VIVA-uOttawa / CBSA at TRECVID 2012: Interactive Surveillance Event Detection [*]

Chris Whiten, Robert Laganière, Ehsan Fazl-Ersi, Feng Shi
VIVA Research Lab
University of Ottawa, Ottawa, Canada
{cwhit025, laganier, efazlers}@uottawa.ca

Guillaume-Alexandre Bilodeau
LITIV Lab
École Polytechnique de Montréal, Montréal, Canada
guillaume-alexandre.bilodeau@polymtl.ca

Dmitry O. Gorodnichy, Jean-Philippe Bergeron, Ehren Choy, David Bissessar
Video Surveillance and Biometrics Section, Science and Engineering Directorate
Canada Border Services Agency, Ottawa, Canada

Abstract

We present an interactive video event detection system for the TRECVID 2012 Surveillance Event Detection (SED) task [16]. Inspired by previous TRECVID submissions, the underlying approach is built on combining automated detection of temporal regions of interest through the extraction of binary spatio-temporal keypoint descriptors in observed video-sequences (Video Analytics module), and efficient manual filtering of false alarms through the use of a custom-designed graphical user interface (Visual Analytics module). We make the automated detection of temporal regions of interest feasible by using efficient binary feature descriptors. These descriptors allow for descriptor matching in the bag-of-words model to be orders of magnitude faster than traditional descriptors, such as SIFT and optical flow. The approach is evaluated on a single task, PersonRuns*, as defined by the TRECVID 2012 guidelines. The combination of Visual Analytics and Video Analytics tools is shown to be essential for the success of a highly challenging task of detecting events of interest in unstructured environments using video surveillance cameras.*

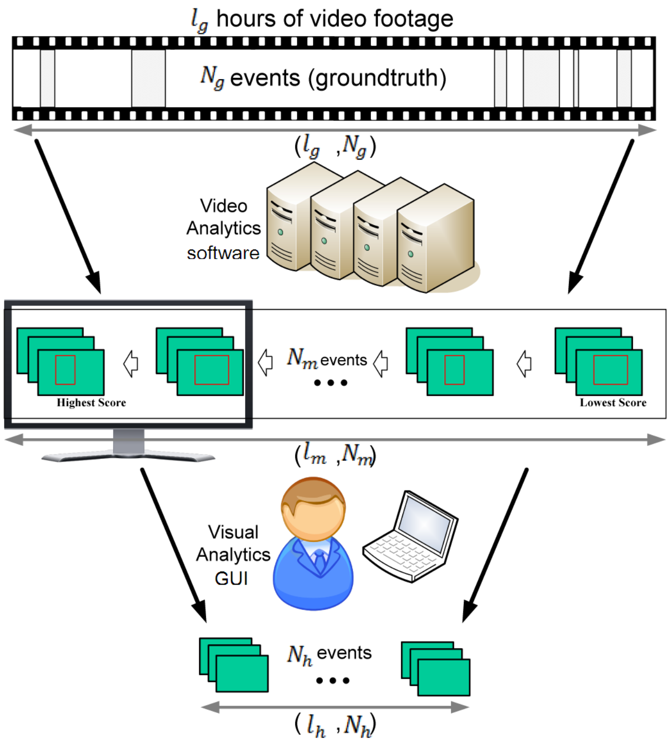
1. Introduction

1.1. Operational need

As a result of the increasingly growing demand for security, many countries have been deploying video surveillance (CCTV) systems as an important tool for enhancing preventive measures and aiding post-incident investigations. Within the Canadian government, many federal departments heavily use CCTV systems, including the Canada Border Services Agency (CBSA) who sees video surveillance as a key technological element in protecting the country's borders as well as in facilitation of travel of legitimate people and goods through the border.

When deployed, CCTV systems are used in either of two modes of operation: a) Live mode (or real-time monitoring), and b) Archival mode (or post-event analysis through recordings). While evaluating the utility of these systems, it has been realized that currently deployed surveillance systems are not fully efficient for either modes of operation. In the real-time monitoring mode, the problem is that an event may easily pass unnoticed due to false or simultaneous alarms and a lack of time required to rewind and analyze all potentially useful video streams. In archival mode, video data storage and manageability is the problem that complicates the efficiency of post-incident investigation the most. Due to the temporal nature of video data, it may take very long for a human to analyze it. A way to resolve these problems is seen in designing automated video analytic tools, which would help a security or investigation officer to do the

[*] This work is done within the CBSA-led PSTP BTS-402 project PROVE-IT(VA) funded by the Defence Research and Development Canada (DRDC) Public Security Technical Program (PSTP). Project lead: Dmitry Gorodnichy (email: Dmitry.Gorodnichy@cbsa-asfc.gc.ca).



Data	length (L)	# events (N)
original video	Lg=15hours	N_g=107 (ground truth)
detected by computer	L_m=2016*2secs	N_m=2016 (TP+FP)
detected by human	L_h =25mins	N_h=15 (6+9)

Figure 1. Formalization of the video analysis problem: Given ℓ_g minutes of video footage, which contains N_g events (ground truth), the objective is to design such Video Analytics software and such Visual Analytics interface that would allow a human operator to detect the maximum number of events N_h in not more than L_n minutes (operational contraint). The table shows the length of video data and the number of events used/obtained/estimated in the current SED task: of original video data, data detected using Video Analytics (by computer) only, and data detected using both Video Analytics and Visual Analytics (by human).

work more efficiently [11]. The design of such a tool tested on TRECVID SED video-sequences captured at Gatwick airport from HomeOffice iLids dataset is presented in this paper.

1.2. Formalization of the problem

The TRECVID 2012 Surveillance Event Detection (SED) task can be formalized as as follows (see Figure 1). Given L_g minutes of video footage, which contains N_g events (ground truth), the objective is to design such a technology that would allow a human analyst to detect the maximum number of events N_h within L_h minutes.

Such a presentation very well represents the actual operational constraint of many agencies, which is the limited time / human resources available to process the data. It also highlights an important dual computer-human nature of the solution defining its two main components: one – executed by a computer, which uses significant machine power and time to search for events of interest in a long video sequence in order to extract the clips corresponding to potential events of interest (N_m), possibly along with their associated metadata; and the other – executed by a human, who further processes the data obtained by a computer in order to detect the true events (N_h) within a given amount of time (L_h).

In particular, it is realized that, regardless of the quality of a video recognition (Video Analytics) algorithm, it will not be able to robustly detect events of interest without generating false alarms, the number of which can be very large in unstructured surveillance environments, such as those used in the SED task.

Therefore our solution is sought in combining *Video Analytics* advances with the recent advances in *Visual Analytics*, the science of using human's visual recognition power for efficient processing of data through the design of problem-tailored graphical user interfaces. In this way, the number of machine-generated alarms N_m, which would take otherwise ℓ_m minutes to view, can be further reduced to N_h alarms which can be viewed L_h minutes.

Specific to the TRECVID 2012 SED task and the results we obtained (see Figure 1), with the use of our Visual Analytic interface we are able to detect $N_h = 15$ events in $L_h = 25$ minutes from the N_m=2016 suspected events detected by our Video Analytics algorithm, the total length of which is $L_m = 2012 * 2$ seconds (each event is extracted as 2 sec clip). While machine detected events N_m consist of both true and false positives, with the majority (over estimated 95%) being false positives, the human detected event N_h should [1] predominantly be true positives.

1.3. Previous work

There has been an extensive amount of research in event detection that is applicable to surveillance environments. Laptev *et al.* [14] use 3-dimensional Harris corners as a spatiotemporal feature detector, where the third dimension is time. An image patch is deemed to be a keypoint when sufficient change is present in both spatial dimensions and the temporal dimension. This method generally detects keypoints with a strong reversal in the motion direction, which returns very few keypoints in many domains. Dollar *et al.* [6] propose a method using Gabor filters in the temporal domain that tends to return many more keypoints, by evoking a strong response when varying image intensities contain periodic frequency components. Gilbert *et al.*

[1] As presented in later sections, this is not always true, as in our experiments only 6 of our $N_h = 15$ manually detected events appeared to be true, with 9 of them being false, which is likely attributed to the fact that the length we chose for extracted clips (2secs) was not large enough to distinguish fast walk from running (especially that of children).

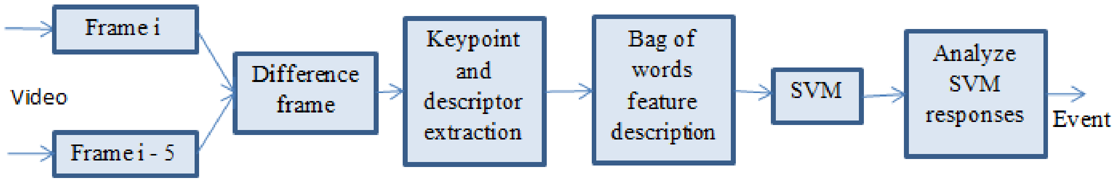


Figure 2. The flow of our event detection system. A pair of frames five (5) frames apart are joined to create an image encoding both motion and appearance. This new image is used in a bag-of-words model to transform keypoint descriptors into SVM features, which are then used to compute the likelihood of an event occurring at that frame.

[9] also extend 2-dimensional features into 3 dimensions. This is achieved by partitioning the 3D space into 3 channels, (x, y), (x, t), and (y, t) and densely computing Harris corners across these spaces. A 3 digit code is created for each corner, corresponding to its scale, which channel it was computed on, and its orientation. The APriori algorithm is used to mine frequently occurring sets of corners based on this 3-digit code, yielding discriminative features for classification. While they have reported accurate performance, dense sampling is extremely space and time costly on a dataset as large as the TRECVID corpus. MoSIFT is a spatio-temporal keypoint descriptor for action recognition, proposed by Chen *et al.* [19]. Rather than simply extending a two-dimensional keypoint into three dimensions, as was done in [14], MoSIFT is an amalgamation of two separate two-dimensional descriptors, SIFT [15] and optical flow. SIFT is used to detect keypoints in the spatial dimension, while the magnitude of the optical flow defines whether there is sufficient motion at those spatial locations to be used as spatio-temporal keypoints. If there is sufficient motion, the optical flow descriptor is appended to the SIFT descriptor to create a descriptor consisting of 256 floating point values. Using MoSIFT for event detection has seen success in recent TRECVID proceedings [4, 8, 7]]. Despite its reported success, MoSIFT remains prohibitively slow for processing ongoing streams of video data.

To this date, no perfect solution has been realized for the SED task. There does not yet exist an algorithm which yields an exact number of true positive responses and no false positives. However, implementing recently proposed efficient algorithms as a tool to reduce the computational load for a human surveillance agent is a very conceivable task.

In the following section, we present an overview of our approach to automated event detection. Following that, in Section 2 we describe an implementation that makes use of our approach to assist human surveillance agents in efficiently locating pre-defined visual events in a large corpus of surveillance footage. The use of a Visual Analytics component is described in Section 3. We follow that up in Section 4 with the results our approach achieved on the TRECVID interactive SED 2012 task, and finish with a discussion in Section 5.

2. Automated detection of events using Video Analytics

2.1. Approach

Our implementation is an incremental process that can be visualized in Figure 2. We first compute an intermediate image representation by the absolute pairwise difference frame between two video frames five frames apart. From this difference frame, we extract several binary spatio-temporal keypoint descriptors and a bag of words model is used to assign descriptors from a temporal sequence to visual words from a precomputed codebook. Finally, this bag of words representation is passed as a feature to a pretrained support vector machine (SVM), yielding a single SVM response. Over time, these responses yield an SVM response distribution and sufficiently large local maxima within that distribution are deemed to be positive events.

2.2. Extracting binary spatio-temporal descriptors

An appearance model that does not sufficiently capture the discriminative features of an event will fail to provide the classifier with sufficient detail to accurately detect that event in a query video sequence. Furthermore, in a video sequence it is not only the local appearance that is of importance, the motion model may be equally important. These models have been thoroughly studied in the computer vision literature, and many have been successfully applied at TRECVID. MoSIFT is a successful appearance and motion model that has been applied in previous iterations of TRECVID [19]. MoSIFT defines a 256 dimensional feature vector consisting of 128 floating point values for the local appearance, via SIFT [15], and 128 floating point values for the motion, via optical flow. While this descriptor has been used to achieve state of the art performance at TRECVID, the descriptors are large and slow to match, making them unsuitable for use as the evaluation corpus grows with time and additional camera views. We follow a similar philosophy to the one described by Chen *et al.* [19], with more compact and efficient descriptors.

Rather than capturing motion with an expensive optical flow computation at each detected keypoint, our approach transforms the current frame by taking the absolute difference image between the frame and the image 5 frames in the past. This representation implicitly encodes both appearance and motion simultaneously, providing us with a new image from which we need only to extract an appearance descriptor from. For this appearance descriptor, we elect to use the FREAK descriptor [1], which uses a series of local binary comparisons to describe a local image patch with a 64 byte binary string. These strings are not only more compact than MoSIFT, but allow for efficient matching with the hamming distance operation. FREAK descriptors captured on this difference frame encode both spatial and temporal information, giving us an informative and efficient descriptor for event detection.

Standard keypoint detection algorithms detect keypoints solely in a spatial manner. We augment keypoint detection by running a second check over each detected keypoint. For each detected keypoint, we sum the values in the corresponding image patch region in the difference image. A high sum in this region indicates significant change between the paired frames, which increases the likelihood of true movement in that patch. We normalize the patch for scale invariance and compute this sum over each detected keypoint. If the sum exceeds a pre-determined threshold, the keypoint is accepted. Otherwise, there is not enough movement within the patch to be useful for event detection, and the keypoint is discarded.

2.3. Bag of words model

As events are temporal in nature, they should be described by a distribution of events over an elapsed time window. This is achieved by computing a bag of words representation across a series of frames. For *PersonRuns*, we have decided to use 50 frames to describe a potential event. Therefore, over 50 frames all of our feature descriptors are extracted and assigned to the nearest *codeword*, a pre-selected feature determined at training time, often selected by clustering the training descriptors. A histogram is constructed, where histogram bin i is incremented each time a descriptor is deemed to be most similar to the i^{th} codeword. In the end, we normalize this histogram to ensure that scenes with more detected keypoints are not overaccounted for.

Studies have shown [12] that using k-means clustering to define the codewords for a bag of features codebook often overfits to the densest region of the feature space, making it no better than random cluster selection. Our experiments validated that hypothesis on video action recognition, leading us to use class-balanced random clusters for visual codebook selection. Given a training set of features, we randomly select features to be the quantizing codewords, while ensuring that there is an even proportion of positive features (features from our target event) and negative features.

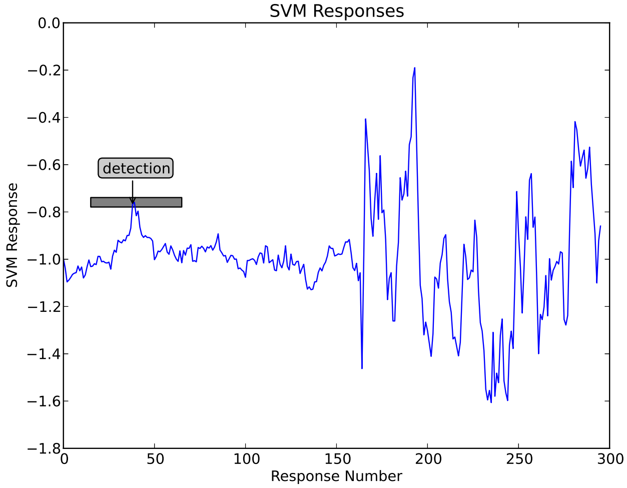


Figure 3. We test whether a peak in the SVM response space is a detected event by checking the window around that peak for greater values. In this case (the gray box), there are no greater peaks and the point would be detected as a *PersonRuns* event.

2.4. Event detection

A query video sequence will incrementally build new bag of words features, which can be fed into an SVM to classify that specific feature. With that in mind, we pose the event detection problem as the problem of finding large local maxima in the SVM response space.

Each bag of words feature is encoded as a normalized histogram. To classify this histogram, we use a support vector machine with the *histogram intersection* kernel. For histograms a and b, the histogram intersection kernel is defined as

$$K_{\cap}(a,b) = \sum_{i=1}^{n} min(a_i, b_i) \tag{1}$$

Our experiments have shown that the histogram intersection kernel works as well as the χ^2 kernel, and sometimes yields better performance.

Each of these SVM responses gives a floating point value corresponding to how likely the feature belongs to the *PersonRuns* event. The set of all bag of words features over an interval gives a distribution with several peaks and valleys, visualized in Figure 3. We are interested in SVM responses that are much higher than the local neighbourhood of that specific response. We begin with a single pass over each response and extract each response x_i where $x_i > \tau$, for some threshold τ, $x_i > x_{i-1}$ and $x_i > x_{i+1}$. This leaves us with a small number of candidate events, the areas with a direct local peak.

With the subset of candidate events extracted, we finalize the *PersonRuns* decision by centering a window around each candidate point i with SVM response x_i and scanning the distribution in that window to test whether x_i is a window-wise maximum. If it is, then the *PersonRuns* event has occurred at location i. The size of the window is relative to the assumed length of a *PersonRuns* event. We previously defined that *PersonRuns* is a 50 frame event, so the window would be for 25 frames before and after i. We also performed experiments with the *PersonRuns* event being a 25 frame event.

2.5. Training

The Gatwick dataset is used as a source for training data. As the ground truth is available, all of the PersonRuns events were extracted from this dataset for use as positive examples. Each event is processed by computing the keypoint descriptors as described in Section 2.2. These keypoints are then transformed into bag-of-words features for use as training examples for the SVM. A similar approach is taken for the negative examples, where random samples of the video are taken from any time frame that a PersonRuns event does not occur. Finally, the features are used to train an SVM with a modified implementation of LibSVM [5], with the histogram intersection kernel.

Clusters for the bag-of-words model were selected at random, as described in Section 2.3. The difference between k-means clustering and random cluster selection was negligible in our experiments. When sampling clusters, we ensured class balance by selecting 500 random clusters from keypoints from positive event instances and 500 random clusters from keypoints among the negative event instances.

3. Filtering false alarms using Visual Analytics

As mentioned in the introduction, a Visual Analytics tool that allows a human officer to efficiently process the data displayed on a computer screen is needed in addition to Video Analytics. Related to the SED task, an interactive graphical user interface that intelligently organizes detected events to allow a human to efficiently analyze the events, deciding whether each detected event is a true positive, is required.

The Canada Border Services Agency has previously developed Video Analytics Platform (VAP) [10], that is well suited to this task (Figure 4). VAP was originally developed for the testing and integration of third party video analytics codes with the existing IP-camera based video surveillance infrastructure. A critical feature of this platform is the *Event Browser* which is an interactive web application for displaying detected events according to filtering criteria. It is designed so that to enable humans to use their Visual Analytics power in order to be able to efficiently find and extract useful information, while ignoring or discarding the information that is not useful.

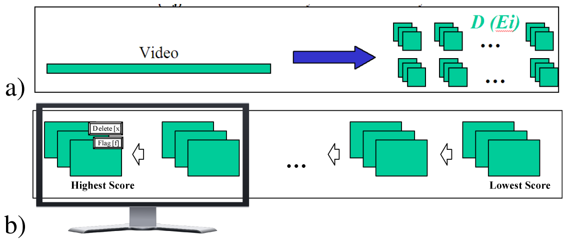


Figure 4. Key idea behind VAP software: a) converting continuous stream to set of bags of graphically annotated images representing the potential events of interest, b) the bag of images which are more likely to be useful are shown to the user first.

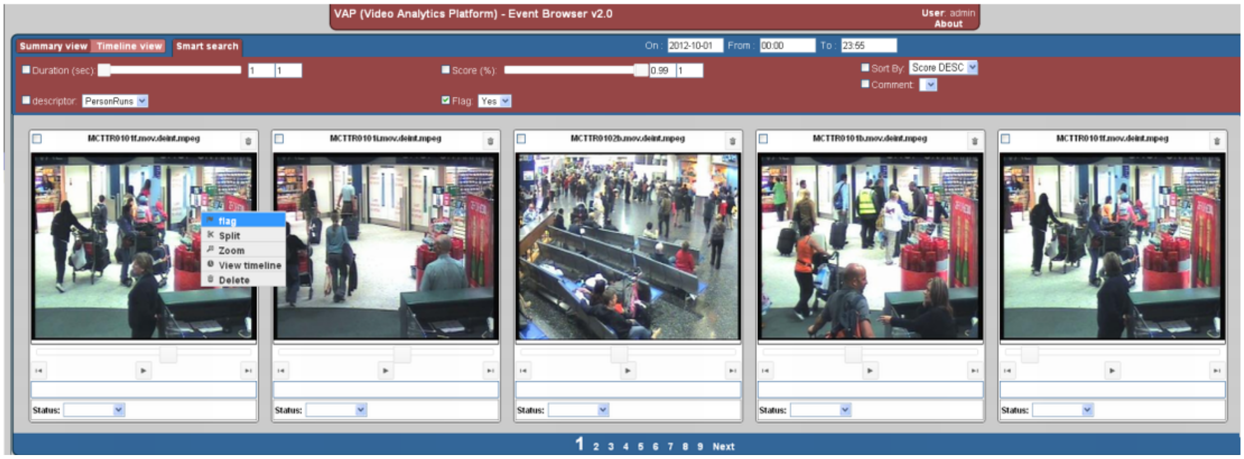
3.1. Making use of human visual recognition power

In “Illuminating the Path” [18], Thomas and Cook define visual analytics as the science of analytical reasoning facilitated by interactive visual interfaces. From [2] *“The purpose of visual analytics techniques is to help a human analyst to understand some data and underlying phenomena. Thus, it can be used in studies of individual movement behaviours, including behaviours of animals.”* While the computer is efficient at processing lots of information quickly, the human is efficient at quickly making sense of the information if it is relatively small and well presented. Visual Analytics is all about showing the data in a format most suited for the human. From [13], Visual Analytics is “The best of both sides” between the machine and the human, it is a collective effort from the machine best at processing data and the human best at perception.

The Video Analytic Platform is built while keeping Visual Analytics in mind. The concept is based on event analysis where every event has a bag of images consisting of a subset of the video. In the VAP lexicon, Event, designated as E, is an instance when certain conditions related to what is observed in the video are met, whereas Details of interest, designated as $D\{Ei\}(E)$, are a set of static images and associated metadata (annotations, timestamp and score) that are extracted and saved from the video when an event of interest happens, of which one image with annotation is chosen to represent the Event.

Based on these definitions, the main VAP task is to replace a continuous video-stream with a list of Details $\{Dj\}$ that can be efficiently browsed and analyzed.

The interface can filter the results by type of descriptor, score, flag, timestamp, camera and comment. This multi-dimensionality gives the user the ability to search for events of interest and prioritize its efforts. The efforts were optimized by sorting the events by descending algorithm score and thus giving a better chance of finding important events in a limited time period. The descriptor was set for only



VAP (Video Analytics Platform) - Event Browser v2.0

User: admin

On: 2012-10-01 From: 00:00 To: 23:55

□ Duration (sec): 1 1

□ Score (%): 0.99 1

□ Sort By: Score DESC

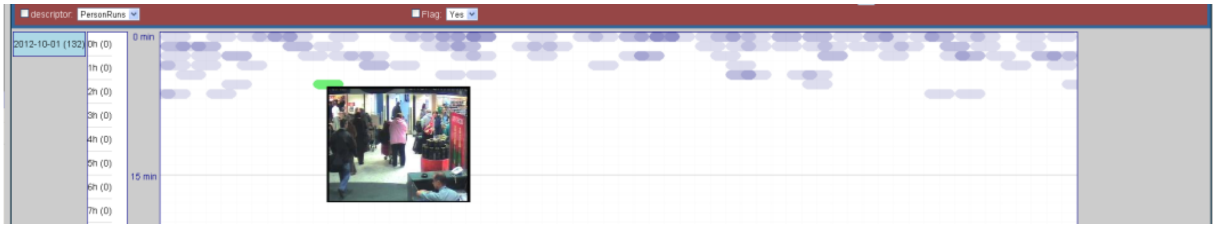
□ Comment:

□ descriptor: PersonRuns

☑ Flag: Yes

About
Summary view
Timeline view
Smart search
MCTTR0101f.mov.deint.mpeg
MCTTR0101i.mov.deint.mpeg
MCTTR0102b.mov.deint.mpeg
MCTTR0101b.mov.deint.mpeg
MCTTR0101f.mov.deint.mpeg
flag
Split
Zoom
View timeline
Delete
Status:
1 2 3 4 5 6 7 8 9 Next

Figure 5. VAP Event Browser interface that allows efficiently reviewing detected events



□ descriptor: PersonRuns

□ Flag: Yes

Figure 6. Timeline View of the VAP Event Browser interface allows analyzing events using the timeline information.

one type of event. Suspicious events would be flagged, after processing the operator would select only flagged events and could review in more detail whether the event is a false positive or not.

To improve the efficiency, peripheral vision and physical navigation was used. From [3], *"The key benefits of exploiting physical navigation over virtual navigation are its physical efficiency (especially eye and head movements), its cognitive efficiency as a user interface, and its natural learnability"*. Multiple events were played at the same time in loop in descending score order. The length of the video was determined by the underlying architecture, but we determined that a one to two second event was optimal. The user could quickly delete a false alarm or flag the video as being important. By having five videos played at the same time, we optimized the human and computer time. The human was able to observe more than one video at the same time as detecting the running action does not need the full attention of the operator. The computer was able to load the latest videos as the operator was watching the oldest ones thus giving a minimal loading time.

For evaluating the efficiency of the peripheral vision and physical navigation components, two tests were run with identical events, with a button to flag a video as a true positive and a button to delete a video in the event of a false positive. The first test displayed only a single video at once, while the second test displayed multiple videos simultaneously. Displaying multiple videos allowed us to process more *PersonRuns* events in the same span of time. In a 3-minute test, a user was able to classify 46 events per minute when multiple videos displayed simultaneously. In constrast, displaying only a single event slowed down user classification time to 34 events per minute.

3.2. Applying VAP software to TRECVID test

In the context of this TRECVID submission, all detected events are imported into VAP and sorted by SVM response, so events deemed most likely by the SVM classifier appear at the start of the interactive process. For each event, the video plays and a simple click can flag an event as a true positive. Alternatively, another click removes an event if it is deemed to be a false positive. This can be visualized by the context menu in Figure 5. We exploit the user's ability to detect events in their peripheral vision by playing the five most likely events simultaneously, in a horizontal row. Our experiments found this technique to be effective, as we were able to identify events occurring in videos which were not the immediate focus, increasing the number of videos analyzed within the allocated time slot. For the submission, the human operator was successful at processing more than

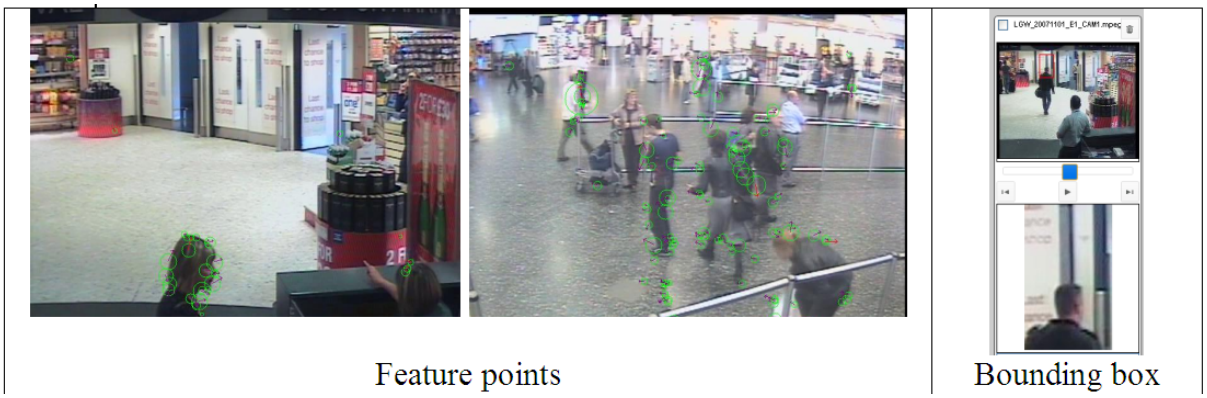


Figure 7. Extracted meta-data, such as rectangles, points, and vectors detected by the detection algorithm, can be used by VAP Event Browser interface to make visual processing more efficient though the use overlaid graphical annotations and extracted regions of interests.

600 events in 25 minutes or 24 events per minute with the help of VAP.

Scalability is achieved by distributing all automated components across multiple machines, and can be furthered by having multiple users classifying events. One could trivially accomplish this, as the platform is web based.

3.3. Additional VAP software features

In addition to Smart Search view, VAP Event Browser can overlay videos with useful visual information (Figure 7) and provides an ability to view events using Timeline View. Timeline View shows events using timestamp information, as shown in Figure 6. — One blue block shows one event in a one hour window. The length of the block is proportional to the length of the event. This view is very useful for processing surveillance footage with real timestamps, since the timestamp may help significantly expedite visual classification. For example, classification between a car and a bicycle could be done by checking the event length. In current TRECVID evaluation however, this view could not be fully used at because the timestamp of the TRECVID video sequences is not known, and also because the length of all video clips extracted by the video analytic module was fixed (2 secs) for all events.

4. Results

All training and event detection was done on the Gatwick dataset, surveillance footage captured at the London Gatwick airport recorded with MPEG2 compression at a resolution of 720x480 at 25 frames per second. This dataset consists of five cameras which contain some overlap. Each camera is broken up into several hours of video across multiple days, which ensures that systems do not overtrain on specific persons and allows for different variations on the same event. Camera 1 observes people exiting and entering through a series of doors, as well as a few shop entrances. Camera 2 observes a dense waiting area with several people sitting and walking through a crowded walking lane. Camera 3 observes many people standing and waiting, as well as a consistent flow of pedestrians passing by. Camera 4 directly observes two elevator doors with very little else in the scene. Finally, camera 5 observes multiple crowded horizontal walking lanes, where pedestrians will walk by at different visual scales. Training was performed on selected portions of 60 hours of the development and evaluation video sequences from these cameras. Event detection was performed on a final 15 hours of test data.

The interactive surveillance event detection task allows for a user to spend 25 minutes with a visual analytics tool to select true positive events from the video analytics system's event detection process. Within this 25 minute timeframe, our test user selected 15 "true positive" events for the *PersonRuns* task. As described in Table 1, nine of these events selected by the user were deemed to be false positives, while six of them were deemed to be true positives. Due to the nature of the underlying event detection system, all events processed were of two fixed lengths as processed, 50 frames or 25 frames. As such, it is possible that some of the false positives were a result of the intersection between the selected event and the ground truth being too small to be considered a true detection. A possible remedy for such a situation would be to add sliders to the visual analytics tool to modify the start and end frames of each event. While this would slow down the human processing speed, it would increase the precision of each detection, allowing us to accurately capture the entire event and less spurious data.

The DET curve for the returned results is displayed in Figure 8. Our returned DCR score is 0.9469. A DCR score is a weighted linear combination of the missed detection probability (the number of missed detections divided by the number of events) and the rate of false alarms. A DCR score of 0 is indicative of a perfect detection system, so lower

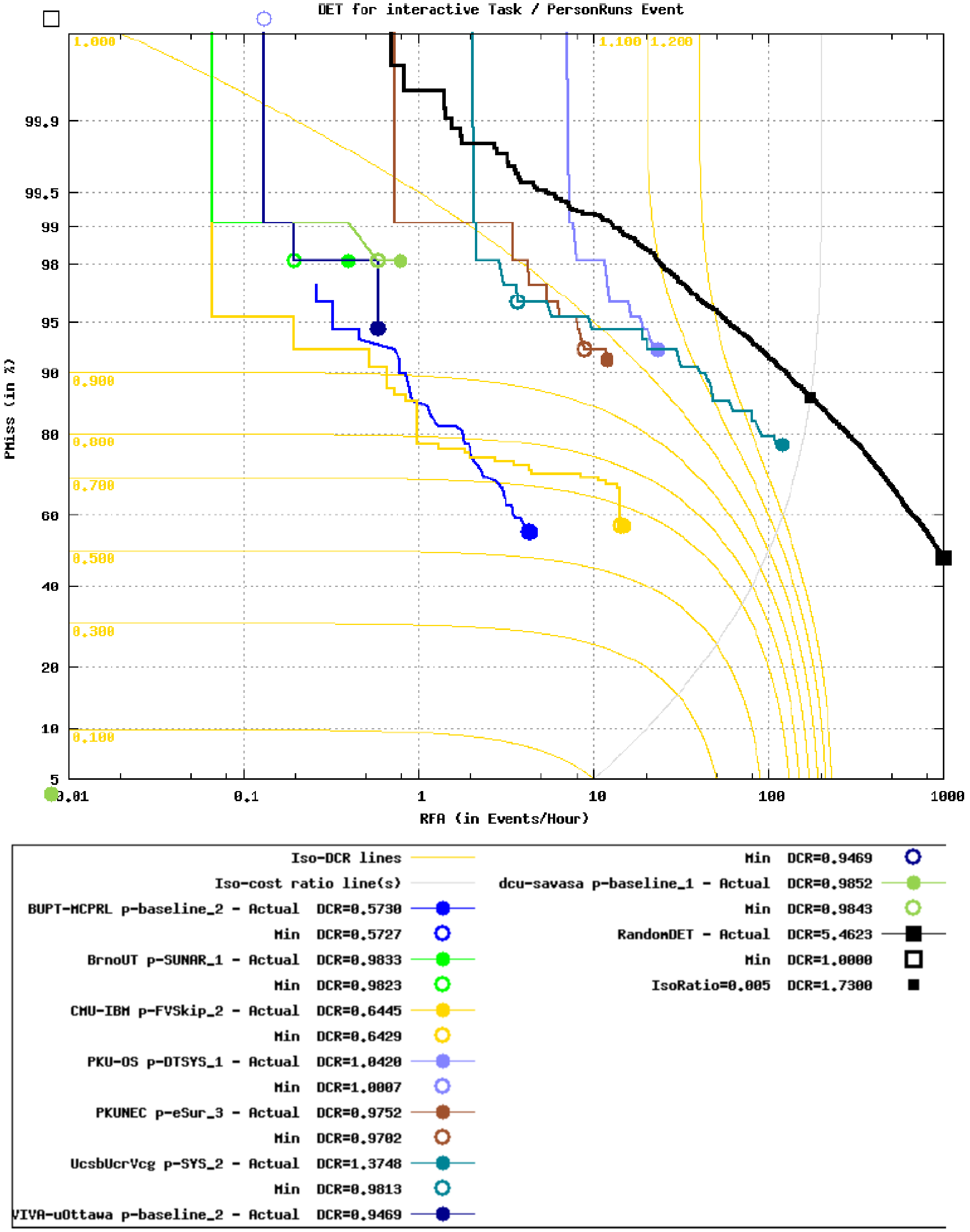


Figure 8. DET curve for our submission for the *PersonRuns* event on the interactive SED task. The DET curve plots the missed detection probability against the rate of false alarms. Our result is the darkest blue curve, as shown in the legend (VIVA-uOttawa p-baseline_2). The actual DCR value is equivalent to the minimum DCR value, so only one node is visible due to the minimum and actual DCR values overlapping.

Title	#Targ	#NTarg	#Sys	#CorDet	#FA	#Miss	RFA	PMiss	DCR
PersonRuns	107	9	15	6	9	101	0.59027	0.944	0.9469

Table 1. Details of our submission for the *PersonRuns* event on the interactive SED task. There were 107 true events (#Targ), while we detected 15 (#Sys). 6 of those 15 events were deemed to be true events (#CorDet), while 9 were deemed false alarms (#FA), giving us 101 missed detections (#Miss). Our rate of false alarms is 0.59027 (RFA), and our percentage of missed detections is 0.944 (PMiss). The weighted linear combination of the false alarm rate and probability of a missed detection is 0.9469 (DCR).

DCR scores are deemed to be representative of a more accurate event detection system, with respect to false alarms and detected events. A thorough explanation of the performance measures is available in [17].

In the context of other TRECVID teams, two groups achieved lower DCR scores, while five teams have higher DCR scores reported than our DCR of 0.9469. The mean DCR is 0.9406, lowered by the best DCR of 0.573, while the highest DCR was 1.3748. Our rate of false alarms (RFA) is 0.59027, the second lowest in this year's TRECVID task. The average RFA is 22.0040075, significantly higher than our own. The disparity in RFA scores indicates that there are multiple methodologies that were used to determine how to prune the automated detection results. The largest RFA value is 119.82510, indicating that rather than manually filtering events sequentially, an approach of tuning parameters to optimize on a subset of the results may have been taken. Conversely, the lowest RFA score is 0.39351. That score corresponds to a submission with very few (8) detected events, likely indicating that a similar approach to our own was taken, through manual sequential filtering.

5. Discussion

In this paper, we propose the use of a standard bag-of-words event detection model with binary feature descriptors. The binary feature descriptors permit efficient matching between detected keypoints and cluster centers, by using the hamming distance rather than a slower distance metric, such as the Euclidean distance. Furthermore, the dimensionality of the descriptors is greatly reduced, leading to further efficiency gains and a reduced disk space cost. Despite the computational gain, the overall recognition performance of the system could be greatly improved.

To address the time constraint requirement of the SED task and in order to filter out efficiently the false alarms, we piped the results obtained in the video analytic component of our solution to the visual analytic component which relies on the human's visual recognition power to enable efficient processing of visual data. This component, originally developed by CBSA for testing third party video analytics in operational CCTV environments, is further custom-tuned to efficiently view, flag and delete TRECVID detection results data.

Further work includes improving the video analytic components (eg. improving the data used to train the classifier, a per-camera classifier, restricting the search space in each camera view to remove unlikely areas for an event to occur, etc), visual analytic component (eg. further tuning of multi-clip viewing, keyboard short-cuts, graphical annotations, and visually salient GUI components) as well as better data pipe-lining between both components and more configurability (eg. to allow detected clips to be of variable length and to have more detailed associated metadata).

5.1. Relevance to operational needs

In order to relate the value of the obtained results to the operational needs, we present in Table 2 the estimates of the value-cost analysis of the presented technology, measured in terms of full-time employeee (FTE) required to perform that task.

Automated event detection technologies garner a great deal of interest from government and private organizations, due to the vast benefits they could bring to real-time surveillance applications, as well as post-event surveillance data analysis. Despite this interest, the current maturity of these technologies does not meet the standard required for real-time deployment in the field. Experiments at TRECVID reveal that with 25 minutes of human analysis, only 6% of true events can currently be identified in a 15 hour testing set.

Despite the current shortcomings, the results at TRECVID showcase growing evidence in the maturing of video analytics techniques over the recent years. Combined with efficient visual analytics interfaces, they may provide an effective means for post-event analysis of complex surveillance data.

6. Acknowledgement

This work was supported by the Defense Research and Development Canada Centre for Security Science Public Security Technical Program (PTSP 03-0402BTS).

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. CVPR 2012 Open Source Award Winner.

[2] G. Andrienko, N. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explor. Newsl.*, 9(2):38–46, Dec. 2007.

	HR cost	computer cost	# Precision (TP)
Manual processing	3 FTE days	3 computer-days	97%
Video Analytics	3 FTE hours	40 computer-days	˜ 12%
Video Analytics + Visual Analytics	125 FTE hours	+3 computer days	˜15%
Video Analytics + 25-minutes of Visual Analytics	25 FTE mins	+ 25 mins	6%

Table 2. Value-cost analysis of the VA solution compared to manual processing. Video Analytics + Visual Analytics numbers are estimates based on typical operational requirements, extrapolated from experimental setups at a smaller scale.

[3] R. Ball and C. North. The effects of peripheral vision and physical navigation on large scale visualization. In *Proceedings of graphics interface 2008*, GI '08, pages 9–16, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[4] L. Bao, L. Zhang, S. Yu, Z.-Z. Lan, L. Jiang, A. Overwijk, Q. Jin, S. Takahashi, B. Langner, Y. Li, M. Garbus, S. Burger, F. Metze, and A. Hauptmann. Informedia @ trecvid 2011. In *TRECVID*, 2011.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[7] X. Fang, H. Zhang, C. Su, T. Xu, F. Wang, S. Tang, Z. Xia, P. Peng, G. Liu, Y. Wang, W. Zeng, and Y. Tian. Pku-nec @ trecvid 2011 sed: Sequence-based event detection in surveillance video. In *TRECVID*, 2011.

[8] Z. Gao, A. Liu, Y.-T. Su, Z. Ji, and Z.-X. Yang. Tjut-tju@trecvid 2011: Surveillance event detection. In *TRECVID*, 2011.

[9] A. Gilbert, J. Illingworth, and R. Bowden. Fast Realistic Multi-Action Recognition using Mined Dense Spatio-temporal Features. In *Proc. Int. Conference Computer Vision (ICCV09)*, 2009.

[10] D. Gorodnichy and E. Dubrofsky. VAP/VAT: Video Analytics Platform and Test Bed for Testing and Deplying Video Analytics. In *Proceedings of SPIE Volume 7667: Conference on Defense, Security and Sensing*, 2010.

[11] T. Gorodnichy, Mungham. Automated video surveillance: challenges and solutions. ACE Surveillanec (Annotated Critical Evidence) case study. In *NATO SET-125 Symposium "Sensor and Technology for Defense against Terrorism"*, 2008.

[12] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, pages 604–610, Washington, DC, USA, 2005. IEEE Computer Society.

[13] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Information visualization. chapter Visual Analytics: Definition, Process, and Challenges, pages 154–175. Springer-Verlag, Berlin, Heidelberg, 2008.

[14] I. Laptev and T. Lindeberg. Space-time interest points. In *IN ICCV*, pages 432–439, 2003.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[16] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[17] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[18] J. Thomas and K. Cooks. Illuminating the path. *IEEE Computer Society*, 2005.

[19] M. yu Chen, H. Li, and A. Hauptmann. Informedia @ trecvid 2009: Analyzing video motions, 2009.