International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:4, 2012

Video Classification by Partitioned Frequency Spectra of Repeating Movements

Kahraman Ayyildiz and Stefan Conrad

Abstract—In this paper we present a system for classifying videos by frequency spectra. Many videos contain activities with repeating movements. Sports videos, home improvement videos, or videos showing mechanical motion are some example areas. Motion of these areas usually repeats with a certain main frequency and several side frequencies. Transforming repeating motion to its frequency domain via FFT reveals these frequencies. Average amplitudes of frequency intervals can be seen as features of cyclic motion. Hence determining these features can help to classify videos with repeating movements. In this paper we explain how to compute frequency spectra for video clips and how to use them for classifying. Our approach utilizes series of image moments as a function. This function again is transformed into its frequency domain.

Keywords—action recognition, frequency feature, motion recognition, repeating movement, video classification

I. Introduction

VIDEO annotation and action recognition plays a central role in computer vision, since video surveillance systems, human-computer interfaces, or motion recognition software have a growing interest. Beside the World Wide Web with online video portals [6] and online video stores [1] there are many other branches and institutions with large video archives. Some examples are museums, the media branch, major corporations, or governmental institutions. All these areas need efficient algorithms for classifying the amount of existing clips or videos automatically.

Provider sided classification ensures quality of indexing process, but has high costs. On the other hand user sided classification has low costs without ensuring quality. Both approaches usually annotate videos with general descriptions, which capture the mean content. Beyond this mean content videos contain activities, which are not captured by common annotations. An automatic indexing and annotating process can reduce costs, capture specific information and ensure quality. A bulk of work regarding automatic video classification exists in literature. Nevertheless video classification by frequencies of repeating movements is sparsely documented.

In this work we explain an approach, which uses frequency features from cyclic motion in videos for classification. As a first step regions with movement are detected for each frame. These regions lead to image moments for each frame, where a series of image moments represents a function. This function again matches one moment to each frame of a video. Transforming this function via fast Fourier transform (FFT) spans a frequency spectrum. A partitioning of this spectrum into intervals of same length reveals different average amplitudes for each interval. Combining average amplitudes gives a multidimensional feature vector for each clip. Once feature vectors are determined, a classifier can assign each clip to a class.

Our approach is related to our previous research work. In [2] we describe an approach, which utilizes main frequencies of repeating motion. The main difference to this work is the feature extraction stage. In [2] we extract just two to six frequencies with maximal amplitudes. In this paper we focus on the whole frequency spectrum and the amplitude height has a stronger affect on classification process. These differences lead to more accurate test results than in [2].

The following section gives an overview of the entire video classification process. It describes how features are extracted and used by classifiers. Moreover we define image moments and how to derive so-called *1D-functions* from these moments in section III. In section IV we define *AAFIs* as the basic of feature vectors (Average Amplitudes of Frequency Intervals). Afterwards in section V our radius based classifier *RBC* is introduced and explained. In section VI we evaluate the idea of video classification by AAFIs. Next section VII discusses and compares work related to our approach. The last section reviews the proposed methods.

II. Classifying Videos by Frequency Spectra

In this section we explain methods used for our approach, where fig. 1 offers an overview of the different stages.

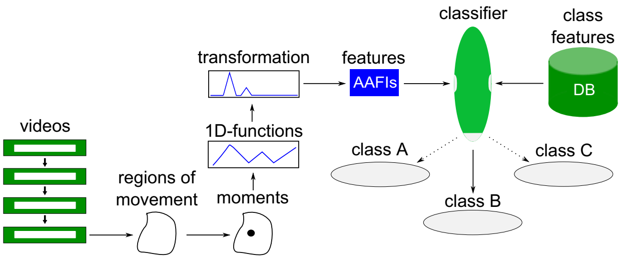

Fig. 1: Flow diagram of whole classification process

The goal of the whole classification process is to classify video sequences with repeating movements properly. Some examples for activity with repeating movement are jumping, playing tennis or hammering. At first regions of movement are detected in every clip for each frame. Region detection

Kahraman Ayyildiz is research assistant at the Department of Databases and Information Systems, Institute of Computer Science, Heinrich Heine University, Duesseldorf, 40225 Germany (e-mail: kahraman.ayyildiz@uni-duesseldorf.de).
Stefan Conrad is full professor at the Department of Databases and Information Systems, Institute of Computer Science, Heinrich Heine University, Duesseldorf, 40225 Germany (e-mail: conrad@uni-duesseldorf.de).

is performed by measuring the color differences of pixels in two consecutive frames (see section III-A). These regions give rise to image moments, where our approach applies two types of moments: centroids and pixel variances. Series of these moments are considered as 1D-functions, which represent the motion in a clip. Transforming 1D-functions via FFT gives the frequency domain. Partitioning the frequency axis into intervals of same length, average amplitudes for each interval can be calculated. We name these averages *AAFIs* (Average Amplitudes of Frequency Intervals). AAFIs constitute the feature vector of a clip with regard to its motion. Once the feature vector of a video is determined, a classifier decides to which class this video fits best by comparing its feature vector to feature vectors of other videos stored in database.

III. Image Moments and 1D-functions

So as to compute frequency domains for video scenes motion has to be localized frame by frame. Image moments and resulting 1D-functions depend on this motion. Next we explain how to detect regions of motion and how to derive 1D-functions from these regions.

A. Regions of Motion

In fig. 2 we illustrate the detection of regions with motion by analyzing one of our clips. We see two consecutive frames with a person troweling a wall. Color differences between these frames are measured for each pixel. A pixel is considered to be a pixel which is part of a movement, if two conditions are fulfilled: Firstly the color difference of a pixel has to exceed a predefined threshold. Secondly there have to be enough neighbor pixels with a color difference beyond the same threshold. Thus we define a region of motion as the affiliation of pixels with motion.

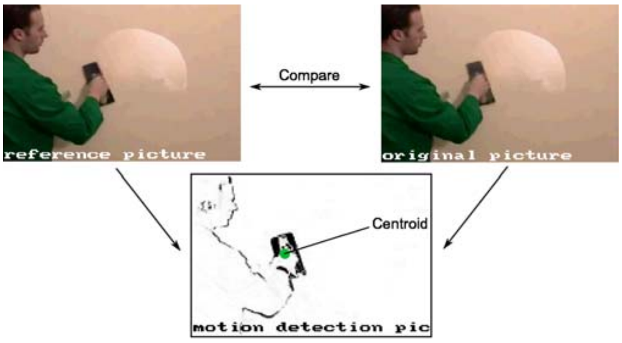

Fig. 2: Regions with pixel activity and centroid

The binary image below the two frames compared shows the regions with movement. Further on the centroid of regions with motion lies exactly on the right hand, because the trowel, the right hand, and the right arm of the home improver represent the main motion in the frame. Hence the centroid follows the movement of the troweling.

B. Image Moments

The weighted average of pixel intensities of a picture is called image moment. An image moment can describe the area, the bias, or the centroid of segmented image parts. Two main types of image moments do exist: raw moments and central moments. Raw moments are sensitive to translation, whereas central moments are translation invariant. For a two dimensional binary image $b(x, y)$ and $i, j \in \mathbb{N}$ a raw moment M_{ij} is defined as follows [14]:

$$M_{ij} = \sum_x \sum_y x^i \cdot y^j \cdot b(x, y) \tag{1}$$

M_{ij} is always of the order $(i + j)$. Hence M_{00} determines the area of segmented parts, where $(\bar{x}, \bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00})$ defines the centroid of segmented parts. Now central moments can be figured by applying centroid coordinates [14].

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i \cdot (y - \bar{y})^j \cdot b(x, y) \tag{2}$$

Here μ_{20} and μ_{02} represent the variances of pixels with regard to x and y coordinates, respectively.

C. Deriving 1D-functions

We define a 1D-function f as a series of one-dimensional moment values. This series corresponds to the order of frames in a video and depends on time t, which leads to function $f(t)$. Let $(\bar{x}_t, \bar{y}_t) = (M_{10_t}/M_{00_t}, M_{01_t}/M_{00_t})$ for the coordinates of a centroid with respect to time t. Then function $f_c(t) = (\bar{x}_t, \bar{y}_t)$ implies:

$$f_{c_x}(t) = \bar{x}_t \wedge f_{c_y}(t) = \bar{y}_t \tag{3}$$

In section VI $f_{c_x}(t)$ and $f_{c_y}(t)$ are used for experimental test series instead of $f_c(t)$, because transforming 1D-functions results in better accuracies than transforming 2D-functions. For the same reason two separate 1D-functions of central moments are implemented and tested:

$$f_{v_x}(t) = \mu_{20_t},\ f_{v_y}(t) = \mu_{02_t} \tag{4}$$

For any 1D-function $f(t)$ we define the speed of an image moment at time t as follows:

$$f_s(t) = |f(t) - f(t-1)| \tag{5}$$

The direction of a moment at time t is defined by 6.

$$f_d(t) = \begin{cases} +1, & \texttt{if } f(t) - f(t-1) > 0 \\ 0, & \texttt{if } f(t) - f(t-1) = 0 \\ -1, & \texttt{if } f(t) - f(t-1) < 0 \end{cases} \tag{6}$$

IV. AAFIs as Feature Vectors

This section explains how 1D-functions lead to feature vectors for clips. As already mentioned each 1D-function can be transformed to its frequency spectrum by FFT. Partitioning this spectrum into intervals of same length, an average amplitude for each interval can be stated.

Fig. 3 depicts this idea by partitioning a frequency spectrum with a length of $m = 256$ units to $n = 8$ intervals. Using the fast Fourier transform variables m and n have to be a power of 2, where $m \geq n$. Further the horizontal, orange lines mark the average amplitude of each interval. Hence with regard to fig. 3 one 1D-function leads to 8 average amplitudes respectively to one 8-dimensional feature vector. Due to the fact, that videos produce two 1D-functions, each video is described by two 8-dimensional feature vectors in this example. Thus a partitioning of the frequency spectrum into n intervals results in a $(2 \cdot n)$-dimensional feature vector for each video.

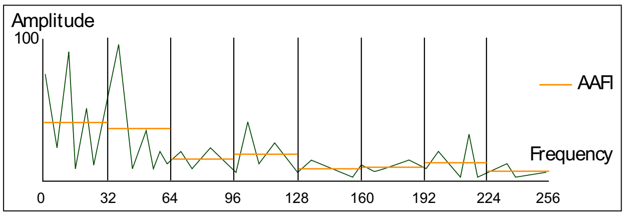

Fig. 3: Average amplitudes of frequency intervals (AAFIs)

Comparing to our previous work [2] feature vectors reveal much more information about the motion type. In [2] we used up to 6 frequency maxima for each video as feature vector, now the whole frequency spectrum is described by AAFIs.

V. Radius Based Classifier

During our experimental phase we built up a novel classifier, which turned out as very effective. It uses a predefined radius around a tested object in order to calculate distances. Because of the importance of this radius we call our classifier *Radius Based Classifier* (RBC).

A. Basic Idea

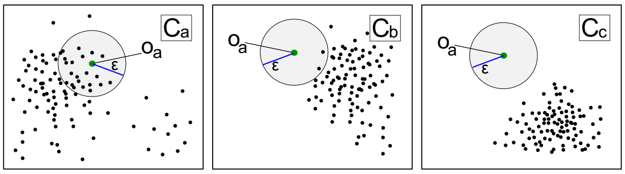

Fig. 4: Classifying with RBC

Fig. 4 illustrates how the RBC works: An object $o_a \in B$ has to be classified. Therefore it is assigned to each existing class in order to calculate the class with the smallest distance $dist(o_a, C_i)$. There are three different example classes C_a, C_b, $C_c \in C$, where each class has its own typical object distribution. Assigning o_a to class C_a reveals, that there are many objects within radius ε. In class C_b only 2 objects are present inside the given metric. Objects of class C_c are far away from o_a, so there is no object of this class within radius ε.

According to these three classes, o_a fits best into class C_a, because it is part of the typical object distribution. At the same time this fact leads to the smallest distance.

B. Formalization

First we define $C = \{C_1, \ldots, C_m\}$ as our set of classes. Each class $C_i \in C$ contains a set of objects, so we define $C_i = \{o_{i_1}, \ldots, o_{i_{n_i}}\}$, $C_i \neq \{\}$ and $C_i \cap C_j = \{\}$ for $i \neq j$. The total of all objects in classes constitutes our training set $A = C_1 \cup \ldots \cup C_m$. Test set objects in $B = \{o_1, \ldots, o_l\} \neq \{\}$ with $A \cap B = \{\}$ do not belong to any class.

For a given object $o_b \in B$, a class $C_i \in C$ and a radius ε the ε-neighborhood $N_\varepsilon(o_b, C_i)$ encloses all objects of class C_i with a distance to o_b smaller than the radius. Furthermore the distance between two objects is measured by Euclidian distance.

$$\begin{aligned} &N_\varepsilon(o_b, C_i) = \\ &\{o_s | o_s \in C_i \wedge dist_{euclid}(o_b, o_s) < \varepsilon\} \end{aligned} \tag{7}$$

Based upon $N_\varepsilon(o_b, C_i)$ the distance between an object o_b and a class C_i is computed:

$$dist(o_b, C_i) = 1 - \frac{|N_\varepsilon(o_b, C_i)|}{|C_i|} \tag{8}$$

Hence the resulting distance lies in interval $[0, 1]$, where 0 means all objects of one class lie inside ε. On the other side 1 means there is no object within the ε-neighborhood of o_b. Utilizing 8 a class with a minimal distance to o_b can be found.

$$\begin{aligned} &cl_{rbc}(o_b, C) = \\ &\{C_i \in C | \forall C_j \in C : dist(o_b, C_i) \leq dist(o_b, C_j)\} \end{aligned} \tag{9}$$

In best case $cl_{rbc}(o_b, C)$ reveals exactly one next class. Then the RBC assigns o_b to this class. If $cl_{rbc}(o_b, C)$ reveals more than just one class, because of equal minimal distances to o_b, one of these classes is chosen at random.

VI. Experiments

We have analyzed all exposed methods through three steps. First, experiments regarding moment type, moment speed and moment direction are considered. In addition these experiments include different interval sizes. Second, translation invariance of motion classification is discussed and analyzed. Third, the runtime of our approach is evaluated. Here class sizes, the amount of classified videos and again interval sizes play an important role.

In both of the first two subsections firstly test series with own video data and secondly test series with external video data from the online video database *youtube.com* are realized [15]. Furthermore tests with own video data are calculated by m-fold cross validation. 10 classes, where each class consists of 20 videos, are tested (total 200 videos). External video data is tested by assigning clips to own classes, because cross validation was not possible due to classes with just few clips (total 102 videos). Each video shows one of the following 10 home improvement activities: filing, hammering, planing, sawing, screwing, using a paint roller, a paste brush, a putty knife, sandpaper and a wrench.

A. Raw Moments and Central Moments

In fig. 5 an example of a 1D-function and its transformation is illustrated. The upper plot shows a 1D-function of a clip with a person using a wrench. This function regards to the x-axis coordinate of centroids. One can see, that the centroid moves from left to right and vice versa, which corresponds to the movement of the person. Below this 1D-function its transformation to the spectral domain is plotted. A partitioning of the frequency axis into $m = 32$ intervals leads to 32 AAFIs. These AAFIs capture the mean information of the frequency domain without considering every single unit. There are significant highs and lows at certain frequencies and wide ranges with constantly high respectively low amplitudes, which are all captured by AAFIs and resulting feature vectors.

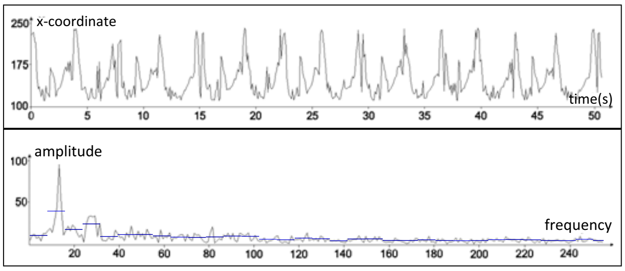

Fig. 5: FFT of a 1D-function: Above 1D-function of a person handling a wrench, bottom FFT of this action

The two line charts in fig. 6 focus on results of test series with raw moments (centroids) regarding interval sizes. Moreover results of tests with directional and speed information of moments are presented. The left chart refers to tests with video data, which was produced especially for our experiments. The right chart relates to external video data from an internet database [15]. At first glance it becomes apparent, that recorded video data results in better accuracies than external video data. This is associated with the fact, that the external videos have lower quality in the sense of regular movement, camera positioning and scaling. Furthermore for both data sources the optimal interval size lies between 4 and 32.

For recorded video data and directional information of centroids our approach achieves a maximal accuracy of 0.87 at an interval size of 8. That means 174 videos of 200 videos are assigned properly. Experiments with external videos and centroid coordinates result in a maximal accuracy of 0.40 at an interval size of 4. Here 41 of 102 clips are assigned correctly.

In the case of external videos centroid coordinates achieve higher accuracies than centroid directions, because external videos contain more irregular movements. For both data sources the speed information of centroids is a weak feature for classifying.

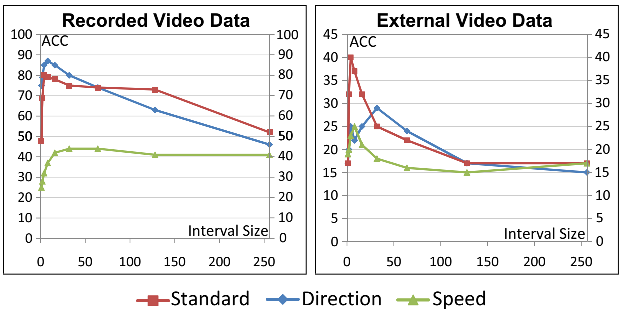

Fig. 6: Accuracies of tests with raw moments

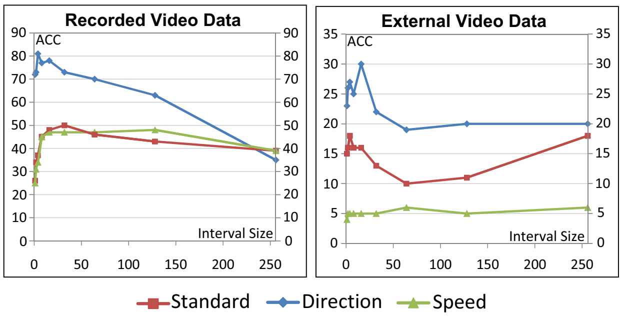

Fig. 7: Accuracies of tests with central moments

Next we see in fig. 7 results of tests with central moments (variances). Here own videos result in higher accuracies than external videos, too. But on the whole accuracies for both data sources decrease comparing with results of raw moments. For own video data and directional information of central moments a maximal accuracy of 0.81 is achieved at an interval size of 4. In contrast to fig. 6 standard and speed information have similar line progressions here. Both lines do not exceed an accuracy of 0.50. Compared with centroids utilizing variances for spectral analysis is a weak approach, because motion inside a scene cannot be tracked properly by variances. However directional information of variances (increase or decrease) is almost as effective as directional information of centroids, because repeating movements of one class can produce very different variances, but the directional information of these variances is mostly similar.

Considering external video data there is a accuracy peak at an interval size of 16 for directional information. The accuracy amounts to 0.30. Standard central moments reveal accuracies between 0.10 and 0.18, speed information of central moments leads to accuracies between 0.04 and 0.06.

B. Translation Invariance

Different positions of one activity in different videos have no effect on classification process (translation invariance). But motion areas shifted within one video have an effect on classification process. Fig. 8 shows how accuracies change in this case. The translation takes places for each classified clip frame by frame. Furthermore tests with different shift velocities and shift directions are plotted. Again own and external video sources are integrated. Tests with own videos are performed via directional and tests with external videos are realized via standard information.

For own videos and centroids the accuracy decreases slightly by increasing velocity of translation, if horizontal or vertical shift of motion is realized. Here accuracy starts at 0.87 and ends at 0.75 for vertical respectively 0.72 for horizontal shift. Moreover accuracies of a diagonal translation decrease rapidly. Starting at an accuracy of 0.87, the accuracy ends up at 0.16. These results differ apparently from horizontal or vertical shifts behavior. The reason for that is, shifting a centroid along just one axis does modify just one coordinate. Unmodified coordinates result in unmodified feature vectors. The yellow line shows the accuracy for central moments (variance). For each translation type and velocity the accuracy stays constantly at 0.81.

Considering test series with external data, it becomes clear, that accuracies react very sensitive on translation. At the beginning each curve falls abruptly. Then the curves for horizontal and vertical translation stay constantly at 0.27 and 0.23. The accuracy curve for diagonal shift ends at 0.17. There are two reasons for this abrupt decrease: First, external videos depend much more on just one 1D-function than own videos. Second, tests with standard moments are more sensitive to translation than directional information of moments. On the other side here central moments lead to constant accuracies, too. For any translation type and velocity the accuracy is 0.30.

According to these experiments it can be stated, that clips with moving objects or moving cameras can often be classified more accurate with central moments than with raw moments.

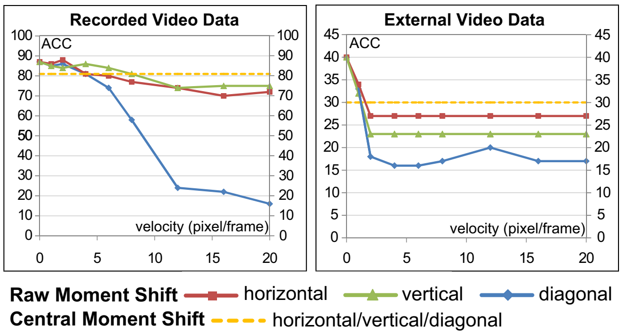

Fig. 8: Accuracies for moments with translation

The stated accuracies in this and the previous subsection result from both selected features and RBC. In further test series, which are not explicitly listed in this work, we compared the RBC with Bayes Classifier, KNN Classifier and Average Link Classifier. We detected that the RBC improves results, but does not affect the relative highs of feature accuracies.

C. Runtime Analysis

After analyzing the accuracy of our system, we focus on runtime performance. Therefore tests with respect to referenced class size, interval size of AAFIs and the amount of classified videos are performed. All experiments are realized by a standard PC with a 2.4 GHz CPU.

Fig. 9 shows the runtime for several class sizes and interval sizes. Analyzing clips with a length of 512 frames via FFT, the frequency axis consists of 256 units. Hence an interval size of 1 unit results in 256 intervals and an interval size of 256 units results in 1 interval.

All line charts in fig. 9 illustrate runtime in seconds for 200 classified clips. There is an increase for each line, when the amount of intervals is rising. At the same time this increase becomes clearer, if the class size of referenced classes increases. This relates to the fact, that the distance of each classified clip to a class is calculated by involving distances to each class object. Hence big class sizes combined with big interval sizes have just little effect on runtime. But big class sizes combined with small interval sizes does have a clear effect on runtime.

For a referenced class size of 200 videos and utilized one interval, the runtime is 859 seconds. Moreover for 256 intervals the runtime is 937 seconds. A referenced class size of 1000 videos combined with just one interval results in 922 seconds. In addition 1486 seconds are needed for 256 intervals. There is a logarithmic growth for increasing amount of intervals.

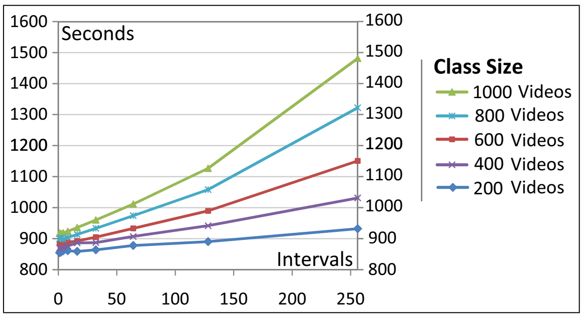

Fig. 9: Runtime with regard to number of intervals and class sizes

In fig. 10 we see a linear growth of runtime. The bar diagram regards to test series with an increasing number of classified clips. In each test the number of intervals is 256 and the number of videos in referenced classes is 200.

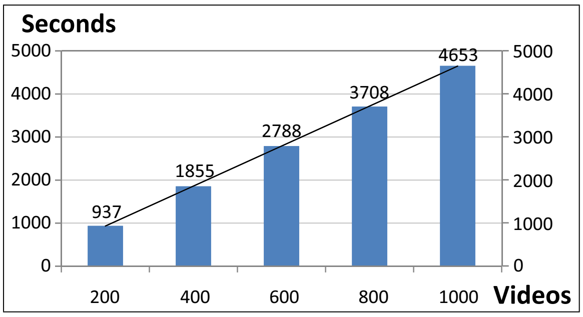

Fig. 10: Runtime with regard to the number of classified videos

Beginning at 937 seconds for 200 classified clips, the runtime peaks at 4653 seconds for 1000 clips. The average time needed for classifying one video is 4.65 seconds. We see, a high amount of classified clips has a stronger impact on runtime than a high amount of clips in referenced classes.

VII. Related Work

Videos reveal a huge amount of information. Hence video annotation and classification can be realized in many different ways. A *key-frame* based approach for example is introduced by Pei and Chan in [11]. After detecting scene changes in videos key-frames are figured out for each scene. Key-frames lead to feature vectors for each frame and scene. Another approach is presented by Lienhart in [8]. Here videos are retrieved by texts within video shots. In [10] Patel and Sethi describe a method, which is able to detect dialog scenes in films by analyzing the audio signal.

Research work considering motion focuses mainly on the gait or the gestures of humans. In [3] the authors use the flow and the strength of change of pixels as features in order to determine the motion type of human body parts. They test seven different action classes by classifying 51 test samples. Accuracies from 0.94 to 0.98 are achieved for different classifiers. This approach is able to handle motion in one direction, but repeating movements cannot be recognized. Moreover in [13] repeating motion of human body parts is analyzed by tracking Moving Light Displays (MLD). Pieces of curves described by these MLDs are used as reference patterns for each clip. We included this idea in our classification system in order to compare with AAFIs as feature vectors. Results have shown that repeating motion patterns are a weak solution, since patterns of the same motion can vary drastically, whereas the frequency is more reliable. He and Debrunner calculate Hu Moments for regions with motion in each frame and count the number of frames until a Hu Moment repeats [7]. This number is defined as frequency. The authors yield high accuracies for three classes tested, where each class consists of 16 samples. Because of the reason that Hu Moments are translation invariant, here the periodic trajectory of an object cannot be ascertained. Further on only one frequency can be captured by this method. Another method is proposed by [12]. In this paper the authors divide each frame of a clip into 16 parts of same size, where 6 frames for each repeating motion are stored. Pixel activities of these 6 x 16 parts give rise to the motion type. Experiments with seven classes and 40 samples in total result in high accuracies, but this approach is strongly sensitive to scaling.

Some research work is strongly related to our frequency domain based approach. For instance in [9] again MLDs are utilized, but here the frequency peaks of transformed MLD curves are considered as features of cyclic motion. The classification of different motion types results in high accuracies from 0.84 to 0.96. Unfortunately authors miss to give rise about the size of tested dataset. Cheng et al. analyze sports videos in [4]. Series of horizontal and vertical pixel motion vectors are transformed and result in two main frequencies for each clip. Again authors state high accuracies for five analyzed sports activities, but the average class size is three and therefore not convincing. Davis and Cutler provide a method, which is able to capture all significant frequency peaks [5]. They obtain frequency domain by transforming measured self-similarity of motion as it evolves in time. Experimental results depict an accuracy of 1.0 for each of three tested classes.

As this comparison to related research shows, our approach remedies deficits of other methods and offers distinct experimental results.

VIII. Conclusion

Previous sections of this paper depict a video classification and action recognition system based on repeating movements. Repetitions of movements lead to frequency spectra by transforming 1D-functions of image moments. Beside different 1D-functions we defined, we explained how frequency spectra can be utilized for feature extraction and video classification. Therefore we introduced AAFIs, which are a strong approach for classifying videos. In addition a novel radius based classifier was presented, which improved the performance of the system.

The experimental stage exposed, that our approach works accurately for centroids as image moments. A maximal accuracy of 0.87 could be measured for recorded video data. For external videos the maximal accuracy was 0.40. On the other side for videos including translation of motion translation invariant central moments work more efficient. Here the highs at 0.81 for own videos and at 0.30 for external videos stay constantly at the same level for each shift velocity of motion. Considering interval sizes, experiments have shown that an interval size between 4 and 16 for AAFIs gives the best results. Furthermore runtime tests with big class sizes combined with small intervals increase the runtime apparently.

In future research work our approach could be modified by using different interval representations. Instead of AAFIs variances of intervals could constitute the basic for feature vectors.

References

[1] Alfamovie. Media LLC. www.alfamovie.com.
[2] K. Ayyildiz and S. Conrad. Video Classification by Main Frequencies of Repeating Movements. In *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011)*, 2011.
[3] R. Babu and K. Ramakrishnan. Compressed domain human motion recognition using motion history. In *ICIP03*, pages 321–324, 2003.
[4] F. Cheng, W. Christmas, and J. Kittler. Periodic human motion description for sports video databases. In *International Conference on Pattern Recognition*, 3:870–873, 2004.
[5] R. Cutler and L. S. Davis. Robust real-time periodic motion detection, analysis, and applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.
[6] Dailymotion. Dailymotion S.A. www.dailymotion.com.
[7] Q. He and C. Debrunner. Individual recognition from periodic activity using hidden markov models. In *Workshop on Human Motion*, pages 47–52, 2000.
[8] R. Lienhart. Indexing and retrieval of digital video sequences based on automatic text recognition. In *Fourth ACM international conference on multimedia*, pages 419–420, 1996.
[9] Q. Meng, B. Li, and H. Holstein. Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach. In *IVC*, pages 795–809, 2006.
[10] N. Patel and I. Sethi. Audio characterization for video indexing. In *SPIE on Storage and Retrieval for Still Image and Video Databases*, pages 373–384, 1996.
[11] S. Pei and F. Chen. Semantic scenes detection and classification in sports videos. In *Conference on Computer Vision, Graphics and Image Processing*, pages 210–217, 2003.
[12] R. Polana and A. Nelson. Detection and recognition of periodic, nonrigid motion. In *International Journal of Computer Vision*, 23:261–282, 1997.
[13] P. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic motion detection. In *Pattern Recognition*, pages 1591–1603, 1994.
[14] W. Wong, W. Siu, and K. Lam. Generation of moment invariants and their uses for character recognition. In *Pattern Recognition Letters*, 16:115–123, 1995.
[15] YouTube LLC. Youtube: Broadcast yourself. www.youtube.com.