Universidad Autónoma de Madrid

Escuela Politécnica Superior

Master's Degree in ICT Research and Innovation
Image Processing and Computer Vision Program

MASTER THESIS

ALGORITHMS FOR ANOMALY DETECTION IN VIDEO SEQUENCES THROUGH DISCRIMINATIVE MODELS

Adrián Tomé Alonso
Director: Luis Salgado Álvarez de Sotomayor

September 2016

ALGORITHMS FOR ANOMALY DETECTION IN VIDEO SEQUENCES THROUGH DISCRIMINATIVE MODELS

Author: Adrián Tomé Alonso
Director: Luis Salgado Álvarez de Sotomayor
Advisor: Jesús Bescós Cano

Escuela Politécnica Superior
Universidad Autónoma de Madrid

September 2016

This work has been partially suported by the Ministerio de Economía y Competitividad of the Spanish Government and the Fondo Europeo para el Desarrollo Regional of the European Union under the project RTC-2015-3527-1 (BEGISE)

GOBIERNO DE ESPANA
MINISTERIO DE ECONOMIA Y COMPETITIVIDAD

Unión Europea
Fondo Europeo
de Desarrollo Regional
“Una manera de hacer Europa”

Abstract

Monitoring public areas with pedestrians is a task that has to be frequently accomplished by means of security systems. Nevertheless, manual detection of these anomalies is a tough task and it is easy to lose interesting events when many areas have to be attended. This is the main reason why the automated detection of these anomalies and interesting events in general has become an important source of research in the past years, specially in the field of computer vision.

Automated anomaly detection is still an open task even though that many methods have been proposed. One of the reasons is that a successful and accurate anomaly detection algorithm strongly depends on the context and the definition of the anomalies to detect and the objects that produce them. The state of the art included in this work has been developed to make a complete study of all these aspects in detail, as well as a study of advantages and drawbacks of the main methods of the literature, helping to choose the best techniques and strategies for specific surveillance scenarios.

Since there is a great difficulty to model every anomaly, we have decided to fashion the normality by means of Gaussian mixture models, which are relatively simple methods compared to others in the literature such as [1, 2], but that have shown potential at detecting anomalies. This can be observed on the methods proposed in [3] and [4].

We have decided to work at pixel level. Thus, to feed the model, discriminative descriptors are built based on a robust optical flow method, that has become the main source of motion and textural information of the scene. This fact makes this work different to other state-of-the-art approaches that work at pixel-level, whose optical flow is not capable to give such a detailed information of the scene.

Finally, the evaluation of the final algorithm is performed exhaustively from a baseline method, whose descriptor grows depending on the best results so far on a publicly available dataset. Detection results are compared with the state-of-the-art methods, concluding that our method is at the same level of the methods proposed in the literature.

Keywords

Anomaly detection, Gaussian mixture model, normality model, video-surveillance, optical flow, global GMM.

Resumen

Vigilar zonas públicas con peatones es una tarea que ha de llevarse a cabo frecuentemente mediante el uso de sistemas de videovigilancia, y en muchas ocasiones, ser capaz de detectar anomalías en dichos escenarios es crucial para asegurar el éxito del sistema. Sin embargo, la detección manual de estas anomalías es una tarea tediosa y es muy fácil no detectar eventos de interés cuando varias áreas tienen que ser atendidas. Esta es la razón principal por la que automatizar la detección de anomalías y eventos de interés se ha convertido en una importante fuente de investigación en los últimos años, especialmente en el campo de la visión artificial.

A pesar de que multitud de métodos han sido propuestos hasta la fecha, la detección de anomalías es aún una tarea abierta. Una de las razones por las que esto ocurre es debido a que un algoritmo para detección de anomalías que sea preciso depende enormemente de la capacidad para identificar el contexto así como la definición de anomalía a detectar y los objetos que las producen. El estado del arte realizado en este trabajo incide sobre estos puntos en detalle, así como el estudio de las ventajas e incovenientes de los métodos más importantes de la literatura para tener un mejor entendimiento de cuáles son los métodos y estrategias que mejor encajan en los diferentes escenarios que se puedan dar en el contexto de la videovigilancia.

Puesto que hay una gran dificultad para modelar cada anomalía que pueda surgir, hemos llegado a la conclusión de que la mejor opción es construir un modelo de normalidad de la escena mediante modelos de mezclas de Gaussianas, ya que son métodos relativamente simples en comparación con otros de la literatura como los propuestos en [1, 2] pero que tienen potencial para la detección de anomalías como ocurre en los trabajos propuestos en [3] y [4].

Además, hemos decidido trabajar a nivel de píxel. Así, para construir los modelos, es necesario crear unos descriptores que sean discriminativos. Para este fin, se ha usado un método robusto para la obtención de flujo óptico, que se ha posicionado como la principal fuente de información de movimiento y textura de la escena. Este hecho hace que nuestro trabajo sea diferente a otros del estado del arte que trabajan a nivel de píxel, cuyos métodos para el cálculo de flujo óptico no son capaces de ofrecer información tan detallada.

Por último, el algoritmo de detección de anomalías llevado a cabo ha sido evaluado exhaustivamente en bases de datos públicas, partiendo desde un método base cuyos descriptores crecen en función de los últimos mejores resultados obtenidos, llegando finalmente al nivel de los métodos del estado del arte.

Palabras clave

Detección de anomalías, modelo de mezcla de Gaussianas, modelo de normalidad, videovigilancia, flujo óptico, GMM global.

Acknowledgements

I would like to thank to all people and institutions that has given me support to make this project. In particular, the director of the thesis, Luis, who is always encouraging me to do my best. Thanks to Noelia, who is always by my side. Thanks to my friends and classmates of the degree and IPCV program. It has been an amazing experience. Thanks to my family. Without you, this project would not have been possible. And finally, I would like to thank to UAM, UPM and the universities of the IPCV program this opportunity. From the bottom of my heart, thanks.

Contents

List of Figures

List of Tables

List of Equations

Chapter 1

Introduction

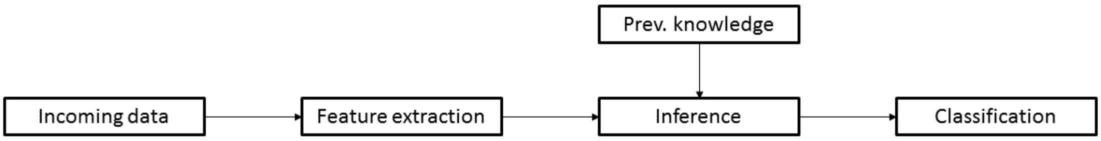


Figure 1.1: General structure of an anomaly detection algorithm

Automated video-surveillance has received remarkable attention by the research community in the past years [10]. In particular, computer vision and machine learning techniques have been employed to recognize actions, events and behaviors, track objects or just modeling a scene, in order to obtain information about the events on scene, which may be interesting for different purposes such as detection of anomalies.

The first main limitation is the lack of a universal definition of anomaly [1]. This is widely seen in some of the strategies from the literature, where the anomalies are considered as suspicious events, irregular, uncommon, unusual or abnormal behaviors [10]. In a try to give an objective definition of anomaly, authors of [6] say: *"Anomaly is a behavior, event or activity that can be considered as unusual, suspicious and/or infrequent, whose information might be known entirely, partially or even have no information at all"*. Alternatively, authors of [8,11] based their work on the definition of abnormality from the dictionary: *"something unusual, irregular and aberrant"*. As we can see, there is no consensus but what we can conclude for sure is that an anomaly occurs when the associated event differs from the concept of normality in the specific context.

After giving an approximate definition of anomaly, it is feasible to study the different methods applied to find them. In general, it is possible to infer the general structure followed in these approaches (a general overview of the panorama is given in the state-of-the-art chapter) in which the final goal is to classify the events in the scene as normal or abnormal, sometimes also locating or recognizing them. For that purpose, it is necessary to make a classification of the events happening on scene, based on features and previous information available. The general procedure is shown in Figure 1.1.

1 Motivation and goals

The purpose of this work is to create a method to detect anomalies in video-sequences taken on pedestrian scenarios. To do so, some goals have to be fulfilled:

- Complete an exhaustive analysis of those methods to detect anomalies available in the literature, organize them by context, strategy and feature.
- Focus on works dedicated to algorithms for the detection of anomalies on video-surveillance scenarios, dedicating special attention to pedestrian environments.
- Based on this, implement an algorithm that detects anomalies in security environments, preferably in C++ and using the OpenCV library, making the necessary contributions, modifications and improvements.
- Evaluate the obtained results in publicly available datasets to compare the results with the state-of-the-art techniques.

2 Structure of the document

- **Chapter 1**: Contains the introduction, motivation and goals.
- **Chapter 2**: Study and analysis of the most important methods and techniques to detect anomalies on video-sequences.
- **Chapter 3**: Architecture of the system. Feature extraction and construction of the normality model.
- **Chapter 4**: Evaluation and results of the anomaly detection method on public datasets.
- **Chapter 5**: Future work and conclusions.
- **Bibliography** and **appendices**.

Chapter 2

State of the art

Making unique classifications of the methods of the literature dedicated to anomaly detection is a hard task, as it stands out from the available reviews for anomaly detection in the last years [10, 12, 13]. It is important to remark that these reviews are focused on specific problems within the field of anomaly detection and therefore, each one gives different perspectives of the available methods, models and characteristics used. For instance, [12] includes the type of sensor used to obtain the sequences, adding other based on non-visible light such as infrared cameras or even audio signals; authors of [10] focus on abnormal human behavior while in [13] the attention is given to crowds.

In spite of these differences, it is possible to infer an underlying scheme common to all of them that will be explained in detail in the next sections. Algorithms in the literature are classified according to three aspects: the targets or objects that produce the anomalies, the level of supervision of the algorithms and the type of model used to characterize the scenarios.

1 Classification by target

The organization of the methods regarding where the information is extracted from, has not a unique classification. This comes from the fact that an anomaly can be produced by many elements on scene. Besides, anomaly detection in general is not always wanted and only anomalies produced by specific objects or events in particular have to be detected. For these reasons, it is convenient to classify the methods depending on the type of target that might produce anomalies. Authors of [12] illustrated this through a Venn diagram, whose concept has been preserved in Figure 2.1.

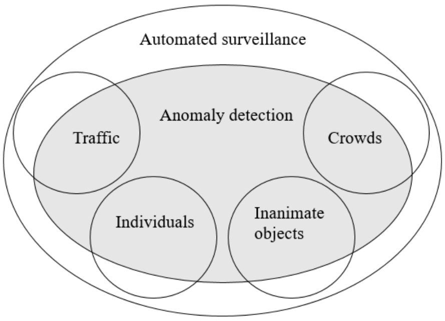


Figure 2.1: Venn diagram illustrating the targets of anomaly detection

Different targets might receive the attention of a security detection system. This is the case of objects in traffic scenarios, individuals and groups of people in public spaces or inanimate objects, not only for anomaly detection: other aspects such as action recognition, trajectory extraction, restricted area access violation or abandoned objects detection might be required.

The main consequence is that it is necessary to analyze exhaustively the context where the method will be applied when choosing an anomaly detection algorithm, sometimes producing at the end methods so specific that commonly, the application of these methods for new contexts becomes a difficult task (a normal behavior in one context may be abnormal in another [10]). Additionally, other methods might be used to different types of target [12].

Computer vision methods that aim to detect or recognize events and actions with people as target form a big branch in the field of automated surveillance. These techniques can be dedicated for instance to look after elder people with methods such as fall detection or pedestrian safety such as tailgating or lawbreaking in general. The strategies proposed in [4, 5, 14–18] are some examples of methods that can be utilized for anomaly detection for individuals and/or crowds.

Particularly remarkable are the strategies proposed in [1, 2] and [6]. The first methods have been proven successful in anomaly detection on crowded scenes by using a mixture of dynamic textures (MDT), in which as they claim, both appearance and dynamics are taken into account to model the normality and see how new samples are likely to be normal. Authors of [6] created an algorithm that learns a hierarchical codebook of dominant behaviors without supervision and in an online manner. Finally, some important methods use trajectories applicable to individuals and other targets [19–22].

1.1 Bottom-up and top-down approaches

Despite the target analyzed, it is possible to organize the approaches depending if they use a “top-down” or a “bottom-up” strategy. By using the first approach, detection and tracking of the objects (high level) is performed on the scene and from that point, the needed information for the specific algorithm is extracted (low level). On the other hand, bottom-up approaches work from pixel-level to higher level, avoiding the use of tracking methods, which could perform poorly in specific situations. Bottom-up approaches have the advantage of working by extracting inherent information such as appearance or motion features, abstracting higher level characteristics of the scene.

Let us analyze the advantages and drawbacks of both approaches: “top-down” approaches are highly dependent on tracking and detection phases. As consequence, if one of them fails, the detection or recognition of the event could be compromised. Besides, the complexity of the problem depends on the number of objects that appear on the scene, making the situation unmanageable when there are occlusions, preventing a good performance at detecting anomalies. Thus, if the scene is crowded, then “bottom-up” approaches are more suitable than top-down due to that “bottom-up” approaches can detect the intrinsic information of the scene reducing the need of trackers and detectors.

Nevertheless, “bottom-up” strategies have one disadvantage. Due to that processing is firstly done at pixel level, the algorithm does not have a real notion about the objects on scene. This means that if this technique is used when a bottom-up approach is more suitable, the performance of the algorithm might be poor. There are plenty of methods in the literature using a “top-down” strategy. This is the case of [19–23]. On the other hand, methods proposed in [1,3,6,8,11] follow a “bottom-up” scheme.

2 Classification by level of supervision

Other possibility is to classify the methods by the level of supervision used when building their models. There are three main groups: supervised learning, unsupervised learning and those in which previous models are used. Besides, it is possible to find semi-supervised and weakly-supervised approaches. The first one uses partially-labeled data and the second does not use exhaustive labeling (noisy labels are used instead), like in [24], in which the user does not need to locate explicitly the behavior of the targets in the training video.

2.1 Supervised learning

In supervised learning, the classes are modeled based on the known labels of the data and, after that, use them to classify new instances. It is important to note that supervised learning approaches depend on the accuracy of the labels of the training data, that have to be unbiased.

This learning method is directly conditioned by the availability of labeled training data. A classification can be made based on the availability of labeled samples [12]:

- Normal and anomalous events are given.
- Only normal events are provided.
- Only anomalous events are given.
- Different classes of labeled events are provided.

Even if there are available samples for every type of event, the samples of one of the classes might be scarce. In fact, there are usually much more data labeled as normal that abnormal so anomalous events have to be detected when these differ enough from a normality model. Sometimes, the anomalous events are the only events labeled. The best of the cases occurs when both normal and abnormal events can be used. Another case corresponds to the case when apart from the anomalous data label, the events have others tags, allowing the detection and also recognition of specific events.

One remarkable work is given by the authors of [1]. They use a set of normal events to train a hierarchical mixture of dynamic textures. The anomalies are detected if the new samples differ enough from the model created. In [8,11], a supervised learning approach is also used in order to create a dictionary of normal samples. After that, sparse reconstruction cost (SRC) over this dictionary is used to detect anomalous events.

Some other works using supervised learning are also important. This is the case of [17], where features based on motion, size and texture from each cell are used for posterior anomalous event detection. The authors of [25] extract motion information and spatial-temporal filters before the use of K-nearest neighbors in the test videos to find anomalous events. Finally, in [3] a Gaussian mixture model is built from the normal events using optical-flow-based data.

2.2 Unsupervised learning

Unlike supervised learning, unsupervised learning schemes do not need labeled training data and relies on automated clustering techniques to build the models. Unsupervised learning approaches have the advantage of the non-dependency on labels coming from users so that the learning procedure is not vulnerable to human errors. However, clustering techniques tend to group samples based on some kind of distance, so the frontiers between classes might not be as well defined as in supervised learning.

Some authors have developed algorithms for anomaly detection using this type of learning. For instance, authors of [26] propose an unsupervised sparse coding approach for detecting unusual events. If a test sequence can be reconstructed from a learned event dictionary, then it is considered as normal. Their procedure works in an online manner, modifying the model as new data enters the system.

The method proposed in [6] works under this scheme. Therein, a codebook of normal and abnormal behaviors is automatically constructed assuming that normal events are much more frequent than anomalous. Thus, training data including abnormalities can be used. At first, spatio-temporal volumes of the sequence are used and after that, information at pixel level (low-level) is extracted from them. Finally, the volumes are organized into contextual graphs to give contextual information about their co-ocurrence statistics.

Authors of [27] proposed a method in which “scan statistic” is applied. Under this method, the videos are processed using windows of different sizes and shapes. Then, a likelihood ratio is applied to check whether or not the features within the window are similar to the outer ones.

2.3 A priori modeling

The use of previous information such as models or parameters is a way to avoid a training phase. This kind of approach is only applicable when the nature of the event in question is supposed to remain the same, since the previous information is usually fixed and does not change during the execution of the algorithms. This fact might generate problems if the correct assumptions are not done a priori.

Working under this scheme, authors of [28] make use of a previously constructed model of normality on crowds under the principle of social force for pedestrians. The abnormalities are detected if the associated events differ enough from the given model.

On the other hand, authors in [29] propose the use of interaction models to make inference about new interactions. In the field of surveillance of people (especially for elder and disabled people), [30] applies a fixed threshold for abnormality detection after applying a previously built model.

3 Classification by type of model

Organizing the different types of models used for anomaly detection is possibly the most striking task due to the multitude of approaches in the literature and the difficulty of assigning a unique category to the methods. Despite this, a general organization can be done through the similarities between them. Authors of [10,12,13] have tried to make this classification, helping us to make our own.

For instance, they agree in highlighting dynamic Bayesian networks (DBN) as one of the main techniques for modeling behaviors and events, including the frequently-used hidden Markov models; Bayesian Topic Models (or bag-of-words model); Clustering in order to extract inner groups of features; and Sparse Representation models, in which a basis of representative samples is constructed.

3.1 Dynamic Bayesian networks

A Bayesian network is a graphical model that represents the conditional dependencies between random variables [31]. The maximum exponent of these networks for anomaly detection are hidden Markov models (HMM) that are into a category of Bayesian networks known as dynamic Bayesian networks. In Hidden Markov models, the information used is temporal, so only past events influence next events (Markov assumption).

An HMM is defined by matrices encoding the possible states and the observation probabilities (transition and emission matrices). Usually, these matrices are calculated by using the Baum–Welch algorithm. After that, by using the Viterbi algorithm a new set of observations is evaluated. HMM can solve problems such as determining the most likely sequences of states of an abnormal activity or giving an estimation of the parameters that fit better in the sequence of states.

HMMs can solve very well the correspondence between sequential events, fitting well in the context of anomaly detection, as the anomaly can be represented by a set of actions of events. The high attention that HMMs have received in the literature proves the success that this model obtains. However, they have some disadvantages. One of

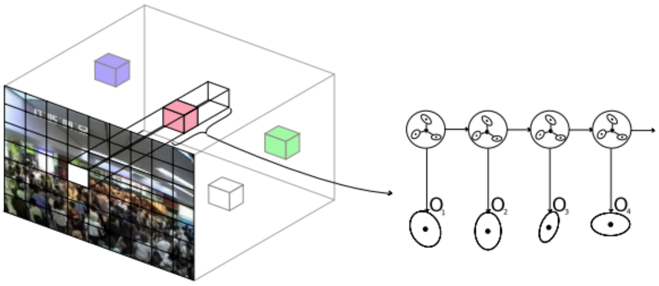


Figure 2.2: Behavioral models of each spatio-temporal volume by using HMM in [5]

them is the computational and memory cost of the Viterbi algorithm for evaluating new sets of observations.

Another difficulty is to determine the appropriate number of hidden states of the model. Some variations have been proposed in order to solve these problems. For instance, authors of [32] apply an infinite hidden Markov model (iHMM). Some other methods such as Multi-Observation HMM (MOHMM) have been proposed for anomaly detection.

Finally, we can conclude that the main advantage of HMMs is that they fit very well in videos with sequential actions, characterizing the change between states through probability distributions. On the other hand, Viterbi algorithm is expensive in terms of computational cost and memory and due to its markovian nature, usual HMMs take only into account the present state and not the past ones (memoryless).

The method proposed by the authors of [5] is a good example of anomaly detection in crowded scenes using HMMs. They fashion the motion variation of spatio-temporal volumes to create a behavioral model of the different events that appears on scene. Their overall strategy is summarized in Figure 2.2, where the succession of events is characterized though the use of a HMM.

3.2 Bayesian topic models

Approaches working under this model were originally created to classify documents, in which the words contained are clustered into topics. When similar words are found on new documents, its topic can be deduced. The name of this approach is known as bag-of-words. The same strategy can be applied on video sequences in order to model the normality of the scene, where instead of words, discriminative features of the image are used to define the “topics” of the video. The anomaly would be detected as an outlier, meaning that it cannot be well explained by the previously created model.

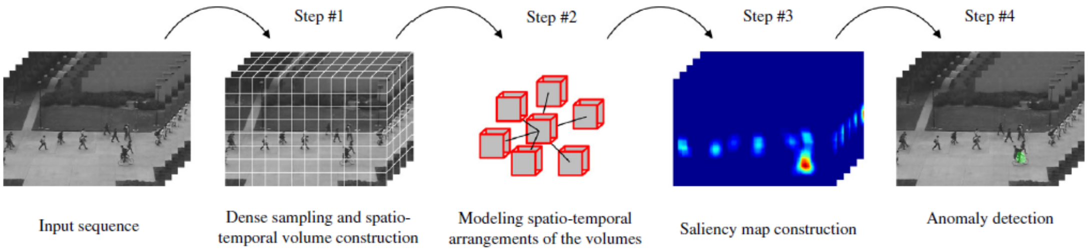

Step #1
Step #2
Step #3
Step #4
Input sequence
Dense sampling and spatio-temporal volume construction
Modeling spatio-temporal arrangements of the volumes
Saliency map construction
Anomaly detection

Figure 2.3: Extended *bag-of-words* approach used in [6]

The most noticeable method of this section for anomaly detection is presented in [6], in which video events are learned at each pixel. In Figure 2.3 the main steps of their strategy for anomaly detection are presented. Besides, authors of [33] used generative Bayesian models such as Latent Dirichlet allocation (LDA) and Hierarchical Dirichlet Processes (HDP) in order to model activities and interactions in crowded and complicated scenes. Niebles *et al.* [34] used probabilistic latent semantic analysis (pLSA) in addition to LDA to characterize human actions into the sequences.

LDA makes the assumption that the videos are formed by a set of actions (topics) to create a probabilistic model and infer if an action is likely to belong to it. However, it has some disadvantages, such that the number of actions or topics needed to be known a priori. In contrast to LDA, pLSA can take into account correlation but over-fitting may appear. There exist variations of the original algorithms solving, at least partially, the problems of these approaches. For instance, HDP is a non-parametric generalization of LDA that might improve the results of the others, since it automatically finds the number of topics.

3.3 Gaussian Mixture Models

Clustering activities through Gaussian Mixture Models (GMM) has also been used for anomaly detection. GMMs are parametric models in which it is assumed that the probabilistic distribution of a set of samples can be estimated with a mixture of a fixed number of Gaussian components.

The estimation of the empirical distribution of the GMM is unfeasible because the real assignments of the samples are not observed so they have to be estimated. This is usually done by means of the Expectation-Maximization algorithm [35], which aims to maximize the likelihood given the data and obtain the parameters (weights, means and covariances) of each of its components.

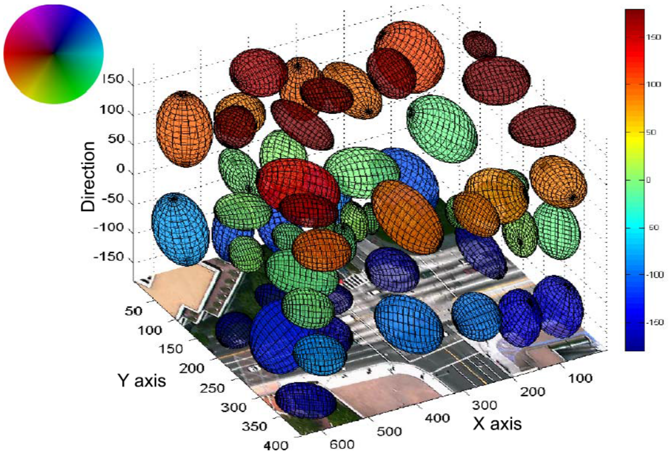


Figure 2.4: Clusters of features based based on optical flow from [7]

GMMs have been also used for anomaly detection using different features to build the model, for instance trajectories and features extracted from spatio-temporal volumes. Examples of methods that use trajectories are [7, 14, 31]. The first one is based on the popular method by Stauffer and Grimson [36] to model the background of the scene but adding the modeling of distributions at pixel-level of speed and size of the objects by using multivariate GMMs. Authors of [14] use also a GMM to train a model with video-clips of normal crowd flows based on particle trajectories and chaotic invariant features. Authors of [7] cluster the patterns of motion described by optical flow in the scene to detect anomalies. A simplified representation of the clusters that they obtain is displayed in Figure 2.4, where the features such as mean, covariance and direction of the events are represented over the scene.

In addition to trajectories, other features such as textures of optical flow and velocity have been included on descriptors to train GMMs. This is the case of [3] and the posterior method proposed by the authors of [4], who added orientations of optical flow and acceleration features to detect anomalies in pedestrian scenarios.

The principal advantage of a GMM is that after the training phase, soft assignments of incoming samples to the clusters are given. On the other hand, it is necessary to fix the number of components of the GMM a priori and if the descriptors are large, the computational cost of the training phase can be elevated, specially if the number of components is elevated.

3.4 Mixture of dynamic textures

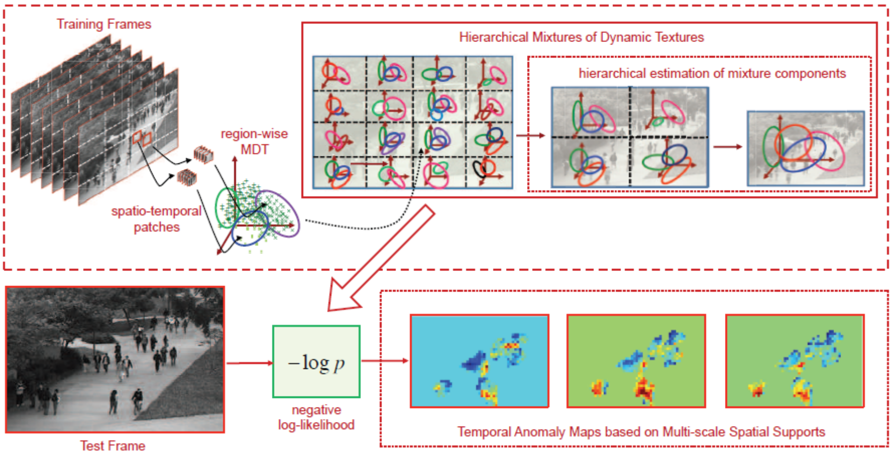

Training Frames
Hierarchical Mixtures of Dynamic Textures
hierarchical estimation of mixture components
region-wise MDT
spatio-temporal patches
Test Frame
$-\log p$
negative log-likelihood
Temporal Anomaly Maps based on Multi-scale Spatial Supports

Figure 2.5: Hierarchical MDT for anomaly detection from [1]

A dynamic texture is a spatio-temporal generative model that represents video sequences as observations from a linear dynamical system [37]. It is based on the original dynamic texture method proposed by [38], in which the characterization of problems of modeling, learning, recognition and synthesis of dynamic textures is proposed, following the assumption that sequences of moving scenes have inherent stationary properties.

Under these ideas, the Mixture of Dynamic Textures (MDT) [1, 2, 37] was proposed to detect anomalies in video sequences, by means of clustering dynamic textures after the division of the videos into spatio-temporal patches. As the authors claim, while other method focus on appearance (such as texture and color) or motion (given by optical flow), MDT integrates both appearance and dynamics to get a robust anomaly detector. They get this by analyzing the videos in the temporal and spatial axes. Anomalies in test sequences are detected as outliers of the MDT model learned in the training phase. It integrates both temporal and spatial strategies together with multi-scale support under a hierarchical approach as it is represented in Figure 2.5.

MDT has the advantage of considering both appearance and motion dynamics of the scene in order to detect temporal and spatial abnormalities but, on the other hand, it is a complex and costly model. Besides, other simpler methods get comparable results detecting abnormalities.

3.5 Sparse representation models

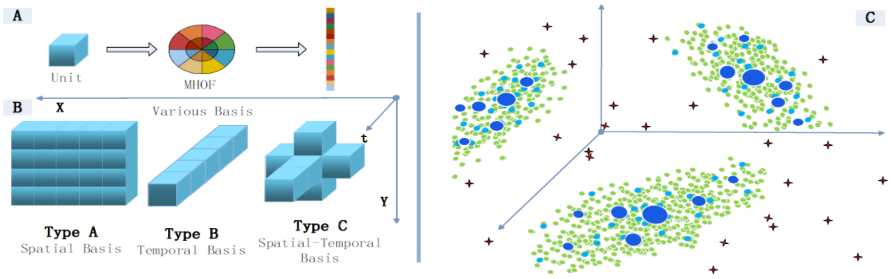


Figure 2.6: Strategy and representation of samples from [8]

The success of this technique has been proved by the authors of [8,11]. They propose to detect abnormal events through the sparse reconstruction of new samples over normal bases, using for that the Sparse Reconstruction Cost (SRC) over the trained dictionary to see how the test sample is likely to be normal. They assume that in a class of samples, there are strong features acting like a basis so the samples can be reconstructed by a combination of them.

They use multi-scale histograms of optical flow as feature extracted from every patch under different basis in space and time. The strategy followed and the representation of the samples in three dimensions is displayed in Figure 2.6. Blue points represent the robust samples that act as basis, the green points the normal samples and in red the detected anomalies. Another method using this approach is given by the authors of [39], who combine the use of sparse representation models with textures defined through local binary patterns (LBP).

Sparse reconstruction methods have the advantage of being intuitive and the dictionary learning helps to reduce the dimensionality and discard noisy samples. It also offers good results, making it an important method in the literature. Its main drawback, as shown in [1], it is the large amount of time needed per frame to detect the anomalies.

3.6 Other models and techniques

Some of the important works have been classified into categories but additionally, there are plenty of methods dedicated to anomaly detection that may not be assignable to one of these groups. The reasons to not include them into this classification are that they do not get remarkable results, the methods proposed are too similar to other more important methods or the use of their techniques is not extended. This can be the case of methods based on neural networks [40], manifold learning [41], fuzzy reasoning [42], decision trees [43], etc.

4 Classification by type of feature

Essentially, what one aims to detect when looking for anomalies into surveillance videos are objects whose features differ somehow from those considered normal. Therefore, features play a fundamental role within the algorithms since they have to discriminate well between normal and anomalous events.

The use of optical flow is widely extended among the methods that aim to detect anomalies in video sequences, essentially because it extracts rich information about motion orientation and magnitude at pixel-level. There are plenty of possibilities when building an optical-flow-based descriptor. One recurrent option is to group the orientation of the motion by using histograms combined with other features such as optical flow magnitude [8, 11, 25]; other possibility is to include directly the magnitude from both vertical and horizontal flow [3] or optical flow acceleration from one frame to another [4]. In addition, authors of [3] make use of what they call textures of optical flow, which measures the uniformity of spatio-temporal volumes by summing up the optical flow values of neighboring pixels at a specified distance.

Other authors prefer the use of gradients. In [6], authors use histograms of oriented gradients (HOG) for each spatio-temporal volume; in [5], authors model the distribution of spatio-temporal gradients using a three-dimensional Gaussian distribution.

If the method is object-based instead of pixel-based, other features are more suitable than the prior ones, such as position, velocity, size or centroid. Histograms of color and pixel change, saliency and curvature and contour information are other features proposed in the literature [10].

Chapter 3

Construction of a normality model

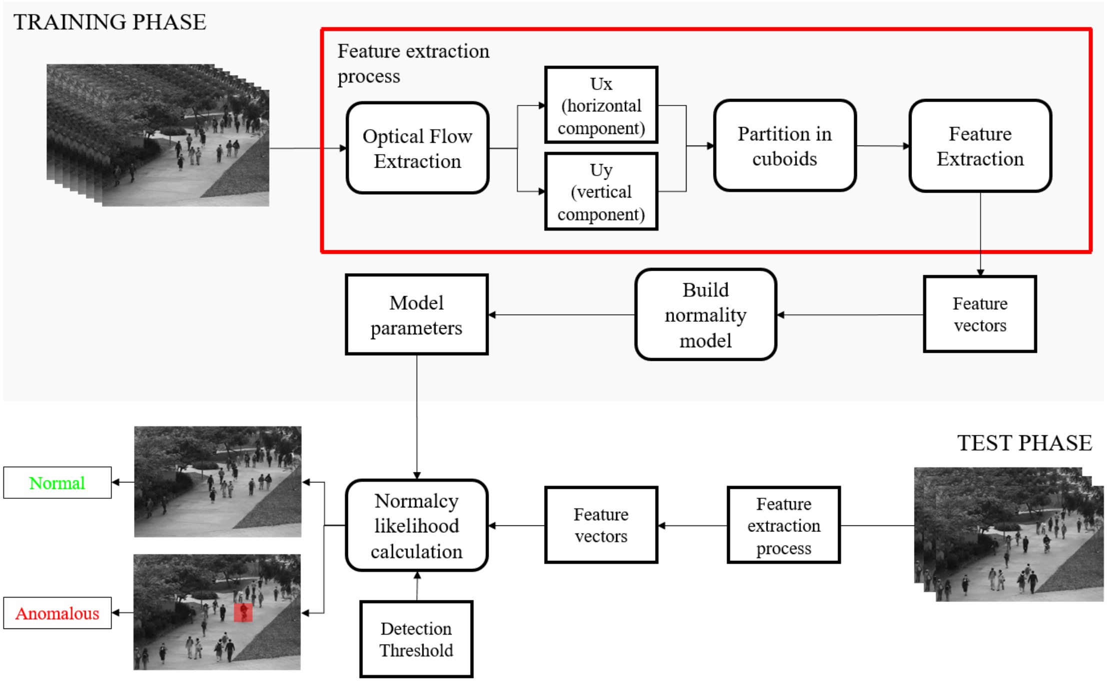


Figure 3.1: Architecture of the system

The first thing we have to think about is the context in which we want to detect anomalies. In our case, we want to detect anomalies in pedestrian scenarios (which might be crowded) and the anomalies, as in the majority of the literature methods, are not well-defined. For instance, unusual motion patterns, people running, circulation of forbidden vehicles or restricted areas could be reasons to raise an alarm. Usually, within these scenarios, these anomalies are not frequent, making their detection even harder.

Taking these aspects into account, some strategies fit better than others. This is commented in section 2.1, where it is explained that even though there are different ways to train models to detect anomalies, the task of anomaly detection is usually conditioned by the scarcity of data containing anomalous events. Thus, the strategy that presumably fits better with our purpose is training a normality model of the scene, in order to detect the anomalies when they differ from the normal events.

1 Architecture of the system

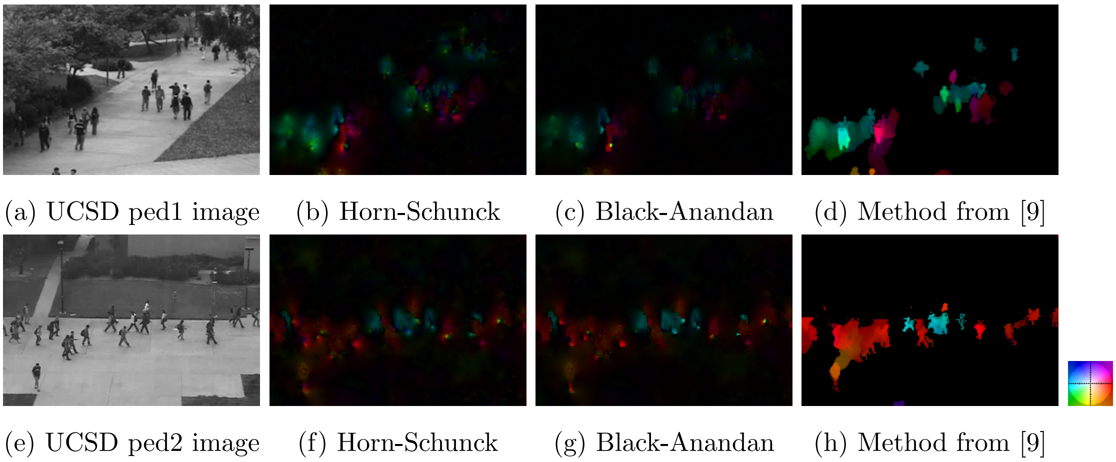
Figure 3.1 portrays the architecture of the system, that includes training and test phases. During the training phase there are two well-differentiated sections, corresponding to the feature extraction process and the construction of the model, with the feature vectors as inputs. Then, the resulting parameters of the model are given to the test phase, in which after the feature extraction step, the likelihood of the features of the frame under the normality model is calculated. A threshold is set here to establish when an anomaly should be detected.

1.1 Feature extraction: Robust optical flow

In this project, a "bottom-up" strategy is used. This means that the information used to build the descriptor has to be extracted at pixel level so in ours, as in many other approaches, we obtain optical flow fields describing horizontal and vertical components of the displacement of the pixels u_x and u_y from one frame to the next.

$$E(u_x, u_y) = \int [(I_x u_x + I_y u_y + I_t)^2 + \lambda(\|\nabla u_x\|^2 + \|\nabla u_y\|^2)]dxdy \tag{3.1}$$

Multitude of methods from the literature work with optical flow, and some of them employ methods based on the classical Horn-Schunck (HS) formulation (Equation 3.1) that works with brightness constancy and spatial smoothness constraints. The objective is to minimize the functional, that includes a regularization parameter λ (the larger the parameter λ the smoother the optical flow field). Therein, the first part is the data term, that includes the derivative of the images intensity values over x, y and temporal axes. The second part is the penalty term, formed by the norm of the flow field to be optimized.



(a) UCSD ped1 image (b) Horn-Schunck (c) Black-Anandan (d) Method from [9]

(e) UCSD ped2 image (f) Horn-Schunck (g) Black-Anandan (h) Method from [9]

Figure 3.2: Optical flow comparison. Horn and Schunck vs Black and Anandan vs method proposed in [9].

In practice, this method offers reasonable results. However, since the creation of the original algorithm, the method has been outperformed by new approaches proposed in the last years that handle problems that HS did not, such as reflections. This is the main reason why, for our purpose, optical flow field is obtained by means of a more robust method proposed in [9] to calculate the motion between successive frames so we can construct more reliable descriptors.

The principal advantage of using this robust algorithm is that is based on the original formulation by Horn and Schunck. However, it includes modern techniques that get remarkable improvements in terms of accuracy. Each of these techniques implemented by the authors of [9] was tested to check the impact that its addition produces on the final result, extracting the best performance from all the parameters. The details of the method are given in Appendix A, although we can introduce the main techniques and their implications here.

To gain robustness against illumination changes, they propose to use a method to separate structure and texture of each of the images as proposed in [44]. Additionally, the estimation of the flow is done gradually with a multi-resolution pyramid, interpolating the last flow towards the next image at the higher resolution using bi-cubic interpolation. In any case the two techniques that produce remarkable improvements are the use of a graduated non-convexity approach (for the penalty function) and a weighted median filter to remove outliers.

Figure 3.2 portrays a visual comparison between different methods to get the optical flow, extracted from two different sequences of UCSD dataset [2]. The first thing that attract the attention is the energy that has the method we use and that it has less

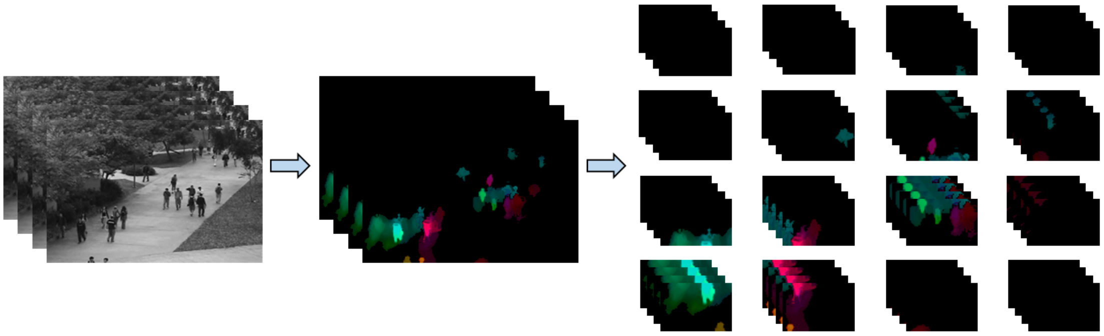


Figure 3.3: Construction of cuboids with optical flow fields

outliers. Besides, compared with Horn-Schunk and Black-Anandan (also used in the literature) methods, objects appear sharper, giving a better description of the movement of the object. A more detailed comparison of these optical flow methods is given in Appendix A.

Once we have obtained the optical flow between frames, we can build from them spatio-temporal patches (depending on the number of temporal frames selected), as we explain in the next section.

1.2 Partition in spatio-temporal patches

To extract the information required to build the model, it is necessary to separate the optical flow fields in groups of a number of frames fixed a priori, corresponding to the temporal depth of the patches. The number of frames has to be large enough to gather the minimum information about the events on the scene while being relatively small so that the context does not change much.

The next step is to divide these groups of frames in a spatial manner so spatio-temporal patches (also called cuboids) are formed, making possible the analysis of local events. Spatial size of these cuboids need to be large enough to contain the minimum information about the events but small enough so no more than one event are included into the cuboid. In Figure 3.3, a schematic representing the procedure to build the cuboids is displayed. Note that we always handle the horizontal and vertical components of the optical flow, even though in the images we show magnitude and orientation on the same figure.

1.3 Normality model: Gaussian mixture model

Applying GMMs to anomaly detection permits, as introduced in Section 3.3, the construction of a probabilistic framework that allows soft assignments of the test samples to the components of the mixture, in order to classify them either as normal or abnormal. The use of GMMs for anomaly detection is not as extended as other models such as hidden Markov models. Nevertheless, GMMs have shown potential, as it can be observed in the methods proposed in [3] or [4], getting results comparable to the best state-of-the-art methods, but with less conceptual complexity compared to methods such as MDT, used in [1].

The process to obtain the model is as follows. Firstly, a GMM is built to model the normality of the scene using the training samples. Once the model is created, it is used in the detection phase to calculate how the new events are likely to be normal. This could be seen as a discriminative model, since the probability of being anomalous is calculated given the probability of being normal. By doing so, GMMs have the advantage that they allow to train the model with sequences that only contains events considered as normal, even with abnormalities if we assume that they are much less frequent. Thus, in the test phase the abnormality is detected if the associated event is not modeled in the training set.

In order to understand how GMMs can be implemented to apply them to anomaly detection, definition, construction as well as limitations are described in the next sections.

1.3.1 Definition

A GMM is a parametric model formed by K multivariate Gaussian distributions. Each of the distributions has a weight π_k, a covariance matrix Σ_k and a vector μ_k containing the means of each of the variables the data samples $X = \{x_1, \ldots, x_n\}$ (n is the total number of samples). Thus, a GMM is completely defined by the three components from above for each of the mixture components $\Theta = \{\pi_k, \mu_k, \Sigma_k; k = 1, \ldots, K\}$. The likelihood of a sample given the parameters is a weighted sum of its probability under each of the components (Equation 3.2).

$$p(x|\Theta) = \sum_{k=1}^{K} \pi_k p(x|\mu_k, \Sigma_k), \qquad p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \tag{3.2}$$

The idea of using a GMM is to model the normalcy of the scene using a probabilistic framework with the information contained in the descriptor, that is supposed to discriminate between normal and anomalous events.

1.3.2 Construction: Expectation-Maximization algorithm

The construction of the GMM from a given dataset can be done by means of the maximization of log-likelihood of the data (Equation 3.3). The problem with this approach is that its maximization is hard to perform because the real assignments of the data samples to the components of the GMM are not known. In any case, we can approximate the real distribution of the GMM by means of the EM algorithm. In order to understand why it does works, the formulation proposed in [45] is helpful, which is based at the same time in the method proposed by [35].

$$\hat{L}(\Theta; X) = \log p(X|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(x_n|\mu_k, \Sigma_k) \tag{3.3}$$

$$\phi\left(\int g(x)f(x)dx\right) \geq \int \phi(g(x))f(x)dx \tag{3.4}$$

As detailed in [35] and [45], we can use a latent variable H and Jensen's inequality [46] for concave functions (Equation 3.4) to find a lower bound of the real log-likelihood of the empirical distribution $\tilde{p}(X)$ through Equation 3.5. This permits us to understand how it is possible to estimate the lower bound of the real log-likelihood of the model without knowing the real assignments of the data to the components of the GMM.

Firstly, we multiply and divide by $q(H|X)$ so we can apply Equation 3.4 (inequality number 1 on 3.5) and the logarithm enters the integral. Note that after applying Jensen's inequality, distribution $q(H|X)$ multiplies the logarithm and because of that, after equality number 2 the estimation of the lower bound is over $q(H|X)\tilde{p}(X)$ instead of $\tilde{p}(X)$ only. Besides, if $q(H|X) = p(H|X, \Theta)$ (equality number 3), then the lower bound is maximized. Finally, the last term of Equation 3.5 corresponds with the lower bound of the real distribution of the dataset X but jointly distributed with the distribution of the latent variable H.

$$
\begin{aligned}
\mathcal{L}(\Theta; X) &= \mathbb{E}\left[\log \int q(H|X)\frac{p(X,H|\Theta)}{q(H|X)} dH\right]_{X\sim\tilde{p}(X)} \\
&\overset{1}{\geq} \mathbb{E}\left[\int q(H|X) \log \frac{p(X,H|\Theta)}{q(H|X)} dH\right]_{X\sim\tilde{p}(X)} \\
&= \mathbb{E}\left[\log \frac{p(X,H|\Theta)}{q(H|X)}\right]_{(X,q)\sim q(H|X)\tilde{p}(X)} \\
&\overset{2}{=} \mathbb{E}\left[\log p(X,H|\Theta)\right]_{(X,q)\sim q(H|X)\tilde{p}(X)} - \mathbb{E}\left[\log q(H|X)\right]_{(X,q)\sim q(H|X)\tilde{p}(X)} \\
&= \mathbb{E}\left[\log p(X,H|\Theta)\right]_{(X,H)\sim p(H|X,\Theta)\tilde{p}(X)} - \mathbb{E}\left[\log p(H|X,\Theta)\right]_{(X,H)\sim p(H|X,\Theta)\tilde{p}(X)} \\
&\overset{3}{=} \mathbb{E}\left[\log \frac{p(X,H|\Theta)}{p(H|X,\Theta)}\right]_{(X,H)\sim p(H|X,\Theta)\tilde{p}(X)}
\end{aligned}
\tag{3.5}
$$

It is important to observe that the last term of Equation 3.5 is composed by the log-likelihood of the model, with the variables X and H observed (samples of the dataset and latent variable) and the entropy of the latent variable H. In the case of a GMM, the latent variable are the point-to-cluster assignments. This is the utility of Equation 3.5), since we can firstly use random assignments of the data to the GMM components and use them as observed to maximize the lower bound of the log-likelihood of the real distribution.

Therefore, EM algorithm can be used in an iterative manner by executing the expectation step (E-step) and the maximization step (M-step). In the expectation step, $p(H|X,\Theta)$ is updated given X and Θ, while in the maximization step the lower bound is maximized taking the results of the expectation step, since now X and H are observed.

- **E-step**: $p(H|X,\Theta)$ is evaluated. This is, which assignments point-to-cluster are expected given the parameters of the GMM Θ and the dataset$X$. At the first iteration, the initial parameters can be assigned randomly or by using an unsupervised clustering technique. The distribution $p(H|X,\Theta)$ can be seen as a matrix Q with size $n \times K$, in which each of the elements Q_{ik} is the probability that the sample i belongs to the mixture component k. The calculation of Q_{ik} is shown in Equation 3.6.

$$
Q_{ik} = \frac{\pi_k p(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j p(x_i|\mu_j, \Sigma_j)} \tag{3.6}
$$

- **M-step**: Within the M step, the parameters of the mixture components and their weights are estimated using the distribution evaluated in the E-step. This can be done because the assignments to the clusters now are fixed, making the estimation simple. With this data, the log-likelihood of the data $\hat{L}(X|\Theta) = \sum_{n=1}^{N} \log p(x_n|\Theta)$ is evaluated. The updated parameters of the GMM are calculated in the following manner:

$$\mu_k = \frac{\sum_{i=1}^{n} Q_{ik} x_i}{\sum_{i=1}^{n} Q_{ik}}, \quad \Sigma_k = \frac{\sum_{i=1}^{n} Q_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} Q_{ik}}, \quad \pi_k = \frac{\sum_{i=1}^{n} Q_{ik}}{\sum_{i=1}^{n} \sum_{j=1}^{K} Q_{ij}} \tag{3.7}$$

EM algorithm always converges but can take too many iterations. To stop its execution, one possibility is to check if the degree of change of the log-likelihood is small enough or setting a maximum number of iterations.

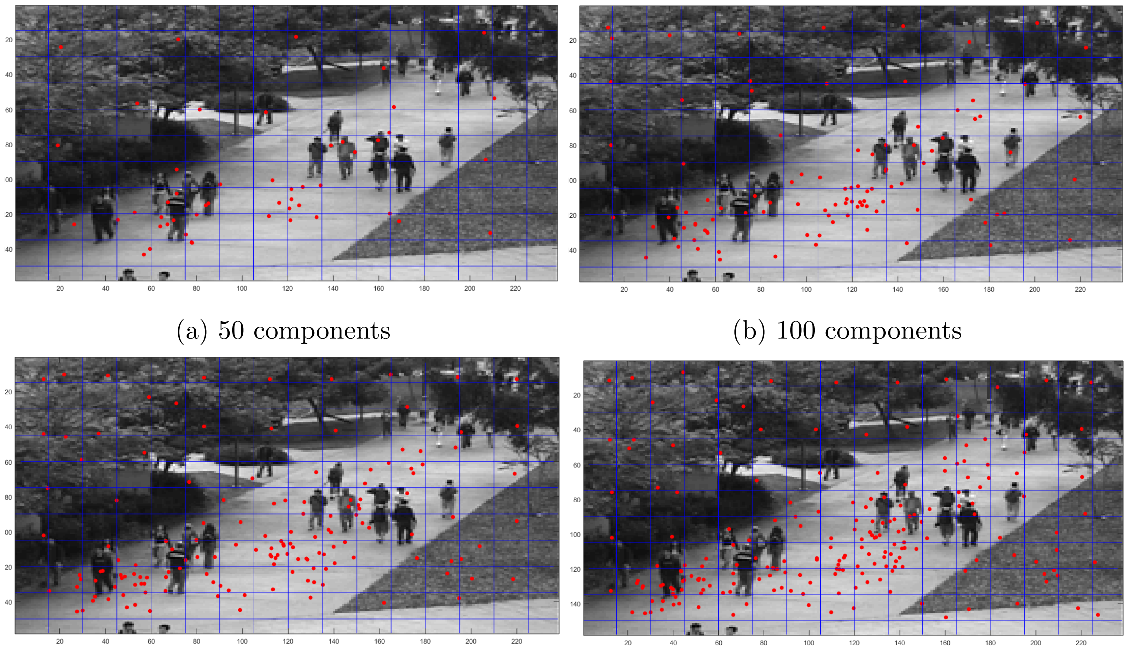
1.3.3 Limitations

Although the use of a GMM has the advantages commented for anomaly detection specially when a large enough normality dataset is given, there exist some drawbacks that is convenient to analyze:

- **Number of components in the mixture**: The main drawback of the GMM is that before the training phase, the number of mixtures of the model has to be fixed. This implies that EM algorithm is forced to assign the data to that number of components, even if a different number of components is optimal.

$$BIC = -2\hat{L}(X;\Theta) + K \log(N) \tag{3.8}$$

To mitigate this and to try to automatize the search of the optimal number of components, a possibility is to use the Bayesian Information Criterion (BIC) presented in [47] and shown in Equation 3.8. $\mathcal{L}(X;\Theta)$ is the log-likelihood of the data given the parameters of the GMM; K is the number of components in the mixture and N is the number of samples of the dataset.



(a) 50 components

(b) 100 components

(c) 150 components

(d) 200 components

Figure 3.4: Clusters initialization for different number of components performed with K-means++

The simplicity of this criterion is that the model with the lowest value of BIC can be chosen to automate the selection of the components. When BIC is computed, the log-likelihood value obtained is penalized depending on the number of components and the size of the dataset. The idea under this is that if the growth of the log-likelihood value does not compensate the penalty, then the value of BIC is increased (a low value of BIC is better) reaching a number of components optimal where the value of BIC is the lowest.

When BIC is used the EM algorithm has to be performed several times with an incremental number of components until the minimum value of BIC is obtained as it is done in [3] or [4].

- **Initialization of the clusters**: GMMs are very sensitive to the initialization of the centers of the mixture components. If this process is done randomly, the precision of the GMM to model the normal events is likely to be poor. An unsupervised clustering method such as Lloyd's algorithm (K-means) is presented as a better option for this purpose.

  At the same time, the initialization of the position of the centroids (seeding) on this last can be improved by means of a method that forces the initial centroids to be separated. This is done by using probability distributions that depends on the distance to the already set centroids, so the probability to assign the position of a centroid near to another is lower that if it is far. This initialization differs from that of the original K-means, where this step was done randomly with a uniform probability distribution.

This modified algorithm is called K-means++ [48]. In Figure 3.4, we can see how this algorithm distributes the centers of the clusters over the scene with the red dots. We can see how the clusters tend to accumulate on the road, area where the majority of events are concentrated. Thanks to this step, the EM algorithm used to build the GMM distribution needs less iterations to reach an optimal position.

1.3.4 Global vs local GMM

There are different ways to integrate a GMM to the task of anomaly detection. For instance, a GMM could be applied onto the whole scene (global GMM). Another possibility is to build a GMM into every spatio-temporal cuboid (local GMM).

The main problem with a global model is that it is not possible to take advantage of the spatial information of the sequence, since a constant behavior all over the scene is wrongly assumed (a normal behavior on an area of the scene does not have to be normal on another). The alternative, in order to continue using a global GMM, is to include into the descriptors some spatial information, such as the position of the cuboids (coordinate (x,y) of each cuboid central point). Since we decided to use a global GMM approach, the final descriptor look as described in Section 1.4.2.

A global GMM is an elegant solution, since the problem of anomaly detection is solved with just an unique model for the whole scene. One advantage over the use of local GMMs, is that the final GMM distribution will ignore areas where there are no interesting events, such as areas with trees, zones without movements, sky, etc. This fact is visible in Figure 3.4, where only a few clusters are located on this areas. This means that a more detailed representation of the normality on the interesting areas is automatically performed. We will see in the next chapter, that the global GMM is able to successfully build a model to detect anomalies.

1.4 Construction of discriminative descriptors

Usually, into public video-surveillance scenarios, the targets are pedestrians walking around sidewalks, parks, malls and so on. Thus, it is reasonable to assume that anomalous events would be created by people running or going through unusual and restricted areas or with unusual motion patterns. Other objects appearing on scenarios where they are not expected, such as vehicles or animals, should also raise an alarm in specific conditions.

On the other hand, the anomaly could be produced on determined infrastructures such as elevators, escalators and close to different types of furniture such as automated doors. On these cases, detecting events should be done in case of malfunction or mishandling. As we see, there are many contexts, whose anomalies can be very different. Because of these reasons, the descriptors of the model have to be chosen carefully, studying the best features for each specific scenario.

In our approach, the descriptors are extracted from each cuboid, whose content is its correspondent optical flow field. Thus, the process to improve the descriptors from the flow fields is done in an incremental process. This is, add features, check the best combination of parameters and finally, add new features, obtaining the descriptors described in the next sections, ensuring that they fit with the context.

1.4.1 Initial descriptor: space and motion features

As we see, descriptors depend strongly on the context where the anomalies are produced. In any case, one basic information to include into the descriptor can be motion, because in any case, it can model the expected magnitude and orientation of the movements on scene and use it to detect outliers. Optical flow field between frames gives this information by means of magnitude and orientation of the motion at pixel-level.

There are different possibilities to include this data into the descriptor. One is to use histograms of orientations and magnitude of motion over the cuboids. Other possibility is to sum up all the values of horizontal and vertical flows over the cuboid, maintaining the sign (Equation 3.9), to have both average magnitude and orientation over the cuboid, as it is done in [3]. This is the baseline feature of our descriptor that has the advantage that it does not enlarge the descriptor in excess, giving magnitude and sign of both vertical and horizontal components of the optical flow in just two features.

$$\hat{\sigma}_x = \sum_{p \in C} u_x, \ \hat{\sigma}_y = \sum_{p \in C} u_y \tag{3.9}$$

Optical-flow features extracted from each cuboid C gives the minimal information to detect anomalies with data about motion. Additionally, to have notion about the spatial location of the cuboids, coordinates (x, y) of the center of the cuboids are also included, since a global GMM approach is used. The descriptor would look as follows:

$$\textbf{Initial descriptor (D1)}: \quad (x, y, \hat{\sigma}_x, \hat{\sigma}_y) \tag{3.10}$$

1.4.2 Improved descriptor: uniformity of optical flow

$$U_\delta = \sum_{p \in C} u_x(p) u_x(p+\delta) + u_y(p) u_y(p+\delta) \tag{3.11}$$

Thanks to the information packed in the initial descriptor (D1), it is possible to detect anomalies based only on events with anomalous motion patterns. One solution is to use uniformity of optical flow as proposed in [3], also referred as textures of optical flow (Equation 3.11). For example, if the feature is extracted from the optical flow produced by pedestrians, the texture is likely to be non-uniform because of the movements of the limbs. If extracted from the optical flow of rigid objects, the texture is likely to be uniform, allowing the detection of anomalies not produced by anomalous motion patterns but because of different appearances.

Uniformity values are based on the summation of the product of optical flow values at one point and a neighbor separated by δ pixels in the three dimensions of the cuboid. Uniformity with different offset values can be combined and added to the descriptor to get a better representation of the events. Note that since we want to describe the uniformity of the optical flow, uniformity between frames does not have sense, so that the offset in the temporal dimension will be always zero.

$$\textbf{Improved descriptor (D2)}: \quad (x, y, \hat{\sigma}_x, \hat{\sigma}_y, U_\delta) \tag{3.12}$$

Uniformity features extracted from different distances can be obtained in order to get a richer descriptor. In particular, we will use three offsets (1, 3 and 5 pixels) as in [3] to build the descriptor.

Chapter 4

Evaluation and analysis of results

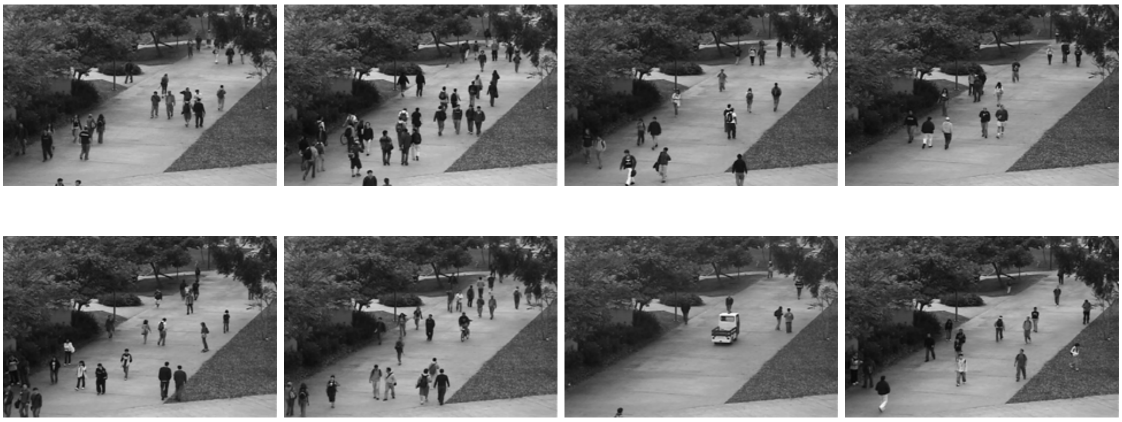


Figure 4.1: Training and test frames of UCSD ped1 dataset (first perspective)

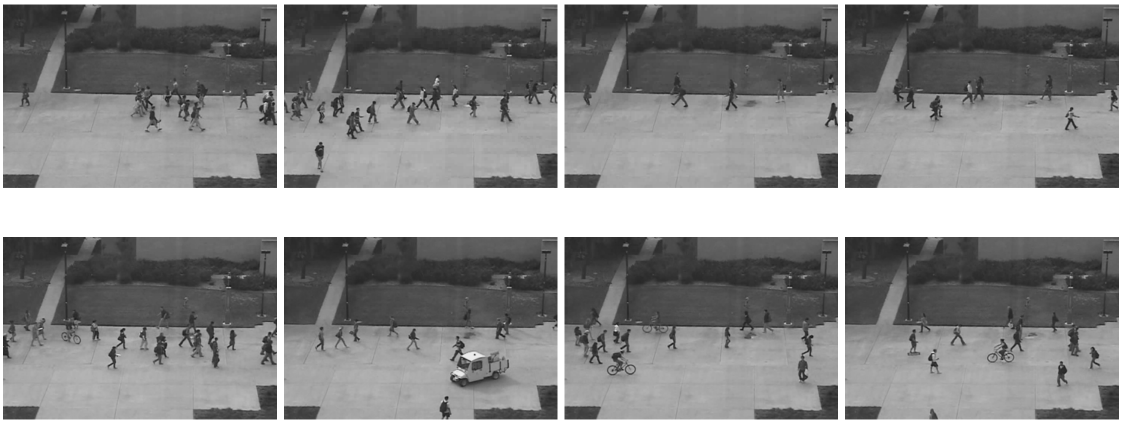


Figure 4.2: Training and test frames of UCSD ped2 dataset (second perspective)

In order to develop a robust anomaly detection algorithm, an exhaustive analysis of the context of the scene in particular is required. Chapter 2 presents different relevant possibilities at choosing strategies to build an anomaly detection algorithm that permits to select the best-fitting techniques concerning the problem. As introduced before, we have to work with pedestrian scenes (that can be crowded) so it is necessary to find subjective and objective methods to evaluate the detection results.

1 Databases

Objective evaluation will be performed on the UCSD dataset [2]. The characteristics of this dataset are very similar to other surveillance environments, specially in terms of variability of events and objects. It was captured in real conditions from a fixed camera pointing to a sidewalk, where pedestrians walk with different motion patterns. This dataset includes training and test sequences. Training sets include only normal events, while test sets include anomalies produced by pedestrians, skaters and vehicles such as bicycles or wheelchairs, with different motion patterns, appearance and speed.

There are sequences taken from two perspectives. With the first (ped1), there are 34 videos for training and 36 for test. With the second perspective (ped2) there are 16 videos for training and 12 for test. The sequences from the two perspectives have a flag indicating whether a frame contains anomalies or not. However, only the sequences without perspective distortion have masks with the exact location of the objects that produce the anomalies. The resolution of ped1 set is 238x158 while the resolution of ped2 is 360x240, although this last one is resized to have the size of the first, to reduce computational cost and to have the same resolution on both datasets.

Some images of ped1 and ped2 sets are displayed in Figures 4.1 and 4.2. First rows show some images from the training sets and the second rows some test frames with anomalies. In Figure 4.1, the anomalies shown are produced, from left to right, by a skater, a cyclist, a vehicle and a person walking through an unusual area over the grass on the right. In Figure 4.2, the anomalies are produced by the same type of objects, with two anomalies in the last two frames produced by two cyclists, and one skater and one cyclist respectively.

2 Evaluation

Baseline method works under a global GMM approach. Recall that to train the global GMM, it is necessary to keep the spatial information between cuboids, so their central coordinates over the frame are included into the descriptors.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \tag{4.1}$$

Evaluation of the detections are performed at frame-level. If the normality likelihood of a cuboid is below a fixed threshold, the frame is marked as a true detection; if not, is marked as false detection. If there is a misdetection, the frame is marked as a false negative. Otherwise, it is marked as true negative. Note that since the cuboids have temporal dimension, the detection of anomalies has a latency that depends on the moment when the anomaly is detected.

The results are given in the form of Receiver Operating Characteristic (ROC) curves, whose horizontal axis indicates the False Positive Rate (FPR) and the vertical indicates the True Positive Rate (TPR), extracting from them the Area Under the Curve (AUC) and Equal Error Rate (EER) values so we can compare our results with those in the literature. AUC is a measure that gives an idea of how well the classification is done. Its values are between zero and one, being one the perfect classification and zero the worst. In any case, a value of AUC smaller than 0.5 means that the classifier works worse than if the classification were done randomly. The EER is the accuracy at the ROC operating point where the false positive and false negative rates are equal, so the lowest EER the better.

We keep the value of EER for further comparison with state-of-the-art methods and to discriminate two results with similar AUC (between two results with similar AUC, the one with the lowest EER is better). The equations of both true positive and false positive rates are in Equation 4.1.

Since we do not know a priori the optimal number of Gaussian components to model the normality with the GMM, this number is incremented in a value of 20, testing from 90 to 550. Then, we represent ROC curves and calculate AUC and EER for each cuboid size and number of components. Additionally, values of BIC (Equation 3.8) are also included. We perform different experiments to tune the parameters of the model and extract the best results. Thus, the strategy is to obtain the best size for the cuboids with a baseline descriptor and use it to perform the evaluation with improved descriptors.

2.1 First experiment: finding the best cuboid size

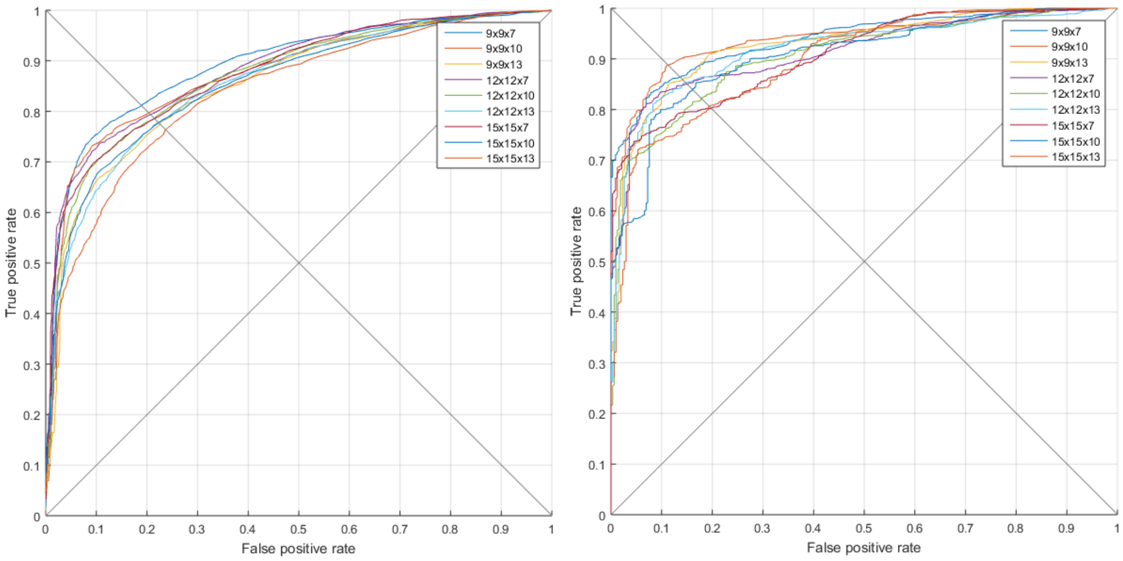


Figure 4.3: ROC curves of the best results for each cuboid size. UCSD ped1 and ped2.

The evaluation focuses on analyzing the performance of the baseline system for variable cuboid sizes to confirm the cuboid size that offers the best results. The initial descriptor used for this purpose is the baseline descriptor specified in Section 1.4 and referred as D1. This, includes the position and summation of the optical flow values of the cuboid.

To begin, cuboid sizes similar to those in the literature are evaluated, slightly changing spatial and temporal sizes to see how results change. Specifically, we begin with a size of 9x9x13 pixels, since it is one of the sizes that performs better in the literature [3, 4]. After that, we modify in a value of 2 pixels the spatial sizes in the vertical and horizontal axes. Temporal size is changed in a value of 3 frames.

To conclude which cuboid size is the best, the highest AUC with its EER is extracted from all the components evaluated for each cuboid size. Then, among the best values of AUC and EER for each cuboid size, the size with best results is selected for both ped1 and ped2 UCSD datasets. In Appendix B, the complete tables containing AUC, EER and BIC values for all components and size of cuboid are given, with the interesting results highlighted in bold, either because the best results are close to each other in AUC or because the difference between the number of GMM components of the best result and the second best is too large (giving priority to models with less Gaussian components).

UCSD ped1 dataset			
Size	AUC	EER (%)	comp
9x9x7	**0.8926**	**18.61**	**530**
9x9x10	0.8751	19.96	410
9x9x13	0.8547	22.28	270
12x12x7	0.8883	20.76	330
12x12x10	0.8711	21.35	250
12x12x13	0.8570	22.26	530
15x15x7	0.8764	21.59	430
15x15x10	0.8571	22.33	370
15x15x13	0.8386	23.59	330

UCSD ped2 dataset			
Size	AUC	EER (%)	comp
9x9x7	0.9386	12.95	210
9x9x10	**0.9401**	**11.04**	**250**
9x9x13	0.9287	13.58	230
12x12x7	0.9142	14.46	370
12x12x10	0.9065	17.67	370
12x12x13	0.9184	14.24	110
15x15x7	0.9092	19.88	270
15x15x10	0.9074	15.77	490
15x15x13	0.8974	19.21	490

Table 4.1: Best AUC and EER results for each cuboid size on UCSD ped1 and ped2 datasets

In any case, to build Table 4.1 only the best value of AUC is chosen, no matter the number of components.

2.1.1 Objective analysis

ROC curves for the different cuboid sizes and datasets are given in Figure 4.3, whose AUC and EER values are given in Table 4.1. Therein, we can see that the two best values of AUC are obtained with the lowest spatial size: 9x9x10 on ped2 dataset and 9x9x7 on ped1 dataset. Additionally, we can see that the values of AUC for all the sizes of the cuboids changes with a similar pattern when spatial size is increased and the temporal is fixed. Except for sizes 12x12x13 on ped1 and 15x15x10 on ped2, the rest decrease their values of AUC when spatial size is increased. When we fix spatial sizes and move on the temporal, we see that a temporal size of seven frames is always better on ped1. This is not clear on ped2, where, only when a spatial size of 9x9 is selected, a temporal size of ten frames seems better.

Further analysis is required to find out which size is appropriate. For this purpose, the next approach is followed. We plot the average AUC at the surroundings of the number of components where the maximum AUC is reached. The goal is to see how differs the average AUC as we move from the optimum to see the global performance over the complete set of Gaussian components used and how flexible we can be selecting a number of components if we cannot select the optimal. Thanks to this approach we can make better conclusions that if we just plot with the AUC values and the corresponding number of components, whose figures are in Appendix B.

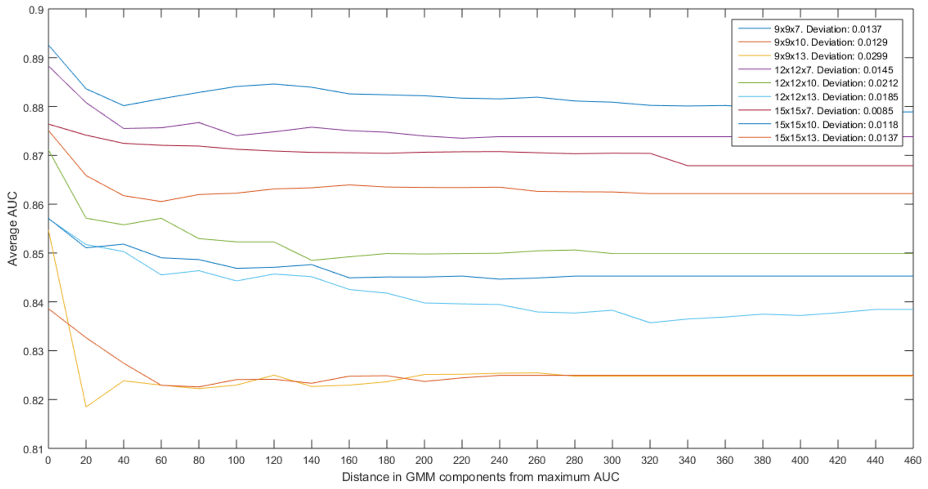


Figure 4.4: AUC average on the neighborhood of the best number of components. UCSD ped1.

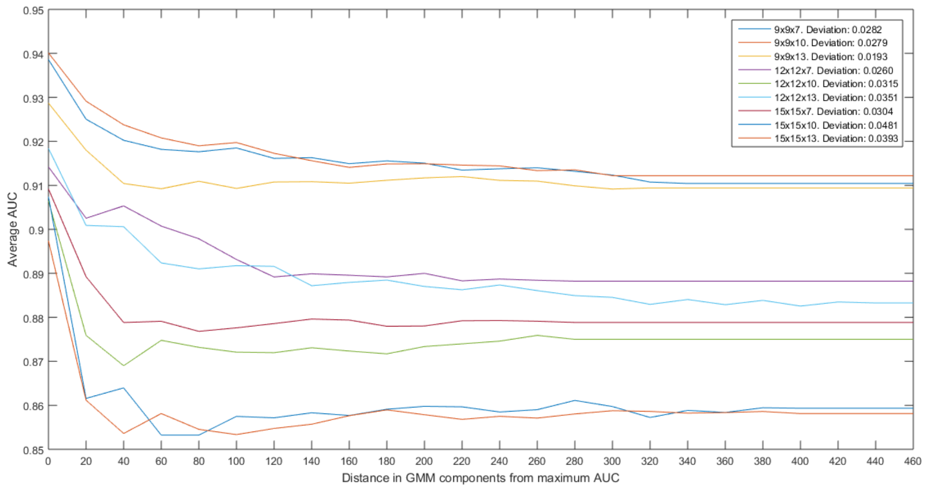


Figure 4.5: AUC average on the neighborhood of the best number of components. UCSD ped2.

Thus, on the horizontal axis we plot the increment in the size of the neighborhood where the average is calculated, against the average value on the vertical axis. These plots can be seen in Figures 4.4 and 4.5. Their utility is to see how the values of AUC variate in average from the maximum. Note that the final value of the functions are the average AUC over all the number of components tried.

In particular, observing Figure 4.4, we can confirm that the best cuboid size is 9x9x7, whose average AUC is always larger than the rest. More interesting is the case of Figure 4.5. Even though the maximum value of AUC is obtained with a size of 9x9x10, the average AUC for the size 9x9x7 is at the same level of 9x9x10. Moreover, a temporal depth of seven frames is better than ten for spatial sizes 12x12 and 15x15 even though the best result is obtained with a temporal size of ten frames. This means that in average, would be preferable to use seven frames.

2.1.2 Subjective analysis

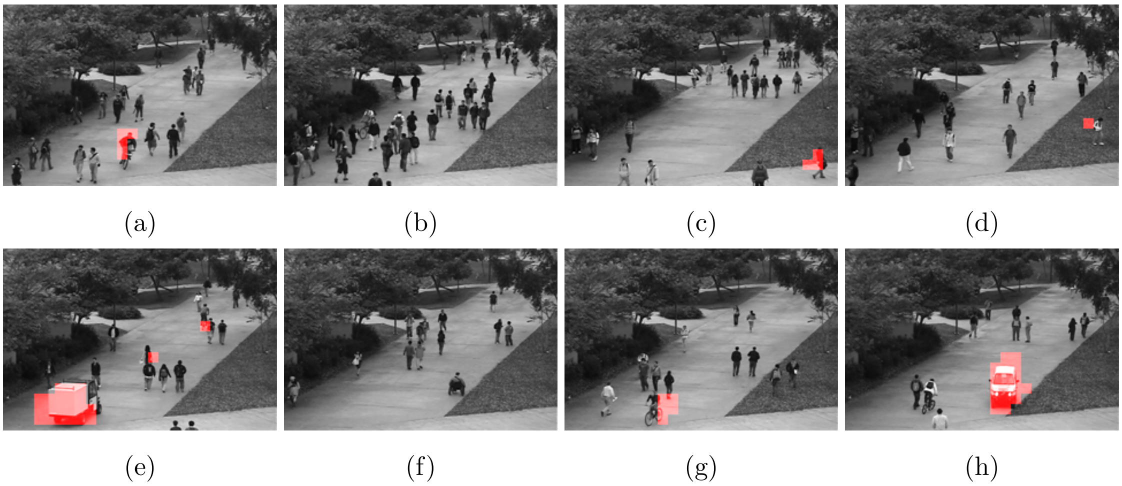

(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 4.6: Examples of anomalies detected with the descriptor D1. UCSD ped1.

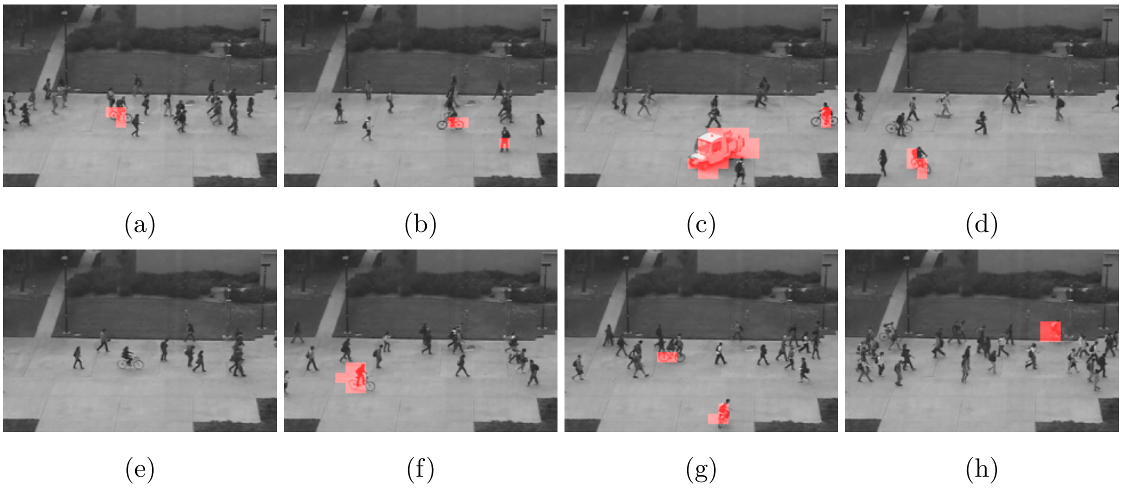

(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 4.7: Examples of anomalies detected with the descriptor D1. UCSD ped2.

It is very difficult to give a visual interpretation of which size is the best based on the precision detecting anomalies. Nevertheless, we can analyze which anomalies are detected or not, to see the weaknesses and capacities of the baseline descriptor and add new features in consequence. Some representative frames, with anomalies detected onto them, are displayed in Figures 4.6 and 4.7.

We have used the best parameters that give the best results on the objective analysis to detect the anomalies. Since the anomalies are detected depending on a threshold, for the visual results these are set manually to avoid many false detections on the images and check the quality of the true detections as we improve the method.

On UCSD ped1, we can see some successful detections on images 4.6a, 4.6c, 4.6g and 4.6h. The detection on image 4.6e is remarkable, since the algorithm is able to detect the three anomalies that appear on scene: the van, the girl with unusual motion pattern and the cyclist. Image 4.6d contains another successful anomaly created by an unusual motion pattern. On the other hand, there are some troubles at the detection in the rest. In particular, on 4.6b the anomaly produced by the cyclist is not detected. Other misdetection exists on image 4.6f, where the person moving on the wheelchair is not detected. These false negatives are produced by the limitation of the descriptor, since its speed and pattern are not anomalous.

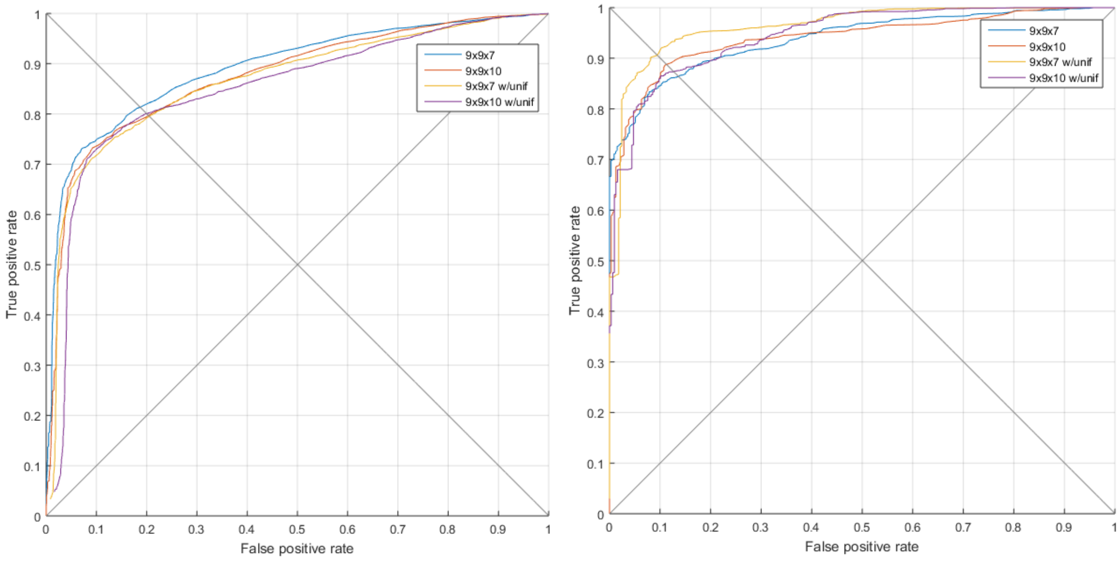
On ped2 dataset, detection on images 4.7c, 4.7f, 4.7g is performed well. However, on image 4.7d, the skater is not detected, as well as the person walking on the side of the bike. There is no detection on image 4.7e, extracted from a sequence where the cyclist is advancing very slowly, because the descriptor is not able to discriminate these anomalies. Finally, during the sequence of image 4.7h, there are difficulties to detect the cyclist on the left, probably for the same reasons commented before.

2.2 Second experiment: uniformity of optical flow

As we concluded at the end of the first experiment, on ped1 dataset we can select a size of 9x9x7 pixels for the cuboids. On ped2 dataset, we can conclude that the spatial size should be 9x9, with a temporal size of 7 or 10 frames.

Thus, the evaluation of the improved descriptor D2 (containing optical flow uniformity as detailed in Section 1.4) will be performed in both sizes (9x9x7 and 9x9x10) for both UCSD ped1 and ped2 datasets. Added to the descriptor so far, three uniformity features are added, with offsets $(1, 1, 0)$, $(3, 3, 0)$ and $(5, 5, 0)$. This is, uniformity is calculated with the same spatial distance for horizontal and vertical axis, always on the same frame.

2.2.1 Objective analysis



(a) (b)

Figure 4.8: Best ROC curves with and without optical flow uniformity. UCSD ped1 and ped2.

In Figures 4.8a and 4.8b, the ROC curves obtained for the best cuboid sizes (9x9x7 and 9x9x10) with the descriptors D1 and D2 (without and with uniformity of optical flow) are represented to see the changes on the detections.

In particular, we can see that on UCSD ped2 dataset the number of right detections have been improved clearly for the cuboid size of 9x9x7 pixels, whose ROC curve has better AUC and EER. For a cuboid size of 9x9x10 pixels, the curve is similar to those obtained before but with a particularity.

Observing the shape of the function for a false positive rate above 0.3, the function goes up above the functions obtained without uniformity, almost converging with the curve for a cuboid size of 9x9x7 pixels. This means that from that point, there are anomalies that were not detected before.

On the other hand, the results on UCSD ped1 are quite different. In fact, we obtain worse results than before, when we applied only optical flow summation and cuboid position. This is a logical behavior if we think that UCSD ped1 and ped2 have different perspectives. Since ped2 does not have any distortion, the computation of the optical flow uniformity helps to discriminate pedestrians from another objects because the size of one object does not change while it is moving through the image and the values of

UCSD ped1 dataset			
Size	AUC	EER (%)	comp
9x9x7	**0.8914**	**18.04**	**430**
9x9x10	0.8751	19.96	410
9x9x7 w/unif	0.8662	19.97	350
9x9x10 w/unif	0.8502	19.48	430

UCSD ped2 dataset			
Size	AUC	EER (%)	comp
9x9x7	0.9386	12.95	210
9x9x10	0.9401	11.04	250
9x9x7 w/unif	**0.9629**	**9.34**	**90**
9x9x10 w/unif	0.9455	11.99	170

Table 4.2: Best AUC and EER results with and without uniformity. UCSD ped1 and ped2.

uniformity are almost the same for the same objects. This is not the case for ped1 dataset, where, because of the perspective distortion, one object does not have the same uniformity value through the scene, giving worse results, as we have seen before.

The effect of the uniformity is more clear if we observe Figures 4.9 and 4.10. If we focus on the case of UCSD ped2, we will see that the best values of AUC are better than before. Moreover, uniformity gives stability to the values of AUC, making the detection less dependent to the number of Gaussian components of the GMM. This stability is almost observed on ped1, where the results extracted with uniformity have a small deviation from the maximum AUC. Nevertheless, the performance is poor as we saw before, being close to the worst results.

It is clear that at this point, it is necessary to apply some kind of perspective normalization to UCSD ped1 dataset, considering than UCSD ped2 has experimented a great improvement on the detection with the same type of descriptor. Thus, the next experiment evaluates if there is any improvement on the detection rate.

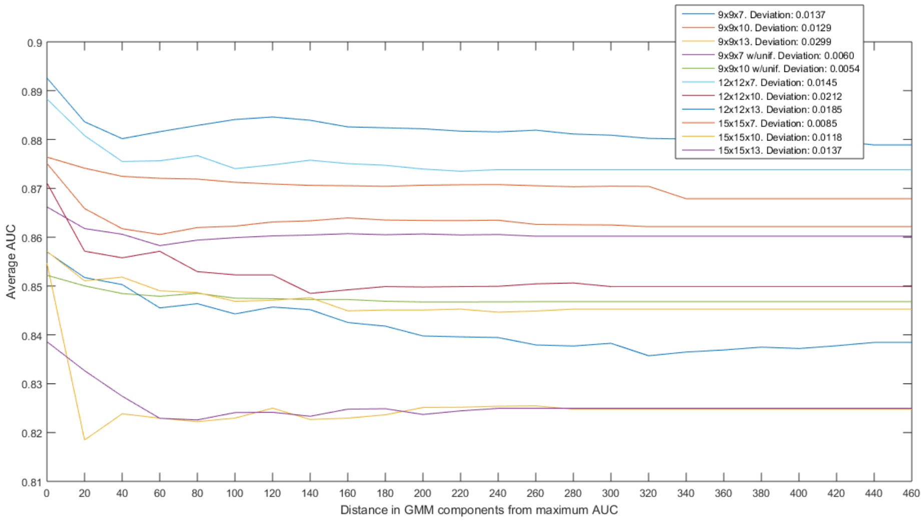


Figure 4.9: AUC average on the neighborhood of the best number of components (with uniformity). UCSD ped1.

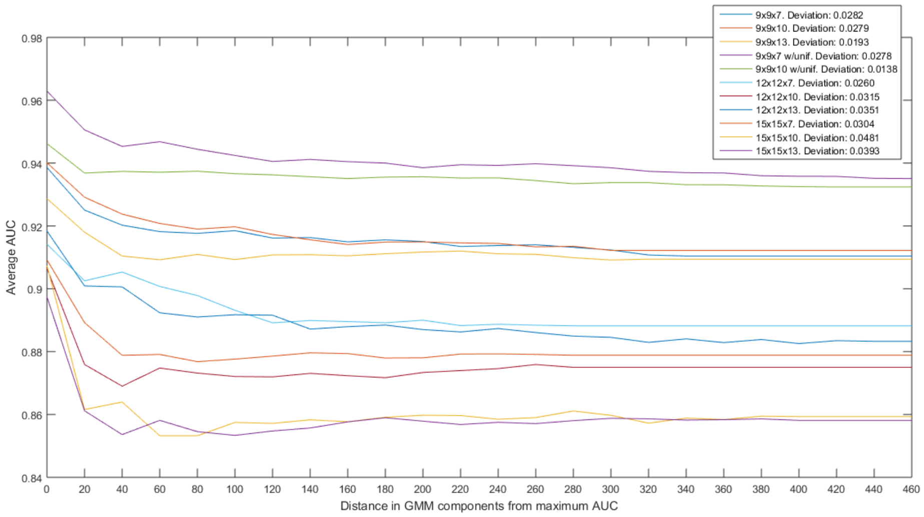
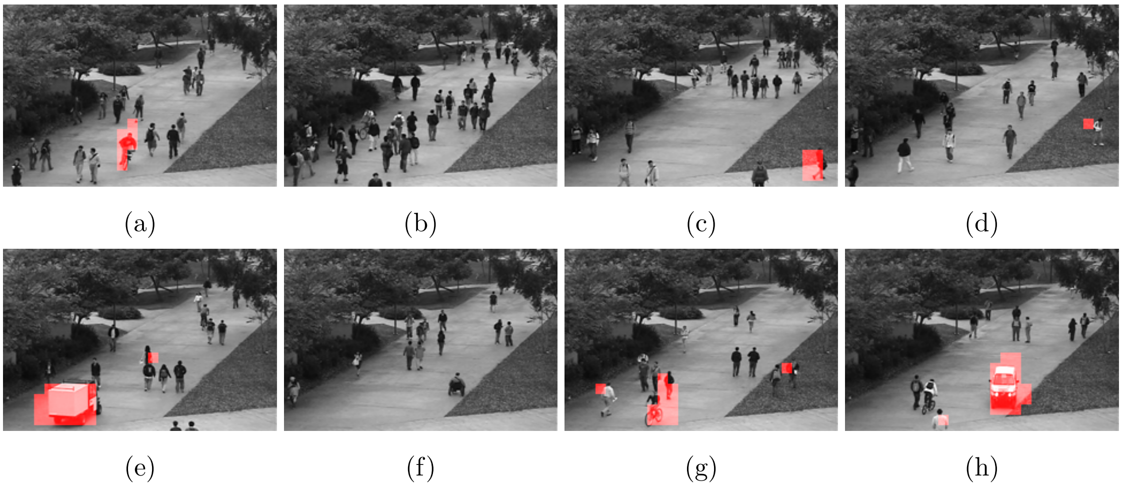


Figure 4.10: AUC average on the neighborhood of the best number of components (with uniformity). UCSD ped2.

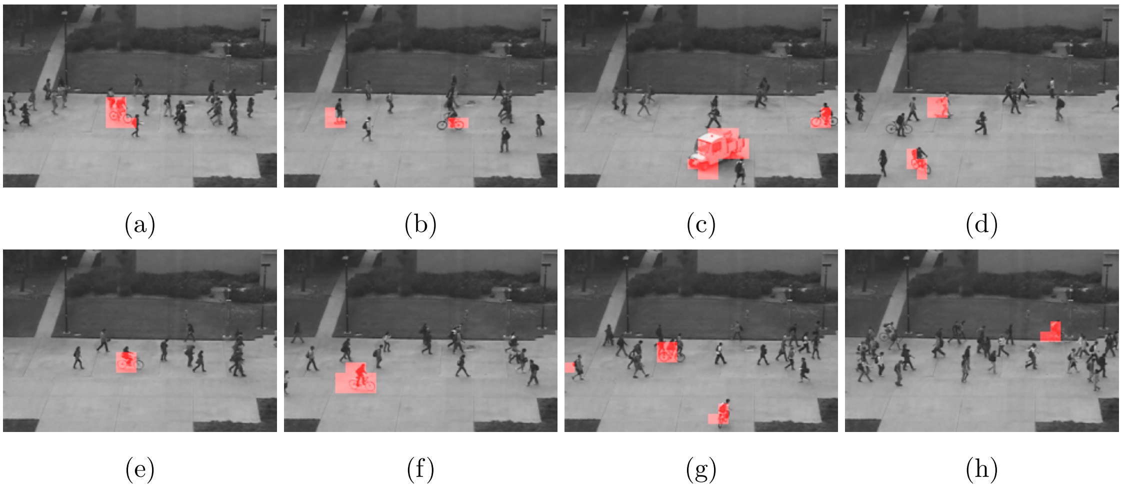
2.2.2 Subjective analysis

With the use of the descriptor D2, the objective is to detect the anomalies that with the baseline descriptor were not detected. These anomalies are produced by another factors different to motion patterns. As we have seen on the objective results, we obtained a great improvement on the UCSD ped2 dataset but not on ped1, where the results are worse.


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

(e) (f) (g) (h)

Figure 4.11: Examples of anomalies detected with the descriptor D2 (optical flow uniformity). UCSD ped1.


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

(e) (f) (g) (h)

Figure 4.12: Examples of anomalies detected with the descriptor D2 (optical flow uniformity). UCSD ped2.

Visually, we can confirm this fact. Generally, the detections are done but there are more false detections as we can observe in Figure 4.11. Specifically, with optical flow uniformity, now it is more difficult to avoid the false detections in the sequence of the image 4.11g. The same occurs in Figure 4.11h.

On the other hand, visual results of the detections on ped2 dataset are significantly better. This is the case of the sequence of the image 4.12e, where now we can detect the slow bicycle that before was not detected. Also, the skaters of the sequence of the image 4.12b and 4.12d are now correctly detected.

2.3 Third experiment: perspective normalization

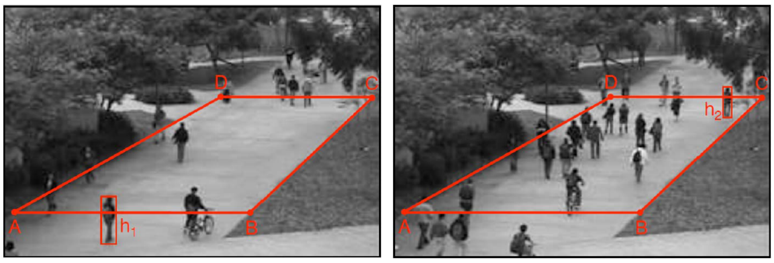


Figure 4.13: Reference width and height of a pedestrian for perspective normalization

The final results of the second experiment are the proof that perspective influences directly the results on the detection of anomalies. This is particularly visible in the sequence with perspective distortion, where the size of the objects of the scene is very different over the image, conditioning the performance of the descriptors that include uniformity of optical flow.

This perspective distortion can be corrected by modifying the optical flow field between images as it is done in [49] and applied in [4]. The modification of the flow is done by scaling the magnitude of the optical flow based on the position y on the optical flow field. The value of this scaling is calculated taking the width and height of a reference pedestrian at the closest point of the image h_1 and w_1 and at the furthest h_2 and w_2 (line AB and CD in Figure 4.13 respectively). The measures in the two points of this reference pedestrian give us the maximum scaling that has to be applied at the top the images.

The scaling $S(y)$ on the values of optical flow varies from the minimum (value of one from the bottom of the image to the line AB of Figure 4.13) and the maximum (from line CD to the top) with the Equation 4.2. In practice, the maximum value of the scaling $S(y)$ has a value of three approximately on ped1 dataset.

$$S(y) = \frac{h_1 w_1}{h(y)w(y)} \tag{4.2}$$

2.3.1 Objective analysis

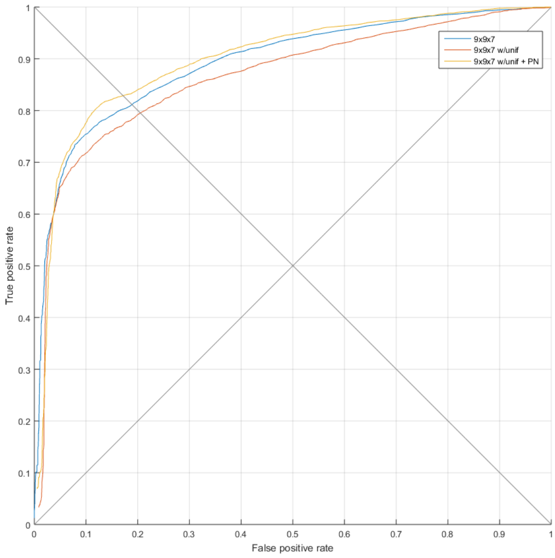


Figure 4.14: Best ROC curves for spatial size 9x9. UCSD ped1.

Size	AUC	EER (%)	comp
9x9x7	0.8914	18.04	430
9x9x7 w/unif	0.8662	19.97	350
9x9x7 w/unif + PN	**0.8977**	**16.38**	**370**

Table 4.3: Best AUC and EER values for a cuboid size of 9x9x7 pixels

For the objective evaluation to compare the results with perspective normalization (PN), we have selected only those results for a cuboid size of 9x9x7, since we have already seen that is the best. With the inclusion of the perspective normalization it occurs something similar comparing the ROC curves with those without uniformity: the improvement in absolute terms is not very high, as we can see in Figure 4.14. AUC and EER values are displayed in Table 4.3.

Nevertheless, observing Figure 4.15, we can see that the average improvement is elevated. We can also observe that with this combination of descriptors and the scaling of the optical flow, the AUC values are very stable as we move away from the best AUC value. Besides, the deviation from the maximum is minimum.

At this point, we can conclude that a good performance on both ped1 and ped2 datasets has been reached. Besides, thanks to objective and subjective analysis, the modification on the algorithm have been done in an incremental manner, trying to solve the weaknesses and advantages in the specific experiments. We have seen that the performance of the baseline method offers reasonable results and, as descriptor and algorithm are improved, the method gains robustness against the number of components selected. On the other hand, it is necessary to be careful with features that change too much on scenes with perspective, as we have observed with the uniformity of optical flow.

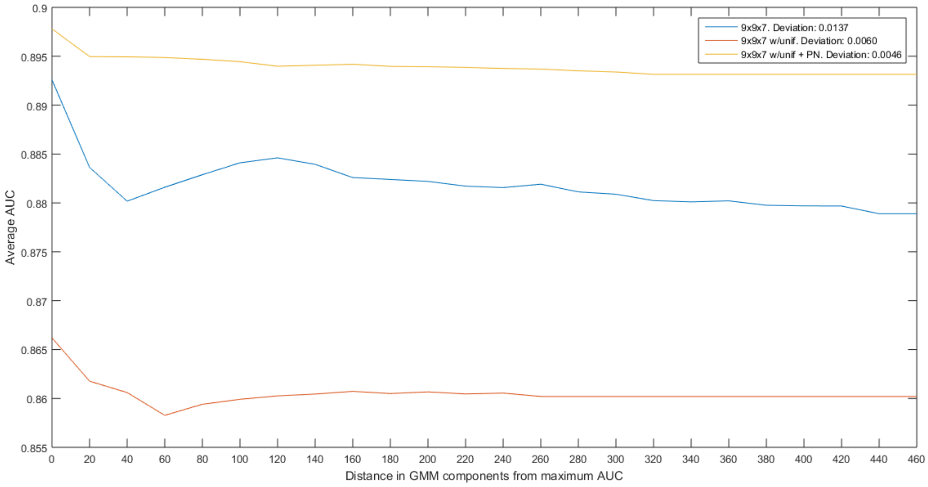
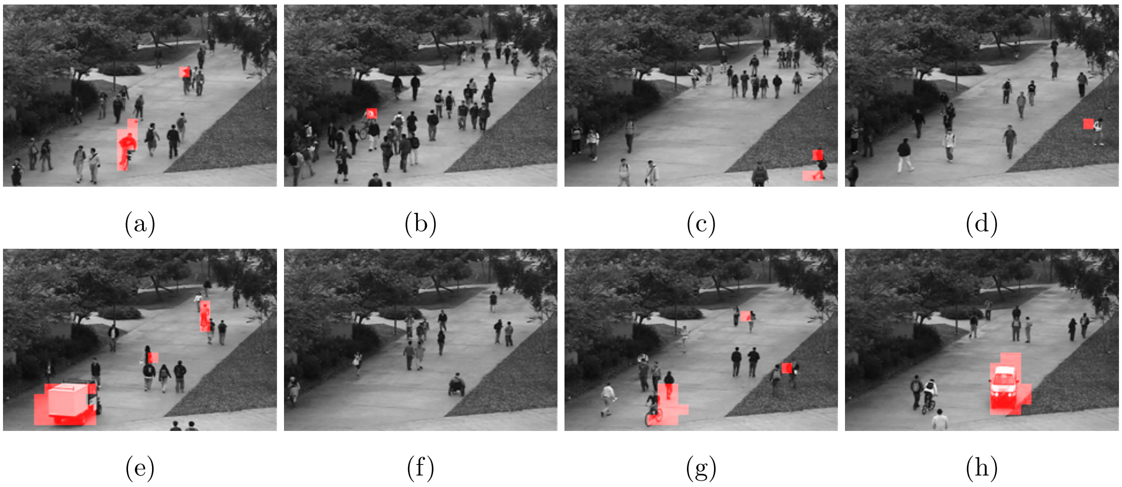


Figure 4.15: AUC average on the neighborhood of the best number of components (with uniformity and perspective normalization). UCSD ped1.

2.3.2 Subjective analysis



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

(e) (f) (g) (h)

Figure 4.16: Examples of anomalies detected with the descriptor D2 and perspective normalization. UCSD ped1.

Figure 4.16 contains some of the images from UCSD ped1 dataset with anomaly detections. The images that were already successful, are still correct. This happens in Figure 4.16a, where now that optical flow vectors at the top of the images are scaled, the results are better. Visually, there are more cuboids covering the real anomalies on the sequences and additionally, some of the anomalies that were not detected at the top of the images, now are correctly detected thanks to the normalization of perspective.

On the other hand, there are still some difficulties when the bicycle of Figure 4.16b appears but now there are more detections during the sequence than before. The results are still poor when the man in the wheelchair appear and the only way to detect it is allowing many false detections.

2.4 Bayesian information criterion (BIC) results

Some methods, such as the proposed in [4] and [3] base their results apparently on the lowest BIC obtained. Nevertheless, in our case, the best results do not coincide with the lowest BIC, as we can see on the figure of Appendix B. This suggest that it is not recommendable to base an automated selection of components via BIC calculation.

Nevertheless, we can confirm that at least, a low value of BIC guarantees that the model has been trained with a minimum number of components since its value is based on the log-likelihood of the data. Added to this, since the values of AUC and EER are not greatly influenced by the number of GMM components, BIC value might be useful to avoid significant computational cost and get reasonable results.

In any case, the values of BIC oscillate too much if we observe Figures in Appendix B, meaning that this value is very sensitive to different number of components. Thus, to find more exhaustive conclusions, it would be necessary to see the values of BIC for all the GMM components tat we have not tried, considering that for now we increase it in a value of twenty between simulations.

2.5 Comparison with state-of-the-art methods

Method	UCSD ped1	UCSD ped2
MDT [1]	17.8	18.5
Roshtkhari et al. [6]	15.1	-
SRC [11]	19	-
Saligrama et al. [25]	16	-
Ryan et al. [3]	23.1	12.7
GMM-MRF [4]	14.9	4.89
Ours	**16.38**	**9.34**

Table 4.4: Comparison of EER (%) values with state-of-the-art methods

Method	UCSD ped1	UCSD ped2
SRC [11]	0.8600	0.8610
Saligrama et al. [25]	0.927	-
Ryan et al. [3]	0.8380	0.9390
GMM-MRF [4]	0.9080	0.9790
Ours	**0.8977**	**0.9629**

Table 4.5: Comparison of AUC values with state-of-the-art methods

For comparison, we have selected some of the most representative methods for anomaly detection: the Mixture of Dynamic Textures (MDT) proposed in [1], textures of optical flow using global GMMs proposed in [3], the GMM-MRF method proposed in [4], the method that uses a codebook of dominant behaviors proposed in [6], the method that uses sparse representations proposed in (SRC) [11] and the method proposed in [25], based in Local Statistics Aggregates.

There are several reasons to include these methods in the comparison. Some of them are frequently cited in the literature, meaning that they are a reference in anomaly detection. Some others, [4] and [3], are not highly referenced but they use GMMs to model the normality of the scenes and have similar features. Besides, they give the results on the evaluation on UCSD dataset. Additionally, some of these methods belong to different of the categories explained in Section 3 so we can compare our method with representative methods of each category of model.

In particular, our method is only behind the method [4] in AUC and EER, as we can see on Tables 4.5 and 4.4 for both ped2 and ped1 and only ped1 in the case of the work from [25]. The differences are not big, specially if we think that descriptors in [4] has twelve features and they use a Markov random field to improve the results. Our method has only seven features.

We find a bigger difference with the method proposed in [3]. They obtain a EER of 23.1 against our 16.38 and in ped1 and 12.7 instead of 9.34 on ped2. With the AUC values occurs the same. Our method obtains a value of 0.8977 on ped1 and 0.9629 on ped2 while they obtain 0.8380 and 0.9390 respectively. The main difference in the method respect to us is that they use another optical flow method and perspective normalization is not performed on ped1 but the descriptors are the same that we use, meaning that the improvement with this two techniques is remarkable.

It is interesting to see that on UCSD we obtain better results than [1] in terms of EER, an important reference in anomaly detection. Another important method is the proposed in [11] and we also obtain better results.

As a conclusion, we can say that our method has potential. We are able to outperform important methods of the literature by using GMMs, which are not as common as another techniques. We can say also that we have outperformed the method in which we are based, almost reaching the level of other more sophisticated methods that use GMMs, as is the case of [4].

Chapter 5

Conclusions and future work

Given the amount of methods to implement an anomaly detection algorithm on video-surveillance scenarios, it is not easy to choose the optimal combination of models, techniques and features for one specific context in particular. However, for better understanding, we have tried to give a general panorama of the principal methods for anomaly detection by separating the state-of-the-art methods by type of anomaly, learning, models and feature extraction strategies. This process has been essential to identify those approaches that better fit with our purpose, that mostly comprehend pedestrian scenarios.

After this exhaustive study, we concluded that the optical flow features could be a reasonable and detailed source of information, from which multiple features can be extracted. In fact, a major part of the methods work under this idea. Nevertheless, we saw potential into changing the technique to extract the optical flow field between frames since the two most frequent methods on anomaly detection are inaccurate. The method we use has proved that the improvement on the detection of anomalies is not negligible.

Through the information we are able to extract from the optical flow fields, we have managed to build discriminative descriptors from baseline features extracted from spatio-temporal volumes, composed by their coordinates and their motion magnitude and orientation. With only these four features, the detection offers good results, improved by the use of uniformity of the new optical flow method and normalization of perspective.

These descriptors feed a global Gaussian mixture model (GMM), that is able to fashion the normality of the scene, including spatial notion of the events thanks to coordinates of the spatio-temporal cuboids included in the descriptors. This is not a common solution in the literature, but its conceptual simplicity and its promising results makes the GMM a relevant alternative for surveillance scenarios, thanks to the possibility to work without modeling the anomalies that might appear on scene.

On the other hand, the use of a global GMM implies that the best number of Gaussian components to model the normality has to be found making the evaluation by increasing its number. In any case, with the proper pre-processing and descriptors, it is possible to make the algorithm more robust to the number of components, relaxing the search.

Having said this, there is much room for improvements. The most logical step is to improve the descriptors, that could be as complete as we want by adding new features. In our case, we have seen that by using only optical flow features it is possible to build a complete anomaly detection algorithm but as commented in the study of the state of the art, there are plenty of features likely to improve the results.

Adding new features to the descriptors would increase the computational cost of the system, specially in ours, that we have to deal with hundreds of Gaussian components. For these reasons, it would be interesting to use local GMMs for each spatio-temporal cuboids, whose computation could be parallelized. This would reduce significantly the needed number of components for the models, although it would be necessary to manage each mixture independently, making the search of the detection threshold more difficult.

In the future, would be interesting to make further research about Bayesian Information Criterion by analyzing its values more exhaustively by increasing the number of Gaussian components in a value of one and conclude if it is a good method to find the optimal number of components.

Another possibility to improve the results, would be to add notion about information of neighboring cuboids on both training and test phases. By doing that, it would be possible to know not only the state of the actual cuboid but also the state of the nearby cuboids. This data would be taken into account in the detection phase, making the detection of anomalies more accurate. In terms of evaluation, the most widely used available dataset has been used. However, some other datasets can be utilized.

Implementation can also be improved. The construction of model for different number of components is already implemented in parallel. Nevertheless, the method does not work yet on real time. By making the code more efficient and using GPU to process the optical flow, for instance, the processing time could be reduced.

Bibliography

[1] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(1):18–32, 2014.

[2] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, volume 249, page 250, 2010.

[3] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Textures of optical flow for real-time anomaly detection in crowds. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 230–235. IEEE, 2011.

[4] Hajananth Nallaivarothayan, Clinton Fookes, Simon Denman, and Sridha Sridharan. An mrf based abnormal event detection approach using motion and appearance features. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 343–348. IEEE, 2014.

[5] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1446–1453. IEEE, 2009.

[6] Mehrsan Roshtkhari and Martin Levine. Online dominant and anomalous behavior detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2611–2618, 2013.

[7] Imran Saleemi, Lance Hartung, and Mubarak Shah. Scene understanding by statistical modeling of motion patterns. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2069–2076. IEEE, 2010.

[8] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[9] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[10] Oluwatoyin P Popoola and Kejun Wang. Video-based abnormal human behavior recognition—a review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):865–878, 2012.

[11] Yang Cong, Junsong Yuan, and Ji Liu. Abnormal event detection in crowded scenes using sparse representation. *Pattern Recognition*, 46(7):1851–1864, 2013.

[12] Angela A Sodemann, Matthew P Ross, and Brett J Borghetti. A review of anomaly detection in automated surveillance. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1257–1272, 2012.

[13] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. Crowded scene analysis: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(3):367–386, 2015.

[14] Shandong Wu, Brian E Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060. IEEE, 2010.

[15] Ramin Mehran, Brian E Moore, and Mubarak Shah. A streakline representation of flow in crowded scenes. In *Computer Vision–ECCV 2010*, pages 439–452. Springer, 2010.

[16] Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee. Robust fall detection using human shape and multi-class support vector machine. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 413–420. IEEE, 2008.

[17] Vikas Reddy, Conrad Sanderson, and Brian C Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 55–61. IEEE, 2011.

[18] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda. A suspicious behaviour detection using a context space model for smart surveillance systems. *Computer Vision and Image Understanding*, 116(2):194–209, 2012.

[19] Brendan Tran Morris and Mohan Manubhai Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2287–2301, 2011.

[20] Fan Jiang, Ying Wu, and Aggelos K Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 18(4):907–913, 2009.

[21] Frederick Tung, John S Zelek, and David A Clausi. Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. *Image and Vision Computing*, 29(4):230–240, 2011.

[22] Rikard Laxhammar and Göran Falkman. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[23] Tianzhu Zhang, Si Liu, Changsheng Xu, and Hanqing Lu. Mining semantic context information for intelligent video surveillance of traffic scenes. *Industrial Informatics, IEEE Transactions on*, 9(1):149–160, 2013.

[24] Timothy M Hospedales, Jian Li, Shaogang Gong, and Tao Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2451–2464, 2011.

[25] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2112–2119. IEEE, 2012.

[26] Bin Zhao, Li Fei-Fei, and Eric P Xing. Online detection of unusual events in videos via dynamic sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3313–3320. IEEE, 2011.

[27] Yang Hu, Yangmuzi Zhang, and Larry Davis. Unsupervised abnormal crowd activity detection using semiparametric scan statistic. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 767–774, 2013.

[28] Ramin Mehran, Akira Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009.

[29] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.

[30] Kyungseo Park, Yong Lin, Vangelis Metsis, Zhengyi Le, and Fillia Makedon. Abnormal human behavioral pattern detection in assisted living environments. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, page 9. ACM, 2010.

[31] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001.

[32] Iulian Pruteanu-Malinici and Lawrence Carin. Infinite hidden markov models for unusual-event detection in video. *Image Processing, IEEE Transactions on*, 17(5):811–822, 2008.

[33] Xiaogang Wang, Xiaoxu Ma, and W Eric L Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):539–555, 2009.

[34] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3):299–318, 2008.

[35] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[36] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[37] Antoni B Chan and Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):909–926, 2008.

[38] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[39] Jingxin Xu, Simon Denman, Sridha Sridharan, Clinton Fookes, and Rajib Rana. Dynamic texture reconstruction from sparse codes for unusual event detection in crowded scenes. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, pages 25–30. ACM, 2011.

[40] Jie Feng, Chao Zhang, and Pengwei Hao. Online learning with self-organizing maps for anomaly detection in crowd scenes. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3599–3602. IEEE, 2010.

[41] Myo Thida, How-Lung Eng, and Paolo Remagnino. Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes. *IEEE transactions on cybernetics*, 43(6):2147–2156, 2013.

[42] Javier Albusac, David Vallejo, Luis Jimenez-Linares, Jose Jesus Castro-Schez, and Luis Rodriguez-Benitez. Intelligent surveillance based on normality analysis to detect abnormal behaviors. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1223–1244, 2009.

[43] Duarte Duque, Henrique Santos, and Paulo Cortez. Prediction of abnormal behaviors for intelligent video surveillance systems. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 362–367. IEEE, 2007.

[44] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[45] authors of VLFeat. GMM Fundamentals. `http://www.vlfeat.org/api/gmm-fundamentals.html`, 2013.

[46] Frank Hansen and Gert K Pedersen. Jensen's operator inequality. *Bulletin of the London Mathematical Society*, 35(4):553–564, 2003.

[47] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[48] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[49] Hajananth Nallaivarothayan, David Ryan, Simon Denman, Sridha Sridharan, and Clinton Fookes. An evaluation of different features and learning models for anomalous event detection. In *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[50] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[51] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

Appendix A: Robust optical flow

$$AEPE := \frac{1}{N}\sum_{i=0}^{N}(u_i - \tilde{u}_i)^2 + (v_i - \tilde{v}_i)^2 \tag{A.1}$$

$$AAE := \frac{1}{N}\sum_{i=1}^{N}\arccos\left(\frac{u_i\tilde{u}_i + v_i\tilde{v}_i}{(u_i^2 + v_i^2 + 1)^{1/2}(\tilde{u}_i^2 + \tilde{v}_i^2 + 1)^{1/2}}\right) \tag{A.2}$$

The method proposed in [9] apply some important functions thanks to that the accuracy of the final flow field is higher than the original methods from Horn-Schunck or Black-Anandan. To evaluate how better this method is, authors checked its results on the training and test sets of the Middlebury optical flow benchmark [50]. The measures used for this purpose are the average end-point error (AEPE) and the average angular error (AAE), whose equations are presented in Equations A.1 and A.2.

1 Techniques used

Before introducing the techniques proposed in [9], let us remember the classical objective function of the optical flow in its discrete form:

$$\begin{split}E(u,v) = \sum_{i,j}\{\rho_D(I_1(i,j) - I_2(i + u_{i,j}, j + v_{i,j}))+ \\ \lambda[\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})]\}\end{split} \tag{A.3}$$

The idea is to minimize this function, where u and v are the vertical and horizontal components of the optical flow field estimated given the images I_1 and I_2; ρ_D refers to the penalty function of the data term; ρ_S is the penalty of the spatial term and λ is a regularization parameter.

The list of the main techniques employed in [9] to get an improved optical flow field are the following:

- **Structure and texture decomposition**: A method called Rudin-Osher-Fatemi proposed in [44] to separate the structure and texture of the images to gain robustness against lighting changes.
- **Multi-resolution**: A multi-resolution pyramid is created so the optical flow estimated at a coarse level is used to warp the second image toward the first at the next finer level. The flow increment is calculated between the first image and the warped second image.
- **Derivative filter**: Instead of using the differences between images for the data term of the optical flow objective function, a derivative filter $1/12\,[-1\ 8\ 0\ -8\ 1]$ is used to find the changes in the directions of x and y and then warp the result over the first image using the current flow estimation by means of bicubic interpolation.
- **Weighted median filter**: Used after the warping steps to remove outliers and avoid errors before the next iteration of the optical flow estimation. There are spatial and color weights, increasing the accuracy and the energy of the final field. This also gets sharper fields around the objects.
- **Graduated non-convexity (GNC)**: The original method from Horn and Schunck makes only use of a quadratic penalty function $\rho(x) = x^2$ as spatial penalty. However, it is not robust to outliers, making the algorithm inaccurate, as explained in [9]. To mitigate this problem, different penalty functions have been used in the literature. For example, the Lorentzian penalty $\rho(x) = \log(1 + x^2/2\sigma^2)$ proposed by Black and Anandan [51] and the Charbonnier penalty $\rho(x) = (x^2 + \epsilon^2)^a$, which has different shapes depending on the value of a, as we can see in Figure A.1.

  Under the GNC strategy used in [9], at first a simple penalty function (convex) is used and then, a more robust penalty function (non-convex) is applied. Thus, there is a first GNC iteration where the estimation of the flow is done over the entire pyramid using the quadratic penalty function. The second GNC iteration starts from the last estimation of the flow obtained to again calculate the flow, but now with the Charbonnier penalty function, which is non-convex when $a < 0.5$. The best accuracy on the flow is obtained with $a = 0.45$.

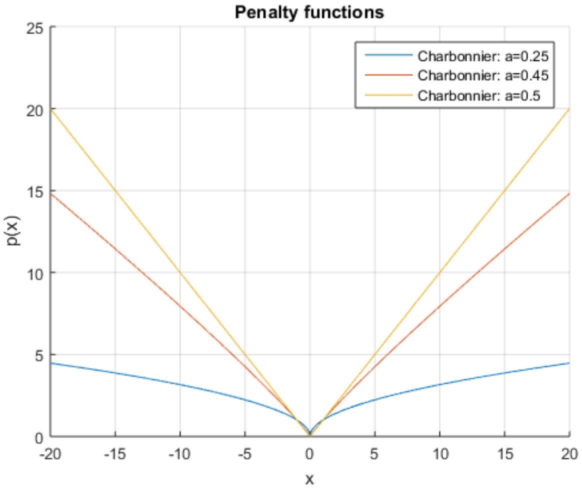


Figure A.1: Charbonnier penalty function for different values of a

For our purpose, two modifications have been done on the parameters and techniques used to lighten the computational cost of the algorithm. Firstly, we saw that the technique to get robustness against illumination changes did not changed too much the accuracy of the final flow field. However, the computational cost was remarkable so we do not apply it. The method proposed in [9] uses three iterations to compute the optical flow for each level of the pyramid. We use only one. The rest techniques remain the same.

Since the Middlebury optical flow benchmark has one public set of images with ground-truths, we have decided to test our implementation of the method and compare the results in terms of visual results (Figure A.2), AAE and AEPE with the most common optical flow methods for anomaly detection: Horn-Schunk and Black-Anandan. Thus, even though we have replicated the code from the original method, we want to check is the behavior is the expected and the results are better than the aforementioned methods.

We have used eight pair of images in total, all in gray scale. In Figure A.2 we can see seven of the eight images used to compare the results of the optical flow. The first thing we see is that we can confirm the problems of Horn-Schunk (HS) method associated with the smoothness constraint used. In general, the results obtained with this methods are blurry.

The results with the method proposed by Black-Anandan (BA) are better in this respect, mainly because the penalty function is not quadratic. Still, the objects are not as sharper as with the method proposed in [9] with our modifications. This is more visible in the fifth row, where the objects at the bottom are almost as defined as in the ground-truth. This fact is also visible at the borders of the building of the sixth row.

Another remarkable flow field is the one in the second row, where the flows obtained with HS and BA appears darker because there is an error in at least one pixel, whose motion magnitude is very high respect to the others. Thus, when the plot is done, the image is normalized using the brightest pixel and the image is darker. The result with the method from [9] does not have this problem. On the other hand, there is less accuracy in terms of orientation in homogeneous areas, as we can see in the first and seventh rows for example.

AAE and AEPE results are given in Table A.1. As we can see, the best technique in terms of AAE is Black-Anandan and in terms of AEPE the best is the one we use based on the method proposed in [9]. In any case, for our purpose, thanks to the techniques that this method integrates objects appear sharper and with less outliers as we described in the last paragraphs.

Measure	Horn-Schunck	Black-Anandan	method in [9]
AAE	5.8326	4.5894	5.03
AEPE	0.4683	0.3571	0.3503

Table A.1: Average end-point and angular error for different optical flow methods of Middlebury dataset

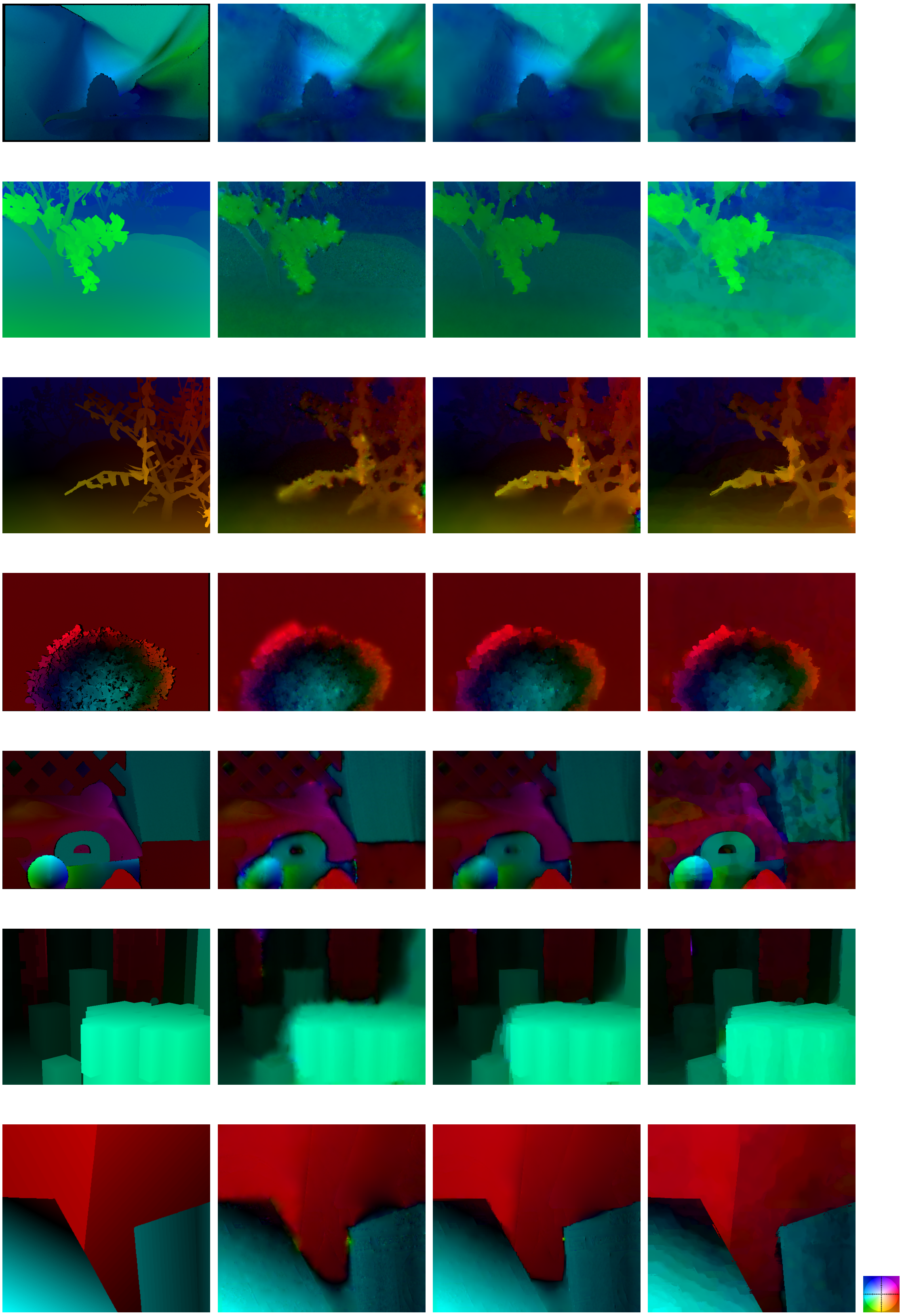


Figure A.2: Optical flow comparison on Middlebury dataset. From left to right: Ground-truth, Horn-Schunck, Black-Anandan, method proposed in [9].

Appendix B: Tables of results

UCSD ped1

Spatial size: 9x9

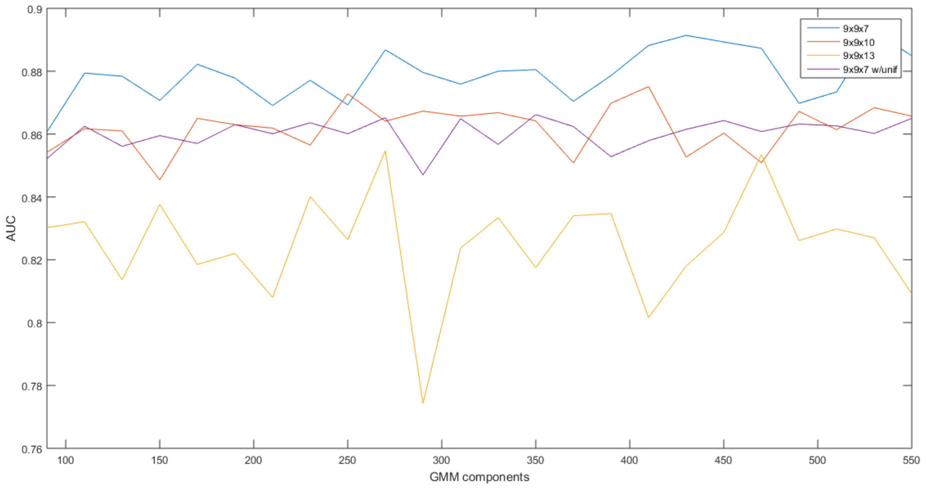


Figure B.1: AUC values with spatial size of 9x9. UCSD ped1.

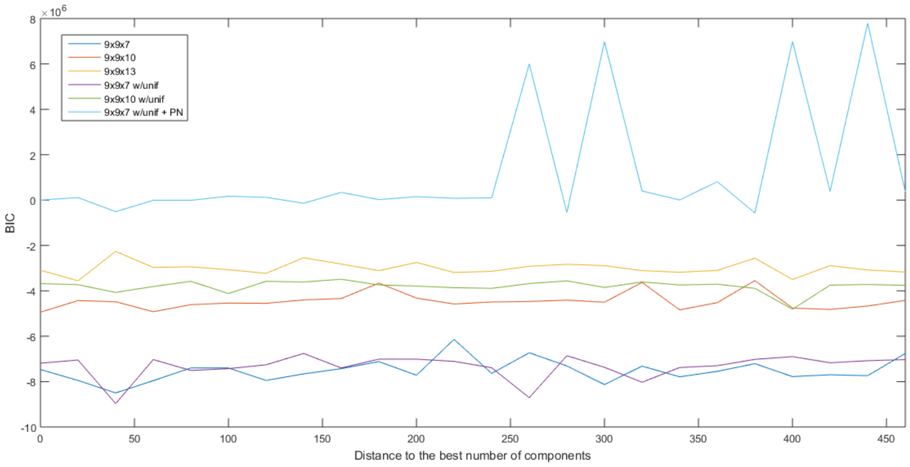


Figure B.2: BIC values with spatial size of 9x9. UCSD ped1.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8607	20.80	-6.76
110	0.8794	18.87	-6.14
130	0.8784	19.04	-6.73
150	0.8707	20.00	-7.55
170	0.8822	19.27	-7.66
190	0.8778	19.37	-7.31
210	0.8691	21.43	-7.74
230	0.8771	20.13	-8.13
250	0.8693	20.17	-7.699
270	0.8868	18.51	-7.4
290	0.8796	19.77	-7.72
310	0.8759	19.04	-7.32
330	0.8800	18.57	-7.12
350	0.8805	20.10	-7.43
370	0.8704	20.27	-7.21
390	0.8786	20.00	-7.64
410	0.8882	19.34	-7.96
430	**0.8914**	**18.04**	**-7.95**
450	0.8893	18.84	-8.50
470	0.8873	18.84	-7.40
490	0.8698	19.54	-7.78
510	0.8734	20.10	-7.79
530	**0.8926**	**18.61**	**-7.47**
550	0.8849	19.30	-7.95

Table B.1: Results for UCSD ped1. Cuboid size: 9x9x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8542	22.03	-3.55
110	0.8617	20.57	-4.41
130	0.8610	20.54	-3.63
150	0.8454	22.94	-4.42
170	0.8650	20.60	-3.66
190	0.8630	22.09	-4.49
210	0.8619	20.30	-4.47
230	0.8565	23.01	-4.52
250	**0.8728**	**20.33**	**-4.43**
270	0.8641	20.50	-4.58
290	0.8673	20.74	-4.61
310	0.8657	21.21	-4.40
330	0.8668	21.01	-4.55
350	0.8642	21.04	-4.32
370	0.8508	20.91	-4.67
390	0.8698	21.42	-4.48
410	**0.8751**	**19.96**	**-4.94**
430	0.8527	22.40	-4.76
450	0.8603	22.40	-4.84
470	0.8509	22.70	-4.82
490	0.8672	21.08	-4.54
510	0.8614	21.79	-4.50
530	0.8684	20.94	-4.92
550	0.8657	21.08	-4.34

Table B.2: Results for UCSD ped1. Cuboid size: 9x9x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8302	24.56	-2.82
110	0.8321	23.87	-2.54
130	0.8136	25.74	-2.56
150	0.8376	24.42	-2.97
170	0.8185	25.63	-3.11
190	0.8220	24.70	-2.89
210	0.8080	25.42	-2.89
230	0.8401	23.90	-2.26
250	0.8264	25.25	-3.14
270	**0.8547**	**22.28**	**-3.09**
290	0.7743	31.30	-3.17
310	0.8237	24.84	-2.83
330	0.8334	23.56	-3.23
350	0.8175	25.29	-3.10
370	0.8340	24.97	-3.07
390	0.8347	25.49	-2.94
410	0.8016	26.91	-3.08
430	0.8180	25.77	-3.18
450	0.8287	25.42	-2.75
470	**0.8535**	**22.79**	**-3.56**
490	0.8261	26.12	-2.92
510	0.8298	25.22	-3.11
530	0.8270	24.28	-3.19
550	0.8092	26.12	-3.50

Table B.3: Results for UCSD ped1. Cuboid size: 9x9x13. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8522	22.29	-7.08
110	0.8625	20.46	-7.01
130	0.8561	21.26	-6.90
150	0.8595	20.03	-8.03
170	0.8570	20.50	-7.30
190	0.8630	20.76	-6.76
210	0.8601	20.10	-6.86
230	0.8636	19.14	-7.43
250	0.8601	20.40	-7.37
270	0.8652	19.90	-7.05
290	0.8470	19.77	-7.03
310	0.8649	19.54	-7.03
330	0.8567	20.60	-7.02
350	**0.8662**	**19.97**	**-7.19**
370	0.8624	19.70	-7.01
390	0.8528	20.43	-7.17
410	0.8579	20.86	-7.38
430	0.8615	20.03	-7.11
450	0.8643	19.40	-7.51
470	0.8608	18.91	-7.39
490	0.8632	19.14	-7.26
510	0.8626	19.70	-7.40
530	0.8602	19.67	-8.71
550	0.8650	19.70	-8.97

Table B.4: Results for UCSD ped1. Cuboid size: 9x9x7. Feature: cuboid position, optical flow summation and uniformity

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8477	21.18	-3.61
110	0.8470	21.25	-3.68
130	0.8469	22.40	-3.56
150	0.8492	21.21	-3.58
170	0.8405	20.77	-3.72
190	0.8463	21.45	-3.61
210	0.8456	21.01	-3.71
230	0.8479	21.52	-3.58
250	0.8476	20.77	-3.49
270	0.8512	20.81	-3.73
290	0.8474	21.72	-3.79
310	0.8447	21.59	-4.81
330	0.8473	21.04	-3.86
350	**0.8522**	**20.47**	**-3.68**
370	0.8506	20.64	-4.07
390	0.8475	21.59	-3.74
410	0.8456	21.55	-3.89
430	**0.8502**	**19.48**	**-3.81**
450	0.8383	21.08	-3.76
470	0.8458	22.09	-3.74
490	0.8466	21.25	-3.85
510	0.8483	21.15	-4.12
530	0.8472	20.81	-3.89
550	0.8415	22.09	-3.75

Table B.5: Results for UCSD ped1. Cuboid size: 9x9x10. Feature: cuboid position, optical flow summation and uniformity

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8876	17.65	+0.37
110	0.8908	17.08	-0.57
130	0.8897	16.92	+0.38
150	0.8925	18.04	+6.98
170	0.8917	17.48	+0.81
190	0.8925	17.58	+0.40
210	0.8934	17.71	+0.08
230	0.8905	17.58	+6.99
250	0.8957	16.95	-0.51
270	0.8946	17.08	+0.12
290	0.8887	17.78	+7.80
310	0.8941	16.62	+0.02
330	0.8932	16.65	+0.1
350	0.8943	17.68	-0.14
370	**0.8977**	**16.38**	**+0.11**
390	0.8928	18.08	+6.00
410	**0.8978**	**16.58**	**-0.001**
430	0.8943	17.91	+0.34
450	0.8921	17.58	+0.005
470	0.8951	16.58	-0.01
490	0.8949	17.05	-0.01
510	0.8926	16.75	-0.55
530	0.8941	17.65	+0.15
550	0.8949	16.82	+0.17

Table B.6: Results for UCSD ped1. Cuboid size: 9x9x7. Feature: cuboid position, optical flow summation, uniformity and perspective normalization.

Spatial size: 12x12

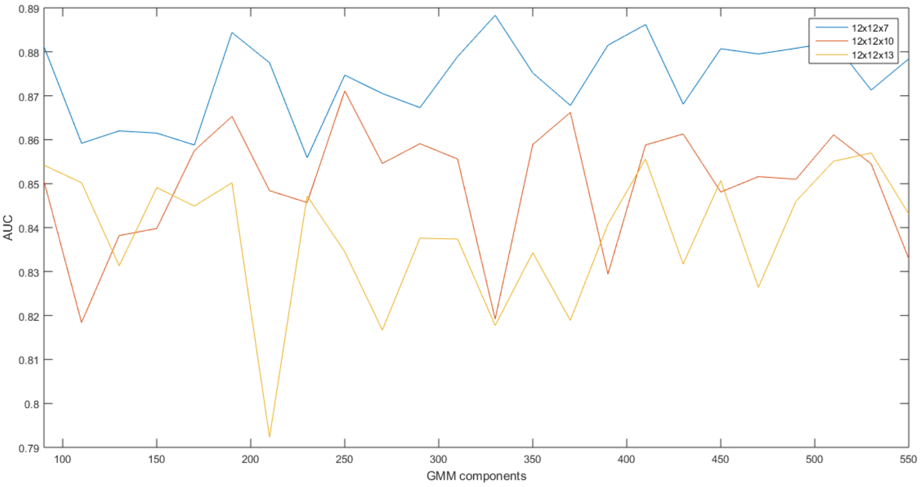


Figure B.3: AUC values with spatial size of 12x12. UCSD ped1.

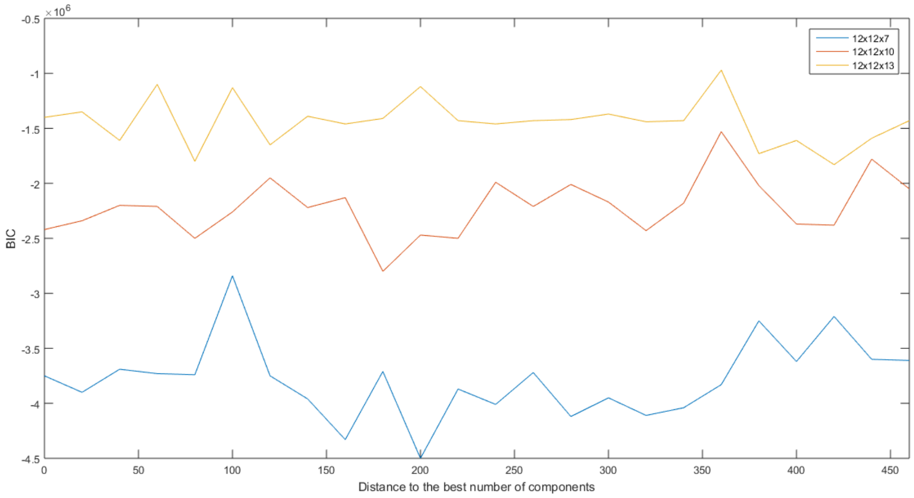


Figure B.4: BIC values with spatial size of 12x12. UCSD ped1.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8812	19.73	-2.84
110	0.8592	21.82	-3.21
130	0.8620	20.93	-3.25
150	0.8615	22.12	-3.62
170	0.8588	21.16	-3.60
190	**0.8844**	**19.70**	**-3.69**
210	0.8775	20.13	-3.87
230	0.8559	21.69	-3.61
250	0.8747	20.20	-3.72
270	0.8705	20.46	-3.95
290	0.8673	21.92	-3.83
310	0.8789	19.93	-3.71
330	**0.8883**	**20.76**	**-3.75**
350	0.8752	20.30	-4.01
370	0.8678	21.26	-4.04
390	0.8815	20.10	-3.74
410	0.8862	20.27	-3.90
430	0.8681	21.19	-4.11
450	0.8807	21.13	-3.96
470	0.8795	19.57	-4.33
490	0.8808	19.90	-3.75
510	**0.8822**	**18.81**	**-3.73**
530	0.8713	20.83	-4.12
550	0.8784	19.04	-4.50

Table B.7: Results for UCSD ped1. Cuboid size: 12x12x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8506	21.55	-2.01
110	0.8184	25.11	-2.05
130	0.8382	23.55	-2.02
150	0.8398	22.91	-1.53
170	0.8575	23.21	-2.13
190	0.8653	21.45	-2.20
210	0.8484	22.81	-2.17
230	0.8457	22.30	-2.18
250	**0.8711**	**21.35**	**-2.42**
270	0.8546	21.92	-2.47
290	0.8591	21.96	-2.26
310	0.8556	22.23	-2.80
330	0.8192	25.11	-1.78
350	0.8589	22.81	-1.95
370	0.8662	21.45	-2.34
390	0.8294	25.31	-2.38
410	0.8588	22.09	-2.22
430	0.8613	21.62	-2.21
450	0.8481	23.25	-2.43
470	0.8516	22.40	-1.99
490	0.8510	23.35	-2.21
510	0.8611	21.55	-2.50
530	0.8545	23.31	-2.50
550	0.8329	24.43	-2.37

Table B.8: Results for UCSD ped1. Cuboid size: 12x12x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	**0.8542**	**21.69**	**-1.10**
110	0.8502	22.13	-1.13
130	0.8313	23.89	-0.97
150	0.8491	22.37	-1.39
170	0.8449	22.74	-1.12
190	0.8502	22.84	-1.65
210	0.7923	27.04	-1.43
230	0.8472	23.42	-1.46
250	0.8345	24.01	-1.37
270	0.8167	25.42	-1.59
290	0.8376	24.18	-1.43
310	0.8374	24.25	-1.42
330	0.8177	24.56	-1.83
350	0.8343	24.66	-1.44
370	0.8189	25.94	-1.61
390	0.8408	23.45	-1.46
410	0.8556	22.67	-1.35
430	0.8317	24.06	-1.43
450	0.8507	22.94	-1.80
470	0.8264	24.36	-1.73
490	0.8460	22.57	-1.41
510	0.8551	22.06	-1.61
530	**0.8570**	**22.26**	**-1.40**
550	0.8431	22.40	-1.43

Table B.9: Results for UCSD ped1. Cuboid size: 12x12x13. Feature: cuboid position, optical flow summation.

Spatial size: 15x15

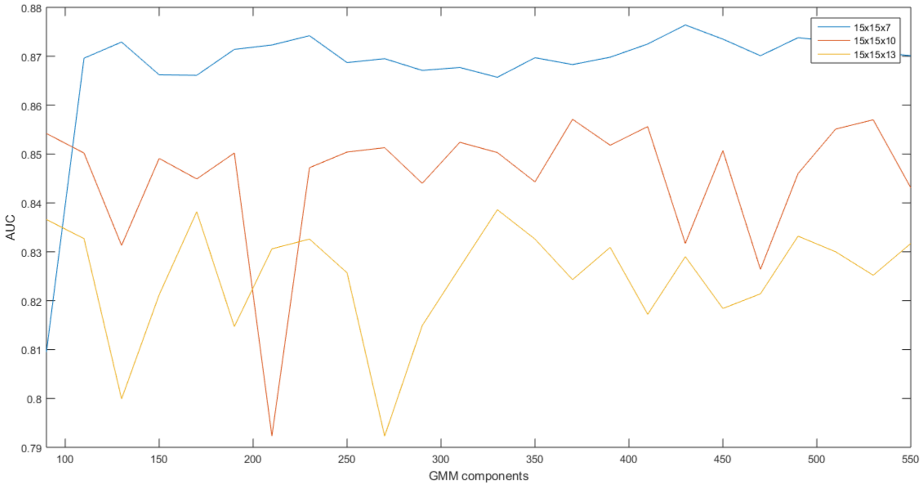


Figure B.5: AUC values with spatial size of 15x15. UCSD ped1.

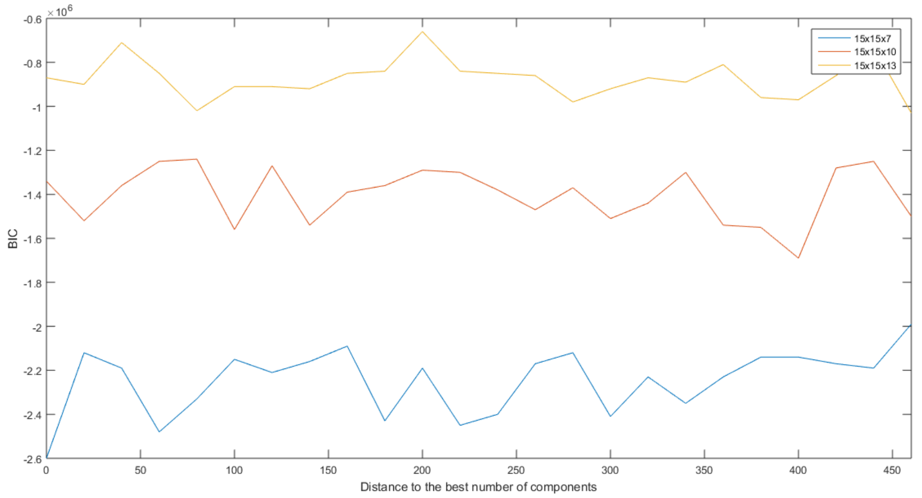


Figure B.6: BIC values with spatial size of 15x15. UCSD ped1.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8095	26.70	-1.99
110	0.8696	21.19	-2.12
130	0.8729	21.86	-2.15
150	0.8662	21.82	-2.14
170	0.8661	21.99	-2.17
190	0.8714	21.72	-2.09
210	0.8723	21.09	-2.16
230	**0.8742**	**19.90**	**-2.12**
250	0.8687	21.72	-2.23
270	0.8695	21.23	-2.41
290	0.8671	20.70	-2.14
310	0.8677	21.69	-2.23
330	0.8657	21.26	-2.19
350	0.8697	21.59	-2.17
370	0.8683	21.96	-2.35
390	0.8698	21.16	-2.40
410	0.8725	21.49	-2.21
430	**0.8764**	**21.59**	**-2.60**
450	0.8735	20.83	-2.48
470	0.8701	20.73	-2.19
490	0.8738	19.93	-2.19
510	0.8730	20.23	-2.33
530	0.8709	21.39	-2.43
550	0.8701	21.06	-2.45

Table B.10: Results for UCSD ped1. Cuboid size: 15x15x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	**0.8542**	**22.64**	**-1.24**
110	**0.8502**	**22.13**	**-1.30**
130	0.8313	24.84	-1.28
150	0.8491	23.52	-1.47
170	0.8449	22.74	-1.44
190	0.8502	22.84	-1.38
210	0.7923	27.45	-1.50
230	0.8472	23.79	-1.37
250	0.8504	23.18	-1.36
270	0.8513	22.64	-1.54
290	0.8440	24.06	-1.54
310	0.8524	22.94	-1.56
330	0.8503	23.75	-1.29
350	0.8443	23.42	-1.30
370	**0.8571**	**22.33**	**-1.34**
390	0.8518	22.26	-1.27
410	0.8556	23.38	-1.36
430	0.8317	24.84	-1.69
450	0.8507	22.94	-1.39
470	0.8264	26.09	-1.25
490	0.8460	23.38	-1.51
510	0.8551	22.60	-1.25
530	**0.8570**	**22.26**	**-1.52**
550	0.8431	23.25	-1.55

Table B.11: Results for UCSD ped1. Cuboid size: 15x15x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	**0.8366**	**24.14**	**-0.71**
110	0.8327	23.66	-1.02
130	0.7999	27.01	-0.73
150	0.8212	26.32	-0.89
170	**0.8382**	**23.97**	**-0.90**
190	0.8147	25.74	-0.86
210	0.8306	25.18	-0.84
230	0.8326	23.38	-0.91
250	0.8257	25.87	-0.86
270	0.7923	28.78	-1.03
290	0.8149	26.84	-0.97
310	0.8268	24.87	-0.85
330	**0.8386**	**23.59**	**-0.87**
350	0.8326	25.15	-0.91
370	0.8243	24.97	-0.92
390	0.8309	23.83	-0.85
410	0.8172	25.63	-0.96
430	0.8290	24.80	-0.84
450	0.8184	26.05	-0.81
470	0.8214	25.11	-0.87
490	0.8332	24.70	-0.85
510	0.8300	23.97	-0.66
530	0.8252	24.35	-0.98
550	0.8317	24.46	-0.92

Table B.12: Results for UCSD ped1. Cuboid size: 15x15x13. Feature: cuboid position, optical flow summation.

UCSD ped2

Spatial size: 9x9

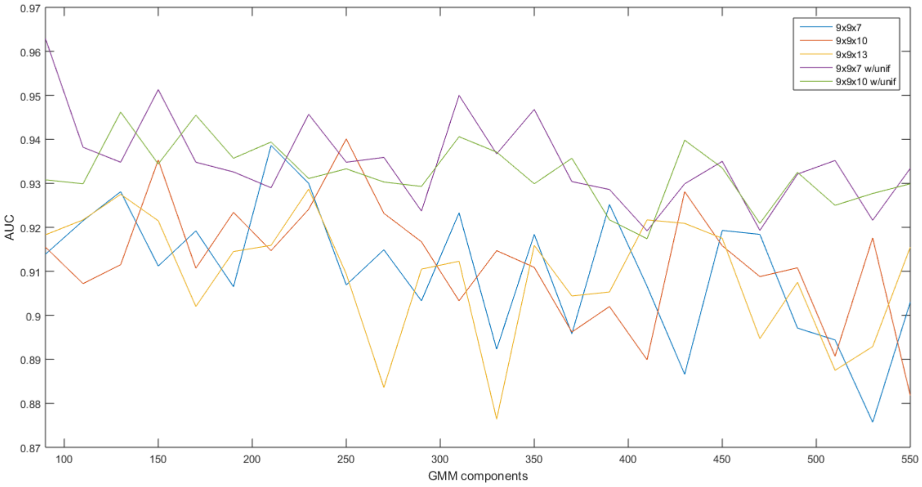


Figure B.7: AUC values with spatial size of 9x9. UCSD ped2.

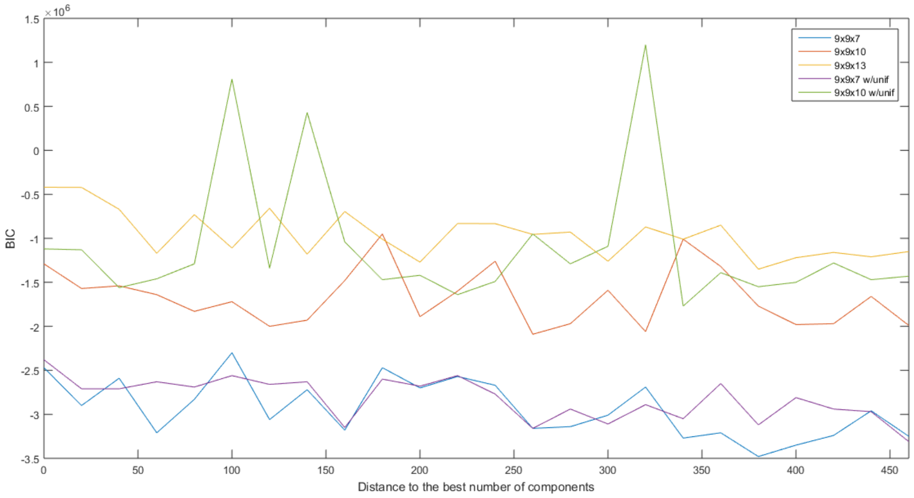


Figure B.8: BIC values with spatial size of 9x9. UCSD ped2.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.9139	15.36	-2.57
110	0.9215	13.86	-2.3
130	0.9281	12.35	-2.59
150	0.9112	13.86	-2.67
170	0.9192	14.16	-2.72
190	0.9065	12.65	-3.01
210	**0.9386**	**12.95**	**-2.47**
230	0.9300	12.95	-2.90
250	0.9069	14.16	-3.16
270	0.9149	13.86	-2.70
290	0.9033	14.46	-2.69
310	0.9233	15.36	-2.83
330	0.8923	18.37	-3.24
350	0.9184	12.65	-3.18
370	0.8958	14.46	-3.48
390	0.9252	11.75	-3.21
410	0.9066	13.55	-3.14
430	0.8866	17.17	-2.96
450	0.9193	12.95	-3.06
470	0.9184	15.06	-2.47
490	0.8971	15.66	-3.21
510	0.8944	15.66	-3.35
530	0.8757	16.57	-3.25
550	0.9030	15.36	-3.27

Table B.13: Results for UCSD ped2. Cuboid size: 9x9x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times10^6$)
90	0.9155	15.77	-0.95
110	0.9072	16.09	-1.01
130	0.9115	15.77	-1.26
150	0.9353	11.99	-1.57
170	0.9107	14.20	-1.59
190	0.9234	16.09	-1.83
210	0.9147	14.20	-1.89
230	0.9241	14.51	-1.64
250	**0.9401**	**11.04**	**-1.29**
270	0.9232	12.30	-1.72
290	0.9167	13.56	-1.93
310	0.9033	15.77	-1.32
330	0.9247	14.20	-1.6
350	0.9109	16.09	-2.09
370	0.8962	19.56	-1.98
390	0.9020	15.46	-1.77
410	0.8899	15.77	-1.66
430	0.9281	11.99	-1.54
450	0.9158	13.98	-1.48
470	0.9088	16.40	-2.06
490	0.9108	14.83	-1.97
510	0.8907	14.51	-1.97
530	0.9176	13.56	-2.00
550	0.8818	14.51	-1.99

Table B.14: Results for UCSD ped2. Cuboid size: 9x9x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times10^6$)
90	0.9183	12.91	-0.658
110	0.9217	13.25	-0.67
130	**0.9275**	**12.25**	**-0.421**
150	0.9215	12.91	-0.731
170	0.9020	15.56	-0.85
190	0.9145	14.24	-0.831
210	0.9159	11.92	-0.695
230	**0.9287**	**13.58**	**-0.419**
250	0.9094	14.90	-0.929
270	0.8836	15.89	-1.21
290	0.9105	12.91	-0.955
310	0.9123	13.91	-0.833
330	0.8764	21.52	-1.15
350	0.9159	13.91	-1.01
370	0.9044	14.24	-1.01
390	0.9053	13.58	-0.87
410	0.9217	13.25	-1.17
430	0.9209	14.90	-1.11
450	0.9175	13.58	-1.18
470	0.8947	17.88	-1.35
490	0.9075	13.91	-1.26
510	0.8875	14.57	-1.16
530	0.8929	13.91	-1.22
550	0.9155	15.56	-1.27

Table B.15: Results for UCSD ped2. Cuboid size: 9x9x13. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	**0.9629**	**9.34**	**-2.38**
110	0.9382	13.55	-2.56
130	0.9348	14.76	-2.68
150	0.9513	12.05	-2.71
170	0.9348	15.06	-2.56
190	0.9326	17.47	-2.94
210	0.9290	16.57	-2.65
230	0.9457	11.45	-2.69
250	0.9348	13.55	-2.77
270	0.9359	14.76	-2.63
290	0.9237	17.47	-2.81
310	0.9500	12.35	-2.71
330	0.9367	13.55	-2.66
350	0.9468	12.05	-2.63
370	0.9304	14.46	-2.89
390	0.9286	16.87	-3.12
410	0.9192	17.17	-3.31
430	0.9299	16.57	-3.05
450	0.9350	14.16	-2.60
470	0.9193	17.17	-2.97
490	0.9321	13.25	-3.11
510	0.9352	15.06	-3.15
530	0.9216	14.76	-2.94
550	0.9334	14.16	-3.16

Table B.16: Results for UCSD ped2. Cuboid size: 9x9x7. Feature: cuboid position, optical flow summation and uniformity.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.9308	15.46	-0.95
110	0.9299	14.20	-1.09
130	0.9402	12.62	-1.12
150	0.9344	13.25	-1.04
170	**0.9455**	**11.99**	**-1.13**
190	0.9357	12.93	-1.34
210	0.9394	13.88	-1.29
230	0.9311	15.14	-1.49
250	0.9333	14.51	-1.42
270	0.9303	14.51	-1.29
290	0.9293	14.83	-1.39
310	0.9406	11.99	-1.56
330	0.9371	11.99	+0.81
350	0.9299	13.25	+1.20
370	0.9357	14.25	+0.43
390	0.9217	16.09	-1.28
410	0.9174	19.24	-1.43
430	0.9398	14.51	-1.46
450	0.9335	13.88	-1.47
470	0.9209	16.40	-1.47
490	0.9325	13.56	-1.64
510	0.9250	15.14	-1.50
530	0.9277	15.46	-1.55
550	0.9299	15.46	-1.77

Table B.17: Results for UCSD ped2. Cuboid size: 9x9x10. Feature: cuboid position, optical flow summation and uniformity.

Spatial size: 12x12

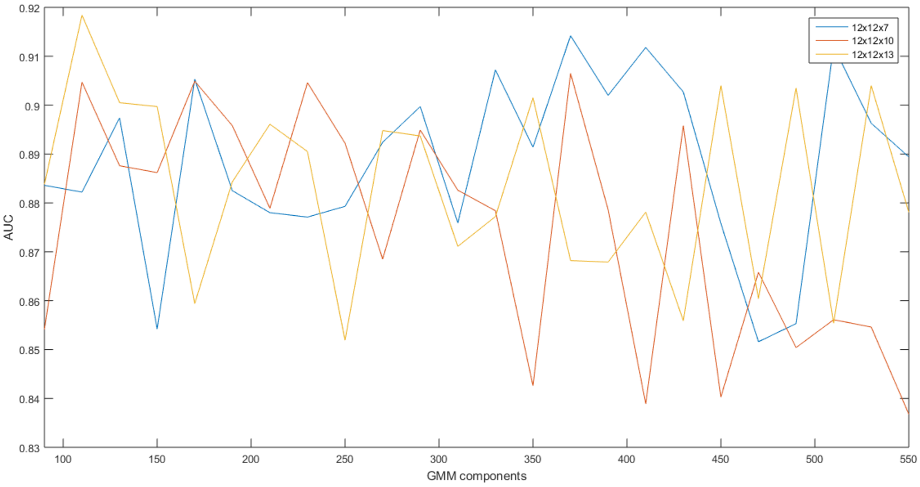


Figure B.9: AUC values with spatial size of 12x12. UCSD ped2

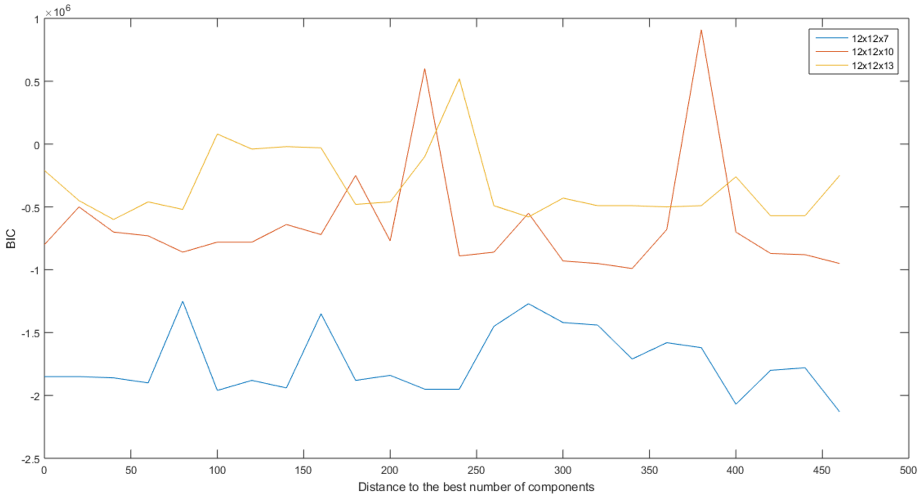


Figure B.10: BIC values with spatial size of 12x12. UCSD ped2

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8836	19.88	-1.45
110	0.8822	19.88	-1.42
130	0.8974	17.17	-1.35
150	0.8542	21.69	-1.78
170	0.9053	17.47	-1.25
190	0.8825	16.87	-1.27
210	0.8780	20.18	-1.71
230	0.8771	19.58	-1.58
250	0.8793	19.58	-1.44
270	0.8924	19.88	-1.84
290	0.8997	17.17	-1.94
310	0.8759	21.69	-1.62
330	0.9072	17.17	-1.90
350	0.8914	18.67	-1.95
370	**0.9142**	**14.46**	**-1.85**
390	0.9020	17.77	-1.88
410	0.9118	17.77	-1.86
430	0.9027	15.96	-1.96
450	0.8758	21.69	-2.07
470	0.8516	22.59	-2.13
490	0.8553	21.99	-1.80
510	0.9120	17.17	-1.85
530	0.8963	18.07	-1.88
550	0.8894	20.78	-1.95

Table B.18: Results for UCSD ped2. Cuboid size: 12x12x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8541	23.34	-0.68
110	**0.9047**	**17.35**	**-0.70**
130	0.8876	16.09	-0.72
150	0.8862	17.67	-0.25
170	0.9049	18.61	-0.50
190	0.8958	16.40	-0.86
210	0.8789	21.45	-0.60
230	0.9046	18.93	-0.73
250	0.8922	20.19	-0.64
270	0.8685	22.08	-0.55
290	0.8949	14.83	-0.78
310	0.8826	21.45	-0.77
330	0.8784	20.50	-0.86
350	0.8426	24.61	-0.70
370	**0.9065**	**17.67**	**-0.80**
390	0.8786	22.40	-0.89
410	0.8389	24.61	-0.88
430	0.8958	18.93	-0.78
450	0.8403	24.92	-0.87
470	0.8658	23.03	-0.93
490	0.8504	24.61	-0.91
510	0.8561	22.40	-0.95
530	0.8546	24.61	-0.99
550	0.8368	26.18	-0.95

Table B.19: Results for UCSD ped2. Cuboid size: 12x12x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8838	19.87	+0.52
110	**0.9184**	**14.24**	**-0.21**
130	0.9005	16.23	+0.08
150	0.8997	14.57	-0.04
170	0.8594	22.85	-0.26
190	0.8843	18.21	-0.10
210	0.8961	16.56	-0.02
230	0.8905	17.55	-0.46
250	0.8519	23.18	-0.25
270	0.8948	18.21	-0.03
290	0.8937	16.89	-0.48
310	0.8711	20.86	-0.49
330	0.8772	18.21	-0.43
350	0.9015	16.23	-0.52
370	0.8682	22.19	-0.49
390	0.8679	21.19	-0.50
410	0.8781	23.51	-0.49
430	0.8559	23.51	-0.57
450	0.9040	14.57	-0.45
470	0.8604	23.51	-0.49
490	0.9035	15.89	-0.46
510	0.8554	24.17	-0.57
530	0.9040	16.89	-0.60
550	0.8781	20.20	-0.58

Table B.20: Results for UCSD ped2. Cuboid size: 12x12x13. Feature: cuboid position, optical flow summation.

Spatial size: 15x15

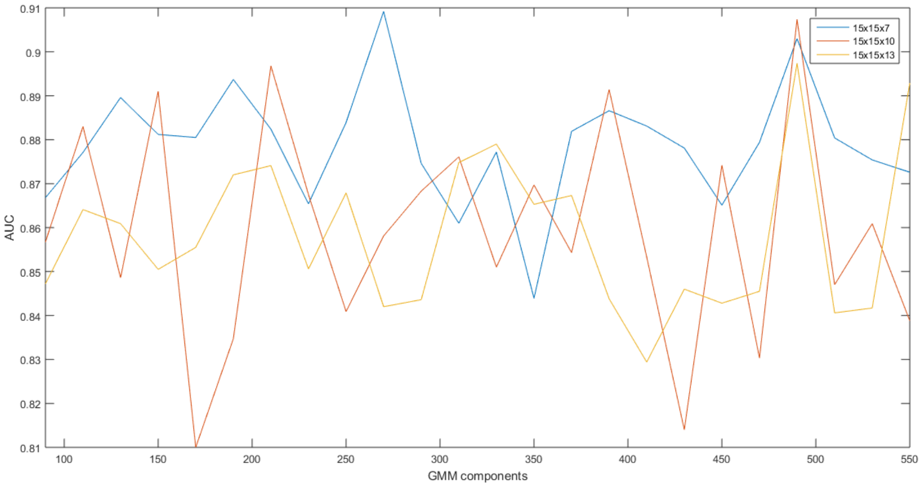


Figure B.11: AUC values with spatial size of 15x15. UCSD ped2.

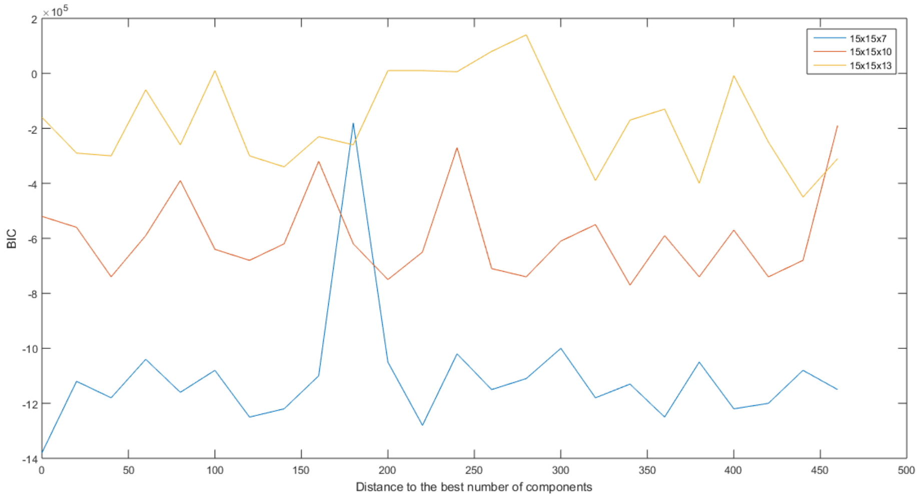


Figure B.12: BIC values with spatial size of 15x15. UCSD ped2.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8668	23.80	-1.05
110	0.8771	22.89	-1.00
130	0.8896	20.18	-1.04
150	0.8812	20.18	-1.18
170	0.8805	21.08	-1.05
190	0.8937	21.08	-1.18
210	0.8824	22.89	-1.22
230	0.8654	23.49	-1.22
250	0.8839	21.69	-1.08
270	**0.9092**	**19.88**	**-1.38**
290	0.8746	23.80	-1.13
310	0.8610	22.59	-1.08

Comp	AUC	EER (%)	BIC ($\times 10^6$)
330	0.8772	23.80	-1.11
350	0.8439	25.90	-1.15
370	0.8819	21.69	-1.10
390	0.8866	20.48	-1.16
410	0.8831	20.48	-1.25
430	0.8781	24.10	-1.15
450	0.8651	21.39	-1.20
470	0.8794	22.59	-1.02
490	**0.9030**	**18.37**	**-1.12**
510	0.8804	21.69	-1.28
530	0.8754	22.59	-1.18
550	0.8726	22.59	-1.25

Table B.21: Results for UCSD ped2. Cuboid size: 15x15x7. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8567	22.40	-0.27
110	0.8830	17.98	-0.39
130	0.8486	26.18	-0.55
150	0.8910	17.98	-0.59
170	0.8100	25.87	-0.19
190	0.8347	25.87	-0.57
210	0.8968	18.30	-0.56
230	0.8675	21.77	-0.62
250	0.8409	27.44	-0.59
270	0.8581	23.97	-0.65
290	0.8683	26.18	-0.32
310	0.8761	20.82	-0.64
330	0.8510	27.76	-0.61
350	0.8697	23.66	-0.62
370	0.8543	24.92	-0.71
390	0.8914	17.35	-0.74
410	0.8533	24.61	-0.74
430	0.8140	28.71	-0.68
450	0.8742	22.08	-0.68
470	0.8303	28.39	-0.74
490	**0.9074**	**15.77**	**-0.52**
510	0.8470	27.44	-0.77
530	0.8609	19.87	-0.75
550	0.8388	23.66	-0.74

Table B.22: Results for UCSD ped2. Cuboid size: 15x15x10. Feature: cuboid position, optical flow summation.

Comp	AUC	EER (%)	BIC ($\times 10^6$)
90	0.8472	24.83	+0.14
110	0.8641	27.15	-0.26
130	0.8609	27.48	+0.01
150	0.8505	26.49	+0.08
170	0.8555	26.16	+0.01
190	0.8720	23.51	+0.01
210	0.8741	21.52	-0.26
230	0.8506	26.82	+0.006
250	0.8679	21.52	-0.30
270	0.8420	29.47	-0.008
290	0.8436	26.82	-0.13
310	0.8748	21.52	-0.06
330	0.8790	22.52	-0.30
350	0.8653	21.85	-0.23
370	0.8673	24.50	-0.34
390	0.8438	26.16	-0.17
410	0.8294	27.48	-0.31
430	0.8460	25.17	-0.13
450	0.8428	27.48	-0.40
470	0.8455	22.85	-0.39
490	**0.8974**	**19.21**	**-0.16**
510	0.8406	24.50	-0.45
530	0.8417	26.16	-0.25
550	0.8929	19.87	-0.29

Table B.23: Results for UCSD ped2. Cuboid size: 15x15x13. Feature: cuboid position, optical flow summation.