July 23-27 2007, Vienna, Austria
www.indin2007.org

INDIN 2007

CONFERENCE PROCEEDINGS
VOLUME 2

5th IEEE International Conference on Industrial Informatics

SIEMENS

TU VIENNA — Technische Universität Wien — Vienna University of Technology

OVE — o'neida — IEEE — ies

Hidden Markov Models for Traffic Observation

Dietmar Bruckner, *Member, IEEE,* Brian Sallans, Gerhard Russ

***Abstract* — An automated method for traffic monitoring based on statistical models of sensor behavior in combination with a Markov model of monitoring points is described. A model of normal traffic flow is automatically constructed. The model's structure and parameters are optimized using a mini-batch model-merging and parameter-updating algorithm. Incoming velocity vectors are conveyed to the model and the most probable path through the tunnel's Markov model is computed. An alarm is generated when the sensor values have a low probability under the model. The performance, strengths and weaknesses of the automated traffic flow analysis system are discussed.**

I. Introduction

The Semantic Concept Recognition System (SCRS) is a system designed to extract "scenarios" from sensor data [1], [6]. A "scenario" is a sequence of multiple sensor values which could be understood by a person to form a single conceptual unit. For example, "drive to the airport" includes a huge number of sensory inputs and decisions, but we condense this entire concept into a single semantic "scenario" with a brief label.

The SCRS is designed to extract such scenarios from sensor data. This is done by detecting the statistical co-occurrence of sensor values or statistically-significant sequences of repeating sensor values. The system is based on statistical time-series models; in particular various types of hidden Markov models [10]. The goal is to condense the huge amount of sensor information currently available into a form which can be understood and quickly acted upon by human operators.

Although the SCRS was originally designed to extract scenarios from building automation systems [1], [6], it is a general tool which can be used in a number of different application areas. In this paper we describe the implement-tation of a traffic analysis system to promote safety in tun-nels using the SCRS. We implemented and tested a model structure based on HMMs with Gaussians as emission models. The traffic situation of tunnels is used as the test environment. Unusual traffic situations should be automati-cally detected by the system.

Manuscript received January 25, 2007.

Dietmar Bruckner is with the Vienna University of Technology, Institute of Computer Technology, 1040 Vienna, Austria. (corresponding author phone: 0043-1-58801-38423; fax: 0043-1-58801-38499; e-mail: bruckner@ict.tuwien.ac.at).

Brian Sallans and Gerhard Russ are with the Austrian Research Centers (e-mail: {brian.sallans,gerhard.russ}@arcs.ac.at).

In this case study a modified scenario concept is used. The entire traffic flow through the tunnel is seen as the sce-nario. The scenario consists of a number of paths - or sub-scenarios. These paths - generated from the actual data - can be more or less normal over time. The main advance com-pared to systems that observe the velocities at a particular point in the tunnel (seen from a single camera) is that this system has the overview of the traffic situation of the whole tunnel and can therefore deliver more informative alerts.

II. Hidden Markov Models

Hidden Markov models (HMMs) are used where it is not possible or useful to directly model observation sequences, but rather to model the underlying source for the change in observations. The following sections give an introduction into HMMs and their most useful algorithms [3].

1) Markov chain: The discrete time Markov Chain of 1st order is defined as $P(Q_{t+1} = q_{t+1} | Q_t = q_t, Q_{t-1} = q_{t-1}, ..., Q_0 = q_0) = P(Q_{t+1} = q_{t+1} | Q_t = q_t)$, where Q_t is the random variable at time t and q_t is a variable for some state at time t. This means that the probability for being in some state at some time is only dependent on the previous state.

2) Hidden Markov model: If the states of a process are directly observable, then we can model the process with a Markov model. Under some circumstances the process we want to model is not described sufficiently by a Markov Model. Consider a situation where you can measure - or observe - some value, but you would like to infer from those observations the driving force behind the values. In those cases, Hidden Markov Models are used. Their states cannot be directly observed, they are hidden. Each state has a probability distribution over some or all possible output symbols. In other words the Hidden Markov Model extends the Markov Model by emission probability distributions. The complete definition of a HMM is: A Hidden Markov Model is a variant of a finite state machine having a set of states Q, a transition probability matrix A, an output alphabet Σ, a confusion or emission probability matrix B and initial state probabilities π. The states are not observable and therefore called hidden. Instead, each state produces some output symbol according to the emission probability distribution (B). Characterizing for HMMs are:

- The number of states N
- The number of possible output symbols M
- The transition probability matrix $A = \{P_{ij}\}$

$P_{ij} = P(Q_{t+1} = j|Q_t = i),\ 1 \le i, j \le N,$

- An emission probability distribution in each of the states $B = \{b_{ik}\}$
  $b_{ik} = P(O_t = k|Q_t = i),\ 1 \le i \le N,\ 1 \le k \le M$
- Finally, the initial state distribution vector $\pi = \{\pi_i\}$
  $\pi_i = P\{Q_0 = i\},\ 1 \le i \le N$

III. HIDDEN MARKOV MODEL ALGORITHMS

After having selected the HMM to model a specific process, there are three possible tasks to accomplish with it [10].

1) Inferring the probability of an observation sequence given the fully characterized model (evaluation).

2) Finding the path of hidden states that most probably generated the observed output (decoding).

3) Generating a HMM given sequences of observations (learning).

In case of learning a HMM, *structure learning* (finding the appropriate number of states and possible connections) and *parameter estimation* (fitting the HMM parameters, such as transition and emission probability distributions) must be distinguished.

In our Semantic Concept Recognition System we used the following algorithms for the three tasks:

1) Forward algorithm: Consider a problem where we have different models for the same process and a sample observation sequence and want to know which model has the best probability of generating that sequence. This task is accomplished by the Forward Algorithm. As an example in a surveillance application we will have a set of possible semantic concepts like “person walking” or more abstract “normal day in an office”. Given a sequence of values we want to know which symbol could be the most probable cause for the sequence we see.

2) Viterbi algorithm: The Viterbi algorithm addresses the decoding problem. Thereby we have a particular HMM and an observation sequence and want to determine the most probable sequence of hidden states that produced that sequence.

3) Baum-Welsh algorithm: This algorithm addresses the third - and most difficult - problem of HMMs: to find a method to adjust the model’s parameters to maximize the probability of the observation sequence given the model. Unfortunately, there is no analytical way to accomplish this task. All we can do is locally optimize the probability of the observation sequence given different models.

IV. SURVEILLANCE SYSTEMS FOR TUNNELS

Tunnels play a crucial role in the importance of the transport sector for Europe's economy. In recent years the safety risks of operating tunnels have increased as tunnels age and are used more intensively. Controlling traffic in tunnels is a complex and challenging task with serious requirements, due to unique tunnel conditions (illumination, environment, intrinsic characteristics of the tunnel) and the small timeframe available to respond correctly to problems. When an incident occurs, tunnel operators have little time to recognize the incident, verify its nature, and react properly. Depending on the magnitude of the incident (e.g. driving in the wrong direction, presence of fire or smoke, or accidents between vehicles), tunnel operators must notify the proper authorities (police, roadway authorities, drivers, etc.), start standard procedures, and activate alarms among many other tasks. Taking the correct measures during the first few minutes after an incident is crucial to ensuring the safety of the people involved.

As a consequence, it is required that operators pay careful attention during the monitoring task. To facilitate this task, many road tunnels are already equipped with video systems allowing tunnel operators to supervise tunnel activities. These video systems operate 24/7, generating a huge amount of information, which cannot be completely supervised by the operators at all times. Further, the length of the tunnel complicates the problem because a longer tunnel implies more cameras, and therefore more information. The result is an increase in the demand for automatic or semi-automatic tools to aid tunnel operators in detecting and managing abnormal behaviors and unexpected events (see for example [2]).

Automatic traffic scene analysis has gained interest in the context of advanced transportation. In recent years, and as a result of the progress in intelligent transportation systems and advances in hardware and software, many potentially reliable and efficient new incident detection methods have emerged. Automatic incident detection has received the most attention, and different areas such as artificial intelligence, computer vision, neural networks, fuzzy logic, and video image processing contribute with a variety of algorithms to this new field.

The research in computer vision applied to intelligent transportation systems is mainly devoted to providing the transportation systems with situational awareness. A combination of computer vision methods and video technology has the potential to detect all major incident types, as well as measure statistical information such as speed and vehicle classification. Advantages of video-based systems are a higher detection rate with a shorter mean detection time and the simplified recording of raw data, among many others. However, traffic lights, reflections, and varying weather conditions are still challenges for video image analysis systems. In tunnels, reflections and low illumination conditions are intrinsic problems that need to be overcome. The ability to detect incidents like slow-moving traffic, traffic jams, and to classify moving objects is demonstrated by previous research work ([4], [5], [7], [8]) and commercial systems like ABT2000[1], INVIS[2], VisioPad[3], Traffic Analysis System[4], Autoscope[5], Video Trak 910[6], SiADS - SITRAFFIC[7], and Traficam[8] among many others - this list is not comprehensive.

The above-mentioned systems focus on the analysis of video to provide information about traffic flow and conditions. Additional software then uses this information to analyze the traffic conditions and draw conclusions. In this work we focus on the second task, analysis, and assume that reliable information about vehicle speed is already available. To the best of our knowledge, no previous work has been reported on the analysis of traffic data with statistical generative models (SGMs) and HMMs in tunnels.

The goal of this work is to automatically recognize unusual traffic-flow conditions without the need of pre-programmed rules, user-entered parameters, or experienced and alert operators. The system observes sensor data over time, constructs a model of "normality", and issues error alerts when sensor values - or combinations thereof - vary from normal. The result is a system that can recognize abnormal traffic activity with minimal manual configuration of the system. Further, if sensor readings vary or drift over time, the system can automatically adapt itself to the new "normal" conditions, adjusting its error criteria accordingly.

The system, called SCRS (Semantic Concept Recognition System) [1], which was initially investigated for use in building automation systems to detect human behavior, was tested with data generated from a traffic simulator. This article presents the results of this trial, highlights the strengths and weaknesses of the automated system, and suggests future areas of improvement.

V. System Structure

The goal of the SCRS model is to automatically discriminate system behavior in a running dynamic system (in this case a tunnel traffic system). It does this by learning about the behavior of the system and by observing data flowing through the system. The SCRS builds a model of the sensor data in the underlying system, based on the data flow. The model comprises not only SGMs describing the possible sensor values but also a model of the underlying events that cause a change in system behavior. From that model, a HMM, the system can identify recurring scenarios - patterns within the sensor values - with slightly varying sensor values represented by the SGMs that model the emission probability distributions of the HMM. The system is also capable of launching an alarm in case of the occurrence of new scenarios or variations within scenarios with very low probability under the model.

We use a set of statistical generative models to represent knowledge about the system under consideration. A statistical generative model takes as input a sensor value, status indicator, time of day, etc., and returns a probability between zero and one. Additionally, HMMs can be queried to deliver the most probable path through the model. In our case the most probable path can be interpreted as a similar (the best matching) traffic situation compared to the current one, which has already been learned.

Using SGMs has several advantages. First, because the model encodes the probability of a sensor value occurring, it provides a quantitative measure of "normality", which can be monitored to detect abnormal situations. Second, the model can be queried as to what the "normal" state of the system would be, given an arbitrary subset of sensor readings. In other words, the model can "fill in" or predict sensor values, which can help to identify the source of system (mis-)behavior, the unusual traffic situation. Third, the model can be continuously updated to adapt to sensor drift or to slightly changing operation conditions - varying speed limits due to road works for example - of the system.

For the application described in this case study, HMMs were used with Gaussian models as emission probability distributions. It is not necessary to use a mixture of Gaussians for the emissions, because each Gaussian belongs to a certain state of the HMM. Under this point of view, the whole system behaves like a set of mixtures of Gaussians, but the priors of the mixture distribution - coming from the transitions - vary with respect to the past.

In this surveillance application the idea of scenarios is used in a different way than in building automation applications of the SCRS. Here the length of the state chain is fixed with the number of cameras C in the tunnel (the number of cameras in the tunnel is specified by national authorities; in Austria, the distance between cameras is recommended – de facto fixed – to 200m for short tunnels (<500m) and 120m otherwise [9]).

Each "snapshot" of velocity values (a vector with one value per camera, even if it is not new) is said to be a sensor value chain. The chains of C values are then fed into the (empty) model and during a procedure of 3 steps (see also [1], [6]) the parameters of the model are updated:

1. Comparison of the chain's beginning/end
2. Merging of identical states
3. Merging of consecutive states

[1] www.artibrain.at
[2] www.invis-security.com
[3] www.citilog.fr
[4] www.crs-its.com/main.htm
[5] www.imagesensing.com, www.autoscope.com
[6] www.peek-traffic.com
[7] www.siemens.com/page/1,3771,1129794-0-14_0_0-10,00.html
[8] www.traficon.com

These steps ensure the creation of a HMM with a fair number of states. Too many states make the model very specific to particular situations and assign a very low probability for previously unseen situations whereas too few states correspond with a very general model structure that would not be able to give reliable information about the probability of the current situation.

The system architecture is depicted in Fig. 1.

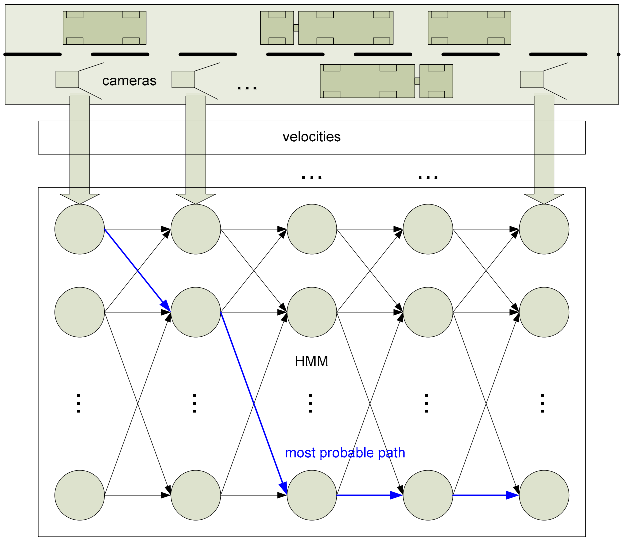

Fig. 1: System Structure of the Surveillance System: Cameras are mounted equidistantly on the ceiling of the tunnel. Each time a camera recognizes a new vehicle and computes its velocity, a vector with the velocities of all cameras is passed to the HMM. The HMM on the one hand computes the most probable path and its log-likelihood and on the other it saves the vector for later incorporation into the model.

VI. System Adaptation

One important parameter of the SCRS for the analysis of the traffic situation is the number of velocity vectors to learn. In the current implementation the system takes the first 10 velocity vectors to form an initial model. Afterwards each new vector is compared to this model. The SCRS computes the Viterbi path for the new vector and the path's likelihood. After additional 10 vectors are received, the model computes a second model for the latest values. Afterwards, these two models are merged according to their priors. This procedure ensures that the model will learn new traffic situations, but with a low prior. The prior is the ratio of the number of new values to the number of values already seen. Fig. 2 shows the outcome of the just presented procedure for the first 450 velocity vectors. In a later phase - when the model is assumed to already have learned all possible states - the Baum-Welsh algorithm can be used to adapt the model parameters.

In the following 3 Figures the impact of the learning rate – the number of vectors which is taken for updating the model – is investigated. Fig. 3 depicts a simulation run where every 2 new values are merged into the model. The likelihood is therefore not as “flat” as shown in Fig. 2, but peaks are not as broad, because the model adapts very quickly to new conditions.

The simulator starts to simulate a traffic jam after about 450 vectors; therefore the likelihood decreases dramatically (about -30 per camera). In Fig. 3, the likelihood increases to about -13 after the first few vectors of the jam, indicating a decrease of the likelihood caused by a traffic jam of about -6 in the case of very fast learning.

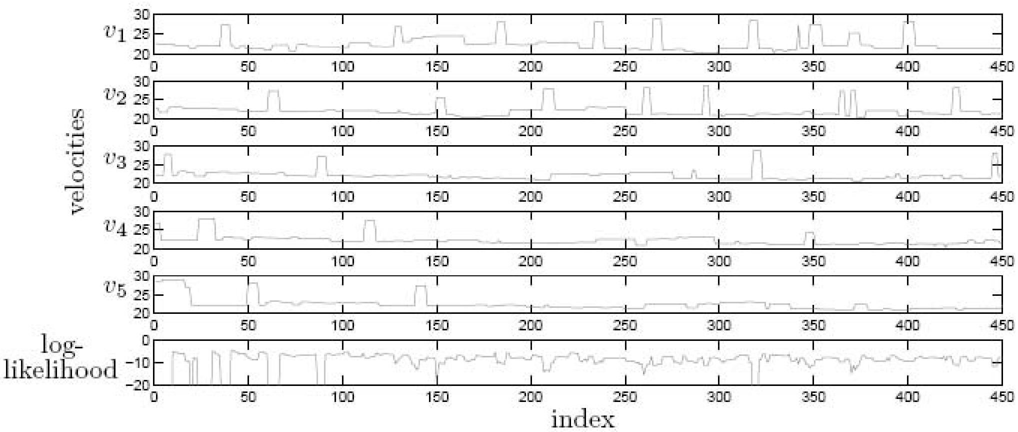

Fig. 2: Simulated velocity values of a tunnel with 5 cameras. The bottom part of the figure is the overall likelihood of the model. At the beginning each new situation causes the system to drop in the likelihood, while it stays around -7 to -10 during normal conditions. The large drop of the likelihood at index 320 is caused by the coincident occurrence of large velocity values at camera 1 and 3.

The next simulation run had a lower learning rate. Each new 20 data vectors are merged into the system. The result is shown in Fig. 4. We recognize a longer learning phase with likelihood drops, but a higher likelihood in the later phase for common situations (about -4 to -5). This observation is continued in Fig. 5. The average likelihood sinks, but the values for common situations can be very high (up to -1.5).

This behavior can be explained by having the structure of the model in mind: an HMM with Gaussians to model the velocities. When many values are used to learn each of the Gaussians, the variance becomes low and therefore the Gaussians become narrow. A narrow Gaussian gives very high values for "fitting" velocities and low for unusual ones. However, for ensuring correct detection of situations the model needs many sensor values to learn about all possible combinations thereof.

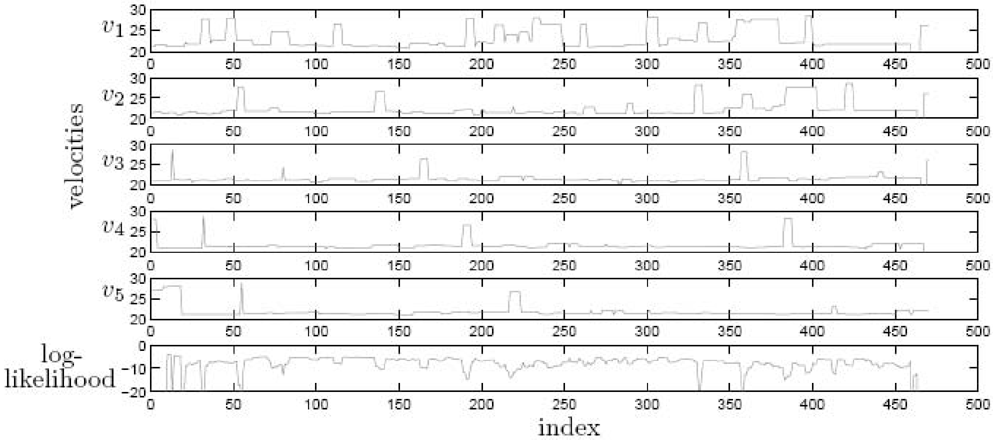

Fig. 3: Simulated velocity vectors for the SCRS with a very high learning rate. The system adapts very quickly to new conditions.

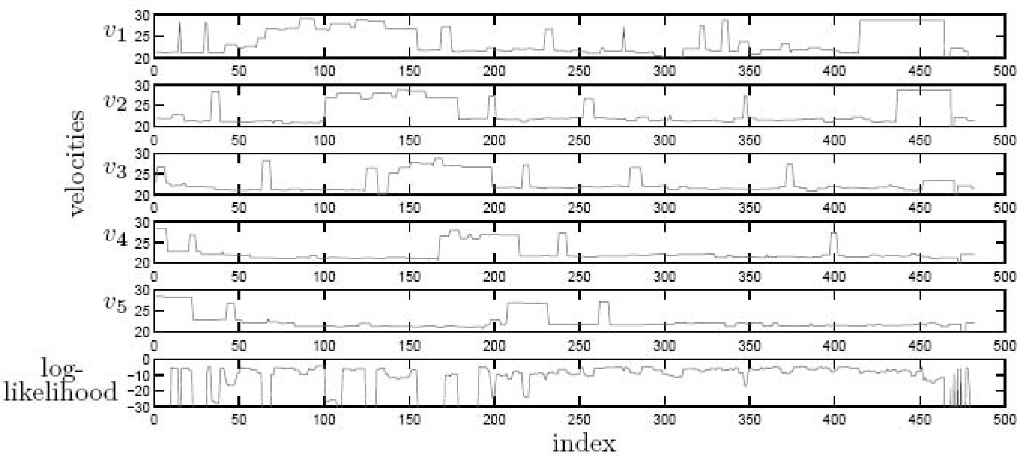

Fig. 4: Simulated velocity vectors for the SCRS with a medium learning rate. The time in which the likelihood drops because of new situations is very long. On the other hand the likelihood for common situations is higher than in the case of faster learning rates.

This is because it learns not just about parameters of one camera, but the overall situation. The simulator used for these experiments was designed to deliver approximately 500 velocity values, after 450 of which it simulated a traffic jam. The 500 velocities are enough to make observations about the system behavior, but too few to fully learn the model. The work presented here could be supplemented with a more simulated data or with actual data from real tunnels, because the "normal" traffic situation in a particular tunnel may not be usual for another one.

The model was capable - independent of the learning rate - to successfully detect the traffic jam incident. Because the model uses the Viterbi algorithm to choose from already seen traffic situations, it would be easy - as it is a built-in function of the model - to detect at which camera the problem occurred (in the sense that it finds the corresponding path). If - during some negotiation procedure - the user has already defined a reaction for such an incident, the system could automatically direct the user's attention to that camera.

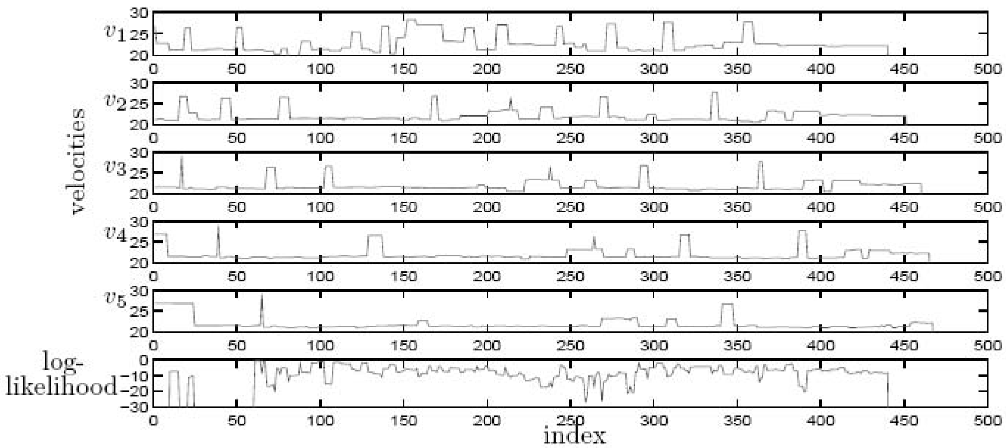

Fig. 5: Simulated velocity vectors for the SCRS with a slow learning rate. The differentiation between very common and uncommon situations becomes sharper. The likelihood for common situations reaches -1.5 several times, but is lower on average than in the case of faster learning.

Another advantage of the system comes to light in case the velocities at all cameras are within the values seen before, but the simultaneous occurrence of groups of values or the sudden change of values can indicate an unusual situation and therefore launch an alarm.

VII. Conclusion

The work presented in this paper describes a test of the HMM approach for the automatic detection of abnormal traffic situations in tunnels. The SCRS system can automatically build a model of normal traffic situations. Each path through the model represents a particular traffic situation for the whole tunnel. To ensure that even unusual situations are represented correctly, auxiliary models of newly arrived sensor values are built, and these are merged with the original model. The SCRS possesses the clear advantage of providing an overview of the whole tunnel, while traditional systems use the values of each camera independently.

This work shows that the SCRS is capable of differentiating between usual and unusual traffic situations. With the limited data available, we have shown that the method can detect system-wide disturbances, and offers a clear advantage over systems which consider each camera alone. The system has shown its ability to give a quantitative measure of the likelihood of the traffic situation, which can then be used by other systems like fire or smoke detection systems, systems for counting vehicles or systems to detect crashes and jams. The detailed study of the system using a comprehensive set of simulations for various alarm conditions must be left as future work.

References

[1] Bruckner, D.; Sallans, B.; Russ, G.: *Probabilistic Construction of Semantic Symbols in Building Automation Systems.* In: Proceedings of the 5th IEEE International Conference on Industrial Informatics, 2006

[2] Strobl B., et. al.: *Vitus - Tunnel Safety through video-based Analysis.* In: Proceedings of the 13th World Congress and Exhibition on Intelligent Transport Systems and Services, 2006

[3] Bishop, C. M.: *Neural Networks for Pattern Recognition.* New York NY.: Oxford University Press Inc., 1995

[4] Bertozzi, M.; Broggi, A.; Fascioli, A.; Tibaldi A.; Chapuis, R.; Chausse, F.: *Pedestrian Localization and Tracking System with Kalman Filtering.* In: IEEE Intelligent Vehicles Symposium, 2004

[5] Cucchiara, R.; Piccardi, M.; Prati, A.; Scarabottolo, N.: *Real-time Detection of Moving Vehicles.* In: Proc. of 10th International Conf. on Image Analysis and Processing, 1999, S. 618–623

[6] Bruckner, D.: *Probabilistic Models for Building Automation: Recognizing Scenarios with Statistical Methods*; Vienna, Austria : Institute of Computer Technology, Technical University of Vienna, 2007. – Dissertation thesis

[7] Remagnino, P.: *An Integrated Traffic and Pedestrian Model-Based Vision System.* In: Proceedings of the Eight British Machine Vision Conference, 1997, S. 380–389

[8] Viola, P.; Jones, M.: *Rapid Object Detection using a Boosted Cascade of Simple.* In: IEEE CVPR, 2001

[9] ASFINAG: *Planungshandbuch Videosysteme*. (2005), S. 76–77

[10] Rabiner, Lawrence R. ; Juang, Biing-Hwang: *An Introduction to Hidden Markov Models.* In: IEEE ASSAP Magazine 3 (1986), January, S. 4–16