On-line human action detection using space-time interest points

Ivo Reznicek and Pavel Zemcik

Faculty of Information Technology, Brno University of Technology,
Brno Bozetechova 1/2, 612 66 Brno , Czech Republic,
ireznice@fit.vutbr.cz,
WWW home page: http://www.fit.vutbr.cz/∼ireznice
zemcik@fit.vutbr.cz,
WWW home page: http://www.fit.vutbr.cz/∼zemcik

Abstract *The on-line human action detection is an important task in human-machine interaction and related applications. One of the possible approaches to the detection is exploitation of space-time interest points. Such points are typically identified using feature extractor and then they are processed and classified. The classification can be performed using codebooks built based on feature vectors statistics. The individual feature vectors are transformed into bag of words representation using such codebooks and then the code words are classified using SVM. The proposed approach improves the training process and extends the known approaches. The training part of the dataset is split into shorter shots with equal duration and these are annotated and classified using a SVM classifier. This ensures that the time-local motion is captured by the SVM while the longer time behavior is left on further processing mechanisms, such as, e.g. HMMs. In the proposed approach, the output of the SVM classifier is simply compared to a threshold and the presence of a value above the threshold indicates that the desired human activity occurred. The contribution describes the approach summarizes the achieved results and draws conclusions.*

1 Introduction

Human behavior detection is one of the most wanted functionality in the surveillance systems; the detection of unusual behavior has a potential to increase security of monitored public places similarly to the object detection.

Object detection can helping in handling potentially dangerous situations, such as, left baggage (may contain some harmful objects), animals (in places, where they are not allowed to occur), etc. Human behavior detection can avoid problems in places, where some activities are prohibited, for example, smoking, using a cellphone, etc.

On the other side, human behavior detection may help in other situations, such as interactive distributed discussions or teleconferences, where the bigger amount of participants obviously want to say something, try to take the word, etc. While attempting these actions, all of them are performing some gestures which can be detected and successfully used for teleconference system.

The human behavior detection may be achieved using the space-time interest points processing. Significant amount of related work for space-time interest point detectors and their processing was published [1], [6], [5]. These approaches are presented below and all share one main disadvantage; the algorithms are working offline and the detectors are not able to process the long video streams. Typically, the space-time feature vectors are extracted as follows:

– the whole video sequence is stored in the memory,
– the video "cube" is analyzed and interest points are established,
– the feature vector is created for each interest point.

Growing the resolution of the video to be procesed requires increase of the memory which needs to be used, the amount of the memory is increased to the order of units of gigabytes; this of cource is not feasible.

The whole system contains the feature extractor, feature vectors processing unit and the output classifiers. The whole procedure needs to be converted into the on-line version. Globally, the most problematic part is the feature extractor; without its online version, the online action detection system cannot be built.

This paper presents the approach for converting the offline processing to online system, which can be applied to various types of feature extractors. Another issue is the ability to process the input video frames in real-time (in this case, we need to speed-up the online algorithm to run in wanted environment, CPU type, memory size etc.). The next part of the work is the presentation of the on-line detection system. This system yields many input parameters which need to be set and which affect the quality of the detection process.

This paper is organised as follows: In Chapter 2, the space-time interest points extractors and following description processes are presented; the structure of the on-line system is presented. Chapter 3 presents the algorithm for feature extractor adaptation to work in on-line mode, Chapter 4 presents the settable parameters of all components of the online system and discusses the possibilities of its setting and evaluation. Chapter 5 describes the dataset used for creation of classifiers, the definition of human actions and describes all experiments and their results. Finally, Chapter 6 concludes the paper with the discussion.

2 Interest points extractors and processing

This Chapter introduces the possibilities of space-time interest points detection, description process, and the whole proposed on-line system with all its components.

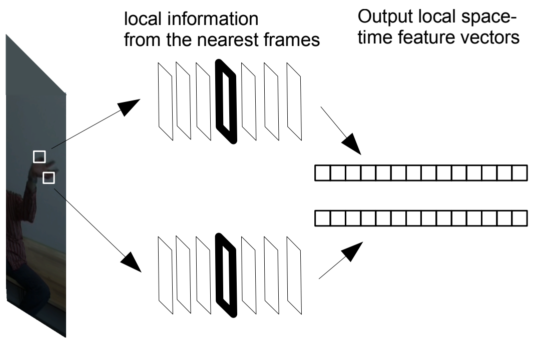

Figure 1. space-time interest points visualization

In Figure 1 shows the scheme of the principle of space-time feature extraction. The video stream is processed frame by frame and the local extrema are located according to two domains; the spatial domain is represented by the width and height of the single video frame; the temporal domain is represented by time domain which is digitalized using the video frame rate. When these local extrema are determined, the local neighborhood of the extrema is used for computing of the extrema local descriptor. These descriptors are computed from all detected extrema and the feature extraction process returns the list of extrema positions and its descriptors.

2.1 Space-time interest points detection

Laptev and Lindeberg proposed [1] the *Harris3D* space-time interest points detector; it is extension of the Harris detector [8]. The spatio-temporal second-moment matrix is computed at each video point

$$\mu(\cdot;\sigma,\tau) = g(\cdot;s\sigma,s\tau) * (\nabla L(\cdot;\sigma,\tau)(\nabla L(\cdot;\sigma,\tau))^T) \quad (1)$$

using independent spatial and temporal scale values σ, τ, a separable Gaussian smoothing function g and space-time gradients ∇L. The final locations of space-time interest points are given by local maxima of the following function:

$$H = det(\mu) - k\ trace^3(\mu), H < 0, k \approx 0.005 \quad (2)$$

Another approach is proposed by Dollr et al. [6], it is called *Cuboids* detector and it is based on a spatial Gaussian smoothing filters and temporal Gabor filters. The response function has the form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \quad (3)$$

where $g(x, y; t)$ is the 2D spatial Gaussian smoothing kernel, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters, which are applied only temporally. The Gabor filters are defined by $h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$ with $\omega = 4/\tau$. The two parameters σ and τ of the response function R corresponds approximately to the temporal and spatial scale of the detector. The interest points are defined as the local maxima of the response function R.

The *Hessian* detector was proposed by Willems et al. [5] as a spatio-temporal extension of the Hessian saliency measure for blob detection in images. The detector measures the saliency with the determinant of the 3D Hessian matrix. The position and scale of the interest points are simultaneously localized without any iterative procedure. In order to speed up the detector, the approximative box-filter operations on an integral video structure may be used. Each octave is divided into 5 scales, with a ratio between subsequent scales in the range 1.2 – 1.5 for the inner 3 classes. The determinant of the Hessian is computed over several octaves of both the spatial and temporal scales. Finally, a non-maximum suppression algorithm selects joint extrema over space, time and both scales.

Dense sampling may be used for defining the space-time interest points. The whole spatio-temporal space is covered by uniform 3D grid which defines the points which will be used for description; instead of evaluation of some response function. The space-time interest points may be generally extracted with different settings for spatial and temporal scales and spatial and temporal grid parameters; they may be individually adjusted according to the processed situation.

2.2 Space-time interest points description

The *HOG/HOF* descriptors were introduced by Laptev et al. [1]. Local motion an dapperance is characterized using histograms of spatial gradient and optic flow accumulated in space-time neighborhoods of detected interest points. For the combination of HOG/HOF descriptors with interest point detectors, the descriptor size is defined by $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma, \Delta_t(\tau) = 8\tau$. Each volume is divided into a $n_x \times n_y \times n_t$ grid of cells; for each cell, 4-bin histogram of gradient orientations (*HOG*) and 5-bin histogram of optic flow (*HOF*) are computed. Histograms are further normalized; parts can be used separately or as a combined feature vector.

The *HOG3D* descriptor was proposed by Kläser et al. [2]. It is based on histograms of 3D gradient orientations and can be seen as an extension of the popular SIFT descriptor [3] to video sequences. The gradients are computed using an integral video representation. Regular polyhedrons are used to uniformly quantize the orientation of spatio-temporal gradients. Therefore, the descriptor, combines shape and motion information at the same time. A given 3D patch is divided into $n_x \times n_y \times n_t$ cells. The corresponding descriptor concatenates gradient histograms of all the cells and is then normalized.

ESURF (extended SURF) which extends the image SURF descriptor [4] proposed by Willems et al. [5]. Similary to previous desctiptor, the 3D patches are divided into $n_x \times n_y \times n_t$ cells. The size of the 3D patch is given by $\Delta_x(\sigma) = \Delta_y(\sigma) = 3\sigma, \Delta_t(\tau) = 3\tau$. For the feature descriptor, each cell is represented by a vector of weighted sums $v = (\sum d_x, \sum d_y, \sum d_t)$ of uniformly sampled responses of the Haar-wavelets d_x, d_y, d_t along the three axes.

The *Cuboid* descriptor was introduced by Dollar et al. [6]. At each interest point, position of the small cube is extracted and called cuboid. On such cuboids, the transformations are applied: (1) pixel values normalisation, (2) brightness gradient, and (3) windowed optical flow. The brightness gradient is calculated at each spatio-temporal location (x, y, t), giving rise to three channels (G_x, G_y, G_t) each the same size as the cuboid. Lucas-Kanade optical flow [7] can be extracted to obtain motion information; it is calculated between each pair of consecutive frames creating two channels (V_x, V_y). Each channel is the same size as the cuboid minus one frame.

The resulting feature vector can be created using the results of one of the methods presented above or their combination. The simplest method involves flattening of the cuboid to the vector; however, the resulting vector is potentially sensitive to small cuboid perturbations. An alternative method involves histogramming the values in the cuboid. Such a representation is robust to perturbations but also discards all positional information (spatial and temporal). Thus the cuboid is divided into a number of regions and a local histogram is created for each region. The goal is to improve robustness to small perturbations while retaining some positional information. The possible output feature vector is then the joined local histograms of optical flow or the joined local histogram of gradient extracted into a cuboid. The descriptor quality may be increased when both of these representations are joined together and produce the longer local feature vector.

2.3 Interest points processing pipeline

The input video stream is processed as follows (see Figure 2); the space-time interest points are extracted from the individual frames of input video and are stored to the queue in the memory. The length of the queue is constant and is defined by the length of the video sub-shot which is used for classification of the activity in the input video.

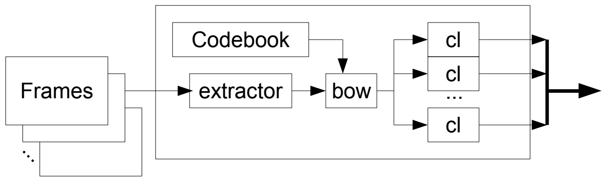

Figure 2. On-line processing system schema

When the queue is filled, its content is used to create the bag-of-words [11] representation (feature vector with constant size) which characterizes the current video sub-shot in the new feature space.

The main component needed for creatinon of such vectors is the codebook. It is created from the training part of the dataset (see Chapter 5) using the clustering algorithm. Subset of the local feature vectors is selected and used for the modelling such local feature space and clustering algorithm creates the finite number of representatives of local feature space.

As mentioned above, the codebook is used for translation of local features of the current video sub-shot into constant size feature vector representation; the dimensionality of resulting bag-of-words representation equals to the codebook size. The output feature vectors are classified using the series of classifiers (one classifier for every distinguished class) whose response is thresholded to achieve the decision whether the activity in the video is presented or not.

The bag-of-words creation and classification do not have to be specifically converted into the on-line version unlike the feature extractors. Their purpose is generally to do the conversion of finite set of input data (associated to currently processed shot) into output data. For bag-of-words descriptor creation, the input low level feature vectors are converted to the output descriptor using codebook, while in classification task, the input feature vector is converted into the classifiers response. None of these procedures is too computationally time-consuming task in comparison to the extraction of the space-time features.

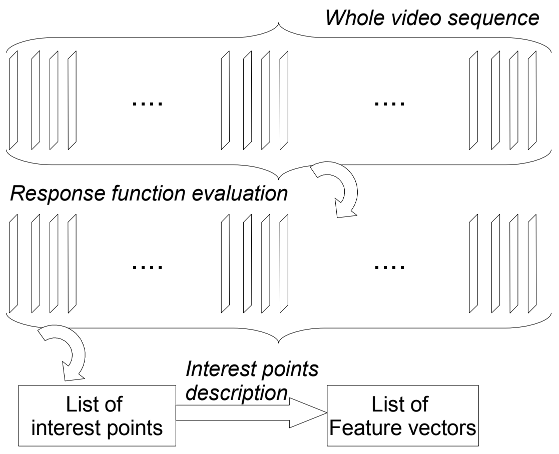

Figure 3. Offline feature extractor schema

3 On-line adaptation

The two main components of the space-time interest points extractor are the *detection of the interest points* and the *output descriptors creation function* that must be performed for all such points.

In the offline processing (see Figure 3), the whole input video cube is presented in memory (usually another same-sized cube with the response function is created in the memory and the response cube is processed with the algorithm to obtain the list of interest points). The list is sequentially processed to obtain the output descriptors; the input data for that procedure is usually obtained from the input video cube.

The on-line processing adaptation is depicted in Figure 4. The main approach is to extend the extractor using two queues, *the input video queue* and *the partial response queue*. The Input video queue is filled by input video frames and the partial response queue is used for storing the partial results of the response function. The interest points detection is running on the partially collected results of response function and the points list is established. The points description process is then executed similarly to the offline mode; the difference from the offline processinh is that the input data is read from the input image queue and the sufficient number of frames has to be present in that queue.

The whole process is constrained and every subcomponent needs some amount of results of preceding operations to produce its results. The size of both of the queues must fulfill the following equation: $s_q \geq max(l_r, l_{ep}, l_{dc})$, where l_r – the minimum number of frames needed for the response function evaluation for one frame; l_{ep} – the minimum number of frames for

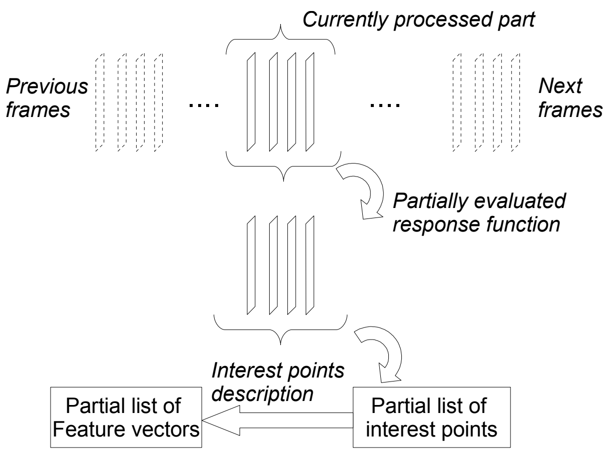
Currently processed part
Previous frames
....
....
Next frames
Partially evaluated response function
Interest points description
Partial list of Feature vectors
Partial list of interest points

Figure 4. On-line feature extractor schema

which the response function must be evaluated in order to get all the interest points for the current frame; l_{dc} – number of frames needed for descriptor creation.

The above presented procedure can be summarized in following way:

Preparation steps:

1. Create the queue of input images with minimum length s_q.
2. Create the queue for response function with minimum length s_q.
3. Re-implement all functions affected by added queues.
4. Fill in the input queue with the video frames.

Processing steps:

1. Execute the response function for all computable frames and insert the results to the response function queue.
2. If the response function queue is filled enough:
3. (a) Analyze the response function queue and establish the space-time interest points for the computable frame;
   (b) execute the description process for all interest points detected , in the step above and produce the feature vectors.
4. Add the new frame into the queue and go to step 1; perform the procedure for all frames in input video.

4 Parameters selection

The whole system presented in Chapter 2.3 contains number of procedures which have many parameters which have to be adjusted. The main resource for obtaining these parameters is the annotated dataset of actions.

The dataset is usually organised as a set of videos with annotations describing occurence of specific actions, and has two main parts: training part and testing part. Training part is used for adjustment of components, such as codebook and classifiers, and the testing dataset is used to assess the quality of the whole processing.

The input parameters of all the components can be searched in many ways, the simplest one being the brute-force technique which we used. The set of all the possibilities is identified and all of them are computed and evaluated; the best "setup" is selected as a final solution.

The main parameters that needs to be adjusted in the proposed on-line system are:

- Feature extractor parameters,
- the classified shot duration,
- size of the codebook used for bag-of-words processing,
- resolution of the input video,
- the classifier creation general parameters.

The *classifier creation* parameters are dependent on the type of the classifier used and may be optimized while the training process is in progress. For example, the cross-validation and bootstrapping approaches may be used.

Cross-validation uses splitting of input data to divide the input data into exact number of subparts. Usually, the bigger amount of subparts is used for training and the rest is used for testing. In such setup, all possibilities of training parameters are tried and evaluated. Thhen the splitting of the subparts is changed and the whole process is repeated several times; finally, the algorithm selects the best solution.

Bootstrapping approach is based on selecting training subparts the same way as described above, the training is performed on specific subpart and the testing is performed on the second one. When the training is evaluated, the miss-classified part of the testing subpart is removed from the testing subpart and added to the training one. Training is then repeated and the testing subpart is updated again in the similar way. After several iterations, the process generates the classifier model.

The methods mentioned above may improve the classifier creation process such so that we can see the classifier creation process as a black box where only the input feature vectors with annotations are known and the process creates the best model automatically. The model is then used for evaluation using the testing part of dataset. This improvement can be used for classifier creation part of the system and yields the reduction of the systems' parameters space.

The *input video resolution* parameter has to be set according to the constitution of the video data used. If the detected action is performed in the small area of the acquired video shot the reduction of the resolution shurely decreases the output system quality. In the case when the detected action is performed on the whole video frame, the reduction of the resolution may decrease the quality minimally or the accuracy may be even increased. The reduction of the video resolution has the main advantage, namely, the processing speed is increased.

The *shot duration* parameter has to be set according to the constitution of the actions detected in the video. It can be expected that longer actions need longer video sub-shot to be processed but the longer delay the final classifier decision on the sub-shot.

Feature extractor parameters are dependent on the type of the extractor. When dense sampling is used in such extractor, its basic parameters are the space and temporal step that control the creation of the spatio-temporal grid and also control the number of points in such grid.

Creation of the codebook component and all classifiers has to be performed using a framework, which is able to automate the process. This procedure is called *learning phase* and its purpose is creation of the codebook and the classifiers components.

Learning phase extracts all feature vectors from all videos in the training and testing dataset. Using the training part, the codebook is created and all local low level features are translated to bag-of-words representation. The training bag-of-words vectors are used for creation of the output classifiers while the testing ones are used for accuracy checking. The whole process has to be repeated for each input parameters settings.

5 Experiments and results

The proposed approach was used to study the human behavior during the place-distributed video conference where the particular groups of participants are connected using the internet connection. All groups have the television and a set of cameras in the room and the conference controlling system needs to do the cutting of the incoming video streams acquired using all cameras and produces the output video which is showed on the televisions. The detection of such human actions may significantly improve the cutting process.

The detected actions were waving and clapping performed by the persons in the video. For creation of such on-line detection system, we have created dataset with the following properties: 15 videos, 9 performing persons, each video contains the block where the waving is performed, the block where the clapping is performed, and also the block where the other activities are performed. All of these videos are annotated and the resolution of the videos is cropped to 380x426 pixels; the unused regions of the video (in which nothing is happening) were removed.

As the offline testing framework, our solution [9] has been used with some amount of modifications. In all of our experiments, the dense sampling feature extractor was used; the interest points description is done using the cuboids descriptors based on local histograms of brightness gradients (see subchapter 2.2). The linear SVM classifier [10] was used and the classifier parameters were set automatically using the cross-validation approach; thus, the main processing parameters which are left and need to be adjusted are:

- The processed size of the video sub-shot,
- the dense sampling parameters of the feature extractor,
- the codebook size,
- the resolution of the input video.

The parameter space of the system is very large and it presents full evaluation of all parameters. For examination of the system properties, some subset of the parameters space needed to be selected.

The metrics for evaluation of system quality were used the average precision which is defined as the area under the precision-recall curve [12]. The system quality denotes the frequency of miss-predicted samples.

The subspace of parameters space has been searched. The results achieved are presented in Table 1; The Shot column depicts the length of classified shot (in frames). Res column depicts the input resolution parameter, R means the regular (full) size, S means the shrunk size. DS and DT column specifies the dense sampling parameters for the spatial domain (DS) and the temporal domain (DT). CBsize column specifies the length of the codebook used and WAVING and the CLAPPING columns specify the average precision of the resulting classifier using specified creation parameters.

Table 1 shows all the possible combinations of the input parameters which were tried in our experiments. The boundaries of the shot duration parameter was specified to the interval from 40 to 60. The main goal is to set this parameter to the minimum possible value. The numbers above this interval can be used and possibly can gain the system accuracy, but the extension of this parameter causes the extension of the detection delay; in the environment, where the resulting system will be used, is this extension infeasible.

The resolution of the input video parameter was set to two sizes: The regular resolution (the original resolution) and the shrunk one (the resolution is reduced to one half in both ways). The interesting research can be done in this parameter evaluation, the bigger amount of possibilities of this parameter should be evaluated; The most interesting may be the survey

Table 1. The accuracy of the system (average precision measure) as a dependency to the input parameters.

Shot	Res	DS	DT	CBsize	WAVING	CLAPPING
40	R	5	5	512	0.602	0.487
				4096	0.666	0.614
				6100	0.655	0.591
40	S	5	5	512	0.679	0.546
				4096	0.725	0.577
				6100	0.717	0.648
40	R	10	5	512	0.619	0.705
				3000	0.632	0.614
				4096	0.659	0.614
				5000	0.610	0.614
				6100	0.618	0.591
40	S	10	5	512	0.491	0.679
				3000	0.502	0.653
				4096	0.528	0.620
				5000	0.501	0.674
				6100	0.540	0.666
42	R	7	7	512	0.559	0.669
				3000	0.548	0.646
				4096	0.568	0.629
				5000	0.531	0.616
				6100	0.558	0.608
42	S	7	7	512	0.455	0.323
				3000	0.468	0.405
				4096	0.550	0.456
				5000	0.511	0.461
				6100	0.533	0.456
49	R	7	7	512	0.591	0.649
				3000	0.509	0.545
				4096	0.548	0.595
				5000	0.559	0.635
				6100	0.547	0.603
49	S	7	7	512	0.484	0.354
				3000	0.552	0.403
				4096	0.580	0.384
				5000	0.512	0.433
				6100	0.591	0.439
55	R	5	5	512	0.637	0.612
				4096	0.583	0.615
				6100	0.698	0.663
55	S	5	5	512	0.684	0.598
				4096	0.714	0.654
				6100	0.591	0.489
60	R	5	5	512	0.640	0.644
				4096	0.705	0.601
				6100	0.706	0.551
60	S	5	5	512	0.680	0.606
				4096	0.74	0.62
				6100	0.658	0.572

of the turning point, where the system accuracy drastically decreases. This evaluation also depends on the constitution of the videos in the training dataset and on the size and the position of the object which actions needed to be recognised.

The dense sampling parameters were set experimentally and are closely related to the video resolution parameter. For example, when the video resolution is reduced to one half in both ways, it is the same as the increasing of the extractor's spatial sampling parameters two times, in both adjustments, the feature extractor produces the same results. In Table 1 can be seen , that the extension of the dense sampling parameters to numbers higher than 10 in spatial domain and the numbers higher than 7 in temporal domain will be probably causing significant decrease of systems accuracy.

The codebook size were selected experimentally, similarly as in image categorisation task (where the image features are used). The optimum codebook size cannot be easily recommended; every experiment has its own best codebook size and it cannot be said generally that some size is the best one. The experiments suggest the need to try another codebook configurations, such as 256 words and some number bigger than 6100 should be explored too.

In Table 1, the configurations in which the system produces the best solution are marked with underline for both classes (it is anticipated that only one codebook will be used in the future). The last but not least aspect for whole system is the processing speed. It can be affected by *dense sampling* parameters and *the resolution of the video* parameter. The increasing of the dense sampling parameters causes that the smaller number of the space-time interest points needed to be described in the feature extractor. Analogous situation is in the input video resolution parameter. When building analogous system and when some combination of the parameters needed to be selected, the best solution is to execute some amount of experiments and search the parameter space and then select the best parameters combination according to the systems requirements, such as the processing speed, the memory size constraint and the output system accuracy.

6 Conclusion

This contribution presented a novel approach to online space-time interest point processing intended for human action detection. The approach has been successfully designed, implemented, and evaluated from the point of view both computational requirements and latency caused by the number of video frames needed for detection of the events.

The evaluation of the approach also included density of sampling, length of video sequence (number of frames), resolution of frames parameter evaluation with respect to its impact on gesture recognition performance.

Future work includes further evaluation especially of the latency issues in order to enable as quick as possible "on-line" reaction to the events in video.

Acknowledgement

This work has been supported by the project TA2: Together Anywhere, Together Anytime ICT-2007-214793, grant no 214793.

The research has also been supported by the Czech Ministry of Education, youth and sports grant project Security-Oriented Research in Information Technology MSM0021630528 and Faculty of Information Technology, Brno University of Technology internal grant project Advanced recognition and presentation of multimedia data FIT-S-11-2.

References

1. Laptev, I., Lindeberg, T.: Space-time interest points. In ICCV (2003)
2. Kläser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3D-gradients. In BMVC (2008)
3. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004), 91-110
4. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In ECCV (2006)
5. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest points detector. In ECCV (2008)
6. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In VS-PETS (2005)
7. Lukas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In IJCAI (1981), 674-679
8. Harris, C., Stephens, M.J.: A combined corner and edge detector. In Alvey Vision Conference (1988)
9. Reznicek, I., Barina, D.: Classifier creation framework for diverse classification tasks. In: Proceedings of the DT workshop, Zilina (2010)
10. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9 (2008), 1871-1874
11. Willamowski., J., Arregui, D., Csurka, G., Dance, C.R., Fan, L.: Categorizing nine visual classes using local apperance descriptors. In ICPR Workshop on Learning for Adaptable Visual Systems (2004)
12. Olson, David L., Delen, Dursun: Advanced Data Mining Techniques. Springer; 1 edition (February 1, 2008)