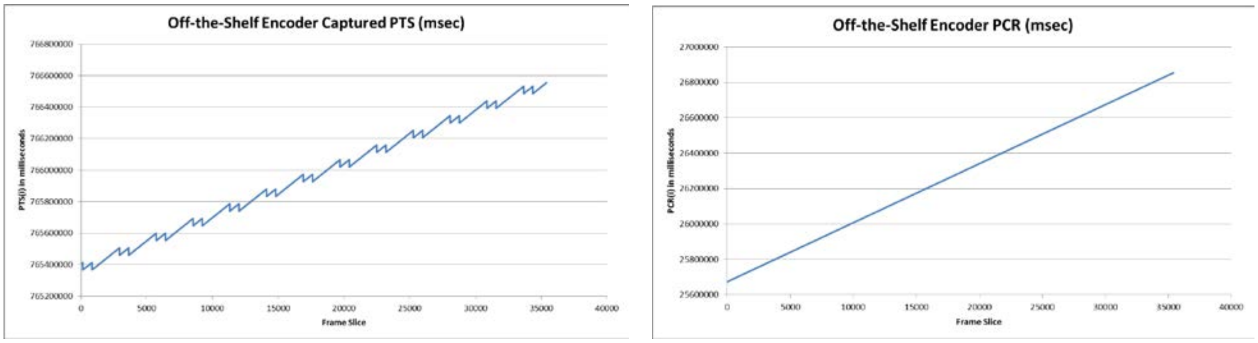

Figure 5. Frame Slice PTS and PCR Plots. *The off-the-shelf encoder tested produced H.264 required PTS/DTS, the presentation and decode time-stamps which varied between 30 and 400 milliseconds (one frame to one full GoP) as well as PCR, the program clock from the encoder to keep the decoder synchronized. The PCR has expected monotonically increasing value, but the PTS has some negative values indicating no buffer delay as well as positive where the encoder is specifying delay. As noted in the H.264 specification, DTS was zero, which means that it should be assumed to be the same as PTS.*

from the computed PTS values in Figure 5, but this can be done – the OV showed that while transport was on the order of 3 or 4 frames of real latency between the encoder and decoder, the observed pattern to display pattern was at least 12 frames latent or more (up to 21). Computation of the presentation time could be accomplished based on the PTS/DTS parsing information, but even without the frame ordering information that would be required, the variation of the PTS adjusted for frequency (1/90,000) shows that the delay introduced by the encoder is about one full 12 frame GoP.

To further refine the preliminary analysis presented here, with more information on the $\text{Encoder}_{\text{order}}$, the delay due to the PTS settings captured could be computed to exactly account for this buffer-delay contribution to verify the observation that it appears to be set to one full GoP on average. For example, based on the parsed I, P, and B frames starts and the PTS in each GoP, the time of presentation can be computed[11]. The computation depends upon knowledge of GoP display order, encoding order, the PTS time-stamp, the GoP size (many frames for the off-the-shelf H.264 encoder), display frame rate (typically 30 or 60), and frequency of the encoder system time clock. One reason this was not done in this study is that with the off-the-shelf encoder, the system time clock frequency and details of the GoP (Group of Picture) ordering and size were not readily available – one downside of using all off-shelf-equipment for this testing. The computation to present a frame in a GoP for H.264 is summarized as:

$$Time\ to\ Present = PTS_i - \frac{(N_{GoP} - (Display_{order} - Encoder_{order}))}{(Framerate\ x\ Frequency)}$$

It is theorized based on observations of the KLV transport latency in this preliminary analysis that the estimated frame rate, and observed actual time from capture-to-display using the OV is such that the PTS is being set high by the encoder with defaults for H.264 of at least one full GoP, which accounts for the difference of 8 to 17 frames between OV frame latency and the frame latency that would be attributed to transport alone.

Method to Automate Optical Verifier Frame Latency Analysis

The preliminary work presented here relied upon human observation of captured pattern generator samples to compare, which required tedious (and potentially error prone) comparison of grids and indicator positions from the two OV analog cameras. Methods to automate this comparison were explored and the authors believe this could be automated with machine vision algorithms to segment the grid and to find the center of the indicated and compare it to the center of each grid location. The closest grid location to the indicator position would be the most likely true indicator position. A simple machine vision automation solution for this could likely be built using OpenCV (Open Computer Vision software), MATLAB or any number of machine vision toolkits and well known algorithms to segment images and to locate the centroids of segmented objects like the indicator and the grid locations.

Conclusion

With an off-the-shelf encoder using default settings it was found that PTS/DTS adds significant latency between scene observation and display – in the case of our experiment 8 additional frames at least to a worst case of 4 frames of required encoder and transport latency – therefore tripling observing latency with decoder buffer-and-hold that was designed for presentation of smooth video rather than low latency video. The method to verify this used was a secondary analog optical method, which did not require any modification of the off-the-shelf SDI camera, encoder or decoder and display. More work to adjust PTS/DTS and GoP size settings on the encoder could be done, but the focus of this work was to simply test the concept for secondary optical latency measurement and to verify the transport segment latency measurement using KLV GPS time injection compared to encode and buffer-and-hold decode. The results of this analysis show that the method of secondary optical latency measurement provides valuable insight into system latency performance and that perhaps PTS/DTS is often set too conservatively by default (appears to by one full 12 frame GoP of added delay by default), contributing to unduly long buffer-and-hold delays on decoders.

References

[1] SMPTE 336M-2007, Society of Motion Picture & Television Engineers, 2007.
[2] Motion Imagery Standards Board, "*UAS* Datalink Local Metadata Set", Standard 0601.4, 4 March, 2010.
[3] SMPTE 12M-2-2008, "Television – Transmission of Time Code in the Ancillalry Data Space (Revision of RP 188-1999)", Society of Motion Picture & Television Engineers, 2007.
[4] EBU Recommendation R 122, Version 2.0, "Material Exchange Format Timecode Implementation", EBU (European Broadcasting Union), UER (Union européenne de radio-télévision), Geneva, November 2010.
[5] Rhys Hill, Christopher Madden, Anoton van den Hengel, Henry Detmold, Anthony Dick, "Measuring Latency for Video Surveillance Systems", Australian Centre for Visual Technologies, School of Computer Science, The University of Adelaide, 1997.
[6] Ralf M. Schreier, Albrecht Rothermel, "A Latency Analysis on H.264 Video Transmission Systems", International Conference on Consumer Electronics, January 2008.
[7] SMPTE 259M, "For Television — SDTV Digital Signal/Data — Serial Digital Interface", Society of Motion Picture & Television Engineers, 2008.
[8] SMPTE 292M, "1.5 Gb/s Signal/Data Serial Interface", Society of Motion Picture & Television Engineers, 2012
[9] ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding, Second Edition, October 2004.
[10] David W. Allan, Neil Ashby, Clifford C. Hodge, "*The Science of Timekeeping – Application Note 1289*", Hewlett Packard, 1997.
[11] Zhou Jin, Xiong Hong-kai, Song Li, Yu Song-yu, "Resynchronization and remultiplexing for transcoding to H.264/AVC", Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200030, China, Feb. 19, 2006.