[1]

Operations and a Surveillance Scripting Language

Natalya SEGAL

***Abstract*—This paper was originally submitted to Xinova as a response to a Request for Invention (RFI) focusing on new Decision and Response Techniques for Security Applications. In this paper, a powerful tool that allows for efficient operations on large sets of dynamic surveillance information is proposed.**

In more detail, this describes the implementation of a scripting language that can facilitate responses and reduce times for actionable security challenges.

I. INTRODUCTION

It is challenging for a human operator to extract actionable data from a large amount of visual information. This solution report introduces a powerful tool that allow for efficient operations on large sets of dynamic surveillance information (video, voice, actions of the agents in place, etc.).

II. SUMMARY OF THE INVENTION

To address the challenges, I propose

1. Introducing a simple scripting language tailored for surveillance to enable the human operators to operate with large sets of video streams and the data extracted from those streams.

2. Enabling actions concatenation, meaning that an action can be performed on the results of the previous action/s.

3. Enabling mixing logical operations on the data with physical world operations commands (given to people via devices, to robots, flying cameras, etc.).

III. DETAILED EXPLANATION

The technical description includes the following;

1. The language is interpreted per line of code.

2. Results of every code line are presented and visualized to the operator

3. Results of every line of code can be used in the subsequent lines, not only the last line result. This allows for a tree-like logic concatenation.

4. The language is object oriented with predefined objects. Objects and their properties and methods are matching the surveillance system capabilities. Ideally (if the systems are standardized), the interpreter decides on it's objects / methods / properties based on the surveillance system configuration.

5. The language is optimized for massive data operations on the metadata extracted from the video streams: searches, segmentation, etc.

6. The language allows for real live operations, such as: sending more drone-mounted cameras, etc. The language allows for sending command messages to groups of people (e.g. police). The IDE for the language requires human confirmation before each step that is translated to a real-world action or a command message.

7. Frequently used combinations of operations are auto-suggested as new methods (functions).

The types of operations that this language supports may include map, filter, segment, action, show, and many others.

A. Map – maps each element from an input to an element or a group of elements from another type in an output. Example 1: map from a list of video streams to a list of their GPS locations. Example 2: for each threat in a list, give a list of cameras (video streams) that provide video streams including this threat. Here there can be more than one camera per threat.

B. Filter – filters the list of input data. Example: find threats above a particular severity threshold

C. Segment – segment the input data according to a list of parameters. The parameters may be provided by an operator (if there are a few parameters he/she is interested in, example: segment the threats into groups according to the weapons they hold /not hold). The parameters may be taken from one of the previous operations, for example: an operator extracts parameters of interest from the recorded data of a previous attack that looks similar and then segments according to those parameters. The parameters may be taken from a system configuration (upon a setup, the system is fed with knows parameters from previous experience and save this information on parameters in a data base). Example: an operator doesn't have a particular idea or comparison with a previous attack and asks for segmentation of threats according to the system parameters, for instance this segmentation may reveal that a group of threats (people) approaching a stadium from different directions are closely related (studied together, connected in social networks, live in the same neighborhood, attend the same religious places, etc.)

D. Action – perform a real-life action on the results of the previous operations. For example, send more drone-delivered cameras to a list of GPS locations from the previous calculation. Example2: send a particular message for all the human agents in the field that are on the list. This list can be either a fixed list or a result of any previous operation, for instance, search (filter + map) for every human agent from a particular group that is in a particular area.

E. Show – visualize one of the results (lines are numbered, so result of each performed lines of code can be addressed by number, or simply selected by touch) in a particular way. The ways of visualization include at least: video and map. Alternative visualizations may include augmented videos (Example: show a video with pointed out visual details, like weapons, luggage, etc. to point out an operator attention,), charts, graphs, etc..

Examples of the possibilities opened for security operators by the proposed solution:

1. A scripting **language** to operate with a large number of streams. The language can be defined to perform: search, segmentation, selection on the large group of streams; operations on the results of the operation that return a stream/group of streams, like: get location, get metadata of the stream; and actions in the physical world on the results of the previous operations, like: search for nearby objects by location, for instance search for cameras nearby, or add drone-mounted cameras to the designated location, facing the designated threats from particular angles.

2. A **concatenation of actions** performed by an operator/s. Simple concatenation: Every next action is performed on the output of the previous one. Another option for concatenation is performing next action on a results of one of the previous operations (not necessarily the last one) or on a combination of results of several previous operations.

3. **Segmentation** of the streams into several groups according to parameters, like: armed/not, with what weapons, number of people detected as threats, among them their zip code, place of birth, study, work, religion, football club, etc.
Looking for a group of threats with similarities (possibly in different places, in different video streams), occurring approximately at the same time.

4. Saving data of the attacks. Then **extracting data from a previous attack** and looking for similar parameters in the on-going streams. The attach ID (identifier) is given by an operator, by id the system extracts recorded video streams over the time of that attack with the recorded metadata. Results may look like: in stream 5,8 and 9 there are people that are defined as threats from the same birth town. Their connection to the people from the previous attack will be investigated (searching online as well as searching surveillance data bases)

5. An operator can ask to find similarities/differences in a group of on-going threats that are found in some video streams. sample answer: 2-3 people, guns, suspects living in the same zip code, etc.

6. Providing operators with **shortcuts for the most commonly asked questions types**. For example: the most common questions may be asked by pressing icons (half-transparent, presented on the video stream as a layer). The icons presentation method may be taken from mobile apps. There is one half-transparent button over a video stream, when pressed, additional buttons of a small menu appear, also as partially transparent icons.

7. An operator can also ask more complicated questions using a simple scripting language (It is reasonable to assume that with advancement of NLP, later those questions may be asked using a natural language)

8. An operator can perform a **search** over a stream of meta information that includes at least a number, type

and severity of threats in the stream

9. On a result of the search (stream/s) for any manually selected object on the stream, an operator can request to **track** this object/person

10. On any result of the search (streams from particular subset of cameras) or any manually selected subset of cameras, an operator can request **more information from the locations covered by those cameras** or a location near it. One of the way to provide this information may be by adding dynamic cameras, delivered by drones, automatically giving those drones GPS locations and the objects/people to track.

11. An operator can **request alerts** (colour, voice, text, etc.) on particular type of events.

12. Operator can request to **visualize** which cameras (from a previous filter) cover what area with conus images going from a camera on a map

13. Showing all the video streams that cover a particular point/s on a **map** upon pressing on this point on a map. For example, a set of GPS locations with the detected threats.

14. On a result of the previous action (a list of cameras that cover location of the threats) an operator can perform a next operation, for example displaying the video streams from those cameras according to one of the previously suggested algorithms (according to severity, in a loop, etc.). A powerful option of **concatenating operation** allows to perform the next operation on the results of a previous filter.

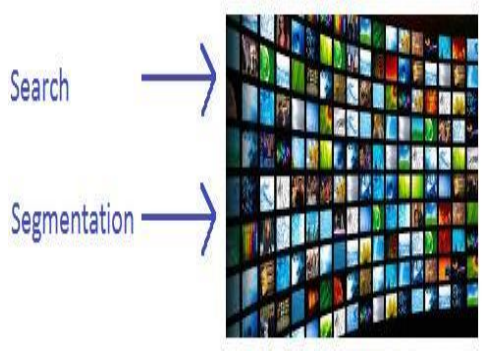

Fig. 1. Extract actionable data

IV. CONCLUSION

Comparatively, more work can be performed by less people, enabling operations that cannot be performed manually, enabling those operations to be performed dynamically in a time of attack, and not just as post-mortem analysis.

V. ACKNOWLEDGMENT

This research was originally submitted to Xinova, LLC by the author in response to a Request for Invention. It is among several submissions that Xinova has chosen to make available to the wider community. The author wishes to thank Xinova, LLC for their funding support of this research.

More information about Xinova, LLC is available at www.xinova.com.