I J C T A, 9(2) 2016, pp. 709-716

Human Action Recognition Using LK Method and SVM Classifier

P. Kalaivani*1 and S. Vimala#2

ABSTRACT

In this paper we addresses the view-independent activity recognition from a different perception. In the geometry-based methods we require identification of body parts and the estimation of analogous points between the video sequences. The earlier view-based methods assume multi-view action samples for training and the testing of images. But in this paper we explored the human action video as the training video and test with another action video by matching the local self-similarity descriptors. The training video is taken to test for the anomaly action recognition. Also the class of actions are analyzed and tested whether they are like walking, running or jumping actions. The human action recognition under different view changes is taken with angle 0, 45, 90 degrees. In this we have proposed Lucas–Kanade (LK) algorithm for optical flow analysis and Support Vector Machine (SVM) for classification.

Keywords: Static camera, multi-view, self-similarity, LK Algorithm, SVM.

I. INTRODUCTION

Human activity recognition is an important area of computer vision research today. The goal of human activity recognition is to automatically analyze ongoing activities from an unknown video (i.e. a sequence of image frames). In a simple case where a video is segmented to contain only one execution of a human activity, the objective of the system is to correctly classify the video into its activity category. In more general cases, the continuous recognition of human activities must be performed by detecting starting and ending times of all occurring activities from an input video. The ability to recognize complex human activities from videos enables the construction of several important applications. Automated surveillance systems in public places like airports and subway stations require detection of abnormal and suspicious activities, as opposed to normal activities. For instance, an airport surveillance system must be able to automatically recognize suspicious activities like "a person leaving a bag" or "a person placing his/her bag in a trash bin." Recognition of human activities also enables the real-time monitoring of patients, children, and elderly persons. The construction of gesture-based human computer interfaces and vision-based intelligent environments becomes possible with an activity recognition system as well. The human activity and posture recognition have been extensively studied during the past few years. A detailed survey of video based motion and activity recognition systems is discussed in [1, 2]. The projects successfully implemented for abnormal human activity recognition.

There are various types of human activities. Depending on their complexity, we conceptually categorize human activities into four different levels: gestures, actions, interactions, and group activities. Gestures are elementary movements of a person's body part, and are the atomic components describing the meaningful motion of a person. "Stretching an arm" and "raising a leg" are good examples of gestures. Actions are single-person activities that may be composed of multiple gestures organized temporally, such as "walking,"

*1 Ph.D. Scholar, Department of Computer Science, Mother Teresa Women's University, Kodaikanal, India, [1]*vanivijay2012@gmail.com*

#2 Associate Professor, Department of Computer Science, Mother Teresa Women's University, Kodaikanal, India, [2]*vimalaharini@gmail.com*

"waving," and "punching." Interactions are human activities that involve two or more persons and/or objects. For example, "two persons fighting" is an interaction between two humans and "a person stealing a suitcase from another" is a human-object interaction involving two humans and one object. Finally, group activities are the activities performed by conceptual groups composed of multiple persons and/or objects: "A group of persons marching," "a group having a meeting," and "two groups fighting" are typical examples.

II. LUCAS–KANADE METHOD FOR OPTICAL FLOW ESTIMATION

In computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It assumes that the flow is essentially constant in a local neighbourhood[3,4] of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion.

By combining information from several nearby pixels, the Lucas–Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

The Lucas–Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p. Namely, the local image flow (velocity) vector (V_x,V_y) must satisfy

$$
\begin{aligned}
I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\
I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\
&\vdots \\
I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n)
\end{aligned}
\tag{1}
$$

where $q_1, q_2, \ldots, q_n$ are the pixels inside the window, and $I_x(q_i)V_x$, $I_y(q_i)V_y$, $I_t(q_i)$ are the partial derivatives of the image I with respect to position x, y and time t, evaluated at the point q_i and at the current time. These equations can be written in matrix form $Av = b$, where

$$
A = \begin{pmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}
$$

and

$$
b = \begin{pmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{pmatrix} \tag{2}
$$

This system has more equations than unknowns and thus it is usually over-determined. The Lucas–Kanade method obtains a compromise solution by the least squares principle. Namely, it solves the 2×2 system

$$
A^T A v = A^T b
$$

or

$$v = (A^TA)^{-1}A^Tb$$

where A^T is the transpose of matrix of A. That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \tag{3}$$

with the sums running from i=1 to n.

The matrix $A^T A$ is often called the structure tensor of the image at the point p.

(A) The Weighted Window

The plain least squares solution above gives the same importance to all n pixels q_i in the window. In practice it is usually better to give more weight to the pixels that are closer to the central pixel p. For that, one uses the weighted version of the least squares equation,

$$A^TWAv = A^TWb$$

or

$$v = (A^TWA)^{-1}A^TWb$$

where W is an n×n diagonal matrix containing the weights $W_{ii} = w_i$ to be assigned to the equation of pixel q_i. That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i)I_y(q_i) \\ \sum_i w_i I_x(q_i)I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i w_i I_x(q_i)I_t(q_i) \\ -\sum_i w_i I_y(q_i)I_t(q_i) \end{bmatrix} \tag{4}$$

The weight w_i is usually set to a Gaussian function of the distance between q_i and p.

(B) LK Method Using Conditions And Techniques

In order for equation $A^TAv = A^Tb$ to be solvable A^TA should be invertible, or A^TA's eigen values satisfy $\lambda_1 \geq \lambda_2 > 0$. To avoid noise issue, usually λ_2 is required not too small. Also, if λ_1/λ_2 is too large, this means the point *p* is on an edge, and this method suffers from the aperture problem. So for this method to work properly, the condition is λ_1 and λ_2 are large enough and have similar magnitude. This condition is also the one for Corner detection. This observation shows that one can easily tell which pixel is suitable for Lucas–Kanade method to work on by inspecting a single image. One main assumption for this method is that the motion is small (less than 1 pixel between two images for example). If the motion is large and violates this assumption, one technique is to reduce the resolution of images first and then apply the Lucas-Kanade method.

(C) Improvements and Extensions

The least-squares approach implicitly assumes that the errors in the image data have a Gaussian distribution with zero mean. If one expects the window to contain a certain percentage of "outliers" (grossly wrong data values that do not follow the "ordinary" Gaussian error distribution), one may use statistical analysis to detect them, and reduce their weight accordingly.

The Lucas–Kanade method per se can be used only when the image flow vector $\mathbf{V_x}, \mathbf{V_y}$ between the two frames is small enough for the differential equation of the optical flow to hold, which is often less than the pixel spacing. When the flow vector may exceed this limit, such as in stereo matching or warped document registration, the Lucas–Kanade method may still be used to refine some coarse estimate of the same, obtained by other means; for example, by extrapolating the flow vectors computed for previous frames, or by running the Lucas-Kanade algorithm on reduced-scale versions of the images. Indeed, the latter method is the basis of the popular Kanade-Lucas-Tomasi (KLT) feature matching algorithm. A similar technique can be used to compute differential affine deformations of the image contents.

III. BACKGROUND SUBTRACTION METHOD

Background subtraction (BS) is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. As the name suggests, BS calculates the foreground mask performing a subtraction between the current frame and a background model, containing the static part of the scene or, more in general, everything that can be considered as background given the characteristics of the observed scene.

In 1990s background subtraction was known as a powerful preprocessing step but only in controlled indoor environments. In 1998, Stauffer and Grimson [5] presented the idea of representing each pixel by a mixture of Gaussians (MoG) and updating each pixel with new Gaussians during run-time. This allows background subtraction to be used in outdoor environments. Normally the updating was done recursively, which can model slow changes in a scene, but not rapid changes like clouds.

The method by Stauffer and Grimson has today become the standard of background subtraction. However, since 1998 a number of advances have been seen which can be divided into background representation, classification, background updating, and background initialization.

IV. ACTION RECOGNITION

The field of action and activity representation and recognition is relatively old, yet still immature. This area is presently subject to intense investigation which is also reflected by the large number of different ideas and approaches. In scene interpretation, the representations should be independent from the objects causing the activity and thus are usually not meant to distinguish explicitly, e.g., cars from humans. On the other hand, some surveillance applications focus explicitly on human activities and the interactions between humans. Here, one finds both, holistic approaches, that take into account the entire human body without considering particular body parts, and local approaches. Most holistic approaches attempt to identify ‘‘holistic’’ information such as gender, identity, or simple actions like walking or running.

Researchers using local approaches[6,7] appear often to be interested in more subtle actions or attempt to model actions by looking for action primitives with which the complex actions can be modeled. We have structured this review according to a visual abstraction hierarchy yielding the following: scene interpretation where the entire image is interpreted without identifying particular objects or humans, holistic recognition where either the entire human body or individual body parts are applied for recognition, and action primitives and grammars where an action hierarchy gives rise to a semantic description of a scene. Before going into these topics we first look closer at the definition of the action hierarchy used in this survey since it has influence on the remaining categories.

Ricky J. Sethi, Amit K. Roy-Chowdhury[8] presents a multi-disciplinary framework for recognizing human actions. Sotirios Spanogianopoulos et al [9], they review the basic principles and current methodologies used for collecting the raw gesture data from the user for recognize actions the users perform and the technologies currently used for gesture-HCI in games enterprise. Stefan Oniga, Jozsef Suto [10] presents the state of our work related to the development of an assistive assembly consisting of a smart and assistive environment, a human activity and health monitoring system, an assistive and telepresence robot, together with the related components and cloud services.

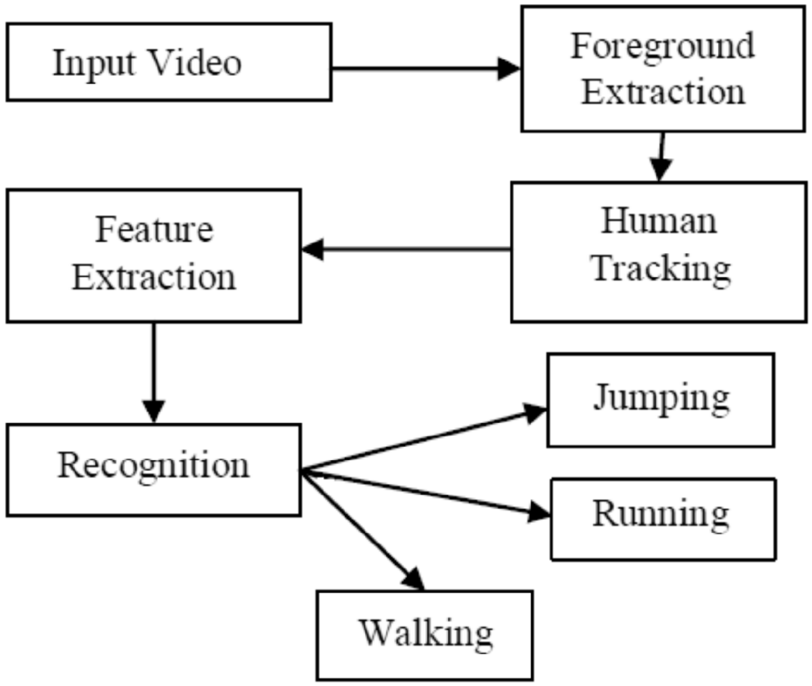

Figure 1: Our system model for human action recognition

In this paper we have analyzed the action sequences like walking running and jumping by extracting their feature using LK Algorithm. Fig. 1 shows our system model for human action recognition.

(D) Feature Extraction

Feature extraction is an important problem in action recognition. In this paper, we adopt two common features like optical flow and silhouette and try to represent them via global and local divisions. Our final feature descriptors are histograms of the divisional silhouette or optical flow. The detail of the process is following

Human Activity Recognition Algorithm

1) Input is fed to the system as a single video sequence.
2) Frames are extracted from the input video, which are used for further processing.
3) Convert the action video into a static image sequence and segment the motion target by using the method of background subtraction.
4) Obtain the moving target region and body silhouette using the morphology processing.
5) Extract optical flow between two continuous frames by using Lucas-Kanade algorithm[11].
6) SVM classifier is used for classification.
7) The features of each action from the parameters of human model acts as the features for classifying all four activities (walking, jumping, and running).
8) The features depend on the following criteria: Walking, Jumping, and Running.

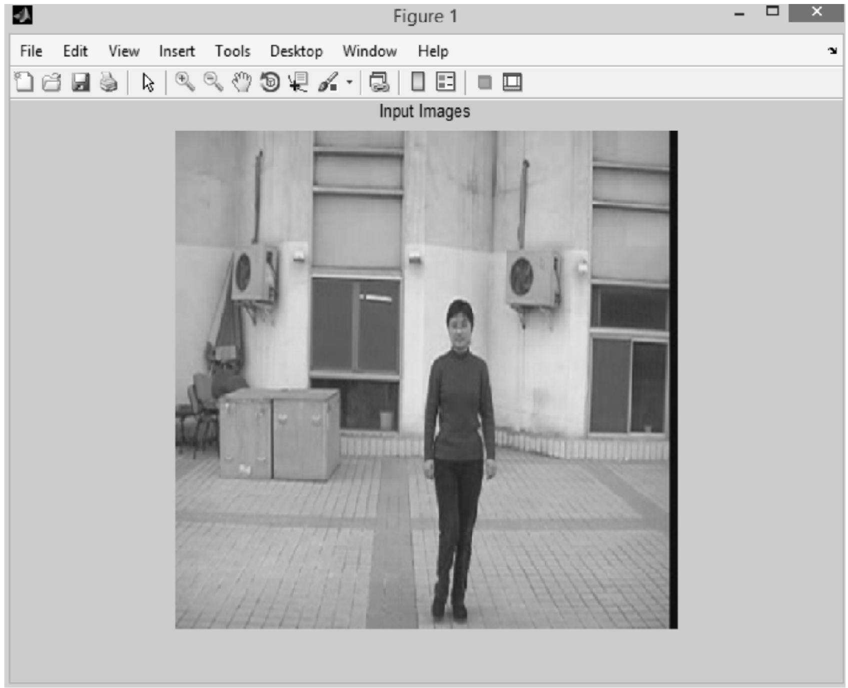

Figure 2: The 90º input image taken for action recognition

In Fig. 2 shows the 90º input image action recognition using the MATLAB software and Fig.3 shows the preprocessing image of that video. After the test video has been taken for classification and recognition of the action.

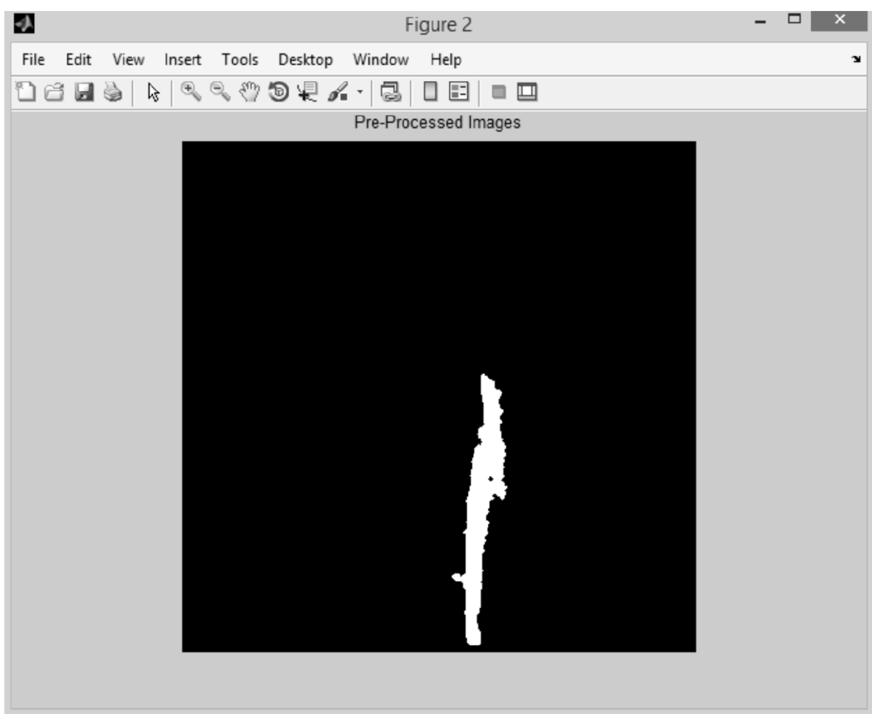

Figure 3: After preprocessing the input image for action recognition

(E) Support Vector Machine (SVM)

The Support Vector Machine is one of the most popular margin-based supervised classifier in the pattern recognition. It is used to separate a data set into two classes the training and the testing. The goal of designing an SVM is to find the optimal dichotomic hyperplane which can maximize the margin (the largest separation) of two classes.

The data points on the margin of the hyperplane are called support vectors. Schuldt *et al.* [12] apply SVMs to recognize human activities by extracting local space-time features in a video. Moreover, Laptev et al. [13,14] use a nonlinear SVM with a multi-dimensional Gaussian kernel for recognition of various natural human activities, including AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp and StandUp by building spatial-temporal bag-of-features (space-time grids).

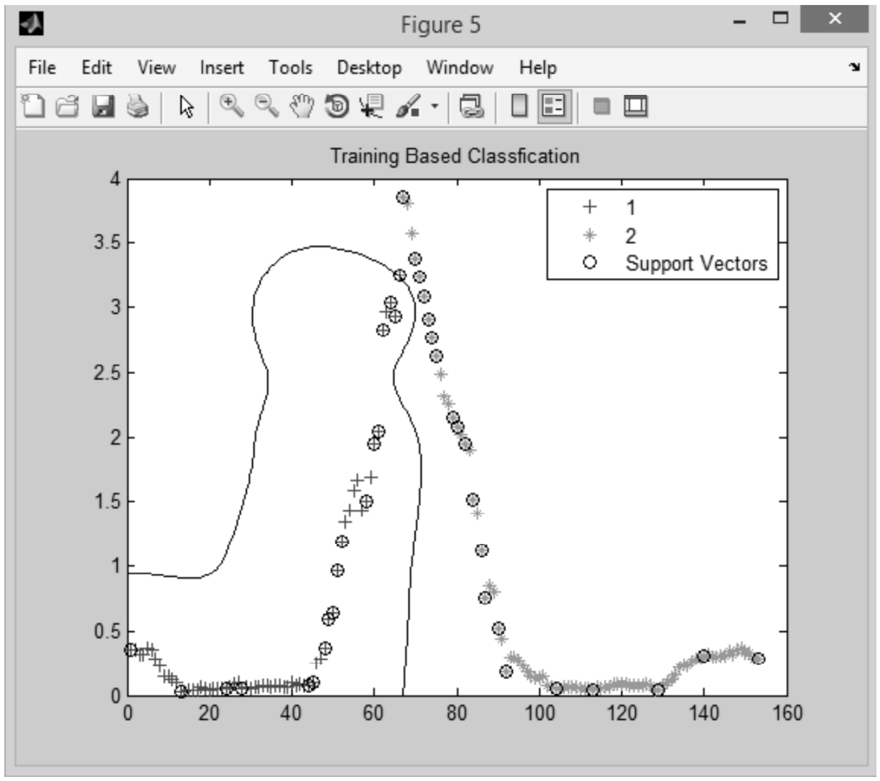

Figure 4: The SVM classification result of the training video

In this research SVM is used to find the dichotomic hyper plane between training and the test data sets. The SVM is used to recognize the human activities by extracting the local space and time features in the input video as shown in the Fig. 4.

V. RESULTS AND DISCUSSIONS

In this research, the human activity detection system focuses towards the security issues. So many researchers have proposed best in class perception systems to help with a part of the security issues with evolving accomplishment. Late studies have suggested that the limit of these observation structures to learn essential natural behavior outlines as wells as to perceive and expect astonishing, or peculiar, practices considering those learnt samples are possible moves up to those systems. In this we have proposed LK Algorithm for optical flow analysis and SVM for classification. The Lucas–Kanade[6] method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point *p* under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at *p*. The training video is taken to test for the anomaly action recognition. Also the class of actions are analyzed and tested whether they are like walking, running or jumping actions.

VI. ACKNOWLEDGMENT

The Authors expresses their sincere thanks to Meenakshi College of Engineering, Chennai for providing necessary facilities to conduct the research work.

REFERENCES

[1] Moeslund, T. B., Hilton, A., and Kruger, V., “A survey of advances in vision-based human motion capture and analysis,” Computer Vision and Image Understanding, vol. 104, pp. 90-126, 2006.

[2] Nevatia, G. R., and Cohen, I., “Event Detection and Analysis from Video Streams” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.23, Aug. 2001.

[3] Baker, S. and Kanade, T. “Limits on super-resolution and how to break them” IEEE Conference on Computer Vision and Pattern Recognition, pp. 372–379, 2010.

[4] Baker, S. and Matthews, I. Equivalence and efficiency of image alignment algorithms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1090–1097,2001.

[5] C. Stauûer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking” Computer Vision and Pattern Recognition, Santa Barbara, CA, June 1998.

[6] K. Kim, T.H. Chalidabhongse, D. Harwood, L. Davis, Real-time foreground-background segmentation using codebook model” Real-Time Imaging 11 (3) pp. 172–185,2005.

[7] P. Fihl, R. Corlin, S. Park, T.B. Moeslund, M.M. Trivedi, “Tracking of individuals in very long video sequences” International Symposium on Visual Computing (ISVC), Stateline, Lake Tahoe, Nevada, USA, Nov 6–8, 2006.

[8] Ricky J. Sethi, Amit K. Roy-Chowdhury, “The Human Action Image Pattern Recognition” ICPR, 20th International Conference, IEEE Conference pp: 3674 - 3678, 2010.

[9] Sotirios Spanogianopoulos, Konstantinos Sirlantzis, Markos Mentzelopoulos, Aristidis Protopsaltis, “Human computer interaction using gestures for mobile devices and serious games: A review” Interactive Mobile Communication Technologies and Learning (IMCL), International Conference, IEEE Conference, Pages: 310 - 314, ,2014.

[10] Stefan Oniga, Jozsef Suto, “Human activity recognition using neural networks Control” ICCC, IEEE Conference Pages: 403 - 406, 2014.

[11] Justin graham, “Target tracking with Lucas kanade optical flow” University of Texas, 2010.

[12] Schuldt, C., Laptev, I., Caputo, B. “Recognizing Human Actions: A Local SVM Approach”, In Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR), Cambridge, UK, Volume 3, pp. 32–36, (23–26 August) 2004.

[13] Laptev and P. Perez. “Retrieving actions in movies” In ICCV, 2007.

[14] Laptev, I, Marszalek, M., Schmid, C., Rozenfeld, B. “Learning Realistic Human Actions from Movies” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, pp. 1–8, (23–28 June) 2008.