

**Anomaly Detection for Data Reduction in an Unattended
Ground Sensor (UGS) Field**

by Laurel C Sadler, Robert Winkler, and Niranjan Suri

ARL-TR-7047

September 2014

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1138

ARL-TR-7047

September 2014

Anomaly Detection for Data Reduction in an Unattended Ground Sensor (UGS) Field

Laurel C Sadler, Robert Winkler, and Niranjan Suri
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE*Form Approved*
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) September 2014		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Anomaly Detection for Data Reduction in an Unattended Ground Sensor (UGS) Field				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Laurel C Sadler, Robert Winkler, and Niranjan Suri				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-CII-B 2800 Powder Mill Road Adelphi, MD 20783-1138				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-7047	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report describes the design and implementation of a data reduction technique for video sensors that are part of a larger unattended ground sensor (UGS) network. The data reduction technique is based on anomaly detection in full-motion video and subsequent statistical analysis techniques that allow the system to identify abnormal or otherwise interesting behavior that acts as a notification trigger. These techniques have been integrated into an existing distributed sensor framework at the US Army Research Laboratory (ARL) that is based on the Open Standards for Unattended Sensors (OSUS), developed in collaboration with the Defense Intelligence Agency (DIA). Furthermore, the statistical technique applied to this problem does not require any training data, which are often impractical in battlefield environments. Instead, it operates by being bootstrapped on mission profiles and templates established by the user and supplemented by incremental and online learning algorithms. Overall, these techniques combine to provide an effective approach to monitoring large distributed sensor fields without network and operator overload.					
15. SUBJECT TERMS Unattended ground sensor, UGS, anomaly detector, full-motion video, data reduction					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 20	19a. NAME OF RESPONSIBLE PERSON Laurel C Sadler
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-1221

Contents

List of Figures	iv
List of Tables	iv
1. Introduction	1
2. Anomaly Algorithm	2
3. Implementation in the ARL Experimentation Framework	3
3.1 Databases and Queries.....	7
4. Modification for UGS System Implementation	8
5. Conclusion and Future Work	9
6. References	11
List of Symbols, Abbreviations, and Acronyms	12
Distribution List	13

List of Figures

Fig. 1	Software implementation of the anomaly detector.....	3
Fig. 2	FPSS full-motion tracker.....	4
Fig. 3	Anomaly detection setup menu.....	5
Fig. 4	Anomaly viewer.....	6
Fig. 5	LINQ-to-SQL query.....	7
Fig. 6	UGS deployment in a network.....	8

List of Tables

Table 1	FPSS tracker outputs (these parameters are available in the database).....	4
---------	--	---

1. Introduction

A desire to provide increased situation awareness, coupled with the reduced cost and ready availability of sensors, has led to an increasingly instrumented battlefield environment. Unfortunately, deployment of large numbers of sensors does not necessarily correspond to an improvement in situation awareness. The tactical networks that interconnect these sensors and the Soldiers are often limited in bandwidth and unable to disseminate all of the data that can be potentially gathered from large numbers of sensors. Furthermore, even if the networks were capable of transporting all of the gathered data, the end result would be a deluge of information being delivered to the Soldier. Doing so would place a large cognitive burden on the Soldier and turn into a distraction from the actual mission at hand. Therefore, we need architectures and approaches that are sufficiently smart to process the sensor data, identify patterns that are important, and only convey such higher level information to the Soldiers. These data reduction techniques help address both the network congestion problem, as well as the cognitive overload problem.

This report describes the design and implementation of a data reduction technique for video sensors that are part of a larger unattended ground sensor (UGS) network. The data reduction technique is based on anomaly detection in full-motion video and subsequent statistical analysis techniques that allow the system to identify abnormal or otherwise interesting behavior that acts as a notification trigger. These techniques have been integrated into an existing distributed sensor framework at the US Army Research Laboratory (ARL) that is based on the Open Standards for Unattended Sensors (OSUS), developed in collaboration with the Defense Intelligence Agency (DIA). Furthermore, the statistical technique applied to this problem does not require any training data, which are often impractical in battlefield environments. Instead, it operates by being bootstrapped on mission profiles and templates established by the user and supplemented by incremental and online learning algorithms. Overall, these techniques combine to provide an effective approach to monitoring large distributed sensor fields without network and operator overload.

ARL previously developed an Image Enhancement Experimentation Framework¹ to evaluate various image processing research algorithms being developed at ARL. The Experimentation Framework integrates super-resolution, contrast, and deblur research algorithms as well as the Force Protection Surveillance System (FPSS)^{2,3} a full-motion video tracker developed at ARL, into a realistic environment simulating Army-relevant scenarios. It was designed to allow the operator the flexibility to run the image processing algorithms in a realistic environment, providing a streamlined, dynamic, reconfigurable workflow. It allows for easy modification of the parameters for each of the algorithms, offering the ability to pan, tilt, and zoom the networked cameras and provide the full-motion video, as well as annotates and saves the

enhanced images for later analysis. The ARL Image Enhancement Experimentation Framework application can be executed on a laptop PC using a traditional graphical user interface (GUI) or via an intuitive touch-based interface on the Microsoft PixelSense™ touch table and Microsoft Surface™. This report discusses the addition of a database to store the FPSS full-motion video tracker output data in conjunction with an anomaly or outlier algorithm to the existing FPSS full-motion tracker element, which currently exists in the ARL Experimentation Framework.

Anomaly detection is important to the user, in this case, to reduce workload by automatically flagging the relevant video segment based on the number of tracker detections outside of the statistically calculated normal range.

2. Anomaly Algorithm

The anomaly algorithm chosen to use in conjunction with the FPSS tracker is based on descriptive statistics. Descriptive statistics are widely used and are very useful in describing databases and the relationships between variables. They bring together large amounts of data so they can be comprehended with minimal effort.⁴

The algorithm detects outliers, or data points that are distinctively separate from the rest of the data. In this case, an outlier is defined as any data point more than 1.5 interquartile ranges (IQRs) below the first quartile or above the third quartile where the IQR is defined as the difference between the first quartile and the third quartile of a set of data. The first quartile is defined as the median of the data, which are data that are less than the overall median or a number for which 25% of the data are less than that number. The third quartile is defined as the median part of the data, which are data that are greater than the median or the number for which 75% of the data are less than that number.⁵

For example, the five-number summary algorithm is performed on the input dataset. The output of which is stored in the array of doubles stats as follows:

Stats[0] = the minimum of the data set

Stats[1] = the first quartile of which 25% of the data fall below

Stats[2] = the median or midway point in the data. 50% of the data fall below the median

Stats[3] = the third quartile of which 75% of the data fall below.

Stats[4] = the maximum of the data set.

The IQR is found by subtracting the first quartile from the third quartile. The lower and upper bounds for the data are defined as the IQR multiplied by 1.5 and subtracted from the first quartile, $\text{Stat}[1] - 1.5 * \text{IQR}$, and the IQR multiplied by 1.5 and subtracted from the third

quartile, $Stat[3] - 1.5 * IQR$, respectively. The outliers or anomalies are any data values that are above or below the upper and lower bounds.

The implementation of the algorithm in C# is shown in Fig. 1.

```
public static class AnomalyDetector
{
    public static bool[] Outliers(this IEnumerable<double> data)
    {
        double[] stats = data.FiveNumberSummary();
        double iqr = data.InterquartileRange();
        double lower = stats[1] - (1.5 * iqr);
        double upper = stats[3] + (1.5 * iqr);
        bool[] outliers = new bool[data.Count()];
        int i = 0;
        foreach (double d in data)
        {
            outliers[i++] = (d < lower || d > upper) ? true : false;
        }
        return outliers;
    }
}
```

Fig. 1 Software implementation of the anomaly detector

3. Implementation in the ARL Experimentation Framework

The integration of a database, an anomaly detector algorithm, and an anomaly viewer element as an enhancement to the existing FPSS full-motion tracker are described below.

The existing FPSS full-motion tracker is shown in Fig. 2. The detected moving objects are encompassed with blue, rectangular boxes. The corresponding track identification number is denoted in blue above the upper-left corner of the rectangle. The output data of this tracker are shown in Table 1. All of these data, in combination with the timestamp of the DVR image frame when the track is first detected as well as the timestamp of the DVR image frame when the track is no longer detected, are stored in the tracker database. These data are committed to the database when each track is no longer detected. The data are later queried to formulate the input to the anomaly detector algorithm.

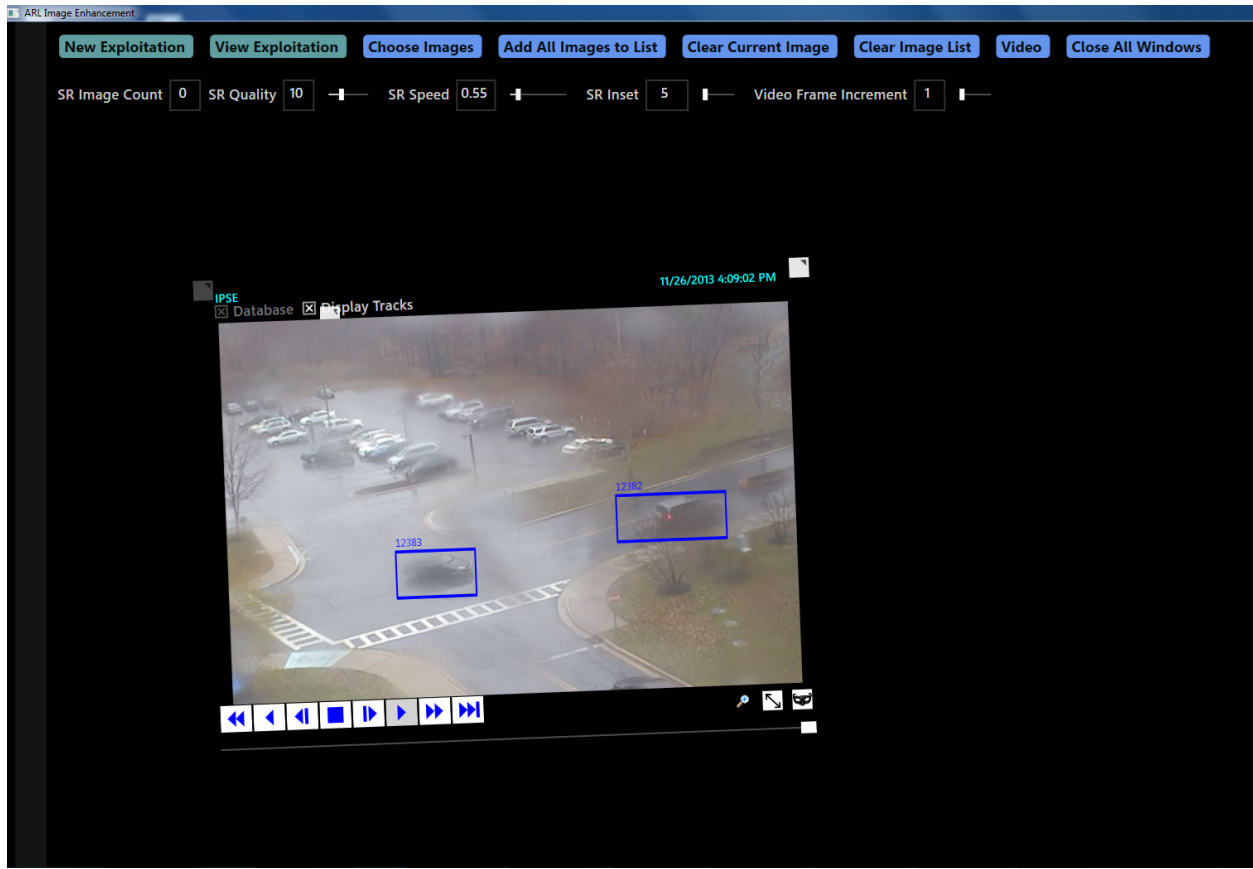


Fig. 2 FPSS full-motion tracker

Table 1 FPSS tracker outputs (these parameters are available in the database)

Tracker Output Data	Camera Output Data
HumanOrVehicle –Boolean: 0 for human, 1 for vehicle	Camera Name
Target ID	Camera Horizontal Field of View
Target Length – total number of frames in which the target is detected	Camera Vertical Field of View
Target Height (pixels)	Camera Frame
Target Width (pixels)	Camera Latitude
X-location –X pixel location on the image	Camera Longitude
Y-location –Y pixel location on the image	Camera Altitude
Target Size (pixels)	Camera Pitch
Target Status – for internal programming purposes	Camera Roll
Target Angle	Camera Heading
Target Average Height (pixels)	
Target Average Width (pixels)	
Target Average Confidence Value	
Target Confidence Value	
Target X-Velocity	
Target Y-Velocity	
Target-X Average Velocity	
Target Y-Average Velocity	

The input to the anomaly detection algorithm comprises an array of counts for each time segment. Each bin of the array contains a count of all tracks detected during a specified time segment on a specified day for a specified period of time. For example, one of the bins could contain the count of all tracks detected for each Thursday from 8:00–8:05 for a period of several weeks.

The user interface (UI) gives the user the ability to modify the above input parameters for the database query to acquire the input to the anomaly detector and also decide whether to use the anomaly detector for analysis purposes with a start and stop date and time or run the anomaly detector in real time given only a start date and time. The length of the time segment for the database query can be selected using the slider. The menu also allows the user to decide whether or not to compare all days equally or to compare only the specific day of the week. This is shown in Fig. 3.

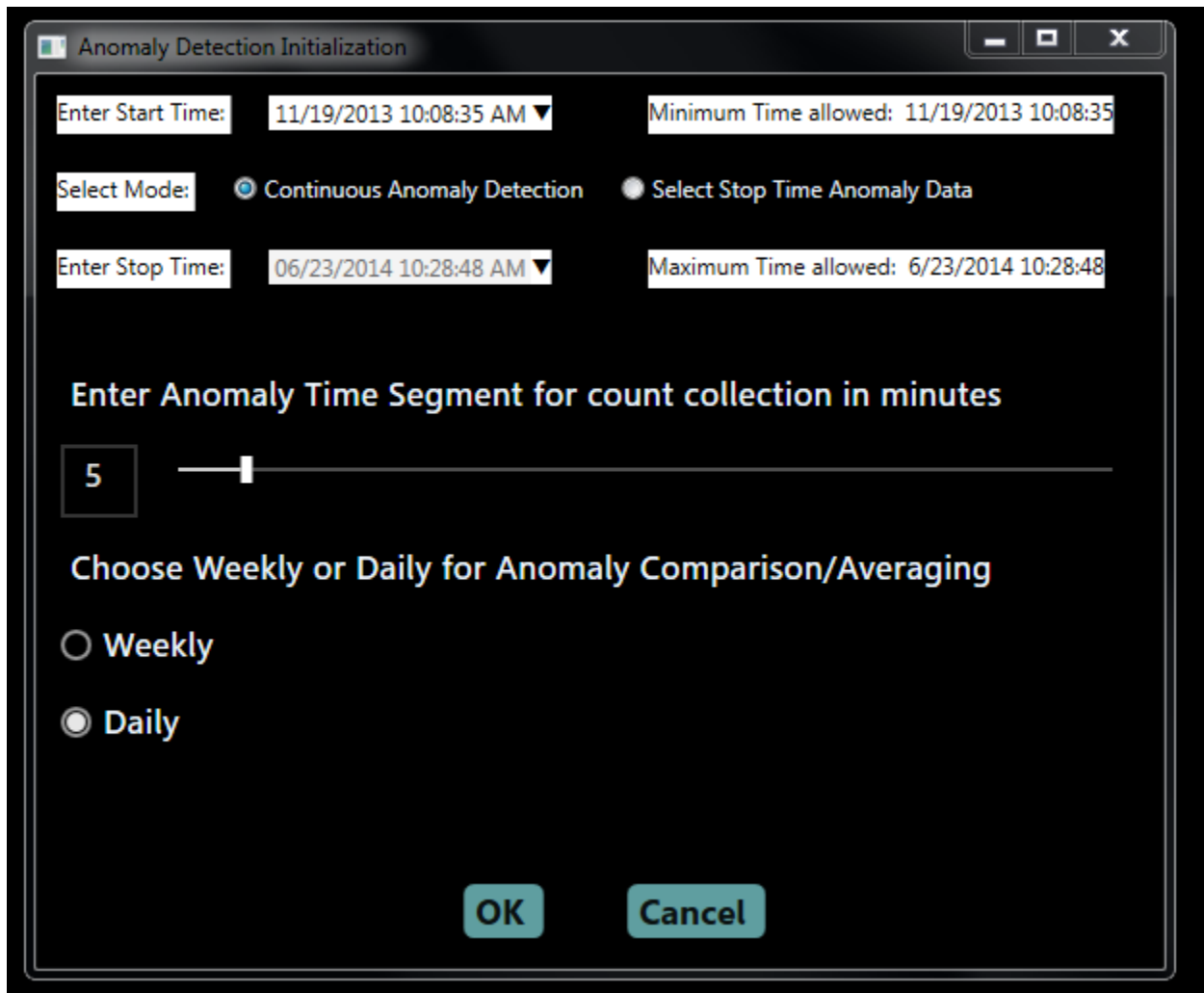


Fig. 3 Anomaly detection setup menu

The output of the anomaly detector is an array of binaries. Each time segment is considered either an anomaly time segment or not an anomaly time segment. The UI allows the user to visualize the anomalies for each period of time and to scroll through the stored DVR data, if available, to evaluate the video for the time segment of interest. The red lines on the slider denote the start of time segments that are considered anomalies. The corresponding detection or track count for that anomaly time segment is shown in red above the image. The blue lines on the slider denote the start time of all other time segments that were taken into consideration in the anomaly algorithm and are not considered anomalies. The track count is always shown; however, it is depicted in cyan when it is not considered an anomaly. Clicking on the slider line takes the video directly to that time segment. The DVR controls play, fast forward, etc., can be used to move through each time segment. This viewer is shown in Fig. 4.

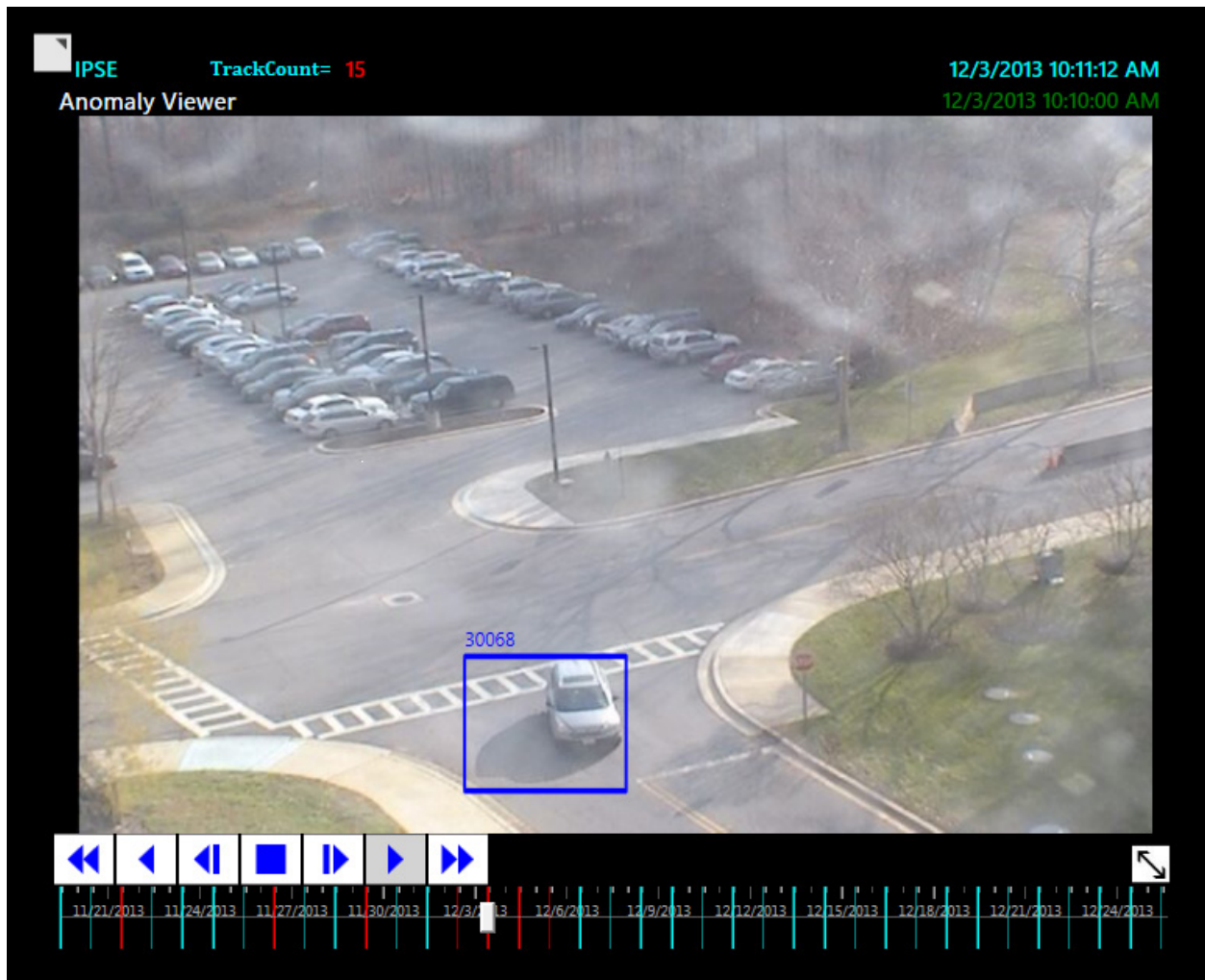


Fig. 4 Anomaly viewer

When operating in real time, the anomaly detector automatically updates the input to the anomaly detection algorithm, applies the algorithm, and presents the new anomaly data on the

anomaly viewer every 5 min or for the time frame specified by the user. In order to run the anomaly detector in real time, implementation of an algorithm that does not depend on laborious learning or training techniques is required. In the case of the current implementation, the longer the data have been collected and stored in the database, the better the algorithm will perform.

Notice that a detection or tracking box is shown surrounding the detection in the anomaly viewer. Each track for each individual frame is not stored in the database. However, the FPSS tracker with the identical setup parameters as were used during data collection is applied to the viewed video and displays detections boxes around the moving objects. This is done to aid the analyst in understanding what detections were most likely detected and counted during the initial data collection and store to the database.

3.1 Databases and Queries

A local SQL database was developed to store the tracker data parameters. An additional SQLite database was developed to store the DVR video frames.

All the data parameters from the FPSS full-motion tracker including the detection arrival time and the detection exit time are committed to the tracker database at the time of the detection exit. For this specific experiment, the database is queried to return the number of detections within a specified time frame. The query is shown in Fig. 5.

```
int count = (from t in tinfo.TrackDatas
             where (t.TimeAppears >= startTime && t.TimeAppears < endDatabaseQueryTime)
                || (t.TimeLeaves >= startTime && t.TimeLeaves < endDatabaseQueryTime)
             select t).Count();
```

Fig. 5 LINQ-to-SQL query

In this case, the anomaly would be the number of detections outside the statistically calculated normal range of detections for the specified time segment. Additional types of anomalies could be measured by varying the database query based on the different parameters that have been stored in the tracker database. These additional types of anomalies could include the size of the detection, the direction in which the detection is traveling, the velocity of the detection, or the location of the detection within the video frame.

The current anomaly detection algorithm depends on having baseline information regarding the vehicle counts in order to determine anomalous behavior. In an ideal environment, this would be learned by the system over time. However, in the target tactical environments envisioned for the developed system, there is typically no opportunity for the system to be in a training mode where it can train on observed data to establish a baseline. Any learning that occurs must be online, as part of the anomaly detection process. Therefore, the best likely approach is to pre-program the anomaly detector with some notion of expected behavior, which it gradually adjusts over time during operation. This initial baseline data are considered to be a template, which can be provided to the system for bootstrapping purposes. In the future, this template will be generalized

into a broader mission program that specifies expected behaviors as well as data that are of particular interest given the type of mission or activity being conducted.

4. Modification for UGS System Implementation

In the context of UGSs, anomaly detection is an effective approach to data reduction. As discussed in the introduction, increasing deployment of UGSs overloads both the network (which has to transport and disseminate the data) and the Soldier (who might be distracted by receiving large amounts of unnecessary data). Therefore, an UGS system would benefit from techniques such as anomaly detection, where the system could potentially filter out large numbers of unimportant detections. Furthermore, detections deemed to be anomalous could be flagged as high priority events and disseminated to the interested users, which increases the likelihood of the users paying attention to the anomalies.

Figure 6 shows an example deployment of multiple UGSs in a tactical network. This example shows two clusters of UGSs with four sensors in each cluster. The sensors within each cluster are interconnected via a network that allows them to exchange information (shown with solid lines in the diagram). Typically, this would be a mobile ad-hoc network (MANET). The clusters are connected to other nodes, which could be consumer nodes (e.g., a dismounted Soldier or a tactical operations center [TOC]) or could be harvester nodes. Harvester nodes typically enable the dissemination of data from the sensor network to other consumers that do not have a direct link. Examples of harvester nodes are unmanned aerial vehicles (UAVs), ground robots, or other mobile networked nodes such as vehicles in a convoy. In this deployment scenario, the network links that interconnect consumers, harvesters, and UGS clusters are shown with dotted lines in the diagram. These links could be intermittent and low bandwidth, which typically should not be overloaded with unimportant data.

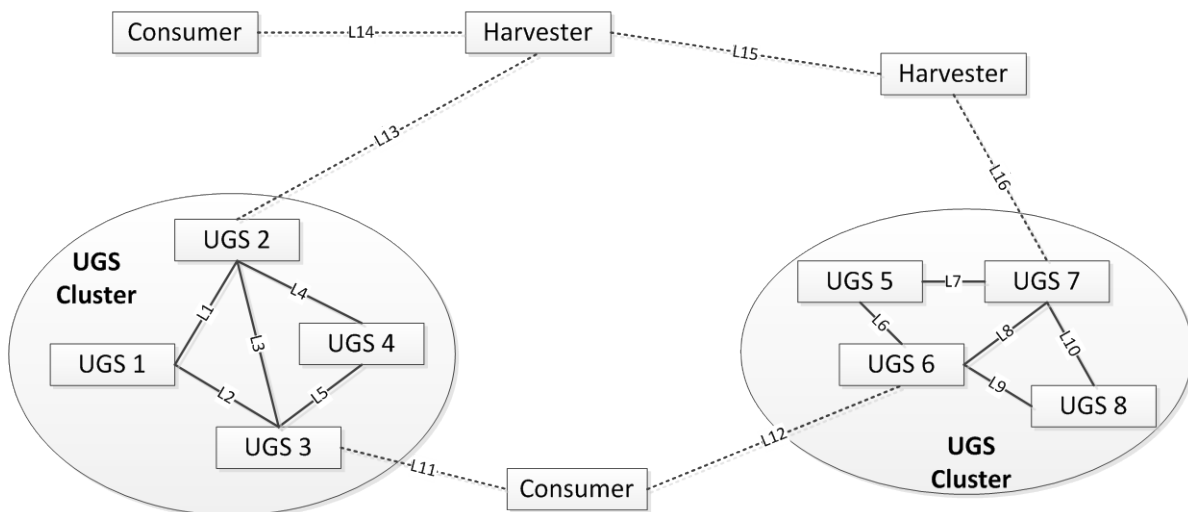


Fig. 6 UGS deployment in a network

Integrating anomaly detection into such a deployment scenario allows the data being generated at the UGS nodes to be locally filtered (or flagged as important) prior to being disseminated to the consumers (or harvested). For example, if the harvester is a UAV, the expectation is that it will be in intermittent contact with the UGS cluster. Therefore, it would be better to exfiltrate data that indicate anomalies from the UGS cluster as opposed to including all of the data (especially if the contact period between the harvester and the UGS cluster is short and not all of the data can be offloaded).

The anomaly detection algorithm can run at different levels of scope. In the simplest case, the algorithm can operate independently at each of UGS system, examining local data to determine anomalies. The next level of scope would involve UGSs within a cluster performing anomaly detection in a cooperative manner. For example, consider a scenario where UGS 1 and UGS 2 are at two ends of a road passing through a region. If a vehicle passes by UGS 1 and continues at an expected rate of speed and passes by UGS 2, that could be considered normal behavior and may not be flagged as an anomaly. On the other hand, if it passes by UGS 1 and does not pass by UGS 2 (or takes longer than usual), that could be considered an anomalous event. This would require that the detection algorithms be able to coordinate observations over the local MANET that interconnects UGSs within the cluster.

Extrapolating from that scenario, there could be anomaly detection that happens over successively larger scopes—for example, across multiple UGS clusters or even coordinated by consumers (i.e., analysts at the TOC). The architecture being developed as part of OSUS is sufficiently flexible to support these multiple modalities of exploiting anomaly detection within networks of UGSs.

5. Conclusion and Future Work

This report has described an anomaly detection algorithm applied to tracks generated by analyzing full-motion video data from a live sensor feed. The anomaly algorithm could also be applied to other data than FPSS track data from streaming video. It could be integrated into a tripwire sensor detecting the number of trips within a specified time frame. Rather than send every image generated when the sensor is tripped, it could down select and transmit images generated within the specified time frame if the time frame is flagged as an anomaly. The tripwire sensor or tracking sensor might also send an event message specifying that there are no data generated during a time period that expects to have significant amounts of activity, as the lack of activity can be an equally important anomaly.

Anomaly detection can be used with various types of data collected by multiple types of sensors. It can be used as a filter to a sensor to reduce the transmission of data reducing bandwidth or battery power consumption. It can also reduce the cognitive load on the operator by reducing the

volume of data or add value to the data by flagging or highlighting more significant data, thereby reducing the chance of the operator disregarding it with the rest of the data. As discussed above, the anomaly detection algorithm can also generate additional information beyond the scope of the sensor by sending an event message when there is a lack of information as well as when there is an overload.

It is the intent of ARL's Battlefield Information Processing Branch to integrate these types of anomaly algorithms into an UGS system in fiscal year 2015 (FY15) and perform experiments to determine its usefulness in these and other scenarios.

6. References

1. Sadler L. US Army Research Laboratory Image Enhancement Test Bed User's Manual. Adelphi (MD): US Army Research Laboratory (US); 2013. Report No.: ARL-TR-6512.
2. Chan AL. Force Protection Surveillance System: Algorithm and Performance. Adelphi (MD): US Army Research Laboratory (US); 2010. Report No.: ARL-TR-5322.
3. Chan AL. A Robust Target Tracking Algorithm for FLIR Imager. Proceedings of the SPIE. 2010;7696:769603-769603-11.
4. Carroll S. Dissertation Statistics Website [accessed 08/04/2014. <http://www.dissertation-statistics.com/descriptive-statistics.html>].
5. Simmons B. Mathwords: Terms and Formulas from Algebra I to Calculus, 28 July 2014 [accessed 08/04/2014. <http://www.mathwords.com/o/outlier.htm>].

List of Symbols, Abbreviations, and Acronyms

ARL	US Army Research Laboratory
DIA	Defense Intelligence Agency
FPSS	Force Protection Surveillance System
FY15	fiscal year 2015
GUI	graphical user interface
IQRs	interquartile ranges
MANET	mobile ad-hoc network
OSUS	Open Standards for Unattended Sensors
TOC	tactical operations center
UAVs	unmanned aerial vehicles
UGS	unattended ground sensor
UI	user interface

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

5 US ARMY RESEARCH LAB
(PDF) RDRL CII B
LAUREL C SADLER
ROBERT WINKLER
NIRANJAN SURI
MARK THOMAS
JESSE KOVACH

1 US ARMY RESEARCH LAB
(PDF) RDRL CII
BARBARA BROOME

1 US ARMY RESEARCH LAB
(PDF) RDRL CI
JOHN PELLEGRINO

1 US ARMY RESEARCH LAB
(PDF) RDRL SES
JOHN EICKE

1 US ARMY RESEARCH LAB
(PDF) RDRL SES-A
NINO SROUR

1 US ARMY RESEARCH LAB
(PDF) RDRL SES-E
RAGHUVVEER RAO

1 USARMY CERDEC (US)
(PDF) RDER-IW-IWP
ALAN HANSEN

1 USARMY CERDEC (US)
(PDF) RDECOM-CERDEC
DANIELLE DUFF

1 UNAFFILIATE
(PDF) LARRY TOKARCIK

INTENTIONALLY LEFT BLANK.