Temporal Video Segmentation by Event Detection: A Novelty Detection Approach

Mahesh Venkata Krishna,[1,*] Paul Bodesheim,[1,**]
Marco Körner,[1,***] and Joachim Denzler[1,****]

[1] *Computer Vision Group,*
Friedrich Schiller University Jena,
07743 Jena, Germany
http://www.inf-cv.uni-jena.de

Temporal segmentation of videos into meaningful image sequences containing some particular activities is an interesting problem in computer vision. We present a novel algorithm to achieve this semantic video segmentation. The segmentation task is accomplished through event detection in a frame-by-frame processing setup. We propose using one-class classification (OCC) techniques to detect events that indicate a new segment, since they have been proved to be successful in object classification and they allow for unsupervised event detection in a natural way. Various OCC schemes have been tested and compared, and additionally, an approach based on the *temporal self-similarity maps* (TSSMs) is also presented. The testing was done on a challenging publicly available thermal video data-set. The results are promising and show the suitability of our approaches for the task of temporal video segmentation.

Keywords: Temporal video segmentation, one-class classification, novelty detection, temporal self-similarity maps, unsupervised video analysis.

[*] Electronic address: mahesh.vk@uni-jena.de
[**] Electronic address: paul.bodesheim@uni-jena.de
[***] Electronic address: marco.koerner@uni-jena.de
[****] Electronic address: joachim.denzler@uni-jena.de

1. INTRODUCTION

In many computer vision applications, like surveillance, microscopy videos etc., it is often the case that there is a large video corpus/stream with interesting events sparsely spread over it. It then becomes a challenging task to extract interesting portions from the huge amount of data. The aim of this work is to achieve this interesting event extraction by segmenting the input video into various semantic *phases*.

In addition, temporal video segmentation gets us into important theoretical issues such as what defines an interesting event and how can a machine *guess* by itself what an interesting event may be. As definitions of events are application dependent, a generic definition of an event can only be stated as : "Something that is not normal in the video", *i.e.*, something *novel*. This leads us to the main contribution of this work: the application of one-class classification (OCC) algorithms to generic and unsupervised temporal video segmentation. In our approach, detecting temporal novelties in a video is a big step towards temporal video segmentation. To perform one-class classification, we look among the various approaches that have been proposed in the field of object classification, such as one-class support vector machine (1SVM) [18], support vector data description (SVDD) [21], Gaussian process regression (GPR) [10], or the recently introduced null space approach of [1] based on kernel null Foley-Sammon transform (KNFST). One can clearly see that the OCC setup matches our problem scenario: a model of normality (or known patterns) has to be built by clustering in the feature space and novelty is declared in case an outlier is met in the testing phase. We show in the following sections how one-class classification techniques can be used for temporal video segmentation.

Furthermore, as an additional contribution, the application of *temporal self-similarity maps* (TSSMs) to temporal video segmentation is studied. This method has been applied for activity recognition tasks in works such as [9], and we adapt this approach to our problem.

The remainder of this paper is organized as follows. First, in section 2, we review related work in the field of temporal video segmentation. A brief discussion of their relative advantages and shortcomings is provided. In section 3, we review the OCC techniques we apply within our temporal video segmentation framework. We then present our approach for temporal video segmentation in section 4 together with the explanation how to use OCC techniques for event detection in videos. In addition, we describe the temporal self-similarity maps and how we adapt it for the present problem. The results on the thermal videos of the CVPR change detection dataset [7] are presented in section 5 highlighting the suitability of our approach. A summary of our findings and suggestions for future research directions conclude the paper.

2. PREVIOUS WORK ON TEMPORAL VIDEO SEGMENTATION

The work by Koprinksa and Carrato [11] provides a good survey of temporal video segmentation techniques based on a very diverse range of theoretical concepts and for various applications. Most of the algorithms presented there concentrate on directly finding the differences between frames through some distance measure between feature vectors of consecutive frames. A threshold is then applied to this distance to achieve segmentation. Some examples of the features they used are global or local color histograms [20, 23] and edge pixels [22] beside others. These approaches yield reasonable results but are based on directly finding inter-frame differences without modeling the underlying scenario, which limits their applicability to scenarios requiring semantic modeling.

In the work of Boreczky and Wicox [2] Hidden Markov Models (HMM) are used for temporal video segmentation. The states in the HMM model various camera parameters and changes in states represent indicate a new temporal segment. The model transition probabilities are learned from labeled training data and segmentation is performed using the standard Viterbi algorithm. This method is suitable only in situations where the states of the HMM are defined clearly and well-labeled training data is available.

Liu *et al.* [13] have presented an approach based on the perceived motion energy feature, where optical flow vectors in each frame are averaged and multiplied with a factor arising out of the dominant direction of motion. These features are then clustered to form segments of the video corpus. This method is very useful for videos containing a lot of motion, but when events happen that do not alter the motion profile of the frames (*e.g.*, color changes), it is likely to fail.

[5] is a more recent work providing a framework for segmentation in various media including video. Here, the authors use a similar approach to our Temporal Self-Similarity Maps. They extract low-level features from the frames and then construct an inter-frame similarity matrix. Thereafter, they perform a matched filter operation on the matrix and then supervised classification (k Nearest Neighbors in their case). But this approach requires supervision and is thus not applicable for the present problem situation.

Another interesting work which also performs multimedia segmentation is [19]. The authors use a modification of the well-known Scene Transition Graphs (STGs) to perform segmentation in modality and in the end, fuse the segmentation results probabilistically. This probabilistic fusion basically calculates an overall segmentation probability by weighted merging of individual STG outputs. This method is basically designed for multi-modal data, with multiple parallel STGs for each modality. When reduced to single modes like in our case, it effectively compares feature similarities between sub-segments of the video called shots and clusters them accordingly into more refined segments. While this method yields good results, the speed of execution is still a possible issue, as the inter-shot distances will have to be calculated on a large number of shot combinations in long videos.

We propose a fast, generic and unsupervised framework for temporal video segmentation without assumptions on application scenarios. The next section provides the theoretical aspects of our proposed approach based on one-class classification.

3. ONE-CLASS CLASSIFICATION TECHNIQUES

In this section, we briefly describe the idea behind OCC as well as the different methods we are using within our framework, namely one-class support vector machine, Gaussian process regression, and kernel null Foley-Sammon transform.

3.1. The Task of One-Class Classification

In an OCC scenario, there are only training samples $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\} \subset \boldsymbol{\mathcal{X}}$ of a single class available. Thus, all of them have the same constant label, *e.g.*, $\boldsymbol{y} = \mathbf{1} = (1, 1, \ldots, 1)^{\mathsf{T}}$. This class is often referred to as target class and the corresponding samples as target data or target set [21]. The aim is to find an appropriate description of the class distribution to distinguish this single class from every other possible and currently unknown class. Therefore, a novelty score should be inferred for each test sample $\boldsymbol{x}^*$ such that a large score indicates strong membership to the target class. If this score is below a certain threshold, the test sample will be treated as an outlier not belonging to the estimated distribution. The following methods allow for a suitable modeling with samples that only stem from a single class.

3.2. One-Class Support Vector Machine

Probably the most common method for one-class classification is one-class support vector machine (1SVM) introduced by Schölkopf *et al.* [18]. The aim of this approach is to separate the distribution of class samples from the origin in some (often high-dimensional) kernel feature space $\boldsymbol{\mathcal{F}}$ by a hyperplane with maximum margin. There is no need to specify the mapping $\boldsymbol{\Phi}$ of feature vectors $\boldsymbol{x}$ to the kernel feature space $\boldsymbol{\mathcal{F}}$ explicitly, since it is usually given implicitly by a kernel function κ that computes the inner product in $\boldsymbol{\mathcal{F}}$ solely using representations in the input space $\boldsymbol{\mathcal{X}}$. The optimal separating hyperplane can be obtained by solving the following quadratic optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\xi},\rho} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \\ \text{s.t.} \quad & \left\langle \boldsymbol{w}, \boldsymbol{\Phi}\left(\boldsymbol{x}^{(i)}\right)\right\rangle \geq \rho - \xi_i \\ & \qquad \xi_i \geq 0 \qquad\qquad \forall\, 1 \leq i \leq N \quad , \end{aligned} \tag{1}$$

where $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}$ describes the hyperplane that has an additional offset $\rho \in \mathbb{R}$, $\boldsymbol{\xi} \in \mathbb{R}^N$ is the vector containing slack variables (one for each of the N training samples), and ν is a parameter indicating the upper bound on the fraction of outliers that are located on the other side of the hyperplane. According to [18], the hyperplane $\boldsymbol{w}$ is specified by a linear combination of mapped input features:

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{\Phi}\left(\boldsymbol{x}^{(i)}\right) \quad , \tag{2}$$

such that inner products in Eq. (1) can be computed by the kernel function κ. This also allows for computing the novelty score of a test sample $\boldsymbol{x}^*$ as follows:

$$\begin{aligned} s\left(\boldsymbol{x}^*\right) &= \left\langle \boldsymbol{w}, \boldsymbol{\Phi}\left(\boldsymbol{x}^*\right)\right\rangle - \rho \\ &= \sum_{i=1}^{N} \alpha_i \kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^*\right) - \rho \end{aligned} \tag{3}$$

Closely related to 1SVM is SVDD [21], where the class distribution is described by a hypersphere with minimum volume but enclosing the training samples. It is shown in [18] that using kernel functions with constant self-similarities $\kappa\left(\boldsymbol{x}, \boldsymbol{x}\right)$, both methods 1SVM and SVDD solve the same optimization problem and hence generate equivalent classification models. In our experiments, we apply 1SVM and use the implementation of libsvm [4].

3.3. Gaussian Process Regression

The Gaussian process framework is a well-known probabilistic methodology that is successfully used for tasks such as regression and classification [17]. In the case of Gaussian process regression (GPR), outputs $y(\boldsymbol{x})$ are assumed to be generated according to a latent function g and a noise term ε:

$$y(\boldsymbol{x}) = g(\boldsymbol{x}) + \varepsilon \quad . \tag{4}$$

Following a Bayesian framework, output values of unknown samples $\boldsymbol{x}^*$ are predicted probabilistically by marginalizing over both latent function values and noise. While this is in most cases infeasible to realize exactly, a few assumptions on the entities in equation 4 make the prediction tractable. The first assumption is that latent functions g are drawn from a Gaussian process prior with zero mean and covariance function κ that is usually a kernel function: $g \sim \mathcal{GP}\left(\mathbf{0}, \kappa(\,\cdot\,,\cdot\,)\right)$. Second, the noise term ε is assumed to be zero mean Gaussian: $\varepsilon \sim \mathcal{N}(0, \sigma_{\mathsf{n}}^2)$.

Using these assumptions, the predictive distribution over output values is normally distributed, *i.e.*, $y^* | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}^* \sim \mathcal{N}(\mu_*, \sigma_*^2)$, where moments μ_* and σ_*^2 can be computed in closed form. The work of [10] shows how GPR can be used to solve OCC problems. The authors propose using either the predictive mean μ_* (GPR-Mean) or negative variance $-\sigma_*^2$ (GPR-Var) as novelty scores:

$$\begin{aligned} \mu_* &= \boldsymbol{k}_*^{\mathsf{T}} \left(\boldsymbol{K} + \sigma_{\mathsf{n}}^2 \boldsymbol{I}\right)^{-1} \mathbf{1} \quad \text{and} \\ -\sigma_*^2 &= -\left(k_{**} - \boldsymbol{k}_*^{\mathsf{T}} \left(\boldsymbol{K} + \sigma_{\mathsf{n}}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{k}_* + \sigma_{\mathsf{n}}^2\right) \quad , \end{aligned} \tag{5}$$

where $\boldsymbol{K} = \kappa\left(\boldsymbol{X}, \boldsymbol{X}\right), \boldsymbol{k}_* = \kappa\left(\boldsymbol{X}, \boldsymbol{x}^*\right), k_{**} = \kappa\left(\boldsymbol{x}^*, \boldsymbol{x}^*\right)$, and $\boldsymbol{I}$ is the unit matrix. To evaluate our video segmentation approach equipped with both GPR methods, either GPR-Mean or GPR-Var, we use the code provided by the authors of [10].

3.4. Kernel Null Foley-Sammon Transform

In [1], the authors propose using a null space approach based on the kernel null Foley-Sammon transform (KNFST) for novelty detection. The idea of this method is to project all samples of the same class to a single target point but different classes to different target points in some subspace called the null space of the training data. This projection is carried out by KNFST and the novelty score of a test sample is calculated based on the distances in the null space between the projected test sample and target points of the classes known during training. The transformation is fully described by the null projection directions $\boldsymbol{\varphi}$ that maximize the Fisher discriminant criterion J based on the between-class scatter matrix $\boldsymbol{S}_b$ and the within-class scatter matrix $\boldsymbol{S}_w$:

$$\mathrm{J}\left(\boldsymbol{\varphi}\right) = \frac{\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_b \boldsymbol{\varphi}}{\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{\varphi}} \quad . \tag{6}$$

The null projection directions achieve $\mathrm{J}\left(\boldsymbol{\varphi}\right) = \infty$ and thus best separability with respect to this criterion, since they set the within-class scatter to zero but ensure a positive between-class scatter:

$$\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{\varphi} = 0 \quad , \tag{7}$$

$$\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_b \, \boldsymbol{\varphi} > 0 \quad . \tag{8}$$

Although focusing on multi-class novelty detection in their experiments, the authors also provide two strategies for one-class classification:

1. Separating the target class from the origin in the kernel feature space.
2. Separating the target class from an artificial second class created by switching the sign of the features of the target class samples.

Since they claim that the first strategy is more appropriate in general and the artificial second class can be avoided, we apply this one within our framework. The source code of all methods described in [1] has already been made publicly available and is used in our experiments.

4. VIDEO SEGMENTATION BY EVENT DETECTION

In this section, we describe how OCC and TSSMs can be used to perform the task of temporal video segmentation.

4.1. The Idea of Our Approach

To achieve temporal video segmentation, an approach similar to work flow segmentation can be used, where different *phases* of the video are marked based on their semantic content. We assume that each stage has a certain minimum number of frames, denoted by the parameter F. These frames are used to build a model of the scene for the current segment. Then, for each succeeding frame, we compare it with the model and classify a detected novelty as the start of a new segment. Figure 1 demonstrates the basic idea behind our method.

Thus, in our approach we detect events that lead to a change of the current phase by using OCC methods as explained in the following subsection.

4.2. Our One-Class Classification Approach

We follow an OCC approach similar to the one used for novelty detection in [1]. Let us assume that there are features available for each frame stored in a specific feature vector.

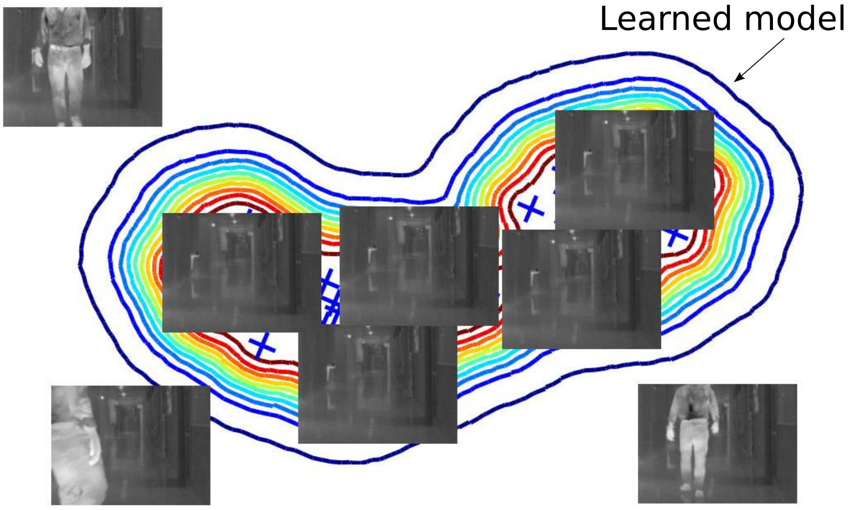


Figure 1. Viewing video segmentation as an OCC problem. The first few frames, parametrized by F, model the *normal* situation in the scene and deviations are marked as the start of next segment.

The feature extraction methods will be explained in section 5. We start with learning a one-class model (Sec. 3) using the features of the first F frames of the video. Here, we assume that there is no event within these first frames and assign them with phase count 1. In most real-life situations, a few frames in the order of 20 to 60 correspond to 1-3 seconds, where it is reasonable to assume that no interesting event happens. For every consecutive frame, we evaluate the learned model to obtain its novelty score (depending on the OCC method that is used). This is done until the score of a frame drops below a specified threshold T. If this is the case, we have detected an event leading to a phase change.

Assuming that such events, which indicate a phase change, are sparsely spread over time (*i.e.*, do not happen closely, with a gap of at least F frames between them), we learn a new one-class model with the features of the next F frames. The unlabeled previous frames are assigned with the phase count of the old model and we update the phase count of the current model. The video sequence is segmented this way, completely unsupervised. An overview of our approach can be seen in Fig. 2.

Note that our approach does not need a training step using manually labeled sequences to learn a suitable model. Moreover, it can be directly applied to any video sequence since the model is learned on-the-fly within the sequence that should be segmented. We may only have to adjust the parameters F and T as well as method specific parameters of the OCC model. Additionally, we are able to process a video online without knowing the whole

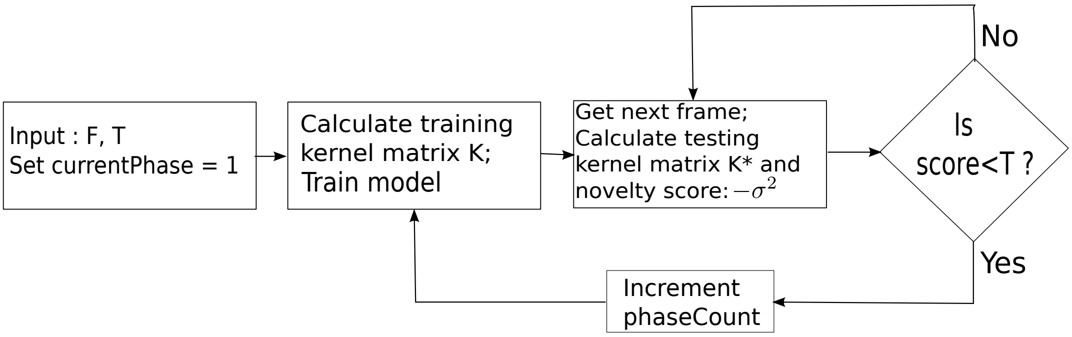

Input : F, T
Set currentPhase = 1
Calculate training kernel matrix K; Train model
Get next frame; Calculate testing kernel matrix K* and novelty score: $-\sigma^2$
Is score<T ?
No
Yes
Increment phaseCount

Figure 2. Overview of our one-class classification approach for temporal video segmentation.

sequence at the beginning.

4.3. Event Segmentation using Temporal Self-Similarity

In order to put our approach and its performance into perspective and present baseline results, we will briefly describe an alternative way to segment videos into shorter clips of certain actions, which in contrast does not rely on a prior learning phase but bases on heuristic assumptions.

When capturing videos of actions performed by humans, this can be regarded as observations of dynamic systems. Then, the concept of *temporal self-similarity maps (TSSM)* [12] can be used in order to detect abrupt changes in their evolution, which are assumed to separate certain actions. Here, we will briefly describe the concept of TSSMs and show how they can be used for event detection tasks.

Given a sequence $\boldsymbol{I}_{1:N} = \{I_1, \ldots, I_N\}$ of images $I_i, 1 \leq i \leq N$, a temporal self-similarity map is generically defined as a square and symmetric matrix

$$\boldsymbol{S}_{f,d}^{\boldsymbol{I}_{1:N}} = \left[d(f(I_i), f(I_j))\right]_{i,j} \in \mathbb{R}^{N \times N} \tag{9}$$

containing pairwise similarities $d(\cdot, \cdot)$ of low-level image features $f(\cdot)$ computed independently for every frame. In the literature, it has already been shown that TSSMs preserve invariants of the dynamic systems they capture [15], they are stable wrt. different embedding dimensions [8, 15], and invariant under isometric transformations [15]. Though not being invariant under projective or affine transformations, TSSMs are heuristically shown to be stable under 3d view changes [9].

4.3.1. Image Features

The choice for low-level image features $f(\cdot)$ is of inherent importance and has to suit the given scenario. In the following, we will discuss some possible alternatives.

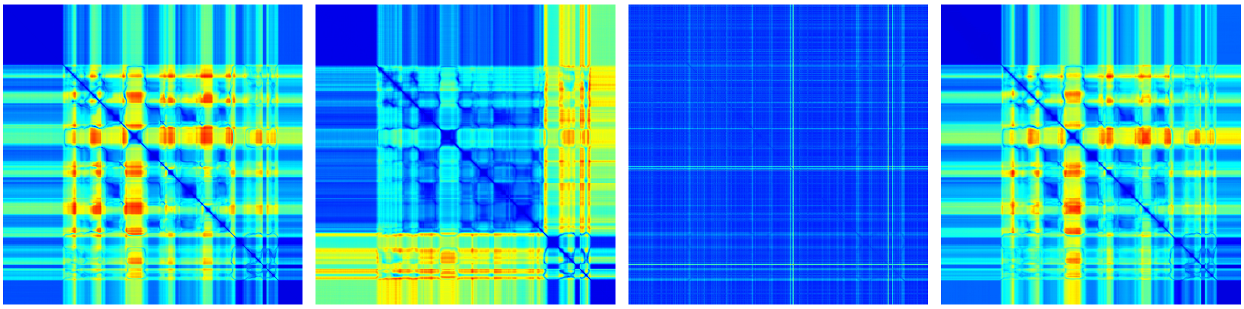
Intensity Values The simplest way to convert an image into a descriptive feature vector $f_{\text{int}}(I) \in \mathbb{R}^{M \cdot N}$ is to append its intensities, as proposed for human gait analysis [6]. While this is suitable for sequences with a single stationary actor, it yields large feature vectors and is very sensitive to noise and illumination changes.

Landmark Positions Assuming to be able to track anatomical or artificial landmarks of the actor over time, their positions $f_{\text{pos}}(I) = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots), \boldsymbol{x}_i = (x_i, y_i, z_i)$, can be used to represent the current system configuration [9]. This is sufficient as long as the tracked points are distributed over moving body parts, but it demands points to be able to be tracked continuously.

Histograms of Oriented Gradients (HOG) Histograms of Oriented Gradients have been shown to give good representations of shape for object detection. For this purpose, the image is subdivided into overlapping cells, where the distribution of gradient directions is approximated by a fixed-bin discretization. These certain local orientation histograms are normalized to the direction of the strongest gradient in order to obtain local rotation invariance. Appending those local gradient histograms gives the final descriptor $f_{\text{HOG}}(I) = (\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots), \boldsymbol{h}_i = (n_i^0, n_i^1, \ldots)$ [9].

Histograms of Optical Flows (HOF) When analyzing the displacements of each pixel between two succeeding frames, this *optical flow* field represents an early fusion of temporal dynamics. Building a global histogram over discretized flow orientations or appending histograms obtained from smaller subimages yield the HOF descriptor $f_{\text{HOF}}(I)$.

Fourier Coefficients When computing the 2-dimensional discrete Fourier transform (DFT) $\hat{a}_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{m,n} \cdot \mathrm{e}^{-2\pi\mathrm{i}\left(\frac{mk}{M}+\frac{nl}{N}\right)}, 0 \leq k \leq M-1, 0 \leq l \leq N-1, \hat{a}_{k,l} \in \mathbb{C}$ of an image patch I, the series $[\hat{a}_{k,l}]$ of Fourier coefficients contain spectral information up to a given cutoff frequency $0 \leq k \leq M_c - 1, 0 \leq l \leq N_c - 1$ and inherently provides invariance against translation. Since the first Fourier coefficient $\hat{a}_{0,0}$ represents the mean intensity of the transformed image patch I, the Fourier coefficient descriptor $f_{\text{Fourier}} = (\hat{a}_{0,1}, \hat{a}_{0,2}, \ldots, \hat{a}_{1,N_c-1}, \ldots, \hat{a}_{M_c-1,N_C-1})$ is further invariant wrt. global illumination changes. By tuning the cutoff frequencies M_c, N_c, statistical noise can be suppressed as it is represented by higher-order frequencies. Since DFT can be implemented in parallel on modern GPU environments, these features can be computed very efficiently.



(a) Intensity (b) HOG (c) HOF (d) Fourier

Figure 3. TSSMs extracted from the *diningRoom* sequence from the *CVPR 2012 Change Detection* dataset using different low-level image features.

A qualitative comparison of these features extracted from different action classes is given in Fig. 3. It can be seen that the HOF feature shows many abrupt changes, while the other TSSMs contain more smooth transitions between the certain similarity values. The HOG feature seems to be more sensitive so temporal changes at small time scale, which could be explained by image noise and might harm the further processing. Hence, we concentrate on using the proposed Fourier coefficients, since they are easily and fast to compute and provide some handy invariants by design.

4.3.2. Similarity Measures

Beside the choice for a suitable image representation $f(\cdot)$, the distance measure $d(\cdot,\cdot)$ plays an important role when computing self-similarities. Distance measures are related to similarity measures as follows: small distances correspond to high similarity and vice versa.

Euclidean Distances The euclidean distance $d_{\text{eucl}}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \|\boldsymbol{f}_1 - \boldsymbol{f}_2\|_2$ serves as a straightforward way to quantify the similarity between two image feature descriptors $\boldsymbol{f}_1 = f(I_1)$ and $\boldsymbol{f}_2 = f(I_2)$ of equal length, as proposed by [9]. While this is easy to compute, it might be unsuited for histogram data [14], since false bin assignments would cause large errors in the euclidean distance.

Normalized Cross-Correlation From a signal-theoretical point of view, the image feature descriptors $\boldsymbol{f}_1, \boldsymbol{f}_2$ can be regarded as D-dimensional discrete signals of equal size. Then, the normalized cross-correlation coefficient $\text{NCC}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \left\langle \frac{\boldsymbol{f}_1}{\|\boldsymbol{f}_1\|}, \frac{\boldsymbol{f}_2}{\|\boldsymbol{f}_2\|} \right\rangle \in [-1, 1]$ measures the cosine of the angle between the signal vectors $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$. Hence, the distance measure $d_{\text{NCC}}(\boldsymbol{f}_1, \boldsymbol{f}_2) = 1 - \text{NCC}(\boldsymbol{f}_1, \boldsymbol{f}_2) \in [0, 2]$ is independent from their lengths.

Histogram Intersection The intersection $\text{HI}(\boldsymbol{h}_1, \boldsymbol{h}_2) = \sum_{i=0}^{D-1} \min(h_{1,i}, h_{2,i})$ of two histograms $\boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^D$ was shown to perform better for codebook generation and image classification tasks [16]. In case of comparing normalized histograms, the histogram intersection distance $d_{\mathrm{HI}}(\boldsymbol{h}_1, \boldsymbol{h}_2) = 1 - \mathrm{HI}(\boldsymbol{h}_1, \boldsymbol{h}_2)$ is bounded by $[0, 1]$.

4.3.3. Observations

According to Körner *et al.*[12], atomic action primitives induce similar structures within the corresponding TSSM. They further observed, that the local structure of these TSSMs reflects the temporal relations between different system configurations over time, as summarized in Tab. 1.

Having these observations in mind, action segmentation can be performed by identifying phases of high similarity (*i.e.*, stationary phases), abrupt changes of similarity values, or descriptive structures within the TSSM. For this purpose, we project all similarity values of a TSSM $\boldsymbol{S}_{f,d}^{\boldsymbol{I}_{1:N}} \in \mathbb{R}^{N \times N}$ onto one of the matrix dimensions and smooth it by convolution with a *Gaussian* kernel g_{Gauss} in order to obtain a *self-similarity signature*

$$\boldsymbol{s}_{f,d}^{\boldsymbol{I}_{1:N}} = \left(\boldsymbol{S}_{f,d}^{\boldsymbol{I}_{1:N}} \cdot \mathbf{1}_N^\top\right) * g_{\mathrm{Gauss}}, \quad \boldsymbol{s}_{f,d}^{\boldsymbol{I}_{1:N}} \in \mathbb{R}^N \tag{10}$$

of the complete image sequence $\boldsymbol{I}_{1:N}$. Then, zero crossings of the first-order derivative $\nabla \boldsymbol{s}_{f,d}^{\boldsymbol{I}_{1:N}}$ indicate possible break points of the video stream, which are further filtered wrt. plausibility considerations, *e.g.*, the minimal clip length. In contrast to the OCC methods described before, this approach is designed to act in an offline manner, so the whole video data is available at evaluation time.

5. EXPERIMENTS

As generic temporal semantic segmentation of video sequences is a new challenge in computer vision, we are not aware of any existing dataset specifically intended for this purpose. For this reason, the change detection dataset for the CVPR 2012 change detection workshop [7] is used to test our approach and to demonstrate its suitability. The thermal images from this dataset have been used because they contain sequences with large variance in object size and intensity contrast. The ground truth data provided with the dataset is oriented towards motion detection and hence, we created our own ground truth for our application.[1] The data is in the form of grayscale frames of size 320x240 pixels. There are

[1] Readers interested in obtaining the ground truth for their own further research or verification of our methods can contact the authors.

Table 1. Semantic interpretations of patterns shown in TSSMs introduced by recorded actions.

Pattern	Interpretation
Homogeneous areas	The corresponding atomic action represents a stationary process
Fading in corners	The recorded action represents a Non-stationary process
Periodic structures	The recorded action contains a cyclic/periodic motion
Isolated points	The recorded action contains an abrupt fluctuation
(Anti-) Diagonal straight lines	The recorded action contains different atomic actions with similar evolutionary characteristics in (reversed) time
Horiz. & vert. lines	No or slow change of states for a given period of time
Bow structures	The recorded action contains different atomic actions with similar evolutionary characteristics in reversed time with different velocities

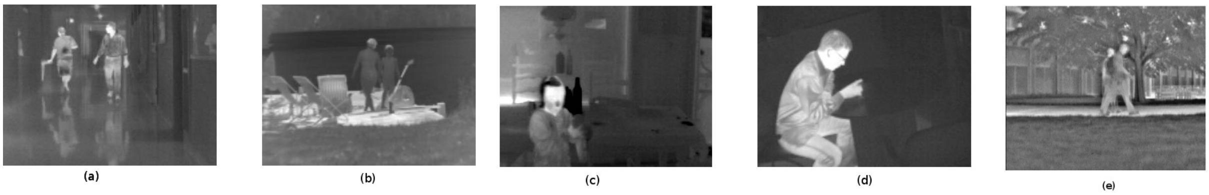
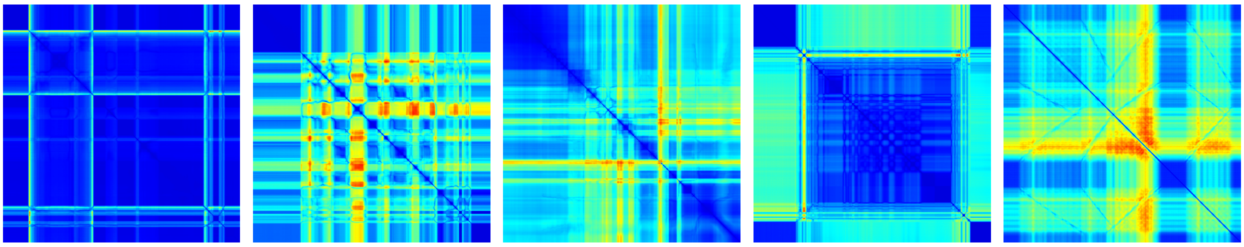


Figure 4. Example frames of the five sequences within the CVPR *change detection* dataset: (a) *corridor*, (b) *lakeSide*, (c) *diningRoom*, (d) *library*, (e) *park*.

five sequences, namely *corridor, diningRoom, lakeSide, library,* and *park.* Example frames of the dataset are shown in Figure 4. Each sequence contains different kind of motion and different zoom levels such that the object sizes are very different.

5.1. *Features for Each Frame*

Due to the fact that we want to build a generic framework for temporal video segmentation without using specific knowledge about the target application scenarios and definition of events, it is extremely challenging to choose a suitable feature set for the algorithm. For the present implementation for OCC methods, evaluated on the thermal surveillance videos, we use pyramidal histogram of oriented gradients (PHOG) as proposed in [3]. This is a very suitable feature because in surveillance sequences, events are defined by entry of a person, change in the normal movements of the people in the scene, etc., and PHOG features are



(a) *corridor* (b) *diningRoom* (c) *lakeSide* (d) *library* (e) *park*

Figure 5. TSSMs extracted from the *CVPR 2012 Change Detection* dataset using Fourier descriptors as low-level image features.

very efficient at representing object shapes in the frame. In this feature representation, local features are represented as histograms of edge orientations. Each histogram bin represents the number of edges orientated in a specific direction. To represent shapes of various sizes within a frame, an image pyramid is built for the frame and histograms from each pyramid level are concatenated. For more details, the reader is referred to [3]. In our implementation, we used 4 levels in the pyramid and 16 bins at each level. For the TSSM part, we used the Fourier features as discussed in Sec. 4.

5.2. Experimental Setup

For evaluation we designed two sets of experiments. In the first experiment, the TSSM-based method was tested in order to provide baseline results. For this purpose, we used Fourier descriptors as low-level features and histogram intersection to compute similarities, as proposed by Körner *et al.* [12]. The obtained temporal self-similiarity maps are shown in Fig. 5.

In the second experiment, the OCC methods were tested. Here, we tested the effects of the two parameters: T and F. The threshold T was varied depending on the OCC method used, as different methods yielded different ranges on scores. Table 2 gives a list of threshold values for different methods.

The parameter F, *i.e.*, the number of frames used for training the model, was varied from 20 to 75 in steps of 5. Thus, the first F frames in each new phase are assumed not to have any special event. At the normal rate of 25 frames per second, this means we are assuming constancy for only about 1 to 3 seconds, which is a very realistic and reasonable assumption.

Table 2. Threshold ranges used in the experiments (parameter T).

Method	Threshold range
GPR-Var	0.015 to 0.028, in steps of 0.001
GPR-Mean	-0.984 to -0.997, in steps of 0.001
1SVM	0.012 to 0.04 , in steps of 0.002
KNFST	0.001 to 0.025, in steps of 0.002

5.3. Evaluation Criteria

In most of the published works including [13], the performance measures used are subjective and do not lend themselves to comparison. The reason is that in problems like temporal video segmentation, it is very difficult to define a good performance measure for the algorithms. Hence, we concentrated on the fact that our algorithm works on the principle of detecting events and use performance measures for event detection. To evaluate the results of the algorithm qualitatively, we used the detection rate η:

$$\eta = \frac{\text{number of correct detections}}{\text{number of events in ground truth}} \quad . \tag{11}$$

It is often the case that the detection of the algorithm and the ground truth vary by about 20-25 frames, because the algorithm makes hard decisions using a threshold and ground truth is marked by human observers. This is not a serious problem, since in real-life videos 25 frames corresponds to a time span of 1 second, in which generally not many events happen. For most applications, this difference is not a major problem.

Additionally, over-segmentation is expected, because our algorithm works completely unsupervised. As the threshold is set without any prior knowledge about the video and is thus completely independent of it, often very small and insignificant changes in the video result in a new segment being reported. However, this is also not a cause for alarm as this stage is usually intended to be followed by a higher processing stage or a human observer in most applications. Thus, at the higher level, we can choose to ignore these particular extra segments in a post-processing step. We represent the effect of over-segmentation with the over-segmentation ratio γ:

$$\gamma = \frac{\text{number of false detections}}{\text{number of events in ground truth}} \quad . \tag{12}$$

This represents the average number of extra segments for every segment in the ground truth.

Table 3. Results on the thermal video subset of the CVPR change detection dataset for the TSSM approach using detection rate η (Eq. (11)) and over-segmentation ratio γ (Eq. (12)).

Video	η	γ
corridor	0.95	0.94
diningRoom	1.00	0.55
lakeSide	0.65	1.76
library	1.00	4.56
park	1.00	0.50

Table 4. Results in terms of detection rate η (Eq. (11)) and over-segmentation ratio γ (Eq. (12)) on the thermal video subset of the CVPR change detection dataset using the GPR-Var method.

Video	η	γ
corridor	0.68	0.49
diningRoom	0.9	0.48
lakeSide	0.23	0.05
library	1	1.67
park	0	0

5.4. Results

As can be seen in Tab. 3, the results of the TSSM approach show quite a good performance. However, this algorithm works in an offline mode, limiting its applicability. Therefore, we need the OCC methods, which offer the dual advantages of online capabilities and the ability of trade-off accuracy and over-segmentation ratio. Table 4 shows the results for each video in the dataset, using the GPR-Var OCC method. The parameters are set as follows: $F = 50$ and $T = 0.02$.

The result for the *park* video is not very promising at the first look. No event is detected in this video at this threshold level. The reason is that contrast in this video is very low and the events are detected only when the threshold is raised much higher. For example, at a threshold of $T = 0.28$, we have $\eta = 0.17$ at $\gamma = 0.17$. Furthermore, the KNFST algorithm performs better here, achieving $\eta = 0.83$ at the same over-segmentation ratio (see Fig. 8). The results for the *lakeSide* video appear to be poor compared to the other videos. It is heavily under-segmented, *i.e.*, many events are missed. This is due to the fact that the video has extremely low contrast and even for a human observer it is very challenging to locate

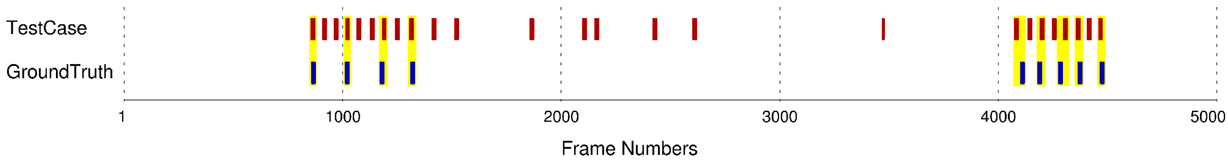


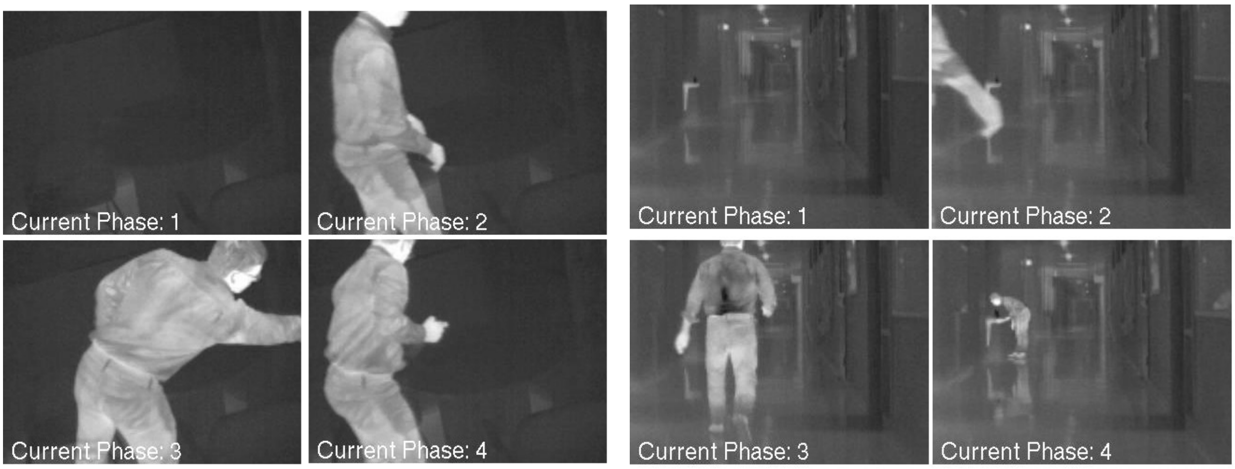
Figure 6. Segmentation timeline for the *library* video (with $T = 0.175$ and $F = 50$), with the ground truth. The yellow patches indicate the matched detections (the figure is best viewed in color)

the events. In addition, the events and objects in the video are of very small size. Again, to detect events in this scenario, one has to set the threshold extremely low, which would increase the false detection rate in other generic cases. On the other hand, we see that the *library* video is over-segmented, which is a result of a threshold value that is too low.

Often, it is even desirable to have over-segmented videos, *e.g.*, in the *library* video, the detected extra events are basically the person under observation turning pages. These are not labeled in the ground truth because they are minor events but could be interesting for the application. In the *diningRoom* video, the extra events detected are basically the person turning, which is an interesting change but again not labeled in the ground truth. Furthermore, a higher processing stage or a human observer is generally present after segmentation and that stage may be used to handle over-segmentation, whereas a missed event is a more difficult problem to tackle.

Figure 6 shows a segmentation timeline for the *library* video and Fig. 7 shows segmentation example frames for *library* and *corridor* videos. Similar to ROC curves showing true positive detections against false positive detections, we plot η-vs.-γ curves as can be seen in Fig. 8 for the five videos. We have varied T as in Tab. 2.

Figures 9 and 10 show the changes in η and γ as T is varied for the case of *corridor* and *diningRoom* videos. The parameter F is set to 50, and the OCC method used is KNFST. Smaller values of the threshold result in an over-segmented video, *i.e.*, high γ, whereas larger values result in under-segmented videos. Clearly, there is the expected strong correlation between η and γ, and the parameter T is an important factor in trading-off between the two. An interesting point to note is that at some points, we sometimes obtain slightly lower detection rates η at higher values of γ than those at lower γ (*e.g.*, for the *corridor* video, as the threshold is increased from 0.015 to 0.02). This is due to the training time F. Often, an event is detected when there is none (*i.e.*, over-segmentation) and during the following training time, an actual event is missed.



(a) *library* (b) *corridor*

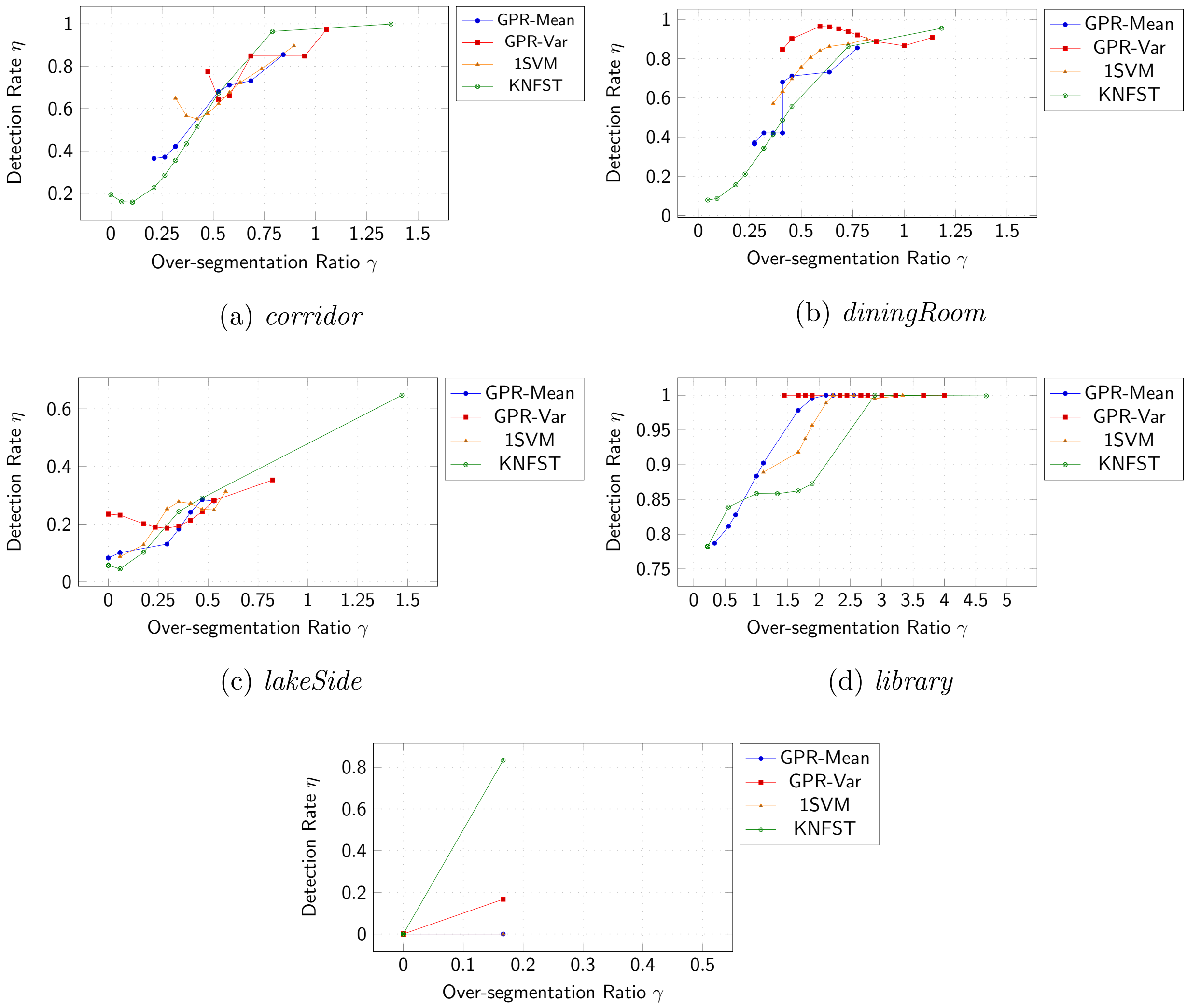
Figure 7. Segmentation results on the *library* and the *corridor* videos.

The effect of the parameter F can be observed in Fig. 11 and Fig. 12. For this, we used the KNFST-based OCC method and the parameter T is set to 0.02 for the *corridor* and *diningRoom* videos. Clearly, the parameter F does not have such a large impact on the detection rate compared to T. Other than that in some cases, if an event occurs sooner than the first F frames after the previous event, that event will be missed. Also, as in the case of the *diningRoom* video, the events happen in quick succession and the parameter F doesn't have a predictable effect on the performance. A similar trend can be observed in the over-segmentation ratio. Overall, a value of F chosen too large clearly results in poor performance. Such high values lead to models that cover a large amount of variations and detect only very large deviations as events. On the other hand, very low values result in models that don't cover the corresponding segment well enough and result in larger number of segments. Thus, the often unpredictable effects of this parameter make it unsuitable for managing the trade-off between η and γ and generally, our experiments show that setting F to some value between 40 and 60 and varying T is a better solution.

Overall, the results are promising and there are strong indications that a video dependent threshold determination will solve most of these existing issues.

6. CONCLUSIONS AND FUTURE WORK

Our work was aimed at segmenting the video sequences into activity phases in an unsupervised way, enabling us to further process the smaller units and extract the interesting parts. We used a one-class classification approach, comparing various well-known methods



(a) *corridor*

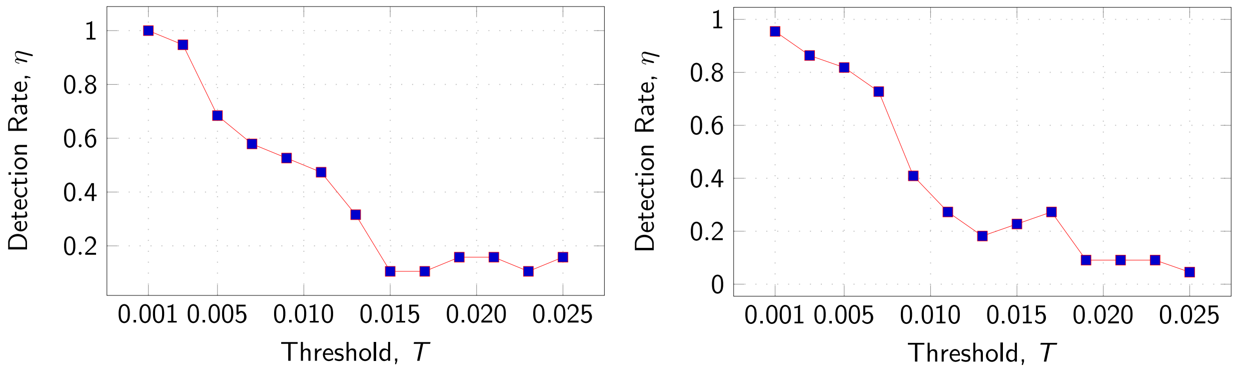
(b) *diningRoom*

(c) *lakeSide*

(d) *library*

(e) *park*

Figure 8. Change point detection rate η vs. over-segmentation ratio γ.



(a) *corridor*

(b) *diningRoom*

Figure 9. Effects of the parameter T on detection rate η for the *corridor* and *diningRoom* videos.

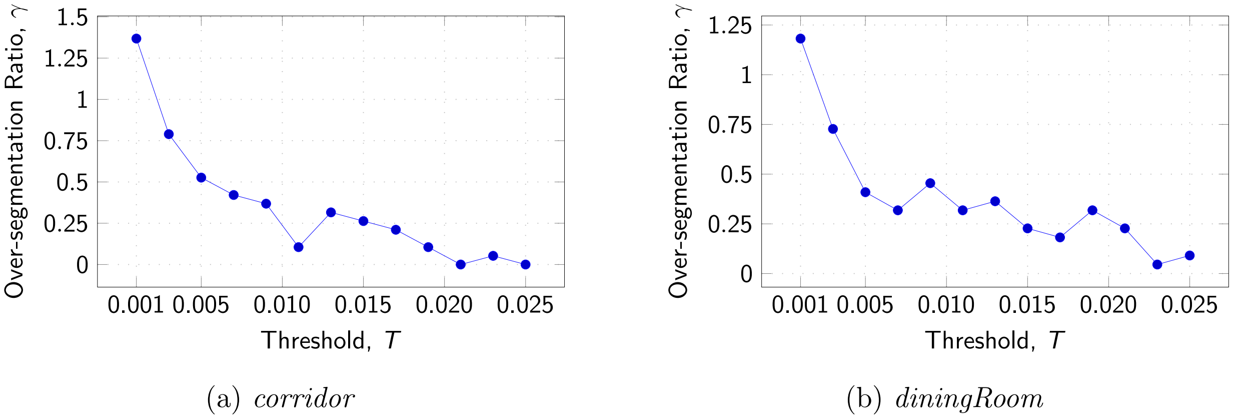


Figure 10. Effects of the parameter T on over-segmentation ratio γ for the *corridor* and *diningRoom* videos.

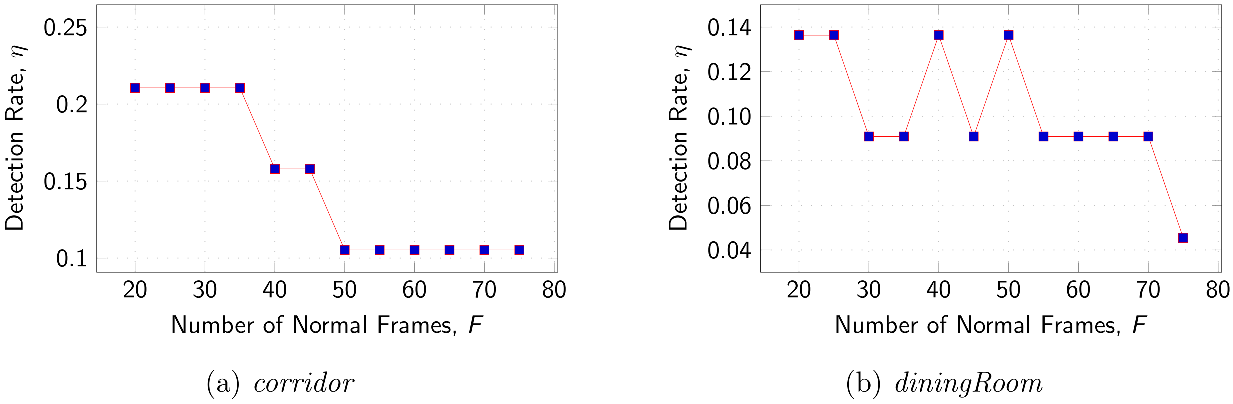
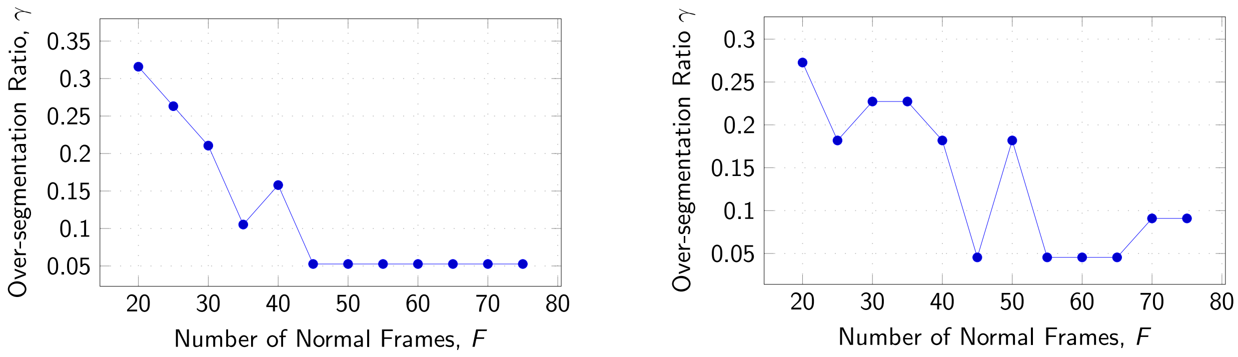


Figure 11. Effects of varying the parameter F on the detection rate η for the *corridor* and *diningRoom* videos.

to do this successfully as seen in the previous section. We also tested an approach based on the temporal self-similarity maps and achieved good performance. The accuracy in event detection is quite impressive even with the simple PHOG features used in this implementation. This demonstrates the possibility of using one-class classification schemes, especially KNFST and GPR-Var, for the task of temporal video segmentation.

As noted earlier, the accuracy of the system can be further enhanced by intelligent selection of parameters. Automated parameter optimization is one topic of future work. This may yield better results because then the parameters will be dependent on the video instead of being universal and it may avoid situations as the one encountered in the case of the *lakeSide* video.

Furthermore, feature selection should be a part of further research. The authors believe that feature selection is of critical importance in this case and the use of more sophisticated features could drastically improve the performance of the approach. To exploit features



(a) *corridor*

(b) *diningRoom*

Figure 12. Effects of varying the parameter F on the over-segmentation ratio γ for the *corridor* and *diningRoom* videos.

representing shape and motion in a combined descriptor is a promising idea for future work.

7. ACKNOWLEDGEMENTS

Mahesh Venkata Krishna is funded by a grant from the Carl Zeiss AG, through the "Pro-Excellence" scholarship of the federal state of Thuringia, Germany. We thank Stefan Saur and Christian Wojek for useful discussions and suggestions.

1. P. Bodesheim, A. Freytag, E. Rodner, M Kemmler, and J. Denzler. Kernel null space methods for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, page (accepted for publication), 2013.
2. J.S. Boreczky and L.D. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3741–3744 vol.6, 1998.
3. A. Bosch, A. Zisserman, , and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval (CIVR'07)*, pages 401–408, 2007.
4. C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 2011.
5. M. Cooper, T. Liu, and E. Rieffel. Video segmentation via temporal pattern classification. *IEEE Transactions on Multimedia*, 9(3):610–618, 2007.

6. R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):781–796, 2000.
7. Goyette, Jodoin N., Porikli P.-M., J. F., Konrad, and P. Ishwar. changedetection.net: A new change detection benchmark dataset. In *Proceedings of the IEEE Workshop on Change Detection (CDW'12) at CVPR'12*, 2012.
8. J. S. Iwanski and E. Bradley. Recurrence plots of experimental data: To embed or not to embed? *Chaos*, 8(4):861–871, 1998.
9. I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):172–185, 2011.
10. M. Kemmler, E. Rodner, and J. Denzler. One-class classification with gaussian processes. In *Proceedings of the Asian Conference on Computer Vision (ACCV'10)*, pages 489–500, 2010.
11. Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477 – 500, 2001.
12. Marco Körner and Joachim Denzler. Temporal self-similarity for appearance-based action recognition in multi-view setups. In *Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2013. (to appear).
13. Tianming Liu, Hong-Jiang Zhang, and Feihu Qi. A novel video key-frame-extraction algorithm based on perceived motion energy model. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(10):1006 – 1013, 2003.
14. S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1–8, 2008.
15. G. McGuire, N. B. Azar, and M. Shelhamer. Recurrence matrices and the preservation of dynamical properties. *Physics Letters A*, 237(1–2):43–47, 1997.
16. F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180, 2005.
17. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.
18. B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
19. P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso. Tem-

poral video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, 2011.

20. Deborah Swanberg, Chiao-Fe Shu, and Ramesh C. Jain. Knowledge-guided parsing in video databases. pages 13–24, 1993.
21. D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.
22. Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2):119–128, 1999.
23. HongJiang Zhang, Atreyi Kankanhalli, and StephenW. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

Mahesh Venkata Krishna, born in 1984, received the Bachelor degree in Telecommunications Engineering in 2006 from the Visvesvaraya Technological University, India and obtained the MSc degree in Communication Engineering from the RWTH, Aachen in 2011. He is currently a holder of a scholarship from the Graduate Academy for Image Processing of the Free State of Thuringia, Germany, funded by Carl Zeiss AG. He is a member of the Computer Vision Group of Joachim Denzler at the Friedrich Schiller University, Jena. His research interests include video analysis, event detection, extraction of rules of a scene based on visual data etc.

Paul Bodesheim, born in 1987, received the Diploma degree in Computer Science (“Diplom-Informatiker”) in 2011 from the Friedrich Schiller University Jena, Germany. He is currently a holder of a scholarship from the Graduate Academy of the University Jena partially funded by the Free State of Thuringia, Germany (“Landesgraduiertenstipendium”) and a PhD student in the Computer Vision Group of Joachim Denzler at the University Jena. His research interests are in the field of computer vision and machine learning, especially one-class classification and novelty detection as well as incremental, large-scale, and life-long learning for visual object category recognition.

Marco Körner, born in 1984, received the Diploma degree in Computer Science ("Diplom-Informatiker") in 2008 from the Friedrich Schiller University Jena, Germany. He is currently a PhD student at the Computer Vision Group of Joachim Denzler at the University Jena. His research interests are in the field of 3d computer vision and machine learning, especially action recognition in multi-sensor environments.

Joachim Denzler, Joachim Denzler earned the degrees "Diplom-Informatiker", "Dr.-Ing.," and "Habilitation" from the University of Erlangen in the years 1992, 1997, and 2003, respectively. Currently, he holds a position of full professor for computer science and is head of the Chair for Computer Vision, Faculty of Mathematics and Informatics, Friedrich-Schiller-University of Jena. His research interests comprise active computer vision, object recognition and tracking, 3D reconstruction, and plenoptic modeling, as well as computer vision for autonomous systems. He is author and coauthor of over 200 journal and conference papers as well as technical articles. He is a member of the IEEE, IEEE computer society, DAGM, and GI.