Automatic aggression detection inside trains

Zhenke Yang*
*Sensorsystemen
Netherlands Defence Academy,
Den Helder, The Netherlands,
Email: z.yang@nlda.nl, l.j.m.rothkrantz@nlda.nl

Leon J.M. Rothkrantz*†
†Man-Machine Interaction Group
Delft University of Technology,
Delft, The Netherlands,
Email: l.j.m.rothkrantz@tudelft.nl

***Abstract*—This paper presents work addressing the challenges of video analysis for automatic detection of aggression in a train. Using data from surveillance cameras, the system assists human operators in their work. It is unobtrusive and respects the privacy of passengers. We used existing algorithms to recognize and classify human behavior. While evaluating the algorithms we paid special attention to their ability to cope with environment specific issues, such as varying lighting conditions and (self)occlusions. A passenger behavior model was developed based on many hours of observing and studying professional operators as they analyze and respond to surveillance data. Experiments were conducted in a real train to evaluate the detection system.**

***Index Terms*—aggression, surveillance, train, rule-based.**

I. Introduction

Aggression in public transport causes destruction of property as well as mental and physical harm to passengers. To prevent aggression in trains, the Dutch Railway company (NS) has equipped some trains with surveillance cameras. To maintain a safe train, human operators need to monitor the camera images and take actions when necessary. All the cameras are therefore connected to a central control room where human operators can keep watch.

As the number of camera is expected to increase over time, it is expected that human operators will have difficulty to keep up with the ensuing data explosion. Another problem with humans is that they lack the ability to concentrate on repetitive and monotonous tasks for lengthy periods of time [1], such as monitor camera images. Computers do not suffer from this concentration problem. Thus, from this perspective, computers seem to be better candidates for the surveillance function. However, object detection tasks that seem easy or even basic for humans proof difficult even for the state of the art object detection algorithms. Making sense of situations and predicting possible aggressive outcomes of situations poses an even greater challenge.

This paper presents research done in cooperation with the NS to explore the opportunities and possibilities for computer assisted aggression detection. During the project, we interviewed and analyzed human surveillance operators to find out what cues they use to detect and to assess aggressive situations. Next we designed and implemented an aggression detection system based on these findings. The system consists of two parts: a low level observation part and a high level reasoning part. The implemented system is designed to function as a support system for the human observer. If an aggressive situation occurs in the compartment, the detection system will warn a human operator in the control central to take further actions. To test and evaluate the system we conducted several experiments with actors playing aggressive as well as normal roles. In the future, each train compartment is to be equipped with such a system.

The remainder of this paper is structured as follows. First we will delve deeper into the problem and discuss the work and research already done in this area. Next the design of the system will be presented in which we will go into more detail in the observation part and the reasoning part separately. Finally we will describe the experiments conducted in the evaluation phase and end the paper with a discussion and conclusions.

II. Background and related research

A prerequisite for an automated aggression detection system is the ability to detect (abnormal) behavior patterns in the input data. In the field of pattern recognition, many techniques for detecting all kinds of patterns have already been developed. In the train we have to deal with additional challenges. Some of these challenges are technical in nature. They stem from the unpredictable world of the train environment in which environment parameters are only partly under control, such as lighting conditions, background noise, train movement (shaking), and occlusions. The main challenge however lies in the correct interpretation of detected patterns and reasoning with them in the context of unpredictable human (aggressive) behavior.

Numerous methods have been used by other researchers to detect and recognize patterns that are related to aggressive behavior. These methods range from low level event detection [2], [3] to emotion recognition [4] and activity and behavior modeling [5], [6]. In order to get a broader understanding of the problem domain and to get a feel of the kinds of cues related to aggression, we will first discuss the concept of aggression.

A. Aggression

Because of the broad definition of aggression, many discussions about aggression can be encountered in literature. The current consensus is for at least two broad categories [7]: instrumental aggression and affective aggression.

- The instrumental aggressor acts to obtain an apparent goal e.g. theft. Our approach to recognize instrumental

978-1-4244-6587-3/10/$25.00 É2010 IEEE

aggression is to detect the patterns (actions and behaviors) that are associated or correlated with the goal. Most security experts will also concur that this type of aggression usually occurs when the situation and the environment permits. Being able to identify and detect these "high risk" environments thus forms a part of the detection process.
- Affective aggression is associated with strong emotional feelings. Anger and fear are usually the dominant emotions. Often affective aggression follows as a defensive response to a perceived provocation. If this happens, the aggression can be detected as a cycle of provocations and responses.

The main targets for our detection system are the specific aggressive situations that have the most impact or occur most frequent in the train. Co-incidentally, these situations (e.g. robbery, theft, violence (towards conductor), abandoned luggage, vandalism) fall under instrumental aggression. Therefore, we will assume the definition of instrumental aggression, and consider emotion as a pattern correlated with the aggression (where appropriate).

B. Behavior recognition

Analyzing spatio-temporal changes in a dynamic scene is an important aspect of aggression detection. During this process one tries to detect unusual deviations from logical sequences of usual activity patterns. Many techniques to model behavior have been suggested e.g. Hidden Markov Models [8], Bayesian networks [9], Finite state machines [10] and stochastic context free grammars [6]. In [11], [12] behavior recognition was also applied to surveillance systems in public transport. Compared to previous approaches, the method adopted in this paper follows a more human centered route concerning the inference of aggressive situations. We interviewed security experts from the NS, which resulted in heuristics and a stepwise procedure for aggression detection and control summarized below.

1) Trigger: detect something unusual based on experience, external warning, a hunch, etc.
2) Orient: find the most salient cues in the scene and create one or more hypotheses.
3) Observe: find other cues that support or refute the hypotheses.
4) Conclude: when a threshold of plausibility is reached the hypothesis can be concluded.
5) Act: take the appropriate actions to solve the problem.

From the interviews, it was clear that the difficulty in the aggression detection and control process lies mainly in the unpredictability of human behavior. This also makes it difficult to create generic methods for aggression detection. However, given the specific scope of the problem however, we can limit the behaviors to those associated with normal situations (e.g. entering, sitting, exiting) and those associated with aggression. The detection steps leading to the conclusion can then be captured in a domain specific rule-base for the limited number of relevant behaviors. Rule-based inference can then be applied to reason about the aggressiveness or aggression potential of a specific train compartment.

To obtain information of behaviors that occur most frequently, we analyzed the incident database of the NS. This database contains all occurrences of major and minor incidents that have been reported in all train in the Netherlands. These incidents are divided into 8 categories: suspicious behavior, theft, violence, serious inconvenience, small inconvenience, vandalism, accident and fire.

C. Aggressive behaviors

To illustrate the kind of aggressive behavior we are interested in, this section gives a breakdown of the most frequent forms of aggression in the NS incident database. Each form of aggression is complemented with the possible ways of detection we extracted from the interviews with experts.

- Theft. The detection method applied in the system is to recognize the behavior of a potential thieve. A thieve is usually lingering around waiting for an opportunity and the deed itself shows an approach-strike-disperse pattern.
- Violence, such as fighting, can be detected by analyzing frequency and magnitude of physical contact and the reaction of the victim. Physical aggression will often be accompanied by shouting. Other passengers in the train may react to the situation as well.
- Serious inconveniences, such as intimidation, is a subtle form of aggression with little or no physical contact. This includes signals such as obscene or threatening gestures. Angry facial expressions are also an indication of this form of aggression. The aggressor often stands very close to the victim confronting him. There may be shouting involved.
- Vandalism causes objects in the train to be destroyed or defaced (e.g. graffiti). It is important to identify all train objects and their normal usage. The reactions and body language from other passengers can indicate when vandalism is going on.
- Abandoned luggage has become an incident that needs special attention since the recent bombings in the London underground and in trains in Madrid. Luggage forgotten by accident can cause great distress among worried passengers.

III. System design

The system is designed to function as a support system for the human observer. In the future, each train compartment is to be equipped with an aggression detection system. When an aggressive situation occurs in the compartment, the detection system will warn a human operator in the control room to take further actions. Since the system has to fulfill the purpose of reducing the risk of physical harm or other illegal activities in a public area, it has to adhere to several general requirements relating to the nature of the environment.

- The system should not be invasive of privacy or should be less invasive than alternative methods. The privacy

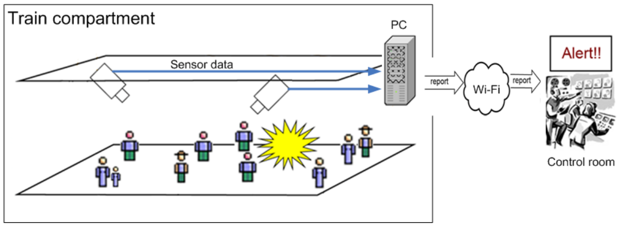

Fig. 1. General architecture of the aggression detection system

implications of such systems has been explored in other works e.g. [13].

- The positive effects of the system should not wear off over time. This problem appears with dummy camera's or surveillance systems lacking human operators, where people quickly learn the weakness of the system. An automatic detection system keeping a constant watch would not suffer from this effect.
- The system should have a classification of severity. This classification is needed to prioritize actions.
- The system should be non-obtrusive. As a result, sensors may need to be placed in sub-optimal locations.

Because of the complexity of aggression and the many objective and subjective factors that affect the emergence and perception of aggression, it is unlikely that all the factors can be cramped into one single algorithm. Instead, several classification sub-algorithms are necessary. Each sub-algorithm performs detection or classification of a separate contributing factor (possibly from different data sources) while all the detected contributing factors combine into the final classification system to yield the final classification.

A. System architecture

The general architecture of the aggression detection system has the properties shown in Figure 1. It consists of multiple independent, autonomous aggression detection units in each train compartment. A detection unit is a computer (located at a safe place in the compartment) connected to distributed sensors (multiple cameras) inside the compartment that provide it with raw sensor data. The computer combines the sensor data to assess the level of aggression in that compartment. The communication between units from neighboring compartments takes place to inform them of inter-compartment activities (e.g. fire, aggressive person moving to another compartment). Each computer has a wireless connection to a control room where a human operator is available.

In each computer, multiple classification algorithms are run in parallel. Each algorithm focuses on a different aspect of the aggression spectrum: one algorithm might perform face recognition, another applies gesture recognition etc. For many classification tasks, algorithms already exist. A detailed overview of the algorithms used in the detection process will be discussed in following sections.

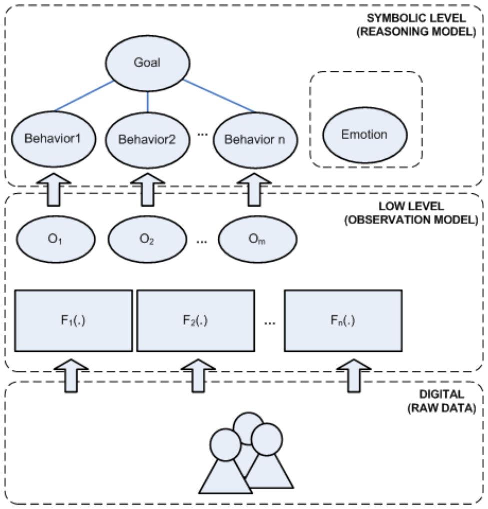

Fig. 2. The observation model describes how low level features ($F_1 ... F_n$) from raw data are combined into high level concepts. The reasoning model describes how high level concepts ($O_1 ... O_m$) are reasoned with in order to infer the presence of aggression

B. Aggression detection approach

The aggression detection approach is divided into two parts: (1) the observation model which describes how low level features from raw data are combined into high level concepts and (2) the reasoning model in which high level concepts are reasoned with in order to infer the presence of aggression (see Figure 2). The following sections explores both models in more detail.

IV. The observation model

The most common sensors in surveillance are microphones and cameras. This section describes the features that are extracted from the stream of images coming from a surveillance camera.

A. Camera alignment

We use the Direct Linear Transform (DLT) to determine the camera parameters of the captured images. With the knowledge of the camera parameters (such as orientation) we have a better understanding of the positions of objects in a image in relation to their real coordinates. This is important for object localization and multi-camera object tracking. The images produced by a projective camera can be interpreted as a sequence of three projective transformations: given a point $p = (x_w, y_w, z_w, 1)$ in homogeneous world coordinates and a point $q = (f \cdot x_i, f \cdot y_i, f)$ in image coordinates corresponding to the projection of p onto the image, the mapping of p to q can be expressed as:

$$q = \begin{bmatrix} \sigma_x & \sigma_\theta & u_0 \\ 0 & \sigma_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot M \cdot p \qquad (1)$$

The first matrix represents the intrinsic parameters of the camera, with (u_0, v_0) the coordinates of the principal point, and σ_x and σ_y the scale factors along the axes of the image.

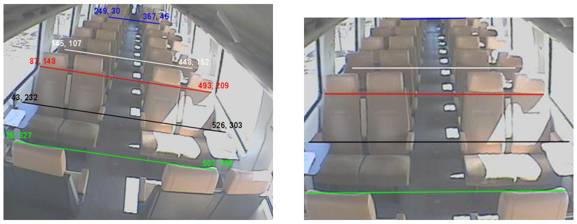

Fig. 3. Lines known to lie parallel to the x-axis can be used for determination of ϕ (left). Validation of the result: after rotation in the reverse angle, all the chosen lines are approximately horizontal (right)

The parameter σ_θ describes the skewness of the two image axes. For most cameras σ_θ is very close to zero.

M represents the extrinsic parameters of the camera and is given by:

$$M = \begin{bmatrix} \ddots & \vdots & & \vdots \\ \cdots & R & \cdots & T \\ & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Where R is the rotation and T the translation which relates the world coordinate system to the camera coordinate system. The optical center of the images can be approximated at the true image center. The camera orientation relative to a xyz axis system may be specified by three Euler angles: ϕ, θ and ψ.

Any rotation of the camera about the z-axis (ϕ) of the camera coordinate system, causes all horizontal lines to be rotated the same amount in the resulting camera images (as ϕ is the angle between a line along the x axis of the world coordinate system and the projection of it into the image plane). As a result, ϕ can be estimated by the rotation of some lines in the image known to be horizontal. Since all the seats in the train are ordered in straight rows, finding a few of these lines is straightforward (see Figure 3).

In a similar fashion θ and ψ can be estimated. For more details we refer to [14].

B. Motion segmentation

Advanced processing algorithms, such as tracking, need to know the objects to track in order to function. So first we need motion segmentation, to differentiate between pixels belonging to moving objects and pixels belonging to static background. One method is to use background subtraction techniques [15]: a model of the background is kept in memory, and when there appears a change, that is not consistent with the background model, it is seen as foreground. Because changes in lighting or weather can influence the background, an adaptive background model is used.

The codebook algorithm [16] that we used for this task (see Figure 4) adopts a quantization/clustering technique, inspired by Kohonen to construct a background model from long observation sequences. For each pixel, it builds a codebook consisting of one or more codewords. Samples at each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds.

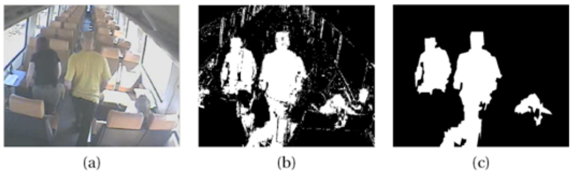

Fig. 4. Background subtraction method using the codebook method (b) of the original image (a). Post-processing connects foreground pixels to create blobs and can recover errors (c)

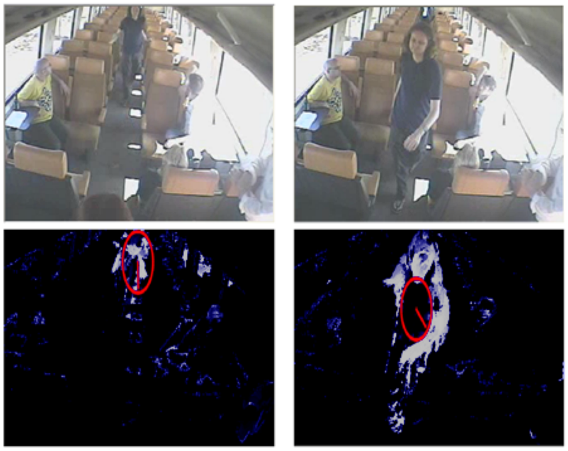

Fig. 5. Motion estimation of a person walking through the train corridor using silhouettes of detected blobs overlaid in time

1) Motion direction: To get an indication of the general movement of the detected foreground areas (a.k.a. blobs), motion templates are used. The silhouette of the blob is used to track its movement. By the movement of the blobs over several frames new silhouettes are captured and overlaid with the (new) current time stamp. Older motions gradually fade. These sequentially fading silhouettes record the history of previous movements and thus are referred to as the motion history image (MHI). Once the motion template has a collection of object silhouettes overlaid in time, we can derive an indication of overall motion by taking the gradient of the MHI image (see Figure 5). Matching a motion template gives a (context independent) indication of the general direction of motion.

2) Energy signatures: As the MHI image assigns values to a pixel according to the amount of change (motion) observed in the last few frames, it can be used to create an energy signature of the image by summing these values. The energy signature shows the amount of change in the scene and thus gives an indication of the amount of action that is taking place in the scene (Figure 6). Energy signatures provide a quick way to calculate the amount of motion in the train compartment. To be useful for aggression detection however, this information needs to be combined with other attributes of the train compartment (e.g. train status) in order to infer if the amount of motion is typical for the type of situation.

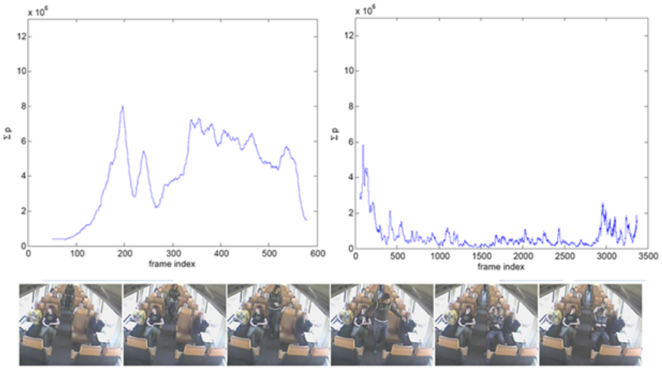

Fig. 6. The amount of motion in the compartment during certain actions results in different energy distributions. The graphs show the energy during the 'enter train' scenario

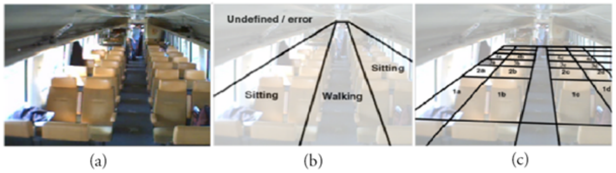

Fig. 8. Empty train (a) and designation of areas for masks and seat positions where faces are unlikely or likely to occur (b and c)

TABLE I

FACE DETECTION RESULTS OF A VIDEO SEQUENCE OF 1420 FRAMES. THE FALSE POSITIVE RATE (FP) IS REDUCED BY MASKING AREAS (MS) AND ANALYZING A DETECTED FACE IN RELATION TO A BLOB (MT)

Detection rate	FP + MS	FP + MT	FP + MS + MT
68%	20%	11%	4%

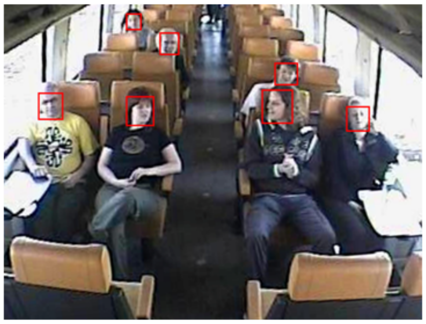

Fig. 7. Frontal face detection using Viola and Jones algorithm

C. Human detection

The most important object to detect regarding aggression is obviously the human. Several human characteristics can be used to detect humans e.g. faces, body poses or speech. Depending on the environment, setup and capability of the sensors one method performs better than the other. In large public spaces cameras may not get a sufficiently high resolution to achieve good results in face recognition. In this case, body recognition is a better choice. On the other hand, in confined spaces it is better to use face recognition since occlusion might influence the performance of human body recognition.

1) Number of people: The method we consider for the purpose of human detection is face detection, because other human characteristics are less effective for the train compartment. When a face is detected in an image, this means a person has been detected as well. In addition, the number of faces, indicates the number of people in the compartment. We choose the algorithm proposed by Viola and Jones [17] mainly for its speed. It is however not very robust under noisy circumstances, and very susceptible to changes in face orientation. Nevertheless, in larger frontal faces of sizes around 50x50 pixels, we achieved very good detection rates (see table I). A snapshot of a video frame where all faces are detected successfully is shown in Figure 7. If the size of a face drops below this threshold, detection rates fall rapidly.

With our train videos data, we also saw a high number of false positives (>20%). To lower the number of false positives, we first used a mask filter (see Figure 8)to remove all faces from areas where no faces are expected. This includes areas such as the windows and the ceiling, where the false positives commonly occur.

Another method to reduce false positives is to use the results of the motion segmentation process. Since the motion segmentation algorithm is independent of the face detection algorithm, we can cross reference a detected face with a moving blob (see Figure 4): the face should appear at the top of the blob.

2) Clusters and personal space: Another attribute of the train compartment derived from face detection is the distance between people in the train and the existence of clusters of passengers. For the clustering algorithm we used an algorithm in which we execute exactly one iterative refinement step per frame. Initially, each face is assigned to a separate cluster $m_1^{(1)}, ..., m_k^{(1)}$, where K is the number of detected faces. For each iteration step:

1) For each cluster, calculate new cluster influence range (R) by taking into account the average Mahalanobis distance of a face (x) to the center of the cluster (μ).

$$R_k = \frac{2}{N} \sum_{n=1}^{N} \sqrt{((x-\mu)^T S^{-1} (x-\mu))} \qquad (3)$$

2) Assign each new face (x) to the cluster (C) with the closest mean but within the influence range .

$$C_k = \{x_j : \|x_j - \mu\| \leq R_k\} \qquad (4)$$

3) For faces that don't belong to any cluster, a new cluster is created forming a single cluster.
4) Two single clusters are merged if the distance between the clusters is less than a threshold (half a seat length in this case).
5) Calculate the new means of each cluster.

$$\mu_k = \frac{1}{\|C_k\|} \sum_{x_j \in C_k} x_j \qquad (5)$$

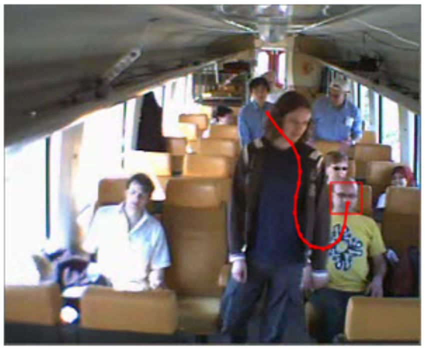

Fig. 9. Track plot for fragment of scenario 8b, overlaid on last frame of the scenario. The track is manually enhanced to improve visibility

The influence range (R) gives an indication of how close people are within a cluster. More specific behaviors, such as invasion of privacy and confrontations, can be found by analyzing the distance of a pair of people over time. A feature that is extracted is thus the minimum distance of the closest pair of faces in each cluster. The conclusion whether there is actual invasion of private space however is delayed till the reasoning phase, because this depends on other factors, such as the occupation of the compartment.

3) Unusual direction of attention: An interesting side effect of frontal face detection is that the direction of the visual attention of a person is known when a face is detected. In the train compartment, people tend to look forward or out the window. If the majority of the passengers (especially passengers that do not belong in the same cluster) focus their attention towards the same direction, then there is probably something going on in the compartment (e.g. a majority of people looking backward is very unusual).

D. Tracking

State-of-the-art tracking methods can be divided in several ways. An important first distinction is the way in which the tracked persons are represented. When there is no predefined explicit shape model, some possibilities are a box, an ellipse, the contours of a blob, or the blob itself. If there is an explicit shape model, a stick figure can be used, or every body part can have its own box. For the sake of efficiency we used the rectangle around a detected faces as shape model.

For the tracking algorithm, we adapted the Mean Shift algorithm to use the results of the face recognition algorithm. The 'mean shift' is the estimated direction and distance in which the target moves, and this is computed by comparing an already defined model target with candidate targets [18]. The advantages of this method are that no dynamic model is needed in advance. This produces satisfactory results as can be seen in Figure 9.

In the Initialization step of our tracking algorithm, a new track is created for every detected face. After that our tracking algorithm is continues as follows:

1) For every track, calculate the histogram of the last detected face. (We use the histogram of the face as a density estimate of the target.)
2) For every detected face, calculate the histogram.
3) Assign detected faces to existing tracks by analyzing a similarity measure and a search range. For target t and candidate location c, the similarity measure is the Bhattacharyya distance:

$$d(t,c) = -ln\left(\sum_{x\in H}\sqrt{t(x)-c(x)}\right) \qquad (6)$$

4) For the tracks that have no newly detected faces assigned, we use the similarity measure to find a suitable candidate in the search range. The search range is represented by an elliptical region with axis S_x and S_y that is expanded every frame to a maximum of 12 frames. The expansion value is based on the average motion of the previous points (p) of the track in both directions (x, y).

$$S_x = \frac{1}{N}\sum_{i=1}^{N}\|p_{ix}-\mu_x\|, S_y = \frac{1}{N}\sum_{i=1}^{N}\|p_{iy}-\mu_y\| \quad (7)$$

5) For remaining faces that have no tracks assigned, new tracks are created

By connecting the tracked points over consecutive frames we obtain motion paths. Typical motion paths can be distinguished. Most people enter a train compartment, sit, and leave it when they have reached their destination. Sporadicly, people hang around or move without any plan.

For typical behaviors of passengers we use a template based classifier that compares the observed path of a person to typical paths for certain actions. A typical path or template is a specific observation of a complete action, that is, a path of observed locations over time. The template consists of the coordinates and times of actions in the train (e.g. walking, running, begging). The templates are manually determined from experiment data.

The templates are then used to compute the similarity with an observed path, expecting similar behaviors to have similar paths. We use the Mean Square Error (MSE) as a measure for similarity. We can then compute the MSE for all coordinates separately. To score an entire measurement against a template we compute the sum of these errors, and divide it by the number of points compared. The template matching algorithm chooses the best match with the template most closely resembling the track taken by the object.

V. The reasoning process

The previous sections have presented a number of techniques to detect different cues or objects in the train compartment. In order to draw conclusions concerning the situation, these observations have to be viewed in the proper perspective. To this end we have created a behavior model, which specifies the meaning and relationship of relevant concepts to aggression (and to each other).

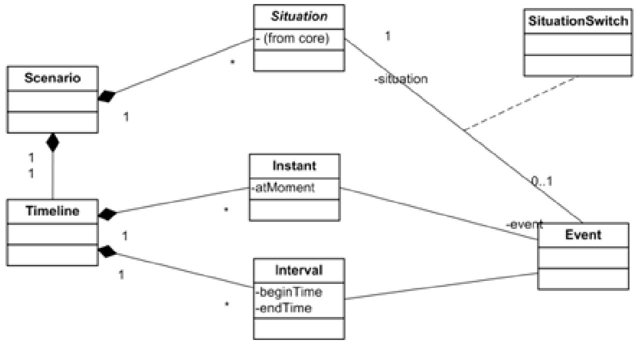

Fig. 10. Overview of the scenario sub ontology. A Scenario consists of a sequence of Situations and a timeline on which Events can be placed

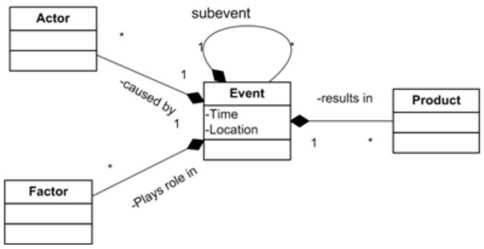

Fig. 11. Overview of the event sub ontology. An event has a location, a time and optional relations to actors, factors and products

A. Behavior model

The behavior model contains a formal definition of behaviors and objects in the train compartment to accommodate reasoning. The behavior model consists of a static context that specifies the objects in the compartment that play a role in aggression and their spatial relationships. The second part is the dynamic context, which describes the temporal aspects of the domain, for example how objects can interact within the environment. Because the static context is essentially just an enumeration of objects, we will not go into detail here.

The central classes in the dynamic context are ‘Situation’ and ‘Event’ (Figure 10). A ‘Situation’ includes an arrangement of concepts (from the static context) and dynamic relationships among them. ‘Event’ contains information about events in the real world observed at a specific time, it represents a way by which we can classify certain useful and relevant patterns of change. ‘Scenario’ describe situations that can occur in the environment, more specifically it summarizes the events and actions that are usually observed and their order of occurrence when such a situation takes place. A ‘Scenario’ consists of a sequence of situations and a timeline on which events can be placed. An ‘Event’ that leads to a transition of one situation into another is represented by a ‘SituationSwitch’.

The ‘Timeline’ concept addresses temporal information as depicted in Figure 11. A number of intervals and instants can be defined by which events can be associated to the ‘Timeline’. An ‘Event’ thus has a location, a time and optional relations to actors, factors and products.

An ‘Actor’ is the object that caused the event (e.g. passenger), a ‘Product’ is the possible appearing or disappearing objects as a result of the event (e.g. train stops in station results in passengers exiting), while a ‘Factor’ is an object that also plays a role in the event but does not belong in the aforementioned categories. In addition, an event can consist of a number of sub events. The ‘Event’ concept forms the connection between the objects in the static context and situations and scenarios in the dynamic context.

B. Unusual situations

Aggression detection by humans is triggered by an observation of a cue or heuristic that compels him to investigate further (see Section II-B). Following this approach, we created a list of triggers for unusual situations that, when detected, requires further investigation. The list is summarized below. All the situations can be detected by (combining) the features detected in the observation phase.

- Crowding,
- Running through the compartment,
- Lingering around,
- Moving against the general flow,
- Same direction of attention of passengers,
- Looking backwards or around continuously,
- Sudden high motion energy without apparent reason,
- Motion at unexpected places,
- Invasions of personal space.

Once an unusual event has been identified, it triggers the scenario based detection discussed in the next section.

C. Scenario based detection

In order to draw conclusions concerning the situation in the compartment, inferences have to be made given detected observations by the observation model. We used a rule-based detection approach in which a set of rules describes which conclusion to draw given the input. The idea is inspired by the schema theory [19]. In this theory, schemas are cognitive structures that link declarative and procedural knowledge together in patterns that facilitate comprehension of behavior within a context. The declarative part is comprised of object classes together with associated features and arranged in hierarchies in space and/or time. The procedural knowledge for the understanding and enacting of behavioral patterns and routines is encoded in scripts. A classic example of a script is Schank and Abelson’s restaurant script [20] that includes a structure for entering a restaurant, ordering, eating etc.

Following the schema theory, the possible aggressive scenarios are modeled as scripts, containing information about the concepts and events that usually occur when the aggressive scenario is encountered. A script is stimulated if the concepts in the script are really being observed. When a concept is observed, it adds to the support of the script. Each concept adds a different weight depending on the salience of the observation to the script. When a script reaches a threshold the script is triggered and the result is that the system recognizes the scenario.

A key idea in the theory is that some events can be more salient for a script than others (in a car accident, a damaged car is more salient than a police officer although both are things that one would expect to see) and that events need not necessarily occur in a strict sequence. Furthermore, the existence of a threshold ensures that not all events in the script need to be observed before the situation can be recognized.

D. *Implementation overview*

The aggression detection approach modeled after scripts in the schema theory is implemented as a rule-based reasoning scheme. In the beginning, when a small number of events are observed, many competing scripts may be in the reasoning system. But by further observation and the detection of new concepts, over time, a final hypothesis emerges (similar to expert recognition steps described in Section II-B). A script is implemented as a sequence of events and their consequences in the form of rules. An expert system's inference engine controls the proper application of these rules. CLIPS [21] was used as the expert system shell.

To construct the rule base, the expert knowledge obtained from interviews with human experts and results from analyzing surveillance videos have been formalized. The first stage in creating the rule base is to create a list of possible aggressive scenarios and a list of high-level concepts that have been observed in these scenarios. We have defined more than 40 scenarios. The scenarios are grouped in the incident types defined by the NS (see Section II-A).

The salience of the relationship between a concept and a scenario (i.e. the weight that a concept adds to a script when observed) is specified in an influence matrix. The matrix indicates for each feature how much the feature contributes to the likelihood of each scenario.

To limit the influence of old observations, a saliency decay function is also implemented. Whenever a concept enters the system, a timestamp (t) is attached to it (if the feature is already in the system, t of the old observation is updated). When the support for a scenario is calculated, the salience of each concept is adjusted with a value depending on t. For simplicity, the salience decreases linearly over time with a fixed decay factor. If the saliency reaches zero, the feature is removed from the system.

VI. Evaluation and results

Several experiments were conducted in a real train compartment to collect usable data for testing and evaluating the aggression detection system. During the experiments, actors had to perform scenarios as described in storyboards. These storyboards where previously validated by security experts for their realism. As the actors performed the scenarios, data was captured using the cameras in the train. As opposed to rigidly scripted approaches, storyboards offer more flexibility. As a result, many different versions of the same scenario were captured.

A. *Experiment setup*

The sensor setup used to capture the scenarios consists of four cameras. Their location and orientation is shown in Figure 12. The cameras are mounted in a straight line along the roof of the compartment and face downward with an angle. The orientation of the cameras is such that they face the center of the compartment.

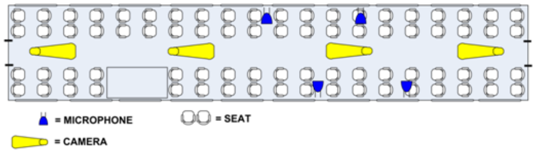

Fig. 12. The locations of the sensors seen from a top view of the train compartment. All cameras face downward with an angle and the center of the compartment has the largest camera coverage and overlap

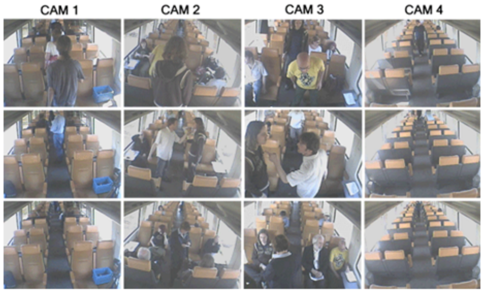

Fig. 13. All cameras face downward with an angle and the center of the compartment has the largest camera coverage and overlap

The cameras have zooming, panning and tilting capabilities, but these settings were fixed during the recordings. Aggressive and normal scenarios are recorded in sequences which total up to about four and a half hours of audio and video data. The data contains the aggressive scenarios as well as recordings of normal and spontaneous situations. The video cameras captured colored video at about 12 frames per second, at a resolution of 640x256 pixels.

The captured data was annotated with ground truth values about the situation in the train compartment. The annotation includes all the objects, their locations and characterizing features and relations with other objects. We used the concepts defined in the behavior model discussed earlier as annotation language. Essentially, an annotation file starts with the introduction of the sensors, the introductions of the actors and objects and then a list of events in which the actors and objects play a role.

B. *Evaluation*

The internal workings of the rule-based system are not that complicated. Therefore, extensive testing of the software was straightforward. The application was implemented such that it logs all the processing steps, allowing clear-cut evaluation and validation of intermediate results and decisions. In practice validating the application entails running the system, while a user manually checks whether or not the output (which includes the positive, negative and currently activated features) matches the desired output. Any unexpected or undesired results which could be mapped to flaws in the rule-based code (reasoning phase) were subsequently fixed. Undesired results that could be attributed to errors in the feature extraction and classification algorithms (observation phase), we tried to fix with modifications to the algorithms. These modifications have been discussed throughout Section IV. Using this method, all 10 aggressive scenarios in our dataset could be correctly classified.

C. Discussion and conclusions

In this paper, we presented a system that is able to detect aggressive scenarios in a dataset of videos taken from surveillance cameras in a train compartment. This dataset contains the most frequent aggressive behaviors. The system consists of a rule-base modeled using knowledge of the train and passengers. Modeling the rule-base constituted the largest part of the work, as knowledge extracted from security experts and personal observations had to be formalized and captured in rules.

Not all the concepts and conditions that experts reason with can be detected accurately and consistently at the moment. The technology needs to mature to a state that the results achieved by current algorithms can cope with the challenging conditions in the train compartment. For example, a face recognition algorithm was able to detect the position of the faces, from which the position of the human body is estimated. However, non of the algorithms to recognize emotions (such as anger, surprised) from facial expressions of the detected faces performed well. Thus the workable input space of the system is limited to the features that we are currently able to detect. Despite this limitation, the system was able to recognize the target scenarios correctly. At this moment we cannot give a statistical evaluation of the performance of the system because we only have a few test examples per scenario. In addition, the data used to train the system were captured in only two experiment session. Therefor, any conclusion regarding the performance of the system might be biased by over-training. The bottom line is that more experiment data is needed to truly evaluate the system.

A drawback of the rule-based approach is the complexity of building and maintaining large rule-based systems. A rule base of thousands of rules may require a trained staff to maintain. In addition, even though the threshold in the scripts allows for some flexibility in the recognition of scenarios, the reasoning system is essentially unable to cope with unexpected situations. That is why we looked to alternative approaches. For example fuzzy inference systems based on fuzzy rules instead of the crisp rules. Another option is the Bayesian alternative that summarizes complex relations between entities in terms of uncertainty values.

Acknowledgment

This research was done at the Delft University of Technology and funded by the MultimediaN project in collaboration with NS/ProRail.

References

[1] H. Koelega, M. Verbaten, T. H. van Leeuwen, J. L. Kenemans, C. Kemner, and W. Sjouw, "Time effects on event-related brain potentials and vigilance performance," *Biological psychology*, vol. 34, no. 1, pp. 59–86, 1992.

[2] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 318–325.

[3] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, 2001, pp. 123–130.

[4] M. Pantic and L. J. M. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *IEEE Transactions on Systems, Man and Cybernetics Part B*, vol. 34, pp. 1449–1461, 2004.

[5] S. Hongeng, F. Brémond, and R. Nevatia, "Representation and optimal recognition of human activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 818–825.

[6] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, August 2000.

[7] L. Berkowitz, *Aggression: Its Causes, Consequences, and Control*. Philadelphia: Temple University Press, 1993.

[8] V. Kettnaker, "Time-dependent hmms for visual intrusion detection," in *IEEE Workshop on Event Mining: Detection and Recognition of Events in Video*, 2003.

[9] H. Buxton and S. Gong, "Advanced visual surveillance using bayesian networks," in *IEEE International Conference on Computer Vision*, 1995, pp. 111–123.

[10] D. Mahajan, N. Kwatra, S. Jain, P. Kalra, and S. Banerjee, "A framework for activity recognition and detection of unusual activities," in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2004)*, 2004.

[11] F. Cupillard, A. Avanzi, F. Brémond, and M. Thonnat, "Video understanding for metro surveillance," in *Proceedings of the IEEE International Conference on Networking, Sensing & Control*, vol. 1, Taipei, Taiwan, March 2004, pp. 186–191.

[12] S. A. Velastin, B. L. Maria Alicia Vicencio-Silva, and L. Khoudour, "A distributed surveillance system for improving security in public transport networks," *Meas Control, Special Issue on Remote Surveillance Measurement and Control*, vol. 35, no. 8, pp. 209–213, September 2002.

[13] Z. Yang, S. Fitrianie, D. Datcu, and L. Rothkrantz, *Advances in Artificial Intelligence for Privacy Protection and Security*. Word Scientific, Singapore, 2009, ch. 11: An Aggression Detection System for the Train Compartment, pp. 249–286.

[14] Z. Yang, "Train image calibration using the DLT model," Delft University of technology, Tech. Rep., 2006.

[15] D. E. Butler, V. B. Jr., and S. Sridharan, "Real-time adaptive foreground-background segmentation," *EURASIP Journal on Applied Signal Processing*, vol. 14, pp. 2292–2304, 2005.

[16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, pp. 167–256, 2005.

[17] P. Viola and M. Jones, "Robust real-time object detection," in *2nd International Workshop On Statistical And Computational Theories Of Vision Modeling, Learning, Computing, And Sampling*, 2001.

[18] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.

[19] J. M. Mandler, *Stories, Scripts and Scenes: Aspects of Schema Theory*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1984.

[20] R. Schank and R. Abelson, *Scripts, Plans, Goals and Understanding*. Hillsdale, New Jersey: Lawrence Erlbaum, 1977.

[21] J. C. Giarratano and G. D. Riley, *Expert Systems: Principles and Programming*, 3rd ed. PWS Publishing Company, 1998.