Applying Filters to Repeating Motion based Trajectories for Video Classification

Kahraman Ayyildiz, Stefan Conrad
Department of Computer Sciences
Heinrich Heine University Duesseldorf
Universitätsstraße 1, 40225 Duesseldorf, Germany
kahraman.ayyildiz, stefan.conrad@uni-duesseldorf.de

ABSTRACT

The presented video classification system is based on the trajectory of repeating motion in video scenes. Further on this trajectory has a certain direction and velocity at each time frame. As the position, direction and velocity of the motion trajectory evolve in time, we consider these as motion functions. Later on we transform these functions by FFT and receive frequency domains, which then represent the frequencies of repeating motion. Moreover these frequencies serve as features during classification phase. Our current work focuses on filtering the functions based on the motion's trajectory in order to reduce noise and emphasize significant parts.

Keywords
Action Recognition, Video Classification, Repeating Motion, Frequency Feature, Filter, Occlusion

1 INTRODUCTION

Today there is a strong demand for computer vision research, since recognition and tracking of objects or motions are core subjects for some major industries. Face tracking for videoconferencing, computer controlling by gestures, size measurement of components on band conveyors or positioning of industrial robots are only some examples, where computer vision has already been established successfully. Moreover computer vision is also needed when it comes to video annotation and classification for video databases.

Current research work brings action recognition and classification by repeating motion into focus. In [AC2012] we already presented the basic idea of our approach. Now we extend our system by adding different filters in order to smoothen or to emphasize repeating motion in videos. Hence the experimental phase is concerned with accuracy and runtime analysis for different filters. Especially when recording conditions for videos differ, filters can compensate these differences. This pertains for varying illumination, resolution, occlusion, shaking or angle.

The analyzed filters in the experimental part of this research work are applied to repeating motion based trajectories. These trajectories serve as the basis for feature extraction. In the field of motion analysis filtering is sparsely researched. Thus our contribution at hand points out the effect of filters on motion trajectories and resulting features.

2 RELATED WORK

Videos can contain key-frames, texts, audio signals, motions or meta-data. Hence video classification can be realized in various ways. In our research work we focus on repeating motion, which is also discussed in a similar way by [MLH2006] and [CCK2004]. [MLH2006] deals with repeating motion of human body parts tracked by Moving Light Displays (MLD). Frequency peaks of Fourier transformed MLD curves are considered as features of repeating motion. In [CCK2004] Cheng et al. analyze sports videos by using a neural network based classifier. They receive two main frequencies for each video by transforming series of vertical and horizontal pixel motion vectors. The transformation takes place by a modified fast Fourier transform. Furthermore the authors of [FZP2005] propose a hybrid model for human action recognition, which is robust against occlusion. This model is based on position, velocity and appearance of body parts.

The filters we consider in this work are particularly applied in image processing and hardly in video analysis. Research in [VUE2010] and [MAS1985] shows that the Lee filter performs better than the average or median filter when it comes to noise reduction for images. Alsultanny and Shilbayeh analyze a series of filters by applying them to satellite images [AS2001]. Here median, average and low-pass filters lead to similar results. Concerning edge detection filters

the so-called *Prewitt filter* works more accurate than Laplace filter.
In the field of video content and motion trajectory analysis there is sparse research done on the application of filters.

3 APPLICATION OVERVIEW

The flow diagram in figure 1 illustrates the different phases of our system [AC2012]. It starts with video data input containing repeating motions as painting, hammering or planing for instance (home improvement). Next regions with motion are detected for each clip frame by frame. For region detection the color difference of pixels in two sequential frames is measured. On the basis of motion regions we calculate image moments. We consider the chronological order of image moments as a *1D-function*, which again represents the main motion in a video sequence. This 1D-function is filtered in order to remove noise respectively to weight important parts. Moreover the result is transformed and we receive a frequency domain describing the frequencies of repeating motion in the video. By dividing the frequency axis into intervals of same length, average amplitudes for each interval are calculated. We name these averages *Average Amplitudes of Frequency Intervals* and refer to them as *AAFIs*. AAFIs set up the final feature vector for each video. At last a radius based classifier (RBC) utilizes this feature vector for the purpose of computing the nearest class for a video.

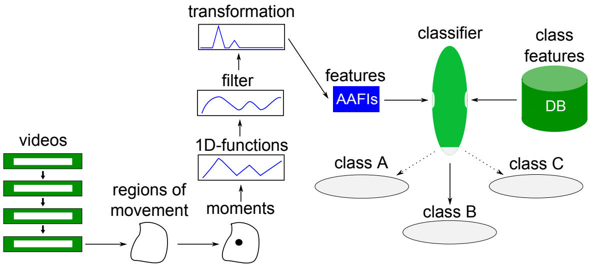

Figure 1: Flow diagram of the whole classification process

4 IMAGE MOMENTS AND 1D-FUNCTIONS

Once motion areas in a video scene are detected image moments can be determined. These image moments lead to 1D-functions, which are explained and defined formally in this section.

Regions of Motion

Figure 2 shows a person painting a wall. We detect regions with movement by comparing two sequential frames of this activity. Further on we measure color differences between these two frames for each pixel. The color difference of a pixel exceeding a predefined threshold combined with a minimum number of neighbor pixels with a color difference beyond the same threshold defines a pixel to be part of a movement. Thus a region with motion is represented by the entirety of pixels with motion. Pixel differences of the two frames shown in figure 2 point out regions with movement, which again are visualized by a monochrome image on the right. It is obvious that the most active areas are the paint roller, the hand, the forearm and the upper arm. Therefore the centroid of regions with motion follows exactly the right forearm. As a result the painting activity sets a specific motion trajectory.

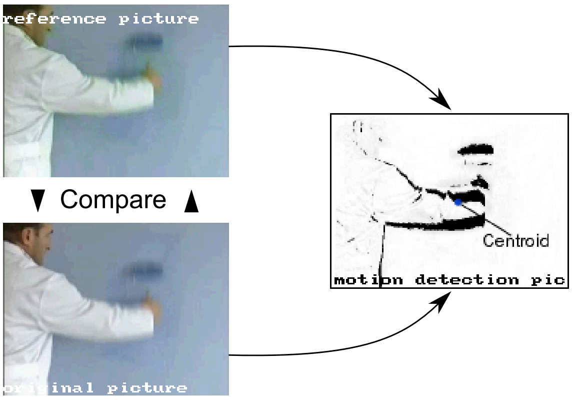

Figure 2: Regions with pixel activity and centroid

Image Moments

An image moment is defined as an image's weighted average of pixel intensities. It can describe the bias, the area or the centroid of segmented image areas. The two main image moment types are raw moments and central moments. Raw moments are sensitive to translation, whereas central moments are translation invariant. The next equation defines a raw moment M_{ij} for a two dimensional monochrome image $b(x,y)$ with $i,j \in \mathbb{N}$ [WSL1995]:

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

The order of M_{ij} is always $(i+j)$. M_{00} is the area of segmented parts. Consequently $(\bar{x},\bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00})$ determines the centroid of segmented parts.

Deriving 1D-functions

Video frames have a chronological order. Hence a series of moment values is also depending on time t. Now we define a 1D-function $f(t)$ as a series of these moment values by considering only one dimension. For centroid coordinates $(\bar{x}_t,\bar{y}_t) = (M_{10_t}/M_{00_t}, M_{01_t}/M_{00_t})$ we decompose function $f_c(t) = (\bar{x}_t,\bar{y}_t)$:

$$f_{c_x}(t) = \bar{x}_t, \; f_{c_y}(t) = \bar{y}_t \qquad (2)$$

Experiments in section 6 use only $f_{c_x}(t)$ and $f_{c_y}(t)$ instead of $f_c(t)$, because the 1D-function transforms result in more accurate frequency domains than 2D-function transforms. By equation (3) we define the direction of an image moment at time t for any 1D-function $f(t)$.

$$f_d(t) = \begin{cases} +1, & \texttt{if } f(t) - f(t-1) > 0 \\ 0, & \texttt{if } f(t) - f(t-1) = 0 \\ -1, & \texttt{if } f(t) - f(t-1) < 0 \end{cases} \tag{3}$$

Now the speed of an image moment at time t is defined as follows:

$$f_s(t) = |f(t) - f(t-1)| \tag{4}$$

5 FILTERS FOR 1D-FUNCTIONS

In real world videos motions of the same activity are never exactly the same and motion trajectories differ from ideal mathematical functions. Unexpected motions, occluded motion or low recording quality can reduce the clarity of 1D-functions and therefore the system's accuracy. In order to improve the clarity various filters can be applied. Filters can reduce noise, smoothen trajectories or emphasize edges, which mean the change of direction in the case of 1D-functions.

Maximum Filter

A maximum filter substitutes each value of a data sequence by a maximum value inside a predefined radius. Let sequence (a_i) with $a_i \in \mathbb{N}$, $i = 0, \ldots, n$ and let radius $r \in \mathbb{N}$. Further on we define $N_r(i)$ as the set of neighborhood indices of sequence element a_i:

$$N_r(i) = \{x \mid 0 \leq x \leq n \land i - r \leq x \leq i + r\} \tag{5}$$

By these definitions we can compute the maximum value around a_i:

$$max_r(a_i) = \max\nolimits_{x \in N_r(i)} a_x \tag{6}$$

Now applying the maximum filter the new sequence $(q_{i_{max}})$ gives:

$$(q_{i_{max}}) = (max_r(a_0), max_r(a_1), \ldots, max_r(a_n)) \tag{7}$$

Median Filter

The median filter substitutes each value of a sequence by a medium value inside a given radius. Again we consider sequence (a_i) with $a_i \in \mathbb{N}$ and $i = 0, \ldots, n$, radius $r \in \mathbb{N}$ and $N_r(i)$. For each value a_i we compute a sorted subsequence $(s_j) = (s_1, s_2, \ldots, s_m)$ inside radius r, where again $N_r(i)$ determines the indices neighborhood. For m as the length of (s_j) we define:

$$med_r(a_i) = \begin{cases} \frac{1}{2}\left(s_{\frac{m}{2}} + s_{\frac{m}{2}+1}\right), & \texttt{if m even} \\ s_{\frac{m+1}{2}}, & \texttt{if m odd} \end{cases} \tag{8}$$

For (a_i) the usage of a median filter results in $(q_{i_{med}})$:

$$(q_{i_{med}}) = (med_r(a_0), med_r(a_1), \ldots, med_r(a_n)) \tag{9}$$

Average Filter

By applying the average filter each value of a sequence is replaced by the average of all values inside radius $r \in \mathbb{N}$. For sequence (a_i) and $N_r(i)$ as the indices neighborhood we replace each value a_i as follows:

$$avg_r(a_i) = \frac{\sum_{x \in N_r(i)} a_x}{|N_r(i)|} \tag{10}$$

Hence we formulate sequence $(q_{i_{avg}})$ as:

$$(q_{i_{avg}}) = (avg_r(a_0), avg_r(a_1), \ldots, avg_r(a_n)) \tag{11}$$

Lee Filter

J. S. Lee proposes a statistical filter for digital images [LEE1980]. Lee assumes that each image contains natural noise, which can be removed pixelwise. Let σ^2 the variance inside radius r, δ a predefined noise energy and $\sigma^2 < \delta$, then a pixel is replaced by the average inside r. For $\sigma^2 > \delta$ the original value is replaced by another functional value: A high variance σ^2 means that the original value stays almost the same, because it is significant. Lee's filter can also be applied to 1D-functions. For sequence (a_i), radius r and $\beta = max(\frac{\sigma^2 - \delta}{\sigma^2}, 0)$ with $\beta \in \mathbb{R}^+$ we define the Lee filter as:

$$lee_r(a_i) = \beta \cdot a_i + (1 - \beta) \cdot avg_r(a_i) \tag{12}$$

So for the new, filtered sequence $(q_{i_{lee}})$ we receive:

$$(q_{i_{lee}}) = (lee_r(a_0), lee_r(a_1), \ldots, lee_r(a_n)) \tag{13}$$

Laplace Filter

A Laplace filter is usually utilized for signal and image processing in order to emphasize edges [VYB1989]. It is based on the *Laplace operator*, which simply means the second derivative in the context of 1D-functions.

Hence 0 as the second derivate points to a local minimum or maximum. This again gives a hint for an edge inside a signal or an image. So the discretization of the second partial derivative results in:

$$\begin{aligned}\Delta f(i) &= \frac{\partial^2 f(i)}{\partial i^2} \\ &\approx \frac{\partial (f(i+1) - f(i))}{\partial i} \\ &\approx f(i+1) - f(i) - (f(i) - f(i-1)) \\ &= f(i+1) - 2 \cdot f(i) + f(i-1)\end{aligned} \tag{14}$$

Consequently the Laplace operator can be described as a convolution matrix.

$$D_i^2 = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \tag{15}$$

An extension of equation (14) allows determining edges with varying properties.

$$\Delta f_{r,t}(i) = (f(i+r) - 2 \cdot f(i) + f(i-r))^t \tag{16}$$

Variable $r \in \mathbb{N}$ extends or reduces the radius for the local minimum and maximum search. Parameter $t \in \mathbb{N}$ has a further influence on the filtering process. For instance $t = 2$ leads to only positive results.

Let (a_i) with $a_i \in \mathbb{N}$, $i = 0, \ldots, n$ and $f(i) = a_i$, where $\Delta f_{r,t}(i)$ is undefined for $(i - r) < 0$ or $(i + r) > n$. Now by these preconditions a Laplace filtered sequence $(q_{i_{lpc}})$ based on equation (16) can be determined:

$$(q_{i_{lpc}}) = (\Delta f_{r,t}(0), \Delta f_{r,t}(1), \ldots, \Delta f_{r,t}(n)) \tag{17}$$

6 EXPERIMENTS

This section focuses on accuracy and runtime performance of our system with respect to the filters introduced in section 5.

Motion Filtering and Transformation

Figure 4 shows filtered example 1D-functions on the left and corresponding transforms on the right side. Moreover the basic 1D-function stems from a person's motion while using a wrench. Particularly the charts in figure 4 plot x-axis coordinates of centroids and capture the main motion. It is obvious that the 1D-functions correspond to the left-right and right-left movements. Transforming these 1D-functions by fast Fourier transform (FFT) results in a frequency domain with peaks at 13 and 27. The first amplitude peak at 13 corresponds to the number of left-right movements. In addition the second peak at 27 arises from a slight centroid movement along the x-axis between two repetitions. This typical centroid movement results from the overall body motion.

Without a filter the spatio-temporal motion trajectory has many highs and lows inside a small time frame. If we consider maximum, median or average filter, these highs and lows disappear and the original chart appears smoothed. In addition maximum and medium filter lead on to edged charts. For each filter the corresponding high frequency domain has lower amplitudes than the original high frequency domain without filter usage. Especially the average filter reduces amplitudes of the high frequency ranges. However Lee filter smoothens only parts of the 1D-function, which are below a predefined noise level. Other parts with strong movements even inside small time frames stay nearly unmodified. So only high frequency amplitudes belonging to noisy parts are reduced.

The last chart in figure 4 shows the transform for Laplace filter. Frequency 27 is emphasized strongly, because corresponding edges in the 1D-function are emphasized. By using a small or large radius it is even possible to focus on high frequency or low frequency domains, respectively.

Motion Occlusion

Figure 3 illustrates how occlusion changes motion detection pictures for video scenes. A planing video, with the main motion taking place along the horizontal axis, is occluded by a vertical bar. The occluded motion area is not visible inside the motion detection picture and therefore its image moment and depending 1D-functions change. We adjust the alignment and the width of the bar manually for each class in order to achieve a maximal distraction of the motion centroid. This means the bar has always a relative thickness to the main motion area as shown in figure 3 and furthermore that this bar is always in the middle of the motion.

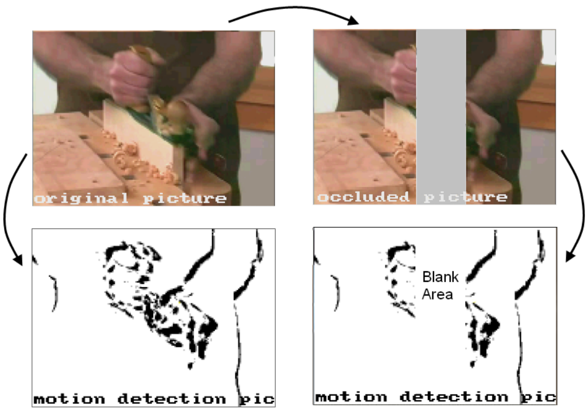

Figure 3: Regions with movement for an occluded planing video

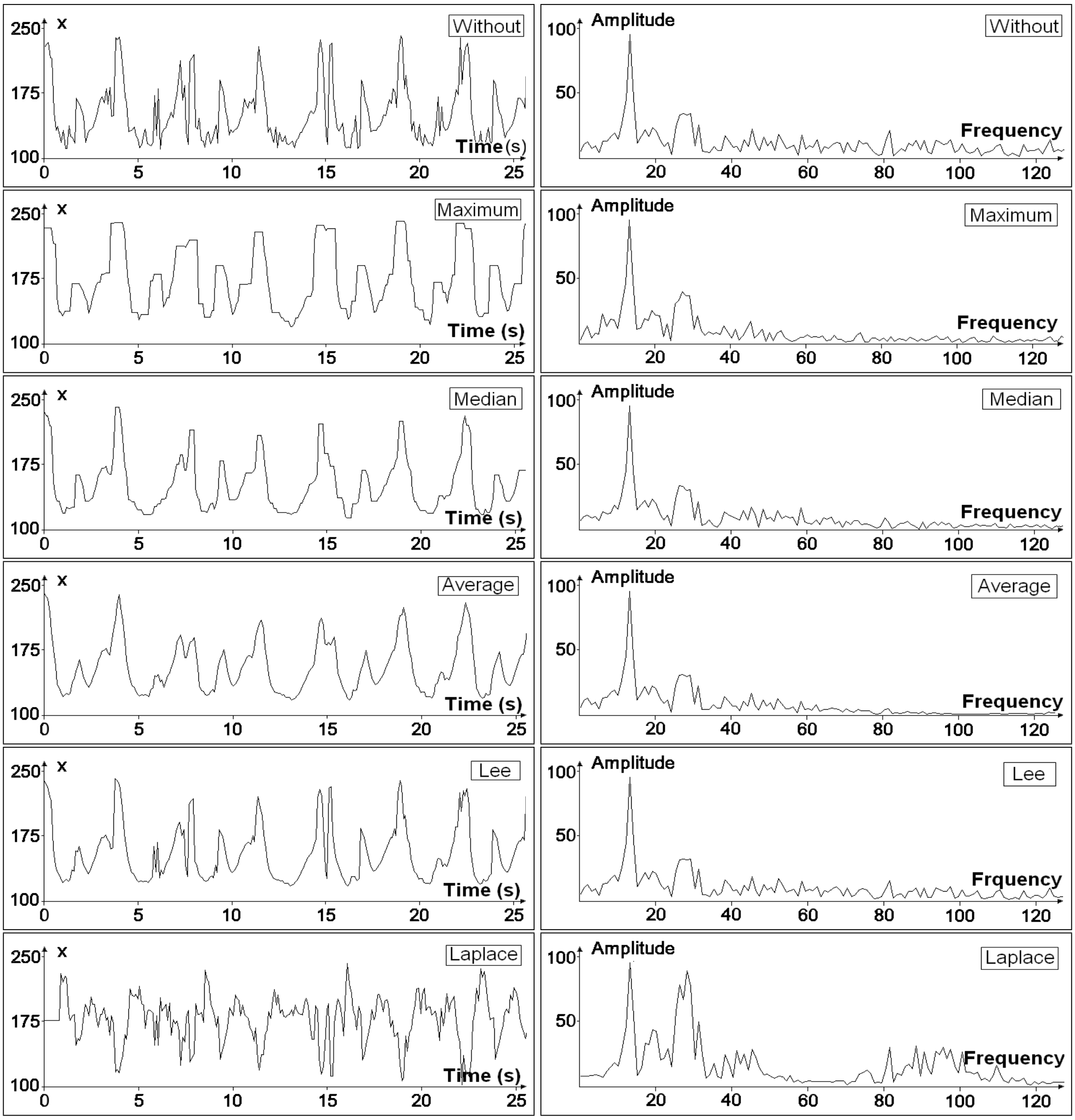

Figure 4: Filtered wrench handling 1D-functions with corresponding transforms

Filter Accuracies

In total we assign 200 own and 102 external videos [YT2010] to one out of ten home improvement classes. These classes contain following activities: filing, hammering, planing, sawing and using a paint roller, paste brush, putty knife, sandpaper, screwdriver, wrench. For our own video data we use twenty-fold cross validation, whereas the external videos are assigned directly to the video classes, because cross validation was not possible due to classes with just too few video clips.

Table 1 shows resulting accuracies for different 1D-functions and filters. Here accuracy means the correct classification ratio. Additionally we check the same video classes with occlusion. Our purpose is to find out, how occlusion affects the classification process and how far filter can balance out irregularities caused by occlusion.

At first glance it becomes apparent that own videos achieve much higher accuracies than external videos. The reason for this behavior is that all own videos have similar recording conditions, whereas all external videos have different recording conditions. Therefore extracted features for external videos vary more than for own videos.

The experimental results in table 1 depict, that occlusion decreases accuracies. But the system is still able to classify own videos via centroid location and

Filter	None	Maximum	Median	Average	Lee	Laplace
Own Videos						
Direction	0.89	0.87	**0.86**	**0.92**	0.89	**0.86**
Location	0.81	**0.72**	0.73	0.73	0.81	**0.84**
Speed	0.48	0.45	**0.37**	**0.49**	0.47	0.44
Own Videos with Occlusion						
Direction	0.70	0.71	0.73	0.75	**0.81**	0.71
Location	0.72	**0.67**	0.69	**0.67**	**0.73**	0.68
Speed	0.35	0.39	0.35	**0.43**	0.36	**0.33**
External Videos						
Direction	0.28	0.22	0.27	**0.20**	0.27	0.26
Location	0.37	0.29	0.36	0.33	**0.39**	**0.26**
Speed	0.23	**0.17**	0.22	0.21	0.23	0.22
External Videos with Occlusion						
Direction	0.25	0.21	0.24	**0.16**	0.18	0.25
Location	0.37	0.32	0.34	0.33	0.37	**0.26**
Speed	0.21	**0.25**	0.21	0.21	**0.25**	**0.18**

Table 1: Overall accuracies for different filter types and 1D-functions

direction based 1D-functions properly. Furthermore for each 1D-function of our own videos there is at least one filter type that increases the accuracy. Especially for occluded videos classified by directional motion data we measure a significant accuracy increase. In this case Lee filter raises the accuracy from 0.70 to 0.81.

For occluded videos and 1D-functions derived by the speed of image moments there is a further significant increase. Here the average filter increases the accuracy from 0.35 to 0.43. With respect to external video data there are only three cases with an accuracy improvement. External videos contain more irregular motions, which again means that for instance the maximum filter substitutes values by maximal noise values and increases therefore the number of false classifications. Moreover the Laplace filter emphasizes noise and the average filter reduces important high frequency amplitudes, which are typical for some external videos.

An overall comparison of all filters leads to the result that the Lee filter is the most accurate filter for repeating motion based video classification. Accuracy increases can be strong and decreases are slight. Here the selective noise reduction seems to be effective. On the other hand Laplace filter tends to increase noise. Hence almost all experimental results show up accuracy decreases. Besides the average filter works only for videos containing clear and smooth motion.

Table 1 shows that Lee filter raises accuracy by 0.11 for directional centroid data of own and occluded videos. Average filter raises accuracy by 0.08 for 1D-functions based on the centroid's speed. By contrast 1D-functions based on the centroid's location do not show any remarkable accuracy raise by applying filters. The reason is that an occlusion influences location based 1D-functions in various ways. Different parts of the frequency domain can be emphasized or declined, whereby filters cannot compensate these changes. Beyond that the location based 1D-functions are the most robust ones, because an occlusion has a minor effect on the overall motion trajectory.

By adding occlusion to video frames the centroid's speed is often raised. This leads to clearer highs and lows inside the 1D-function. Considering that speed information in general is noisy, these clear highs and lows become only apparent in the frequency domain, when the average filter is applied.

Furthermore occlusion weakens the clarity of motion, consequently the centroid direction becomes noisy. Most often this noise stays below a certain amplitude value, so that the Lee filter can remove exactly this specific noise type. This improvement becomes even more apparent, if the original movement without occlusion was wide and clear. In figure 5 classes paint roller, plane and wrench confirm this behavior. Since we consider 10 classes with 20 videos, the maximal number of proper classifications is 20 for each class.

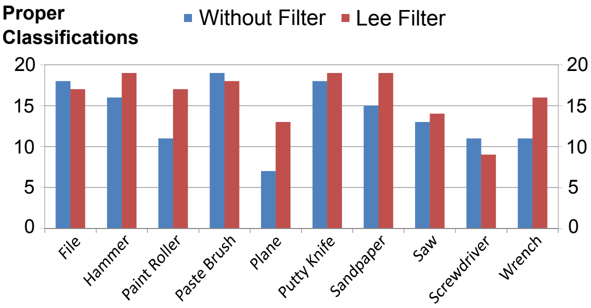

Figure 5: Number of proper classifications with and without Lee filter for occluded own videos

Concluding we can state that filtering 1D-functions can improve accuracy in some cases, but on the whole filters reduce the system's accuracy. They reduce the information content or emphasize noise for motion trajectories, so that the resulting feature vectors cannot be assigned properly.

Runtime Analysis

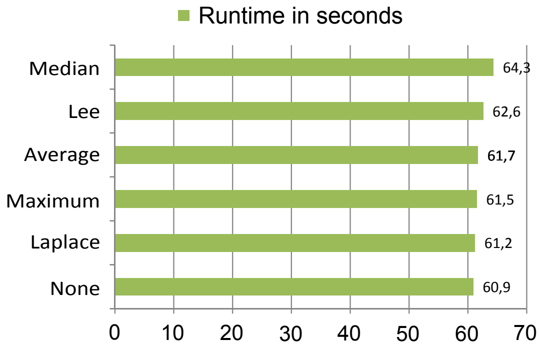

Figure 6: System's runtime with different filters

Figure 6 shows runtime results for each introduced filter. For runtime analysis a 2.2 GHz CPU is used. We assign 1000 videos to one out of 10 classes containing home improvement video data (see figure 5). We reuse our 200 videos covering database five times. Each class consists of 20 videos and each video again consists of 512 frames with a 320×240 resolution. Moreover the filter radius is set to 10. Depicted filter runtimes are averages of five separate test iterations. Averaging is necessary, since runtime differences are marginal and system operations can influence the runtime.

Figure 6 shows up small runtime increases, when filters are applied. Standard classification without filter takes 60.9 seconds for 1000 videos. Applying Laplace, maximum or average filter the runtime increase stays below 1 second. These three filters have got similar algorithmic setups. Utilizing Lee filter runtime is 62.6 seconds and therefore longer than the runtime for the previous three filters. Due to additional operations in order to find out the variance, Lee filter requires more runtime. Further on we measure a maximum runtime at 64.3 seconds for median filter. The median filter has to arrange data values in order to find a median. Sorting data values needs more operations than calculating the variance. Thus median filter takes more runtime than Lee filter.

7 CONCLUSION

In this paper we have shown a video classification system based on the frequency of repeating movements. Frequency spectra are computed by transforming spatio-temporal image moment trajectories (1D-functions). The experimental part focused on filtering 1D-functions in order to receive more decisive frequency domains. Test results show that the Lee filter performs best, since this filter smoothens only noisy parts of a 1D-function. However maximum or Laplace filter reduce the system's accuracy in most cases, because either high frequencies are smoothed too strongly or noisy parts are emphasized, respectively. Runtime analysis turns out that Lee filter needs more operations than maximum, average or Laplace filter, but less operations than median filter. Applying filters to 1D-functions can improve the system's accuracy in some cases, but in general the accuracy is decreased. Particularly smoothing filters like maximum, median and average filter reduce the information content.

But there are still edge detection filters as the Prewitt filter or noise removing filters as the harmonic mean filter, which have to be analyzed and could reveal more accurate test results.

8 REFERENCES

[AS2001] Alsultanny, Y. and Shilbayeh, N., Examining filtration performance on remotely sensing satellite images, SSIP, pages 75–80, 2001.

[AC2012] Ayyildiz, K. and Conrad, S., Video classification by partitioned frequency spectra of repeating movements, WASET, pages 154–159, 2012.

[CCK2004] Cheng, F., Christmas, W., and Kittler, J., Periodic human motion description for sports video databases, ICPR, pages 870–873, 2004.

[FZP2005] Fanti, C., Zelnik-Manor, L., and Perona, P., Hybrid models for human motion recognition, CVPR, pages 1166–1173, 2005.

[LEE1980] Lee, J., Digital image enhancement and noise filtering by use of local statistics, TPAMI, pages 165–168, 1980.

[MAS1985] Mastin, G. ,Adaptive filters for digital image noise smoothing: An evaluation, CVGIP, pages 103–121, 1985.

[MLH2006] Meng, Q., Li, B., and Holstein, H., Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach, IVC, pages 795–809, 2006.

[VUE2010] Vanithamani, R., Umamaheswari, G., and Ezhilarasi, M., Modified hybrid median filter for effective speckle reduction in ultrasound images, ICNVS, pages 166–171, 2010.

[VYB1989] Vliet, L., Young, I., and Beckers, G., A nonlinear Laplace operator as edge detector in noisy images, CVGIP, pages 167–195, 1989.

[WSL1995] Wong, W., Siu, W., and Lam, K., Generation of moment invariants and their uses for character recognition, PRL, pages 115–123, 1995.

[YT2010] YouTube, L., Youtube: Broadcast yourself, `www.youtube.com`, 2010.