Scene Segmentation and Interpretation-
Pattern Recognition
Project Report

on

Human Activity Recognition from Videos

by

Gopikrishna Erabati
Mohit Kumar Ahuja

under the guidance of
Dr. Desire Sidibe

Master in Computer Vision

UNIVERSITE DE BOURGOGNE
Centre Universitaire Condorcet - UB, Le Creusot

10 June 2017

Contents

1. Introduction:

Nowadays, it's a very hot topic on video-based human action detection, which has recently been demonstrated to be very useful in a wide range of applications including video surveillance, tele-monitoring of patients and senior people, medical diagnosis and training, video content analysis and search, and intelligent human computer interaction [1]. As video camera sensors become less expensive, this approach is increasingly attractive since it is low cost and can be adapted to different video scenarios.

Actions can be characterized by spatiotemporal patterns. Similar to the object detection, action detection finds the reoccurrences of such spatiotemporal patterns through pattern matching. Compared with human motion capture, which requires recovering the full pose and motion of the human body, the task of action detection only requires detecting the occurrences of a certain type of actions.

Video features for action detection

The development of video-based action detection technology has been ongoing for decades. The extraction of appropriate features is critical to action detection. Ideally, visual features are able to handle the following challenges for robust action detection:

1. Viewpoint variations of the camera
2. Performing speed variations for different people
3. Different anthropometry of the performers and their movement style variations
4. Cluttered and moving backgrounds.

Previously, human bodies were tracked and segmented from the videos to characterize actions and motion trajectories are popularly used to represent and recognize actions. Unfortunately, only limited success has been achieved because robust object tracking is itself a nontrivial task. Recently, interest point based video features show promising results in the action detection research. Such interest point-based video features do not require foreground/background separation or human tracking [2].

We searched a lot and found many techniques to identify actions in videos, like space-time interest point (STIP), which is developed by Laptev and Lindeberg. STIP features have been frequently used for action recognition. However, the detected interest points are usually quite sparse, and it is time consuming to extract STIP features for high-resolution videos. And then we finalized to work on few types of interest-point based feature extractions like;

1. The first type of interest point features is called 3-D SIFT, developed by Scovanner et al [3]. This descriptor is similar to scale invariant feature transformation (SIFT) descriptor except that the gradient direction for each pixel is a three-dimensional vector. It can work with any interest point detector.
2. The second type of interest point features is named spatiotemporal interest point (STIP) [2].
3. The third type of classification is done by using Histograms of Oriented Optical Flow (HOOF), Histogram of Oriented Optical Flow (HOOF) features are independent of the scale of the moving person as well as the direction of motion. Extraction of HOOF features does not require any prior human segmentation or background subtraction.

However, HOOF features are non-Euclidean, and thus the evolution of HOOF features creates a trajectory on a nonlinear manifold.

We found these three techniques for feature extraction as the most powerful techniques, why we choose these three algorithms and which one is the best among all of them will be explained one by one in the detailed explanation of the report. We implemented our algorithms on KTH dataset which has six actions like boxing, hand waving, hand clapping, jogging, running and walking of nearly 100 videos each.

2. Methodologies

2.1 3-D SIFT:

Action recognition is a well-studied yet very difficult problem in the task of automatically understanding video data.Intra-class variation is often very large and confusion is common between actions such as running and jogging. Actions depicted by video data inherently contain spatiotemporal information, which implies that descriptors are needed which can robustly encode this kind of information.

Earlier methods to 3D Sift tested only simple features such as gradient magnitude. However, note that these features do not explicitly describethe true spatiotemporalnature of the video data. The 3D SIFT descriptor encodes the information local in both space and time in a manner which allows for robustness to orientations and noise. In addition, after describing the videos as a bag of spatio temporal words using the proposed SIFT descriptor, we discover relationships between words to form spatiotemporalword groupings.

This 3D SIFT descriptor is able to robustly describe the 3D nature of the data in a way that vectorization of a 3D volume cannot. Using sub-histograms to encode local time and space information allows 3D SIFT to better generalize the spatiotemporal information than features used in previous works. All of this translates into a fast and accurate method of action recognition.

The first step is to compute the overall orientation of the neighborhood. Once this is computed we can create the sub-histograms which will encode our 3D SIFT descriptor.

2.1.1 Orientation:

The 2D gradient magnitude and orientation[3] for each pixel is defined as follows:

$$m_{2D}(x,y) = \sqrt{L_x^2 + L_y^2}, \quad \theta(x,y) = \tan^{-1}(\frac{L_y}{L_x}).$$

Where Lx and Ly are respectively computed using finite difference approximations: L(x + 1, y, t) and L(x -1, y, t) and L(x, y+1, t) - L(x, y – 1, t). Similarly, in 3D (x, y and t), the spatiotemporal gradient (Lx, L – y, L - t) can be computed, where Lt is approximated by L(x, y, t + 1) - L(x, y, t - 1).Now the gradient magnitude and orientations in 3D are given:

$$m_{3D}(x,y,t) = \sqrt{L_x^2 + L_y^2 + L_t^2},$$
$$\theta(x,y,t) = \tan^{-1}(L_y/L_x),$$
$$\phi(x,y,t) = \tan^{-1}(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}).$$

It can be observed that ∅ now encodes the angle away from the 2D gradient direction. Due to the fact that $\sqrt{L_x^2 + L_y^2}$is positive, ∅ will always be in the range (-$\pi/2$,$\pi/2$). This is a desired effect, causing every angle to be represented by a single unique (θ,∅) pair.

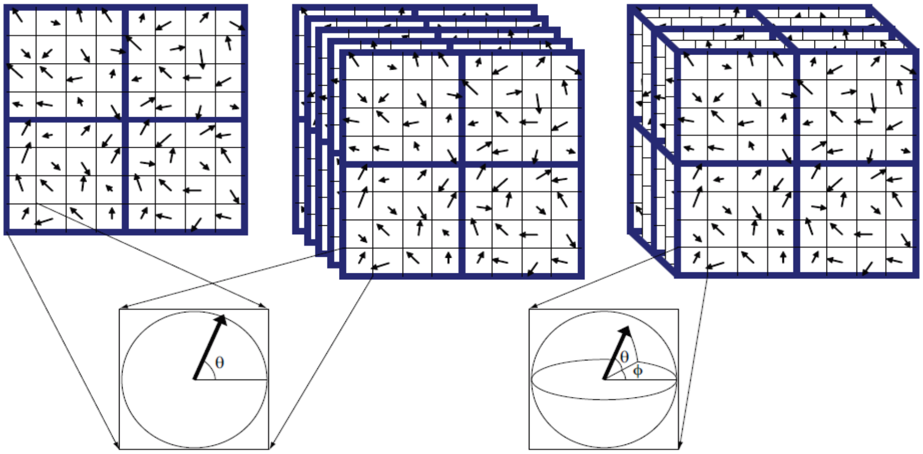

Figure 1: 2D SIFT and 3D SIFT

In Figure 1, the left image shows the familiar 2D SIFT descriptor. The center shows how multiple 2D SIFT descriptors could be used on a video without modification to the original method. The right shows the 3D SIFT descriptor with its 3D sub-volumes, each sub-volume is accumulated into its own sub-histogram. These histograms are what makes up the final descriptor.

2.1.2 Action Classification

The first step is to select the salient regions from the spatiotemporal video cube. For this purpose we carry out random sampling of a video at different locations, times, and scales. The interest points could also be extracted from video content using other methods. However, these methods require additional processing stages which can be hectic. Once the points are sampled the second step is to describe the spatiotemporal region around the points using the proposed 3D SIFT descriptor. The length of the descriptor is based on the number of sub-histograms, and the number of bins used to break represent the θ and ∅ angles. We observed slight improvements when using the larger feature vectors and the results of this technique uses the larger descriptor, however the abbreviated feature descriptor could be used to improve runtime test speeds. The descriptors gathered from all the interest points are then quantized by clustering them into a pre-specified number of clusters.

Now that our vocabulary is computed, the 3D SIFT descriptors from the videos are matched to each `word' and the frequency of the words in each video is accumulated into a histogram. This word frequency histogram, referred to as a `signature', is used to generate an initial representation of the video. Finally, SVM learning is used to train using grouping histograms as feature vectors. And prediction is done on this model and test video whose 3D Sift descriptors are computed same as train data.

2.2 Optical Flow and Histograms of Oriented Optical Flow(HOOF):

2.2.1 Optical Flow

Optical flow is the apparent 2D image motion of pixels. The main initial assumption of optical flow is 'brightness constancy assumption', where intensity of pixels in small variations in 'x', 'y' and 't' directions be the same to original pixel.

The brightness constancy equation is given by,

$$I(x,y,t) = I(x+dx, y+dy, t+dt)$$

When we apply Taylor series approximation on above equation, it turns out to be,

$$I_x u + I_y v + I_t = 0$$

where, I_x is derivative of image in x-direction, I_y is derivative of image in y-direction , I_t is derivative of image in time (between frames) and 'u' is velocity in x-direction and 'v' is velocity in y-direction.

In the above equations there are 2 unknowns (u and v) but there is only one equation, so its an underdetermined system.

We can solve this by estimation of flow velocity using

1. Horn and Shunck (Global method) [4]

2. Lucas-Kanade (local Method) [5]

We used Lucas-Kanade as it provides better results than H&S because of local smoothing. The main idea of Lucas-Kanade is the optical flow is constant on the neighborhood of the current point (x,y). Each neighbor gives one equation. So , if we take a 5 x 5 neighborhood around a pixel we would get 25 equations in 2 unknowns which is over determined system which can be solved by SVD.

But Lucas-Kanade fails in case of large motion, so we go for multi-resolution Lucas-Kanade to determine optical flow. The multi resolution Lucas-Kanade is as depicted in Fig. 2.

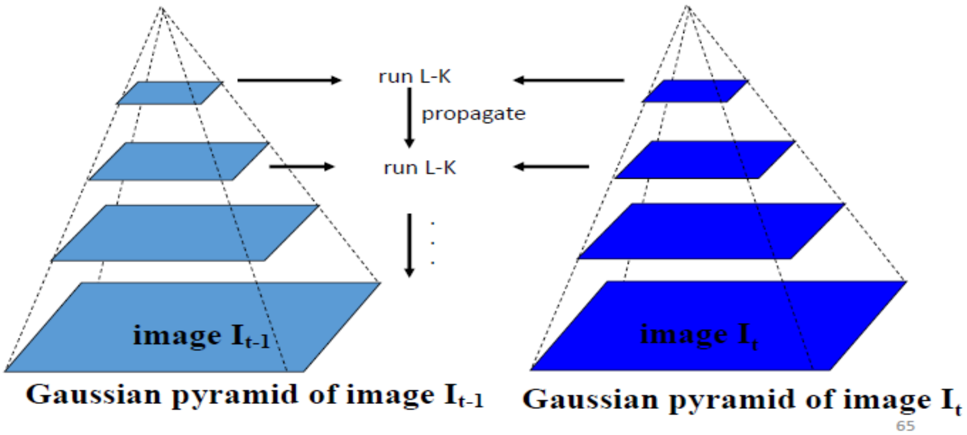
run L-K
propagate
run L-K
image I_{t-1}
image I_t
Gaussian pyramid of image I_{t-1}
Gaussian pyramid of image I_t
65

Fig. 2 Multi resolution Lucas-Kanade

But we cannot use optical flow directly as a feature because it is susceptible to background noise, scale and directionality. So we go for distribution of optical flow by using histograms.

2.2.2 Histogram of Oriented Optical Flow

Much of the success of recent object recognition techniques relies on the use of more complex feature descriptors, such as SIFT descriptors or HOG descriptors, which are essentially histograms. Since histograms live in a non-Euclidean space, we can no longer model their temporal evolution with LDSs, nor can we classify them using a metric for LDSs. So, we used each frame of a video using a histogram of oriented optical flow (HOOF) and to recognize human actions by classifying HOOF time-series [6].

Global approaches use global features such as optical flow to represent the state of motion in the whole frame at a time instant. With static background, one can represent the type of motion of the foreground object by computing features from the optical flow. Example is shown in figure 3 for optical flow and HOOF.

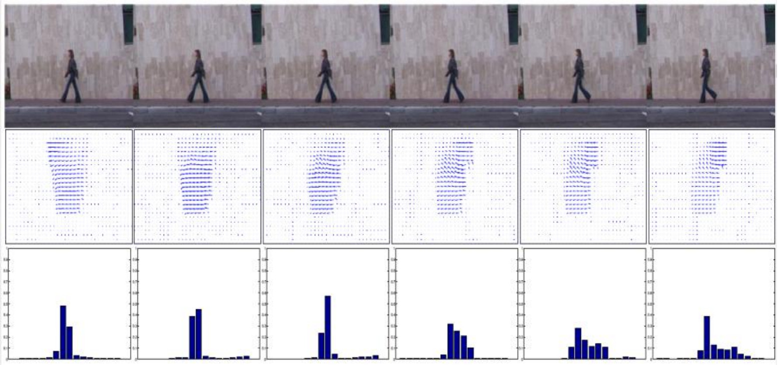

Figure 3 Optical Flow and HOOF

Because of many successful results of histograms of features in the object recognition community, we posit that the natural feature to use in a motion sequence is optical flow. However, the raw optical flow data may be of no use, as the number of pixels in a person (hence the size of the descriptor) changes over time. Moreover, optical flow computations are very susceptible to background noise, scale changes as well as directionality of movement. So, to avoid these issues, we use instead the distribution of optical flow, when a person moves through a scene with a stationary background, it induces a very characteristic optical flow profile as shown in figure 3.

However, it was noticeable that the observed optical flow profile could be different if the activity was performed at a larger scale. For example a zoomed in walking person versus a far-away walking person. The magnitude of the optical flow vectors would be larger in the zoomed in case. Similarly, if a person is running from the left to the right, the optical flow observed would be a reflection in the vertical axis to that observed if the person was running from the right to the left. We thus need a feature based on optical flow that represents the action profile at every time instant and that is invariant to the scale and directionality of motion.

To overcome these issues, we propose the Histogram of Oriented Optical Flow (HOOF), which is defined as follows. First, optical flow is computed at every frame of the video. Each flow vector is binned according to its primary angle from the horizontal axis and weighted according to its magnitude. Thus, all optical flow vectors,v $= [\mathrm{x}, \mathrm{y}]^T$with direction, $\theta = \tan^{-1}(\frac{y}{x})$ in the range:

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B}$$

And will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin b, $1 \leq \mathrm{b} \leq \mathrm{B}$, out of a total of B bins. Finally, the histogram is normalized to sum up to 1.

Binning according to the primary angle, the smallest signed angle between the horizontal axis and the vector, allows the histogram representation to be independent of the (left or right) direction of motion. Normalization makes the histogram representation scale-invariant. We expect to observe the same histogram whether a person is moving from the left to the right or in the opposite direction, whether a person is running far away in the scene or very near the camera. Since the contribution of each optical flow vector to its corresponding bin is proportional to its magnitude, small noisy optical flow measurements have little effect on the observed histogram. Assuming a stationary background, there is no optical flow in the background. Using the magnitude-based addition to each bin, we can simply compute the optical flow histogram on the whole frame rather than requiring to pre-compute a segmentation of the moving person.

2.3 Space-time Interest Points

Interest points provide compact and abstract representations of patterns in an image. So, to extend the notion of spatial interest points into the spatiotemporal domain and show how the resulting features often reflect interesting events that can beused for a compact representation of video data as well asfor its interpretation.

To detect spatiotemporal events, we build on the idea of the Harris and Forstner interest point operators and detect local structures in space-time where the image values have significant local variations in both space and time. We then estimate the spatiotemporal extents of the detected events and compute their scale-invariant spatiotemporal descriptors. Using such descriptors, we classify events and construct video representation in terms of labeled space-time points.

So, till now we implemented approaches for motion analysis mainly involve the computation of optic flow or feature tracking. Although very effective for many tasks, both of these techniques have limitations. Optic flow approaches mostly capture first-order motion and often fail when the motion has sudden changes. Feature trackers often assume a constant appearance of image patches over time and may hence fail when this appearance changes, for example, in situations when two objects in the image merge or split.

An example of Result of detecting the strongest spatiotemporal interest point is shown in figure 4, where a football sequence with a player heading the ball. The detected event corresponds to the high spatiotemporal variation of the image data or a “space-time corner” as illustrated by the spatiotemporal slice on the right [2].

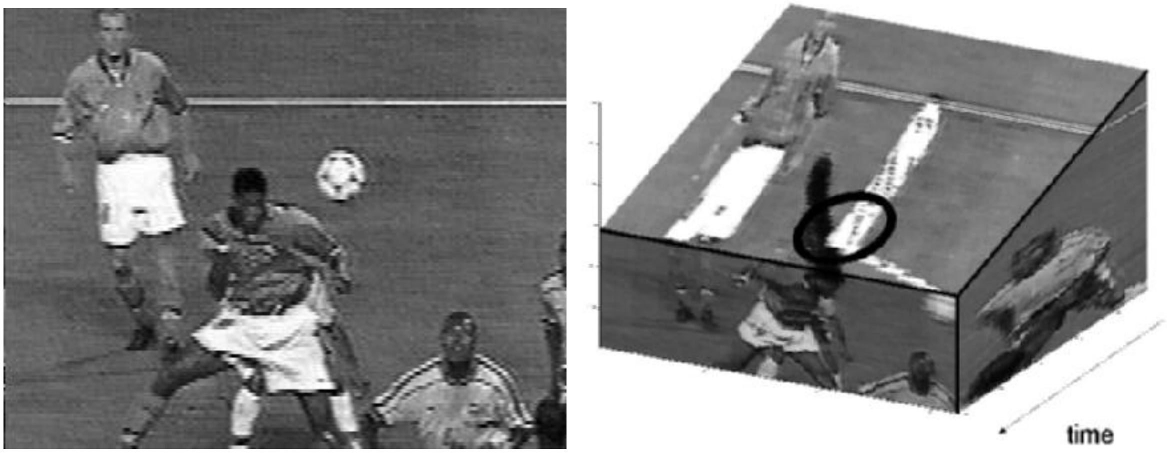

Figure 4: Detecting the strongest spatiotemporal interest point

To capture events with different spatiotemporal extents, we compute interest points in spatiotemporal scale-spaceand select scales that roughly correspond to the size of the detected events in space and to their durations in time.

2.3.1 Interest point detection

The idea of the Harris interest point detector is to detect locations in a spatial image *fsp* where the image values have significant variations in both directions. For a given scale of observation σ_l^2, such interest points can be found from a windowed second moment matrix integrated at scale:

$$\sigma_i^2 = s\sigma_l^2$$

$$\mu^{sp} = g^{sp}(\cdot;\, \sigma_i^2) * \begin{pmatrix} (L_x^{sp})^2 & L_x^{sp} L_y^{sp} \\ L_x^{sp} L_y^{sp} & (L_y^{sp})^2 \end{pmatrix}$$

Where L_x^{sp} and L_y^{sp} are Gaussian derivatives defined as:

$$L_x^{sp}(\cdot;\, \sigma_l^2) = \partial_x (g^{sp}(\cdot;\, \sigma_l^2) * f^{sp})$$
$$L_y^{sp}(\cdot;\, \sigma_l^2) = \partial_y (g^{sp}(\cdot;\, \sigma_l^2) * f^{sp}),$$

And where g_{sp} is the spatial Gaussian kernel:

$$g^{sp}(x, y;\, \sigma^2) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/2\sigma^2)$$

As the eigenvalues $\lambda 1$, $\lambda 2$, ($\lambda 1 \leq \lambda 2$) ofμ^{sp}represent characteristic variations of *fsp*in both image directions, two significant values of $\lambda 1$, $\lambda 2$ indicate the presence of an interest point. To detect such points, Harris and Stephens propose to detect positive maxima of the corner function

$$H^{sp} = \det(\mu^{sp}) - k\, \mathrm{trace}^2(\mu^{sp}) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2.$$

2.3.2 Interest point detection - Interest points in the space-time

The idea of interest points in the spatial domain can be extended into the spatiotemporal domain by requiring the image values in space-time to have large variations in both the spatial and the temporal dimensions. Points with such properties will be spatial interest points with a distinct location in time corresponding to the moments with non-constant motion of the image in a local spatiotemporal neighborhood.

To model a spatiotemporal image sequence, we use a function f , R2×R → R and construct its linear scale-space representation L: R2 × R ×R_+^2→ R by convolution of f with an anisotropic Gaussian kernel1 with distinct spatial variance $\sigma_l^2$and temporal variance τ_l^2.

$$L(\cdot;\, \sigma_l^2, \tau_l^2) = g(\cdot;\, \sigma_l^2, \tau_l^2) * f(\cdot),$$

Where the spatiotemporal separable Gaussian kernel is defined as:

$$g(x, y, t;\, \sigma_l^2, \tau_l^2) = \frac{\exp(-(x^2 + y^2)/2\sigma_l^2 - t^2/2\tau_l^2)}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}}$$

Similar to the spatial domain, we consider the spatiotemporal second-moment matrix which is a 3-by-3 matrix composed of first order spatial and temporal derivatives averaged with a Gaussian weighting function $g(\cdot, \sigma_l^2 , \tau_i^2)$

$$\mu = g(\cdot;\, \sigma_i^2, \tau_i^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$$

Where the integration scales are $\sigma_i^2 = s\, \sigma_l^2and\tau_i^2 = S\tau_l^2$, while the first-order derivatives are defined as $L\xi(\cdot, \sigma_l^2, \tau_l^2) = \partial\xi(g * f)$. The second-moment matrix μ has been used previously by Nagel and Gehrkein the context of optic flow computation.To detect interest points, we search for regions in *f*having significant eigenvalues $\lambda1$, $\lambda2$, $\lambda3$ of μ. Among different approaches to find such regions, we choose to extend the Harris corner function defined for the spatial domain into the spatiotemporal domain by combining the determinant and the trace of μ in the following way

$$H = \det(\mu) - k\,\mathrm{trace}^3(\mu) = \lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3$$

For sufficiently large values of *k*, positive local maxima of *H* correspond to points with high variation of the image gray-values in both the spatial and the temporal dimensions. Thus, spatiotemporal interest points of *f* can be found by detecting local positive spatiotemporal maxima in *H*.

2.3.3 Histogram of oriented Gradients (HOG)

We used HOG as feature descriptor to spatial temporal interest points. The HOG features can be detected similar to HOOF features except that in HOOF the angle will be calculated between velocities in x and y directions, instead in HOG it will be calculated between gradients along x and y-directions.

3. Implementation in MATLAB

Note : The code for each methodology is in each folder on the name of method. The code is well commented for further reference. There is a README.txt file in the code folder which explains how to run the code and information regarding files.

3.1 STIP

The videos to be trained are selected for different actions. Each video is taken at a time and is divided into frames. We shall take certain number of frames and compute STIP for each frame.

This is implemented by,

[featPos, featVal] = computeSTIP(frame, kParameter, sigma, sSigma, nPoints)

After we get the interest points we compute HOG features for it and form a feature vector.

The train data is then saved to 'FeatureTrainSTIP' folder which can be used later.

The computing of STIP features for training data is implemented by,

trainData( ) in STIP folder.

After we get train features, we can test a video by computing same features on test video and giving the train data to SVM (linear kernel) to build a model and then predict the test video by this model.

This is implemented by,

testData( ) in STIP folder.

3.2 Optical Flow

For train features, we take the train videos and divide into frames. We take pair of frames in a video and compute optical flow between them and give this optical flow to HOOF to get the descriptor. This is done for all frames and all test videos.

This is implemented by,

trainDataHOOF( ) in Optical Flow folder.

We can similarly test the video as said in 3.1.

3.3 3D Sift

The same method as above applies here, we first detect interest points using Harrris detctor in each frame and compute sift descriptor for all videos. We then form bag of words using k-means clustering. And then we form word frequency histogram called 'signature'.

This is implemented by,

trainData3DSift( ) in 3DSift folder

The same would be applied on test video to get features and we use SVM to classify the test video.

3.4 Calculating Accuracy and Confusion Matrix

For each of the above three methods the accuracy of test video set and consfusion matrix is computed.

This is implemented by,

getAccuracySTIP( )

getACcuracyHOOF( )

getAcuracy3DSift( )

4. About Dataset : KTH

We used KTH dataset which contain six actions : boxing, hand waving, hand clapping, jogging, running, walking of 25 persons with four different types of video views.

The dataset example of a frame is as shown in fig. 5.

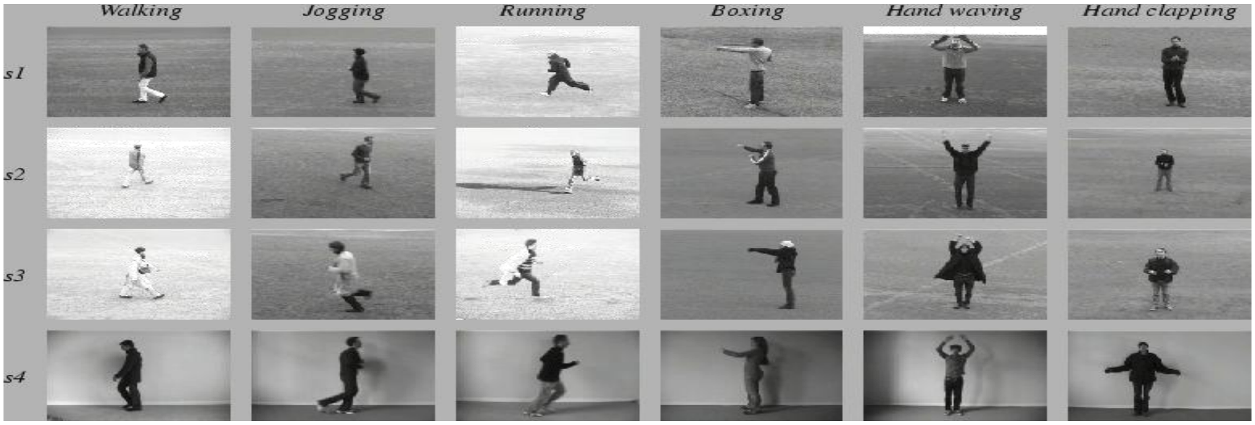

Fig. 5 KTH Dataset

In the above fig.5 , s1 is for normal video of person, s2 is for scale variation , s3 is for using different clothes and s4 is for different illumination conditions.

5. Results:

5.1 3D Sift

In 3D Sift implementation, we used 54 training videos of six actions and 30 test videos of six actions from KTH dataset.

The accuracy achieved was 40%.

The confusion matrix is as given

TABLE 1 Confusion matrix of 3D Sift result

	Boxing	Hand clapping	Hand waving	Jogging	Running	Walking
Boxing	60 %		20 %	20 %		
Hand Clapping			80 %			20 %
Hand Waving		20 %	40 %	40%		
Jogging				40 %	20 %	40 %
Running					80 %	20%
Walking				40 %		60 %

The first column represented ground truth class and first row represent predicted class.

5.2 Optical flow + HOG

In OF + HOOF implementation, we used 126 training videos of six actions and 54 test videos of six actions from KTH dataset.

The accuracy achieved was 60%.

The confusion matrix is as given in Tab. 2.

TABLE 2 Confusion matrix of OF + HOOF result

	Boxing	Hand clapping	Hand waving	Jogging	Running	Walking
Boxing	50 %	9 %	30 %			10 %
Hand Clapping	30 %	29 %	30 %			11 %
Hand Waving	50 %		50 %			
Jogging	8 %	8 %	84 %			
Running	8 %	10 %			62 %	10 %
Walking	8 %	8%				84 %

5.3 STIP + HOG

In STIP + HOG implementation, we used 126 training videos of six actions and 54 test videos of six actions from KTH dataset.

The accuracy achieved was 48%.

The confusion matrix is as given in Tab.3.

TABLE 3 Confusion matrix of STIP + HOG result

	Boxing	Hand clapping	Hand waving	Jogging	Running	Walking
Boxing	56 %	22 %	11 %		11 %	
Hand Clapping		56 %	33 %		11 %	
Hand Waving	11 %	22 %	67 %			
Jogging	22 %	11 %	11 %	12 %	33 %	11 %
Running			11 %		89%	
Walking		11 %	33 %	11 %	33 %	12 %

6. Conclusion

We can infer from our results that the feature selection plays a vital role in detecting the actions in video. We used three features to get the videos detected, 3D Sift, Optical flow + HOOF and STIP + HOG. Among the three methods we got higher accuracy for Optical Flow method of 60%. The accuracy of STIP + HOG is around 48% and for 3D Sift it is 40 %. Sift accuracy can be improved by taking more number of descriptors and more number of frames. In our case Sift takes lot of time to get the result, which can be resolved using parallel computing.

References

[1] Junsong Yuan and Zicheng Liu"TechWare, "Video-Based Human Action Detection Resources", IEEE Signal Processing Magazine, Sept. 2010.

[2] Ivan Laptev and Tony Lindeberg, "Space-time Interest Points", 9th IEEE Intnl. Conf. in Computer Vision (ICCV), 2003.

[3] Paul Scovanner, Saad Ali and Mubarak Shah, "A 3-Dimensional SIFT Descriptor and its Application to Action Recognition", *MM'07,* September 23–28, 2007, Augsburg, Bavaria, Germany.

[4] Horn B.K.P and Shunck B.G., "Determining optical flow", Artificial Intelligence. Vol 17 pp 185-203

[5] Lucas, B. and Kanade, T., "An iterative image registration technique with an application to stereo vision", In *Proc. of the Int. Joint Conf. on Artificial Intelligence.*

[6] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager and Ren´e Vidal, "Histograms of Oriented Optical Flow and Binet-Cauchy Kernels on Nonlinear Dynamical Systems for the Recognition of Human Actions".