2015 IEEE International Conference on Computer Vision Workshop

A Scalable Architecture for Operational FMV Exploitation

William R. Thissell[1], Robert Czajkowski[2], Frank Schrenk[1], Timothy Selway[1], Anthony J. Ries[3], Shamoli Patel[4], Patricia L. McDermott[5], Rod Moten[6], Ron Rudnicki[7], Guna Seetharaman[8], Ilker Ersoy[9], and Kannappan Palaniappan[9]
[1]Chenega Technical Innovations, Lorton, VA, USA, william.thissell@chenegati.com
[2]U.S. Army Intelligence and Information Warfare Directorate, Aberdeen Proving Ground, MD, USA
[3]U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, USA
[4]mail@shamoli.org, FL, USA
[5]Alion Science and Technology, Boulder, CO, USA
[6]DataNova Scientific, Baltimore, MD, USA
[7]CUBRC, Inc., Buffalo NY, USA
[8]Naval Research Laboratory, Washington D.C., USA
[9]Dept. of Computer Science, University of Missouri, Columbia, MO, USA

Abstract

A scalable open systems and standards derived software ecosystem is described for computer vision analytics (CVA) assisted exploitation of full motion video (FMV). The ecosystem, referred to as the Advanced Video Activity Analytics (AVAA), has two instantiations, one for size, weight, and power (SWAP) constrained conditions, and the other for large to massive cloud based configurations. The architecture is designed to meet operational analyst requirements to increase their productivity and accuracy for exploiting FMV using local cluster or scalable cloud-based computing resources. CVAs are encapsulated within a software plug-in architecture and FMV processing pipelines are constructed by combining these plug-ins to accomplish analytical tasks and manage provenance of processing history. An example pipeline for real-time motion detection and moving object characterization using the flux tensor approach is presented. An example video ingest experiment is described. Quantitative and qualitative methods for human factors engineering (HFE) assessment to evaluate cognitive loads for alternative work flow design choices are discussed. This HFE process is used for validating that an AVAA system instantiation with candidate workflow pipelines meets CVA assisted FMV exploitation operational goals for specific analyst workflows. AVAA offers a new framework for video understanding at scale for large enterprise applications in the government and commercial sectors.

1. Introduction

Armed forces and police agencies across the world have made a significant in-vestment in fielding a wide variety of full-motion video (FMV) electro-optical and infrared sensors to provide superior situational awareness and surveillance. These sensors collect an increasingly unmanageable amount of data, up to terabytes per hour from a single wide area motion imagery sensor. Even with conventional FMV sensors, the data being produced often far exceeds the manpower available to manually exploit the data.

While automated computer vision algorithms exist, the software solutions are often proprietary, fragmented, incompatible, and unable to work at scale on massive data sets. Motion imagery is a rapidly developing technology area that has the potential for providing unprecedented situational awareness and intelligence information to warfighters in the field. This potential as yet is far from being fully tapped. A great many motion imagery sensor systems are "stove-piped." That is, they are vertically but not horizontally integrated. Video from the field can only reach operators and analysts with immediate access to their control stations and operators with remote viewing receivers. Also, many motion imagery transmission data formats do not adhere to established standards. This greatly inhibits interoperability for data sharing and analysis since custom non-standard interfaces must be provided. In addition (but by no means intended to be inclusive of all issues), the volume of anticipated motion imagery data in future systems will, if not so already, over-whelm operators and analysts especially as new sensors and systems such as body-worn cameras are widely deployed [1]. Motion imagery data processing exploitation tools must be provided to allow operators and analysts to deal with the volume of motion imagery data and to provide actionable outputs. Organizations invest vast amounts of resources to address their unique imagery exploitation needs. It is very difficult to collaborate on these efforts or share new capabilities and algorithms that have been built by other groups.

1.1. High Level Requirements for a Full Motion Video (FMV) Computer Vision Analytics Assisted Exploitation Architecture

Most instantiations of a FMV computer vision analytic aided exploitation architecture will be as a component in an integrated system that will enable the information extracted

978-1-4673-9711-7/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCVW.2015.139

from the FMV data to be combined with information from other systems in order to produce knowledge and context [2]. Many electro optical infrared (EO-IR) FMV sensors for military and surveillance applications will have a size, weight, and power (SWAP) constrained computer system on a disadvantaged network for operators to control the sensors. The former, larger architecture, has forensic exploitation as a focus, whilst the latter has near real-time (*e.g.*, low exploitation latency relative to the frame rate of the sensor), typically defined as less than 200 ms for a complete computer vision analytic stream processing pipeline. This is the upper temporal latency limit where humans may maintain control of the sensor without experiencing a high cognitive burden and frustration level.

The operational goals for a FMV computer vision aided exploitation architecture, relative to conventional manual exploitation, are:

1. Reduce the human factors burden for exploiting FMV to extract information from geo-spatially disperse and temporally rare low observable activities of interest in a high clutter environment.
2. Increase the amount of FMV data that may be exploited per unit of human labor.
3. Increase the rate of actionable intelligence produced from FMV exploitation.
4. Facilitate the use of FMV exploitation into a multi data source exploitation environment.

The high level operational requirements above result in a series of implicit technical requirements that define the resulting material solution architecture towards:

1. An open architecture with open and extensible application programming interfaces.
2. An open and extensible data model for the information extracted from the FMV data.
3. An open compartmentalized computer vision analytic plug-in architecture to encapsulate unique intellectual property associated with many computer vision analytics.
4. Compliance with relevant industry standards (e.g., Motion Imagery Standards Board [MISB], NATO Standardization Agreement [STANAG], World Wide Web Consortium [W3C]), and industry best engineering practices (*e.g.*, Representational State Transfer [REST]).

2. A Scalable FMV Computer Vision Analytics Assisted Exploitation Architecture

The architecture developed to meet the requirements in Section 2.1 is the Advanced Video Activity Analytics (AVAA). The principal sub-components of this architecture are:

1. The Video Processing and Exploitation Framework (VPEF) with VBench and VProfiler.
2. The VPEF Distribution Server and Client (VDSC).
3. The Video Data Model (VDM).
4. The VDM Annotation Web Service (VAWS).

The sub-sections below describe AVAA and its sub-components in more detail.

2.1. AVAA

AVAA is instantiated in both a large, scalable cloud and a small system virtual machine architecture. Table 1 and Table 2 list the external software dependencies for both AVAA instantiations.

Figure 1 shows sub-component processing flow diagrams of the AVAA architecture for ingest (a); on demand computer vision analytic processing (b); and querying the enriched analytic results (c). In Figure 1a, upon ingest, the files are placed in a directory or the streams are passed directly to the VDSC. The original files and streams are written either to the Hadoop Distributed File System (HDFS) or the VM filesystem. The VDSC sends the files and streams to the AVAA clients for both transcoding to MP4 format, using H.264/AVC compression, compliant to MISB RP 0802.2, and for performing the ingest pipelines. The analytic data from the ingest pipelines are written to the VDM. The VAWS exposes the original videos, transcoded videos, and the analytic data to external systems and user interfaces. On demand processing (Figure 1b) is launched through the web service where the VDSC brokers the requested pipelines, unique pipeline configuration parameters, and selected files to the system clients. Querying the system (Figure 1c) occurs through VAWS, which executes W3C and OGC compliant SPARQL and GeoSPARQL queries to the VDM and returns the query results and associated videos.

2.2. VPEF Distribution Server and Client (VDSC)

The VPEF Distribution Server and Client is responsible

Table 1: AVAA Cloud Instantiation Software Dependencies

Component	Version
Accumulo	1.5.0
CentOS Linux 64 bit	6.5
Hadoop	2.0.0
httpd main proxy	2.4.10
JBoss	7.2.0
Puppet	2.7.22
RabbitMQ	3.2.2
Zookeeper	3.4.5

Table 2: AVAA Small system instantiation software dependencies

Component	Version
CentOS Linux 64 bit	6.5
Hibernate-release	4.3
Hibernate-spatial	4.3
JBoss	7.2.0
Postgis2_93	2.1
Postgresql	9.3
RabbitMQ-server	3.5

for managing the cluster of virtual machine and cloud nodes that are running VPEF. It is based on the RabbitMQ open source enterprise middleware message broker. FMV file and stream ingest requests are passed through RabbitMQ, where the requested VPEF pipelines to perform ingest and on-demand processing are managed.

2.3. Video Processing and Exploitation Framework

The agile, modular based architecture of VPEF greatly facilitates the construction of pipelines to achieve the operational objectives for computer vision analytic assisted FMV exploitation. VPEF is based on the open source GStreamer 0.10.35.0 baseline [3], with enhancements and plug-ins to process MISB standards compliant metadata, perform utility operations, sub-component APIs, and helper applications to facilitate the development, optimization, and configuration of complex stream pipelines, and create JavaBeans class compliant pipelines that are distributed among the nodes for execution via the VDSC. VPEF compartmentalizes computer vision analytics as plug-ins with standardized inputs (sinks) and outputs (sources). VPEF outputs (sources) from a pipeline include annotations, camera transformations, overlays, salient regions, objects, and image chips. VPEF enables standardization, integration, and parallelization of computer vision algorithms, thereby making them interoperable and testable.

Numerous automated computer vision algorithms, including FMV preprocessing, filtering, super-resolution, image-registration, metadata correction (such as camera pose), precise geo-registration, image quality and interpretability measurement, object detection and classification, object tracking, face detection and recognition, optical character recognition, scene classification, overlay masking, event and activity detection are being matured and integrated as plug-ins into VPEF. Macro level operations such as video shot detections, summarization, *etc.*, are realized using two or more of these modules, with localized shared-context buffers. The VPEF 2.0 Beta 3.6 release contains 935 plug-ins, and an additional 85 or so advanced computer vision and metadata processing plug-ins from commercial, government, and academic sources are under current development; but it is expected that not all of them will pass the rigorous software engineering tests to be verified and integrated into the AVAA environment.

VPEF uses industry and government accepted standards whenever possible, and extends these standards when required to meet performance objectives. Table 3 shows the current list of standards supported by VPEF.

VPEF has two important developmental applications, VBench and VProfiler. VBench is a GUI for creating, debugging, and tuning pipelines of plug-ins that extract information in FMV and enhance and correct the metadata.

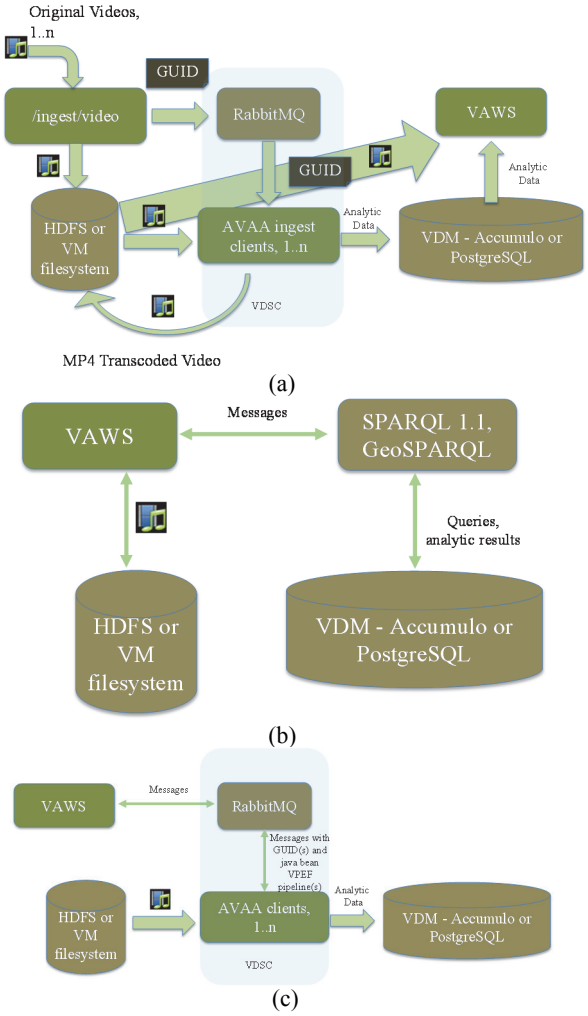

Figure 1: (a) Processing schematic for the ingest process; (b) Processing schematic for on demand FMV exploitation; and (c) Processing schematic for querying the analytic results.

VProfiler provides a GUI based tool for determining processing latency at the plug-in granularity. Figure 2 shows an example VBench display of a VPEF pipeline using the Flux Tensor and Blob Extractor plug-ins [4]. The flux tensor motion detection algorithm is a computational vision technique for robustly detecting moving objects in cluttered scenes using a temporal variation extension of the optical flow constraint equation. The implementation uses the trace of the flux tensor matrix and is computed using windowed integration kernels as,

$$Tr\_J_F = \int_{\Omega} W(\mathbf{x}-\mathbf{y})(I_{xt}^2(\mathbf{y}) + I_{yt}^2(\mathbf{y}) + I_{tt}^2(\mathbf{y}))d\mathbf{y} \qquad (1)$$

where I_{xt}, I_{yt} and I_{tt} are spatiotemporal partial derivatives of the image and $W(x\text{-}y)$ is the local smoothing kernel for the integration operator [5-7]. The flux tensor has been extended with a split Gaussian approach to detect very slow moving and stopped objects [8]. The flux tensor can also

be incorporated as part of motion plus appearance-based object tracking algorithms for aerial and ground-based FMV [9-12].

Both VPEF and VBench in combination produce a more powerful environment to rapidly prototype a task flow, using an ecosystem of algorithmically diverse, functionally similar modules. Higher level composition allows for competing methods to be concurrently tried, and reconciled downstream. VPEF is being used to develop a near real-time on the move multi-sensor exploitation system [13].

Figure 2: VBench GUI of a pipeline using the Flux Tensor [4 - 7] and Blob Extractor VPEF plug-ins. The real-time plug-in configuration properties are shown in the sliders in the upper left of the image. The upper right of the display is the graphical representation of the pipeline where the individual plug-ins are connected together. The three lower images show the intermediate and final results of the pipeline, which are updated on a per frame basis.

2.4. Video Data Model (VDM)

The design of the VDM facilitates the incorporation of FMV exploitation products in a multi-source exploitation computing environment.

The outputs from VPEF pipelines are written in AVAA to the VDM as Resource Description Framework (RDF) V1.1 triples [14]. The VDM itself is implemented either in an object - relational database, for the small computer system implementation, or as entries into an Accumulo column store for the cloud architecture instantiation. The VDM uses the motion imagery ontology, compliant to the web ontology language (OWL) specification, and is hierarchically mapped to the Actionable Intelligence Retrieval System (AIRS) as a domain-specific ontology [15]. The AIRS set of ontologies contains the Basic Formal Ontologies [16]. The taxonomy of the motion imagery ontology is shown in Figure 3. The brown lines pointing to the observer box in Figure 3 illustrate an important concept for a data model to support CVAs. CVA pipelines that are not robust, or with improperly adjusted object detection analytic parameters, or applied to ill-suiting FMV content, will potentially generate a large number of VDM entries of little or no value. The combination of CVA suggested, and analyst confirmed objects reduce this clutter to a potentially manageable level.

Table 3: Standards supported by VPEF

Standard	Description
MISB 0102.11	Security Metadata Universal and Local Sets for Digital Motion Imagery
MISB RP104	Predator Metadata Set
MISB RP210	SMPTE Metadata Set
MISB 0601.8	Unmanned Aerial System (UAS) Datalink Local Set
MISB 0602.4	Annotation Metadata Set
MISB 0801.5	Photogrammetry Metadata Set for Digital Motion Imagery
MISB 0901.2	Video-National Imagery Interpretability Rating Scale (VNIIRS)
NITF 2.1 PRI	Portable Reference Image
STANAG 4676 V1	NATO ISR Tracking Standard

2.5. VDM Annotation Web Service (VAWS)

VAWS is an extensible representative state transfer (REST) interface with an application programming interface. VAWS calls are the means by which external systems and user interfaces interact with the AVAA architecture. Table 4 lists the services currently available through VAWS.

3. Implementation Results

An instantiation of the AVAA architecture is sized according to the number of FMV streams it is required to ingest, and the amount of FMV to be stored, the peak number of analysts that will be using the system, and the workflows that will be typically executed. A thorough analysis of the architecture performance is required to develop the engineering data to support instantiation sizing, including individual plug-in latencies, pipeline latencies, CVA plug-in and composite pipe-line accuracies, errors, and confidence levels as a function of input data quality and FMV scene conditions. Thorough human factors

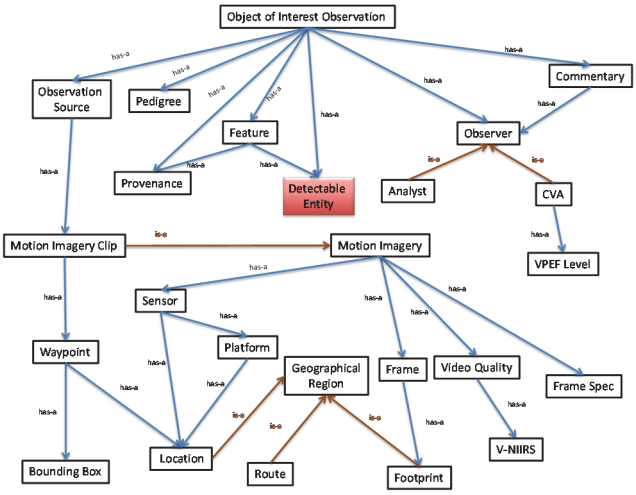

Figure 3: Taxonomy of the motion imagery ontology.

Table 4: VAWS Available Services

Name	Purpose
Video Links Service	Displays a list of links for all the transcoded videos available
Video Stream Service	Provides services for streaming videos
Camera Positions Service	Provides services for the camera positions of an ingested video
Video Quality Service	Provides services for the video quality of an ingested video
Video Frame Rate Service	Provides services for the video frame rate of an ingested video
Video Frame Size Service	Provides services for the video frame size of an ingested video
Video KLV Packet Service	Provides services for the Key-Length-Value (KLV) Packet of an ingested video
Annotation Service	Provides services for the annotations of an ingested video
Attribute Service	Provides services for the attributes of an ingested video
Comment Service	Provides services for the comments of an ingested video
Search Service	Provides services for searching analytics
Metrics Service	Provides services for collecting metrics on service usages
Classification Service	Provides services for classifications

engineering assessments of workflows are required for validating how a given instantiation satisfies the operational requirements and goals for FMV exploitation, described in Section 1.1.

Initial results of an example VPEF plug-in testing, scale testing ingest performance, and human factors engineering assessments are presented.

3.1. VPEF Plug-In Testing

The flux tensor based motion detection and morphology-based blob extraction plug-ins were benchmarked for performance within an aerial video processing pipeline as part of an object tracking analytics framework. Aerial imagery acquired from a moving platform first needs to be stabilized to compensate for background motion. An OpenCV stabilization module was included as part of the benchmarking pipeline. Table 5 summarizes the performance of each plug-in combination within the pipeline in terms of the latencies involved for each algorithm and the overall pipeline throughput or sustained framerate. The flux tensor and blob extraction plug-ins take about 5.1ms and 3.1ms respectively for small 352x240 pixel video frames and 24.6ms and 10.5ms for larger 720x480 frames. The stabilizer plug-in uses a buffer of ten frames and its latency is high at 682 ms and 1400ms for the small and larger video frames, respectively. The sustained framerate, once the pipeline plug-in buffers are filled, is near real time at about 25ms for small 352x240 pixel FMV frames and 51ms for larger 720x480 sized video frames; this includes all three plug-in modules for frame stabilization, flux tensor and blob extraction. The pipeline is essentially being executed in parallel with plug-ins distributed across multiple threads. So the overall performance of the pipeline is bounded by the module requiring the longest time to complete which in this case is the stabilization module.

3.2. Scalability Testing

Preliminary scale testing of the AVAA architecture was reported previously [17]. The results showed that the VDM is scalable to massive levels and query times for text annotations and date time groups were less than 0.5 seconds for an ingested data store of 2160 hours of FMV. Geo-spatial queries took substantially longer, up to about 3.75 seconds for 2160 hours of FMV, indicating that this was an area that warranted additional development work.

Scale testing of ingest performance is currently underway to quantify the hardware requirements needed by an installation in order to support a given FMV collection rate. A key consideration is to identify the hardware requirements such that the rate of ingesting and processing FMV is equal to or faster than the rate of collection.

The testing is identifying issues that affect ingest performance and accuracy, such as the optimum number of cores per VPEF ingest client, required time-out values, and minimum required batch sizes, as well as FMV file conditions that result in ingest plug-in failures.

The ingest pipelines used for this testing reads the Key-Length-Value (klvSpring) and video info (ImprovedVideoInfoProcessor) metadata, copies (fastUnreliableTranscoder) or transcodes the FMV file or stream to H.264 (Back-upMp4Transcoder) if it is not already encoded in this standard, computes the MISB 0901.2 values, and writes out this stream to the VDM (*i.e.*, input and enriched metadata) and HDFS or VM file system (*i.e.*, input data and H.264 MP4 file). These pipelines are wrapped as JavaBeans classes for execution within the VPEF clients.

The AVAA ingest nodes used for this testing are dual processor Intel Xeon E5-2670 (2.6 – 3.3 GHz, 20 MB L3 cache, 8 cores, 16 threads) 2U servers configured with 128 GB RAM and a NVidia TESLA M2090 Graphical Processing Unit (GPU) module. Table 6 shows the test configuration and example results from one of the ingest experiments. All of the pipelines used in this experiment did not utilize the GPU to accelerate the processing. The FMV resolution used for testing was 480p. A 30-minute timeout occurred 61 times during KLV extraction within the klvSpring pipeline. This represents 0.47% of the time in this experiment and is the principal cause for the large standard deviation in execution time for this pipeline.

Table 5: Flux tensor, blob extraction and stabilization motion analysis plug-in performance using an Intel Core i7-3960X 3.3 GHz processor with 12 cores and 32 GB of memory

352x240 Video Frames		Plug-In Performance (ms)				Pipeline Throughput (ms)			
Test	Count	Min	Max	Avg	Std Dev	Min	Max	Avg	Std Dev
Flux only	1192	4.767	5.324	5.114	0.067	7.770	36.164	25.366	3.494
Blob only	1188	2.810	3.808	3.125	0.086	7.705	35.998	25.190	3.494
Flux and Blob	1192	7.504	9.925	8.304	0.228	7.649	35.928	25.143	3.519
Stabilizer only	1196	281.069	817.298	682.151	59.831	0.604	35.968	25.105	3.994
Stabilizer, Flux and Blob	1192	397.203	828.795	692.580	54.363	7.692	36.083	25.175	3.490

720x480 Video Frames		Plug-In Performance (ms)				Pipeline Throughtput (ms)			
Test	Count	Min	Max	Avg	Std Dev	Min	Max	Avg	Std Dev
Flux only	1790	20.357	27.042	24.675	1.914	31.089	61.363	51.875	2.671
Blob only	1786	10.192	12.962	10.544	0.405	36.117	60.878	51.470	2.270
Flux and Blob	1790	35.266	38.887	36.204	0.478	35.812	61.000	51.426	2.287
Stabilizer only	1794	706.774	1779.029	1399.641	62.296	2.146	66.802	51.575	4.648
Stabilizer, Flux and Blob	1790	1180.032	1811.558	1431.875	47.485	35.805	60.972	51.504	2.295

3.3. Human Factors Engineering

A human factors engineering (HFE) process was developed to access and validate the ability of an architecture instantiation to achieve the operational goals described in Section 2 for desired analyst workflows [18]. A process of continuous evaluation is followed in the capability development. Insight gained from the HFE evaluations influence the developmental requirements and priorities for software development sprints.

The procedure includes quantitative, qualitative, and computational modeling measurements of analyst performance executing workflows. The quantitative measurements include mouse click analytics, eye tracking, physiological and electroencephalogram (EEG) measurements of neural activity. The qualitative measurements include four questionnaires and surveys. Figure 4 shows a photograph of the EEG and gaze tracking data collection station.

The participants were experienced imagery analysts with recent deployments conducting operational imagery analysis. The analysts had a mean of 9.85 (SD = 5.75) years of experience in the Imagery Analysis military operational specialty (MOS). Two assessments have been completed thus far, concentrating on the impact of a single video quality metric the VNIIRS (MISB 0901.2) [18]. The test scenario FMV data had various elements of military intelligence significance that the participants were asked to identify. Each analyst was given four scenarios to search through and given a short synopsis of the importance of the operational tasking for each scenario. They saw two scenarios in the baseline condition and two that were filtered using the MISB 0901.2 value set at a threshold. Four of the eight participants completed the scenarios while using the EEG and eye-tracking equipment.

Table 6: Example ingest experiment configuration and results

Input	**Value**		
AVAA Version	1.7.3-1		
Total Hours Ingested	1000.02		
Total Files Ingested	13017		
Number of Nodes Running VPEF_client	8		
Number of VPEF client per Node	4		
Output			
Test Duration, hours	37.49		
Effective Number of Streams	26.68		
Average Bit Rate, KB/s	6405.21		
FMV Ingest Statistics, seconds of VPEF client per file			
Java Beans	**Mean**	**Median**	**Std. Dev.**
ImprovedVideoInfoProcessor	1.99	0.88	3.00
fastUnreliableTranscoder	16.45	20.15	11.60
BackupMp4Transcoder	94.62	70.97	88.26
klvSpring	114.15	42.99	224.28
MISB 0901.2 v2.0.3-193	93.35	84.04	80.54

The EEG data analysis used the B-Alert classification model for cognitive work-load metrics [18-21]. This model utilizes a discriminant function derived from a large normative database and then refines this model for each subject based on their unique patterns of EEG acquired during three baseline task conditions. The user-specific workload classification model uses power spectral density (PSD) from 1-40 Hz in 1 Hz bins across all electrodes to produce a score indicating the probability of a high state of work-load.

Eye movement data were recorded using the Tobii X120 eye-tracker. Prior to testing each operator performed a nine point calibration. Eye tracking data were used to measure fixation and blink frequency as well as provide estimates of gaze distribution. Eye fixations were calculated using the ILAB toolbox and the Widdel algorithm [22, 23].

The baseline condition had a mean of 14.07 videos returned from each participant's search. Incorporating the video quality VNIIRS MISB 0901.2 metric improved the mean number of retrieved videos to 6.27, a reduction of 55 percent compared to the baseline condition. The VNIIRS filtered condition resulted in a reduction of 30 percent in the number of FMVs viewed compared to the baseline condition, 3.73 versus 5.36 mean FMVs, respectively.

The analysts found 40 percent more primary targets and 16 percent more total targets when using the MISB VNIIRS quality metric to filter videos selected for viewing, compared to not using this CVA module. Analysts took

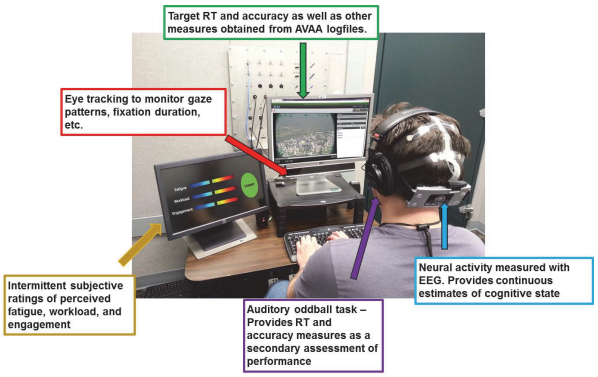

Figure 4: Experimental configuration. (Green) Primary task monitor used to view full-motion video from the AVAA software environment. (Red) Remote desktop eye-tracker provided ocular metrics during software interaction. (Yellow and Purple) A touch screen monitor was used as a response input device during an auditory probe task as well as a digital version of the NASA TXL. (Blue) Wireless EEG system used to derive neural estimates of cognitive workload and provide auditory-evoked potentials.

longer to find the primary targets when low quality videos are pre-filtered, by an average of three and a half minutes, compared to the base case without using VNIIRS due to the presence of more targets and faint targets; this is expected to be task and analyst dependent.

Figure 5 shows the cumulative sum of the standardized (*Z*-score) workload scores over the course of the test for two participants. Scores were standardized using the mean and standard deviation from both the Baseline and VNIIRS conditions. The data depict the workload fluctuations over time for each mission in each condition. Figure 6 shows the average eye blink and fixation frequency during the target search with much longer fixation times for videos that have been screened using VNIIRS. Figure 7 shows the gaze distribution from one subject during one of the missions using VNIIRS filtering. The gaze distribution map is generated from fixation duration.

4. Analysis and Discussion

The ingest experiment described in Table 6 sometimes exhibited a rare (0.47%) pipeline failure (30 minute timeouts) during klv extraction, which normally ought to execute quite quickly because it does not involve any complex mathematical operations. Ingest pipelines are required to be absolutely stable, because all subsequent analysis of the data relies on accurate ingestion. Ingest failures typically result in data that either is not ingested at all, or is ingested and may not be discoverable during subsequent annotation queries. Development is ongoing to optimize the multi-source multi-format media stream ingestion pipelines described in Table 6 and make them more robust. The results listed in Table 6 show that about 59 percent of the total median ingest times are spent in pipelines that involve extensive mathematical operations, the H.264 MP4 transcoding and the MISB 0901.2 VNIIRS

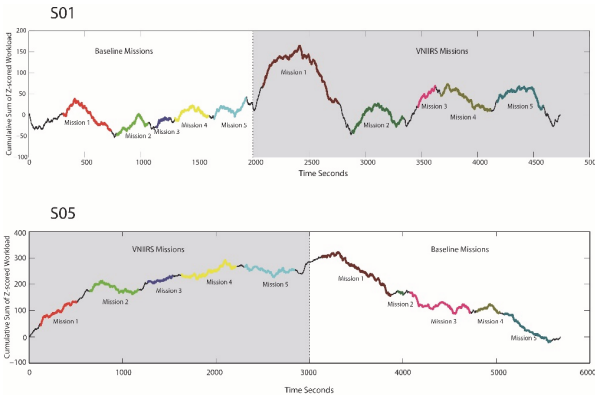

Figure 5: Workload classification estimates derived from EEG for each mission within the Baseline and with VNIIRS (shaded) conditions. The top panel shows data from observer S01 and the bottom panel represents data from observer S05.

quality computations. Both of these operations may be speeded-up considerably by utilizing many core GPU hardware at the ingest nodes in future versions of compute intensive VPEF plugins. Typical performance improvements by moving these operations from the CPU to the GPU (using CUDA) are expected to be a factor of 5 to 7.

The modular and scalable nature of this agile plug-in based CVA assisted FMV exploitation architecture provides a great deal of adaptability for meeting the operational requirements described in Section 1.1 for specific analyst workflows and sensor streaming data content. Optimization of the increased analyst productivity due to incorporating CVA exploitation modules requires an iterative process of pipeline design and refinement, CVA algorithm parameter tuning, and human factors design and engineering evaluation for specific analyst workflows using candidate pipelines on relevant sensor data scene content. The HFE procedure developed to support this process, described in Section 3.3, provides the quantitative data required to understand and estimate the labor required to

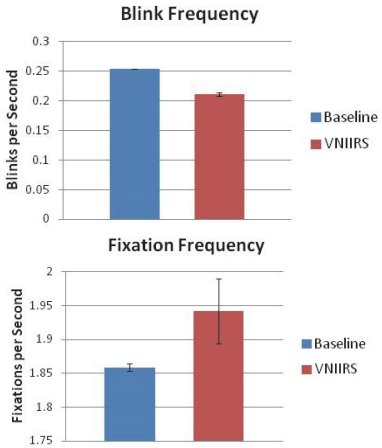

Figure 6: Average blink and fixation frequency during target search. Error bars equal standard error.

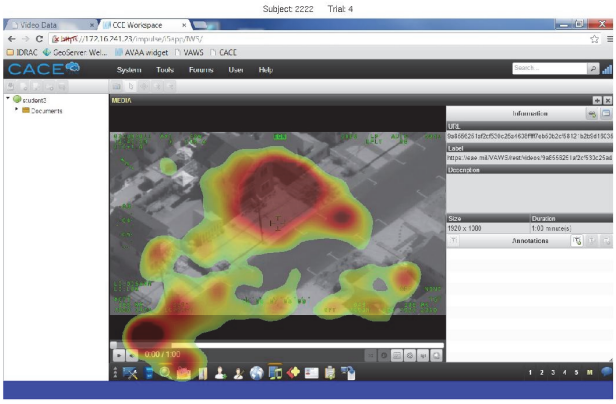

Figure 7: Fixation distribution from analyst S05 during the fourth mission in the MISB 0901.2 value filtered condition. The video frame depicted is for illustrative purposes only.

achieve FMV exploitation mission objectives, and assess the amount of improvement CVA assistance provides in meeting the operational goals described in Section 1.1. The example assessment described in Section 3.3, which included a single CVA, the VNIIRS MISB 0901.2 video quality metric, quantified a significant improvement in analyst productivity towards meeting the test mission objectives. The longer time the analysts took to find the primary targets when using the VNIIRS filtered condition is because the analysts found so many more targets in the higher quality videos, including hard to find targets. In contrast, the smaller number of targets found in the baseline condition were more obvious and quicker to locate. The eye-tracking data revealed that observers tended to make fewer blinks and more fixations on average on higher quality VNIIRS screened video with respect to the baseline. The gaze data shown in Figure 7 suggest this analyst primarily searched for targets in the center of the video feed and continuously monitored or interacted with the timing parameters of the video. The EEG data and behavioral performance indicated similar workload levels between the baseline and VNIIRS filtered conditions. The eye-tracking data suggest a trend toward higher cognitive workload in the VNIIRS filtered condition.

The HFE assessments highlighted numerous areas for system improvement that have been incorporated into the spiral software development cycles. An example are the improvements made to the architecture instantiation between the first and second assessments, spaced three months apart, that resulted in an increase in favorable ratings from 43 to 74 percent between the first and second evaluations. The cognitive impact of one software issue is shown in Figure 5, for participant S01, VNIIRS filtering, mission one, which shows a large, broad peak in cognitive workload when the system became unresponsive.

5. Conclusions

AVAA is a scalable open system and standards derived software ecosystem for computer vision analytics (CVA) assisted exploitation of FMV that will continue to evolve and mature as it is deployed. The ecosystem is targeted for highly scalable enterprise level video analytics with cloud computing and large scale data. The predecessors, such as VSIPL and the Interactive Image Spread Sheet (IISS), *etc*., focused on workstation based analytics with high performance computing [24-25]. The new AVAA software ecosystem for multimedia stream processing is designed to meet operational analyst requirements and increase their productivity and accuracy for exploiting FMV. The CVA encapsulated plug-in architecture protects unique intellectual property with standardized inputs and outputs and makes alternative algorithmic approaches readily testable. FMV processing pipelines constructed by combining these plug-ins to achieve an analytical outcome are modular and easily adapted using the GUI based VBench tool.

A quantitative and qualitative human factors engineering (HFE) assessment process is used for validating that a system instantiation with candidate workflow pipelines meets CVA assisted FMV exploitation operational goals for specific analyst workflows. This HFE assessment process involves discrete subjective NASA TLX ratings, augmented with multiple continuous objective measures including electrophysiology, eye-tracking, behavioral performance, and computational modeling. The measurement approach can be implemented in different settings and can be used to assess various cognitive states. A benefit of this approach is that it provides evaluators the ability to continuously track fluctuations in cognitive state during system interaction with higher temporal resolution than that provided by traditional self-assessment approaches. Evaluators may leverage this information to understand how system implementations may impact cognitive state and in turn operator performance within the system. The human factors evaluations of adding a single video analytics component for quality assessment using VNIIRS to screen videos, resulted in less work to be done per target, more targets found, and a longer search time to find the primary target when there are many targets to assess.

Acknowledgements

This research was partially supported by U.S. Air Force Research Laboratory (AFRL) under agreement AFRL FA875014-2-0072. Chris Geyer at EOIR assisted with integrating and benchmarking the flux tensor and blob extraction VPEF plug-ins. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of AFRL, I2WD, ARL, NRL, or the U.S. Government. This document is approved for public release, Distribution Unlimited.

References

[1] L. Miller, J. Toliver, S. Yanda, and C. Fischer, "Implementing a Body-Worn Camera Program: Recommendations and Lessons Learned," Office of Community Oriented Policing Services, Dept. of Justice, Washington, DC, 2014, http://www.nij.gov/topics/law-enforcement/technology/pages/body-worn-cameras.aspx.

[2] A. Cordova, A. D. Millard, L. Menthe, R. A. Guffey, and C. Rhodes, "Motion Imagery Processing and Exploitation (MIPE)" Santa Monica, CA: RAND Corporation, 2013. http://www.rand.org/pubs/research_reports/RR154. Also available in print form.

[3] http://gstreamer.freedesktop.org/

[4] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman, "Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking," *Journal of Multimedia,* Vol. 2, No. 4, Aug, 2007, pp. 20-33.

[5] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman, "Geodesic active contour based fusion of visible and infrared video for persistent object tracking," *8th IEEE Workshop Applications of Computer Vision (WACV 2007)*, pages Online, 2007.

[6] K. Palaniappan, I. Ersoy, G. Seetharaman, S. Davis, R. Rao, and R. Linderman, "Multicore energy efficient flux tensor for video analysis," *IEEE Workshop on Energy Efficient High-Performance Computing (EEHiPC)*, 2010.

[7] K. Palaniappan, I. Ersoy, G. Seetharaman, S. Davis, P. Kumar, R. M. Rao, and R. Linderman, "Parallel flux tensor analysis for efficient moving object detection," *14th Int. Conf. Information Fusion*, 2011.

[8] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split Gaussian models," *Proc. of IEEE CVPR Workshop on Change Detection*, 2014.

[9] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. Rao, and G. Seetharaman, "Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video," *13th Int. Conf. Information Fusion*, 2010.

[10] K. Palaniappan, R. Rao, and G. Seetharaman, "Wide-area persistent airborne video: Architecture and challenges," *Distributed Video Sensor Networks: Research Challenges and Future Directions*, Springer, pages 349--371, 2011

[11] I. Ersoy, K. Palaniappan, G. Seetharaman, and R. Rao, "Interactive tracking for persistent wide-area surveillance," *Proc. SPIE Conf. Geospatial Info Fusion II (Defense, Security and Sensing: Sensor Data and Information Exploitation)*, volume 8396, 2012.

[12] R. Pelapur, S. Candemir, F. Bunyak, M. Poostchi, G. Seetharaman, and K. Palaniappan, "Persistent target tracking using likelihood fusion in wide-area and full motion video sequences," *15th Int. Conf. Information Fusion*, pages 2420--2427, 2012.

[13] K. Green, C. Geyer, C. Burnette, S. Agarwal, B. Swett, C. Phan, and D. Deterline, "Near real-time, on-the-move software PED using VPEF," *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, 94540O, 2015.

[14] http://www.w3.org/RDF/

[15] R. Rudnicki, http://ncor.buffalo.edu/ontologies/AIRS_Ontologies.pdf.

[16] http://ifomis.uni-saarland.de/bfo/

[17] P. Heaney, W. Thissell, U. Patel, F. Schrenk, S. Patel, T. Selway, J. Hauris, and B. Swett, "Army Cloud Video Analytics for Automated Imagery Exploitation at Massive Scale," *Military Sensing Symposium*, Washington D.C., October 2012, www.dtic.mil.

[18] P. L. McDermott, B. M. Plott, A. J. Ries, J. Touryan, M. Barnes, and K. Schweitzer, K., "Advanced Video Activity Analytics (AVAA): Human Factors Evaluation," US Army Research Laboratory, ARL-TR-7286, May 2015, www.dtic.mil.

[19] C. Berka, D. J. Levendowski, M. M. Cvetinovic, M. M. Petrovic, G. Davis, M. N. Lumicao, and R. Olmstead, "Real-Time Analysis of EEG Indexes of Alertness, Cognition, and Memory Acquired With a Wireless EEG Headset," *International Journal of Human-Computer Interaction*, *17*(2), 151–170, 2004. doi:10.1207/s15327590ijhc1702_3.

[20] C. Berka, D. J. Levendowski, M. N. Lumicao, D. G. Yau, V. T. Zivkovic, and P. L. Craven, "EEG Correlates of Task Engagement and Mental Workload in Vigilance, Learning, and Memory Tasks" *Aviation, Space, and Environmental Medicine*, *78*(5), B231–B244, 2007.

[21] R. R. Johnson, D. P. Popovic, R. E. Olmstead, M. Stikic, D. J. Levendowski, and C. Berka, "Drowsiness/alertness algorithm development and validation using synchronized EEG and cognitive performance to individualize a generalized model," *Biological Psychology*, *87*(2), 241–250, 2011. doi:10.1016/j.biopsycho.2011.03.003.

[22] D. R. Gitelman, "ILAB: A program for post experimental eye movement analysis," *Behavior Research Methods, Instruments, & Computers*, *34*(4), 605–612, 2002. doi:10.3758/BF03195488.

[23] H. Widdel, "Operational Problems in Analyzing Eye Movements," In Alastair G. Gale and Frank Johnson (Ed.), *Advances in Psychology*, Vol. 22, pp. 21–29, 1984.

[24] R. Janka, R. Judd, J. Lebak, M. Richards, and D. Campbell, "VSIPL: an object-based open standard API for vector, signal, and image processing," *IEEE Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 949-952, 2001, doi: 10.1109/ICASSP.2001.941073.

[25] K. Palaniappan, A. Hasler, J. Fraser, and M. Manyin, "Network-based visualization using the distributed image spreadsheet (DISS)," *17th Int. AMS Conf. on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography and Hydrology*, pages 399-403, 2001.