World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:7, 2012

Analyzing Transformation of 1D-Functions for Frequency Domain based Video Classification

Kahraman Ayyildiz and Stefan Conrad

Abstract—In this paper we illuminate a frequency domain based classification method for video scenes. Videos from certain topical areas often contain activities with repeating movements. Sports videos, home improvement videos, or videos showing mechanical motion are some example areas. Assessing main and side frequencies of each repeating movement gives rise to the motion type. We obtain the frequency domain by transforming spatio-temporal motion trajectories. Further on we explain how to compute frequency features for video clips and how to use them for classifying. The focus of the experimental phase is on transforms utilized for our system. By comparing various transforms, experiments show the optimal transform for a motion frequency based approach.

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

I. Introduction

MOTION and video analysis for content based retrieval or action recognition purposes is an intensely investigated research area. Further topics connected to this research field are video surveillance, human-computer interfaces or object tracking. Research on these fields is encouraged by obvious demands. Most of today's digital video cameras for instance utilize face tracking methods in order to focus on faces and zoom into important parts of a photo. Moreover video surveillance is needed to protect company buildings, public places and private properties. Beyond that video databases can be found in major corporations or online video portals. So motion and video analysis has relevance for industry, technique and practical life.

In this work we present an approach, which uses frequency features from cyclic motion in video sequences for classification. Its basic idea stems from our previous research work [2]. The main difference between these two papers is the feature extraction stage. In [2] we used up to six frequency maxima as features for one video. Now we apply average amplitudes from the whole frequency spectrum as feature vector by partitioning the spectrum (see V). Our approach works for every motion type and is not limited to human gait recognition as described in [3], [18]. At first our method detects regions with motion framewise. These regions lead on to image moments for each frame, where a series of image moments represents a function. By transforming this function we obtain its frequency spectrum and are able to extract certain features for classification.

In the experimental stage of our research work we inspect the accuracy and runtime of different transforms in order to determine the best transform for our system. Results are representative for other approaches based on repeating movements, since this aspect is hardly researched in video analysis.

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).

II. Classifying Videos by AAFIs

In this section we offer an overview of the whole classification process. Fig. 1 illustrates the different steps of this process.

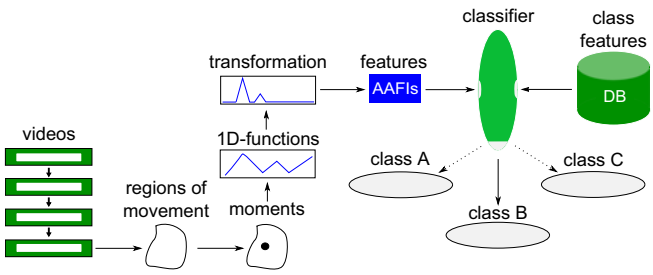

Fig. 1: Flow diagram of whole classification process

The flow diagram starts with the video data input. Valid video data contains repeating movements like hammering, planing, or filing for instance (home improvement). It is even possible to extend our approach to other topic areas: Playing tennis (sports) or accordion (music) are possible activities. Our approach is not capable for classifying activities with strongly differing frequency spectra or periodic texture motion.

Nevertheless the aim of the whole classification process is to classify video sequences with repeating movements properly. At first regions of movement are detected in every clip frame by frame. Regions are detected by measuring the color difference of pixels in two consecutive frames (see section III-A). With these regions image moments are calculated, where we use only centroids based on the *raw moment* type (see section III-B). A chronological series of these moments is considered as 1D-function and represents the motion in a video sequence. The transform of one 1D-function reveals its frequency domain. By partitioning the frequency axis into intervals of same length, average amplitudes for each interval are computed. We name these averages *AAFIs* (Average Amplitudes of Frequency Intervals). AAFIs constitute the final feature vectors for each clip with respect to its motion. After resolving the feature vectors a classifier can decide to which class a video fits best. Next sections explain each of the outlined stages in detail.

III. Image Moments and 1D-functions

In order to compute frequency spectra for video scenes the motion in each frame has to be localized. Once the motion is detected image moments and resulting 1D-functions can be figured. Next we define regions of motion and explain how these regions lead to 1D-functions.

A. Regions of Motion

Fig. 2 shows a person using a paint roller in two frames following each other. By analyzing these two frames we detect regions with motion. Color differences between the first and the second frame are measured for each pixel. If the color difference of a pixel exceeds a predefined threshold and if there are enough neighbor pixels with a color difference beyond the same threshold, this pixel is considered to be a part of a movement. Thus a region of motion is represented by the conflation of pixels with motion.

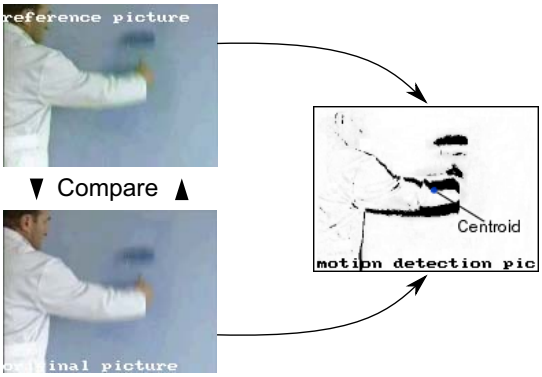

Fig. 2: Regions with pixel activity and centroid

A comparison of these two frames results in a binary image arising from regions with movement. Furthermore the centroid of regions with motion lies exactly on the right forearm, because the most active areas are the paint roller, the hand, the forearm and the upper arm. Hence painting the wall with a paint roller determines a specific centroid motion path as it evolves in time.

B. Image Moments

An image moment is the weighted average of pixel intensities of a picture. It can describe the area, the bias, or the centroid of segmented image parts. We distinguish two types of image moments: raw moments and central moments. Raw moments are sensitive to translation, whereas central moments are translation invariant. Next equation defines a raw moment M_{ij} for a two dimensional binary image $b(x, y)$ and $i, j \in \mathbb{N}$ [5]:

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

The order of M_{ij} is always $(i+j)$. M_{00} determines the area of segmented parts. Hence $(\bar{x}, \bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00})$ defines the centroid of segmented parts. Moreover the computation of central moments applies centroid coordinates [5].

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

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 chronological order of frames in a video, which leads to function $f(t)$ with t as time. For $(\bar{x}_t, \bar{y}_t) = (M_{10_t}/M_{00_t}, M_{01_t}/M_{00_t})$ as the centroid coordinates depending on time t function $f_c(t) = (\bar{x}_t, \bar{y}_t)$ can be decomposed as follows:

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

For the experimental stage in section VII we use $f_{c_x}(t)$ and $f_{c_y}(t)$ instead of $f_c(t)$, because the transformation of 1D-functions results in more decisive frequency spectra than transforming 2D-functions. For any 1D-function $f(t)$ the direction of a moment at time t is defined by equation 4.

$$f_d(t) = \begin{cases} +1, & \texttt{if } f(t) - f(t-1) > 0 \\ 0, & \texttt{if } f(t) - f(t-1) = 0 \\ -1, & \texttt{if } f(t) - f(t-1) < 0 \end{cases} \tag{4}$$

IV. Transformation of 1D-Functions

Transforms explained in this section are used during experimental stage. All introduced equations transform time discrete and periodic signals to their frequency domain.

A. Fast Fourier Transform

The most important transform in science is the Fourier transform. Since 1D-functions are finite and periodic, we utilize the discrete Fourier transform (DFT) [11]. Let $\hat{f} = (\hat{f}_0, ..., \hat{f}_{N-1}) \in C^{\mathbb{N}}$ as the discrete Fourier transform of a complex vector $x = (x_0, ..., x_{N-1}) \in C^N$, which is constituted by the interpolation of measured values. The transform's size is set to $2n$. Then formula 5 for $m = 0, \ldots, 2n-1$ results.

$$\hat{f}_m = \sum_{k=0}^{2n-1} x_k \, e^{-\frac{2\pi i}{2n} mk} \tag{5}$$

Here $\hat{f}_m$ is also called Fourier coefficient or Fourier component. Realizing the algorithm of Cooley and Tukey we get the fast Fourier transform (FFT) [4]. FFT works with DFTs of half length and consequently needs the quarter of complex calculations. This transform implements a divide-and-conquer method. Depending on the length of the measured value sequence the method can be applied multiple times, whereby the runtime amounts to $O(n \cdot log(n))$.

B. Fast Cosine Transform

The fast Cosine transform (FCT) is based up on the discrete Cosine transform (DCT) [1] and works similar to FFT. By contrast to DFT the DCT calculation takes place only with real coefficients. A finite data sequence is expressed as a finite sum of cosine functions. Since data values of discrete transforms are finite, a sequel before and after this sequence is presumed. The DCT computation is realized by an *even extension*.

Even Function A function f with a domain D is called an even function, if $\forall x \in D\ f(x) = f(-x)$.

Extending a sequence by even and odd functions allows four combinations (see IV-C). Furthermore it is possible to extend a sequence directly at a value or between two values. Thus $4 \cdot 4 = 16$ combinations are possible. The 8 marginal conditions with even sequels at the beginning belong to the cosine transforms. We use DCT-I with even marginals around x_0 at the beginning and around x_{N-1} at the end. Let $x_n = (x_0, \ldots, x_{N-1}) \Rightarrow X_n = (X_0, \ldots, X_{N-1})$ a real value mapping and $k = 0, \ldots, N-1$, then DCT-I is defined as:

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \quad (6)$$

By pre- and post-processing steps the DCT can be combined with FFT and solved with $O(N \cdot log(N))$ operations.

C. Fast Sine Transform

Fast Sine transform (FST) is based on discrete Sine transform (DST) [8], which is related to DCT. Now DST transforms a sequence of values by sine functions. Further on DST extends the beginning of a finite sequence by odd functions. Here again 8 different extensions are possible.

Odd Function A function f with domain D is called an odd function, if $\forall x \in D\ f(-x) = -f(x)$.

For our experiments we use DST-I among all 8 extensions. Its marginals around x_{-1} at the beginning and around x_N at the ending are odd. If a real value mapping $x_n = (x_0, \ldots, x_{N-1}) \Rightarrow X_n = (X_0, \ldots, X_{N-1})$ and $k = 0, \ldots, N-1$ is given, then one has

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N+1}(n+1)(k+1)\right] \quad (7)$$

The FST is computed analog to FCT by combining the incoming signal with equation (7) and FFT. Here again runtime amounts to $O(N \cdot log(N))$.

D. Fast Haar-Wavelet Transform

Now we define the Haar-Wavelet [12]:

$$\psi(x) = \begin{cases} 1 & \texttt{if } 0 \le x < 1/2 \\ -1 & \texttt{if } 1/2 \le x < 1 \\ 0 & \texttt{else} \end{cases} \quad (8)$$

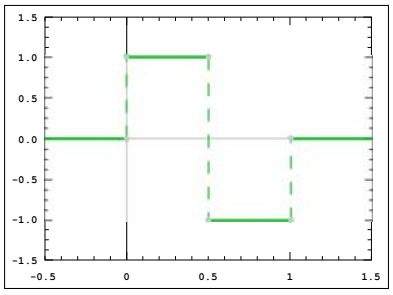

Fig. 3: Haar-Wavelet

Fig. 3 illustrates the Haar-Wavelet plot. For $j \in \mathbb{N}, 0 \le k \le 2^j - 1$ it is feasible to swage and to shift the Haar wavelet inside interval [0,1]:

$$\psi_{j,k}(x) = \psi(2^j \cdot x - k) \quad (9)$$

Functions $\psi(x)$ and $\psi_{j,k}(x)$ are orthogonal inside interval [0,1]. These assumptions lead to Haar matrices of different orders [12]. In (10) for instance a second order Haar transform matrix H is presented.

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (10)$$

Considering fast Haar wavelet transform (FWT) a finite discrete signal is transformed into a sequence of two dimensional vectors. This process is the so-called *polyphase partition*. Hereby data sequence is joined pairwise without changing the order of its elements.

$$f_p = \left(\ldots, \begin{pmatrix} f_{-2} \\ f_{-1} \end{pmatrix}, \begin{pmatrix} f_0 \\ f_1 \end{pmatrix}, \begin{pmatrix} f_2 \\ f_3 \end{pmatrix}, \ldots\right) \quad (11)$$

Each element of this new sequence is multiplied with transform matrix H.

$$\begin{pmatrix} s \\ d \end{pmatrix} := H \cdot f_p = \left(\ldots, \begin{pmatrix} s_{-1} \\ d_{-1} \end{pmatrix}, \begin{pmatrix} s_0 \\ d_0 \end{pmatrix}, \begin{pmatrix} s_1 \\ d_1 \end{pmatrix}, \ldots\right) \quad (12)$$

Then one has $s_k = \frac{f_{2k} + f_{2k+1}}{\sqrt{2}}$ and $d_k = \frac{f_{2k} - f_{2k+1}}{\sqrt{2}}$. Moreover the input signal can be partitioned into bigger blocks, if a corresponding orthogonal Haar matrix with entries $1/\sqrt{s}$ in its first line is generated. The FWT conforms to complexity class $O(N \cdot log(N))$.

E. Fast Hadamard Transform

The Hadamard transform is an orthogonal, symmetric, involutional und linear function, which is realized by 2^m input values [13]. Its calculation takes place with a $2^m \times 2^m$ Hadamard matrix H_m. This matrix transforms a 2^m sized real sequence x_n into a 2^m sized sequence X_k by multiplying H_m with x_n. For $H_0 = 1$ and $m > 0$ the Hadamard matrix is defined recursively as follows:

$$H_m = \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix} \quad (13)$$

H_0 yields that all entries of H_m must be 1 or -1. Let k and n indices with k_j and n_j as their binary digits:

$$k = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \cdots + k_1 2 + k_0$$
$$n = n_{m-1}2^{m-1} + n_{m-2}2^{m-2} + \cdots + n_1 2 + n_0$$

Hence each entry (k, n) of a Hadamard matrix is computed by equation (14).

$$(H_m)_{k,n} = \frac{1}{2^{m/2}}(-1)^{\sum_j k_j n_j} \tag{14}$$

The Hadamard transform involves $O(N^2)$ operations. Fast Hadamard transform (FHT) requires $O(m \cdot log(m))$ calculation steps [14]. It also uses a divide and conquer algorithm in order to break down Hadamard transforms of size N into two smaller Hadamard transforms of size $N/2$ recursively.

V. AAFIs as Feature Vectors

In our previous work [2] we used up to 6 frequency maxima for each video as feature vector. Now the whole frequency spectrum is described by AAFIs and feature vectors reveal much more information about the motion type. Each significant frequency high or low has an influence on concerning AAFI.

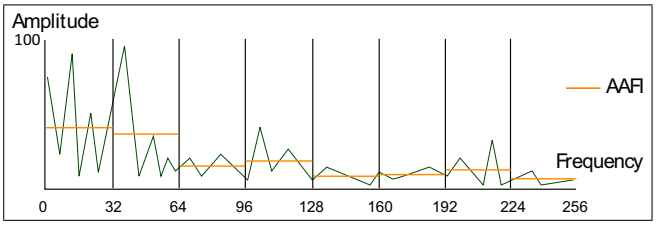

Fig. 4: Average amplitudes of frequency intervals (AAFIs)

As already mentioned each 1D-function can be transformed to its frequency spectrum. Partitioning this spectrum into intervals of same length, an average amplitude for each interval can be stated.

Fig. 4 depicts this idea by partitioning a frequency spectrum with a length of $m = 256$ units to $n = 8$ intervals. Using the fast Fourier transform for instance 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. 4 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.

VI. Radius Based Classifier

Now we explain our *Radius Based Classifier* RBC [2]. This classifier turned out as very effective for distance computations between objects and object classes with frequency domain based features. The RBC uses the radius ε around a given object o and counts all objects of a class C_i around o with a distance smaller than ε. The normalized sum of all objects leads to a distance $dist(o, C_i) = 1 - \frac{|N_\varepsilon(o,C_i)|}{|C_i|}$ between tested object and class. After computing distances to all existing classes, the RBC assigns o to the class with the smallest distance.

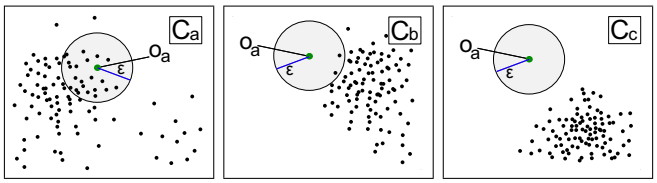

Fig. 5: Classifying with RBC

Fig. 5 illustrates how the RBC works: An object o_a of an unspecified class has to be classified. Therefore it is assigned to each existing class in order to calculate the class with the minimal distance. Three different example classes C_a, C_b and C_c are given and 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 the typical object distribution. At the same time this fact leads to a minimal distance.

VII. Experiments

In this section we evaluate accuracy and runtime performance of our system concerning different transforms. Test series are performed by own and by external video data. Own videos are recorded especially for the evaluation phase and external video data is taken from the online video database *youtube.com* [20]. In addition experiments with own video data are computed by m-fold cross validation. For classification process we use 10 classes, where each class consists of 20 videos (total 200 videos). External videos are analyzed by assigning them to especially recorded video classes, because cross validation was not possible due to classes with just few clips (total 102 videos). Each video shows one of the next 10 home improvement activities: filing, hammering, planing, sawing, screwing, using a paint roller, a paste brush, a putty knife, sandpaper and a wrench.

A. Motion Transformation

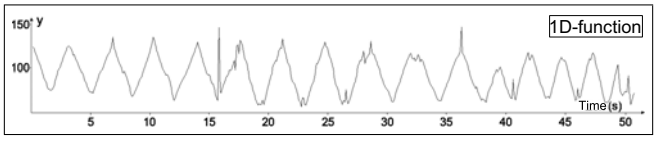

Fig. 6: 1D-function of a person using a paint roller

Next in fig. 6 an example 1D-function is illustrated. It regards to the y-axis coordinate of centroids and captures the mean motion inside an external video clip. The motion in this clip arises from a person using a paint roller. Furthermore the

1D-function corresponds to the movement of the person, since the centroid moves from bottom to top and vice versa. Plots in fig. 7 present frequency spectra of 1D-function from fig. 6. Each spectrum results from one of five transforms explained in section IV.

At first glance it becomes apparent, that all transforms except FHT have a high amplitude around a frequency of 12. This frequency corresponds to the main movement of the paint roller in fig. 6. FCT presents frequencies more detailed and partially with higher amplitudes, but roughly frequencies are identical with FFT. Now FST shifts FFT and FCT highs to the left, whereas FWT has high amplitudes at different frequencies. There is even a peak at a frequency of 165.

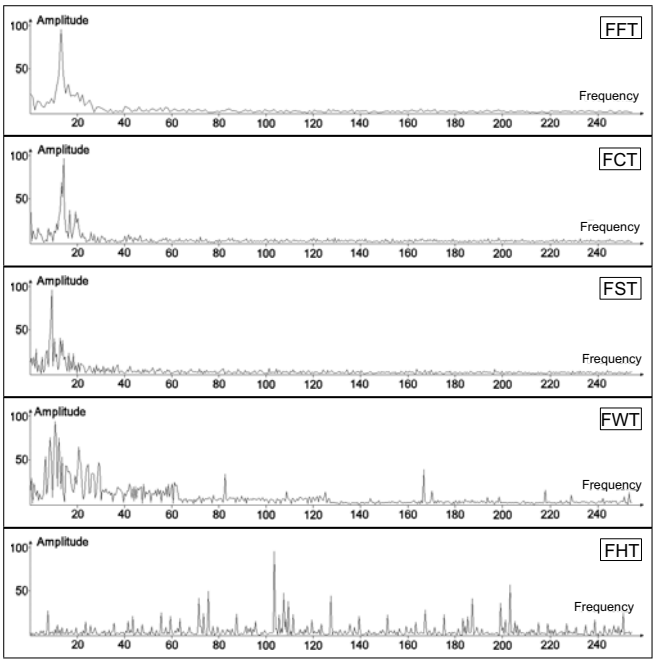

Fig. 7: Transforms of paint roller 1D-function

B. Accuracy of transforms

Next tables I and II show accuracies for own and external videos, respectively. *Overall accuracies* are not the average of *class accuracies*, but the ratio of correct classifications to the total number of classifications. Now own videos are assigned to classes by direction information, whereas external videos are assigned using location information of image moments (see section III-C). External video data contains more irregular motion, so for this case direction information is less reliable. Tables show overall and single class accuracies. In table I the transform with the maximum accuracy 0.87 is FFT. The transforms FCT, FST and FWT also achieve high accuracies. As shown in fig. 7 here again FHT differs clearly from the other transforms. Its accuracy marks the minimum with 0.29. Considering the class accuracies for each transform we find high accuracies for home improvement activities with clear motion and an unique frequency feature.

For external videos FCT leads to a maximum accuracy of 0.41. Here again FHT has the lowest accuracy with 0.14.

TABLE I: Test results for own videos

Class ACC	FFT	FCT	FST	FWT	FHT
File	0.85	0.85	0.70	0.35	0.25
Hammer	0.84	0.83	0.79	1.00	0.25
Paint Roller	1.00	1.00	0.80	0.55	0.45
Paste Brush	0.90	0.85	0.85	1.00	0.40
Plane	0.55	0.75	0.65	0.70	0.11
Putty Knife	0.95	0.95	1.00	0.80	0.40
Sandpaper	0.88	0.56	0.53	0.75	0.05
Saw	0.75	0.65	0.80	0.90	0.20
Screwdriver	0.95	1.00	1.00	0.70	0.75
Wrench	1.00	1.00	0.85	0.95	0.05
Overall ACC	**0.87**	**0.85**	**0.80**	**0.77**	**0.29**

TABLE II: Test results for external videos

Class ACC	FFT	FCT	FST	FWT	FHT
File	0.00	0.00	0.00	0.75	0.00
Hammer	0.00	0.00	0.22	0.00	0.11
Paint Roller	0.69	0.92	0.50	0.79	0.15
Paste Brush	0.00	0.25	0.00	0.25	0.25
Plane	0.00	0.00	0.00	0.00	0.00
Putty Knife	0.57	0.55	0.52	0.43	0.04
Sandpaper	0.36	0.18	0.18	0.45	0.18
Saw	0.33	0.22	0.22	0.00	0.10
Screwdriver	0.57	0.57	0.14	0.00	0.43
Wrench	0.50	0.63	0.13	0.38	0.38
Overall ACC	**0.40**	**0.41**	**0.27**	**0.32**	**0.14**

If we compare results for own and external video scenes as shown in fig. 8, it becomes apparent that own videos are assigned twice as accurate as external videos. Moreover it can be stated, that FFT and FCT work best for our purpose. FST and FWT result in moderate accuracies and FHT is the weakest approach.

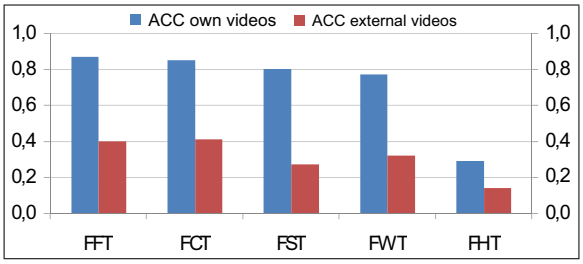

Fig. 8: Overall accuracies for own and external clips

Fast Fourier transform and fast Cosine transform plot similar frequency spectra. Fast Sine transform is also similar to fast Fourier transform and fast Cosine transform, but there are accuracy decreasing differences. Due to the fact that fast Sine transform uses odd extensions at the beginning and at the end of a data sequence, we detect more high amplitudes for low frequencies. Furthermore comparing fast Sine transform to fast Fourier transform amplitudes are shifted to the low frequency range a bit. On the whole the low frequency range is not as reliable as the low frequency range of fast Fourier transform or fast Cosine transform. Fast Wavelet transform works not as accurate as fast Fourier transform, because wavelets are

localized in both time and frequency whereas the standard Fourier transform is only localized in frequency. This means fast Wavelet transform is capable to represent discontinuities and sharp peaks of a 1D-function. Our repeating motion based approach is impaired by representing stationary movements inside the frequency spectrum. Fast Hadamard transform is extremely sensitive to time shift and not invariant under circular time shift of the input data [17]. Even a minimal difference of the starting point of the dataset changes the frequency domain drastically. Hence fast Hadamard transform is not useful for repeating motion based video classification.

C. Runtime Analysis

This subsection concerns runtime of our approach with regard to different transforms. The bar chart in fig. 9 shows runtimes for fast Fourier-, fast Cosine-, fast Sine-, fast Hadamard- and fast Wavelet transform.

When classifying 100 videos with FHT the system runs 114 seconds. This means with FHT the system works faster than with any other transform. As we have explained in section IV FHT belongs to complexity class $O(m \cdot log(m))$, where $2m$ is the size of data elements. Other transforms belong to the same complexity class, but the number of data elements is m. FFT follows with 117 seconds at position two. Although FFT, FCT, FST and FWT belong to the same complexity class, FFT works fastest. Since FCT and FST modify a sequence resulting from FFT their runtime is the sum of FFT's runtime and additional operations. Thus FCT and FST run up to 120 seconds. With 123 seconds FWT is located at the last position. Here the polyphase partition and following transform-matrix multiplication with data elements are more expensive than the divide and conquer approach of Cooley and Tukey.

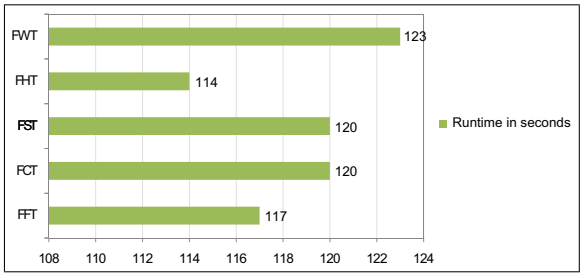

Fig. 9: System's runtime with different transforms

VIII. Related Work

Videos reveal a huge amount of information. Hence video annotation and classification is realized in various manners. Main techniques base on key-frames [6], texts in frames [7], audio signals [9] and motions.

Cheng et al. analyze sports videos in [10]. Series of horizontal and vertical pixel motion vectors are transformed by a modified FFT and result in two main frequencies for each clip. Here authors state accuracies up to 1.0 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 [16]. They obtain frequency domain by transforming measured self-similarity of motion via Fourier transform. Experimental results depict an accuracy of 1.0 for each of three tested classes. Moreover in [3] and [18] repeating motion of human body parts is analyzed by tracking Moving Light Displays (MLD). In [3] the frequency peaks of transformed MLD curves are considered as features of cyclic motion. Here again transformation takes places via Fourier coefficients. 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. In paper [18] optimal MLD curve patterns are determined by FFT. We tested this idea, but results have shown that repeating motion patterns are a weak solution, since patterns of the same motion type can vary drastically.

Direct comparison of transforms is provided by image processing research: Authors of [19] compare discrete cosine, Walsh Hadamard, Haar-Wavelet and Kekre's transform for an image retrieval system. Here images are transformed before the feature extraction stage. Discrete cosine, Walsh-Hadamard and Haar-Wavelet transforms have a maximum accuracy at 0.41 and Kekre's transform at 0.42. In [15] HWT, WHT, DTT and DCT are compared and analyzes show that HWT works best for image compression using orthogonal basis functions. WHT is positioned at the fourth place with a marginal difference to the other transforms.

Summarizing it can be stated, that FFT and DCT are the most commonly used transforms in image and video retrieval context. For our frequency domain based classification method we could not identify related research in computer vision, which directly compares each of the above transforms.

IX. Conclusion

In this paper we have shown a video classification method, which is based on the frequency of repeating movements. Frequency spectra are computed by transforming spatio-temporal image moment trajectories (1D-functions). We explained how frequency spectra can be utilized for feature extraction and video classification. Thereto we introduced AAFIs, which are a strong approach for classifying videos.

Here the experimental stage discussed the transform process of 1D-functions extensively in order to determine the best transform for our approach. Results expose that FFT and FCT lead on to best accuracies, where FFT has a marginal runtime advantage. FHT is the fastest transform, but at the same time it brings about the lowest accuracy values.

Beyond that it remains an open issue to adapt and analyze the presented approach for real time action recognition. In future work we intend to extend the presented approach, so as to recognize different activities inside a single video.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, 90–93, 1974.
[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*, April 13-15, 2011.
[3] Q.G. Meng, B.H. Li, and H. Holstein, "Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach," *IVC*, 795–809, 2006.
[4] James W. Cooley, and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp. 19*, 297–301, 1965.
[5] W. Wong, W. Siu, and K. Lam, "Generation of moment invariants and their uses for character recognition," *Patt. Recog. Lett.*, 115–123, 1995.
[6] S.C. Pei and F. Chen, "Semantic scenes detection and classification in sports videos," in *Conf. on Comp. Vision, Graphics and Image Proc.*, 2003, 210–217.
[7] R. Lienhart, "Indexing and retrieval of digital video sequences based on automatic text recognition," in *Fourth ACM int. Conf. on Multimedia*, 419–420, 1996.
[8] S. A. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Trans. Sig. Processing SP-42*, 1038–1051, 1994.
[9] N. Patel and I. Sethi, "Audio characterization for video indexing," in *SPIE on Storage and Retrieval for Still Image and Video Databases*, 373–384, 1996.
[10] Fangxiang Cheng, William Christmas, and Josef Kittler, "Periodic human motion description for sports video databases," *International Conference on Pattern Recognition*, vol. 3, 870–873, 2004.
[11] Matteo Frigo, "A fast Fourier transform compiler," *SIGPLAN Notices*, 642–655, 2004.
[12] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms," *Comm. Pure Appl. Math.*, 141–183, 1991.
[13] Hakob Sarukhanyan, Sos Agaian, Karen Egiazarian, and Jaakko Astola, "Reversible Hadamard Transforms," *ELEC. ENERG.*, 309–330, 2007.
[14] B.J. Fino, and V.R. Algazi, "Unified Matrix Treatment of the Fast Walsh Hadamard Transform," *IEEE Trans. on Comp.*, 1142-1146, 1976.
[15] O. Hunt and R. Mukundan, "A comparison of discrete orthogonal basis functions for image compression," *Image and Vision Computing New Zealand*, 234-245, 2004.
[16] Ross Cutler and Larry S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 781–796, 2000.
[17] Hozumi Tanaka, "Hadamard transform for speech wave analysis," in *Technical Report at Stanford University*, 1972.
[18] P. Tsai, M. Shah, K. Keiter, and T. Kasparis, "Cyclic motion detection," in *Pattern Recognition*, 1591–1603, 1994.
[19] H. B. Kekre, S. D. Thepade, and A. Maloo, "Performance comparison of image retrieval using fractional coefficients of transformed image using dct, walsh, haar and kekres transform," *Int. Journal of Image Proc. (IJIP)*, 142–155, 2010.
[20] YouTube LLC. Youtube: Broadcast yourself. www.youtube.com.