Silhouette-based human action recognition using SAX-Shapes — Source link

Imran N. Junejo, Khurrum Nazir Junejo, Zaher Al Aghbari

Institutions: University of Sharjah, National University of Computer and Emerging Sciences

Published on: 01 Mar 2014 - The Visual Computer (Springer Berlin Heidelberg)

Topics: Silhouette

Silhouette-based Human Action Recognition using SAX-Shapes

Imran N. Junejo [1] [2] [a] Khurrum Nazir Junejo [b]
Zaher Al Aghbari [a]

[a] *University of Sharjah, Sharjah, U.A.E.*

[b] *National University of Computer and Emerging Sciences, Department of Computer Science, Karachi, Pakistan*

Abstract

Human action recognition is an important problem in Computer Vision. Although most of the existing solutions provide good accuracy results, the methods are often overly complex and computationally expensive, hindering practical applications. In this regard, we introduce the combination of time series representation for the silhouette and *Symbolic Aggregate approXimation* (SAX), which we refer to as **SAX-Shapes**, to address the problem of human action recognition. Given an action sequence, the extracted silhouettes of an actor from every frame are transformed into time-series. Each of these time series is then efficiently converted into the symbolic vector: SAX. The set of all these SAX vectors (SAX-Shape) represents the action. We propose a rotation invariant distance function to be used by a random forest algorithm to perform the human action recognition. Requiring only silhouettes of actors, the proposed method is validated on two public datasets. It has an accuracy comparable to the related works and it performs well even in varying rotation.

Key words: action recognition; computer vision; Time Series Shapelets; Symbolic Aggregate approXimation

1 Introduction

Recognizing human action from the videos is one of the most intensively studied problem in the field of Computer Vision. Application for this can be found in a wide range of industries: biometrics, sports, medial sciences, search and the structuring of large video archives, video surveillance, human-computer interaction, gesture

[1] This research is funded by University of Sharjah (Project 120227)
[2] Corresponding Author: ijunejo@sharjah.ac.ae, Tel: +97165053705 Fax: +97165050102

recognition and video editing. However, despite many efforts by the researchers, accurate recognition of human actions is still quite a daunting task, this is generally due to the complexity of the human action, which is dynamic, ambiguous, and interactive with other objects.

Generally, most of the approaches rely on extracting complex features from the data. These approaches can be roughly categorized on the basis of representation. Time evolution of human silhouettes was frequently used as action description. For example, [1] proposed to capture the history of shape changes using temporal templates and [2] extends these 2D templates to 3D action templates. Similarly, the notions of *action cylinders* [3], and *space-time shapes* [4,5] have been introduced based on silhouettes. Recently, space-time approaches analyzing the structure of local 3D patches in the video have been shown promising in [6,7,8,9]. Almost all of the works mentioned rely mostly on an effective feature extraction technique. Which is then combined with machine learning or pattern recognition techniques. These feature extraction methods can be roughly categorized into: *motion-based* [10,11], *appearance based* [5], *space-time volume* based [12,4,5], and *space-time interest points* or local features based [6,7,8,13]. Motion-based methods generally compute optical-flow from a given action sequence, followed by appropriate feature extraction [14]. However, optical-flow based methods are known to be very susceptible to noise and are easily led to inaccuracies. Appearance based methods are prone to differences in appearance between the already seen sequences (i.e. the training dataset) and the new sequences (i.e. the testing sequence). Volume or shape based methods require highly detailed silhouette extraction, which may not be possible in given real-world noisy video dataset. In comparison with these approaches, the space-time interest point (STIP) based methods [6,8] has received considerable attention and are thus popular. STIP-based methods are more robust to noise and camera movement and also seem to work quite well with low resolution inputs. However, these methods rely solely on the discriminative power of individual local space-time descriptors. Information related to the global spatio-temporal distribution is ignored. Thus due to lack of this temporal information, smooth motions cannot be captured using STIP methods. In addition, issues like optimal space-time descriptor selection and codebook clustering algorithm selection have to be addressed, with fine-tuning various parameters, which is highly data dependent [15].

Space-time interest point (STIP), 3D trajectories, space-time shapes and other constructs described above,however, are computationally complex. And at the same time, accurate location for these features is also an issue. Extracting trajectories from action sequences is very noise prone; STIP can be very sparse, and 3D shape reconstruction requires highly accurate low-level processing. These issues in turn make the solution impractical for large datasets. Furthermore, these features have to be stored for performing any kind of recognition, which also leads to storage issues. Dimensionality reduction helps somewhat, but at the cost of reduced accuracy. Most of the above mentioned works assume a fixed camera and the action have matching, which is a big assumption in real-world action recognition. In this regards, a simple, rotation invariant, and an efficient method is desired that scales well to large datasets, and yet provides comparable or better accuracy than existing methods.

In this work, given an action sequence, the silhouettes of tracked actors are extracted from every frame and transformed into a time series. This transformation is straight forward and efficient, relying only on the silhouette of a detected actor [16]. We then convert these time series into a new representation: Symbolic Aggregate approXimation (SAX) [17,18] in order to effectively address the action recognition problem. SAX tremendously reduces the dimensionality of time series that represents an action sequence. Moreover, we describe how the representation can be made rotation invariant. The set of all these SAX vectors (SAX-Shape) represents the action. We use the random forest algorithm to classify human actions due to their high accuracy compared to other classifiers as we show in our experiments (cf. Section 6). Other symbolic representations, such as Discrete Fourier Transform (DCT)[19], and Discrete Wavelet Transform (DWT)[20], may be used to reduce the memory space required to store the time series but the intrinsic dimensionality of the symbolic representation is not reduced [21]. Another desirable feature of SAX is its lower bounding property. That is, the Euclidean distance between two time series in the SAX space is guaranteed to be less than or equal to the distance between them in the original space [21]. Due to the above two features, SAX symbolic representation is used in this paper to efficiently and effectively recognize human actions from large sequences of video. Also, we do not extract any features from SAX representation and rely solely on the Euclidean distance.

The proposed method is illustrated in Fig. 1. A sample from a `bend` sequence is shown in Fig. 1[top row] from the Weizmann dataset. Here we show just one frame from the sequence for clarity of presentation. For each frame, the silhouettes are *first* extracted and then converted to a time series, a sample of which is shown in the second column. For this time-series, we compute the SAX-based representation, an example is shown in Fig. 1(last column), where the horizontal axis represents the SAX length segments that map trajectory values into SAX segments and the vertical axis represents the SAX symbols that quantize the trajectory. This particular example is only of length 16 and an alphabet size (i.e. distinct values) of only 16. This process thus not only reduces the time series size, but it also makes the proposed method computationally feasible. *Second*, once we have the SAX representation of the training and the test sequence, we used the machine learning technique defined in Section 5 for performing action recognition. The method does not require time or joint correspondences. We show results comparable to the state of the art methods, and the method is shown to be very fast and rotation invariant as well.

The paper is organized as follows: Section 2 reviews some of the work relevant to the action recognition and the time series representation. We introduce the conver-

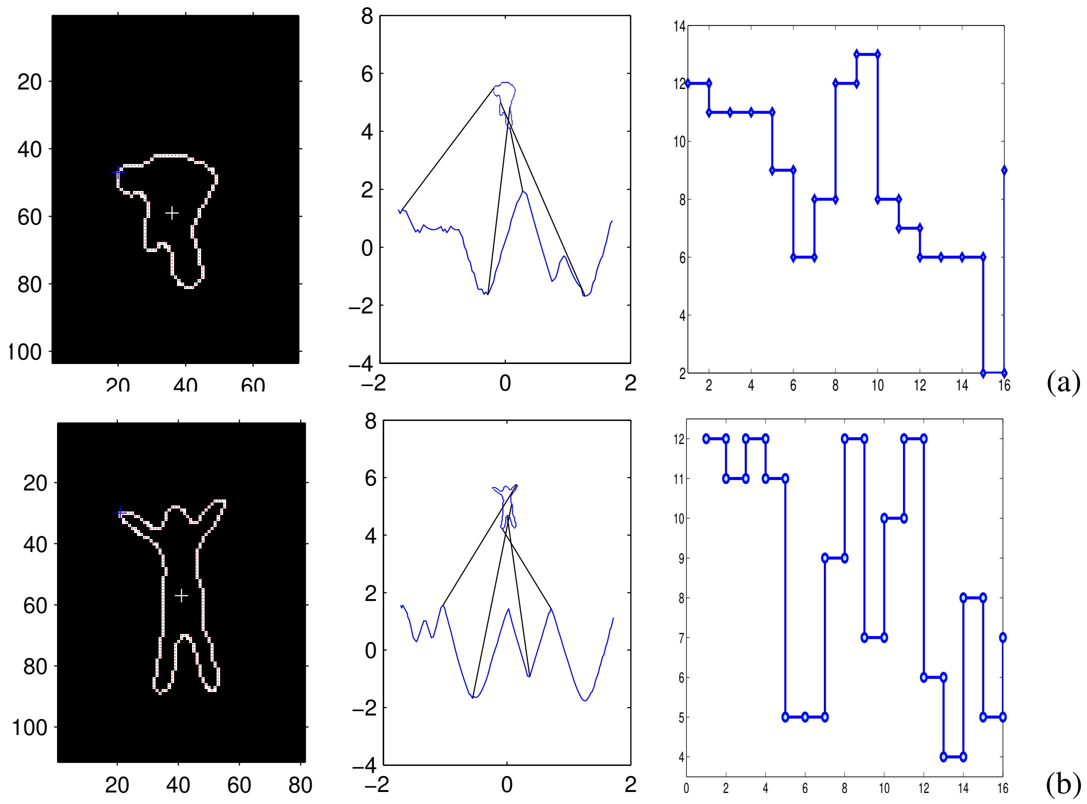


Fig. 1. SAX-Shape based action recognition approach: Two samples from the `bend` (a) and `jumpingjack` (b) sequences. The silhouettes are extracted from the actors. These extracted silhouettes are converted to time series using method in [16], as shown in the second column. The last column shows the SAX representation of the time series, having a length length(w) 16, and alphabet size(p) 16. The horizontal axis represents the SAX breakpoints that map time series values into SAX segments and the vertical axis represents the SAX symbols that quantize the trajectory (the upper limit of the vertical axis is in fact 16, not shown in the figure above due to scaling effects.

sion to time series in Section 3, followed by SAX representation of human action recognition in Section 4. We describe our action recognition scheme in Section 5, along with distance measure used to compare different SAX representations. Finally, we test the proposed method on a public datasets and demonstrate strong results in Section 6, followed by conclusion.

2 Related work

The input to our method is the silhouette of an actor performing a action sequence, which are then converted to time series. Once the silhouette is extracted, most of the techniques use representations such as chain codes [22], Fourier descriptors [19], shape moments, and shape context [23]. [24] represent each silhouette by a planar closed curve corresponding to the contour of the silhouette, and then evolve the shapes of these curves during actions and gestures recognition. [25] propose a local feature called temporal-state shape context (TSSC) to capture local space-time shape characteristics effectively from a silhouette. This feature uses temporal states for compensating time warping effects and shape contexts for space-time shape variations. Space-time shapes or tunnels have been explored by [4,5], where silhouettes from various frames are stacked together. [26] performs a geometric manifold embedding based on a tangent bundle obtained from the silhouette frames from given sample frames.

Treating silhouettes as a time-series data has a close relation to the works [27,28,29,30] (a full review of all the related work is beyond the scope of the current work). Using velocity history of (KLT) tracked points, [28] present a generative mixture model for the video sequences. A log-polar uniform quantization is performed on these tracks, with 8 bins for direction, and 5 for magnitude. Activity classes are modeled as weighted mixture of bags of augmented trajectory sequences, where each mixture component models a velocity history feature with a Markov Chain. The model is trained by using the EM algorithm. In addition to choose the best parameters for quantization, depending on the number of mixture components, the approach can be computationally expensive. Employing a probabilistic framework, [27] present a method for augmenting local features with spatio-temporal relationships. Using STIP-HOG based and trajectory-based features, probabilities are accumulated in a Naive-Bayes like fashion into a reduced number of bins, and SVMs are used to perform the final classification. Using concepts from the theory of chaotic systems [29], the trajectories of reference points are treated as non-linear dynamical system that are generating an action. Using delay-embedding scheme, each such reference trajectory is used to construct a phase space of appropriate dimension. Final representation of each action is a feature vector, containing dynamical and metric invariants of the constructed phase space, namely Lyapunov exponent, correlation integral and correlation dimension. Along the same lines, based on extracted silhouettes, [30] introduces a manifold embedding method, called Local Spatio-Temporal Discriminant Embedding (LSTDE), that projects points lying in a local neighborhood into the embedding space. The idea is that the points of the same class are close together in this space and the data points of different classes are far apart. The authors aim to find an optimal embedding by maximizing the principal angles between temporal subspaces associated with data points of different classes. However, the embedding is a costly affair, often relying on fine-tuning of many parameters such an embedding dimension. In this work, we show results comparable to [29] and a significant improvement over [30], in addition to proposing a simple and computationally less complex solution that is very efficient.

Typically, since the size of time series is too large to fit in main memory, reducing the size of time series using an appropriate representation emerged as an issue of interest in the data mining community. Many representation techniques have been proposed in the literature, but we will address in this section the most commonly used ones. Faloutsos et. al. introduced Discrete Fourier Transform, DFT,

[19] to reduce the dimensionality of subsequences which are then indexed using the reduced representation. The indexing was then improved in [20] by using the Discrete Wavelet Transform, DWT, to reduce the dimensionality of time series; however, DWT is effective for time series of lengths that are an integer power of two. The authors of [31] used the Piecewise Aggregate Approximation, PAA, in efficient data mining techniques. Then, the Singular Value Decomposition, SVD, was proposed by [32] to improve the accuracy of time series mining however at the expense of the computational cost. In contrast to the above mentioned techniques, the SAX representation uniquely has both desirable properties [21]: dimensionality reduction of time series and lower bounding of Euclidean distance between time series in the original space. Due to these features, several works have used the SAX representation in different applications, such as image DB classification [33,34].

Recently, although a different approach altogether, [13] introduces Prototype Trees to solve action recognition. We thus compare results of the proposed method with the Prototype Trees approach and demonstrate the effectiveness and efficiency of our tool.

3 Silhouette to Time Series

To recognize actions in the proposed approach, we extract the silhouettes of actors performing the actions from every video frame that contains these actions. These sequences of silhouettes are 2D representations of the performed actions. Each extracted silhouette is converted into a time series, which is a 1D representation of the action at a particular frame [16], as shown in Figure 1. Matching shapes in the 1D space has proven to achieve comparable or superior accuracy as compared to matching in the original 2D space [35,36].

The method of extracting time series from 2D silhouettes does not consider the rotational alignment of these silhouettes and hence the extracted time series might be misaligned. Such rotational misalignment may greatly degrades recognition accuracy. Even a small rotational misalignment in the extracted time series can change the rotation by as much as 90 degrees [37]. Therefore, in the proposed approach, we use a rotation invariant distance function to mitigate the brittleness of comparing 1D representations of action silhouettes.

In the literature, several works have tried to minimize the effect of rotation misalignment, or invariance [35]. These works can be divided into three categories:

- Works that employ a landmark on the silhouette to indicate the start of a rotation. For example, a chin or a nose may be used as the landmark of a face profile [38]. Others have used the major axis of a silhouette as a landmark [39]. However, the difficulty in these works is in locating these landmarks, which in turn becomes the source of recognition inaccuracies.

- Works that extract rotation invariant features such as min curvature, max curvature, circularity, ratio of perimeter to area, elongatedness, etc. These features are stored in a feature vector to represent the image shape. Although this approach seems attractive and can easily be extended to represent 3D shapes, it suffers from poor recognition accuracies [40].
- Works that use exhaustive brute force search over all possible rotations. To find the best match between two rotated silhouettes represented as time series of length n, one sequence is fixed and the other is circularly shifted n times. Such works [41,42] find the best rotation and thus achieve better recognition accuracies as compared to the other two approaches, but at the expense of computational efficiency.

In the proposed approach, we follow the brute force approach since it gives the best recognition accuracies, however we employ a technique to speed it up by orders of magnitude and yet maintain identical accuracies.

Given two time series Q and T of length n that are representing two silhouettes, previous works achieve rotational invariance by fixing one and circularly shifting the other one n times. The rotation that gives the best match (smallest distance) is considered to be the true rotation. Such comparison is performed on the raw time series. In the proposed approach, we first convert both time series into their SAX representations, Q_{SAX} and T_{SAX}, where each time series is represented by w segments, $w \ll n$. That is, w is smaller than n by orders of magnitude and this results in great computational time savings, compared to computations with raw time series.

$$Q_{SAX} : s_1^q, s_2^q, \ldots, s_w^q \tag{1}$$

$$T_{SAX} : s_1^t, s_2^t, \ldots, s_w^t \tag{2}$$

Our approach fixes Q_{SAX} and rotates T_{SAX} w times, as shown in the rotation matrix below:

$$\begin{bmatrix} s_1^t & s_2^t & \ldots & s_{w-1}^t & s_w^t \\ s_2^t & s_3^t & \ldots & s_w^t & s_1^t \\ & & \vdots & & \\ s_w^t & s_1^t & \ldots & s_{w-2}^t & s_{w-1}^t \end{bmatrix} \tag{3}$$

Each row T_{SAX}^i in the rotational matrix is matched with time series Q_{SAX} and therefore w distance computations are performed. The proposed approach considers the minimum distance among the w distances as the best rotational invariance

Table 1
Symbols used in the paper

Symbol	Meaning
T	Time series
Q	Query time series
n	Length of time series
w	Number of PAA segments
p	Number of alphabets used to represent the time series
β_i	A breakpoint under a Gaussian curve that divides the curve into p equi-probable regions.

distance, RID.

$$RID(Q_{SAX}, T_{SAX}) = min_{1<i<w}(distance(Q_{SAX}, T^{i}_{SAX})) \tag{4}$$

RID distance function has a complexity of $O(w^2)$, however w is much smaller than the length of the original time series n. Nevertheless, the proposed approach produces accuracies identical to those produced by the original time series as shown in our experiments.

4 SAX - time series representation

We briefly review the SAX (Symbolic Aggregate approXimation) representation of a time series as presented in [21]. Although there are many representations for time series in the literature, SAX is the only one that reduces the dimensionality and lower bounds the Lp norms (distance functions) [43], which guarantees that the distance between any two vectors in the SAX representation is smaller than, or equal to, the distance between these two vectors in the original space. Table 1 explains the symbols used in this paper.

We begin by a formal definition of time series:

Definition A time series $T = \{t_1, t_2, t_3, \cdots, t_n\}$, t_n is an ordered set of n real-valued variables.

Before dealing with time series, we have to remove the distortions, namely the offset translation and the amplitude scaling, that could greatly affect the results of the action recognition tasks. Thus, we normalize each time series to have a mean of zero and a standard deviation of one, since it is well understood that it is meaningless to compare two time series with different offsets and amplitudes. Eq (5) is used

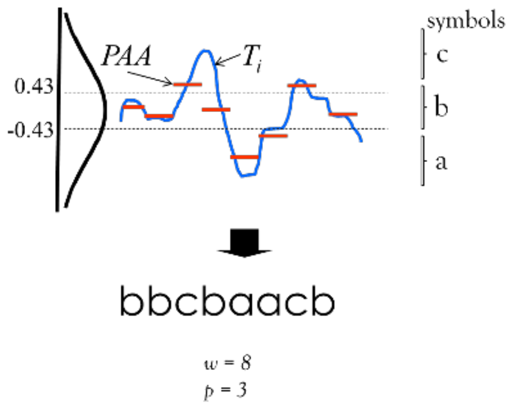


Fig. 2. A time series T_i (smooth curve) is converted to PAA segments and then using the pre-determined breakpoints the PAA coefficients are mapped into symbols (SAX representation).

to normalize the time series.

$$T = \frac{T_o - mean(T_o)}{std(T_o)} \tag{5}$$

where T_o is the original time series, $mean(T_o)$ is the mean value of time series variables, and $std(T_o)$ is the standard deviation of the time series variable.

4.1 Converting time series to SAX representation

After normalizing the time series, it is converted to a Piecewise Aggregate Approximation (PAA) and then to the symbolic representation SAX (see Figure 2).

The conversion process is performed by the *time series to SAX representation algorithm*, which takes as input the time series, T, its length, n, the number of PAA segments, w, and the number of alphabets used to represent the time series, p.

The details of the *time series to SAX representation algorithm* are explained below:

```
Algorithm: time series to SAX representation
Input: $T, n, w, p$
Output: SAX representation

  (1) Normalize $T$ using Eq (5).
  (2) Set the number of PAA segments i.e. $w$.
  (3) Transform $T$ into $PAA$ coefficients using Eq (6).
  (4) Convert the $PAA$ coefficients into SAX symbols
```

- An input time series T of length n is normalized using Eq (5).
- Choose an appropriate value for w.
- Transform T into segments using PAA by dividing the length n of T into w equal-sized "frames". The mean value of the data falling within a frame is calculated using Eq (6) and a vector of these values (PAA coefficients) becomes the data-reduced representation.

$$\overline{x_i} = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} x_j \tag{6}$$

- Since normalized time series follow the Gaussian distribution [18,21], breakpoints are determined so that they produce equal-sized areas under the Gaussian curve. The Gaussian curve used to compute the breakpoints has been defined to represent the distribution of the normalized time series and thus it has a mean of zero and standard deviation of one. Breakpoints are a sorted list of numbers $B = \{\beta_1, \beta_2, \cdots, \beta_{p-1}\}$ such that the area under a Gaussian curve from β_i to $\beta_{i+1} = 1/p$ (β_0 and β_p are defined as $-\infty$ and $+\infty$, respectively), see Fig. 2. These breakpoints can be taken from a breakpoint lookup table [18,21] and they divide the amplitude values of the time series into p equi-probable regions as follows:
  - Depending on the value of p, the breakpoints are determined.
  - All the PAA coefficients that are below the smallest breakpoint are mapped to the first symbol, say `a` all coefficients greater than or equal to the smallest breakpoint and less than the second smallest breakpoint are mapped to the second symbol, say `b`, and so on.
- After mapping all PAA coefficients to their corresponding symbols, we get a SAX representation of the input time series. For example, in Fig. 2 the SAX representation of the time series T_i is `bbcbaacb`.

5 Action Recognition

The outline of the proposed method is given below:

```
Algorithm: Action Recognition
Input: Action Sequence
Output: class association

  (1) perform foreground extraction.
  (2) localize the object silhouette per frame.
  (3) convert the silhouette to the time series (cf. Section 3).
  (4) Convert time series of each frame to SAX representation (cf. Section 4).
  (5) perform model training using the random forest method (cf. Section 6).
  (6) perform action recognition.
```

The method first start with acquiring the action sequence and performing the foreground attraction [44]. For the current setup, we use the silhouettes provided by [4]. Once the silhouettes are extracted, we use the method detailed in Section 3 to convert it in to the time series, these time series are then converted into their SAX representation as detailed in Section 4. Figure 3 graphically illustrates this process. The top left image, from a dataset sequence, is to be matched with the possible rotations of SAX representations of the test sequences (e.g. Figure 3, top right image shows a few rotations of the test image for illustration purpose). The middle row shows that a perfect match is obtained when the test SAX vector matches with a SAX vector of the target sequence. The bottom row shows the votes obtained for each match.

Based on extensive experiments, the authors of [21,43] show that the parameters alphabet size, p, and word size, w, are not too critical and a word size in the range of $5-8$ and an alphabet size of $3-10$ seem reasonable for the task at hand. However, the appropriate value of w and p depends on the data. That is, smaller values of w are more suitable for relatively smooth and slowly changing time series and on the other hand larger value of w are appropriate for fast changing time series. Also, smaller values of p would result in clustering time series values in few alphabets and larger values of p works against the SAX goal to reduce the dimensionality of time series.

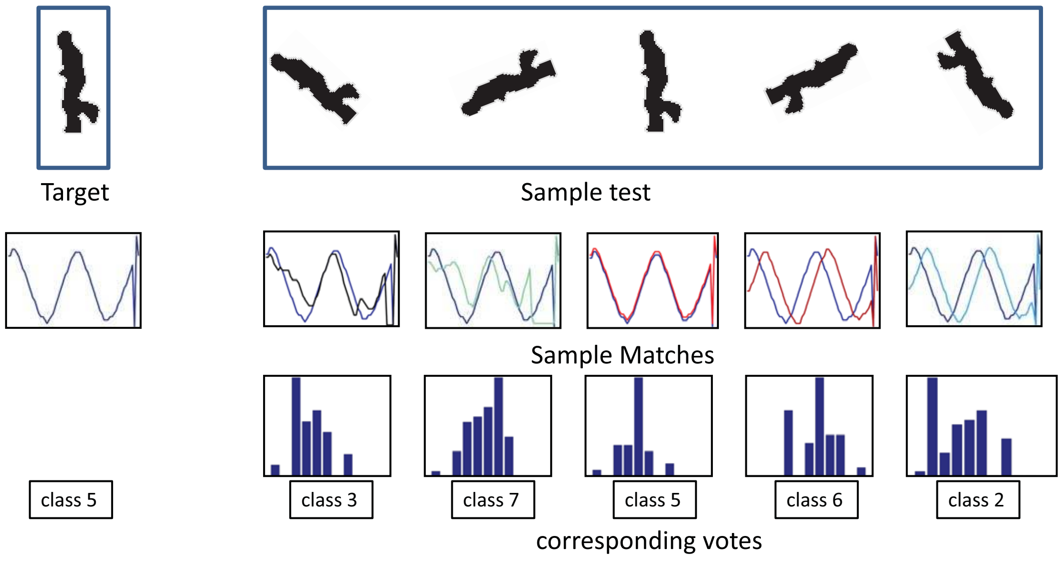


Fig. 3. Top-left denotes a target frame being matched to rotated versions of it, shown in the top-right. The middle figure denotes the obtained time series (which are then converted to the SAX representation) version of the silhouette and it matchings. The bottom figure shows the class votes attained by the target sequence after matching to all frame.

6 Experiments & Results

In this section we evaluate our approach for the task of human action recognition. The input to the method is the silhouette of the actor for every frame of the action sequence. The silhouettes are then converted into time series in the form of xy-trajectories.

To assess the discriminative power of our method on real video sequences, we apply it to a standard single-view video Weizmann dataset [4](see Fig. 4) and to the more advanced IXMAS dataset [45]. The Weizmann dataset consists of videos of 10 different actions performed by 9 actors. Each video clip contains one subject performing a single action. The 10 different action categories: *walking, running, jumping, gallop sideways, bending, one-hand-waving, two-hands-waving, jumping in place, jumping jack*, and *skipping*. Each of the clip lasts about 2 seconds at 25Hz with image frame size of 180×144.

We take one sequence from each class (i.e. 10 sequences) and form the test set; this is about 11.65% of the total data. We report results on the following algorithms:

J48 Decision Trees [46] is a decision tree algorithm that places the attribute having the highest information gain closer to the root of the tree.

RF is a random forest algorithm that grows many classification trees. To classify a new action, its SAX vector is inputted down each of the trees in the forest. Each tree gives a classification, that is the tree “votes” for that class. Then, the forest

Table 2
Percentage accuracy obtained by testing on various methods various methods.

SAX params	**IBK**	**BayesNet**	**J48**	**RF**
$w = 64; p = 8$	70	30	50	60
$w = 64; p = 32$	70	50	50	70
$w = 64; p = 48$	60	30	30	70

chooses the classification having the most votes [47].

IBK is a lazy algorithm based on KNN that supports instance-based learning, which do not maintain a set of abstractions derived from specific instances. It extends the nearest neighbor algorithm, which has large storage requirements [48].

BayesNet: Bayes Network learning using various search algorithms and quality measures. Base class for a Bayes Network classifier. Provides data structures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like **K2** and **B**.

We now report the accuracy in terms of the videos correctly classified by majority vote in the above set: Table 2 shows the results obtained for the dataset by using the methods mentioned above for the different SAX parameters. The worst performance was noticed with **BayesNet** and **J48**. The performance with **IBK** and and **RF** was noticed to be similar. The number of trees used for this experiment for the **RF** was 500. We choose **RF** as our method of preference as its performance improves with increase in number of trees, and it is easier to increase number of trees than increasing the number of attributes.

Table 3 shows the results while using **RF** only while using the leave-one-out cross-validation approach. The first column shows the different SAX parameters used for the experiment and the corresponding percentage errors are mentioned in column two. Although not a huge difference, the best result was obtained with the SAX parameters $w = 64; p = 32$ to be 11.76%.

Table 3
Error rates for various SAX parameters with RF method.

SAX params	Percentage Error
$w = 64; p = 8$	16.47
$w = 64; p = 16$	15.29
$w = 64; p = 32$	11.76
$w = 96; p = 8$	15.29
$w = 96; p = 16$	12.94
$w = 96; p = 32$	15.29

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	89.0	0.0	0.0	11.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	67.0	11.0	0.0	0.0	22.0	0.0	0.0	0.0
pjump	0.0	0.0	22.0	78.0	0.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	89.0	11.0	0.0	0.0	0.0
skip	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
walk	0.0	0.0	0.0	0.0	37.0	0.0	0.0	63.0	0.0	0.0
wave1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
wave2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Fig. 4. Weizman dataset: The top row shows instances from the nine different action sequences. The bottom row depicts recognition results. The dataset contains ninety-three action sequences videos of varying lengths, each having a resolution of 180×144. The nine difference action are: `bending down`, `jumping back`, `jumping`, `jumping in place`, `galloping sideways`, `running`, `walking`, `waving one hand` and `waving two hands`.

The accuracy with **RF** is computed to be around 89% with the number of trees 1500. The corresponding confusion matrices are illustrated in Fig. 4(bottom). Some errors occur when `jump` action is confused with the `skip` action; with `jump` and `pjump` as the two actions are quite similar generating same SAX representation; or when the `walk` is confused with the `running` action due to the similarity of the action, as shown in Figure 4. Although the obtained accuracy is not state of the art on this dataset, it is comparable, if not better, than the methods that treat trajectory data as time series, i.e. 89.7% for [29] (however, they require joint tracking which is impractical), and 80% for [30]. Similarly, [13] recently introduced prototype trees to solve action recognition and reports an accuracy of 88.89% (when using motion information only). However, these methods are computationally expensive. Whereas higher recognition rates on this dataset have been reported, e.g., in [49], the main strength of our method is the fast and the simple solution making our method suitable for various action recognition applications.

IXMAS Dataset

Table 4
Accuracy recognition rate.

Method	Accuracy
Our Method	89%
[30]	80%
[13]	88.89%

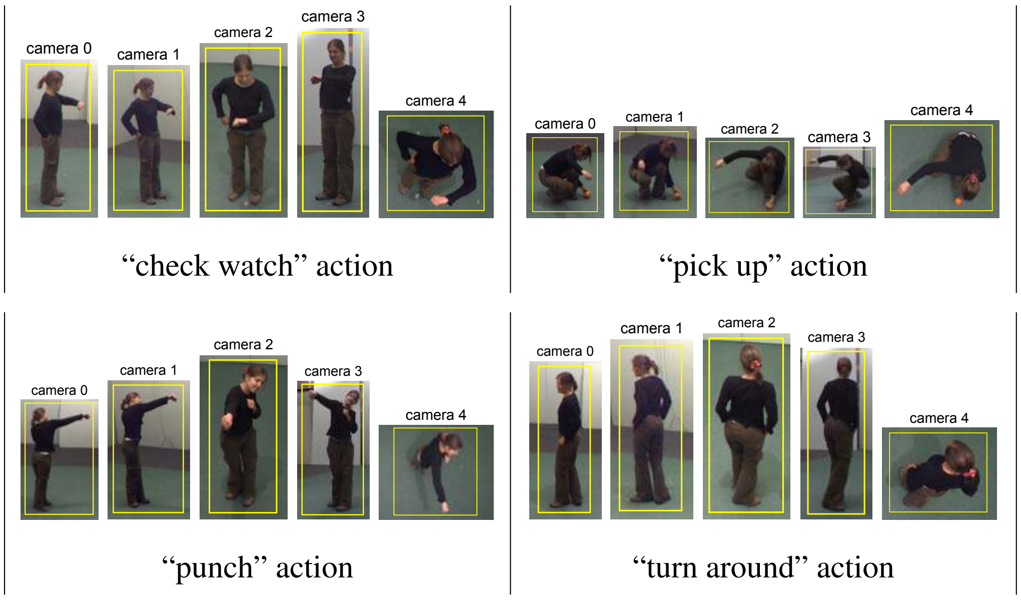


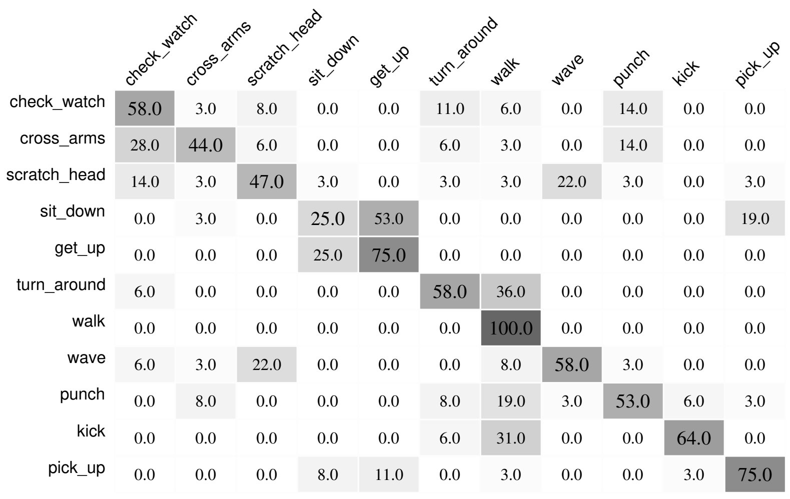
Fig. 5. **Example frames from IXMAS multiview action dataset**: for four classes of action, the five views at a given instant, of one performance of the action is shown.

Table 5
Comparison of recognition results on IXMAS dataset by alternative methods.

	cam 0	cam 1	cam 2	cam 3	cam 4
Our Method	**55.2**%	61.5%	**54.6**%	59.7%	**50.4**%
Weinland [45] 3D	65.4%	70.0%	54.3%	66.0%	33.6%
Weinland [45] 2D	55.2%	63.5%	—	60.0%	—

IXMAS dataset is publicly available and numerous researchers have reported their results on this dataset. Without resorting to engineering a different experimental setup to test view invariance, using this dataset allows for a quick and a fair comparison of our method to the other methods. Thus we present results for IXMAS video dataset [45] with 11 classes of actions performed three times by each of 10 actors and recorded simultaneously from 5 different views. Sample frames for all cameras and four action classes are illustrated in Fig. 5.

We compare our results with the standard method of Weinland [45], as shown in Table 5. We show the results on all cameras, and as can be seen, we out perform



	check_watch	cross_arms	scratch_head	sit_down	get_up	turn_around	walk	wave	punch	kick	pick_up
check_watch	58.0	3.0	8.0	0.0	0.0	11.0	6.0	0.0	14.0	0.0	0.0
cross_arms	28.0	44.0	6.0	0.0	0.0	6.0	3.0	0.0	14.0	0.0	0.0
scratch_head	14.0	3.0	47.0	3.0	0.0	3.0	3.0	22.0	3.0	0.0	3.0
sit_down	0.0	3.0	0.0	25.0	53.0	0.0	0.0	0.0	0.0	0.0	19.0
get_up	0.0	0.0	0.0	25.0	75.0	0.0	0.0	0.0	0.0	0.0	0.0
turn_around	6.0	0.0	0.0	0.0	0.0	58.0	36.0	0.0	0.0	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
wave	6.0	3.0	22.0	0.0	0.0	0.0	8.0	58.0	3.0	0.0	0.0
punch	0.0	8.0	0.0	0.0	0.0	8.0	19.0	3.0	53.0	6.0	3.0
kick	0.0	0.0	0.0	0.0	0.0	6.0	31.0	0.0	0.0	64.0	0.0
pick_up	0.0	0.0	0.0	8.0	11.0	0.0	3.0	0.0	0.0	3.0	75.0

Fig. 6. Confusion matrix for `cam 3` for the IXMAS dataset.

on some of the views. Our method is also comparable to the work in Weinland [45] that is based on extracting 3D shapes from multiple views. The confusion matrix for the best results obtain in `cam 3` is shown in Fig. 6. As mentioned above, this is a difficult datasets, and there is a great variation in the different instances of the action and in the way an action is performed by different actors. For example, `cross-watch` is confused with the `punch` action, as both contain moving one arm forward; `scratch-head` is confused with `check-watch` and wave for similar reasons; `sit-down` is confused with `get-up` etc.

7 Conclusion

In conclusion, we have presented a unique action recognition solution, adopted from the field of data mining and introduced it to the computer vision community. Silhouettes from the actors are first extracted and are converted to the time series. We then use an effective and an efficient method to convert this time series to SAX representation. The SAX representation tremendously reduces the dimensionality of the time-series, thus reducing the strain on the memory space and computational power required, and is shown to be fast as well. We believe the proposed method is very practical and considering the existing solutions, is very suitable for various action recognition applications. We test on two public datasets and report encouraging results. In addition to its simplicity and low computational costs, we have shown that the results obtained are comparable, if not better, to some of the existing methods.

References

[1] A. Bobick, J. Davis, The recognition of human movement using temporal templates, PAMI 23 (3) (2001) 257–267.

[2] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, CVIU 103 (2-3) (2006) 249–257.

[3] T. Syeda-Mahmood, M. Vasilescu, S. Sethi, Recognizing action events from multiple viewpoints, in: Proc. EventVideo, 2001, pp. 64–72.

[4] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, PAMI 29 (12) (2007) 2247–2253.

[5] M. Grundmann, F. Meier, I. Essa, 3d shape context and distance transform for action recognition, in: Proc. ICPR, 2008, pp. 1–4.

[6] I. Laptev, On space-time interest points, IJCV 64 (2/3) (2005) 107–123.

[7] J. Liu, B. Kuipers, S. Savarese, Recognizing human actions by attributes, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3337 –3344.

[8] J. Niebles, H. Wang, F. Li, Unsupervised learning of human action categories using spatial-temporal words, in: Proc. BMVC, 2006.

[9] W. Choi, K. Shahid, S. Savarese, Learning context for collective activity recognition, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3273 –3280.

[10] A. A. Efros, A. C. Berg, E. C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: ICCV, 2003, pp. 726–733.

[11] L.-P. Morency, A. Quattoni, T. Darrell, Latent-dynamic discriminative models for continuous gesture recognition, in: Proc. CVPR, 2007.

[12] A. Yilmaz, M. Shah, Actions sketch: A novel action representation, in: Proc. CVPR, 2005, pp. I:984–989.

[13] Z. Lin, Z. Jiang, L. S. Davis, Recognizing actions by shape-motion prototype trees, in: Proc. ICCV, 2009.

[14] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: CVPR, 2011, pp. 3169–3176.

[15] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in: Proc. CVPR, 2009, pp. 1948–1955.

[16] L. Ye, E. J. Keogh, Time series shapelets: a new primitive for data mining, in: Knowledge Discovery and Data Mining, 2009, pp. 947–956.

[17] E. J. Keogh, M. J. Pazzani, Scaling up dynamic time warping for datamining applications, Knowledge Discovery and Data Mining (2000) 285–289.

[18] J. Lin, E. J. Keogh, L. Wei, S. Lonardi, Experiencing sax: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144.

[19] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time-series databases, SIGMOD Rec. 23 (2).

[20] K.-P. Chan, W.-C. Fu, Efficient time series matching by wavelets, Data Engineering, International Conference on 0.

[21] J. Lin, E. J. Keogh, S. Lonardi, B. Y. chi Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2–11.

[22] H. Freeman, On the encoding of arbitrary geometric configurations, Electronic Computers, IRE Transactions on EC-10 (2) (1961) 260 –268.

[23] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (4) (2002) 509 –522.

[24] Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds, Computer Vision and Image Understanding 115 (3) (2011) 439 – 455.

[25] P.-C. Hsiao, C.-S. Chen, L.-W. Chang, Human action recognition using temporal-state shape contexts, in: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, 2008, pp. 1 –4.

[26] C.-S. Lee, Y. M. Lui, S. Y. Chun, in: ICCV Workshops, 2011, pp. 1318–1323.

[27] P. Matikainen, M. Hebert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in: N. P. K. Daniilidis, P. Maragos (Ed.), European Conference on Computer Vision 2010 (ECCV 2010), 2010.

[28] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, 2009, pp. 104–111.

[29] S. Ali, A. Basharat, M. Shah, Chaotic invariants for human action recognition, in: Proc. ICCV, 2007.

[30] R. Li, T. Tian, S. Sclaroff, Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series, in: Proc. ICCV, 2007.

[31] E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in: In ICDM, 2001, pp. 289–296.

[32] I. Popivanov, Similarity search over time series data using wavelets, in: In ICDE, 2002, pp. 212–221.

[33] Z. A. Aghbari, Effective image mining by representing color histograms as time series, JACIII 13 (2) (2009) 109–114.

[34] I. N. Junejo, Z. A. Aghbari, Using sax representation for human action recognition, J. Visual Communication and Image Representation 23 (6) (2012) 853–861.

[35] E. Keogh, L. Wei, X. Xi, S. hee Lee, M. Vlachos, Lb keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures, in: 33rd Very Large Data Bases Conference, 2006.

[36] X. Xi, E. Keogh, L. Wei, A. Mafra-Neto, Finding motifs in database of shapes, in: In Proc. SIAM Intl. Conf. On Data Mining, 2007.

[37] J. Zunic, P. L. Rosin, L. Kopanja, On the orientability of shapes, Trans. Img. Proc. 15 (11) (2006) 3478–3487.

[38] B. Bhanu, X. Zhou, Face recognition from face profile using dynamic time warping, in: In Proc. Intl. Conference on Pattern Recognition, 2006.

[39] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (1) (2004) 1–19.

[40] A. Cardone, S. K. Gupta, M. Karnik, A survey of shape similarity assessment algorithms for product design and manufacturing applications, Journal of Computing and Information Science in EngineeringJ. Comput. Inf. Sci. Eng. 3 (2003) 109 – 118.

[41] T. Adamek, N. E. O'Connor, A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Trans. Circuits Syst. Video Techn. (2004) 742–753.

[42] E. Attalla, P. Siy, Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching, Pattern Recogn. 38 (12) (2005) 2229–2241.

[43] E. J. Keogh, J. Lin, A. W.-C. Fu, Hot sax: Efficiently finding the most unusual time series subsequence, in: IEEE International Conf. on Data Mining (ICDM), 2005, pp. 226–233.

[44] O. Javed, M. Shah, Tracking and object classification for automated surveillance, in: the seventh European Conference on Computer Vision (ECCV), 2002.

[45] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3D exemplars, in: Proc. ICCV, 2007.

[46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: An update, SIGKDD Explorations 11 (1).

[47] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[48] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.

[49] N. Ikizler, P. Duygulu, Human action recognition using distribution of oriented rectangular patches, in: Workshop on Human Motion, 2007, pp. 271–284.