International Journal of Innovative Computing, Information and Control
Volume **12**, Number **1**, February **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1–13

SPATIOTEMPORAL LBP AND SHAPE FEATURE FOR HUMAN ACTIVITY REPRESENTATION AND RECOGNITION

Sk. Md. Masudul Ahsan, Joo Kooi Tan, Hyoungseop Kim
and Seiji Ishikawa

Department of Control Engineering
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata-ku, Kitakyushu-shi, Fukuoka 804-8550, Japan
{ ahsan; etheltan; ishikawa }@ss10.cntl.kyutech.ac.jp; kim@cntl.kyutech.ac.jp

Received June 2015; revised November 2015

Abstract. *In this paper, we propose a histogram based feature to represent and recognize human action in video sequences. Motion History Image (MHI) merges a video sequence into a single image. However, in this method, we use Directional Motion History Image (DMHI) to create four directional spatiotemporal templates. We, then, extract the Local Binary Pattern (LBP) from those templates. Then, spatiotemporal LBP histograms are formed to represent the distribution of those patterns which makes the feature vector. We also use shape feature taken from three selective snippets and concatenate them with the LBP histograms. We measure the performance of the proposed representation method along with some variants of it by experimenting on the Weizmann action dataset. Higher recognition rates found in the experiment suggest that, compared to complex representation, the proposed simple and compact representation can achieve robust recognition of human activity for practical use.*

Keywords: Human action, DMHI, LBP, Histogram, Support vector machine

1. Introduction.

Automatic analysis of a human motion and recognizing the performed action from video is one of the attractive but challenging problems in computer vision research. In recent years, researches on human action recognition have gained more interests due to its diverse applications like robotics, monitoring normal or suspicious activity by video surveillance, controller-free human computer interaction or gaming, mixed or virtual reality, intelligent environments. Each application domain has its individual demands, but in general, algorithms must be smart enough to detect and recognize various actions performed by different people with several possible body configurations such as view angle, clothing, speed or posture variation. Moreover, the designed algorithms must have the capability of real time recognition as well as the adaptability to various types of environments [1].

The rest of the paper is organized as follows. Some state of the art research works are delineated in Section 2. Section 3 describes the proposed method of how an action can be represented as a histogram of spatiotemporal LBP templates. Experimental results and discussion are presented in Section 4 followed by a conclusion in Section 5.

2. Related Works.

Recently, there have been a lot of works on analyzing human motions in a spatiotemporal way instead of analyzing each individual frame [1]. In the literature, several methods [2] have been proposed for learning and recognizing a broad class of motion or action patterns. Bobick and Davis [3] used Motion Energy Image (MEI) and Motion History Image (MHI) for the first time as temporal templates to represent human actions. Recognition was done by using seven Hu moments. They have developed a virtual aerobics trainer that can watch and respond to the user as he/she performs the workout. Weinland et al. proposed the 3D extension of the temporal templates [4]. They used multiple cameras to build motion history volumes and action classification was performed using Fourier analysis in the cylindrical coordinates. Related 3D approaches have been introduced by Blank et al. [5], and Yilmaz and Shah [6] who used time as the third dimension to form space-time volumes in the (x, y, t) space. Those volumes were matched using features from Poisson equations and geometric surface properties, respectively.

Kellokumpu et al. [1] extracted a histogram of Local Binary Pattern (LBP) from MHI and MEI as temporal templates to represent action. They also used another descriptor called LBP-TOP [7], which extracts LBP information from three orthogonal planes (xy, xt, and yt). They used Hidden Markov Model (HMM) to model the temporal behavior of action and hence to recognize them. Yau et al. [8] used MHI to represent the temporal information of the mouth, which is generated using accumulative frame differencing of the video. The MHIs are decomposed into wavelet sub-images using Discrete Stationary Wavelet Transform (DSWT). Artificial Neural Network (ANN) with back propagation learning algorithm is used for classification.

Huang et al. [9] represented a human action as a Histogram of Oriented Gradient (HOG) of MHI. First, they generated MHI by differential images from successive frames of a video, then HOG features are computed and supplied to a Support Vector Machine (SVM) for action classification.

In the basic MHI method, old motion information can be wiped out by new motion information that occurred in a same region. This overwriting causes poor recognition rate for natural motions that have complex nature and overlapping motion (e.g., sitting down and then standing up). Ahad et al. [10] employed a variant of MHI called Directional MHI (DMHI) to represent that type of actions. They also used Hu moments for the recognition purpose.

A histogram is a popular statistic that has been frequently used in computer vision research. For action recognition, Freeman and Roth [11] used orientation histograms for hand gesture recognition. Recently, Dalal and Triggs used Histograms of Oriented Gradients (HOGs) for human detection in images [12], which is shown to be quite successful.

Ikizler and Duygulu [13] used a Histogram-of-Oriented-Rectangles (HOR) for representing human actions. They represent each human pose in a frame by oriented rectangular patches, and form a histogram to represent the distribution of these patches. They used different classifiers like nearest neighbor, support vector machine, dynamic time warping for the matching purpose.

The past approaches of action descriptor can be roughly classified into two groups: first one that extracts a global feature descriptor from a video sequence [14-16], and assign a single label to the entire video. The other method extracts feature descriptor for each frame and assigns an action label to them [17-19]. However, if required, a local label for the first approach can be obtained by extracting feature set until the desired frame to be labeled. Similarly, a global label for the second approach is usually obtained by simple voting methods.

Some methods such as [3,10] did not use any benchmark dataset to show the efficiency of their descriptors. Some other methods are not suitable for real-time recognition such as the HOR descriptor presented by Ikizler and Duygulu [13] which takes approximately one second per frame only for the rectangle extraction phase. Since Kellokumpu et al. used a volume based descriptor [1], it is also inexpedient for online recognition applications. In their descriptors, they have to wait for the next few frames to arrive to extract the LBP-TOP of a particular frame.

The key motive of this study is to devise a simple and compact action representation scheme that can be applied to real-time recognition problems. To realize the objective, our action descriptor is based on a basic idea that a human action descriptor can be represented as a distribution of local texture patterns extracted from a spatiotemporal template. To fulfil the purpose, rather than analyzing every frame or detecting the exact body parts, we are only interested in the distribution of those spatiotemporal patterns. In this paper, we propose a novel way of constructing a spatiotemporal template, i.e., we use DMHI to infuse a sequence of frames into a single temporal template and then apply the LBP operator which highlights the spatial textures unlike [1]. We form a concatenated block histogram of those LBPs that serve as a feature vector. However, we also use a shape feature descriptor for some selected frames of the action sequence in the form of a histogram.

3. The Proposed Method.

3.1. Foreground extraction. The sample videos in the action dataset we use are taken by a stationary camera. So, without constructing any complex background model, we simply take the absolute difference of the current frame with the static background frame and Otsu threshold [20] is performed to get the binary foreground mask. Before this all the frames are passed through a Gaussian filter to reduce the effect of noise. Figure 1 shows the simple graphical illustration of the foreground extraction method. However, performing the subtraction of background in grayscale or RGB color space loses some foreground information for the action dataset in hand, so we do all the processing in Lab color space, and Equation (1) to Equation (6) show the details of the method, mathematically.

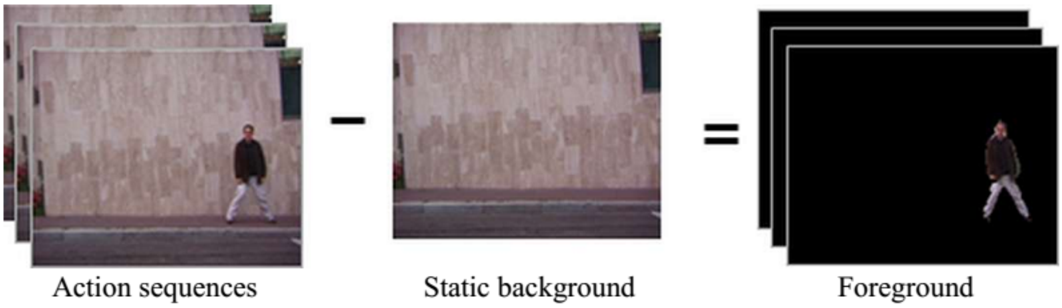


Figure 1. Simple illustration of foreground extraction

The meanings of the used abbreviations are as follows: Df_t is the absolute difference of the current frame, f_t, and the background frame, f_{bg}; L, a, b are the pixel intensity of Df in Lab color space; $\widehat{m}_{fg_t}$ is the single channel intermediate frame where nonzero intensity represents the foreground and, after applying the Otsu threshold, we get m_{fg_t} which is a binary frame containing the foreground mask of size $h_t \times w_t$. The suffix t means the frame at time t, and (x, y) is the spatial position of a pixel in a frame. We use this m_{fg_t} to get the actual foreground frame, f_{fg_t}. The constants in Equation (2) and Equation (3) are experimentally determined.

$$Df_t(x,y) = |f_t(x,y) - f_{bg}(x,y)| \tag{1}$$

$$a_t(x,y) = 0.25L_t(x,y) + 0.6a_t(x,y) + 0.15b_t(x,y) \tag{2}$$

$$b_t(x,y) = 0.2L_t(x,y) + 0.2a_t(x,y) + 0.6b_t(x,y) \tag{3}$$

$$\widehat{m}_{fg_t}(x,y) = \sqrt{L_t^2(x,y) + a_t^2(x,y) + b_t^2(x,y)} \tag{4}$$

$$m_{fg_t} = OtsuThreshold\,(\widehat{m}_{fg_t}) \tag{5}$$

$$f_{fg_t}(x,y) = \begin{cases} f_t(x,y) & \text{if } m_{fg_t}(x,y) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.2. **Spatiotemporal template.** We have used DMHI as a spatiotemporal template. The DMHI [10,21], is an extension of the basic MHI method which splits the motion into four different directions. The MHI $H_\tau(x,y,t)$ can be computed from the following equations using an update function [3]:

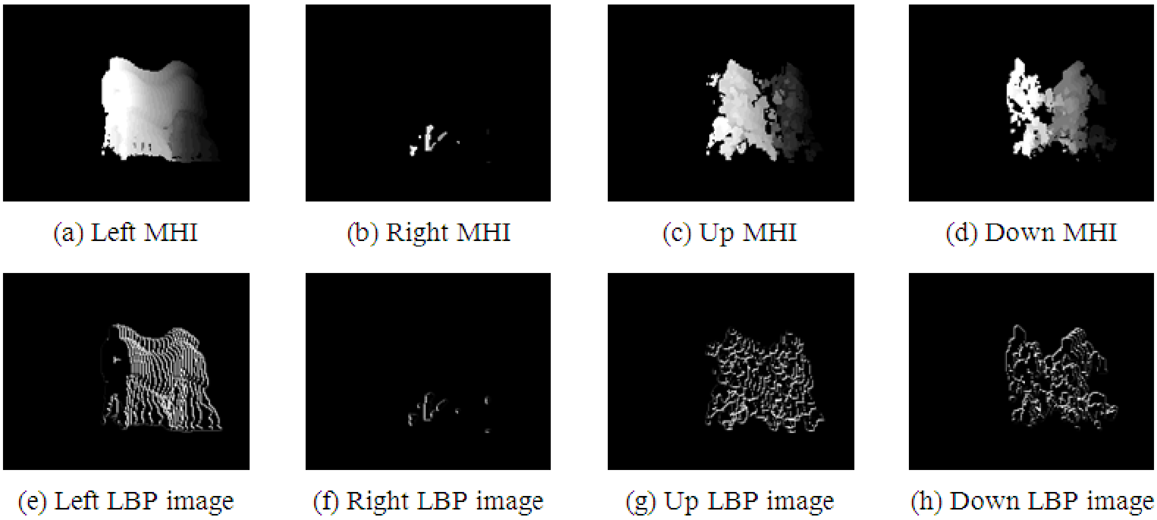
$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } \Psi(x,y,t) = 1 \\ \max(0, H_\tau(x,y,t-1) - \delta) & \text{otherwise} \end{cases} \tag{7}$$

$$\Psi(x,y,t) = \begin{cases} 1 & \text{if } D(x,y,t) \geq threshold \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$D(x,y,t) = |f_t(x,y) - f_{t\pm\Delta}(x,y)| \tag{9}$$

Here, x, y, and t show the position and time; the update function $\Psi(x,y,t)$ signals the presence of motion in the current frame $f(x,y)$; τ decides the temporal duration of MHI, e.g., in terms of the number of frames or in terms of time (second/millisecond); and δ is the decay parameter whose value was 1 in the original MHI [3], and $D(x,y,t)$ gives the absolute difference between pixels with step time difference Δ. This update function is called for every new video frame analyzed in the sequence. The result of this computation is a grayscale image where brighter pixels represent the more recent motion.

To create DMHI, initially the moving region is tracked using a dense optical flow algorithm that generally produces a vector denoting the horizontal (x-direction) and vertical (y-direction) motion of an object. Each of these horizontal and vertical motions are further rectified to positive and negative directions, resulting in four update functions denoting directions right, left, up, and down. These update functions are used in Equation (7) to generate directional MHIs. First row of the Figure 2 presents some example DMHIs of a sidewalk action. Since the motion in the performed action is mostly in leftward direction, we can see that the Left MHI (Figure 2(a)) encapsulates that information, and the right MHI (Figure 2(b)) contains almost nothing. Bouncing upward and downward motions are captured by the Up MHI (Figure 2(c)) and Down MHI (Figure 2(d)).


(a) Left MHI (b) Right MHI (c) Up MHI (d) Down MHI

(e) Left LBP image (f) Right LBP image (g) Up LBP image (h) Down LBP image

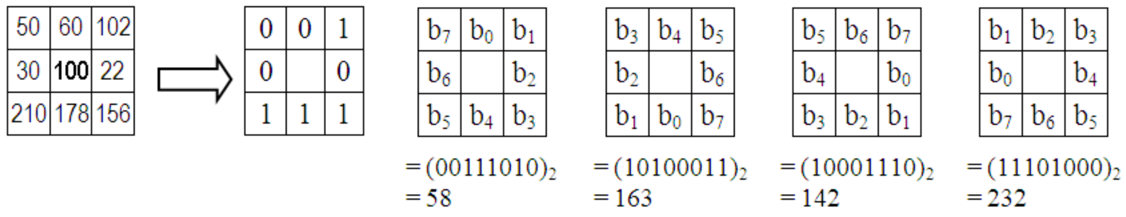
FIGURE 2. Example of DMHIs and corresponding LBP images of a side walk action

3.3. **Generating LBP images.** Using the LBP operator becomes a popular approach in various applications for its computational simplicity. LBP operator [22] describes the local texture pattern of an image with a binary code, which can be obtained by taking a threshold of neighboring pixels with the gray value of their center pixel. Mathematically LBP operator can be written as Equation (10), and Equation (11).

$$LBP(g_c) = \sum_{i=0}^{p-1} B(g_i - g_c) \times 2^i \tag{10}$$

$$B(x) = \begin{cases} 1 & \text{if } x \geq threshold \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Here, g_c is the intensity of the center pixel (x, y) and g_i $(i = 0, 1, \ldots, p-1)$ are the intensities of the neighboring pixels. The neighborhoods can be of different sizes such as 3 × 3, 3 × 5, or 5 × 5 pixel. In this paper, we use 3 × 3 neighborhood, but different arrangement of LBP bit position for different DMHIs, called as rotated bit arrangements. This is to give more strength to the pattern of a particular direction. Figure 3(a) shows a basic LBP operator, whereas Figures 3(b)-3(e) show the LBP bit arrangements used for different DMHIs. Consider Figure 3(b): arrangement of bit positions is chosen in such a way that it will give more emphasis on leftward motion. Other arrangements are chosen to have similar effects. From Figure 3 we can see that same binary output (Figure 3(a)) of the LBP operator can be assigned to a different decimal pattern value by rotating bit arrangements. The LBP images corresponding to the DMHIs are shown in Figure 2 (2nd row).



(a) Output of 3×3 LBP operator (b) Left (c) Right (d) Up (e) Down

FIGURE 3. Illustration of different LBP bit arrangements

3.4. **Selective snippets.** We choose some frames containing the foreground mask from all the m_{fg_t} within the DMHI time duration, τ, of the performed action: We define those frames as selective snippets. We select only three frames as snippets and extract the pose information of the action in the form of a histogram (explained in Section 3.5) and called it as shape feature. To select the snippets, we determine the minimum bounding rectangle of the foreground mask in every m_{fg_t} and choose the frames with the property in Equations (12)-(14).

$$t_1 = \underset{t \in [0,1,\ldots,\tau]}{\arg\max}(h_t \times w_t) \tag{12}$$

$$t_2 = \underset{t \in [0,1,\ldots,\tau]}{\arg\max}\,(h_t / w_t) \tag{13}$$

$$t_3 = \begin{cases} \dfrac{t_1 + t_2}{2} & \text{if } |t_1 - t_2| > \dfrac{\tau}{2} \\ \left\lfloor \max(t_1, t_2) + \frac{\tau - |t_1 - t_2|}{2} \right\rfloor \pmod{\tau} & \text{otherwise} \end{cases} \tag{14}$$

$$S_k = m_{fg_k}; \quad k = t_1, t_2, t_3 \tag{15}$$

Here, h_t, and w_t are the height and width of the bounding rectangle of the foreground mask in frame m_{fg_t} at time t. And, S_{t_1} is the snippet with the pose covering a maximum area in the frame, S_{t_2} is the snippet with the pose having the narrowest possible area and S_{t_3} is simply an in-between snippet of S_{t_1} and S_{t_2}. We choose first two frames with a basic intuition that, the pose with maximum covering area and pose with minimum covering area provide some distinctive information for classification. The third frame is chosen only to make it more robust. Figure 4 displays the snippets, containing the foreground mask, selected for shape feature extraction.

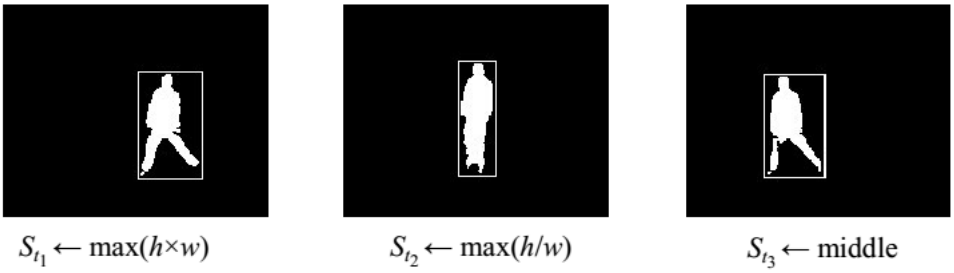

$S_{t_1} \leftarrow \max(h \times w)$
$S_{t_2} \leftarrow \max(h/w)$
$S_{t_3} \leftarrow$ middle

FIGURE 4. The snippets selected for the extraction of shape feature

3.5. **Feature vector generation.** An image texture can be described by its intensity distribution. We use the histogram of LBP images to represent an action. We compute the feature vector only for the action region in the images to make it invariant to translational effect. We use MEI to determine the action region. The MEI is deduced by accumulating all the DMHIs to a single image and then taking the image under a threshold equals

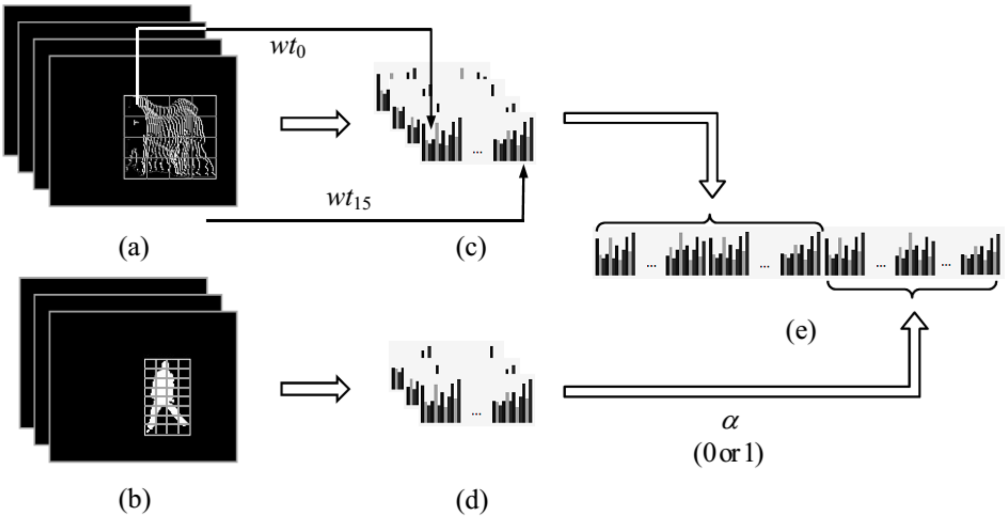


FIGURE 5. Generation of a feature vector: (a) LBP images where action regions are partitioned into 4 × 4 blocks, (b) selective snippets partitioned into 8×4 blocks, (c) concatenated block intensity (pattern) histogram of the LBP images, (d) block nonzero pixel frequency histogram of the snippets, (e) the feature vector – concatenated histogram of LBP images along the snippets

zero [3]. In this phase, we take the smallest bounding rectangle with maximum possible non-zero pixels in MEI as an action region.

Figure 5 illustrates the construction of a feature vector for representing an action. The action regions of the LBP images are partitioned into $p \times q$ disjoint blocks. For each block, we compute a weighted (See Equation (16)) LBP histogram splitting the entire pattern ranges (256 patterns) into r equal sized bins. Since an intensity histogram loses spatial information, rather than computing a single global histogram, we calculate block histograms to encapsulate some spatial information into the feature vector. These block histograms of all the images are concatenated together in a raster scanning fashion to form an action descriptor which can be individually treated as a feature vector for recognition.

However, along with the LBP histogram, we also use shape feature of the action represented as a histogram of selective snippets. In this case, we partition the action region of the selected snippets into a constant 8×4 blocks, and find the non-zero pixel distribution of the blocks which yields the snippet histogram. The snippet histogram is then put together with LBP histogram to form a larger feature vector. Here, we use a parameter α (0 or 1) to make the snippet histogram optional in the action descriptor and thereby measure its importance in a recognition rate. The algorithm presented next explains the details of the steps necessary for computation of the feature vector.

```
Algorithm: CreateFeatureVector
    Input: LBP images, $L_i$, $i = 0, 1, 2, 3$ and selective snippets $S_j$, $j = 0, 1, 2$ of an action
    Initialization: Find the bounding box denoting the action region
    LBP histogram $H_1 := 0$
    Snippets histogram $H_2 := 0$
    For each $L_i$ (for1)
        Partition the action region into $p \times q$ disjoint blocks
        LBP Image histogram $LIH_i := 0$
        For each block $b_k$, $k = 0, 1, \ldots, p \times q - 1$ (for2)
            Calculate weight $wt_i$ for the block
            $BH_k :=$ LBP histogram of $b_k$ splitting the pattern's range (0-255) into $r$ bins
            $BH_k := BH_k \times wt_i$
            $LIH_i := LIH_i || BH_k$, concatenate $BH_k$ with $LIH_i$
        End (for2)
        $H_1 := H_1 || LIH_i$, concatenate $LIH_i$ with $H_1$
    End (for1)
    $H_1 :=$ L2_norm ($H_1$)
    For each $S_j$ (for3)
        Partition the action region into $8 \times 4$ disjoint blocks
        Snippet Image histogram $SIH_j := 0$
        For each block $b_l$, $l = 0, 1, \ldots, 31$ (for4)
            $f_l :=$ Count non-zero pixel frequency of $b_l$
            $SIH_j := SIH_j || f_l$, concatenate $f_l$ with $SIH_i$
        End (for4)
        $H_2 := H_2 || SIH_j$, concatenate $SIH_j$ with $H_2$
    End (for3)
    $H_2 :=$ L2_norm ($H_2$)
    Output: Feature vector, $FV := H_1 || (H_2 \times \alpha)$, $\alpha = 0$ or 1, concatenate $H_1$ with $H_2$.
```

In the above algorithm, || is a concatenation operator, i.e., histograms are put side-by-side to form a larger histogram. The weight wt_i for each block is calculated using Equation (16), where, NP_{B_i} is the number of non-zero pixels (each pixel is a pattern) in block i, and NP_A means the number of non-zero pixels in the action region, and ε is a constant (≈ 0) useful for no action scene. Here, wt_i is always greater than or equal to one and the sum of the inverse of the weights equals one. Equation (17) gives the maximum possible number of dimensions of the feature vector, where $p \times q$ is the number of blocks; r is the number of bins.

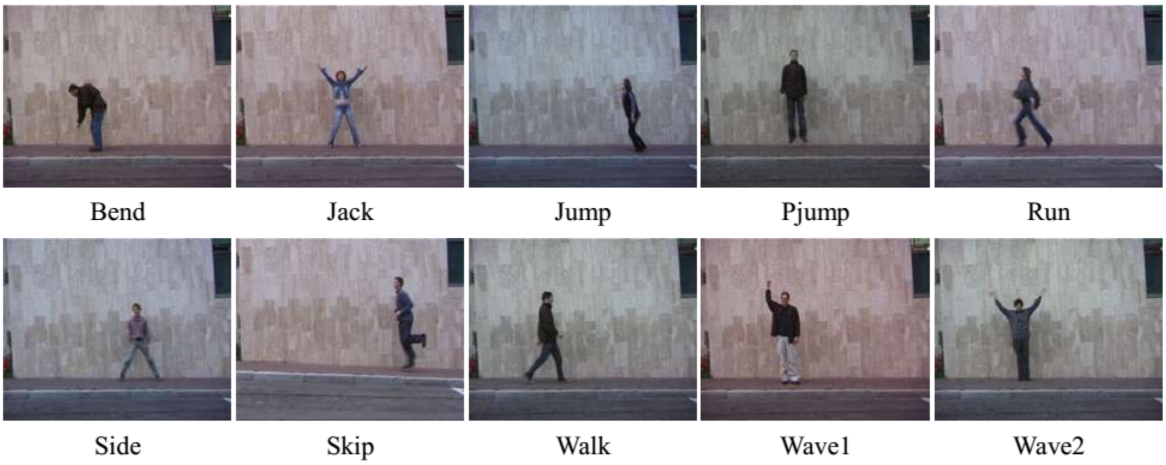
$$wt_i = \frac{1}{1 - NP_{B_i}/NP_A + \varepsilon}, \quad i = 0, 1, \ldots, p \times q - 1 \tag{16}$$

$$FV_{Dim} = 4 \times p \times q \times r + (8 \times 4) \times 3 \tag{17}$$

4. Experiments and Results.

4.1. Experimental setup.

We evaluate the performance of the proposed method by experimenting with popular benchmark database: the Weizmann action datasets [5]. The dataset consists of 90 sample videos showing nine different people, each performing 10 natural actions: “bend”, “jumping-jack” (“jack”), “jump-forward-on-two-legs” (“jump”), “jump-in-place-on-two-legs” (“pjump”), “run”, “gallop-side-ways” (“side”), “skip”, “walk,” “wave-one-hand” (“wave1”), and “wave-two-hands” (“wave2”). Figure 6 illustrates some sample frames of the Weizmann action dataset.



Bend Jack Jump Pjump Run

Side Skip Walk Wave1 Wave2

Figure 6. Sample frames of the different actions from Weizmann dataset

4.2. Method.

For the recognition purpose, we use a Support Vector Machine (SVM) [23,24]. SVMs are state-of-the-art large margin classifiers which have recently become very popular for visual pattern recognition [25,26] and many other applications. The multiclass classification problem is reduced down to binary classification one. To recognize k action classes, we train a bank of k linear one-vs-rest and ${}^kC_2$ linear one-vs-one binary SVMs, each with identical weights. 10 fold cross validation method is used to train and test the classifier. For the cross validation purpose, we stratify each partition of the dataset, i.e., each partition resembles the global distribution of the dataset. During the test, an unknown action is labeled with the class that gets the maximum votes by those classifiers.

For tracking the optical flow, dense Gunnar Farneback [27] algorithm is used and a flow threshold of ±1 pixel is used to separate the x and y directional flow into right ($+x$), left ($-x$), up ($+y$), and down ($-y$) directions. We use $\tau = 0.9$ second and $\delta = 1$ (as it was in the original) in Equation (7), i.e., only 0.9 second frames are used to create DMHI templates. Each DMHI is used to calculate the LBP image, where 3×3 neighborhood and $threshold = 1$ are used for Equations (10) and (11). We use $p = q = 2, 4, 6$ and $r = 8, 16, 32$ for the feature vector generation.

Although we have explained only one method of action representation, we performed the experiment with some variants of it and presented the comparative results. The representation methods used in experiment are (i) histogram of LBP image created from MHI (MHI_LBP_H), (ii) histogram of LBP image created from MHI along with shape feature of the selective snippets (MHI_LBP_H + SF), (iii) histogram of rotated bit arranged LBP image created from DMHI (DMHI_R_LBP_H), and (iv) histogram of rotated bit arranged LBP image created from DMHI along with shape feature of the selective snippets (DMHI_R_LBP_H + SF).

4.3. **Recognition results.** Figure 7 shows the correct classification rate of different representation methods for different number of blocks and bins. The results, presented here, are the average of three runs, i.e., the experiment is performed thrice, and each run consists of 10 fold cross validation. For all cases, DMHI_R_LBP_H + SF shows better accuracy than its corresponding (same p, q, r values) representations. We find that including the shape feature in action representation greatly improves the recognition rate for MHI_LBP_H, but in case of DMHI_R_LBP_H, the impact is more for lower number of blocks. However, in all cases, including the shape feature increases overall performance. Also in Figure 7, increasing the number of bins and blocks does not linearly increase the accuracy, rather after some point it starts to fall down (e.g., 6×6 blocks and 8, 16 bins). The best found recognition rate is 95.37%, which is observed for DMHI_R_LBP_H + SF with $p = q = 4$ and $r = 16$ bins.

Figure 8 presents the comparison of using rotating bit arrangement for creating different LBP images with that of a constant bit arrangement. Here DMHI_C_LBP_H means histogram of constant bit arranged LBP image created from DMHI. We find that in every

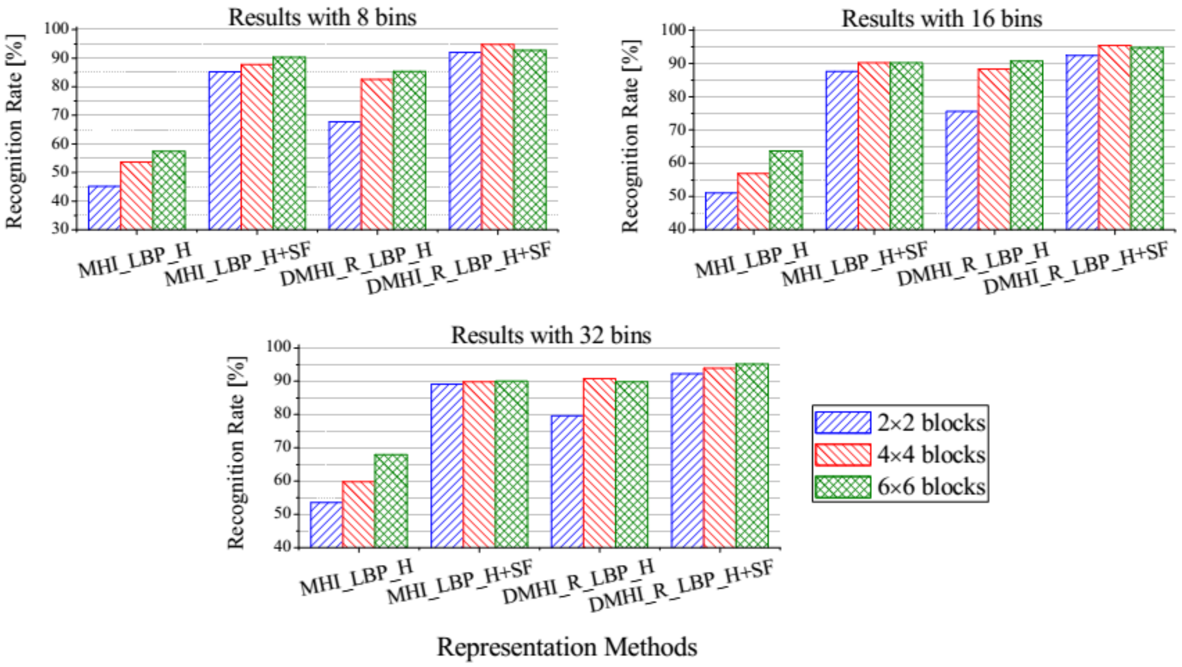


FIGURE 7. Correct recognition rate for different representations with various numbers of blocks and bins on Weizmann dataset

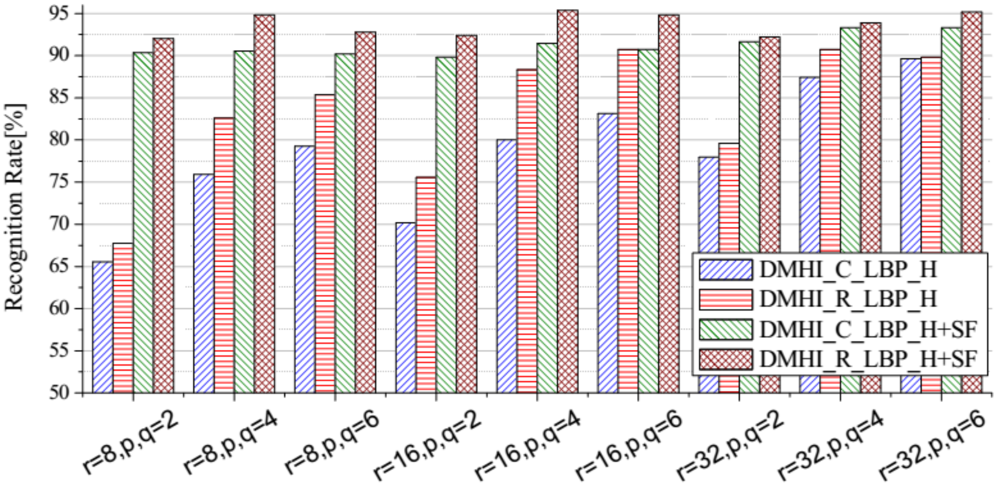


FIGURE 8. A performance comparison of using rotated and constant arranged bits for LBP image creation

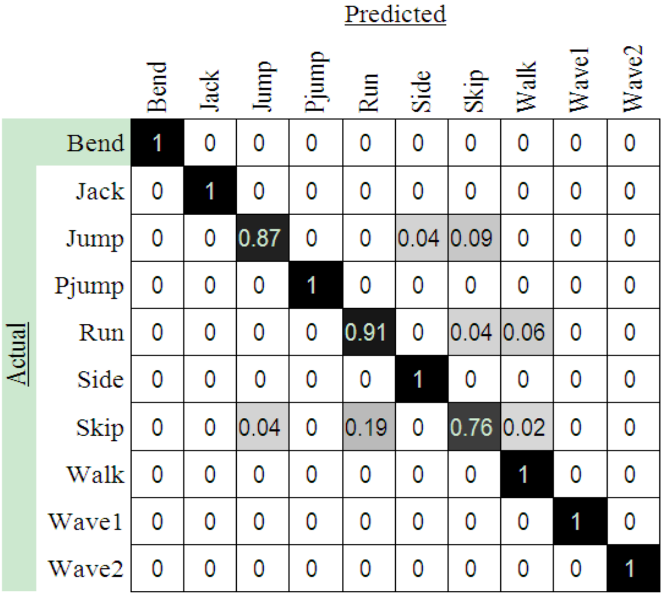
combination of p, q, and r values, DMHI_R_LBP_H performs better than DMHI_C_LBP_H which justify our claim presented in Section 3.3. This is because, if the same actions performed in leftward or rightward direction (e.g., walk, run, side-walk), the rotated arrangement of LBP tends to produce similar pattern value in left or right LBP image. These images in turn yield a histogram which is more consolidated and helps in better classification. The same thing also applies to upward or downward actions.

Table 1 summarizes the best accuracies found by our experiment with different action representation as well as the best results found by other methods on Weizmann dataset. The presented best result of DMHI_R_LBP_H + SF is for the parameter p, $q = 4$, and $r = 16$. These results are just indicative only, since different authors used a different number of frames for the feature vector generation or different types of classifiers and even different testing methods like leave-one-out. Some authors perform their experiments with 9 actions from Weizman dataset excluding the skipping action. We also do the same only for DMHI_R_LBP_H + SF representation, since it gives the best performance for all 10 actions. The best average classification rate found by the proposed method for 9 actions

TABLE 1. Comparison of the recognition rate of the proposed method to other methods reported on Weizmann dataset

	Reference	No. of Actions	Accuracy [%]
Proposed	MHI_LBP_H	10	67.96
	MHI_LBP_H + SF	10	90.37
	DMHI_R_LBP_H	10	90.74
	DMHI_R_LBP_H + SF	10, (9)	95.37, (98.96)
Archived in literature	Kelllokumpu et al. [1]	10, (9)	98.9, (100)
	Scovanner et al. [28]	10	82.6
	Boiman and Irani [29]	9	97.5
	Neibles and Fei-Fei [18]	9	72.8
	Wang and Suter [14]	10	97.8
	Ahsan et al. [30]	10	94.3
	Campos et al. [31]	10	96.7
	Ikizler and Duygulu [13]	9	100

TABLE 2. Confusion matrix of the DMHI_R_LBP_H + SF representation



Actual \ Predicted	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	1	0	0	0	0	0	0	0	0	0
Jack	0	1	0	0	0	0	0	0	0	0
Jump	0	0	0.87	0	0	0.04	0.09	0	0	0
Pjump	0	0	0	1	0	0	0	0	0	0
Run	0	0	0	0	0.91	0	0.04	0.06	0	0
Side	0	0	0	0	0	1	0	0	0	0
Skip	0	0	0.04	0	0.19	0	0.76	0.02	0	0
Walk	0	0	0	0	0	0	0	1	0	0
Wave1	0	0	0	0	0	0	0	0	1	0
Wave2	0	0	0	0	0	0	0	0	0	1

is also presented in the table (parenthesized in some cases). Though the recognition rate found from the proposed method is not the best compared to other methods, the achieved accuracy is reasonable enough and quite fast (See Table 3 for computational time) for practical application.

Table 2 shows the confusion matrix of DMHI_R_LBP_H + SF representation for the best result. Here, the accuracy is scaled down to one. All actions except for skipping the recognition rate are approximately 90% or above. In Weizmann dataset, some frames of the skipping poses have very subtle difference from the running poses, which is very difficult to distinguish even by humans. Therefore, the classifier puts some skipping actions as running ones.

4.4. **Computational time.** The run time of the method can be parted in two phases. First being the time to create the spatiotemporal templates, and the second is the feature vector generation and testing time, which depend on the values of p, q, r. Table 3 shows the average of per frame computational times (in milliseconds) of different phases for DMHI_R_LBP_H + SF representation over Weizman dataset. The feature vector generation and testing times in Table 3 are for best recognition rate parameters p, $q = 4$, $r = 16$ and a frame size of 144 × 180 pixel. It should be noted that the experiment is done on a machine with a processor Intel® Core$^{\mathrm{TM}}$ i7-3770, speed 3.40 GHz, and memory 8GB. We implement the program in Microsoft Visual Studio 2010, and OpenCV 2.4.8 without applying any code optimization method. However, it is worthy to mention that the total time reported in Table 3 excludes the foreground extraction time, since we are only interested in the action representation and recognition time. We are unable to compare the computational time of the proposed descriptor, since most of the state of

TABLE 3. Per frame computational time (in milliseconds) of the proposed method on Weizmann dataset

Template creation time	Feature vector generation time	Testing time	Total time
30.94	2.6	2.3	35.84

the art methods do not report about their computational time, except the HOR [13] that takes approximately one second per frame only for rectangle extraction phase which is far slower than the proposed one.

5. **Conclusion.** In this paper, we propose a novel approach for action recognition that represents a human action as a histogram of LBP images created from DMHIs, i.e., histogram of spatiotemporal texture and uses an SVM for recognition. It has been shown that without constructing any complex model, the proposed simple and compact descriptor performs well on different actions and the recognition rate is promising enough for practical use compared to the state of the art methods. We notice that, correct localization of the action region greatly affects the overall performance. Though we find that DMHI_R_LBP_H + SF representation performs better, we observe that MHI_LBP_H + SF representation with only one MHI image performs quite well too. To mitigate the scale variation, we always partition the action region in a constant number of blocks rather than fixed size blocks. The proposed method does not incorporate any direct mechanism for rotation invariance or view point changes. This can be future work of direction. The application of the descriptor to more complex actions or scenarios could be other possible future work.

This study is performed with a dataset having generic action classification problem. However, the method can easily be incorporated into some real life applications like gaming or human computer interaction without using any controller such as mouse, trackball, and joystick. The potential of the proposed method can also be applied to other related domains like a patient's activity monitoring system or automatic labeling of video sequences in a video dataset [32].

Acknowledgment. This study was supported by JSPS KAKENHI Grant Number 2535-0477, which is greatly acknowledged.

REFERENCES

[1] V. Kellokumpu, G. Zhao and M. Pietikäinen, Recognition of human actions using texture descriptors, *Machine Vision and Applications*, vol.22, no.5, pp.767-780, 2009.

[2] R. Poppe, A survey on vision-based human action recognition, *Image and Vision Computing*, vol.28, no.6, pp.976-990, 2010.

[3] A. F. Bobick and J. W. Davis, The recognition of human movement using temporal templates, *IEEE Trans. PAMI*, vol.23, no.3, pp.257-267, 2001.

[4] D. Weinland, R. Ronfard and E. Boyer, Free viewpoint action recognition using motion history volumes, *Computer Vision and Image Understanding*, vol.104, nos.2-3, pp.249-257, 2006.

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, Actions as space-time shapes, *Int. Conf. on Computer Vision*, pp.1395-1402, 2005.

[6] A. Yilmaz and M. Shah, Action sketch: A novel action representation, *Int. Conf. on Computer Vision and Pattern Recognition*, pp.984-989, 2005.

[7] G. Zhao and M. Pietikäinen, Dynamic texture recognition using local binary patterns with an application to facial expressions, *IEEE Trans. PAMI*, vol.29, no.6, pp.915-928, 2007.

[8] W. C. Yau, D. K. Kumar, S. P. Arjunan and S. Kumar, Visual speech recognition using image moments and multiresolution wavelet images, *Int. Conf. on Computer Graphics, Imaging and Visualisation*, pp.194-199, 2006.

[9] C.-P. Huang, C.-H. Hsieh, K.-T. Lai and W.-Y. Huang, Human action recognition using histogram of oriented gradient of motion history image, *Int. Conf. on Instrumentation, Measurement, Computer, Communication and Control*, pp.353-356, 2011.

[10] M. A. R. Ahad, T. Ogata, J. K. Tan, H. S. Kim and S. Ishikawa, View-based human motion recognition in the presence of outliers, *Int. Journal of Biomedical Soft Computing and Human Sciences*, vol.13, no.1, pp.71-78, 2008.

[11] W. T. Freeman and M. Roth, Orientation histograms for hand gesture recognition, *Int. Workshop on Automatic Face and Gesture Recognition*, pp.296-301, 1995.

[12] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *Int. Conf. on Computer Vision and Pattern Recognition*, pp.886-893, 2005.

[13] N. Ikizler and P. Duygulu, Histogram of oriented rectangles: A new pose descriptor for human action recognition, *Image and Vision Computing*, vol.27, no.10, pp.1515-1526, 2009.

[14] L. Wang and D. Suter, Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model, *Int. Conf. on Computer Vision and Pattern Recognition*, pp.1-8, 2007.

[15] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, Behavior recognition via sparse spatio-temporal features, *Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.65-72, 2005.

[16] C. Schuldt, I. Laptev and B. Caputo, Recognizing human actions: A local SVM approach, *Int. Conf. on Pattern Recognition*, pp.32-36, 2004.

[17] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, Actions as space-time shapes, *Int. Conf. on Computer Vision*, pp.1395-1402, 2005.

[18] J. C. Niebles and L. Fei-Fei, A hierarchical model of shape and appearance for human action classification, *Int. Conf. on Computer Vision and Pattern Recognition*, pp.1-8, 2007.

[19] K. Schindler and L. V. Gool, Action snippets: How many frames does human action recognition require? *Int. Conf. on Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[20] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Systems, Man, and Cybernetics*, vol.9, no.1, pp.62-66, 1979.

[21] M. A. R. Ahad, J. K. Tan, H. S. Kim and S. Ishikawa, Motion history image: Its variants and applications, *Machine Vision and Applications*, vol.23, no.2, pp.255-281, 2012.

[22] T. Ojala, M. Pietikainen and T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. PAMI*, vol.24, no.7, pp.971-987, 2002.

[23] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, New York, USA, 2000.

[24] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[25] C. Wallraven, B. Caputo and A. Graf, Recognition with local features: The kernel recipe, *Int. Conf. on Computer Vision*, pp.257-264, 2003.

[26] L. Wof and A. Shashua, Kernel principal angles for classification machines with applications to image sequence interpretation, *Int. Conf. on Computer Vision and Pattern Recognition*, pp.635-640, 2003.

[27] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, *Scandinavian Conf.*, pp.363-370, 2003.

[28] P. Scovanner, S. Ali and M. Shah, A 3-dimensional sift descriptor and its application to action recognition, *Int. Conf. on Multimedia*, pp.357-360, 2007.

[29] O. Boiman and M. Irani, Similarity by composition, *Neural Information Processing Systems*, pp.177-184, 2006.

[30] S. M. M. Ahsan, J. K. Tan, H. Kim and S. Ishikawa, Histogram of spatiotemporal local binary patterns for human action recognition, *Joint Int. Conf. on Soft Computing and Intelligent Systems and Int. Symposium on Advanced Intelligent Systems*, pp.1007-1011, 2014.

[31] T. D. Campos et al., An evaluation of bags-of-words and spatio-temporal shapes for action recognition, *Workshop on Applications of Computer Vision*, pp.344-351, 2011.

[32] S. M. M. Ahsan, J. K. Tan, H. Kim and S. Ishikawa, Human action representation and recognition: An approach to a histogram of spatiotemporal templates, *International Journal of Innovative Computing, Information and Control*, vol.11, no.6, pp.1855-1868, 2015.