Graz University of Technology

Institute for Computer Graphics and Vision

Master's Thesis

On-line Unsupervised Outlier Detection in Video Streams: A Quantitative Study

---

Georg Kummert

Graz, Austria, October 2011

Thesis supervisor

Univ.-Prof. DI Dr. Horst Bischof

Instructor

DI Dr. Peter M. Roth

TO MY PARENTS
SEFI AND FRITZ

Abstract

The ever growing number of surveillance cameras and the concomitantly increasing amount of video footage have led to a steady growth of research in the field of automated outlier detection in video. Typically this is thought of as a Computer Vision system taking over the tedious task of filtering unexpected events that may require attention of a human operator. Traditionally, mostly models based on object tracking were investigated, which are likely to break down in cluttered scenes with significant occlusions. To circumvent these problems methods based directly on low-level features like motion or texture have been proposed in recent years.

In this thesis an overview of the current state of the research on automated video surveillance based on-low-level features is given and three methods allowing for unsupervised, on-line outlier detection are investigated. Detailed analysis has led to significant performance improvements by algorithmic modification and extension of the original work. To simplify analysis a unified feature is used for all approaches and the most suitable confusion-matrix-based performance measure to assess detection performance is identified.

Two of the three methods are found to perform well on five datasets used in outlier detection literature. One in its original form, the other one improved by extensions proposed in this work: (i) Employing the best suitable distance measure for a given scene and (ii) maintaining multiple scene models instead of one to enhance the modeling of local normality. The third method can be applied to a limited extent, but still significant performance improvements are made due to algorithmic adaptations.

Keywords. Outlier detection, unsupervised on-line learning, model of normality, automated video surveillance, CCTV.

Kurzfassung

Die immer größer werdende Zahl von Überwachungskameras und der damit verbundene Zuwachs an Videomaterial hat zu einem stetigen Wachstum des Gebiets der automatischen Videoüberwachung geführt. Hierbei übernimmt ein Computer-Vision System die für Menschen anstrengende Aufgabe der Filterung von unerwartetem, eventuell sicherheitsrelevantem Verhalten. Bisherige, auf Objektverfolgung basierende Verfahren scheitern an Szenen, in denen höhere Objektdichte zu Verdeckungen führt. Dies hat in den letzten Jahren zu Methoden geführt, die direkt auf Low-Level-Merkmalen, wie Bewegung oder Textur, basieren.

Diese Arbeit gibt einen Überblick über den aktuellen Stand der Forschung auf dem Gebiet der automatischen Videoüberwachung basierend auf Low-Level-Merkmalen und untersucht drei Methoden, die unbeaufsichtigte, adaptive Erkennung von unerwartetem Verhalten ermöglichen. Detaillierte Analyse hat zu signifikanten Verbesserungen der Originalmethoden geführt, einerseits durch algorithmische Modifikationen, andererseits durch Erweiterungen. Um die Analyse zu vereinfachen, wird ein einheitliches Bewegungsmerkmal verwendet sowie das zum Vergleich am besten geeignetste Maß basierend auf Wahrheitsmatrizen identifiziert.

Zwei der drei Methoden liefern auf fünf Videos aus der Literatur gute Ergebnisse. Eine in ihrer ursprünglichen Form, die andere durch Veränderungen, die in dieser Arbeit vorgeschlagen werden: (i) Einsatz des bestgeeignesten Distanzmaßes für eine gegebene Szene und (ii) Verwendung von mehreren Szenemodellen, um die Modellierung von lokalen Szenegegebenheiten zu verbessern. Die dritte Methode ist zwar nur begrenzt einsetzbar, dennoch konnte durch Modifikation des Algorithmus eine signifikante Leistungssteigerung erreicht werden.

Schlüsselwörter. Erkennung von Ausreißern, Detektion von Anomalien, on-line Lernen, automatische Videoüberwachung

Danksagung

Mein Dank gilt zuallererst meinen Eltern, Sefi und Fritz, für die langjährige Unterstützung und die Ermöglichung meiner Ausbildung, und für ihr Verständnis die langen Jahre über, in denen ich das Studium hinter die beruflichen Interessen gestellt habe.

Ein Dankeschön geht an Prof. Horst Bischof für die Beaufsichtigung dieser Masterarbeit und die Unterstützung in terminlich engen Grenzen, und ein besonderes an Peter M. Roth für die fachliche Führung, die stets auch humorvollen Gespräche, die Unterstützung während der Umsetzung und letztlich das Korrekturlesen dieser Arbeit. Weiters danke ich all den Kollegen am Institut für Maschinelles Sehen und Darstellen, die mir mit Rat und Tat zur Seite standen.

Auch den vielen Kollegen, die ich im Laufe meiner bisherigen beruflichen Laufbahn kennenlernen durfte möchte ich danken – eure Kameradschaft ist einer der schönsten Aspekte des Daseins als Ingenieur.

Und schließlich danke ich dir, Sigrid, für das aufmerksame Korrekturlesen, die kompetente sprachliche Betreuung und den Rückhalt während der letzten Studienphase, vor allem aber dafür, dass du an meiner Seite gehst und mir Ratgeberin, Freundin und Gefährtin bist.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

……. ……..
date (signature)

Contents

1 Introduction **1**

1.1 Problem statement 2

1.2 Contribution 3

1.3 Outline 4

2 Literature Review **7**

2.1 Statistical methods 8

2.1.1 Parametric models 9

2.1.1.1 Spatial-temporal co-occurrence GMM 9

2.1.1.2 GMM using PCA and LPP subspaces 10

2.1.2 Nonparametric techniques 10

2.1.2.1 Unusual Event Detection Using Monitors 11

2.2 Neural network models 12

2.2.1 Self Organizing Map 13

2.2.1.1 On-line SOM 13

2.2.2 Growing Neural Gas 15

2.2.2.1 Incremental Neural Network 16

2.2.2.2 Computing trajectory prototypes using GNG 16

2.3 Bayesian network models 18

2.3.1 Probabilistic Latent Semantic Analysis 19

2.3.1.1 Anomaly Detection using hierarchical PLSA 20

2.3.1.2 PLSA scene segmentation 20

2.3.2 Latent Dirichlet Allocation . . . 21
2.3.2.1 Temporal Order Sensitive LDA . . . 21
2.3.2.2 Region LDA . . . 23
2.3.3 Hidden Markov Model . . . 24
2.3.3.1 Temporal Statistics of Spatio-Temporal Motion using HMM . . . 24
2.3.3.2 Markov chained LDA . . . 26
2.4 Other approaches . . . 26
2.4.1 Co-occurrences using Markov Random Fields . . . 26
2.4.2 Mixtures of Dynamic Textures . . . 27

3 Method Selection and Realization 29
3.1 Evaluation and selection . . . 29
3.2 Feature extraction and representation . . . 32
3.3 Implementation . . . 34
3.3.1 Anomaly Detection Using Multiple Fixed-Location Monitors . . . 35
3.3.1.1 Integration procedure . . . 36
3.3.2 Spatial-temporal co-occurrence GMMs . . . 37
3.3.2.1 Outlier detection . . . 38
3.3.3 Modified STCOG . . . 40
3.3.4 Self-Organizing Maps for Anomaly Detection . . . 42
3.3.4.1 Learning the TASOM . . . 43
3.3.4.2 Outlier detection . . . 44
3.3.5 Modified TASOM . . . 46
3.3.6 Baseline method . . . 48

4 Experiments and Results 49
4.1 Obtaining ground truth . . . 49
4.2 Datasets . . . 51
4.2.1 UMN crowd activity dataset . . . 52
4.2.2 UCSD pedestrian dataset . . . 53

4.2.3 Roadmarket dataset 55
4.2.4 Junction dataset 56
4.2.5 Underground train station exit platform dataset 58
4.3 Experiments 60
4.3.1 Experiment configurations 60
4.3.2 Evaluation of performance measures 61
4.3.3 Comparison of performance measures 63
4.4 Results 66
4.4.1 Original vs. modified variants 66
4.4.1.1 Adam's `HIST` approach 66
4.4.1.2 Shi's `STCOG` approach 68
4.4.1.3 Feng's `TASOM` approach 69
4.4.2 On-line vs. off-line performance 72
4.4.3 Best performers 75
4.4.3.1 `UMN crowd activity` dataset 75
4.4.3.2 `UCSD pedestrian` dataset 78
4.4.3.3 `Roadmarket` dataset 81
4.4.3.4 `Junction` dataset 85
4.4.3.5 `Exit platform` dataset 88
4.5 Summary 91

5 Conclusion and Outlook **93**
5.1 Conclusion 93
5.2 Outlook 95

List of Figures **97**

Bibliography **99**

Chapter 1

Introduction

Contents

1.1 Problem statement 2

1.2 Contribution 3

1.3 Outline 4

Our world has seen an enormous growth of surveillance in the past decades. Large networks of Closed Circuit Television (CCTV) cameras have been deployed in key infrastructure such as underground transportation and airports, and a massive spread of surveillance in all areas of the urban space is recognizable. In 2002 nearly one third (29%) of 1.400 publicly accessible spaces in the capitals of six European countries were found to operate a CCTV surveillance system. While the diffusion of CCTV in private, but publicly accessible spaces is similar throughout Europe, its extent in public areas differs significantly. In 2004, in more than 500 cities in Britain an estimated 40.000 cameras monitored public space compared to less than 100 cameras in around 15 German cities, and no open street CCTV in Denmark (Hempel and Töpfer [47]).

In practice only a fraction of installed surveillance cameras is ever watched in real-time. Reported screen to camera ratios lie between 1:4 and 1:78, and the ratio of operatives to screen can be as high as 1:16. Consequently, the majority of CCTV footage is only watched following an incident for investigative purposes rather than using it as a mechanism for real-time alerts during an event (Dee and Velastin [29], Adam et al. [1]).

Furthermore, it is practically acknowledged that an operative can monitor only 1-4 screens at a time, therefore in a typical installation with 100 cameras and 3 operatives the probability that a camera is actively monitored by an operator is approximately 3%. Moreover most current CCTV systems leave the decisions about which camera to monitor to the operators themselves, who often decide which camera to watch based upon the appearance rather than the behavior of the people on the screen. This can leave the system open to abuse (such as the targeting of minority groups) and has attracted the attention of human rights and anti-surveillance groups.

Also, health aspects have to be taken into consideration: Naturally, concentration time is a limiting factor – the optimal span for a person is generally about 25 to 30 minutes – breaks away from the screen are recommended (5-10 minutes each hour), fatigue, operational stress and change blindness greatly affect performance. And, CCTV operators suffer from the obvious problem of boredom: monotonously viewing hours of routinized, uneventful televisual images and – in the vast majority of surveillance situations – nothing happens (McCahill and Norris [70], Noyes and Bransby [81], Scott-Brown and Cronin [90], Smith [97], Wallace and Diffley [105]).

1.1 Problem statement

Apart from human limitations like for example poor concentration or boredom, the ever increasing number of surveillance cameras makes it impossible to have humans constantly monitor all captured video streams. It would be desirable to *filter* these large amounts of data in order to generate alerts, that is, identifying instances that need further attention from a human operator. This would enable a single operator to effectively supervise a large number of cameras. Algorithms performing such filtering with proper detection and false-alarm rates would allow for presenting or recording only the interesting or potentially anomalous sequences of the video feed and using the system as an *attention mechanism* (Tziakos et al. [102], Adam et al. [1]).

Traditionally, Computer Vision technologies applicable to the problem of automated surveillance are based on object tracking and a problem closely related to it, occlusion handling. Nowadays, however, it is widely agreed that this classical ap-

proach has some shortcomings (Zhan et al. [115]). Detection and tracking of individuals break down when the scenes get crowded and are likely to create false alarms due to the creation of false targets (shadows, clouds, ...) or tracking failure. For this reason, in recent years a variety of methods directly based on low-level features instead of tracking have been proposed. Conventional systems deploying trajectories are usually organized in the following hierarchical way: low-level feature extraction, tracking and occlusion reasoning, scene modeling, behavior analysis and finally event detection. With low-level feature based systems tracking and occlusion reasoning is omitted. This methodology allows to circumvent the problems associated with tracking approaches and, additionally, leads to systems that are generally simpler in terms of computational complexity.

While several reviews of approaches based on object tracking are available that outline the methodology as well as summarize the proposed algorithms, e.g.Dee and Velastin [29] or Morris and Trivedi [73], to our knowledge neither an overview on outlier detection methods directly based on low-level features nor a comparison regarding their individual performance exists.

Therefore, the intent of this work is twofold: On the one hand, to give a literature review on low-level outlier detection methods and on the other, to conduct a comparison of a suitable subset of these methods.

1.2 Contribution

The contributions of this work to various areas of outlier detection on video data are detailed in the following:

- **Literature review.** Recent work in the area of outliers detection based on low-level features is summarized and categorized. In addition to a description of each work, the theoretical foundation is outlined briefly to give a self-contained overview.
- **Extension of existing work.** A detailed description of the implementation of a number of methods chosen based on a list of evaluation criteria is given. Additionally, several extensions to enhance their detection performance are proposed. These include the modification of the modeling algorithms as well as

the extension of the filtering algorithm used to eliminate volatile outliers. On one of the methods, the usage of various difference measures plugged in the modeling algorithm is proposed and the maintenance of multiple models to improve the representation of normal scene behavior is introduced.

- **In-depth performance analysis.** First, a suitable measure to assess outlier detection performance is identified by evaluation of experimental results. Finally, using this performance measure, a comparative analysis is conducted that covers diverse aspects of the methods, like original vs. modified method performance and off-line vs. on-line performance.

1.3 Outline

This thesis is organized as follows: in Chapter 2 a self-contained literature review is given, methods are categorized and their theoretical background summarized. Three basic categories of approaches are identified: methods based on statistical models, methods based on neural network models and, finally, those based on Bayesian network models. Moreover, two approaches not assignable to these categories are presented.

Next, Chapter 3 presents a list of – partly qualitative – criteria to guide the selection of methods to be implemented. Based on the type of low-level features used, the localization capabilities, the method's efficiency and most importantly, its on-line adaptiveness, three methods are chosen and their implementation as well as modification to improve performance are described in detail. In addition, a unified feature extraction procedure is introduced, which is used as a common feature supplier to simplify method comparison in the final performance comparison. To that end, the different motion feature extraction procedures are compared and the most expressive one is identified.

Chapter 4 describes five video datasets used in the literature and their subsequent usage to evaluate the implemented methods. First the procedure used to obtain ground truth for these datasets is presented, then a comparison of different statistics based on confusion matrix statistics is conducted to identify the most appropriate performance measure. Using this measure, the five video datasets are tested and various aspects examined in detail. Primarily, the comparison is focused

on the performance improvements possible with the modified methods and the performance differences between on-line and off-line variants. At last the best performers for each dataset are presented.

Finally, Chapter 5 gives a summary of findings and insights gained in this work and an outlook of possible future work.

Chapter 2

Literature Review

Contents

2.1 Statistical methods 8
2.2 Neural network models 12
2.3 Bayesian network models 18
2.4 Other approaches 26

In this Chapter an overview of different outlier detection techniques applied in the literature is given. Methods are categorized according to the main principle used: statistics, neural networks or Bayesian networks. Although outlier detection is a wide field and the approaches proposed to solve it are diverse, the basic scheme depicted in Fig. 2.1 is common to all methods.

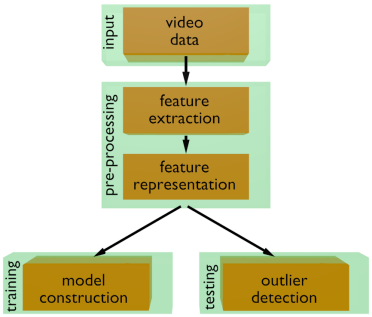


Figure 2.1: Basic outlier detection diagram: Features are extracted and represented appropriately before they are either (1) used for model construction while *training* or (2) used for outlier detection while *testing*.

First, features are extracted and represented in a suitable form. Then, while *training*, a model of normality is fit to the extracted feature instances, whereas while *testing* the deviation of observed data instances as measured by the model is used to carry out outlier detection.

The number of methods proposed in the wide field of outlier detection is large, therefore several restrictions were applied when selecting methods to narrow the scope of this review:

- **Feature-based approaches.** Feature-based approaches are described to avoid shortcomings in connection with object-based approaches, i.e., methods that rely on object tracking.
- **Video context.** We describe techniques used in video context only.
- **Representative methods.** Whenever a larger quantity of methods is available, only a few representative methods are described.

In the following overview, each method is illustrated by (i) the feature extraction and representation used and (ii) the model of normality deployed to describe scenes. The principle used to model normality is used to classify a method into one of three categories: statistical models, neural network model and Bayesian network models. Whenever a presented approach employs more than one technique in modeling data, the central technique is used to categorize the method. Additionally, two methods are presented that do not fit into any of these three categories.

2.1 Statistical methods

When using statistical methods, given data is fit to statistical models during training. Subsequently, statistical inference tests are applied to decide whether an unseen instance is an outlier. Instances that have a low probability of being generated by the learned models are declared anomalies. Hence statistical methods are deployed under the assumption that normal data instances populate high probability regions of a model, while anomalous instances lie in regions with low probability. Chandola et al. [26] differentiate two different method categories:

- **Parametric methods.** Training data is modeled using stochastic distributions and, subsequently, outliers are labeled depending on their relation to this model. Parametric methods assume certain properties of the data, i.e., regarding its smoothness. For this reason, parametric models have been found to be unsuitable in a range of applications as they require extensive knowledge of the problem and do not necessarily fit real data.
- **Nonparametric methods.** When using nonparametric statistical methods, a model's structure is not defined a priori, but is determined from the given data instead. In addition, nonparametric methods do not assume knowledge about the underlying distribution and make fewer assumptions regarding the data.

2.1.1 Parametric models

Parametric techniques assume that the normal data is generated by distributions defined by parameters that can be estimated from the given data. The most prominent parametric models are mixture models, which use a mixture of parametric statistical distributions to model data. An important technique belonging to this category is the Gaussian Mixture Model (GMM), a probability density function consisting of a weighted sum of Gaussian distributions. During training, a GMM's parameters are estimated from the training data. A popular algorithm to estimate these parameters is the iterative Expectation-Maximization (EM) algorithm.

Singh and Markou [96] argue that the main limitation of GMMs is that given high data dimensionality, a large number of training samples is needed to estimate all free parameters of each Gaussian's mean and covariance matrix. When training data is limited, fitting a model may be impossible. Additionally, the computational effort is higher than that of nonparametric techniques. In the following two methods using GMM's are detailed.

2.1.1.1 Spatial-temporal co-occurrence GMM

Shi et al. [95] propose a spatial-temporal co-occurrence Gaussian mixture model (STCOG). The video is divided into local, non-overlapping regions and phase correlation introduced by De Castro and Morandi [28] is used to calculate the local velocities in each region of each frame. The velocities of L subsequent frames combined

from two adjacent regions form what is termed a pair-wise cuboid. For each pair-wise cuboid in a four-neighborhood one GMM is trained using an online K-means approximation similar to that used by Stauffer and Grimson [99].

When detecting, the co-occurrence probability for each cuboid is calculated and averaged over the four-neighborhood. This probability is then thresholded to detect outliers while the parameters of the model are still updated.

2.1.1.2 GMM using PCA and LPP subspaces

Tziakos et al. [102] deploy multiple non-overlapping local motion detectors similar to those Adam et al. [1] use, although the monitors are not directly adjoined. Motion vectors are extracted using block matching and are grouped spatially. Finally, the matrix of motion vectors is concatenated to form a high dimensional feature vector. Training and outlier detection take place in a subspace. Two methods are compared: Principal Component Analysis (PCA) and Locality Preserving Projections (LPP) introduced by He and Niyogi [45]. The results show that due to locality preserving characteristics LPP is insensitive to abnormal instances in the training data as long as normal instances are the majority of the sample. This behavior is not to be expected in global methods like PCA which are sensitive to outliers. Another shortcoming of PCA compared to the LPP approach is that it assumes that the direction of the maximum variance is also the direction of importance.

After subspace projection, a GMM is trained for each monitor using Expectation Maximization (EM) while determining the number of components using the Bayesian Information Criterion (BIC). Outlier detection is then carried out by mapping new instances into the subspace and thresholding their probability of being generated by the GMM. Finally, to reduce false alarms, the integration procedure introduced by Adam et al. [1] is applied. See Chapter 3.3.1.1 for a detailed description of Adam's integration procedure.

2.1.2 Nonparametric techniques

When nonparametric techniques are used the model structure is not defined beforehand but determined from the data. The simplest nonparametric method is to use histograms, a technique also referred to as being *frequency-based* or *counting-based*.

It consists of two steps: First, a histogram is built based on the training data. Second, outlier detection is conducted: either an outlier is reported if a test instance does not fall in one of the histogram bins or if it falls in a bin with a low frequency. Evidently, the width of the bins is the key for anomaly detection. A histogram-based outlier detection method is discussed as follows.

2.1.2.1 Unusual Event Detection Using Monitors

Adam et al. [1] extract optical flow at fixed spatial locations. To avoid the aperture problem the following procedure is used to compute reliable motion:

1. The Sum of Squared Differences (SSD) error between two patches with different displacements is calculated.
2. The resulting error matrix is transformed into a probability distribution – similar to the approach of Rosenberg and Werman [87].
3. Either angular or radial binning is applied to aggregate the distribution.
4. Then, the most likely displacement is checked using an ambiguity test. If the ambiguity exceeds a predefined threshold, the monitor does not produce a motion output.

Afterwards, motion features are binned depending on the video processed: if motion direction is prevalent in the video radial binning is applied, otherwise the magnitude of the motion is binned. Each of the monitors uses a cyclic, fixed length buffer to sequentially collect the observations. Once the buffer is full, the oldest observation is replaced by the new one. When operating in detection mode, the likelihood of a new observation is evaluated using the observations that are currently in the buffer. If its probability drops below a predefined threshold an alert is triggered.

Finally, alerts produced by local monitors are integrated using a simple integration procedure before an event is reported to the user. A detailed explanation of the procedure can be found in Chapter 3.3.1.1.

According to Adam et al. [1] an advantage of their approach is the use of low-level measurements at fixed spatial locations instead of object-based measurements.

Moreover, they claim their algorithm to be computationally undemanding, highly adaptive, simple and intuitive. On the other hand, they point out that the method cannot detect events that are characterized by an unusual sequence of short-term normal actions. Tziakos et al. [102] substantiate this assessment by adding that although the approach has a low complexity, it only considers abnormality based on recent (temporal) motion history and thus detected abnormal events might not be consistent, e.g., what is abnormal at a certain time instant might not be abnormal in the future.

2.2 Neural network models

The second category of approaches described in this overview comprises methods based on neural networks to model scene normality. These techniques operate in two basic steps:

1. **Training.** The neural network is trained on normal data to learn different normal classes. During training, the neurons are adapted to represent the training data in a compressed way.

2. **Testing.** A test data instance is provided as an input to the neural network. If the network accepts the test input, it is normal, otherwise it is an anomaly. As a measurement of acceptance the *reconstruction error* – the ability of the network to reconstruct a given input – can be used. A predetermined threshold can then be employed to detect abnormal instances.

One of the advantages of using neural networks for outlier detection is that during training no a priori information is needed on data distribution and no specific parameters related to data need to be set.

For outlier detection the neural network subclass of *topology preserving feature maps* is used in the literature. Topology preserving feature maps project input data instances onto a network of neural units such that similar instances are projected onto adjacent units and, vice versa, adjacent units code similar instances. Moreover, Martinetz [68] shows that topology preserving maps can be learned using Compet-

itive Hebbian Learning (CHL)[1] in combination with a vector quantization method. In this work methods based on two different topology preserving feature maps are presented: Self-Organizing Maps (SOMs) and Growing Neural Gas (GNG).

2.2.1 Self Organizing Map

The first of the topology preserving feature maps applied to the problem of outlier detection in the literature is the SOM, which is introduced in Kohonen [56]. Theoretical background information on SOM is found in summary box Sum. 2.1. An on-line variant for outlier detection is illustrated in the following.

2.2.1.1 On-line SOM

Feng et al. [33] propose an on-line SOM to model crowd scenes, which is based on the Time Adaptive Self-Organizing Map (TASOM) introduced by Shah-Hosseini and Safabakhsh [92]. To improve the SOM's capabilities of handling non-stationary input distributions and changing environments the learning rate $\alpha(t)$ and the size of the neighborhood $N_c(t)$ are adapted independently for each neuron.

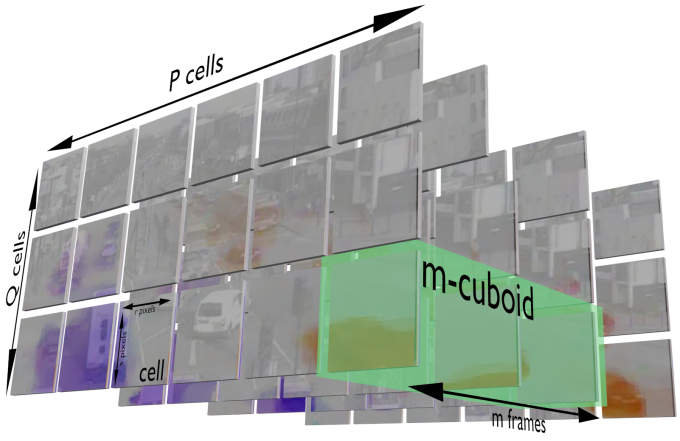


Figure 2.3: Spatial-temporal division of the video: The image plane is divided into $P \times Q$ non-overlapping cells of $r \times s$ pixel size. m temporarily consecutive cells form a volume referred to as m-cuboid in this work.

[1]The basic principle that governs the change of interneural connection strength has been formulated by Hebb [46]. According to Hebb's postulate a presynaptic unit increases the strength of its synaptic link to a postsynaptic unit if both units are concurrently active, i.e., if both activities correlate.

Self-Organizing Map

A SOM – depicted schematically in Fig. 2.2 – is a $K \times L$ grid of neurons that is trained using CHL, its size is fixed and has to be defined beforehand. Weights m_i associated with the neurons $\boldsymbol{\omega_i}$ have the same dimension as the input data instances and are properly initialized, i.e., random initialization will often suffice.

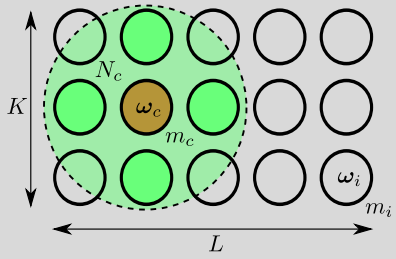


Figure 2.2: Scheme of a SOM while *training*.

Training. Given a data instance x and a distance function $d(x, m_i)$, the "winner" neuron m_c is identified such that

$$d(x, m_c) = \min_i \{d(x, m_i\}.$$

A – possibly time-variable – neighborhood $N_c(t)$ of the winning node m_c is defined by a width or a radius. The net is trained by adapting the weight vectors m_i of the neurons in N_c, all other neurons are left unchanged:

$$m_i(t+1) = \left\{ \begin{array}{c} m_i(t) + \alpha(t)[x(t) - m_i(t)] \\ m_i(t) \end{array} \right\} \begin{array}{l} \text{if } i \in N_c(t) \\ \text{if } i \notin N_c(t) \end{array},$$

where $\alpha(t)$ is a scalar-valued "adaptation gain" $0 < \alpha(t) < 1$, which decreases in time and is often Gaussian in practice.
SOM is an unsupervised technique that identifies clusters in a data set and performs topological ordering while computing the feature map.

Testing. After training, a distance measure of neurons representing clusters can be thresholded to perform outlier detection. The main limitation of this method according to Singh and Markou [96] is the selection of an appropriate threshold.

Summary 2.1: Self-Organizing Map.

First, the extracted motion data is split into non-overlapping spatio-temporal volumes, each consisting of m frames, referred to as m-cuboid in this work (see Fig. 2.3 for an illustration). To increase the robustness to noise as well as occlusions two dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ are used to fit the motion flow vectors $(\Delta x, \Delta y)$ of each clip, referred to as a *motion pattern MP*. Subsequent motion patterns are then concatenated in temporal order to form *behavioral patterns* $BP^i = \{MP^i_j\}$, $j = 1 \dots m$. The distance between behavioral patterns is measured as an averaged sum of Kullback-Leibler (KL-) divergences[2] between pairs of matching motion patterns.

A $K \times K$ two-dimensional lattice of neurons is trained using behavioral patterns as weights. Behavioral patterns acquired in the training phase are drawn randomly and used to train the network. Adaptation of a behavioral patterns is realized by adaptation of the $\boldsymbol{\mu}$ and Σ of its motion patterns. After training, the distances of all training behavior patterns are calculated and used to determine a distance distribution $\mathcal{N}_d(\boldsymbol{\mu}_d, \sigma_d)$.

Finally, in the detection phase, the distance distribution is used to decide whether a given test behavioral pattern is an outlier, i.e., all instances with a larger distance than $n \cdot \sigma_d$ are reported as outliers. Even though topological ordering it performed while learning the model, it is not relevant to the outlier detection capabilities of the TASOM.

2.2.2 Growing Neural Gas

The second topology preserving feature map employed is GNG, proposed by Fritzke [34]. GNG is a combination of CHL and Neural Gas (NG) introduced by Martinetz and Schulten [69]. Unlike NG, GNG has no parameters that change over time and continues learning until a performance criterion is met. Since non-stationary data

[2]The **Kullback-Leibler Divergence** is a natural way of computing a similarity measure between distributions. Myrvoll and Soong [76] propose the following symmetric, positive semi-definite form

$$d(f,g) = \int f(x) \log \frac{f(x)}{g(x)} dx + \int g(x) \log \frac{g(x)}{f(x)} dx.$$

Using multivariate normal distributions, its closed form is

$$d(f,g) = \frac{1}{2} \operatorname{trace}\{(\Sigma_f^{-1} + \Sigma_g^{-1})(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^T + \Sigma_f \Sigma_g^{-1} + \Sigma_g \Sigma_f^{-1} - 2I\}.$$

distributions are often found in real-world processes, this extension of the NG model enables continuous adaptation.

GNG is deployed in the context of outlier analysis exploiting its ability to learn topologies in a subspace, which reflect the topological structure of the data. The dimensionality of the subspace has to be chosen *a priori*. Background information on GNG is found in summary box Sum. 2.2, two approaches based on GNG are described below.

2.2.2.1 Incremental Neural Network

One method based on GNG is found in Shen and Hasegawa [93], who argue that unlike SOM and NG methods, GNG-U can follow a non-stationary input distribution, but the previously learned prototype patterns are completely destroyed. Therefore the removal of nodes prevents GNG-U from being used in the context of *life-long learning*. In this context, *dead nodes* play a major role in that they preserve knowledge of the previous situations for future decisions. They propose an adaptation of the GNU-U: One part of this adaptation is the reformulation of the removal criterion to remove only nodes in regions with low probability density resulting in cluster separation. Additionally, they claim that a periodical application of this strategy removes nodes caused by noisy input data.

The method is deployed in two layers where the first one generates a topological structure of the input patterns and the second reports the number of clusters and gives a typical prototype for each cluster. The advantages of the proposed method are that no a priori decision about network size is needed and, additionally, the permanent increase in the number of nodes and the constant drift of the centers to capture the input distribution are circumvented.

2.2.2.2 Computing trajectory prototypes using GNG

Widhalm and Brändle [107] use particles moved according to optical flow to generate trajectories. Points on the extracted trajectories are represented as a five-dimensional feature

$$f = [x, y, u_x, u_y, s],$$

Growing Neural Gas

The GNG algorithm operates a network consisting of

- a set A of nodes, where each node $c \in A$ has an associated *reference vector* $w_c \in \mathcal{R}^n$ representing the position of the node in input space.
- a set N of connections between pairs of nodes that are not weighted, but only define the topological structure.

The basic idea of the GNG model is to construct and track this network representation of the underlying distribution by successively adding new units and applying the following three actions:

- **Adapting** the network by moving the nodes towards the given training instances and adapt local errors.
- **Inserting** new nodes halfway between the nearest and the second-nearest node that accumulated the maximum local error. The accumulation of local errors aids in identifying underrepresented areas.
- **Removing** edges and nodes that are deserted because the network moved.

The resulting net is a subgraph of the Delaunay triangulation covering those areas of the input space that is populated by the training data distribution. It is called the "induced Delaunay triangulation", which has been shown to optimally preserve topology in a general sense by Martinetz [68].

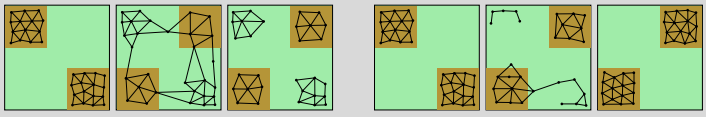


Figure 2.4: (a) GNG, (b) GNG with Utility criterion.

GNG is able to follow non-stationary distributions as long as the changes are slow. Rapid changes cannot be handled properly, so-called *dead units* remain (see Fig. 2.4 (a)). To overcome this problem Fritzke [35] proposes to use a local utility measure – analogous to the local error measures introduced with GNG – to formulate a removal criterion and enable modeling non-stationary data distributions by deleting nodes that are located in regions of low input probability densities (see Fig. 2.4 (b)). This extension is referred to as Growing Neural Gas with Utility Criterion, GNG-U.

Summary 2.2: Growing Neural Gas.

using the image coordinates (x, y), normalized direction components (u_x, u_y) and speed s. Then GNG is used to find prototype trajectories that represent the collected data sufficiently. Subsequently, particle trajectories are expressed using the prototypes obtained and clustered using the self-tuning spectral clustering algorithm proposed by Zelnik-Manor and Perona [114] using a Dynamic Time Warping (DTW) distance measure. The clusters obtained are then represented using their means μ_c and covariances Σ_c and any trajectory which can not be expressed using these clusters is reported as abnormal.

2.3 Bayesian network models

The third category of approaches presented in this overview are Bayesian Networks (BNs). A BN is a probabilistic Graphical Model (GM) that represents knowledge about an uncertain domain in a structure known as a *Directed Acyclic Graph* (DAG). Each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. The conditional independences represented by BNs allow for reduction of the parameters necessary to characterize the Joint Probability Distribution (JPD). This reduction enables an efficient computation of the posterior probabilities when evidence is given.

By marginalization, i.e., summing out over variables, one can evaluate all possible inference queries. But even for the case of binary variables, the JPD has size $O(2^n)$, where n is the number of nodes. Summing over the JPD takes exponential time, full summation over discrete variables is known to be an *NP-hard* problem. Some efficient algorithms exist to exactly solve the inference problem in restricted classes of networks. Alternatively, approximate inference methods are proposed, such as *Monte Carlo* sampling that gives gradually improving estimates as sampling proceeds. A variety of standard Markov chain Monte Carlo (MCMC) methods, including the *Gibbs sampling* and the *Metropolis-Hastings algorithm*, are also used for approximate inference (Ben-Gal [9]).

The Bayesian networks in the following are illustrated in **plate notation** introduced by Buntine [18] and Gilks et al. [40]. Plate notation provides a language for encoding models with repeated structure and shared parameters. Identically dis-

tributed variables that are repeated together are enclosed in a box – or *plate*. Plate models induce (possibly infinite) sets of Bayesian networks (Koller and Friedman [57]).

2.3.1 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (PLSA), introduced by Hofmann [50], is a generative document model that is widely used because of its simplicity. In the following two approaches using PLSA are summarized. Again, see summary box Sum. 2.3 for basic theoretical information on the model.

Probabilistic Latent Semantic Analysis

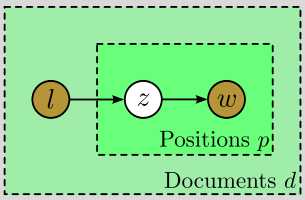


Figure 2.5: PLSA in plate notation.

PLSA is a generative document model that associates each word $w \in \mathcal{W} = \{w_1, \ldots, w_M\}$ in a document $d \in \mathcal{D} = \{d_1, \ldots, d_N\}$ with an unobserved topic $z \in \mathcal{Z} = \{z_1, \ldots, z_K\}$. Terms like "words", "documents", and "topics" are used to guide intuition, in the context of video analysis features are words, clips – m successive video frames – are documents and topics typically relate to specific behaviors in a scene. A schematic illustration of PLSA in plate notation is shown in Fig. 2.5.
The model postulates that a document d and a word w are conditionally independent given a latent topic z. Unlike simpler latent variable models, i.e., the Unigram model or the Mixture of Unigram model, PLSA allows to associate multiple topics with one document. The standard procedure for Maximum Likelihood Estimation (MLE) in latent variable models is the EM algorithm.

Summary 2.3: Probabilistic Latent Semantic Analysis.

2.3.1.1 Anomaly Detection using hierarchical PLSA

Li et al. [62] propose a two-stage hierarchical PLSA model for anomaly detection based on semantically segmented image regions. To extract features, background subtraction is performed and blobs of foreground pixels are represented by a 10-dimensional vector as

$$\boldsymbol{v}_f = [x, y, w, h, r_s, r_p, u, v, r_u, r_v],$$

with (x, y) the centroid position, (w, h) the width and height of the bounding box, r_s the blob ratio, r_p the percentage of foreground pixels, (u, v) the mean optic flow and scaling features $r_u = u/w$ as well as $r_v = v/h$. Extracted from non-overlapping temporal video clips these features are clustered using K-means, where the number of clusters is set to be the average number of image events across all frames of the clip. Clusters are then represented by a 20-component-feature-vector referred to as **atomic events**:

$$\boldsymbol{v} = [mean(\boldsymbol{v}_f), var(\boldsymbol{v}_f)].$$

Scene segmentation is then performed based on the atomic events using a modification of the spectral clustering proposed by Zelnik-Manor and Perona [114]. Finally, the following two-stage hierarchical PLSA model is applied:

1. Each segmented region is treated as a document and one PLSA per region is used to learn the local behavior correlations.
2. The local topics obtained from all regions in the first step are regarded as the visual words and a global PLSA is used to learn global correlations.

Global behavior topics are inferred using the second stage PLSA and anomaly scores for clips are computed.

2.3.1.2 PLSA scene segmentation

Varadarajan and Odobez [104] argue that though PLSA is not a fully generative model, its tractability make it an attractive alternative to fully generative models.

In their work, PLSA is used to model visual words from video using three types of features: location, motion and size. Pixel positions are quantized into non-overlapping cells of 10×10 pixels, e.g., for 360×280 a set of 36×28 cells is obtained. Motion is computed using the Lucas-Kanade algorithm on all foreground pixels remaining after background subtraction. Then motion is thresholded and either categorized as static or as one of the four cardinal directions leading to five motion words. Finally, a size word is computed by applying a simple K-means clustering with $K = 2$. The vocabulary of $36 \times 28 \times 5 \times 2 = 10080$ is reduced by assuming independence between motion and size given an activity and its location. Then the codebook size is $36 \times 28 \times (5 + 2) = 7056$.

The model is learned using EM. Different abnormality measures are evaluated, including the log-likelihood measure at the end of the fitting phase, the normalized log-likelihood measure and measures based on the Kullback-Leibler divergence as well as the Bhattacharyya distance.

2.3.2 Latent Dirichlet Allocation

Blei et al. [14] propose Latent Dirichlet Allocation (LDA), a generative model with greater modeling flexibility than PLSA. LDA overcomes the overfitting difficulties of PLSA – the number of parameters to be estimated grow linearly with the number of training documents – in that it allows for a continuous mixture of topics associated with a document. Additionally, it can handle unseen documents. see summary box Sum. 2.4 for further information. For these reasons, the LDA model has recently received attention in the literature. The following two subchapters give a closer look at two methods employing LDA.

2.3.2.1 Temporal Order Sensitive LDA

Li et al. [63] propose a framework using LDA to detect anomalies across a distributed camera network. First, frame differencing is performed and low level features are extracted on “moving” pixels. Location is encoded as cell index g in the uniformly – in cells of size $l \times l$ – divided image plane $g \in [1, G]$. Motion p is quantized into P cardinal directions, $p \in [1, P]$. These features form the codebook of $G \times P$ words.

Latent Direchlet Allocation

LDA is a three-level hierarchical Bayesian model introduced by Blei et al. [14], in which each item of a collection is modeled as a finite mixture over an underlying set of topic probabilities.

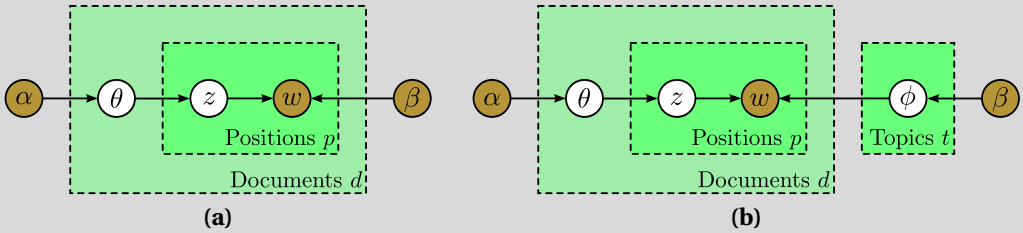


Figure 2.6: (a) LDA and (b) its smoothed variant.

An LDA model representation in plate notation can be seen in Fig. 2.6 (a). The three hierarchies of the model are:

- Corpus-level parameters: sampled once in the corpus generation process
  - α, the parameter of the Dirichlet prior on per-document topic distribution
  - β, the parameter of the Dirichlet prior on per-topic word distribution
- Document-level parameter: sampled once per document
  - θ, the topic distribution of document d
- Word-level parameters: sampled once for each word in a document
  - w, the word and z, its topic

The LDA model allows for a continuous mixture of topics, defined using parameter θ, to be associated with each document d. This approach largely avoids the overfitting problems of other Bayesian models and enables the LDA model to provide better results than other unsupervised methods. Notably, the assignment flexibility allows for words to have low probability relative to a particular topic (Koller and Friedman [57]).

A new document will likely contain words that did not appear in the training corpus. Since the multinomial parameter probability of such words is zero, these documents will be assigned zero probability. Blei et al. [14] suggest to apply variational inference methods to the extended model. Then the model shown in Fig. 2.6 (b) is obtained, where ϕ is a matrix denoting the word distributions for all topics.

Summary 2.4: Latent Direchlet Allocation.

Local behavior topics are computed using LDA on feature words extracted from documents of 25 frames length.

Next, a cell representation is computed where each cell is represented by a feature vector composed of the likelihoods of observing the P possible words with the k^{th} local topic. The similarity between pairs of cells is measured and used as input for spectral clustering (Zelnik-Manor and Perona [114]). This clustering process is carried out on all camera views and returns semantic regions for all views.

Finally, global behavior representation is carried out using standard LDA, but on documents that are composed of a set of successive sliding windows. Specifically, each visual word of a document is indexed by both the region label and the sliding window index t.

Anomaly detection is carried out by predicting the likelihood of local behavior occurrences for all semantic regions and thresholding their probability. In a final step, the contributing local behaviors are identified.

2.3.2.2 Region LDA

Haines and Xiang [42] propose a regional LDA variant – rLDA. The original LDA model is augmented with an identifier i to additionally store the position of a word. Instead of modeling the words as tuples of position and motion, the separation of position and motion is maintained. Position is used to cluster locations and subsequently, each of these clusters is forced to share a single distribution over motion. Both region and topic are then used to index the distribution to draw the word from (see Fig. 2.7 for the rLDA extension of the smoothed LDA).

Each frame is divided into a spatial grid and optical flow is computed and quantized into four cardinal directions. A document is constructed for every clip (5 seconds each) by combining samples from all frames. The standard LDA model is augmented with an identifier, which encodes the quantized position.

Gibbs sampling is applied to estimate the model parameters as in Griffiths and Steyvers [41]. To detect abnormalities multiple samples of region probability are calculated and averaged, normalized and thresholded to find abnormally low probabilities.

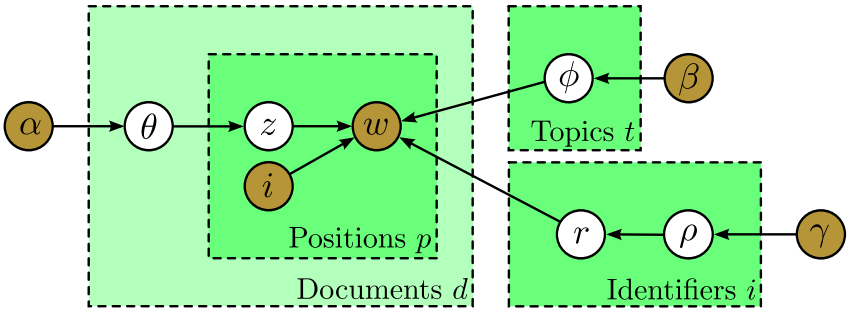


Figure 2.7: Region LDA: The smoothed LDA model is extended with identifiers i to store a word's position. Each sample consists of two parts, the identifier i, which encodes the position, and the word w, which encodes the direction of motion. Each identifier i is associated with a distribution over regions ρ and a region r. After Haines and Xiang [42].

2.3.3 Hidden Markov Model

A Bayesian network model that allows for modeling temporal dependencies is the so called Hidden Markov Model (HMM). In contexts where successive data instances are dependent, the assumption that instances are independent and identically distributed does not hold anymore. In English 'h' is very likely to follow 't' but not 'x'. Such processes consisting of sequences of observations – for example, successive frames of object behavior – cannot be modeled with simple probability distributions. However, a sequence can be characterized as being generated by a random process modeled by a HMM. See summary box Sum. 2.5 for theoretical background information about the HMM. Two approaches employing HMM as a modeling technique are summarized in the following.

2.3.3.1 Temporal Statistics of Spatio-Temporal Motion using HMM

Kratz and Nishino [59] model spatio-temporal relationships with a distribution-based HMM that describes natural motion transitions. A second coupled HMM is used to model spatial relationships between adjoining neighbor cells.

First, the video is divided into local spatio-temporal volumes of fixed size, referred to as *cuboids*. The motion of each cuboid is then represented as a 3D-Gaussian distribution of the spatio-temporal gradient at every pixel of the cuboid. A set of motion distribution prototypes is collected by thresholding the Kullback-Leibler-

Hidden Markov Model

The HMM is a conditional Bayesian Network that represents sequences of variables. The Markov property is assumed: The past has no impact on the future given the present, that is, the model is the same for all t. Therefore an HMM is called a 2-time-slice Bayesian Network, which can be thought of as a sequence of Bayesian network models – or *time slices* – each representing one unit in time. See Fig. 2.8 for an illustration of a *2-TBN* where the time slices are connected through temporal links.

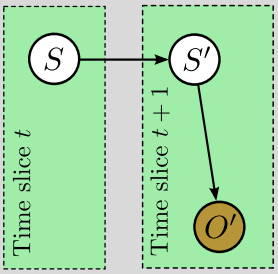


Figure 2.7: HMM as a 2-TBN (after Koller and Friedman [57]).

The HMM is the simplest example of a temporal model having only a single state variable S and a single observation variable O. The network is represented given the initial state distribution and the transition model (Jensen and Nielsen [54], Koller and Friedman [57]).

Summary 2.5: Hidden Markov Model.

divergence. If the KL-divergence exceeds a predefined threshold the new pattern is added to the prototype set, otherwise the prototype set is updated with the new observation.

Afterwards, based on the prototype set, HMMs are trained using EM. One HMM per tube is trained to capture temporal relationships, and a coupled HMM – mapping the spatial relationships between neighboring cells of a tube – is trained to capture spatial relationships. Finally, spatial and temporal confidence measures are combined to identify unusual relationships between motion patterns.

2.3.3.2 Markov chained LDA

Hospedales et al. [51] extend the smoothed LDA model by introducing a third layer that captures co-occurring topics to form what is termed a document category. Additionally, they add a Markov chain to model the change of the behavior category from clip to clip. The model, termed Markov Clustering Topic Model (MCTM), is intractable, but a collapsed Gibbs sampler can be used to approximately learn the model and draw inferences.

The image plane is divided into $C \times C$ cells and motion is quantized into four cardinal directions. Together with a fifth, stationary motion state (calculated using background subtraction) on a 320×240 frame and $C = 10$ a codebook of size $32 \times 24 \times (4+1) = 3840$ is derived.

Learning is performed offline assuming that the training dataset is representative and takes about 4 hours for a 5-minutes-video. The authors claim that without the iterative sweeps of the Gibbs sampler outlier detection can be performed online by calculating the Bayesian surprise of the marginal likelihood of the new observation given all other observations.

2.4 Other approaches

Finally, two methods are described that can not be categorized in any of the three preceding areas. The first uses a Markov Random Field (MRF), which, in contrast to the Bayesian networks described earlier, is a network formed by an **undirected** graph connecting random variables. The second one is the only method not based on motion as a low-level feature, but on texture. It uses Dynamic Textures to model scene normality.

2.4.1 Co-occurrences using Markov Random Fields

Benezeth et al. [10] use a Markov Random Field (MRF) to model co-occurring events in a spatio-temporal volume and detect outliers.

First, using a background subtraction method, a motion label field $X_s \in \{0,1\}$ is obtained where ones denote motion and $s = (\boldsymbol{x}, t)$ is a pixel at location $\boldsymbol{x}$ and time t.

The motion label sequence of length $2\eta + 1$ centered at pixel s is $\boldsymbol{X}_s = [X_{\boldsymbol{x},t-\eta}, \ldots, X_{\boldsymbol{x},t+\eta}]$. Given a video of size $Q_0 \times R_0 \times T_0$ and a spatio-temporal neighborhood of a pixel s, $\mathcal{M}_s$ of size $Q \times R \times T$, where $Q < Q_0$, $R < R_0$ and $T \ll T_0$, the co-occurrence matrix α_{uv} of two spatio-temporal locations $u, v \in \mathcal{M}_s$ is

$$\alpha_{uv} = \frac{\beta_{uv}}{T_0 - T} \sum_{t=T/2}^{T_0-T/2} \text{sim}(\boldsymbol{X}_u, \boldsymbol{X}_v),$$

where $\text{sim}(\boldsymbol{X}_u, \boldsymbol{X}_v)$ is the mutual information between motion label vectors $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$. Benezeth et al. [10] show that the co-occurrence between two sites u and v can be determined using motion label values X_u and X_v instead of sequences $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$. Then u and v co-occur whenever $X_u = X_v = 1$ and α_{uv} can be easily computed.

Given motion label observations O_s of a pixel s, a simple likelihood ratio test is derived to determine if observation O_s is normal according to the co-occurrence matrix α. To be able to deal with multiple moving objects, those sites r are reported which not only co-occur with site s, but are also connected to s, that is, a connected graph of ones between the sites exists in O_s.

2.4.2 Mixtures of Dynamic Textures

Mahadevan et al. [66] use a grid of Mixtures of Dynamic Textures (MDT) to model temporal abnormalities. Spatio-temporal patches are extracted on a finer grid and used to train the nearest MDT. During outlier detection, patches of low probability under the cells MDT are considered anomalies.

Mixtures of Dynamic Textures are based on a characterization of Dynamic textures proposed by Soatto et al. [98], which are generative models for the dynamics and the appearance of video sequences. A linear dynamical system is defined by

$$\begin{cases} x_{t+1} = Ax_t + v_t & x_0 \in \mathbb{R}^n, v_t \sim \mathcal{N}(0, Q) \\ y_t = Cx_t + w_t & w(t) \sim \mathcal{N}(0, R) \end{cases},$$

where $y_t \in \mathbb{R}^m$ is the observed variable encoding the video frame at time t and $x_t \in \mathbb{R}^n$ is a hidden state variable encoding the evolution of the video, typically $n \ll m$. The parameter $A \in \mathbb{R}^{n \times n}$ is a state-transition matrix, and $C \in \mathbb{R}^{m \times n}$ is an observation matrix. Q and R are symmetric positive-definite matrices. According

to Chan and Vasconcelos [24] different methods to estimate the model parameters $\Theta = \{A, C, Q, R\}$ from observation measurements y_1^τ are available.

Chan and Vasconcelos [24] sample the observed video sequence from one of K dynamic textures associated with component priors $\alpha = \alpha_1, \ldots, \alpha_K$, with $\sum_{j=1}^{K} 1$. First a component index z is sampled, then an observation y^τ is sampled from the dynamic texture component $\Theta_z = \{A_z, C_z, Q_z, R_z, \mu_z, S_z\}$.

The probability of a sequence $y_1^\tau = [y_1, \ldots, y_\tau]$ under this model is

$$p(y_1^\tau) = \sum_{i=1}^{K} \alpha_i p(y_1^\tau | z = i),$$

where $p(y_1^\tau | z = i)$ is the class conditional probability of the j^{th} dynamic texture component parametrized by Θ_j.

An algorithm for parameter estimation of the MDT using EM can be found in Chan and Vasconcelos [24].

Chapter 3

Method Selection and Realization

Contents

3.1 Evaluation and selection 29

3.2 Feature extraction and representation 32

3.3 Implementation 34

3.1 Evaluation and selection

The main goal of this work is to compare different approaches from the field of low-level feature based outlier detection methods. As described in the literature review, the following restrictions are applied to narrow the scope of existing work taken into consideration: The method is applied to video stream analysis in the literature, it works unsupervised and is based on low-level features. More than a dozen methods meeting these criteria are described in Chapter 2. To guide the selection as to which of these methods are implemented, a simple, straightforward strategy is applied. A number of desirable – partly qualitative – properties are collected and evaluated on each method. In the following the properties used are described in detail as well as their assignment for the individual approaches explained. The results are then summarized and those method meeting the most criteria are chosen to be implemented.

- **Usage of motion-based features.** Since outlier detection based on motion as a low-level feature is the most common approach in the literature, comparing

those methods provides a wide range of works. Since the goal of this work is to provide a comparison using a unified feature, methods using common motion-based features are ranked higher than methods based on other low-level features.

Adam et al. [1] use an SSD-based probabilistic motion measure, Benezeth et al. [10] a binary motion label using simple background subtraction. Feng et al. [33] use smoothed optical flow to estimate Gaussian flow distributions, Hospedales et al. [51] optical flow quantized into one of four cardinal directions. Li et al. [62] perform background subtraction and use the mean optic flow vector of blobs of foreground pixels besides other low level features. Similarly, in Li et al. [63], optical flow is computed on "moving" pixels detected using frame differencing, which is then quantized in both motion direction and location using a codebook. Shi et al. [95] apply phase correlation for motion estimation, Varadarajan and Odobez [104] use the Lucas-Kanade algorithm to compute optical flow of foreground pixels detected using background subtraction. Finally, Widhalm and Brändle [107] employ particle trajectories based on optical flow.

The diversity of motion-based features used is impressive and only two methods in our review use different feature approaches: Kratz and Nishino [59], who use distributions of spatio-temporal gradients and Mahadevan et al. [66], where dynamic textures are used.

- **Localization of outlier detection.** Methods that inherently report regions responsible for triggering outliers are preferred to methods that report the frame as a whole or require post-processing to localize outliers.

  Adam et al. [1] and Tziakos et al. [102] report outliers localized at predefined spatial positions. Benezeth et al. [10] localize outliers in a sub-window of the video, but require post-processing to handle multiple objects.

  By far most common approach is to divide the image plane in non-overlapping cells and report those which witness outlying behavior. This approach is employed in Feng et al. [33], Haines and Xiang [42], Hospedales et al. [51], Kratz and Nishino [59], Mahadevan et al. [66], Shi et al. [95] and Varadarajan and Odobez [104].

  Finally, Widhalm and Brändle [107] report abnormal trajectories, hence, localized outliers.

In Li et al. [62] and Li et al. [63] localization is provided only indirectly through calculating probabilities of local behavior contributing to a global anomaly, therefore these approaches are not considered localized.

- **Efficiency.** Efficient methods are favored. A method's efficiency is used in the same sense as used in Adam et al. [1]: "...the algorithm is computationally non-demanding, allowing it to perform real-time analysis of the video stream ...".

  Adam et al. [1], Benezeth et al. [10] as well as Shi et al. [95] report that their algorithms run in real-time.

  According to Hospedales et al. [51] the computational cost of MCMC model learning is hard to quantify, because convergence assessment is an open question. Based on the runtime reported in their work – training on 5 minutes of data required 4 hours – the method was categorized as inefficient. Mahadevan et al. [66] report that training the mixtures of dynamic textures takes around 2 hours, while testing per frame is about 25 seconds on a 3GHz CPU and hence is also categorized as inefficient in the sense used in this work.

  Since the rest of the approaches considered does not report performance evaluations, their efficiency assessment is deferred.

- **On-line.** Methods that are able to adapt in a life-long learning sense are preferred to methods whose model is trained off-line and is not adapted later.

  Adam et al. [1] achieve constant adaptation in that cyclic buffers are employed. Although normal behavior already seen is lost using this approach it is still considered an adaptive approach. Feng et al. [33] update the neurons of the SOM whenever new data is available and Shi et al. [95] keep updating the set of GMMs maintained.

  The rest of the methods learns the model of normality off-line, i.e. on a limited footage.

- **Spatially unrestricted.** Methods that are able to report outliers on the whole frame are chosen over those which report outliers only on some fixed spatial locations. As already described under localization aspects, only three methods report outliers on a subset of the image plane: Adam et al. [1], Benezeth et al. [10] and Tziakos et al. [102].

The results of the evaluation are summarized in Tab. 3.1, all methods lacking an efficiency assessment are declared inefficient. Based on this table the top ranked methods fulfilling more than three criteria were chosen to be implemented, these methods are those proposed by Adam et al. [1], Feng et al. [33] and Shi et al. [95]. Interestingly, all other methods are not adaptive, arguably a key requirement of outlier detection.

	motion-based	**localized**	**efficient**	**on-line (adaptive)**	**spatially unrestricted**	**# of criteria met**
Shi et al. [95]	✓	✓	✓	✓	✓	5
Adam et al. [1]	✓	✓	✓	✓		4
Feng et al. [33]	✓	✓	?	✓	✓	4
Haines and Xiang [42]	✓	✓	?		✓	3
Hospedales et al. [51]	✓	✓			✓	3
Varadarajan and Odobez [104]	✓	✓	?		✓	3
Widhalm and Brändle [107]	✓	✓	?		✓	3
Benezeth et al. [10]	✓	✓	✓			3
Kratz and Nishino [59]		✓	?		✓	2
Li et al. [62]	✓		?		✓	2
Li et al. [63]	✓		?		✓	2
Mahadevan et al. [66]		✓			✓	2
Tziakos et al. [102]	✓	✓	?			2

Table 3.1: Method property summary.

3.2 Feature extraction and representation

During method analysis the usage of a single motion feature would simplify method comparison and reveal a method's individual strengths and weaknesses. Since all three methods identified in the evaluation employ different types of motion features, their feature extraction processes are reviewed in the following to be able to identify one unifying motion feature. Terms used in the review are illustrated in Fig. 3.1.

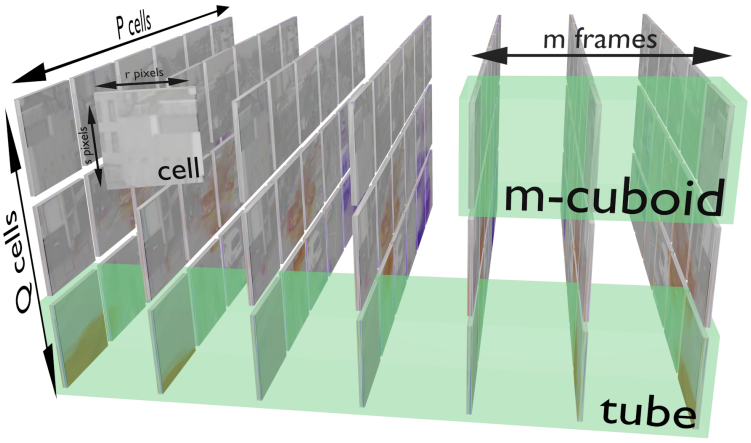


Figure 3.1: Image plane division in a grid of $P \times Q$ non-overlapping cells of $r \times s$ pixels size. The temporal extension of one cell is termed a *tube*, aggregated spatio-temporal volumes of m cells are referred to as an m-*cuboid*.

- Adam et al. [1] define fixed spatial locations to extract flow measurements and apply a method similar to the one used by Rosenberg and Werman [87] to avoid the aperture problem. A cell around a fixed location – referred to as a "monitor" – is used to estimate motion. In this cell the SSD(u, v) error matrix of corresponding discrete shifts is calculated and transformed into a probability distribution.

  Flow is then calculated by aggregating the probability in either radial or angular bins and finally verified using an ambiguity test.

- Feng et al. [33] calculate dense optical flow using the technique by Zach et al. [113]. Subsequently, the obtained flow is divided into m-cuboids and smoothed by median filtering. Modeled using two-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, these *motion patterns* are then used as basic building blocks for *behavioral patterns* – concatenations of motion patterns in temporal order.

- Shi et al. [95] divide the image plane into non-overlapping cells of 16×12 pixels. Then, phase correlation (De Castro and Morandi [28]) is used to calculate the shift between two temporally successive cells.

The feature extraction procedure by Feng et al. [33] can be seen as a common denominator, provided that the fixed spatial positions used by Adam's method are placed at the center of individual cells of the grid. The final unified motion extraction procedure is summarized as Alg. 3.1 and illustrated in Fig. 3.2.

Unified motion feature extraction

1. Crop and resize the video frames to a unified size, Fig. 3.2 (a) - (c).
2. Extract dense optical flow using the freely available optical flow technique provided by Chambolle and Pock [23], Fig. 3.2 (d).
3. Split the optical flow in $P \times Q$ fixed-sized regions of $r \times s$ pixels size, Fig. 3.2 (e).
4. For each region, Fig. 3.2 (f):
   (a) Merge all $m \times r \times s$ flow vectors of an m-cuboid and apply median filtering.
   (b) Estimate the two-dimensional Gaussian flow distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ using maximum likelihood.

Algorithm 3.1: Unified motion feature extraction.

While for Feng et al. [33] and Shi et al. [95] the distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is used, for Adam et al. [1] only the estimated mean $\boldsymbol{\mu}$ is utilized.

3.3 Implementation

Based on the method evaluation summarized in Tab. 3.1 three method are chosen to be implemented: Adam's `HIST` method is based on histograms calculated on cyclic buffers, Shi's `STCOG` maintains a GMM for each pair of adjacent cells to model the scene and Feng's `TASOM` uses a SOM to express scene normality.

Implementation details are found in the following. Additionally, with each method modifications and extensions originating from detailed performance analysis are presented.

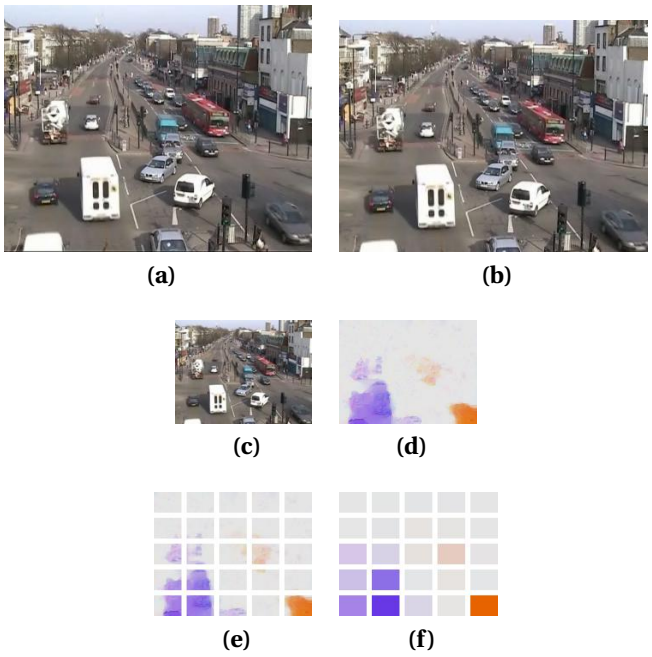


Figure 3.2: The unified motion extraction procedure: The unified motion extraction procedure: (a) video frame, (b) cropped and (c) rescaled to the unified size of 160×120 pixels. (d) extracted optical flow - color-coded, (e) flow divided into $P \times Q$ cells and (f) $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ estimated in each cell - $\boldsymbol{\mu}$ pictured.

3.3.1 Anomaly Detection Using Multiple Fixed-Location Monitors

Adam et al. [1] propose to use local "monitors" at fixed spatial positions to detect outliers. At those positions, which in this work are replaced by $P \times Q$ cells on the image plane, motion information is monitored to detect abnormal instance occurrences.

Depending on the scene to be processed, the system designer or the user decides whether motion direction or speed is monitored. If direction is chosen, then angular binning is applied, otherwise the magnitude is binned radially. The number of bins used is denoted as N_{bins}, a binned observation as $\boldsymbol{\mu}^b$. The binned observations are then committed to a set of cyclic, fixed length buffers, i.e., one buffer of length L_{buffer} for each cell. Provided the scene contains phases, i.e., a light-controlled traffic junction, L_{buffer} is set to cover at least one phase.

Once the buffers are filled, outlier detection can be performed by thresholding an observation's probability $p_{outlier}$ using a predefined threshold th_{alert}:

$$N_{occurrences} = \left|\{\boldsymbol{\mu}^b | \boldsymbol{\mu}_t^b = \boldsymbol{\mu}_{t-i}^b\}\right|, i = \{1 \dots L_{buffer}\},$$
$$p_{outlier} = 1 - N_{occurrences}/L_{buffer}.$$

When a new observation is available it replaces the oldest one in the cyclic buffer. This enables the method's adaptation to scene changes as the video stream is processed.

Name	Typical value	Remarks
type	`speed` or `direction`	
$speed_{max}$		Only necessary if $type =$ `speed`
N_{bins}	8	
L_{buffer}		Depends on the phases present in the scene, usually at least one complete phase is covered
th_{alert}	1/100	

Table 3.2: Parameters of Adam's method.

3.3.1.1 Integration procedure

Subsequently, alerts produced by individual cells are integrated using a simple procedure to ensure that volatile outliers are filtered before reporting an event:

1. **Detecting "alarming" frames.** If the number of alerting cells in the current frame is at least Z, then the frame is considered an "alarming frame". The choice of Z is related to the number of cells used as well as the expected number of cells that will observe an "unusual" observation at the same time. Adam et al. [1] suggest to use at least five monitors covering an object and to use $Z = 1$ to handle occlusions and nonobservations.

2. **Filtering reliable events.** If at least K of Y preceding frames were "alarming frames" the algorithm produces an alert to the user (typical values are $K = 7$ and $Y = 10$). Y is related to the expected duration of the unusual activity, but also to the delay in reporting an alarm to the user. Adam et al. [1] suggest that K should be defined lower than Y to make the procedure more robust to occlusions or missing observations.

In this work an extension of the procedure is used, in which an additional parameter N is used to control the minimal number of cells that finally trigger an alarm, i.e., to avoid "small" events to be reported.

The procedure is then extended by a third step:

3. **Filtering "small" events.** If the number of alarming cells in the frame is greater than N, an alarm is triggered. Using $N = 0$ results in the original integration procedure.

Name	Typical value	Remarks
Z	1	minimal # of alerts in a frame to be "alerting"
Y	10	# of frames taken into account when calculating alarms
K	7	minimal # of "alerting" frames of Y preceding frames to trigger an alarm
N	1	minimal # of "alerting" cells in an "alarming" frame (filtering small events)

Table 3.3: Parameters of Adam's integration procedure.

3.3.2 Spatial-temporal co-occurrence GMMs

Shi et al. [95] use spatial-temporal co-occurrence GMMs (STCOGs) to detect outliers using *pair-wise cuboids*. A pair-wise cuboid at time t and adjacent cells p and q is denoted $C^t_{p,q}$. Adjacency is defined as being a four-neighborhood, therefore every cell is part of $N_c = 4$ pair-wise cuboids. See Fig. 3.3 for an illustration of this concept.

The mean motion vectors $\boldsymbol{\mu} = (\Delta x, \Delta y)$ at two adjacent locations p and q and L successive frames are concatenated to form the pair-wise cuboid

$$C^t_{p,q} = \{\boldsymbol{\mu}^t_p, \ldots, \boldsymbol{\mu}^{t-L+1}_p, \boldsymbol{\mu}^t_q, \ldots, \boldsymbol{\mu}^{t-L+1}_q\}^T, \tag{3.1}$$

a $4 \cdot L$ dimensional vector.

While processing the video stream pair-wise cuboids are extracted and a set of GMMs is adapted to reflect changes in the scene. For every pair of cells on the $P \times Q$ grid one GMM is maintained, therefore the total number of GMMs is

$$N_{GMM} = Q \cdot (P-1) + P \cdot (Q-1) = 2 \cdot P \cdot Q - Q - P.$$

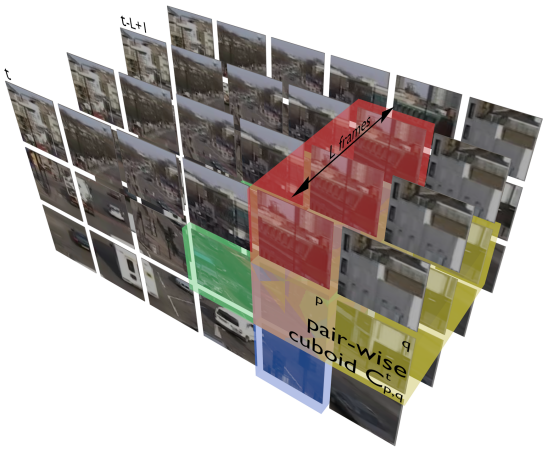


Figure 3.3: Four pair-wise cuboids of L frames depth, each covering the center cell p and one adjacent cell. Pair-wise cuboid $C^t_{p,q}$ at time t and positions p and q in yellow.

Each of the GMMs consists of maximally K_{max} components, the i^{th} component is denoted as $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ and associated with weight ω_i.

For every incoming cuboid $C^t_i, i = 1 \ldots N_{GMM}$, an online K-means approximation (inspired by Stauffer and Grimson [99]) is applied to the associated GMM. See Alg. 3.2 for a summary of the learning procedure.

3.3.2.1 Outlier detection

For each cell, the probability p of an observation is calculated and thresholded to detect outlying instances. It is computed by averaging the probabilities p_{C_j} of all N_C pair-wise cuboids:

$$p = \frac{1}{N_C} \sum_{j=1}^{N_C} p_{C_j}.$$

Probabilities P_{C_j} are computed as a weighted sum of the components of the associated GMM:

$$p_{C_j} = \sum_{i=1}^{k_j} \omega_i \cdot \exp\Big(-\frac{1}{2}(C_j - \boldsymbol{\mu}_i)^T \cdot \Sigma_i^{-1} \cdot (C^t_j - \boldsymbol{\mu}_i)\Big), j = 1, \ldots, N_C.$$

GMM update procedure

A GMM with k components, $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$, $i = 1, \ldots, k$, each of them associated with a weight ω_i, as well as the incoming pair-wise cuboid C^t and the threshold p_{max} for creating new components are given.

1. The probabilities of all components are computed:

$$p_i = \exp\Big(-\frac{1}{2}(C^t - \boldsymbol{\mu}_i)^T \cdot \Sigma_i^{-1} \cdot (C^t - \boldsymbol{\mu}_i)\Big), i = 1, \ldots, k. \tag{3.2}$$

2. If $\exists p_i > p_{max}$
   (a) Choose the component with the highest probability:

$$k = \arg\max_i \ p_i.$$

   (b) Update the k^{th} component $\mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$:

$$\boldsymbol{\mu}_k^t = (1 - \beta) \cdot \boldsymbol{\mu}_k^{t-1} + \beta \cdot C^t, \tag{3.3}$$

$$\Sigma_k^t = (1 - \beta) \cdot \Sigma_k^{t-1} + \beta \cdot (\boldsymbol{\mu}_k^{t-1} - C^t)^T (\boldsymbol{\mu}_k^{t-1} - C^t), \tag{3.4}$$

$$\omega_k = \omega_k + \Delta\omega. \tag{3.5}$$

3. If $p_i \leq p_{max} \forall i = 1, \ldots, k$
   (a) If $k < K_{max}$ add a new component:

$$\boldsymbol{\mu}_k \ = C^t, \Sigma_k = I_{4 \cdot L} \text{ and } \omega_k = \Delta\omega. \tag{3.6}$$

   (b) If $k = K_{max}$ replace the component with the lowest probability:

$$k = \arg\min_i \ p_i$$

   with a new component constructed using Eq. 3.6.

4. Normalize the weights $\omega_i = \omega_i / \sum_{i=1}^{k} \omega_i$.

Algorithm 3.2: GMM update procedure.

Name	Typical value	Remarks
N_c	4	Neighborhood relation used
L	$3 \dots 8$	Number of frames used to form a pair-wise cuboid
K_{max}	100	Maximal number of components of one GMM
β	$1/10$	The learning rate
$\Delta\omega$	$1/10$	The weight reward
p_{max}	$1/10, 1/100$	Probability threshold for creating a new component
th_{alert}	$1 - 10^{-2}$	Probability threshold for outlier detection

Table 3.4: Parameters of Shi's method.

3.3.3 Modified STCOG

The following observations lead to a modification of the STCOG (M-STCOG):

- Issues arising because of constant reward $\Delta\omega$ in Eq. 3.5: The later a component is
  1. updated, the greater is its relative weight gain.
  2. initialized, the greater is its weight in relation to already existing components.
- Since the unified motion extraction procedure provides a Gaussian motion distribution $\mathcal{N}(\boldsymbol{\mu}_k^t, \Sigma_k^t)$ that was not available in the work of Shi et al. [95], motion variance can be incorporated in the learning procedure (Eq. 3.4). Variance is used two-fold: First, to initialize new components, second, to avoid Σ shrinkage when updating component k with $C_k^t \approx \boldsymbol{\mu}_k^t$.

The following modifications were implemented:

- **Aging concept.** Each component is associated with an age_i, which is incremented every time the component is updated. When adding a new component, its weight is initialized relative to the total age age_{sum} of all k components,

$$age_{sum} = \sum_{i=1}^{k} age_i.$$

  Additionally, when updated, a component's weight reward $\Delta\omega$ reflects the total age of the GMM – it is dynamically adapted using $\Delta\omega = 1/age_{sum}$.

- **Incorporation of variance.** Following the concatenation principle in Eq. 3.1, the Σs of the distributions are placed at the diagonal of a $4 \cdot L \times 4 \cdot L$ matrix to form

$$S_{p,q}^{t} = \begin{pmatrix} \Sigma_p^t & 0 & & \dots & & & 0 \\ 0 & \Sigma_p^{t-1} & 0 & & \dots & & 0 \\ 0 & 0 & \ddots & 0 & & \dots & 0 \\ 0 & \dots & 0 & \Sigma_p^{t-L-1} & 0 & \dots & 0 \\ 0 & \dots & & 0 & \Sigma_q^t & 0 & \dots & 0 \\ 0 & & \dots & & 0 & \Sigma_q^{t-1} & 0 & 0 \\ 0 & & \dots & & & 0 & \ddots & 0 \\ 0 & & & \dots & & & 0 & \Sigma_p^{t-L-1} \end{pmatrix}. \tag{3.7}$$

The Σ^k adaptation step is then carried out identically to $\boldsymbol{\mu}$ adaptation in Eq. 3.3. See Alg. 3.3 for a summary of the modified steps of the STCOG update scheme.

Modified steps of the GMM update procedure

2. (b) Update the k^{th} component:

$$age_k^t = age_k^{t-1} + 1, \omega_k^t = \omega_k^{t-1} + 1/age_{sum},$$
$$\boldsymbol{\mu}_k^t = (1-\beta) \cdot \boldsymbol{\mu}_k^{t-1} + \beta \cdot C^t,$$
$$\Sigma_k^t = (1-\beta) \cdot \Sigma_k^{t-1} + \beta \cdot S^t.$$

3. (a) Initialization of a newly added component

$$age_k = 1, \omega_k = 1/age_{sum}, \boldsymbol{\mu}_k = C^t \text{ and } \Sigma_k = S^t.$$

Algorithm 3.3: Modified steps of the GMM update procedure.

3.3.4 Self-Organizing Maps for Anomaly Detection

Feng et al. [33] introduce outlier detection using the Time Adaptive Self-Organizing Map (TASOM) proposed by Shah-Hosseini and Safabakhsh [92]. The TASOM, in contrast to the original SOM, adapts each neuron's neighborhood size σ_i and learning rate η_i individually. Its lattice consists of $K \times K$ neurons each holding a behavioral pattern

$$BP_i = \{MP_j\}, j = 1 \dots m,$$

which is a concatenation of m motion patterns MP_j in temporal order. Each of the MP_j is a Gaussian distribution estimated from a d-cuboid using Alg. 3.1:

$$MP_j = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j).$$

Employing the closed form of KL-divergence proposed by Myrvoll and Soong [76] the distance between two motion patterns MP_i and MP_j is computed as

$$D_{KL}(MP_i, MP_j) = \frac{1}{2}\,\text{trace}\big\{(\Sigma_i^{-1} + \Sigma_j^{-1})(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T + \Sigma_i\Sigma_j^{-1} + \Sigma_j\Sigma_i^{-1} - 2I\big\}$$

and in turn used to calculate the distance between two behavioral patterns BP^a and BP^b as

$$\widehat{D}_{BP_{KL}}(BP^a, BP^b) = \sqrt{\sum_{k=1}^{m} D_{KL}(MP_k^a, MP_k^b)^2}.$$

As in Shi et al. [95] the log of the distance $\widehat{D}_{BP_{KL}}$ is taken to constrain the values within a small range. The final distance measure is

$$D_{BP_{KL}}(BP^a, BP^b) = \begin{cases} \widehat{D}_{BP_{KL}} & \text{if } \widehat{D}_{BP_{KL}} \leq 1 \\ 1 + log(\widehat{D}_{BP_{KL}}) & \text{if } \widehat{D}_{BP_{KL}} > 1 \end{cases}.$$

3.3.4.1 Learning the TASOM

Before the learning algorithm is applied, a pool of motion patterns is collected, i.e., all possible overlapping MP_i are calculated using d-cuboids. Each of the $K \times K$ neurons of the TASOM lattice is initialized with a behavioral pattern *BP* drawn randomly from the pool. As long as none of the motion patterns of *BP* exceeds the dataset's motion magnitude threshold, the draw is repeated. In addition, the neuron's learning rate η is set close to a predefined maximal learning rate η_{max}, i.e., $\eta = 9/10 \cdot \eta_{max}$, and its neighborhood size σ is set to a predefined neighborhood size $\sigma_{max} \leq K$.

Behavioral pattern update scheme
Given learning rate α and two behavioral patterns

$$\begin{aligned} BP_a &= \{MP_{a_i}\} = \{\mathcal{N}(\boldsymbol{\mu}_{a_i}, \Sigma_{a_i})\} \text{ and} \\ BP_b &= \{MP_{b_i}\} = \{\mathcal{N}(\boldsymbol{\mu}_{b_i}, \Sigma_{b_i})\} \end{aligned}$$

consisting of $i = 1, \ldots, m$ motion patterns. The update

$$BP = (1-\alpha)BP_a + \alpha BP_b$$

is carried out on each of the m individual motion patterns

$$\{\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)\} = (1-\alpha)\{\mathcal{N}(\boldsymbol{\mu}_{a_i}, \Sigma_{a_i})\} + \alpha\{\mathcal{N}(\boldsymbol{\mu}_{b_i}, \Sigma_{b_i})\}$$

using

$$\boldsymbol{\mu}_i = \frac{(1-\alpha)n_a\boldsymbol{\mu}_{a_i} + \alpha n_b \boldsymbol{\mu}_{b_i}}{(1-\alpha)n_a + \alpha n_b} \text{ and}$$

$$\Sigma_i = \frac{(1-\alpha)\big((n_a-1)\Sigma_{a_i}\big) + \alpha n_b \Sigma_{b_i}}{(1-\alpha)n_a + \alpha n_n - 1}$$

with n_a and n_b the individual sample numbers and $(1-\alpha)n_a + \alpha n_b$ the total sample number.

Algorithm 3.4: Update scheme for behavioral patterns.

After initialization an iterative learning scheme is executed to adapt the model to the collected pool of training data. In each step the neuron i_c closest to a ran-

domly drawn behavioral pattern $\boldsymbol{x}$ is determined. Again, only behavioral patterns containing significant motion are considered. Then, the learning rates and behavioral patterns of all neurons in a neighborhood of i_c are updated. See Alg. 3.4 for a detailed summary of the update procedure applied when adapting a neuron's behavioral pattern.

At last, the learning rate of i_c is adapted. Overall, a predefined number N_{it} of learning iterations is carried out to allow for model convergence. A summary of the entire learning process can be found in Alg. 3.5.

3.3.4.2 Outlier detection

After the learning procedure is completed, the minimal distances between the neurons of the SOM and all behavioral patterns BP_i in the pool are computed:

$$d_i = \arg\min_j \; D_{BP_{KL}}(BP_i, BP_j), j = 1, \ldots, K^2. \tag{3.11}$$

Again, only those patterns that contain at least one motion pattern which exceeds the dataset's motion magnitude threshold are considered. Afterwards, a Gaussian distribution of the collected distance values d_i is estimated

$$D \sim \mathcal{N}(\boldsymbol{\mu}_d, \sigma_d). \tag{3.12}$$

Using these distribution parameters outlier detection is performed. Given a behavioral pattern BP to examine, the minimal distance d_{BP} between the given pattern and the model is determined using Eq. 3.11. Finally, using a predefined constant parameter th_m, the threshold multiplier, and distribution parameters $\boldsymbol{\mu}_d$ and σ_d classification is performed:

$$c_{outlier} = (d_{BP} - \boldsymbol{\mu}_d) > th_m \, \sigma_d.$$

Iterative TASOM learning scheme

1. Draw a behavioral pattern $\boldsymbol{x}$ from the pool in a random fashion.
2. Identify the winning neuron $i_{\boldsymbol{x}}$:

$$i_{\boldsymbol{x}} = \arg\min_j \; D_{BP_{KL}}(\boldsymbol{x}, BP_j), j = 1, \ldots, K^2. \tag{3.8}$$

3. For each neuron i_j in the $\sigma_{i_{\boldsymbol{x}}}$-neighborhood $\Delta_{i_{\boldsymbol{x}}}$ and $i_{\boldsymbol{x}}$ itself:
   (a) Adapt the learning rate:

$$\eta_j = (1-\alpha)\eta_j + \alpha\, \eta_{max}\, f\left(\frac{D_{BP_{KL}}(\boldsymbol{x}, BP_j)}{d_{max}}\right), \tag{3.9}$$

   where α is a constant parameter, η_{max} the upper bound of update speed, $f(x)$ a monotonically increasing function, i.e., $f(x) = x/(1+x)$ and d_{max} the maximum distance between two patterns.

   (b) Adapt the behavioral pattern using Alg. 3.4:

$$BP_j = \text{update}(\alpha_{lr}, BP_j, \boldsymbol{x}),$$

   with $\alpha_{lr} = \eta_j\, h_j(i_{\boldsymbol{x}}, i_j)$. As in the basic SOM, h_j is a Gaussian weighting function involving the center neuron i_c and the current neuron i_j:

$$h_j(i_c, i_j) = \exp\left(-\frac{||i_c - i_j||^2}{2\sigma_{i_c}}\right).$$

4. Adapt the neighborhood size of the winning neuron $i(\boldsymbol{x})$:

$$\sigma_{i_{\boldsymbol{x}}} = (1-\beta)\sigma_{i_{\boldsymbol{x}}} + \beta\, \sigma_{max}\, f\left(\frac{avgD(i_{\boldsymbol{x}})}{d_{max}}\right),$$

   where β is a constant parameter, σ_{max} the maximum neighborhood size and $avgD(i)$ the averaged distance in a neighborhood Δ_i calculated as

$$avgD(i) = \frac{1}{|\Delta_i|} \sum_{j \in \Delta_i} D_{BP_{KL}}(BP_i, BP_j). \tag{3.10}$$

Algorithm 3.5: Iterative TASOM learning scheme.

Name	Typical value	Remarks
K	$4, 6, 8$	Size of the lattice
m	$1, \ldots, 3$	Number of motion patterns that form one behavioral pattern
d	$1, 3, 5$	Number of frames used to estimate a motion pattern
α	$1/10$	Constant parameter controlling the change rate of the learning rate
η_{max}	1	The upper bound of learning rate update speed
β	$1/10$	Constant parameter controlling the change rate of the neighborhood size
σ_{max}	$2, 4, 6, 8$	The maximal neighborhood size. Also used to initialize a neuron's neighborhood size
th_m	$2, \ldots, 12$	Threshold multiplier used during outlier detection

Table 3.5: Parameters of Feng's method.

3.3.5 Modified TASOM

In the original TASOM model the update rule for the learning rate η_j is

$$\eta_j = (1-\alpha)\eta_j + \alpha\,\eta_{max}\,f\left(\frac{D_{BP_{KL}}(\boldsymbol{x}, BP_j)}{d_{max}}\right),$$

leading to monotonically decreasing learning rates. To increase the adaptability of the model, in the modified model only the last term is used:

$$\eta_j = \eta_{max}\,f\left(\frac{D_{BP_{KL}}(\boldsymbol{x}, BP_j)}{d_{max}}\right).$$

Additionally, the usage of different distance measures is proposed. First, the KL-divergence used in the original model is computationally expensive, since it is complex and matrix inversions are necessary. Second, different measures might be suitable for different scenarios.

The following measures, which only use the mean $\boldsymbol{\mu}$ of the motion pattern $MP = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ are used in the comparison:

$$\begin{aligned} D_A(MP_i, MP_j) &= arccos\left(\frac{\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j}{|\boldsymbol{\mu}_i|\,|\boldsymbol{\mu}_j|}\right)/\pi, \\ D_R(MP_i, MP_j) &= \big||\boldsymbol{\mu}_i| - |\boldsymbol{\mu}_j|\big|/d_{max} \text{ and} \\ D_{AR}(MP_i, MP_j) &= \frac{1}{2}\Big(D_A(MP_i, MP_j) + D_R(MP_i, MP_j)\Big) \end{aligned}$$

Finally, the two following aggregated measures used in the analysis are:

$$D_{BP_A}(BP^a, BP^b) = \sqrt{\frac{1}{m}\sum_{k=1}^{m} D_R\left(MP_k^a, MP_k^b\right)^2} \quad \text{and}$$

$$D_{BP_{AR}}(BP^a, BP^b) = \sqrt{\frac{1}{m}\sum_{k=1}^{m} D_{AR}\left(MP_k^a, MP_k^b\right)^2}.$$

These measures replace $D_{BP_{KL}}$ in Eqs. 3.8, 3.9 and 3.10.

Furthermore, the modeling of multiple models is proposed for different parts of the scene. The original TASOM model misses outliers that constitute a normal event in another part of the scene. In this work multiple models based on a subset of cells are maintained to solve this problem. See Fig. 3.4 for an illustration of the concept, each rectangular area depicted in blue and green forms the basis for one model. To avoid false positives at the borders between models, each model is trained with data from its inner cells as well as its surrounding neighbor cells.

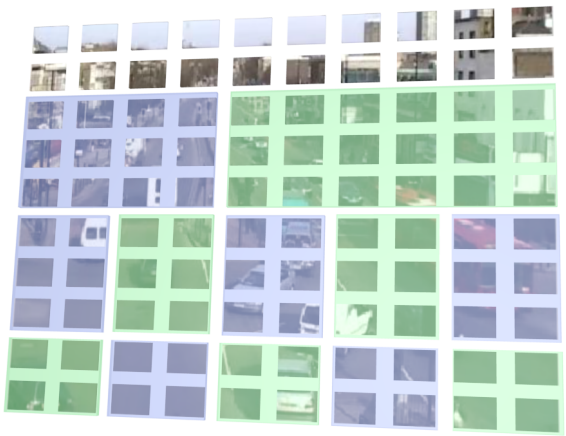


Figure 3.4: To enable modeling localized normal behavior, the usage of multiple `TASOM` models is proposed. Each of the models represents one of the rectangular areas depicted in green and blue.

3.3.6 Baseline method

In addition to the three previously described methods a fourth method – the `BASELINE` method – is implemented. Its results are used to guide the performance assessment of the other methods.

The `BASELINE` method is based on the online K-means algorithm described in Alphaydin [4, p. 276ff, ch. 12] extended to enable cluster creation. Each cluster is associated with a count n_i, representing the number of updates performed. Threshold th_{CC} is used to decide whether a new cluster is created or the nearest existing one is updated. The update scheme is summarized in Alg. 3.6.

On-line K-means update procedure

Given are k cluster centers m_i with associated counts n_i, $i = 1 \ldots, k$. Additionally, the incoming feature vector x and the threshold for creating a new cluster th_{cc} are given.

1. The closest cluster is determined using Euclidean distance:

$$k = \arg\min_i \; ||x - m_i|| \,.$$

2. If $||x - m_k|| < th_{cc}$ the k^{th} cluster is updated:

$$m_k = m_k + \frac{1}{n_k} \cdot (x - m_k).$$

3. If $||x - m_k|| \geq th_{cc}$ a new cluster is initialized:

$$k = k + 1, n_k = 1, m_k = x.$$

Algorithm 3.6: On-line K-means update procedure.

Name	Typical value	Remarks
th_{cc}	$1/10, 5/10$	Threshold for cluster creation

Table 3.6: Parameters of the Baseline method.

Chapter 4

Experiments and Results

Contents

4.1 Obtaining ground truth 49
4.2 Datasets 51
4.3 Experiments 60
4.4 Results 66
4.5 Summary 91

In this chapter experimental results for the three method introduced in Chapter 3 are presented and a method comparison based on confusion matrix analysis is given. To that end, five video datasets used in the literature are manually labeled to obtain ground truth. Then, performance measures based on confusion matrix statistics are evaluated. Based on the most suitable performance measure identified in the evaluation, the individual strengths and weaknesses of the methods and their modified variants are discussed.

4.1 Obtaining ground truth

According to Dee and Velastin [29], establishing ground truth of video footage is problematic. Several difficulties have to be taken into consideration:

- It is hard to obtain genuine footage of interesting events, since they are unlikely to occur in scenarios vision researchers choose to capture, i.e. campus scenes.

When cooperating with CCTV operations to acquire real-world-footage, serious data protection and privacy implications arise.

- When generating ground truth, that is, manually marking those events in a video considered anomalous, only those events that do not fit one's own model are marked leading to subjective ground truth.
- Similar problems emerge when "acted scenes" are used in the evaluation. Questions as to why people acted the way they did arise: What is their relationship to the system designer? Were they told how to act by the system designer?

Due to the fact that no uniform ground truth for the datasets used in this work is available, a certain degree of subjectivity is unavoidable when manually marking the dataset. Nevertheless, the process is carried out carefully bearing the associated problems in mind.

To circumvent the aforementioned problems related to self-captured footage, in this work only video material already used in other work is evaluated. Then, the following procedure is used in the making of the ground truth of a specific dataset:

1. Only events that do not comply with "normal events" in the scene – that is, scene-specific recurring events – are considered anomalous. Normal events are characterized as behavior that is expected on a regular basis. Admittedly, this procedure does not eliminate subjectivity in the perception of exceptional behavior.
2. All $Q \times P$ cells of the image grid that contain a part of the object responsible for an anomaly are marked manually, cf. scenes depicted in the left column of Fig. 4.1.
3. All cells marked anomalous that do not exceed a predefined dataset-specific motion magnitude threshold – and therefore are undetectable – are excluded from the ground truth. The resulting ground truth is shown in the right column of Fig. 4.1.

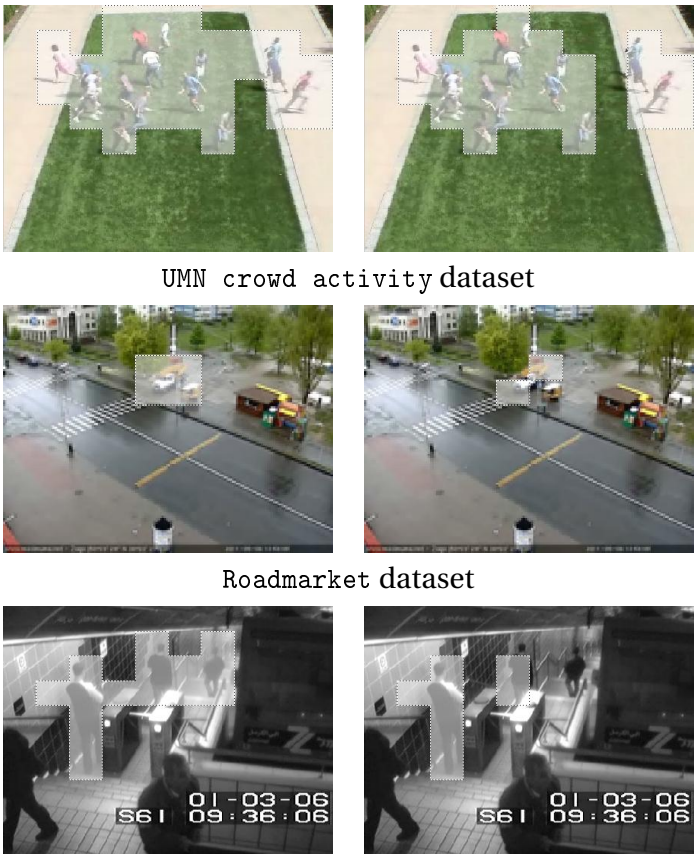


Figure 4.1: Ground truth examples of datasets `UMN crowd activity` dataset, `Roadmarket` dataset and `Exit platform`: First, all cells containing a part of the object triggering an anomaly are manually marked **(left)**. Then, all undetectable outlying cells, i.e., with motion magnitude lower than dataset threshold, are removed. The remaining cells **(right)** are finally used in method analysis.

4.2 Datasets

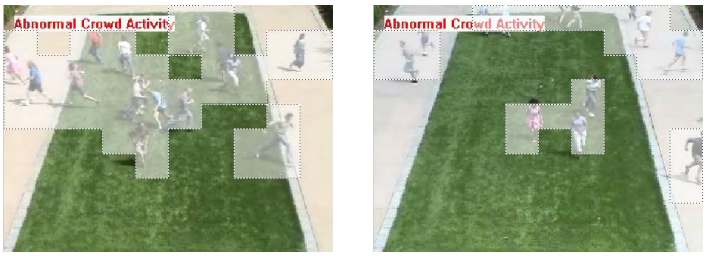
For the evaluation of the implemented outlier detection methods five different video datasets are used – four of them publicly available, one on request. In the following, each dataset is shown with examples of typical and abnormal frames including post-processed ground truth. Additionally, a time-line diagram shows at which points in time outliers are present at frame-level.

4.2.1 UMN crowd activity dataset

The UMN crowd activity dataset is provided by the Department of Computer Science and Engineering of the University of Minnesota and consists of several videos in which crowd escape panic is simulated. Two short clips cut together are used as a "proof of concept" video to show that a method works in principle. See Fig. 4.2 for typical frames and Fig. 4.3 for anomalous frames of the dataset, the label "Abnormal Crowd Activity" in the upper left corner is part of the video. Fig. 4.4 shows the ground truth of the dataset and Tab. 4.1 summarizes properties of dataset.

(a) **(b)**

Figure 4.2: Typical frame of the `UMN crowd activity` dataset dataset: people walking in unpredictable patterns.



(a) **(b)**

Figure 4.3: Anomalous frames of the `UMN crowd activity` dataset dataset: people escaping.

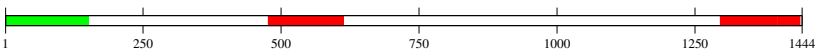


Figure 4.4: Ground truth of the `UMN crowd activity` dataset dataset. Frames used for training **(green)**, frames containing at least one outlier **(red)**.

UMN crowd activity dataset

dimension	crop	scale
320 × 240	**not cropped**	**1/2**
number of frames	frames per second	magnitude threshold
1.444 (edited version)	**30**	**5/100**
maximum motion magnitude		
2.25		

source
Department of Computer Science and Engineering of the University of Minnesota
`http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi`

Table 4.1: Properties of the `UMN crowd activity` dataset dataset.

4.2.2 UCSD pedestrian dataset

The UCSD pedestrian dataset is provided by the statistical visual computing laboratory (SVCL) of the University of California, San Diego. It is used in the work of Mahadevan et al. [66] and contains two datasets composed of clips showing a park scene with pedestrians. Pixel-level ground truth is partly available. Parts of the *peds1* set which consists of 70 clips of 200 frames length each are used in this work. To increase the anomaly rate, clips are cut together.

See Fig. 4.5 for typical frames and Fig. 4.6 for anomalous frames, which consist mostly of golf buggies driven through the scene. Fig. 4.7 shows the ground truth and Tab. 4.2 summarizes properties of the `UCSD pedestrian` dataset dataset.

(a) **(b)**

Figure 4.5: Typical frames of the `UCSD pedestrian` dataset.

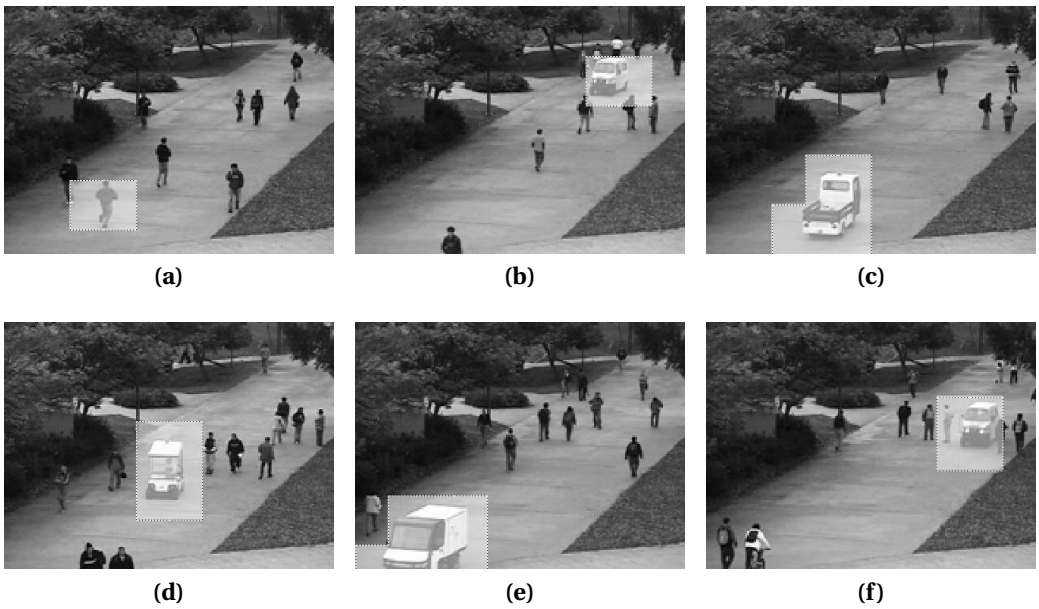


Figure 4.6: Abnormal frames of the `UCSD pedestrian` dataset: (a) person running and (b) – (f) golf buggies driven through the scene.

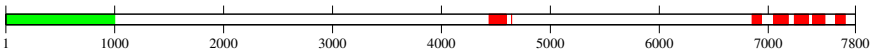


Figure 4.7: Ground truth of the `UCSD pedestrian` dataset. Frames used for training **(green)**, frames containing at least one outlier **(red)**.

UCSD pedestrian dataset

dimension	crop	scale
238 × 158	**not cropped**	**5/6**
number of frames	frames per second	magnitude threshold
7.800 (edited version)	**10**	**1/10**
maximum motion magnitude		
2		

source
Statistical Visual Computing Laboratory of the University of California, San Diego
`http://www.svcl.ucsd.edu/projects/anomaly/UCSD_Anomaly_Dataset.tar.gz`

Table 4.2: Properties of the `UCSD pedestrian` dataset.

4.2.3 Roadmarket dataset

The Roadmarket dataset, recorded during the OUTLIER project[1], observes a street with a T-junction and some market stalls. Three aspects make it challenging: First, the asphalt is wet, therefore many reflections are present, second, the camera moves sometimes, perhaps it is exposed to wind. Third, it is a recording whose frame rate is low and not constant over time, i.e., every once in a while there are larger gaps between frames. Consequently, motion estimation is more prone to error than with other datasets.

See Fig. 4.8 for typical frames and Fig. 4.9 for anomalous frames, which for the most part consist of vehicles turning around illegally. Ground truth is found in Fig. 4.10 and, finally, Tab. 4.3 sums up properties of the `Roadmarket` dataset.

(a) **(b)**

Figure 4.8: Typical frames of the `Roadmarket` dataset.

Roadmarket dataset

dimension	crop	scale
160 × 120	**not cropped**	**1**
number of frames	frames per second	magnitude threshold
5.147	**varying**	**1/10**
maximum motion magnitude		
7.5		

source
OUTLIER project: Joanneum Research, Siemens, TU Graz
`http://www.joanneum.at/?id=2841`

Table 4.3: Properties of the `Roadmarket` dataset.

[1] For details on the OUTLIER project, see the project web page: `http://www.joanneum.at/?id=2841`

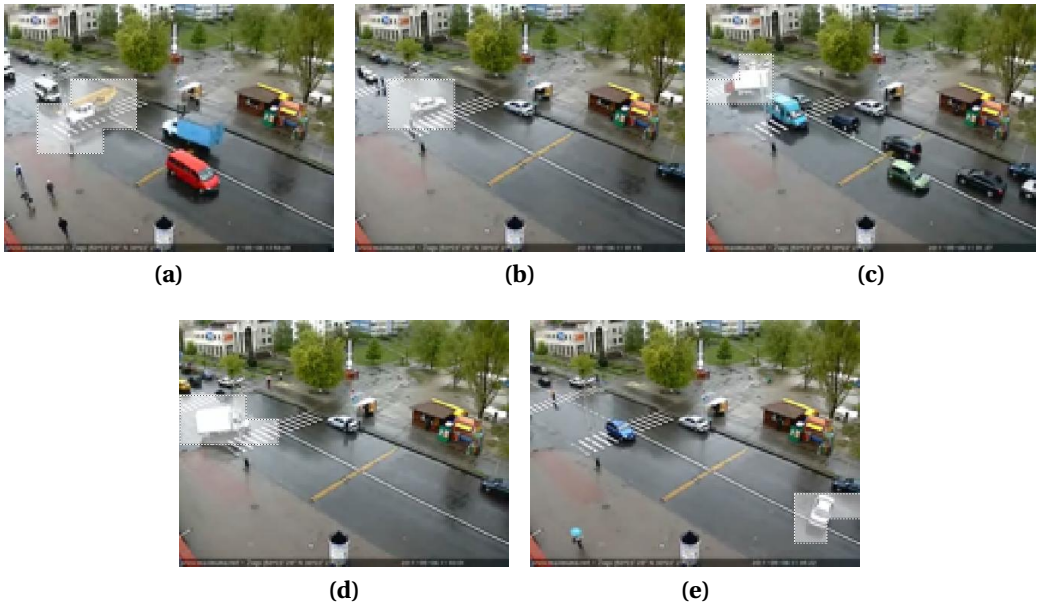


Figure 4.9: Abnormal frames of the `Roadmarket` dataset: (a) construction vehicle crossing the street, (b) vehicle turning illegally., (c) – (e) vehicles turning around illegally.

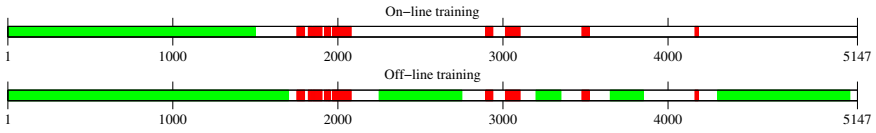


Figure 4.10: Ground truth of the `Roadmarket` dataset. Frames used for training **(green)**, frames containing at least one outlier **(red)**. On-line training of Adam's `HIST` approach is carried out using a window of 1.500 frames length**(above)**, off-line training is carried out using the majority of normal frames **(below)**.

4.2.4 Junction dataset

The Junction dataset is provided by the Computer Vision Group of the Queen Mary University of London and shows a typical, multi-lane traffic junction. The original version of the video is 90.000 frames long, of which in this work only the first half is used. Fig. 4.11 shows typical frames, Fig. 4.12 summarizes different typical anomalies: pedestrians, bikers or ambulances crossing the junction on unexpected paths. Fig. 4.13 shows the ground truth and Tab. 4.4 finally summarizes properties of the `Junction` dataset.

(a) (b)

Figure 4.11: Typical frames of the `Junction` dataset.

(a) (b) (c)

(d) (e) (f)

Figure 4.12: Abnormal frames of the `Junction` dataset: (a) and (b) bikers and (c) and (d) pedestrians crossing in unexpected paths. (e) and (f) ambulances driving in wrong direction.

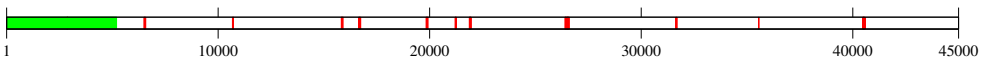


Figure 4.13: Ground truth of the `Junction` dataset. Frames used for training **(green)**, frames containing at least one outlier **(red)**.

junction dataset

dimension	crop	scale
360×288	$[9, 9]$ in $\mathbf{y}$	$4/9$
number of frames	frames per second	magnitude threshold
45.000	30	$3/10$
maximum motion magnitude		
3.5		

source
Computer Vision Group of the Queen Mary University of London
`http://www.eecs.qmul.ac.uk/~jianli/Dataset_List.html`

Table 4.4: Properties of the `Junction` dataset.

4.2.5 Underground train station exit platform dataset

The underground train station exit platform dataset is made available by Amit Adam. It consists of two videos, showing an entrance and an exit platform of an underground train station. In this work, the exit platform video is used – termed `Exit platform` dataset. Typical behavior in this video comprises the train arriving and leaving, people leaving the left platform after arrival of a train as well as people walking from left to right to get to the departure platform. The most remarkable outlier in this video is a group of men, who repeatedly enter the left platform by passing the turnstiles in the wrong direction. See Fig. 4.14 for typical frames, Fig. 4.16 for the ground truth. Finally, Fig. 4.16 again shows the ground truth and Tab. 4.5 summarizes properties of the `Exit platform` dataset.

(a) **(b)**

Figure 4.14: Typical frames of the `Exit platform` dataset: (a) people leaving the platform, (b) people going to the platform on the right.

(a) (b) (c)

Figure 4.15: Abnormal frames of the `Exit platform` dataset: (a) woman cleaning, (b) people crossing turnstiles in wrong direction and (c) person loitering.

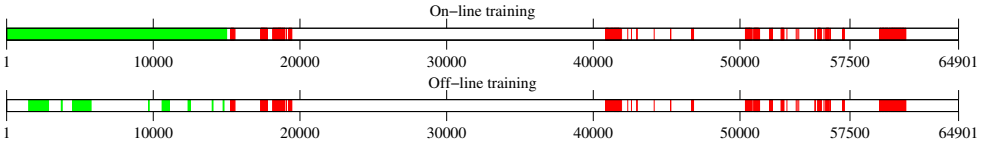


Figure 4.16: Ground truth of the `Exit platform` dataset. Frames used for training **(green)**, frames containing at least one outlier **(red)**. On-line training of Adam's `HIST` approach is carried out using a window of 15000 frames length**(above)**, off-line training is carried out using the marked regions containing significant normal events **(below)**.

Underground train station exit platform dataset

dimension	crop	scale
512×384	$[12, 12]$ in $\mathbf{x}$, $[16, 16]$ in $\mathbf{y}$	$1/3$
number of frames	frames per second	magnitude threshold
64.901	25	$1/10$
maximum motion magnitude		
7.2		
source		
Amit Adam, author of [1]		

Table 4.5: Properties of the `Exit platform` dataset.

4.3 Experiments

To allow for an in-depth analysis of the implemented methods a series of experiments is conducted. Primarily, the experiments are designed to cover the entire parameter range of each method. Secondly, each experiment is post-processed to find out whether the results can be improved by using Adam's Integration Procedure (see Chapter 3.3.1.1).

Adam's integration procedure is run with the following parameter ranges:

- $Z = \{1, \ldots, 5\}$
- pairs of $(Y, K) = \{(10, 9), (10, 8), (10, 7), (5, 4), (5, 3), (3, 2), (2, 1), (1, 1)\}$
- $N = \{0, \ldots, 5\}$

This procedure results in a total number of $5 \cdot 8 \cdot 6 = 240$ integration configurations evaluated.

For the `BASELINE` method, Shi's `STCOG` and Feng's `TASOM` method first the experiment is run and the result is stored. Then, all integration configurations are computed and the result provided by the best configuration is stored.

Another approach is chosen to evaluate Adam's `HISTOGRAM` method. The method is run delivering a result. Afterwards, to provide data to analyze the integration procedure itself, every integration configuration is computed and the corresponding result is stored.

4.3.1 Experiment configurations

The parameterizations and resulting numbers of experiments run on each dataset are:

- `BASELINE` – 1 experiment, which is also integrated, 2 results.
- `HISTOGRAM` – Adam's method is run in on-line and off-line mode and with either motion speed or direction used as feature. This results in $2 \cdot 2 = 4$ experiments, each of them is integrated 240 times. The total result number therefore is $4 + 4 \cdot 240 = 964$.

- `STCOG` – Shi's method is run on-line and off-line and in its original and modified (`M-STCOG`) variant. Parameters are assigned as follows:
  - $N_C = 4, K_{max} = 100, \beta = 1/10, \Delta = 1/10$
  - Number of frames used to form a pair-wise cuboid $L = \{1, \ldots, 5\}$
  - Probability threshold used for component creation $p_{max} = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$

  The total number of results including results obtained by integration is $2 \cdot 2 \cdot 5 \cdot 7 \cdot 2 = 280$.

- `TASOM` – Feng's method is run on-line and off-line and in its original and modified (M-TASOM) form. Additionally, three different distance measures are applied: $D_{BP_{KL}}, D_{BP_A}$ and $D_{BP_{AR}}$. Parameters are assigned as follows:
  - $\alpha = 1/10, \beta = 1/10, \eta_{max} = 1$
  - Lattice size $K = \{4, 6, 8\}$
  - Number of frames used to estimate a motion pattern $d = \{1, 3, 5\}$
  - Number of motion patterns per behavioral pattern $m = \{1, 2\}$
  - Maximal neighborhood size $\sigma_{max} = \{2, 4, 6, 8\}$. This value has to be smaller than the current K configuration.

  The total number including results obtained by integration is $2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 2 \cdot (2 + 3 + 4) = 1296$.

In sum **2.542** experiments are run on each dataset, which are subsequently analyzed with the performance measure discussed next.

4.3.2 Evaluation of performance measures

According to Dee and Velastin [29] the main tools available to compare the performance of outlier detection systems are Receiver Operating Characteristics (ROCs) and confusion matrices.

Both are based on the classification of data instances in one of the four following categories:

- **True positive (TP):** an outlier correctly identified.
- **False positive (FP):** a normal instance incorrectly classified as outlier.
- **True negative (TN):** a normal instance correctly identified.
- **False negative (FN):** an outlier incorrectly classified as normal instance.

ROC curves are plots of true positive rates against false positive rates as the threshold changes. They show the robustness of a method as a function of the threshold and help to identify the most robust threshold (Fawcett [32]). However, when evaluating outlier detection methods of video footage ROC curves are of limited interpretability, since the two classes compared – normal and anomalous data – differ significantly in size and hence the false positive rates are near zero for all useful thresholds.

From confusion matrices, on the other hand, a variety of summary statistics can be derived using the category counts above (Dee and Velastin [29], Altman and Bland [5] and Baldi et al. [6]). The statistics evaluated in this work are summarized in the following:

- The **positive predictive value** or **precision** is the proportion of correct outliers detected:
$$PPV = \frac{TP}{TP + FP}.$$
- The $\boldsymbol{F_\beta}$ **score**
$$F_\beta = (1+\beta^2)\frac{PPV \cdot TPR}{(\beta^2 \cdot PPV) + TPR} = \frac{(1+\beta^2)TP}{((1+\beta^2)TP + \beta^2 FN + FP}$$
is a measure of a test's accuracy and can be interpreted as a weighted average of precision and recall. In the following evaluation, $\beta = \{0.5, 1, 2\}$ is used. Note that for $\beta = 0$ the F_β score is equal to *PPV*.
- **Matthew's correlation coefficient**
$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

is the only statistical measure taking into account all information of the confusion matrix and is essentially a correlation coefficient between ground truth and classification result. The value of MCC lies between -1 and +1. A coefficient of 1 represents a perfect method, 0 an average random method and -1 an inversely perfect method.

4.3.3 Comparison of performance measures

In order to identify the most suitable measure to assess the performance of an experiment configuration, an evaluation of the following performance measures is carried out: three F_β scores, namely $F_{0.5}, F_1$ and F_2, the *PPV* and the *MCC* measure. In all experiments, the best configuration of all 2.542 experiments is determined by simply averaging the scores obtained for the frame statistics and those for the cell statistics, i.e. for the *MCC* measure:

$$MCC = \frac{1}{2}\left(MCC_{frames} + MCC_{cells}\right).$$

The best performers on the `UMN crowd activity` dataset compared to the ground truth according to the different measures are depicted in Fig. 4.17. All measures consistently report Feng's `TASOM` method to be the top performer, although the *PPV* reports the top score for several methods. The respective confusion matrix entries are given in Tab. 4.6.

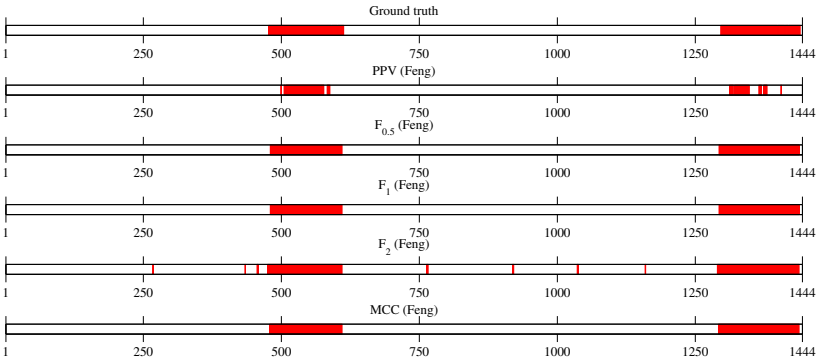


Figure 4.17: Performance measure comparison on the `UMN crowd activity` dataset compared to the ground truth.

	frames				cells			
Method	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**
Ground truth	276	0	0	1.167	5.132	0	0	139.168
PPV	123	0	153	1.167	410	0	4.722	139.168
$F_{0.5}$	268	8	8	1.159	3.353	258	1.779	138.910
F_1	268	8	8	1.159	3.464	304	1.668	138.864
F_2	271	25	5	1.141	3.789	394	1.343	138.774
MCC	269	9	7	1.158	3.357	254	1.775	138.914

Table 4.6: Confusion matrix entries of the best performers regarding to different evaluation measures for the `UMN crowd activity` dataset.

For the `UCSD pedestrian` dataset the best performers are found in Fig. 4.18. Adam's `HISTOGRAM` method outperforms all others with respect to all performance measures except the *PPV* measure, which determines Feng's `TASOM` to be the top performer. The associated confusion matrix entries are listed in Tab. 4.7.

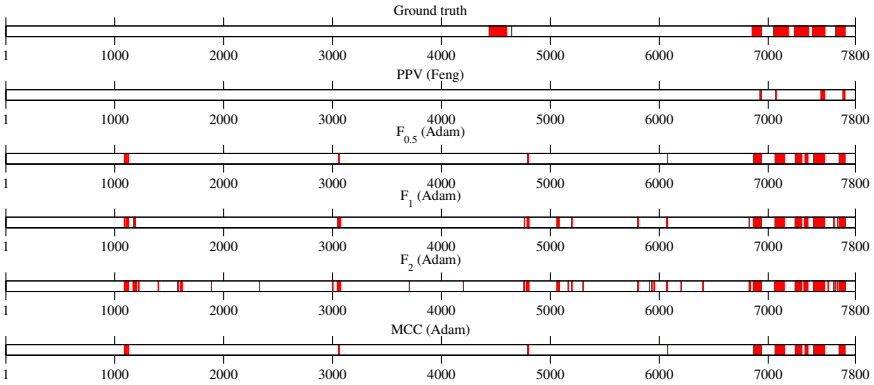


Figure 4.18: Performance measure comparison on the `UCSD pedestrian` dataset compared to the ground truth.

	frames				cells			
Method	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**
Ground truth	697	0	0	7.102	3.245	0	0	776.655
PPV	45	0	652	7.102	54	0	3.191	776.655
$F_{0.5}$	384	20	313	7.083	1.220	176	2.025	776.579
F_1	393	41	304	7.062	1.274	207	1.971	776.548
F_2	476	392	221	6.711	1.336	615	1.909	776.140
MCC	384	20	313	7.083	1.220	176	2.025	776.579

Table 4.7: Confusion matrix entries of the best performers regarding to different evaluation measures for the `UCSD pedestrian` dataset.

As can be seen in Fig. 4.18, *PPV* favors low false positive rates leading to selection of experiment configurations that also have low true positive rates. On the `UMN crowd`

`activity` dataset, *PPV* is the only measure selecting a configuration that misses 153 out of 276 frames, with the `UCSD pedestrian` dataset it even misses 652 of 697 frames. Thus a required trade-off between false and true positives cannot be achieved with *PPV*.

With the F_β scores for $\beta = 1$ and $\beta = 2$ a converse observation is made: high *TP* values are achieved at the cost of additional false positives, which can be observed for F_2 on the `UMN crowd activity` dataset in Fig. 4.17 and for both F_1 and F_2 on the `UCSD pedestrian` dataset in Fig. 4.18.

The other two performance measures, $F_{0.5}$ and *MCC* perform almost equally on the `UMN crowd activity` dataset and select the same experiment configuration on the `UCSD pedestrian` dataset. Therefore, a comparison between those two is conducted on the `Exit platform` dataset. Results are shown in Fig. 4.19 and Tab. 4.8.

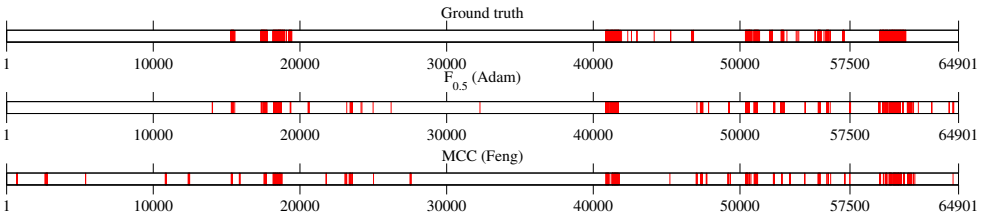


Figure 4.19: Performance measure comparison on the `Exit platform` dataset compared to the ground truth.

	frames				cells			
Method	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**
Ground truth	4.452	0	0	60.449	25.984	0	0	6.464.116
$F_{0.5}$	2.845	1.008	1.607	59.441	8.642	3.363	17.342	6.460.753
MCC	2.831	2.941	1.621	57.508	12.793	12.052	13.191	6.452.064

Table 4.8: Confusion matrix entries of the best performers regarding to different evaluation measures for the `Exit platform` dataset.

The MCC measure attempts to reach a balanced configuration leading to nearly equal true positive and false positive counts. Clearly, this is not the desired objective.

The $F_{0.5}$ score, on the other hand, reaches a desired level of true positives while allowing for an acceptable false positive level. For this reason, the $F_{0.5}$ score was chosen as performance measure. All results presented in the following are based on $F_{0.5}$ score analysis.

4.4 Results

The results are presented in three parts: In the first, a detailed comparison regarding the performance differences between the original and the modified variants of the methods is given. These modified methods encompass: the algorithmic modifications, the application of the extension of Adam's Integration method and, for Feng's `TASOM` approach, the usage of three distance measures and the application of multi-model maintenance. The comparison is carried out with respect to the five video datasets on the three implemented methods, namely, Adam's `HIST` method as well as Shi's `STCOG` as Feng's `TASOM` approach.

Since on-line ability is central to a method's applicability in real-word scenarios, in the second part the performance differences of experiment configurations using on-line and off-line model training are analyzed. Again, the comparison is based on experimental results of all five datasets.

Finally, the best performers on all datasets including detection examples are presented and a result summary is given. All results are ranked using the $F_{0.5}$ score introduced in the previous evaluation.

4.4.1 Original vs. modified variants

To allow for a comparison of the performances of the original methods and their modified variants, results are given for the original method and combinations of modifications used. For example, for Shi's `STCOG` approach, results are given for the original version, the algorithmically modified version only and a version including the algorithmic modification and Adam's integration step. By doing so, a statement on the contribution of individual modifications can be given.

4.4.1.1 Adam's `HIST` approach

The approach of Adam et al. [1] is methodically the simplest of the three compared methods, employing a cyclic buffer to store observations and calculating a new observation's probability based on those currently in the buffer. It tends to report a large number of volatile false alarms, which are efficiently filtered using the integration method proposed. The main modification made to Adam's approach is an

extension of this procedure. In addition to the three original integration parameters: Z, the number of alerting cells making up an "alarming" frame and the pair of parameters, K and Y, controlling the filtering process of volatile events, a forth parameter, N, is added to configure the minimal size of an event. This simple idea is based on the observation that objects in scenes normally cover more than one cell and hence smaller objects can be filtered without decreasing the detection rate. The subsequent comparison analyses the findings in applying the modified integration procedure for both binning strategies proposed: angular binning (A) of the motion direction and radial binning (R) of its magnitude.

In addition to the cyclic buffering approach, experiments with buffers filled once in an off-line fashion were conducted, which for the majority of datasets yielded superior results. Detailed performance discussion regarding on-line vs. off-line performance is deferred to Chapter 4.4.2, in the following the best result is used. Find a result summary in Tab. 4.9 including the true positive, false positive and false negative counts on a frame and cell level. For each dataset the individual experiment configurations are ranked by their achieved overall $F_{0.5}$ score.

		frames			cells			
Dataset	**variant**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**$F_{0.5}$ %**
UMN crowd dataset	orig., A	194	137	82	468	213	4.664	45,2
	mod., A, $N = 4$	171	19	105	1.274	318	3.858	69,0
	orig., R	225	3	51	2.284	211	2.848	85,1
	mod., R, $N = 3$	236	3	40	2.587	300	2.545	86,7
UCSD pedestrian dataset	orig., A	81	158	616	145	510	3.100	18,4
	mod., A, $N = 1$	113	210	584	443	1.755	2.802	23,4
	orig., R	393	37	304	827	54	2.418	71,2
	mod., R, $N = 1$	384	20	313	1.220	176	2.025	76,1
Roadmarket dataset	orig., R	156	400	288	209	556	1.191	26,3
	mod., R, $N = 4$	75	41	369	300	405	1.100	38,4
	orig., A	126	69	318	254	141	1.146	47,0
	mod., A $N = 3$	122	24	322	484	453	916	53,2
Junction dataset	orig., R	53	1.790	569	15	5.525	1.120	1,8
	mod., R, $N = 5$	44	1.316	578	33	12.320	1.102	2,0
	orig., A	150	18	472	460	63	675	64,6
	mod., A, $N = 1$	143	5	479	457	46	678	65,8
Exit platform dataset	orig., R	4.208	5.519	244	23.405	57.902	2.579	40,9
	mod., R, $N = 0$	4.208	5.519	244	23.405	57.902	2.579	40,9
	orig., A	2.845	1.008	1.607	8.642	3.363	17.342	65,0
	mod., A, $N = 0$	2.845	1.008	1.607	8.642	3.363	17.342	65,0

Table 4.9: True positive, false positive and false negatives on all five dataset for Adam's `HIST` approach. Values are given for the original variant and for the modified variant using the extended integration procedure.

As described in Adam et al. [1], depending on a given scene, one of the two binning strategies is generally more suitable than the other. The superior strategy is usually the one a user would choose intuitively when configuring the system. For the shorter and simpler datasets, `UMN crowd` and `UCSD pedestrian`, the radial binning provides better performance, whereas for the three remaining datasets angular binning performs better. The biggest performance difference is found for the `Junction` dataset, where the motion magnitude carries no information about a given data instance being an outlier and hence, the experiment performances are dominated by false positives.

The proposed integration procedure extension improves the results for all datasets except the `Exit platform` dataset. This improvement is not significant for the `UMN crowd` dataset, because most of the outliers cover several dozen cells, nor for the `Junction` dataset, because the reported outliers already are very sparse. For the remaining two datasets, however, significant improvement is observed. For the `UCSD pedestrian` dataset as well as the `Roadmarket` dataset, the vehicles exhibiting unexpected behavior are larger than all other objects.

Depending on the size of the objects expected to show unexpected behavior in a scene, the appliance of the extended integration procedure is advantageous.

4.4.1.2 Shi's `STCOG` approach

For Shi's `STCOG` approach, two modifications are applied, namely algorithmic modifications and the use of the extended integration procedure. Algorithmic modifications comprise an extension of the adaptation mechanism using an aging concept of the GMM components and the incorporation of the motion variance available through the Unified motion feature procedure. For all experiments conducted using the modified method, the extended integration procedure is applied to filter volatile events. See Fig. 4.10 for results of the original and the modified variants on all five datasets.

Shi's `STCOG` approach performs acceptably well on the `UMN crowd`, `UCSD pedestrian` and `Roadmarket` datasets, although it tends to report only on a small part of the object responsible for the outlier. For the other two datasets, however, it does not allow to robustly separate abnormal from normal data instances. Neither

		frames			cells			
Dataset	**variant**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**$F_{0.5}$ %**
UMN crowd dataset	orig.	223	13	53	2.029	398	3.103	79,9
	mod.	261	22	15	3.057	680	2.075	84,4
	mod. int.	250	15	26	3.046	673	2.086	84,8
UCSD pedestrian dataset	orig.	320	77	377	578	312	2.667	56,3
	mod.	298	31	399	805	358	2.440	62,5
	mod. int.	382	23	315	1.059	144	2.186	74,1
Roadmarket dataset	orig.	95	168	349	143	288	1.257	27,3
	mod.	73	53	371	124	86	1.276	33,1
	mod. int.	56	7	388	118	10	1.282	35,5
Junction dataset	orig.	77	1.624	545	60	3.061	1.075	3,7
	mod.	75	199	547	113	230	1.022	22,2
	mod. int.	63	67	559	94	68	1.041	27,0
Exit platform dataset	orig.	3.481	6.928	971	18.343	51.987	7.641	33,8
	mod.	3.126	5.469	1.326	17.693	45.022	8.291	36,1
	mod. int.	3.203	4.861	1.249	19.014	50.638	6.970	37,4

Table 4.10: True positive, false positive and false negatives on all five dataset for Shi's `STCOG` approach. Results are given for the original, the algorithmically modified variant and for the modified variant integrated using Adam's procedure.

from the experimental results nor from the detection behavior a distinct reason for this inability is deducible. Since the motion information is present in the feature vector, it is likely that GMMs are not able to express more complex scene normality. Also, these difficulties remain the same for feature spaces of different dimensions.

In practice, the first cells to fire false alarms are typically the corner cells, since they are modeled using two STCOGs only, but also rare normal events are likely to trigger false alarms, i.e., higher vehicles in the `Junction` dataset such as buses or trucks. Although the false-alarm rate is significantly reduced when using the modified variant and can moreover be remedied to some extent by applying Adam's integration procedure – as results in Tab. 4.10 show – in summary Shi's approach is too sensitive for real-world applications.

4.4.1.3 Feng's `TASOM` approach

The modifications and extensions to Feng's `TASOM` method are threefold: First, algorithmic modifications are implemented to increase the adaptability of the model, second, two additional difference measures are proposed to replace the KL-divergence, namely an angular measure (A) and a combined angular-magnitude measure (AR).

Finally, the usage of multiple TASOM models is analyzed using the Junction dataset. In Tab. 4.11 results ranked by $F_{0.5}$ are found for all five datasets. Again, experiment configurations displayed encompass the original method, the algorithmically modified variant for all three difference measures, and, finally, integrated results for the modified methods. In contrast to the other datasets, an additional set of results is given for the Junction dataset, for which the multiple-model TASOM approach was analyzed.

Similar to the performance results found for the two binning strategies of Adam's approach, the various difference measures suit different scenes in varying degrees. While for the UMN crowd and the UCSD pedestrian dataset the angular measure does not work, it outperforms the other measures significantly for the Roadmarket and the Junction dataset. Apparently, the entropy-based KL-divergence measure – computationally the most complex of the three measures compared – is not suitable for a range of scenes and can be replaced by other measures with comparable performance for all other scenes.

The application of Adam's integration procedure is advantageous for all datasets. The bigger the challenge presented by a dataset, the greater is the advantage of integration, since its primarily filters volatile events and does not affect outliers detected robustly. See, for example, the performance of integration on the Exit platform dataset: although the angular R measure outperforms the mixture measure AR significantly without integration, after integration both perform comparably.

Finally, multiple TASOM models were applied to the Junction dataset since the best standard approach performs weakly with an $F_{0.5}$ score of 31,4%. The poor performance of the original approach is understandable since unexpected object behavior is determined mainly by unexpected motion direction in a local neighboorhood. These outliers, however, cannot be detected by the standard model, since the same motion direction is normal in another part of the scene. Therefore, the image plane is divided into 12 rectangular zones, two bigger zones in the upper part of the plane and 10 smaller zones in the busier lower part. To avoid false alarms at the border between zones, each zones's model is trained using the inner as well as its surrounding cells. See Fig. 4.20 for a depiction of the 12 zones used.

Although computational effort grows linearly with the number of models, it is justified by the increase in detection performance. The two most prominent out-

		frames			cells			
Dataset	**variant**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**$F_{0.5}$ %**
UMN crowd dataset	mod. R	215	226	61	533	394	4.599	41,4
	mod. R int.	161	14	115	986	584	4.146	62,8
	mod. AR	263	17	13	2.495	122	2.637	87,1
	mod. AR int.	244	8	32	2.760	143	2.372	88,7
	orig. KL	270	12	6	3.219	329	1.913	89,7
	mod. KL	271	13	5	3.288	340	1.844	89,8
	mod. KL int.	268	8	8	3.353	258	1.779	91,4
UCSD pedestrian dataset	mod. R	139	581	558	298	868	2.947	19,1
	mod. R int.	52	6	645	315	135	2.930	29,6
	mod. AR	240	68	457	760	106	2.485	59,4
	orig. KL	235	35	462	737	74	2.508	61,5
	mod. KL	252	45	445	792	105	2.453	62,4
	mod. AR int.	203	5	494	811	44	2.434	63,6
	mod. KL int.	238	17	459	793	48	2.452	64,7
Road-market dataset	orig. KL	293	1.571	151	387	2.931	1.013	15,9
	mod. KL	249	1.106	195	301	1.912	1.099	18,0
	mod. AR	149	219	295	163	351	1.237	31,2
	mod. R	186	255	258	270	380	1.130	37,9
	mod. KL int.	81	38	363	201	117	1.199	40,8
	mod. AR int.	131	31	313	184	46	1.216	49,8
	mod. R int.	183	90	261	345	170	1.055	54,7
Junction dataset	orig. KL	66	3.222	556	90	4.718	1.045	2,3
	mod. AR	131	5.909	491	146	7.729	989	2,4
	mod. KL	32	1.530	590	49	2.018	1.086	2,5
	mod. R	65	1.028	557	81	1.129	1.054	6,6
	mod. AR int.	71	663	551	72	760	1.063	9,0
	mod. KL int.	25	91	597	50	189	1.085	11,7
	mod. KL, 12 models	50	152	572	74	152	1.061	17,8
	mod. KL int., 12 models	46	19	576	68	20	1.067	24,5
	mod. AR, 12 models	95	119	527	127	161	1.008	30,0
	mod. R int.	60	55	562	160	126	975	31,4
	mod. R, 12 models	120	66	502	309	116	826	49,2
	mod. AR int., 12 models	108	27	514	283	103	852	49,6
	mod. R int., 12 zones	138	0	484	491	48	644	66,7
Exit platform dataset	orig. KL	3.083	4.508	1.369	11.679	18.402	14.305	42,1
	mod. KL	3.307	4.861	1.145	11.324	18.721	14.660	41,6
	mod. KL int.	1.103	636	3.349	6.082	5.094	19.902	45,7
	mod. R	1.784	822	2.668	6.659	1.752	19.325	57,9
	mod. R int.	1.361	303	3.091	7.039	1.763	18.945	59,4
	mod. AR	2.031	1.436	2.421	7.043	4.481	18.941	52,1
	mod. AR int.	1.684	456	2.768	7.624	3.427	18.360	59,5

Table 4.11: Results on all five datasets using Feng's `TASOM` approach. Values are given for the original method and for individual difference measures on the algorithmically modified method. For the modified method, results after applying the integration procedure are also given as well as those for the multiple model variant on the `Junction` dataset.

liers, i.e., ambulances driving in forbidden directions, are flawlessly detected. On the other hand, outliers triggered by smaller objects such as pedestrian or bikers, are missed, which is due to the cell size used for detection.

To sum up, Feng's original `TASOM` approach performs well on simpler datasets but cannot convince on more difficult ones. The proposed algorithmic modifica-

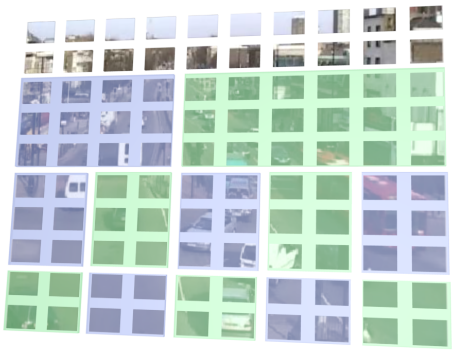


Figure 4.20: Multiple `TASOM` models on the `Junction` dataset. To improve the detection rate and allow for local modeling of normality multiple models are maintained, depicted as green and blue rectangles.

tion improves the method's performance on four datasets, but the improvement is not significant. The application of Adam's integration procedure leads to a further performance increase, but still the method does not perform well on all datasets. The reasons are, on the one hand, KL-divergence being an inappropriate distance measure and, on the other, the approach's inability to model localized scene normality. Only the usage of additional distance measures proposed and deployment of multiple models are able to raise the method's performance to a competitive level.

4.4.2 On-line vs. off-line performance

One of the most important aspects of an outlier detection method is its ability to adapt to the changes in the scene due to environment changes such as changing illumination or weather condition, something that can only be achieved in an on-line fashion. Surprisingly, many of the methods proposed in the literature are not able to adapt to changes of the observed scene, but train their models in an off-line fashion only. The three methods analyzed in this work, however, offer this possibility. In this Chapter their on-line abilities are investigated by comparing their performance to that of their off-line counterparts.

Although the histogram-based method proposed by Adam et al. [1] is originally proposed to only operate in an on-line fashion using a cyclic buffer, its off-line abil-

ities are also examined in this work. Using the same training data as for all other methods, a fixed length buffer is filled and used to detect outliers.

In addition to their off-line capabilities, for Shi's STCOG and Feng's TASOM approach concepts are proposed to constantly adapt the scene models in an on-line fashion. While for the STCOG all GMMs are continuously trained using the incoming data, the TASOM model is updated using a randomly selected subset of data instances only. See Fig. 4.12 for results on all five datasets ranked by $F_{0.5}$ score.

		frames			cells			
dataset	**method**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**$F_{0.5}$ %**
UMN crowd activity dataset	STCOG off-line	255	6	21	2.128	285	3.004	84,3
	STCOG on-line	250	15	26	3.046	673	2.086	84,8
	HIST on-line	213	2	63	3.012	476	2.120	86,3
	HIST off-line	236	3	40	2.587	300	2.545	86,7
	TASOM on-line	252	5	24	3.278	343	1.854	90,1
	TASOM off-line	268	8	8	3.353	258	1.779	91,4
UCSD pedestrian dataset	HIST on-line	247	52	450	799	363	2.446	57,9
	TASOM on-line	213	14	484	842	155	2.403	62,3
	TASOM off-line	238	17	459	793	48	2.452	64,7
	STCOG on-line	363	15	334	946	127	2.299	72,4
	STCOG off-line	382	23	315	1.059	144	2.186	74,1
	HIST off-line	384	20	313	1.220	176	2.025	76,1
Roadmarket dataset	STCOG on-line	85	309	359	217	2.031	1.183	15,7
	STCOG off-line	56	7	388	118	10	1.282	35,5
	TASOM off-line	150	67	294	257	115	1.143	50,8
	HIST off-line	122	29	322	271	127	1.129	51,8
	HIST on-line	122	24	322	484	453	916	53,2
	TASOM on-line	183	90	261	345	170	1.055	54,7
Junction dataset	STCOG on-line	31	274	591	43	529	1.092	7,4
	HIST on-line	37	77	585	128	305	1.007	19,7
	STCOG off-line	63	67	559	94	68	1.041	27,0
	TASOM on-line	138	0	484	464	75	671	64,6
	HIST off-line	143	5	479	457	46	678	65,7
	TASOM off-line	138	0	484	491	48	644	66,7
Exit platform dataset	STCOG off-line	2.917	4.515	1.535	16.380	42.981	9.604	36,9
	STCOG on-line	3.203	4.861	1.249	19.014	50.638	6.970	37,4
	HIST on-line	3.511	4.134	941	13.813	27.815	12.171	43,0
	TASOM on-line	1.361	303	3.091	7.039	1.763	18.945	59,4
	TASOM off-line	1.684	456	2.768	7.624	3.427	18.360	59,5
	HIST off-line	2.845	1.008	1.607	8.642	3.363	17.342	65,0

Table 4.12: True positive, false positive and false negatives on all five dataset with Shi's STCOG approach. Values are given for the original variant and the algorithmically modified variant, in both cases with and without appliance of Adam's integration procedure.

The cyclic buffer of Adam's HIST approach tends to decrease its on-line performance. For all but one dataset, namely, the Roadmarket dataset, the off-line HIST approach outperforms its on-line counterpart significantly. This is because of two reasons: First, only two of the five real-world scenes dataset evaluated

in this work exhibit phases, namely, the `Junction` and `Exit platform` dataset. For all other datasets, the cyclic buffer approach has no real-world equivalence. Second, for the given datasets, off-line training data represents scene normality to a sufficient extent, while, when moving the buffer window along the video, normality is often only poorly represented. At the beginning of the `UCSD pedestrian` dataset, for example, larger crowds are observed encompassing the normal behavior adequately, while only occasionally observed pedestrians later on do not represent the scene's normality sufficiently. Summarized, the simple adaptability concept is often not sufficient for real-world scenes.

The on-line and off-line achievements of the `STCOG` approach proposed by Shi et al. [95] are very similar on three of the datasets, i.e., `UMN crowd activity`, `UCSD pedestrian` and `Exit platform` datasets. For both the `Roadmarket` and `Junction` dataset, the on-line variant suffers from massive false positives and is clearly unable to model the scene. This behavior is related to the modeling issues already analyzed in Sec. 4.4.1.2.

Last but not least, Feng's `TASOM` approach delivers the most convincing on-line performance in that the results are comparable to its off-line performance for all five datasets. Compared to the other methods, this behavior can be interpreted in so far that the on-line adaptation concept is working. Of course, due to the length of the datasets evaluated, a decisive statement regarding the on-line performance cannot be made.

4.4.3 Best performers

Finally, the best performers of all three methods are shown on each of the datasets using three different forms of representation. Again, the confusion matrix statistics are used, but, in addition, time-line comparisons of detected outliers on a frame level and examples of correct detections as well as false alarms are shown. Detailed results for the best performers in 42 different classes of experiment configurations, i.e., difference measure or binning strategy used, on-line/off-line variant, integration, are enclosed. In addition, the results for the `BASELINE` method are included to convey a sense of what is possible with a trivial method.

4.4.3.1 `UMN crowd activity` dataset

The `UMN crowd activity` dataset shows a crowd moving in unpredictable patterns and escaping collectively. Two scenes are edited together giving two outliers.

All four methods perform well on the `UMN crowd activity` dataset. The `M-TASOM` is the only method achieving an $F_{0.5}$ score higher than 90%, all other methods fall back slighty to at around 85% – see Tab. 4.13. See Fig. 4.21 for a time-line comparison of the results with the ground truth at frame-level. Interestingly, the simple `BASELINE` performs better than the `M-STCOG` approach, which is due to its sensitivity to the cut in the middle of the video, which was edited in to concatenate two sequences.

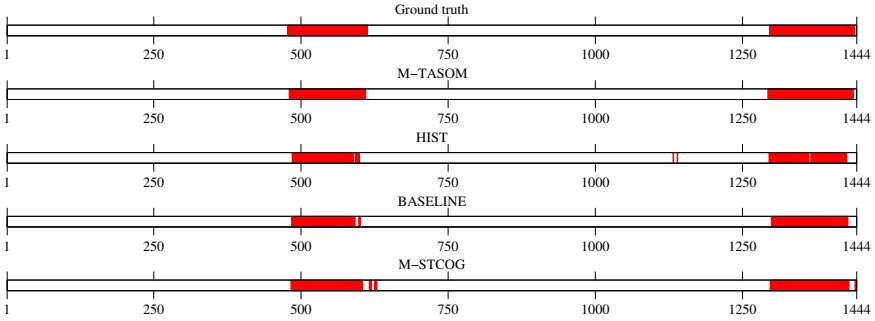


Figure 4.21: Best performers of all four methods on the `UMN crowd activity` dataset.

Examples of detections are shown in Fig. 4.22 for all four methods, detailed results for all 42 categories are found in Tab. 4.13.

(a) **(b)** **(c)** **(d)**

Figure 4.22: Examples of outliers detected in the `UMN crowd activity` dataset. True positives **(red)**, false positives **(yellow)** and false negatives **(white)** for **(a)** Modified Feng's `TASOM` approach, **(b)** Adam's `HIST` approach, **(c)** `BASELINE` and **(d)** modified Shi's `STCOG` approach.

pos.	model	on-line	Δ	parameters	Z, Y, K, N	TP_f %	TN_f %	$F_{0.5,f}$ %	TP_c %	TN_c %	$F_{0.5,c}$ %	∅ $F_{0.5}$ %
1	M-TASOM		KL	$K=4, \sigma_{max}=4, d=5, m=1$	1, 5, 3, 0	97.1	99.3	97.1	65.3	99.8	85.6	91.4
2	TASOM		KL	$K=4, \sigma_{max}=2, d=5, m=1$	1, 3, 2, 0	97.1	99.2	96.8	64.3	99.8	85.5	91.1
3	M-TASOM	✓	KL	$K=8, \sigma_{max}=2, d=5, m=1$	1, 1, 1, 1	91.3	99.6	96.6	63.9	99.8	83.6	90.1
4	M-TASOM	✓	KL	$K=4, \sigma_{max}=4, d=5, m=1$		98.2	98.9	96.0	64.1	99.8	83.7	89.8
5	TASOM	✓	KL	$K=4, \sigma_{max}=2, d=5, m=1$	1, 1, 1, 1	90.2	99.6	96.4	62.3	99.8	83.2	89.8
6	TASOM	✓	KL	$K=4, \sigma_{max}=2, d=5, m=1$		97.8	99.0	96.2	62.7	99.8	83.3	89.7
7	M-TASOM		KL	$K=4, \sigma_{max}=4, d=5, m=1$		97.8	98.8	95.6	64.1	99.8	83.8	89.7
8	TASOM		KL	$K=4, \sigma_{max}=2, d=5, m=1$		97.5	98.9	95.8	62.9	99.8	83.5	89.6
9	TASOM		AR	$K=8, \sigma_{max}=2, d=3, m=1$	1, 2, 1, 1	88.4	99.3	95.0	53.8	99.9	82.4	88.7
10	TASOM	✓	AR	$K=8, \sigma_{max}=2, d=3, m=1$	2, 2, 1, 0	86.6	99.5	95.1	53.1	99.9	81.7	88.4
11	M-TASOM		AR	$K=8, \sigma_{max}=2, d=3, m=1$	1, 2, 1, 1	88.0	99.3	94.9	52.5	99.9	81.8	88.3
12	M-TASOM	✓	AR	$K=8, \sigma_{max}=2, d=3, m=1$	1, 2, 1, 1	88.0	99.3	94.9	52.1	99.9	81.2	88.1
13	TASOM	✓	AR	$K=8, \sigma_{max}=2, d=3, m=1$		95.3	98.5	94.2	48.6	99.9	80.0	87.1
14	TASOM		AR	$K=8, \sigma_{max}=2, d=3, m=1$		94.9	98.5	93.8	48.5	99.9	79.8	86.8
15	HIST		R		4, 3, 2, 3	85.5	99.7	95.8	50.4	99.8	77.5	86.7
16	HIST	✓	R		5, 3, 2, 5	77.2	99.8	93.8	58.7	99.7	78.9	86.3
17	M-TASOM	✓	AR	$K=8, \sigma_{max}=2, d=5, m=1$		96.4	98.5	94.5	46.4	99.9	78.1	86.3
18	M-TASOM		AR	$K=8, \sigma_{max}=2, d=5, m=1$		95.3	98.3	93.4	46.4	99.9	78.1	85.7
19	BASELINE				1, 3, 2, 1	86.2	100.0	96.9	40.2	99.9	74.6	85.7
20	M-STCOG	✓		$L=4, p_{max}=0.5$	1, 1, 1, 1	90.6	98.7	93.6	59.4	99.5	76.1	84.8
21	M-STCOG	✓		$L=4, p_{max}=0.5$		94.6	98.1	92.7	59.6	99.5	76.1	84.4
22	M-STCOG			$L=2, p_{max}=1$	1, 5, 3, 1	92.4	99.5	96.6	41.5	99.8	72.0	84.3
23	STCOG	✓		$L=3, p_{max}=1$	1, 2, 1, 1	79.3	98.9	90.9	45.8	99.6	69.8	80.4
24	STCOG	✓		$L=4, p_{max}=1$		80.8	98.9	91.4	39.5	99.7	68.4	79.9
25	STCOG			$L=3, p_{max}=5E-02$	3, 2, 1, 3	64.1	99.3	87.1	40.0	99.7	69.2	78.2
26	M-STCOG			$L=1, p_{max}=1E-03$		93.8	96.2	87.0	33.1	99.9	66.8	76.9
27	BASELINE					93.5	90.8	74.3	41.9	99.8	71.4	72.9
28	HIST	✓	A		1, 10, 9, 4	62.0	98.4	82.5	24.8	99.8	55.4	69.0
29	HIST		A		1, 10, 9, 5	60.1	98.5	82.0	21.0	99.7	49.5	65.8
30	STCOG			$L=2, p_{max}=0.5$		77.9	91.4	69.8	25.8	99.8	57.4	63.6
31	TASOM		A	$K=4, \sigma_{max}=2, d=5, m=2$	1, 2, 1, 5	58.3	98.8	82.5	19.2	99.6	43.2	62.8
32	M-TASOM		A	$K=6, \sigma_{max}=6, d=5, m=2$	1, 2, 1, 5	50.7	99.3	80.6	16.3	99.7	40.9	60.7
33	HIST		R			98.2	74.3	52.9	46.9	99.4	67.4	60.1
34	TASOM	✓	A	$K=6, \sigma_{max}=6, d=5, m=2$	3, 5, 3, 2	60.5	98.7	83.2	12.0	99.8	35.2	59.2
35	M-TASOM	✓	A	$K=8, \sigma_{max}=4, d=5, m=2$	1, 2, 1, 3	74.3	97.2	83.5	14.7	99.5	33.9	58.7
36	HIST	✓	R			94.6	54.0	37.6	44.4	98.7	53.3	45.5
37	TASOM		A	$K=8, \sigma_{max}=2, d=5, m=1$		77.9	80.6	52.7	10.4	99.7	30.1	41.4
38	M-TASOM		A	$K=4, \sigma_{max}=2, d=5, m=1$		82.2	77.7	51.0	11.4	99.7	31.5	41.3
39	TASOM	✓	A	$K=8, \sigma_{max}=2, d=5, m=1$		78.3	80.7	52.9	10.2	99.7	29.5	41.2
40	M-TASOM	✓	A	$K=6, \sigma_{max}=2, d=5, m=1$		86.2	74.0	48.8	13.2	99.6	33.6	41.2
41	HIST	✓	A			99.3	22.5	27.4	59.8	92.4	25.8	26.6
42	HIST		A			99.6	22.5	27.5	64.3	91.4	24.8	26.2

Table 4.13: Top performers of all 42 categories on the `UMN crowd activity` dataset ranked by $F_{0.5}$ score. Configuration details, true positive and false negative rates shown. Best performers of each method in gray.

4.4.3.2 `UCSD pedestrian` dataset

The `UCSD pedestrian` dataset shows a campus park scene predominated by pedestrians moving uniformly. Outliers such as buggies driven or pedestrians running through the scene are primarily identified due to their higher velocities. In contrast to the simple `UMN crowd activity` video, the `BASELINE` approach is no longer able to keep up with the other methods and reports significantly more false alarms. See Fig. 4.23 for a time-line comparison of the detections with the ground truth at frame-level.

The other three methods identify the majority of the outliers, in fact `M-STCOG` and `M-TASOM` detect all outliers, only Adam's `HIST` approach misses the first outlier, a running pedestrian, completely. `HIST` and `M-STCOG` are able to achieve the highest ground truth congruence at around 75% $F_{0.5}$ score. However, the fact that the top performing `HIST` result is not achieved with the original setup should not go unmentioned – it is an off-line result. Using the cyclic approach originally proposed the best score obtained is 57,9 %, as can be seen with other detailed results in Tab. 4.14. Also, the `UCSD pedestrian` dataset is the only one for which the `STCOG` approach performs better than the `TASOM` approach. The best performer is an off-line variant employing an average of 14 GMM components to model the scene. In comparison, the on-line variant already creates an average 36 components to achieve a similar result. The best `TASOM` performer is a modified off-line variant using KL-divergence as a difference measure. However, the `M-TASOM` approach covers less of the ground truth and suffers from more false alarms compared to the two other methods and, as a consequence, scores lower. Detection examples are found in Fig. 4.24.

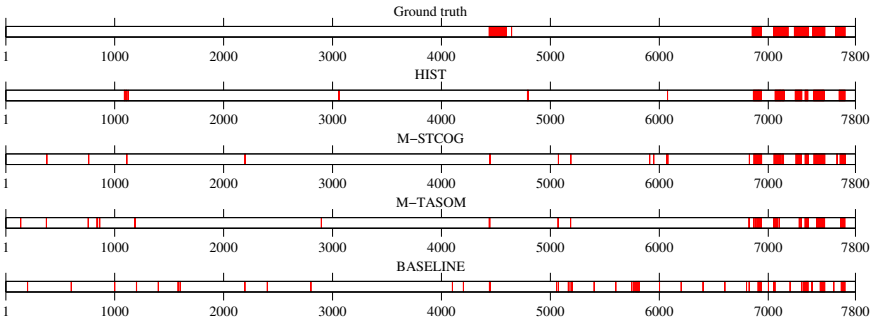
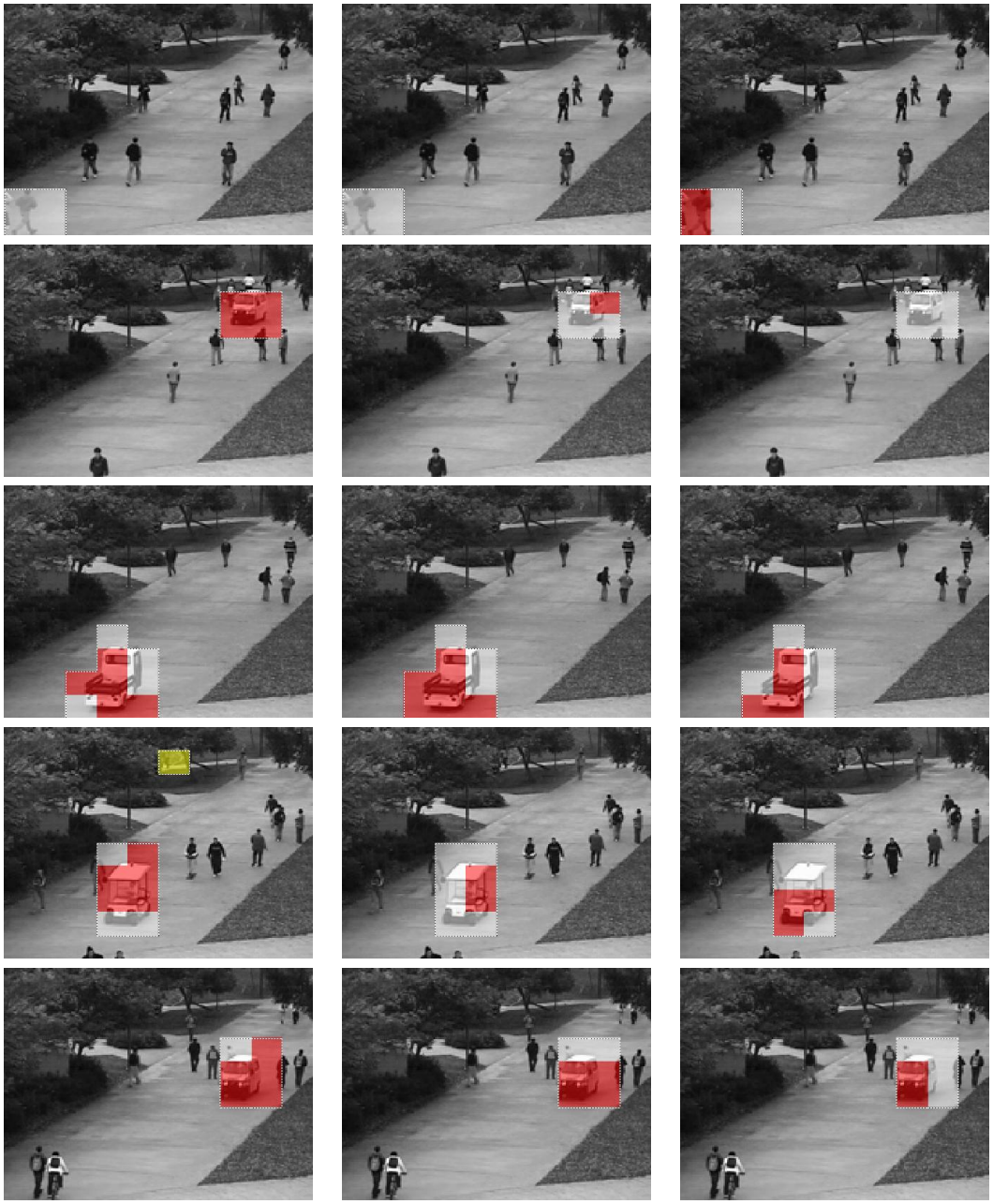


Figure 4.23: Best performers of all four methods on `UCSD pedestrian` dataset.



(a) **(b)** **(c)**

Figure 4.24: Examples of outliers detected in the `UCSD pedestrian` dataset. True positives **(red)**, false positives **(yellow)** and false negatives **(white)** for **(a)** Adam's `HIST` approach, **(b)** modified Shi's `STCOG` approach and **(c)** modified Feng's `TASOM` approach

pos.	model	on-line	Δ	parameters	Z, Y, K, N	TP_f %	TN_f %	$F_{0.5,f}$ %	TP_c %	TN_c %	$F_{0.5,c}$ %	∅ $F_{0.5}$ %
1	HIST		R		2, 5, 3, 1	55.1	99.7	83.0	37.6	100.0	69.1	76.0
2	M-STCOG			$L = 2, p_{max} = 0.5$	1, 5, 3, 0	54.8	99.7	82.4	32.6	100.0	65.7	74.1
3	M-STCOG	✓		$L = 2, p_{max} = 0.5$	1, 5, 3, 0	52.1	99.8	82.2	29.2	100.0	62.8	72.5
4	M-TASOM		KL	$K = 8, \sigma_{max} = 6, d = 5, m = 1$	1, 3, 2, 0	34.1	99.8	69.3	24.4	100.0	60.0	64.7
5	TASOM		AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 1	30.0	99.8	66.1	26.9	100.0	62.5	64.3
6	STCOG			$L = 2, p_{max} = 1E-03$	1, 5, 3, 0	42.8	99.5	73.4	21.6	100.0	53.8	63.6
7	M-TASOM		AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 1	29.1	99.9	66.4	25.0	100.0	60.8	63.6
8	HIST		R			60.0	98.6	75.6	26.3	99.9	50.9	63.3
9	TASOM		KL	$K = 6, \sigma_{max} = 2, d = 5, m = 1$	1, 5, 3, 0	31.6	99.8	67.5	23.4	100.0	58.9	63.2
10	TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 1	30.0	99.8	66.1	25.5	100.0	59.3	62.7
11	M-STCOG			$L = 1, p_{max} = 0.5$		42.8	99.6	74.0	24.8	100.0	51.0	62.5
12	M-TASOM		KL	$K = 8, \sigma_{max} = 6, d = 5, m = 1$		36.2	99.4	66.8	24.4	100.0	58.0	62.4
13	M-TASOM	✓	KL	$K = 8, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 1	30.6	99.8	66.4	25.9	100.0	58.2	62.3
14	M-TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 1	29.1	99.9	66.4	23.8	100.0	57.8	62.1
15	M-TASOM	✓	KL	$K = 6, \sigma_{max} = 2, d = 5, m = 1$		34.9	99.5	67.2	23.0	100.0	56.3	61.7
16	TASOM		KL	$K = 4, \sigma_{max} = 2, d = 5, m = 1$		33.7	99.5	66.1	22.7	100.0	56.8	61.5
17	M-STCOG	✓		$L = 2, p_{max} = 0.5$		56.1	98.5	72.9	32.3	99.9	49.9	61.4
18	TASOM	✓	KL	$K = 4, \sigma_{max} = 2, d = 5, m = 1$		33.6	99.5	65.8	22.2	100.0	55.5	60.7
19	TASOM	✓	KL	$K = 4, \sigma_{max} = 2, d = 5, m = 1$	1, 1, 1, 0	33.6	99.5	65.8	22.2	100.0	55.5	60.7
20	M-TASOM		AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$		34.4	99.0	62.2	23.4	100.0	56.6	59.4
21	M-TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$		34.4	99.0	62.2	23.3	100.0	56.3	59.2
22	TASOM		AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$		36.2	98.5	59.4	25.9	100.0	58.3	58.8
23	TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 5, m = 1$		36.2	98.5	59.4	25.5	100.0	57.6	58.5
24	HIST	✓	R		3, 5, 3, 2	35.4	99.3	65.2	24.6	100.0	50.6	57.9
25	STCOG			$L = 1, p_{max} = 1E-03$		45.9	98.9	70.0	17.8	100.0	42.5	56.2
26	BASELINE				1, 2, 1, 0	22.1	99.7	54.6	15.6	100.0	46.2	50.4
27	BASELINE					22.1	99.7	54.6	13.6	100.0	42.8	48.7
28	STCOG	✓		$L = 3, p_{max} = 1$	1, 5, 4, 1	22.8	98.7	46.4	13.1	100.0	33.7	40.0
29	STCOG	✓		$L = 1, p_{max} = 1$		17.9	98.9	41.6	10.6	100.0	29.7	35.7
30	TASOM		A	$K = 6, \sigma_{max} = 4, d = 3, m = 1$	3, 5, 4, 2	12.5	99.0	32.7	10.6	100.0	29.2	31.0
31	TASOM	✓	A	$K = 6, \sigma_{max} = 4, d = 3, m = 1$	3, 5, 4, 2	12.5	99.0	32.7	10.3	100.0	28.5	30.6
32	M-TASOM		A	$K = 4, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 5	7.5	99.9	28.0	9.7	100.0	31.2	29.6
33	M-TASOM	✓	A	$K = 4, \sigma_{max} = 2, d = 5, m = 1$	1, 2, 1, 5	7.5	99.9	28.0	9.5	100.0	30.5	29.3
34	HIST	✓	R			69.9	81.3	30.6	23.9	99.7	23.1	26.9
35	HIST		A		2, 10, 8, 5	16.2	97.0	28.4	13.7	99.8	18.4	23.4
36	HIST	✓	A		1, 10, 9, 5	11.8	98.8	29.9	7.7	99.9	16.5	23.2
37	M-TASOM		A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		19.9	91.8	19.4	9.2	99.9	18.8	19.1
38	TASOM		A	$K = 8, \sigma_{max} = 8, d = 5, m = 1$		17.1	92.8	18.5	9.3	99.9	19.3	18.9
39	M-TASOM	✓	A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		19.7	91.8	19.2	9.1	99.9	18.6	18.9
40	TASOM	✓	A	$K = 8, \sigma_{max} = 8, d = 5, m = 1$		16.6	92.8	18.0	9.1	99.9	18.8	18.4
41	HIST		A			46.8	73.4	17.1	8.0	99.5	6.8	11.9
42	HIST	✓	A			61.4	66.1	17.8	5.3	99.4	3.8	10.8

Table 4.14: Top performers of all 42 categories on the `UCSD pedestrian` dataset ranked by $F_{0.5}$ score. Configuration details, true positive and false negative rates shown. Best performers of each method in gray.

4.4.3.3 `Roadmarket` dataset

The `Roadmarket` dataset shows a street junction where outliers consist of vehicles taking unexpected paths, i.e. turning around. Reflections due to the wet environment, camera movement and the varying frame rate make it a challenging dataset.

Surprisingly, the best performers on the `Roadmarket` dataset are on-line variants of `M-TASOM` and `HIST` approaches, for a time-line comparison of detection results with the ground truth at frame-level see Fig. 4.25.

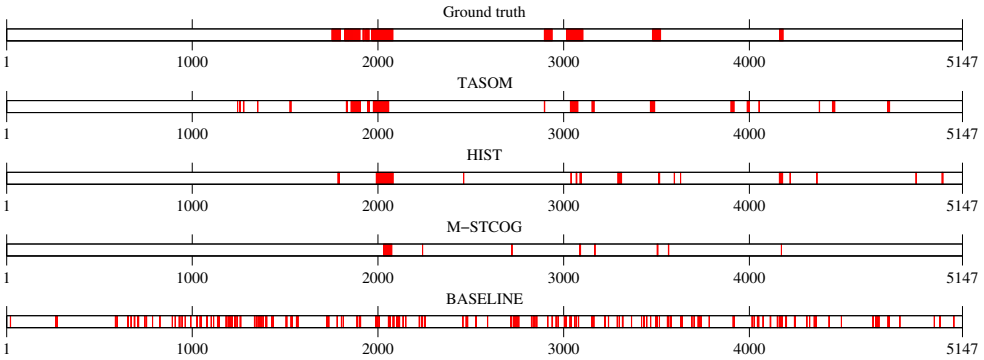


Figure 4.25: Best performers of all four methods on the `Roadmarket` dataset.

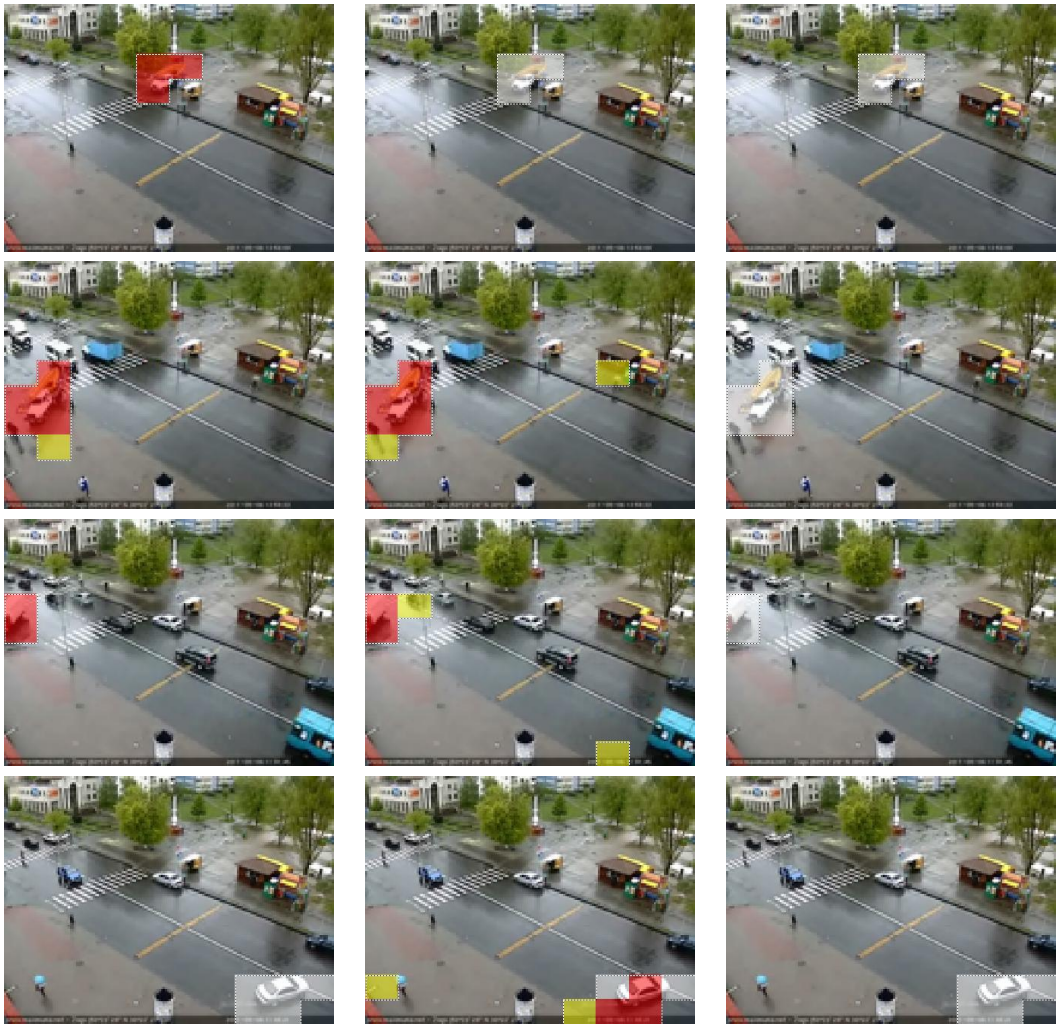
This outcome is particularly interesting given the fact that the off-line training set was chosen to include almost all parts of the dataset showing normal behavior, see Fig. 4.10. However, even the on-line training data used to train the models before applying them to outlier detection is quite large. Since the performance differences between on-line and off-line variants are not particularly large a decisive reason as to what causes these differences cannot be given.

The false alarms reported by the `TASOM` approach are mostly caused by objects that do not move along the main axis of movement, an expected behavior that demonstrates one of the main weaknesses of the approach: its inability to model normal local behavior. The false alarms of the `HIST` approach, on the other hand, are caused by an aspect which could not be observed with other datasets: it is sensitive to sudden motion changes caused by camera movement and time lapses. Nevertheless, it is the only method detecting every outlier even if the congruence with the ground truth is not particularly large.

The best performance of the STCOG method is delivered by a modified off-line variant with approximately 17 average components per GMM. The reported outliers are very sparse, but accurate, and only one outlier is missing. However, a very low probability threshold is necessary to obtain useful results indicating that the model is on the verge of its expressive power.

The BASELINE method is obviously overwhelmed by the task and is consequently dominated by false alarms.

Examples of outliers detected are displayed in Fig. 4.26, detailed results for all 42 categories can be found in Tab. 4.15.



(a) **(b)** **(c)**

Figure 4.26: Examples of outliers detected in the `UCSD pedestrian` dataset. True positives **(red)**, false positives **(yellow)** and false negatives **(white)** for **(a)** modified Feng's `TASOM` approach, **(b)** Adam's `HIST` approach and **(c)** modified Shi's `STCOG` approach. `M-TASOM` detects outliers in congruence with the ground truth, but some are missed. `HIST` shows good detection rates but suffers from false alarms while `M-STCOG` detects outlier sporadically and with little congruence.

pos.	model	on-line	Δ	parameters	Z, Y, K, N	TP_f %	TN_f %	$F_{0.5,f}$ %	TP_c %	TN_c %	$F_{0.5,c}$ %	∅ $F_{0.5}$ %
1	M-TASOM	✓	A	$K=4, \sigma_{max}=4, d=3, m=1$	1, 10, 7, 0	41.2	98.1	59.6	24.6	100.0	49.9	54.7
2	TASOM	✓	A	$K=4, \sigma_{max}=4, d=3, m=2$	2, 10, 9, 0	40.1	98.4	61.0	20.4	100.0	45.8	53.4
3	HIST	✓	A		4, 10, 9, 3	27.5	99.5	59.3	34.6	99.9	47.0	53.2
4	HIST		A		1, 10, 9, 1	27.5	99.4	58.2	19.4	100.0	45.3	51.7
5	M-TASOM		A	$K=4, \sigma_{max}=4, d=5, m=1$	1, 10, 8, 0	33.8	98.6	57.2	18.4	100.0	44.5	50.8
6	TASOM		A	$K=8, \sigma_{max}=8, d=1, m=1$	1, 10, 8, 0	35.8	98.2	56.3	18.4	100.0	44.1	50.2
7	M-TASOM		AR	$K=8, \sigma_{max}=2, d=5, m=1$	1, 10, 7, 0	29.5	99.3	60.0	13.1	100.0	39.7	49.8
8	TASOM		AR	$K=8, \sigma_{max}=4, d=5, m=1$	1, 10, 7, 0	29.1	99.3	58.6	13.4	100.0	39.8	49.2
9	TASOM	✓	AR	$K=8, \sigma_{max}=4, d=1, m=1$	1, 10, 8, 0	25.0	99.7	58.8	11.1	100.0	36.7	47.7
10	M-TASOM	✓	AR	$K=4, \sigma_{max}=4, d=5, m=1$	1, 10, 8, 0	31.3	98.6	54.8	14.4	100.0	38.0	46.4
11	M-TASOM		KL	$K=8, \sigma_{max}=4, d=5, m=1$	1, 10, 7, 1	18.2	99.2	44.0	14.4	100.0	37.6	40.8
12	TASOM		KL	$K=6, \sigma_{max}=2, d=5, m=1$	1, 10, 7, 1	18.5	99.0	42.4	14.9	100.0	37.2	39.8
13	HIST		R		1, 10, 9, 4	16.9	99.1	41.3	21.4	99.9	35.5	38.4
14	M-TASOM	✓	A	$K=4, \sigma_{max}=2, d=3, m=1$		41.9	94.6	42.1	19.3	99.9	33.8	37.9
15	M-TASOM		A	$K=4, \sigma_{max}=2, d=5, m=1$		33.8	96.7	45.0	13.4	100.0	30.5	37.7
16	TASOM	✓	KL	$K=8, \sigma_{max}=2, d=5, m=1$	3, 10, 8, 0	23.0	98.4	44.1	11.1	100.0	30.5	37.3
17	TASOM		A	$K=4, \sigma_{max}=2, d=3, m=1$		36.7	95.7	43.0	15.1	99.9	31.4	37.2
18	TASOM	✓	A	$K=8, \sigma_{max}=4, d=5, m=1$		42.8	94.3	41.7	17.7	99.9	31.2	36.4
19	M-TASOM	✓	KL	$K=6, \sigma_{max}=4, d=5, m=1$	4, 10, 8, 2	11.9	99.7	37.2	11.7	100.0	34.6	35.9
20	M-STCOG			$L=4, p_{max}=1E-03$	1, 10, 7, 0	12.6	99.9	40.2	8.4	100.0	30.9	35.5
21	M-STCOG			$L=3, p_{max}=1E-03$		16.4	98.9	38.5	8.9	100.0	27.7	33.1
22	M-TASOM		AR	$K=8, \sigma_{max}=4, d=5, m=1$		33.6	95.3	38.9	11.6	99.9	23.6	31.2
23	STCOG			$L=3, p_{max}=1E-03$	1, 10, 7, 0	19.6	97.5	34.6	11.0	100.0	27.5	31.0
24	HIST	✓	R		1, 10, 9, 3	16.2	98.6	36.3	13.3	99.9	24.0	30.1
25	M-TASOM	✓	AR	$K=6, \sigma_{max}=2, d=5, m=1$		38.7	92.0	32.6	16.9	99.9	24.4	28.5
26	TASOM	✓	AR	$K=8, \sigma_{max}=6, d=5, m=1$		39.2	90.7	30.2	19.1	99.9	24.9	27.5
27	STCOG			$L=4, p_{max}=1E-02$		21.4	96.4	31.8	10.2	99.9	22.9	27.3
28	TASOM		AR	$K=8, \sigma_{max}=2, d=5, m=1$		30.4	94.4	33.0	11.5	99.9	21.1	27.1
29	M-TASOM	✓	KL	$K=8, \sigma_{max}=4, d=5, m=1$		56.1	76.5	21.2	21.5	99.6	14.7	18.0
30	HIST	✓	R			34.5	90.4	26.7	7.3	99.8	7.4	17.1
31	HIST		A			26.6	90.9	22.5	7.4	99.9	11.5	17.0
32	M-TASOM		KL	$K=8, \sigma_{max}=2, d=5, m=1$		45.3	79.7	19.8	15.9	99.7	12.8	16.3
33	TASOM	✓	KL	$K=8, \sigma_{max}=2, d=5, m=1$		66.0	66.6	18.5	27.6	99.4	13.2	15.9
34	M-STCOG	✓		$L=5, p_{max}=0.5$	5, 10, 9, 3	19.1	93.4	21.0	15.5	99.6	10.4	15.7
35	TASOM		KL	$K=8, \sigma_{max}=2, d=5, m=1$		56.5	71.2	18.3	23.4	99.5	12.9	15.6
36	HIST	✓	A			84.5	65.4	22.2	30.9	98.9	8.3	15.2
37	HIST		R			4.5	99.9	18.2	1.9	100.0	8.3	13.3
38	STCOG	✓		$L=1, p_{max}=1E-03$	5, 10, 9, 5	48.0	72.7	16.6	22.5	97.3	2.7	9.6
39	M-STCOG	✓		$L=5, p_{max}=0.1$		58.1	61.8	14.9	13.6	99.0	4.3	9.6
40	BASELINE				1, 2, 1, 0	9.5	95.1	13.7	1.6	99.9	4.0	8.9
41	BASELINE					9.5	95.1	13.7	1.6	99.9	4.0	8.9
42	STCOG	✓		$L=4, p_{max}=1$		98.0	16.8	12.2	31.3	92.8	1.4	6.8

Table 4.15: Top performers of all 42 categories on the `Roadmarket` dataset ranked by $F_{0.5}$ score. Configuration details, true positive and false negative rates shown. Best performers of each method in gray.

4.4.3.4 `Junction` dataset

The `Junction` dataset observes a traffic junction regulated by traffic lights, i.e., there are phases present in the scene. The scene is modeled well by the `M-TASOM` and `HIST` approaches, although the smaller outliers representing pedestrians walking or bikers driving on unexpected paths are not recognized for the most part. This suggests that the chosen cell size is too small to satisfactorily model outliers of this size. Apart from the missing outliers, the `M-TASOM` delivers very good congruence with the ground truth and does not signal any false alarms. However, this performance is only achieved by modeling multiple `TASOM` models and using the angular difference measure. The originally proposed method falls back broadly by achieving only 12.3% compared to 66.7% $F_{0.5}$ score. See Fig. 4.27 for a time-line comparison of the results with the ground truth at frame-level.

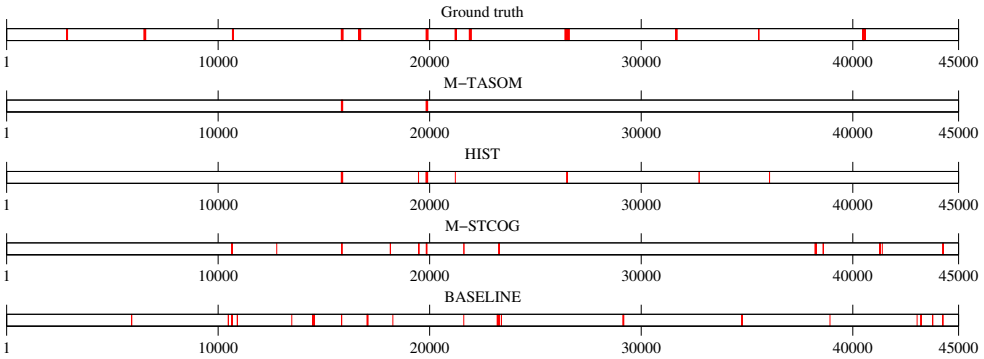


Figure 4.27: Best performers of all four methods on the `Junction` dataset.

The histogram-based `HIST` method detects at least one pedestrian outlier, but suffers from multiple false alarms mostly triggered by objects not seen during training, i.e., double-decker buses. As with the `UCSD pedestrian` dataset, this performance was achieved with the off-line variant of the method, which scores 19.7% compared to 65.8% of the modified method. In this case, the significant deterioration is caused by the short buffer used for the cyclic on-line approach and the accompanying fact that due to varying traffic density scene normality is expressed only insufficiently.

Finally, the `M-STCOG` approach is not able to robustly distinguish normal and abnormal events and suffers from false alarms while the `BASELINE` method delivers no useful result at all.

See Fig. 4.28 for examples of detected outliers for the M-TASOM, HIST and M-STCOG approaches as well as Tab. 4.16 for detailed results in all 42 categories.

(a) **(b)** **(c)**

Figure 4.28: Examples of outliers detected in the Junction dataset. True positives **(red)**, false positives **(yellow)** and false negatives **(white)** for **(a)** Feng's M-TASOM approach, **(b)** Adam's HIST approach and **(c)** Shi's M-STCOG approach.
Small outliers like pedestrians and bikers are missed almost completely by all methods. Two prominent outliers, i.e., ambulances driving the wrong way, are detected in congruence with the ground truth by M-TASOM and HIST, while Shi's M-STCOG misses one and achieves poor congruence on the other.

pos.	model	on-line	Δ	parameters	Z, Y, K, N	TP_f %	TN_f %	$F_{0.5,f}$ %	TP_c %	TN_c %	$F_{0.5,c}$ %	∅ $F_{0.5}$ %
1	M-TASOM		A	$K = 6, \sigma_{max} = 4, d = 3, m = 2$	1, 10, 7, 1	22.2	100.0	58.8	43.3	100.0	74.6	66.7
2	HIST		A		2, 3, 2, 1	23.0	100.0	58.9	40.3	100.0	72.6	65.8
3	M-TASOM	✓	A	$K = 6, \sigma_{max} = 4, d = 3, m = 2$	1, 10, 7, 1	22.2	100.0	58.8	40.9	100.0	70.5	64.6
4	TASOM	✓	A	$K = 6, \sigma_{max} = 2, d = 3, m = 1$		19.9	99.8	44.2	30.3	100.0	58.0	51.1
5	TASOM		A	$K = 6, \sigma_{max} = 2, d = 3, m = 1$		19.6	99.8	43.5	30.4	100.0	58.2	50.9
6	M-TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 3, m = 2$	1, 10, 9, 1	17.4	99.9	46.5	24.9	100.0	52.8	49.6
7	M-TASOM	✓	A	$K = 6, \sigma_{max} = 6, d = 5, m = 1$		19.3	99.9	43.9	27.2	100.0	54.5	49.2
8	M-TASOM		A	$K = 8, \sigma_{max} = 8, d = 3, m = 1$		19.6	99.8	43.5	26.9	100.0	54.1	48.8
9	HIST		A			30.5	99.2	33.8	40.4	100.0	49.4	41.6
10	TASOM		A	$K = 8, \sigma_{max} = 6, d = 3, m = 2$	3, 10, 9, 1	9.5	99.9	28.3	13.9	100.0	35.9	32.1
11	M-TASOM	✓	AR	$K = 4, \sigma_{max} = 2, d = 5, m = 1$		15.3	99.7	32.1	11.2	100.0	27.8	30.0
12	TASOM	✓	A	$K = 6, \sigma_{max} = 2, d = 5, m = 1$	2, 10, 7, 0	9.0	99.9	29.6	8.6	100.0	29.7	29.6
13	M-STCOG			$L = 5, p_{max} = 0.1$	1, 10, 7, 0	10.1	99.8	27.6	8.3	100.0	26.4	27.0
14	TASOM	✓	AR	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		19.6	98.9	19.9	25.2	100.0	29.5	24.7
15	TASOM		AR	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		19.3	98.9	19.6	25.2	100.0	29.5	24.5
16	M-TASOM		KL	$K = 4, \sigma_{max} = 2, d = 1, m = 1$	1, 10, 7, 0	7.4	100.0	26.1	6.0	100.0	22.9	24.5
17	M-TASOM	✓	KL	$K = 4, \sigma_{max} = 2, d = 1, m = 1$	1, 10, 7, 0	7.4	100.0	26.1	6.0	100.0	22.9	24.5
18	M-STCOG			$L = 5, p_{max} = 0.1$		12.1	99.6	21.8	10.0	100.0	22.5	22.2
19	HIST	✓	A		1, 10, 8, 2	5.9	99.8	17.2	11.3	100.0	22.3	19.7
20	M-TASOM		KL	$K = 4, \sigma_{max} = 4, d = 3, m = 1$		8.0	99.7	17.5	6.5	100.0	18.1	17.8
21	M-TASOM	✓	KL	$K = 4, \sigma_{max} = 4, d = 3, m = 1$		8.0	99.7	17.5	6.5	100.0	18.1	17.8
22	TASOM		KL	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		7.1	99.4	11.7	5.8	100.0	13.0	12.3
23	TASOM	✓	KL	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		7.1	99.4	11.7	5.8	100.0	13.0	12.3
24	M-TASOM		AR	$K = 4, \sigma_{max} = 2, d = 1, m = 2$	1, 10, 9, 0	11.4	98.5	10.0	6.3	100.0	8.1	9.0
25	TASOM		AR	$K = 4, \sigma_{max} = 2, d = 1, m = 1$	1, 10, 9, 1	4.0	99.5	7.9	4.4	100.0	7.7	7.8
26	TASOM	✓	AR	$K = 4, \sigma_{max} = 2, d = 1, m = 1$	1, 10, 9, 1	4.0	99.5	7.9	4.4	100.0	7.7	7.8
27	M-STCOG	✓		$L = 1, p_{max} = 0.5$	1, 10, 7, 0	5.0	99.4	8.4	3.8	100.0	6.3	7.3
28	TASOM		KL	$K = 4, \sigma_{max} = 4, d = 5, m = 2$	1, 10, 8, 1	3.1	99.6	6.9	3.3	100.0	6.7	6.8
29	TASOM	✓	KL	$K = 4, \sigma_{max} = 4, d = 5, m = 2$	1, 10, 8, 1	3.1	99.6	6.9	3.3	100.0	6.7	6.8
30	M-STCOG	✓		$L = 1, p_{max} = 0.5$		7.7	98.5	7.0	5.4	100.0	5.0	6.0
31	HIST	✓	A			9.5	98.4	8.0	3.1	100.0	3.6	5.8
32	STCOG			$L = 3, p_{max} = 5E-02$	1, 10, 7, 0	9.0	97.7	5.7	4.2	100.0	2.5	4.1
33	STCOG			$L = 3, p_{max} = 5E-02$		12.4	96.3	5.2	5.3	99.9	2.2	3.7
34	BASELINE				1, 5, 4, 0	1.3	99.7	3.5	0.7	100.0	2.4	2.9
35	M-TASOM		AR	$K = 4, \sigma_{max} = 2, d = 1, m = 2$		21.1	86.7	2.6	12.9	99.8	2.2	2.4
36	BASELINE					1.9	99.1	2.7	1.0	100.0	1.9	2.3
37	HIST	✓	R		4, 2, 1, 5	7.1	97.0	3.6	2.9	99.7	0.3	2.0
38	STCOG	✓		$L = 1, p_{max} = 1$	2, 10, 9, 0	31.2	81.9	2.9	14.0	99.2	0.6	1.7
39	HIST		R		5, 10, 9, 5	12.9	92.7	2.9	4.1	99.3	0.2	1.5
40	STCOG	✓		$L = 1, p_{max} = 1$		46.3	69.2	2.5	21.1	98.6	0.5	1.5
41	HIST	✓	R			16.6	88.6	2.4	1.4	99.7	0.2	1.3
42	HIST		R			94.2	21.4	2.1	78.9	93.3	0.4	1.2

Table 4.16: Top performers of all 42 categories on the `Junction` dataset ranked by $F_{0.5}$ score. Configuration details, true positive and false negative rates shown. Best performers of each method in gray.

4.4.3.5 `Exit platform` dataset

The `Exit platform` dataset is the longest and most challenging dataset of this evaluation. Most of its outliers consist of people walking in the wrong direction at a platform shown at the left of the scene thereby passing the turnstiles. The path of a person walking on this platform is a result of two components: Primarily, a vertical component from moving up and down the stairs, secondarily, a diagonal component moving back and forth on the platform, containing the outlier information. Since people are moving in both directions at the same time, these motion components are hard to separate. Other outliers of the datasets are more isolated and therefore easier to detect.

Again, the `HIST` and `TASOM` approaches outperform the other approaches by far, but nevertheless none of the two can solve the dataset satisfactorily. The main reason for the difficulties posed by the dataset, as mentioned above, is the indistinguishability of motion patterns. Another reason is the rareness of normal events in the scene, i.e., too little pedestrians walking the platforms. This is a general problem for all approaches, but especially difficult for the `HIST` approach, which cannot keep up a model of normality despite using a buffer of 15.000 frames length. Shi's `STCOG` approach suffers from too many false positives to be useful, whereas the `BASELINE` approach performers quite well given its simplicity, but suffers from the same problems as the other approaches.

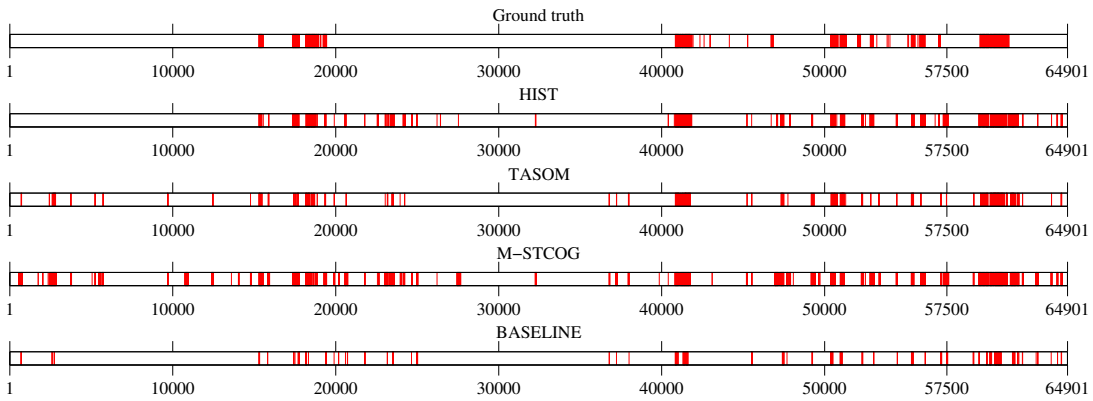
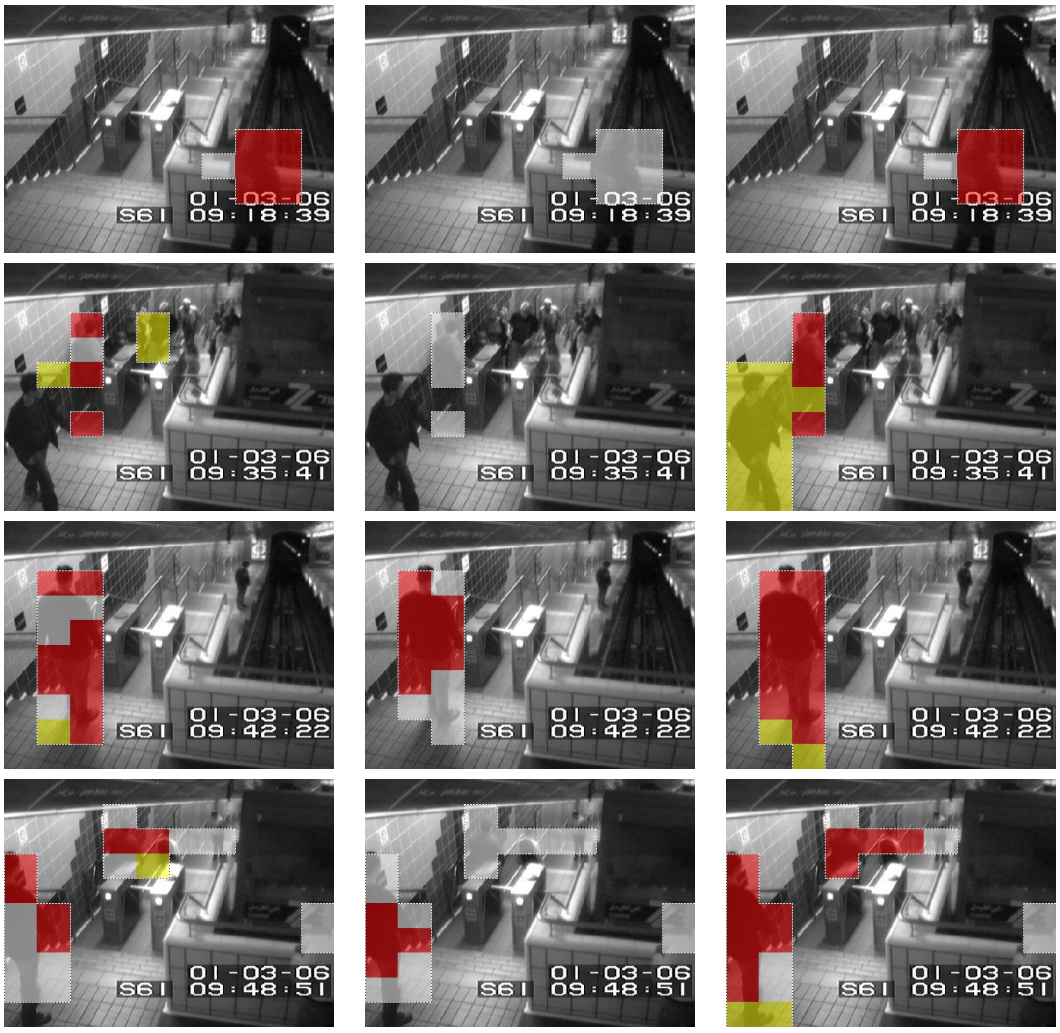


Figure 4.29: Best performers of all four methods on the `Exit platform` dataset.

A time-line comparison of the results with the ground truth at frame-level is displayed in Fig. 4.29 and detection examples are shown in Fig. 4.30. Also, see Tab. 4.17 for detailed results of all 42 categories.



(a) **(b)** **(c)**

Figure 4.30: Examples of outliers detected in the `Junction` dataset. True true positives **(red)**, false positives **(yellow)** and false negatives **(white)** for **(a)** Adam's `HIST` approach, **(b)** Feng's `TASOM` approach and **(c)** the modified Shi's `STCOG` approach.
`HIST` detects most of the outliers, but, like all methods, is not able to completely separate the complex motion patterns at the platform. `TASOM` detects outliers in good congruence with the ground truth but misses more than `HIST`. `M-STCOG` detects the most outliers, but also suffers massively from false negatives.

pos.	model	on-line	Δ	parameters	Z, Y, K, N	TP_f %	TN_f %	$F_{0.5,f}$ %	TP_c %	TN_c %	$F_{0.5,c}$ %	∅ $F_{0.5}$ %
1	HIST		A		1, 3, 2, 0	63.9	98.3	71.6	33.3	99.9	58.4	65.0
2	TASOM		AR	$K = 8, \sigma_{max} = 6, d = 5, m = 2$	1, 10, 8, 0	44.4	98.7	64.0	28.4	100.0	56.2	60.1
3	M-TASOM		AR	$K = 8, \sigma_{max} = 8, d = 1, m = 2$	1, 10, 8, 1	37.8	99.2	64.7	29.3	99.9	54.3	59.5
4	TASOM	✓	A	$K = 8, \sigma_{max} = 6, d = 3, m = 1$	1, 2, 1, 1	33.3	99.4	62.3	28.8	100.0	56.7	59.5
5	M-TASOM	✓	A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$	1, 2, 1, 1	30.6	99.5	61.3	27.1	100.0	57.5	59.4
6	HIST		A			70.8	96.2	60.1	38.3	99.9	58.2	59.1
7	TASOM		A	$K = 8, \sigma_{max} = 6, d = 3, m = 1$	1, 5, 3, 0	33.6	99.3	61.8	24.3	100.0	56.2	59.0
8	M-TASOM		A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$	1, 5, 3, 0	30.6	99.5	61.5	22.9	100.0	55.8	58.7
9	M-TASOM	✓	AR	$K = 8, \sigma_{max} = 8, d = 1, m = 2$	1, 10, 8, 1	37.7	99.2	64.5	28.2	99.9	52.2	58.4
10	TASOM	✓	A	$K = 8, \sigma_{max} = 6, d = 3, m = 1$		43.6	98.4	60.6	27.3	100.0	55.8	58.2
11	M-TASOM	✓	A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		40.1	98.6	60.0	25.6	100.0	55.8	57.9
12	TASOM		A	$K = 8, \sigma_{max} = 6, d = 3, m = 1$		42.6	98.4	59.3	25.0	100.0	51.1	55.2
13	M-TASOM		A	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		39.0	98.6	58.4	23.4	100.0	51.1	54.7
14	TASOM	✓	AR	$K = 8, \sigma_{max} = 6, d = 5, m = 2$	1, 10, 7, 1	36.9	99.1	62.6	25.8	99.9	46.2	54.4
15	TASOM	✓	AR	$K = 4, \sigma_{max} = 4, d = 5, m = 1$		44.7	97.9	56.5	27.9	99.9	51.1	53.8
16	M-TASOM	✓	AR	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		45.6	97.6	55.4	27.1	99.9	48.9	52.1
17	TASOM		AR	$K = 4, \sigma_{max} = 4, d = 5, m = 1$		43.5	97.8	55.0	25.1	99.9	46.0	50.5
18	M-TASOM		AR	$K = 6, \sigma_{max} = 6, d = 3, m = 1$		44.4	97.5	54.0	24.9	99.9	44.8	49.4
19	TASOM		KL	$K = 4, \sigma_{max} = 4, d = 5, m = 1$	1, 10, 7, 3	25.1	99.0	49.4	23.8	99.9	44.3	46.8
20	M-TASOM		KL	$K = 8, \sigma_{max} = 4, d = 5, m = 1$	1, 5, 3, 3	24.8	98.9	48.3	23.4	99.9	43.0	45.7
21	TASOM	✓	KL	$K = 4, \sigma_{max} = 4, d = 5, m = 1$	3, 10, 7, 3	24.7	99.1	49.8	21.8	99.9	41.2	45.5
22	M-TASOM	✓	KL	$K = 4, \sigma_{max} = 4, d = 3, m = 1$	3, 10, 9, 3	23.2	99.2	49.0	21.4	99.9	40.3	44.6
23	HIST	✓	A		1, 5, 4, 0	78.9	93.2	50.1	53.2	99.6	35.9	43.0
24	TASOM	✓	KL	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		69.2	92.5	44.3	44.9	99.7	39.9	42.1
25	M-TASOM	✓	KL	$K = 4, \sigma_{max} = 4, d = 5, m = 1$		74.3	92.0	44.5	43.6	99.7	38.7	41.6
26	M-TASOM		KL	$K = 8, \sigma_{max} = 4, d = 5, m = 1$		64.1	93.2	44.1	32.4	99.8	38.1	41.1
27	HIST	✓	R		1, 3, 2, 0	94.5	90.9	48.5	90.1	99.1	33.3	40.9
28	TASOM		KL	$K = 8, \sigma_{max} = 2, d = 3, m = 1$		68.0	92.4	43.5	42.2	99.7	37.5	40.5
29	HIST	✓	A			59.6	94.4	46.4	20.0	99.9	30.7	38.5
30	HIST		R		1, 5, 4, 0	86.9	90.5	45.1	77.0	99.2	31.3	38.2
31	M-STCOG	✓		$L = 1, p_{max} = 1$	1, 3, 2, 0	71.9	92.0	43.6	73.2	99.2	31.2	37.4
32	M-STCOG			$L = 1, p_{max} = 1$	1, 3, 2, 0	65.5	92.5	42.7	63.0	99.3	31.1	36.9
33	HIST	✓	R			100.0	85.5	38.9	100.0	99.0	33.9	36.4
34	M-STCOG			$L = 1, p_{max} = 1$		70.2	91.0	40.3	68.1	99.3	32.0	36.1
35	M-STCOG	✓		$L = 1, p_{max} = 1$		76.8	89.9	40.2	79.0	99.2	32.0	36.1
36	STCOG			$L = 1, p_{max} = 0.5$	1, 1, 1, 1	65.8	92.3	42.0	68.6	99.2	30.1	36.0
37	STCOG	✓		$L = 1, p_{max} = 0.5$	1, 3, 2, 0	77.2	91.2	43.5	73.7	99.0	27.2	35.3
38	STCOG			$L = 1, p_{max} = 0.5$		78.2	88.5	37.8	70.6	99.2	29.8	33.8
39	STCOG	✓		$L = 1, p_{max} = 1$		56.1	93.3	40.8	43.6	99.4	26.3	33.6
40	HIST		R			100.0	82.7	34.7	100.0	98.9	31.5	33.1
41	BASELINE				1, 2, 1, 0	9.5	99.7	30.7	4.5	100.0	17.8	24.2
42	BASELINE					9.5	99.7	30.7	3.4	100.0	14.2	22.4

Table 4.17: Top performers of all 42 categories on the `Exit platform` dataset ranked by $F_{0.5}$ score. Configuration details, true positive and false negative rates shown. Best performers of each method in gray.

4.5 Summary

To sum up, two methods have proven to perform well on all five datasets: Adam's HIST and Feng's TASOM approach. However, it should be mentioned that the TASOM approach produces its good results mostly due to the modifications proposed and carried out in this work. Although the algorithmic modifications do not bring the desired success, the use of the proposed distance measures proves successful on the three more challenging datasets, where the angular-based difference measures are found to be significantly more distinctive than KL-divergence used in the original method. On the remaining datasets, both the proposed difference measures and KL-divergence provide comparable performance, but the computational complexity of KL-divergence is significantly greater. In addition, employing multiple SOM models – as proposed in this work – proves very effective on the Junction dataset, on which the original version of the method fails. This is due to the fact, that anomalous behavior in one part of the scene is normal in another part and hence, multiple models are necessary to sufficiently describe local normalcy. However, the configuration effort for the TASOM method is very high, even experienced users might find it difficult to intuitively make the right configuration decisions.

Despite its simplicity Adam's HIST method performs well on all datasets. Its greatest weakness is its inability to generalize, which has the consequence that if normal events in a scene cannot observed densely enough the cyclic model approach breaks down. Also, normal events deviating from the average event, i.e. a person taller than others, result in outliers. Advantageous are its low computational complexity and configuration effort. Experiments without application of the integration procedure show that the HIST approach relies heavily on successful filtering of volatile events, since it reports far more events than other methods. In general, Adam's integration procedure has proven to be a simple yet effective tool to filter volatile outliers, although the possible improvement tends to be less significant for more sophisticated models.

The STCOG approach works well on simple tasks, but is too sensitive to perform well for more challenging dataset. In fact, the method suffers from an inability to robustly distinguish normal events from outliers. Although the chosen feature contains motion information in a distinguishable form, the performance achieved makes it most likely that the chosen GMM approach is not able to model the

normality adequately. When updating the model in an on-line fashion, these modeling difficulties even increase.

When detecting outliers on video datasets exhibiting complex motion patterns, like the `Exit platform` dataset, doubts about the employed motion feature extraction strategy arise. Since the behavior observed on the platform in this scene cannot be successfully separated using radial or angular-based motion features, more sophisticated approaches are necessary to solve this dataset. One possibility would be to determine the main components present in the observed motion, e.g., by using k-means clustering. For, if the features, regardless of their structure, do not represent the normal and anomalous data in a separable form, every approach is doomed to fail.

Chapter 5

Conclusion and Outlook

Contents

5.1 Conclusion **93**

5.2 Outlook **95**

5.1 Conclusion

This Master's thesis deals with solutions to the problem of outlier detection in video datasets. The number of surveillance cameras deployed in our world is steadily increasing and an automated detection of unexpected behavior is desirable in order to facilitate the task of monitoring CCTV footage. Typically this is thought of as a system taking over the tedious task of filtering unexpected events that may require attention of a human operator.

Traditionally, this problem is solved based on object detection and tracking, where objects exhibiting unusual trajectories are reported as outliers. However, this methodology has been found to be inadequate since it is likely to break down whenever scenes get crowded and is prone to the creation of false targets such as shadows or clouds. Consequently, in recent years numerous methods have been proposed that circumvent the error-prone task of object tracking by directly using low-level features to model the normality of the observed scene.

While for the area of trajectory-based outlier detection various surveys exist, none are available for low-level-feature based approaches. In addition, method

experiments are often conducted on short videos only and there is neither an established set of test datasets nor an agreement as to which performance measure should be used for evaluation. Hence direct performance comparison of different methods in the literature is difficult.

Therefore, the objectives of this thesis are twofold: First, an overview of approaches proposed in the field of low-level-based outlier detection is given in Chapter 2. Second – based on a list of desirable properties – three of these methods are chosen to be implemented and their performance is analyzed on a set of five real-world video datasets presenting various representative challenges.

Two of the selected approaches are based on statistical methods, namely a histogram-based approach (`HIST`) proposed by Adam et al. [1] and a method employing multiple GMMs (`STCOG`) to model scene normality introduced by Shi et al. [95]. The third approach (`TASOM`) is proposed by Feng et al. [33] and uses a Self-Organizing Map. Alongside a detailed description of the implementation, several modifications and extensions to increase the detection performance of these methods are described in Chapter 3. Besides algorithmic modifications of the `TASOM` and the `STCOG` approach, the usage of two additional difference measures and maintenance of multiple models to describe normal local behavior are proposed for the `TASOM` approach. The integration procedure introduced with the `HIST` method is extended and applied to all three methods to filter volatile outliers.

In Chapter 4 the results of the best performing experiment configurations in different categories are compared. To that end, ground truth on all datasets is manually labeled and an evaluation of various confusion matrix-based statistics is conducted to identify a performance measure used in subsequent analysis. The most suitable measure is found to be a specific F_β score, a weighted average of precision and recall, namely $F_{0.5}$. Results are then presented on different levels. First, the original and modified methods are compared, then on-line vs. off-line performance is analyzed and finally the best performers on all five datasets are presented. In addition, results of an on-line k-means approach are given to provide insight into the performance possible with a `BASELINE` method.

Two of the three methods are found to perform well: Adam's `HIST` method and Feng's `TASOM` method. Shi's `STCOG` method performs acceptably well on the simpler datasets, but is not able to model more complex scene normality. Given its sim-

plicity the HIST approach performs remarkably well although for three out of five datasets the cyclic approach originally proposed by Adam et al. [1] is significantly outperformed by its off-line counterpart additionally tested in this work. The reason for this behavior is that the cyclic buffer strategy used cannot sufficiently generalize normal scene behavior. Nevertheless, the small number of parameters make it the most easily configurable approach.

The STCOG approach works well only on three datasets but tends to report a small part of an outlying object only. The poor performance on the longer and more complex datasets is mainly due to the model's limited expressive power. These modeling difficulties even increase when adapting the model in an on-line fashion.

Although the third approach, Feng's TASOM model, performs well on all datasets, this performance is largely due to the extension proposed in this thesis. Apart from the two simpler datasets the original method is significantly outperformed by the extended variant. Although the TASOM approach is the only one that does not forfeit its descriptive power when applied in an on-line fashion, its difficult configurability complicate its real-world application.

To uncover the strengths and weaknesses mentioned above, five datasets covering a wide range of challenges were tested: Toy examples, a scene with varying frame rate, a complex traffic junction scene and finally, an underground train station platform exhibiting complex pedestrian motion. The two top performing methods, Adam's HIST and Feng's TASOM achieve the best results using either angular or radial measures as these measures contain sufficiently distinctive power for the majority of scenes. However, one of the datasets, namely the underground train station platform, exhibits motion patterns consisting of two main components which would have to be separated to robustly detect outliers, a problem none of the methods analyzed solves satisfactorily.

5.2 Outlook

Although the tested methods perform well on several datasets, there are various possibilities for improvement. To solve surveillance tasks exhibiting behavior consisting of different motion components, like the aforementioned underground train station platform scene, one possibility would be to determine the basic motion components

present in the data, i.e., by applying k-means clustering. The use of features based on these components would enable the presented methods to solve scenes containing complex motion patterns.

Also, an established set of video datasets used in the outlier detection literature, corresponding ground truth and the usage of a pre-defined performance measure would simplify method comparison. Test datasets should include footage covering longer periods of time, i.e. 24 hours of outdoor footage, to be able to adequately demonstrate the adaptability of an approach.

Although grid-based approaches such as the methods presented in this work have proven their applicability in real-world scenarios, their automated tuning remains an open question. On the one hand, the size of the grid cells employed depends on the scene and the method applied. Moreover, perspective distortions might require different cell sizes in different parts of the scene. On the other hand, tuning of the presented methods is a tedious task, especially when there are several parameters to be configured, like for example, with Feng's `TASOM` approach. However, the most important configuration – whether to use a radial or an angular measure – is usually easy to decide. Yet, since generally no ground truth is available, automated system configuration is a future challenge. The incorporation of expert knowledge, i.e, outliers detected by human operators, would be a possibility to facilitate this task.

Finally, one last aspect should be taken into consideration, namely if besides motion other low-level information should be used. Since there are, for example, approaches using texture, future work might deal with incorporating additional features to further enhance the capabilities of outlier detection systems.

List of Figures

2.1 Basic outlier detection diagram 7
2.3 Spatial-temporal division of the video 13
2.2 Scheme of a Self-Organizing Map 14
2.4 Growing Neural Gas 17
2.5 Plate notation of PLSA 19
2.6 LDAs 22
2.7 Region LDA 24
2.8 HMM 25

3.1 Terms used in feature extraction 33
3.2 Unified motion extraction procedure 35
3.3 Pair-wise cuboid concept 38
3.4 Multiple `TASOM` models 47

4.1 Obtaining ground truth 51
4.2 Typical frame of the `UMN crowd activity` dataset 52
4.3 Anomalous frames of the `UMN crowd activity` dataset 52
4.4 Ground truth of the `UMN crowd activity` dataset 52
4.5 Typical frames of the `UCSD pedestrian` dataset. 53
4.6 Abnormal frames of the `UCSD pedestrian` dataset 54
4.7 Ground truth of the `UCSD pedestrian` dataset 54
4.8 Typical frames of the `Roadmarket` dataset. 55
4.9 Abnormal frames of the `Roadmarket` dataset 56

4.10 Ground truth of the Roadmarket dataset 56
4.11 Typical frames of the Junction dataset. 57
4.12 Abnormal frames of the Junction dataset 57
4.13 Ground truth of the Junction dataset 57
4.14 Typical frames of the Exit platform dataset 58
4.15 Abnormal frames of the Exit platform dataset 59
4.16 Ground truth of the Exit platform dataset 59
4.17 Performance measure comparison on the UMN crowd activity dataset 63
4.18 Performance measure comparison on the UCSD pedestrian dataset . . 64
4.19 Performance measure comparison on the Exit platform dataset . . . 65
4.20 Multiple TASOM models (revisited) 72
4.21 Best performers on the UMN crowd activity dataset 75
4.22 Examples of outliers detected in the UMN crowd activity dataset . . . 76
4.23 Best performers on the UCSD pedestrian dataset 78
4.24 Examples of outliers detected in UCSD pedestrian dataset 79
4.25 Best performers on the Roadmarket dataset 81
4.26 Examples of outliers detected in UCSD pedestrian dataset 83
4.27 Best performers on the Junction dataset 85
4.28 Examples of outliers detected in Junction dataset 86
4.29 Best performers on the Exit platform dataset 88
4.30 Examples of outliers detected in Junction dataset 89

Bibliography

[1] Adam, A., Rivlin, E., Shimshoni, I., and Reinitz, D. (2008). Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:555–560.

[2] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In *Proc. Int'l Conf. on Database Theory*, pages 420–434.

[3] Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. In *Proc. ACM SIGMOID Int'l Conf. on Management of Data*, pages 37–46.

[4] Alphaydin, E. (2004). *Introduction to Machine Learning*. MIT Press.

[5] Altman, D. G. and Bland, J. M. (1994). Statistics Notes: Diagnostic tests 2: predictive values. *British Medical Journal*, 309(6947):102+.

[6] Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics*, 16(5):412–424.

[7] Basharat, A., Gritai, A., and Shah, M. (2008). Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection. In *Proc. Conf. on Computer Vision and Pattern Recognition*.

[8] Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). The Infinite Hidden Markov Model. In *Advances in Neural Information Processing Systems*, pages 577–584.

[9] Ben-Gal, I. (2007). *Bayesian Networks*. John Wiley and Sons.

[10] Benezeth, Y., Jodoin, P.-M., Saligrama, V., and Rosenberger, C. (2009). Abnormal Events Detection Based on Spatio-Temporal Co-occurences. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 2458–2465.

[11] Berkhin, P. (2002). Survey Of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA.

[12] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When Is "Nearest Neighbor" Meaningful? In *Proc. Int'l Conf. on Database Theory*.

[13] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[14] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[15] Breitenstein, M., Grabner, H., and Van Gool, L. (2009). Hunting Nessie – Real-Time Abnormality Detection from Webcams. In *IEEE Workshops on Visual Surveilance*, pages 1243–1250.

[16] Bremond, F. and Pusiol, G. (2008). Commentary Paper on "Learning and Classification of Trajectories in Dynamic Scenes: A General Framework for Live Video Analysis". In *Proc. IEEE Int'l Conf. on Advanced Video and Signal Based Surveillance*, pages 162–163.

[17] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In *Proc. ACM SIGMOID Int'l Conf. on Management of Data*, pages 93–104.

[18] Buntine, W. L. (1994). Operations for Learning with Graphical Models. *Journal of Artificial Intelligence Research*, 2:159–225.

[19] Carpenter, G. A. and Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer*, 21:77–88.

[20] Celik, H., Hanjalic, A., and Hendriks, E. (2009a). Towards Unsupervised Learning for Automatic Multi-class Object Detection in Surveillance Videos. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 3521–3524.

[21] Celik, H., Hanjalic, A., Hendriks, E., and Boughorbel, S. (2008). Online Training of Object Detectors from Unlabeled Surveillance Video. In *Proc. IEEE Workshop on Online Learning for Classification in Conjunction with Computer Vision and Pattern Recognition*.

[22] Celik, H., Hanjalic, A., and Hendriks, E. A. (2009b). Unsupervised and simultaneous training of multiple object detectors from unlabeled surveillance video. *Computer Vision and Image Understanding*, 113:1076–1094.

[23] Chambolle, A. and Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.

[24] Chan, A. B. and Vasconcelos, N. (2008). Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(5):909–926.

[25] Chandola, V., Banerjee, A., and Kumar, V. (2007). Outlier detection: A survey. *ACM Computing Surveys*.

[26] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Computing Surveys*, 41:15:1–15:58.

[27] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

[28] De Castro, E. and Morandi, C. (1987). Registration of Translated and Rotated Images Using Finite Fourier Transforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 700–703.

[29] Dee, H. and Velastin, S. (2008). How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19:329–343.

[30] Diehl, C. and Hampshire, J.B., I. (2002). Real-time Object Classification and Novelty Detection for Collaborative Video Surveillance. In *Proc. Int'l Joint Conf. on Neural Networks*, pages 2620–2625.

[31] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, Inc.

[32] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874.

[33] Feng, J., Zhang, C., and Hao, P. (2010). Online Learning with Self-Organizing Maps for Anomaly Detection in Crowd Scenes. In *Proc. Int'l Conf. on Pattern Recognition*, pages 3599–3602.

[34] Fritzke, B. (1995). A Growing Neural Gas Network Learns Topologies. In *Advances in Neural Information Processing Systems*, pages 625–632.

[35] Fritzke, B. (1997a). A Self-Organizing Network that Can Follow Non-stationary Distributions. In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 613–618.

[36] Fritzke, B. (1997b). Some competitive learning methods. Technical report, Institut für Neuroinformatik, Ruhr-Universitat Bochum.

[37] Frome, A., Singer, Y., Sha, F., and Malik, J. (2007). Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In *Proc. Int'l Conf. on Computer Vision*.

[38] Fusier, F., Valentin, V., Bremond, F., Thonnat, M., Borg, M., Thirde, D., and Ferryman, J. (2007). Video understanding for complex activity recognition. *Machine Vision and Applications*, 18:167–188.

[39] Geraci, A. (1991). *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. IEEE Press.

[40] Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994). A Language and Program for Complex Bayesian Modelling. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(1):169–177.

[41] Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proc. of the National Academy of Sciences of the United States of America*, 101:5228–5235.

[42] Haines, T. S. F. and Xiang, T. (2010). Video Topic Modelling with Behavioural Segmentation. In *Proc. ACM Workshop Multimodal Pervasive Video Analysis*, pages 53–58.

[43] Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2nd edition.

[44] Han, M., Xu, W., Tao, H., and Gong, Y. (2007). Multi-object trajectory tracking. *Machine Vision and Applications*, 18:221–232.

[45] He, X. and Niyogi, P. (2003). Locality preserving projections. In *Advances in Neural Information Processing Systems*.

[46] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley.

[47] Hempel, L. and Töpfer, E. (2004). *On the Threshold to Urban Panopticon? Analysing the Employment of CCTV in European Cities and Assessing its Social and Political Impacts*. CCTV in Europe. Final Report.

[48] Hendel, A., Weinshall, D., and Peleg, S. (2010). Identifying Surprising Events in Videos Using Bayesian Topic Models. In *Proc. Asian Conf. on Computer Vision*, pages 448–459.

[49] Hodge, V. and Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22:85–126.

[50] Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. In *Proc. ACM Conf. on Research and Development in Information Retrieval*, pages 50–57.

[51] Hospedales, T., Gong, S., and Xiang, T. (2009). A Markov Clustering Topic Model for Mining Behaviour in Video. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 1165–1172.

[52] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31:651–666.

[53] Japkowicz, N., Myers, C., and Gluck, M. (1995). A Novelty Detection Approach to Classification. In *Proc. Joint Conf. on Artificial Intelligence*, pages 518–523.

[54] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer.

[55] Kim, J. and Grauman, K. (2009). Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *Proc. Conf. on Computer Vision and Pattern Recognition*.

[56] Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3):1–6.

[57] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

[58] Kotsiantis, S. B. and Pintelas, P. E. (2004). Recent Advances in Clustering: A Brief Survey. *WSEAS Trans. on Information Science and Applications*, 1:73–81.

[59] Kratz, L. and Nishino, K. (2009). Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 1446–1453.

[60] Kwak, S. and Byun, H. (2011). Detection of dominant flow and abnormal events in surveillance video. *Optical Engineering*, 50(2):027202.

[61] Latecki, L., Lazarevic, A., and Pokrajac, D. (2007). Outlier Detection with Kernel Density Functions. In *Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer.

[62] Li, J., Gong, S., and Xiang, T. (2008). Global Behaviour Inference using Probabilistic Latent Semantic Analysis. In *Proc. British Machine Vision Conference*.

[63] Li, J., Gong, S., and Xiang, T. (2009). Discovering Multi-Camera Behaviour Correlations for On-the-Fly Global Activity Prediction and Anomaly Detection. *IEEE Int'l Workshop on Visual Surveillance*.

[64] Liao, T. W. (2005). Clustering of time series data — a survey. *Pattern Recognition*, 38(11):1857–1874.

[65] Loy, C., Xiang, T., and Gong, S. (2011). Stream-Based Active Unusual Event Detection. In *Proc. Asian Conf. on Computer Vision*.

[66] Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly Detection in Crowded Scenes. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 1975–1981.

[67] Markou, M. and Singh, S. (2003). Novelty detection: a review — part 2: neural network based approaches. *Signal Process.*, 83:2499–2521.

[68] Martinetz, T. (1993). Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 427–434.

[69] Martinetz, T. and Schulten, K. (1991). A „Neural-Gas" Network Learns Topologies. *Artificial Neural Networks*, I:397–402.

[70] McCahill, M. and Norris, C. (2004). *On the Threshold to Urban Panopticon? Analysing the Employment of CCTV in European Cities and Assessing its Social and Political Impacts*. CCTV Systems in London. Their Structures and Practices.

[71] Meer, P. (2004). *Emerging Topics In Computer Vision*. Prentice Hall Computer.

[72] Mohemmed, A. W., Zhang, M., and Browne, W. N. (2010). Particle Swarm Optimisation for Outlier Detection. In *Proc. Conf. on Genetic and Evolutionary Computation*, pages 83–84.

[73] Morris, B. and Trivedi, M. (2008a). A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(8):1114–1127.

[74] Morris, B. and Trivedi, M. (2008b). Learning and Classification of Trajectories in Dynamic Scenes: A General Framework for Live Video Analysis. In *Proc. IEEE Int'l Conf. on Advanced Video and Signal Based Surveillance*, pages 154–161.

[75] Morris, B. and Trivedi, M. (2009). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 312–319.

[76] Myrvoll, T. A. and Soong, F. K. (2003a). On Divergence Based Clustering of Normal Distributions and Its Application to HMM Adaptation. In *Proc. European Conference on Speech Communication and Technology*.

[77] Myrvoll, T. A. and Soong, F. K. (2003b). Optimal clustering of multivariate normal distributions using divergence and its application to HMM adaptation. In *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 552–555.

[78] Nairac, A., Corbett-Clark, T., Ripley, R., Townsend, N., and Tarassenko, L. (1997). Choosing an appropriate Model for Novelty Detection. In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 117–122.

[79] Norris, V., McCahill, M., and Wood, D. (2004). Editorial: The Growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance and Society*, 2(2/3):110–135.

[80] Nowak, E. and Jurie, F. (2007). Learning Visual Similarity Measures for Comparing Never Seen Objects. In *Proc. Conf. on Computer Vision and Pattern Recognition*.

[81] Noyes, J. and Bransby, M. (2001). *People in control: Human factors in control room design*. IET Control Engineering Series. The Institution of Engineering and Technology.

[82] Papadimitriou, S., Kitawaga, H., Gibbons, P. B., and Faloutsos, C. (2002). LOCI: Fast Outlier Detection Using the Local Correlation Integral.

[83] Papadimitriou, S., Kitawaga, H., Gibbons, P. B., and Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *Proc. Int'l Conf. on Data Engineering*.

[84] Pokrajac, D., Lazarevic, A., and Latecki, L. J. (2007). Incremental Local Outlier Detection for Data Streams. In *Proc. IEEE Symposium on Computational Intelligence and Data Mining*, pages 504–515.

[85] Pruteanu-Malinici, I. and Carin, L. (2008). Infinite Hidden Markov Models for Unusual-Event Detection in Video. *IEEE Trans. on Image Processing*, (5):811–822.

[86] Remagnino, P., Velastin, S. A., Foresti, G. L., and Trivedi, M. (2007). Novel concepts and challenges for the next generation of video surveillance systems. *Mach. Vision Appl.*, 18:135–137.

[87] Rosenberg, Y. and Werman, M. (1997). Representing Local Motion as a Probability Distribution Matrix and Object Tracking. In *Proc. Int'l Workshop on Persistent Object Systems in the MONADS Architecture*, pages 392–405.

[88] Santini, S. and Jain, R. (1999). Similarity Measures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(9):871–883.

[89] Schmidhuber, J. (2009). Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 48–76. Springer.

[90] Scott-Brown, K. and Cronin, P. (2008). Detect the unexpected: a science for surveillance. *Policing: An Int'l Journal of Police Strategies & Management*, 31:395–414.

[91] Shah-Hosseini, H. and Safabakhsh, R. (2000a). Pattern Classification by the Time Adaptive Self-Organizing Map. In *Proc. IEEE Int'l Conf. on Electronics, Circuits and Systems*, volume 1, pages 495–498.

[92] Shah-Hosseini, H. and Safabakhsh, R. (2000b). TASOM: The Time Adaptive Self-Organizing Map. In *Proc. Int'l Conf. on Information Technology: Coding and Computing*, pages 422–427.

[93] Shen, F. and Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19:90–106.

[94] Shen, F. and Hasegawa, O. (2010). Self-Organizing Incremental Neural Network and Its Application. In *Proc. Int'l Conf. on Artificial Neural Networks: Part III*, pages 535–540. Springer-Verlag.

[95] Shi, Y., Gao, Y., and Wang, R. (2010). Real-Time Abnormal Event Detection in Complicated Scenes. *Int'l Conf. on Pattern Recognition*, pages 3653–3656.

[96] Singh, S. and Markou, M. (2004). An Approach to Novelty Detection Applied to the Classification of Image Regions. *IEEE Trans. on Knowl. and Data Eng.*, 16:396–407.

[97] Smith, G. J. (2004). Behind the Screens: Examining Constructions of Deviance and Informal Practices among CCTV Control Room Operators in the UK. *Surveillance & Society 2(2/3)*, pages 376–395.

[98] Soatto, S., Doretto, G., and Wu, Y. N. (2001). Dynamic Textures. In *Proc. ICCV*, pages 439–446.

[99] Stauffer, C. and Grimson, W. E. L. (2000). Learning Patterns of Activity Using Real-Time Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:747–757.

[100] Teh, Y. W. (2010). Dirichlet Processes. In *Encyclopedia of Machine Learning*. Springer.

[101] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

[102] Tziakos, I., Cavallaro, A., and Xu, L.-Q. (2010). Local Abnormality Detection in Video Using Subspace Learning. In *Proc. IEEE Int'l Conf. on Advanced Video and Signal Based Surveillance*, pages 519–525.

[103] van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth.

[104] Varadarajan, J. and Odobez, J.-M. (2009). Topic Models for Scene Analysis and Abnormality Detection. In *Proc. IEEE Int'l Conf. on Computer Vision Surveillance Workshops*, pages 1338–1345.

[105] Wallace, E. and Diffley, C. (1998). CCTV: Making It Work. *Police Scientific Development Branch (PSDB)*.

[106] Wang, X., Ma, X., and Grimson, W. (2009). Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31:539–555.

[107] Widhalm, P. and Brändle, N. (2010). Learning Major Pedestrian Flows in Crowded Scenes. In *Proc. Int'l Conf. on Pattern Recognition*, pages 4064–4067.

[108] Xiang, T. and Gong, S. (2005). Video Behaviour Profiling and Abnormality Detection without Manual Labelling. In *Proc. Int'l Conference on Computer Vision*, pages 1238–1245.

[109] Xiang, T. and Gong, S. (2008). Video Behavior Profiling for Anomaly Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(5):893–908.

[110] Xu, R. and Wunsch, D., I. (2005). Survey of Clustering Algorithms. *IEEE Trans. on Neural Networks*, 16(3):645–678.

[111] Yamasaki, K., Makibuchi, N., Shen, F., and Hasegawa, O. (2010). How to use the SOINN software: user's guide (version 1.0). In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 521–527.

[112] Yu, X., Tang, L. A., and Han, J. (2009). Filtering and Refinement: A Two-Stage Approach for Efficient and Effective Anomaly Detection. In *Proc. IEEE Int'l Conf. on Data Mining*, pages 617–626.

[113] Zach, C., Pock, T., and Bischof, H. (2007). A Duality Based Approach for Real-time TV-L1 Optical Flow. In *Proc. DAGM Symposium*, pages 214–223.

[114] Zelnik-Manor, L. and Perona, P. (2004). Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608.

[115] Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., and Xu, L.-Q. (2008). Crowd analysis: a survey. *Mach. Vision Appl.*, 19:345–357.

[116] Zhong, H., Shi, J., and Visontai, M. (2004). Detecting Unusual Activity in Video. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 819–826.

[117] Zhou, S. K. and Chellappa, R. (2004). Kullback-Leibler Distance between Two Gaussian Densities in Reproducing Kernel Hilbert Space. *IEEE Int'l Symposium on Information Theory*.

[118] Zhu, X. and Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

[119] Zhu, X., Yao, Y., Liu, Z., and Xiong, J. (2010). Activity Clustering for Online Anomaly Detection. In *Proc. Int'l Conf. on Modelling, Identification and Control*, pages 842–847.