Chaotic Invariants for Human Action Recognition

Saad Ali
Computer Vision Lab
University of Central Florida
sali@cs.ucf.edu

Arslan Basharat
Computer Vision Lab
University of Central Florida
arslan@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Abstract

The paper introduces an action recognition framework that uses concepts from the theory of chaotic systems to model and analyze nonlinear dynamics of human actions. Trajectories of reference joints are used as the representation of the non-linear dynamical system that is generating the action. Each trajectory is then used to reconstruct a phase space of appropriate dimension by employing a delay-embedding scheme. The properties of the reconstructed phase space are captured in terms of dynamical and metric invariants that include Lyapunov exponent, correlation integral and correlation dimension. Finally, the action is represented by a feature vector which is a combination of these invariants over all the reference trajectories. Our contributions in this paper include :1) investigation of the appropriateness of theory of chaotic systems for human action modelling and recognition, 2) a new set of features to characterize nonlinear dynamics of human actions, 3) experimental validation of the feasibility and potential merits of carrying out action recognition using methods from theory of chaotic systems.

1. Introduction

Human actions consist of spatio-temporal patterns that are generated by a complex and time varying non-linear dynamical system. A complete description of this system will require enumeration of all independent variables, their interdependencies, differential equations controlling their evolution and a set of boundary conditions to be satisfied by the system. Ideally, one would like to have this complete description so that it can be used to control, predict, and extract features of the dynamical system in a deterministic fashion. However, in practical scenarios obtaining a complete analytic description is extremely hard.

In computer vision literature, the problem of obtaining the description of a dynamical system is often overcome by selecting a set of variables defining the state space, and a function that maps the previous state to the next state. The type of the mapping function determines whether it is a linear, non-linear or stochastic dynamical system. For instance, human actions can be represented in terms of state variables defined as the image locations of body joints, followed by assuming that a linear [7], non-linear [8] or stochastic dynamical model [10] is controlling the evolution of these state variables. The unknown parameters of the dynamical model are learnt using a training data of human actions.

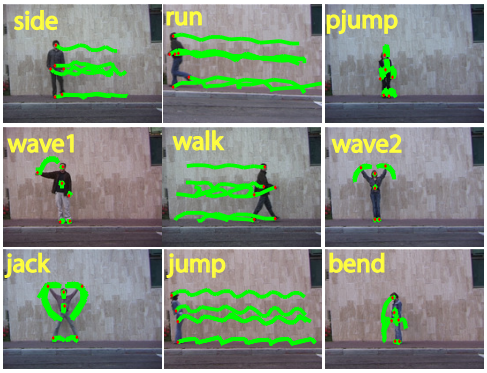


Figure 1. Nine different actions are used from the dataset provided by [12]. Trajectories from six landmarks (two hands, two feet, the head, and the body center) on human body are used as input to our method. These trajectories are used to extract invariant features of the reconstructed phase space that represent the underlying dynamical system.

Our contention in this paper is that by constraining the dynamical system to be of a particular type, one only *approximates* the true non-linear physics of human actions. In other words, by making assumptions about the type of the dynamical model, one tries to fit the experimental data to the model by finding values of the parameters that best explains the data. Rather than letting the data speak for itself about the type of the dynamical system, number of independent variables, degrees of freedom of the system, and values of unknown parameters. An analogous example of this type of approach from the field of probability theory is to assume the type of the probability distribution generating the data,

say Gaussian, and then computing the mean and variance of the Gaussian. Rather than allowing the data to determine the actual shape of the probability distribution using kernel density estimation.

The aim of this paper is to derive a representation of the dynamical system generating the human actions directly from the experimental data. This is achieved by proposing a computational framework that uses concepts from theory of chaotic systems to model and analyze nonlinear dynamics of human actions, by using trajectories of body joints. There are few important points to note here: First, by proposing dynamical system generating human actions as a chaotic system, we are making the statement that there is a *determinism* present in the seemingly stochastic dynamics of human actions. This *determinism*, if exploited, can be used to derive richer features for action recognition. Second, the proposed approach of modelling human actions directly from experimental data is superior to approximate modelling, since no assumptions have to be made about the type or form of the dynamical model.

2. Related Work

In general, approaches for human action and activity analysis can be categorized on the basis of the representations used by researchers. Some leading representations are: learned geometrical models of human body parts [19, 20], space-time pattern templates [12, 16]), 3D information ([17]), shape or form features [21, 13]), interest point based representation [23], motion/optical flow patterns [24, 18] and volumetric features [22, 15].

Our present work is more related to the approaches of learning dynamical models over the state space that represent human motion ([7, 10, 14]). Specifically, the method by Bissacco et. al. [7] used a parametric skeletal model of a moving person and learned a linear dynamical model, while Bregler [14] proposed a mixed-state statistical model with a finite state automaton at the highest level to switch between local linear models to cater for the nonlinear dynamics of human motion. Later on [8, 9] attempted to integrate the nonlinear dynamics directly into the model, rather than using an external mechanism to control the switching.

A common theme of all these approaches is that they approximate the true motion dynamics by putting constraints on the type of the dynamical model. In addition, they require very detailed mathematical and statistical modelling which involves assumptions about the probability distributions of stochastic variables of the model, development of inference methods, and algorithms for learning parameters of the distribution using a large data set. To overcome some of these difficulties, in this paper we are proposing a framework that captures the true non-linear dynamics of the human motion, and generates a more richer set of features by directly working with the experimental data. In addition, our method is not a statistical learning method therefore does not require large training data, instead strong discriminative features can be derived just from one example action.

3. Preliminaries

In this section we present the background material related to the theory of nonlinear dynamics and chaos. We believe that this quick overview will be helpful in understanding the rest of the paper. A dynamical system can be represented as a set of functions which describes how variables change in time. A dynamical system is termed nonlinear if the function defining the change in the system is nonlinear. A dynamical system may be stochastic or deterministic. In a stochastic dynamical system, new values are generated from a probability distribution, while in a deterministic dynamical system a single new value is associated with any current value.

Dynamical systems can be represented by state-space models, where state variables $X(t) = [x_1(t), x_2(t), ..., x_n(t)] \in R^n$ define the status of the system at a given time t. The state variables are often considered to be in subspaces of Euclidian spaces, but they more generally are in n-dimensional manifolds. The space of the state variables is often called the *phase space*. The state of the system evolves in accordance with a deterministic evolution function and the path traced by the systems states as they evolve over time is referred to as a *trajectory* or *orbit*. The collection of all trajectories from all possible starting points in the phase space of the dynamical system is called a *phase portrait*. An *attractor* is defined as the region of the phase space to which all the trajectories settle down to as time limit approaches infinity. If the attractor is not stable it is termed *strange*. The *invariants* of system's attractor are measures that quantify the properties that are invariant under smooth transformations of the phase space or control parameters. Invariants fall into three classes: 1) Metric 2) Dynamical and 3) Topological. Metric invariants include dimensions of different kind and multi-fractal scaling functions, while dynamical invariants include Lyapunov exponent. Topological invariants generally depend on the periodic orbits that exist in the strange attractor. *Embedding* is defined as a process of mapping one-dimensional signal to a m-dimensional signal.

Chaos theory is one of the ways to study nonlinear phenomena. The name 'Chaos Theory' comes from the fact that the systems the theory describes are apparently disordered, but theory is really about finding the underlying order in apparently random data. In other words, a chaotic system is a deterministic system which is globally stable, exhibit clear boundaries and displays sensitivity to the initial conditions. When applying chaos theory to a given a problem, the goal often is to extract information required to identify and classify strange attractors of the dynamical

system from the experimental data. The procedure can be broken down into a few relatively easy steps. These are: find a suitable embedding of the data, verify the existence of deterministic structure, compute dynamical, topological and metric invariants of the periodic orbits, and finally use the invariants for the identification purposes. The proposed framework for action recognition is built around these basic steps. Intuitively speaking, for a computer vision practitioner chaos theory provides a way of determining the description of a dynamical system from a time series data. As long as one has the time series data, analysis steps described above can be applied. Few examples of the time series data that we come across in the field of computer vision would be trajectories, pixel intensity over time, flow vectors etc.

4. Framework

This section describes the algorithmic steps of the proposed action recognition framework. These are: *i*) Given a video of an exemplar action, obtain trajectories of reference body joints, and break each trajectory into a time series by considering each data dimension separately; *ii*) obtain chaotic structure of each time series by embedding it in a phase space of an appropriate dimension using the mutual information [2], and false nearest neighborhood algorithms [5]; *iii*) apply determinism test to verify the existence of deterministic structure in the reconstructed phase space; *iv*) represent dynamical and metric structure of the reconstructed phase space in terms of the phase space invariants, and *v*) generate global feature vector of exemplar action by pooling invariants from all time series, and use it in a classification algorithm. Now, each step is explained in detail in the following subsections.

4.1. Action Representation

A trajectory corresponding to a body joint represents a deterministic nonlinear dynamical system. In our framework six body joints corresponding to two hands, two feet, head and belly are taken as the reference joints. To make the representation scale and translation invariant, trajectories of the first five joints are normalized with respect to the belly point. Hence, for any given action we use five trajectories to represent the action. We choose these reference joints as they provide sufficient information about most of the actions. Another consideration is that these joints are relatively easy to automatically detect and track in real videos, as opposed to the inner body joints which are more difficult to track. Figure 1 shows examples of set of trajectories for different actions in the case of real videos (2D trajectories), while Figure 2 shows trajectories for a running action from the motion capture data (3D trajectories). Note that, we are not solving the tracking problem in this paper, therefore, we assume that the tracks are available to us. Formally, we represent the normalized trajectory corresponding to a joint b as a sequence of locations $Z^b = [\mathbf{z}_1^b, \mathbf{z}_2^b, ..., \mathbf{z}_t^b]$, where $\mathbf{z} \in R^k$ with $k = 2$ for image based measurement, and $k = 3$ for the motion capture data. Finally, we have $k \times N_B$ scalar time series for each exemplar action, where N_B is the number of the reference joints.

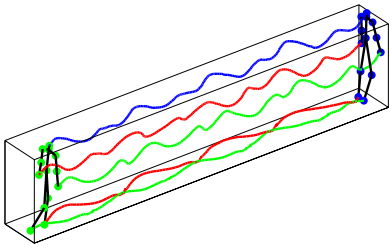


Figure 2. A sample set of 3-dimensional trajectories generated by head (blue), two hands (red & green) , and two feet (red & green) are shown for the running action from the motion capture data set. The stick figure with green landmarks depict the first frame, and the one with blue landmarks represents the last frame.

4.2. Embedding

Embedding, as defined earlier, is a mapping from one dimensional space to a m-dimensional space. It is an important part of study of chaotic systems, as it allows us to study the systems for which the state space variables and the governing differential equations are unknown. The underlying idea of embedding is that all the variables of a dynamical system influence one another. Thus, every subsequent point, z_i^b, of a given one dimensional time series results from an intricate combination of the influences of all other system variables. Therefore, $z_{i+\tau}^b$ can be considered as a second substitute system variable which carries information about the influence of all other variables during time interval τ. Using this reasoning one can introduce a series of substitute variables $z_{i+2\tau}, ..., z_{i+m\tau}$, and thus obtain the whole m-dimensional phase space, where substitute variables carry the same information as the original variables of the system [3].

Formally, the embedding is achieved by using theorem of Takens [1], which states that *a map exists between the original state space and a reconstructed state space*. The theorem assures that one does not have to measure all the true state space variables of the system, as in fact almost any one of the variables will be sufficient to reconstruct the dynamics. It also states that the dynamical properties of the system in the true state space are preserved under the embedding transformation. Thus, for a large enough embedding dimension m, the delay vectors $\mathbf{Y}^b(i) = [z_i^b, z_{i+\tau}^b, z_{i+2\tau}^b, ..., z_{i+(m-1)\tau}^b]$, generate a phase space that has exactly the same properties as that formed by the original variables of the system. Over here, $z_i^b, z_{i+\tau}^b, z_{i+2\tau}^b, ..., z_{i+(m-1)\tau}^b$ represent scalar time series,

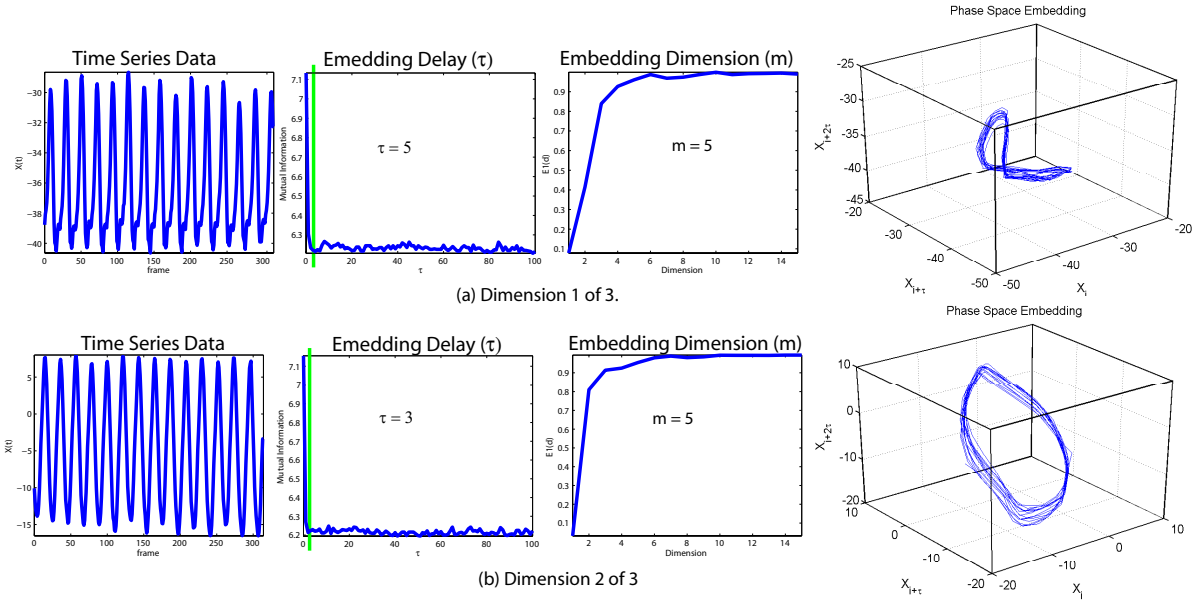


Figure 3. Depicts the embeddings of the time series corresponding to the right foot of the actor shown in Figure 2. The first column shows the time series corresponding to the x and y dimensions of the right-foot trajectory. The second column shows the plot of mutual information which is used to determine τ. The first minima value, marked by the green bar, reflects the optimal values of τ. The third column shows the plot of a measure $E1(d)$ [27], which can be derived from the false nearest neighbor algorithm, against different values of m. The value of m, after which the plot converges to a stable value, is chosen as the optimal embedding dimension. This happens to be at $m = 5$ in the current case. The fourth column shows the 3-dimensional projection of the reconstructed phase space for the chosen values of τ and m. This embedding is used to extract invariant features.

belonging to one dimension of the trajectory, of the body joint b at times $t = idt$ to $t = (i + (m-1)\tau)dt$. Here, τ is known as the embedding delay. However, the embedding theorem does not provide a method to find the optimal values of τ and m. For estimating these values, we use the mutual information [2] and the false nearest neighbor algorithms [4]. In order to make the paper self-contained and readable, we are re-stating these algorithms from [3].

4.2.1 Estimating Embedding Delay

The estimation of delay parameter is based on the idea, that the mutual information between z_i^b and $z_{i+\tau}^b$ can be used to estimate a proper embedding delay τ. The algorithm considers two criterion: First, the value of τ should be large enough so that value of z^b at time $i+\tau$ is measuring something significantly different from the value of z^b at time i, and thus providing us with a new information which we do not have up till now. Second, the value of τ should not be larger than the time in which system loses memory of its initial state. The algorithmic steps are:

1. From the given time series $z_1^b, z_2^b, \ldots, z_t^b$, compute z_{min} and z_{max}.
2. Compute absolute value of their difference, $d = |z_{min} - z_{max}|$, and partition d into j equally sized intervals.
3. Compute: $I(\tau) = -\sum_{h=1}^{j}\sum_{k=1}^{j} P_{h,k}(\tau)\ln\frac{P_{h,k}(\tau)}{P_h(\tau)P_k(\tau)}$, where P_h and P_k denote the probabilities that the variable assumes a value inside the hth and kth bin, and $P_{h,k}$ is the joint probability that z_i^b is in bin h and $z_{i+\tau}^b$ is in bin k.
4. Chose that τ as the embedding delay parameter for which $I(\tau)$ gives the first minima (Figure 3).

4.2.2 Estimating Embedding Dimension

For finding the optimal embedding dimension m we used the false nearest neighbor method proposed in [4]. The idea of the algorithm is to unfold the observed orbits from self overlap arising from the projection of an attractor of a dynamical system on a lower dimensional space. The algorithm makes use of the assumption that the phase space of a dynamical system folds and unfolds smoothly, and there are no sudden irregularities. This translates to the observation that if points are sufficiently close in a reconstructed phase space, then they should remain close during a forward iteration. If a phase space point has a neighbor that does not full fill this criteria then that point is said to have a false neighbor [3]. The steps for finding optimal m are:

1. Pick a point $p(i)$ in a m-dimensional space from the time series Z^b.
2. Find a neighbor $p(j)$ so that $\|p(i) - p(j)\| < \xi$.
3. Compute a normalized distance $R_i = \frac{|z^b_{i+m\tau} - z^b_{j+m\tau}|}{\|p(i)-p(j)\|}$, between $(m+1)$th coordinates of $p(i)$ and $p(j)$.
4. If R_i is larger then threshold R_{th}, then $p(i)$ is marked a having a false nearest neighbor.
5. Apply the equation in step 3 to entire time series for $m = 1, 2, \ldots$, until the fraction of points for which $R_i > R_{th}$is negligible.

Figure 3 pictorially shows the process of finding optimal τ and m for two time series. It also displays 3-dimensional mapping of the reconstructed phase spaces. Once the values of τ and m are known, we slide a window of length m through the time series, and stack the m dimensional vectors row-wise into a matrix

$$X^b = \begin{pmatrix} z^b_0 & z^b_\tau & . & . & z^b_{(m-1)\tau} \\ z^b_1 & z^b_{1+\tau} & . & . & z^b_{1+(m-1)\tau} \\ z^b_2 & z^b_{2+\tau} & . & . & z^b_{2+(m-1)\tau} \\ . & . & . & . & . \end{pmatrix}. \quad (1)$$

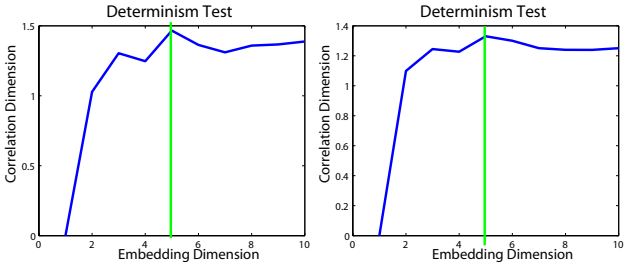
Note that each component of the m-dimensional vector is separated by an interval τ. Each row of the above matrix is now a point in the m-dimensional reconstructed phase space. We repeat the process for each time series, thus obtaining $k \times N_B$ reconstructed phase spaces for each action.

4.3. Determinism Test

The purpose of this test is to get the evidence in support of our assertion, that there is a structure present in the trajectory data that can be exploited to obtain the representation of the underlying dynamics of human actions. It is performed on each of reconstructed phase space to distinguish irregular behavior resulting from deterministic chaos and the one appearing due to the noise. For this purpose, we employ a determinism test proposed in [26], where the idea is that neighboring trajectories in a small portion of the reconstructed phase space should all point in the same direction, thus assure the uniqueness of solutions in the phase space which is a property of determinism. The outcome of this test (as shown in Figure 4) on our data validates the existence of determinism. That is, it reveals that the trajectories of the body joints indeed are generated by a deterministic process, and this justifies further analysis of the data by using the phase space invariants.

4.4. Invariant Features

Metric, dynamical and topological organization of orbits associated with a strange attractor of the reconstructed



(a) Dim 1, m = 5 (b) Dim 2, m = 5

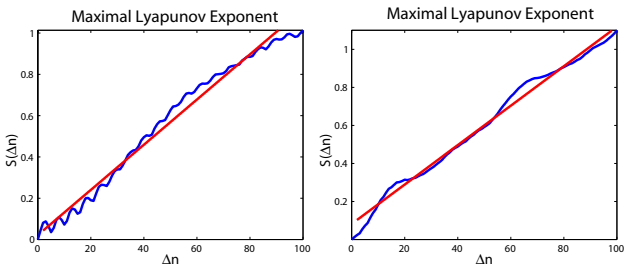
Figure 4. The determinism test is performed by checking the convergence of the correlation dimension for the embedding dimension larger than m. In the case of a stochastic system, the value of correlation dimension (y-axis) increases monotonically with the increasing embedding dimension (x-axis). We show that the data under consideration indeed converges to the value of correlation dimension at the computed values of m (the green line) for the two time series shown in Figure 3.

phase space can be used to distinguish different strange attractors representing different human actions. This organization is quantified in terms of phase space invariants. In this paper, we limit ourselves only to metric and dynamical invariants which include: i) Maximal Lyapunov Exponent, ii) Correlation Integral, iii) Correlation Dimension.

4.4.1 Maximal Lyapunov Exponent

Lyapunov exponent is a dynamical invariant of the attractor, and measures the exponential divergence of the nearby trajectories in the phase space. If the value of maximum Lyapunov exponent is greater than zero, that means the dynamics of underlying system are chaotic. In order to compute maximum Lyapunov exponent of reconstructed phase space, we employ algorithm given in [3]. The algorithm tests the exponential divergence of trajectories directly from the phase space trajectories.

To estimate the maximum divergence around a reference point $p(i)$ in the phase space, we start by finding all the



(a)Dim 1 (b) Dim 2

Figure 5. The computation of maximal Lyapunov exponent (for the right foot trajectory shown in Figure 2) from the plot of $S(\Delta n)$ against Δn. The slope of the line fitted to the curve provides a robust estimate of the maximal Lyapunov exponent. The estimated values here are 0.0104 for (a) and 0.0109 for (b).

neighbors $p(k)$ which are within distance ε. Here $p(i)$ is the ith row of the reconstructed phase space matrix X^b. The neighboring points are used as the starting point of nearby trajectories. The average distance of all the trajectories to the reference trajectory can be computed as a function of relative time Δn as follows:

$$D_i(\Delta n) = \frac{1}{r}\sum_{s=1}^{r} \mid z^b_{k+(m-1)\tau+\Delta n} - z^b_{i+(m-1)\tau+\Delta n} \mid, \tag{2}$$

where s counts the different points $p(k)$, and there are total of r such points. Finally, the average of the logarithm of $D_i(\Delta n)$ is obtained for several reference points to get the effective expansion rate. That is we compute $S(\Delta n) = \frac{1}{c}\sum_{i=1}^{c} ln(D_i(\Delta n))$, where c is the number of reference points over which the process is repeated. Values of $S(\Delta n)$, computed for different Δn, and the maximum Lyapunov exponent is taken as the slope of the line fitted to the graph of $S(\Delta n)$ against Δn. Figure 5 shows this graph for the two time series shown in Figure 3.

4.4.2 Correlation Integral

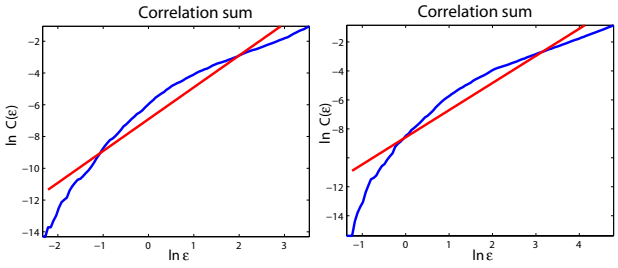
The correlation integral is a metric invariant, which characterizes the metric structure of the attractor by quantifying the density of points in the phase space. It achieves this through a normalized count of pair of points lying within a radius ϵ. Formally, correlation integral $C(\epsilon)$ is defined as:

$$C(\epsilon) = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} \Theta(\epsilon - \parallel x_i - x_j \parallel), \tag{3}$$

where Θ is the Heaviside function. Note that, x_i in this case refers to a point in the phase space i.e. it corresponds to ith row vector of X^b. In our experiments, we computed $C(\epsilon)$ for a fixed values of ϵ and used it as a feature vector. Figure 6 shows the plot of the correlation integral for increasing values of ϵ.

4.4.3 Correlation Dimension

The correlation dimension also characterizes the metric structure of the attractor. It measures the change in the density of phase space with respect to the neighborhood radius ϵ. The correlation dimension can be computed from the correlation integral by exploiting the power law relationship $C(\epsilon) \approx \epsilon^d$, where d is the correlation dimension. The computation of the correlation dimension proceeds by plotting $C(\epsilon)$ and ϵ on a log-log graph. Again, the slope of the linen fitted to this graph provides a robust estimate of correlation dimension ,because the region in which power law is obeyed appears as a straight line in the graph. Figure 6 shows this graph, along with the estimated values of the correlation dimensions for the two time series shown in Figure 3. The region whose slope is an estimate of the correlation dimension.



(a) Dim 1 Correlation Dim: 1.48 (b) Dim 2 Correlation Dim: 1.36

Figure 6. Computation of correlation dimension for the two time series shown in Figure 3. With increasing values of neighborhood radius ϵ (the horizontal axes), the values of the correlation integral (vertical axes) also increases. The slope of the line fitted to the curve provides an estimate of the correlation dimension.

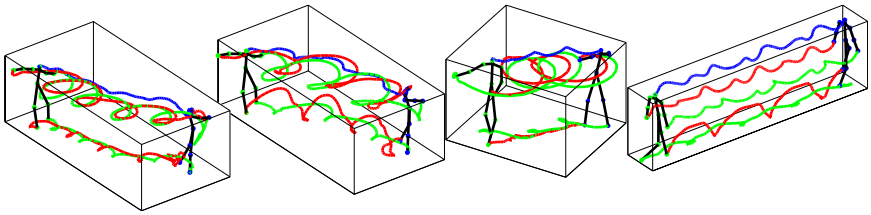
Another useful information about the action can be obtained from the variance of the time series data, which we employ as a part of the feature vector in addition to the phase space invariants.

5. Experiments

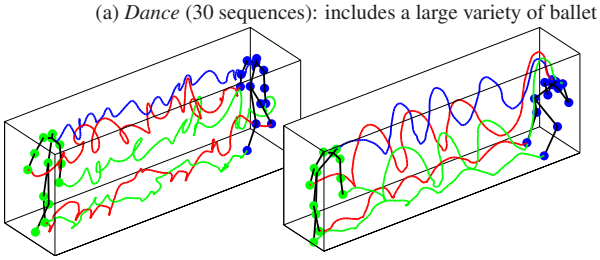
Experimental analysis is carried out on data sets provided by [12] (see Figure 1) and 3D motion capture dataset from [25] (see Figure 7).

5.1. Motion Capture Dataset

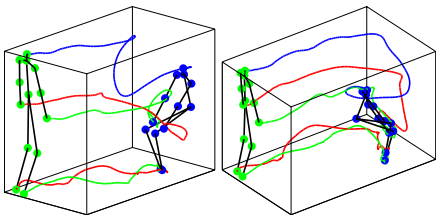
The first set of experiments was performed on the data set containing 3-dimensional motion capture sequences provided by FutureLight [25]. Figure 7 shows some typical sequences from this data set. In total, it contains 155 sequences of 5 action classes, namely *dance*, *jump*, *run*, *sit*, and *walk* with 30, 14, 30, 33, and 48 instances, respectively. All five classes have significant intra-class variations. For example, the *run* class has variations in terms of speed (jog, run), stride length (short, long), bounce (low, high), and arm swing (low, high). The sequences in the run class, therefore, are created by several combinations of these parameters, and also include stopping and turning events. Similarly, the *walk* class contains these variations, in addition to a parameter for the pelvic swing (high, low). There are other variations like walking in a circle, turning around, stopping etc. The *dance* class contains stationary and moving ballet sequences, and some cat-walk sequence, which in fact resembles closely to the *walk* sequences. The *jump* class contains jumping in place as well as jumping/hopping on one foot while walking. Finally, the *sit* class contains variations in the execution styles. In summary, all the action classes contains significant intra-class variations. and therefore, this is a very challenging data set.



(a) *Dance* (30 sequences): includes a large variety of ballet sequences. A subset of these is very similar to the *walk* class.



(b) *Jump* (14 sequences): mostly hopping and jumping while walking



(c)*Sit* (33 sequences): contains variations in sitting postures & directions

Figure 7. Sample sequences of few action classes from the motion capture data set. The stick figures with green joints depicts the first frame of the sequence, while the stick figure with blue joints represent the last frame.

The initial input is in the form of trajectories of 13 body joints of the stick figure shown in Figure 2, but we only use 5 reference joints. We extract scalar time series from all five reference joints, resulting in 3 time series (x,y, & z) per reference joint and 15 time series per action. Each time series is embedded separately using the procedure described in Section 4.2.2. A four dimensional feature vector is then constructed for each time series by computing Lyapunov exponent, correlation integral, correlation dimension and variance. After concatenation, for a given action sequence this results in a 60-dimensional feature vector. For testing, we use the leave-one-out cross validation approach using the K-nearest neighbor classifier with $K = 5$. The classification results achieved by this approach are shown in the Figure 8. We achieved mean accuracy of 89.7% on the entire data set. Four *run* sequences were misclassified as the *walk*, which is understandable considering the similarity between these actions. Another main source of error was the confusion between the walking ballet sequences from the *dance* class and the *walk* class.

	Dance	Jump	Run	Sit	Walk
Dance	28				2
Jump		13			1
Run	2	1	22	1	4
Sit				33	
Walk	3		2		43

Figure 8. Confusion table for the motion capture data set. We achieved mean classification accuracy of 89.7%.

5.2. Video Data Set

The second set of experiments was performed on the action data set [12], which depicts real actors performing different actions. Figure 1 shows some examples of these actions. Specifically, the data set contains 81 videos with 9 different actions performed by 9 different actors. Given the data, the first step in the algorithm is the extraction of joint tracks for the six landmarks on the human body (two hands, two feet, the head, & the belly point). We used a semi-supervised joint detection and tracking approach for this experiment. That is, for computing trajectories for the reference joints, we extracted body skeletons and their endpoints using by using morphological operations on foreground silhouettes of the actor. Then an initial set of trajectories is generated by joining extracted joint locations using the spatial and motion similarity constraint. The broken trajectories and wrong associations were corrected manually. Note that the quality of the phase space embedding is dependent on the length of a time series, which implies that we need to observe the target action for sufficiently long period of time (approximately 200 frames). However, the length of the videos in the data set varies from 27 to 80 frames. We overcame the problem by up-sampling and concatenating the original trajectories and thereby increasing the number of observations. Our experimental results have shown we are able to capture variations present in different actions by employing this approximation. Once the trajectories of five body joints relative to the centroid of foreground blob are recovered, we decomposed each of them into their two spatial components (x & y). This resulted in ten time series in total, which are then used to compute the invariants. After concatenating, for a given action this resulted in a 40-dimensional feature vector.

The testing was performed performed by using leave-one-out cross validation. When using K-nearest neighbor, one sequence is kept as a test sequence while all the remaining sequences were used as training samples. We obtained a mean classification accuracy of 92.6% for all nine actions.

	Bend	Jumping Jack	Jumping Forward	Jumping in Place	Run	Side Gallop	Walk	Wave1	Wave2
Bend	9								
Jumping Jack		9							
Jump Forward			5	2	2				
Jump in Place				9					
Run					8		1		
Side Gallop					1	8			
Walk							9		
Wave1								9	
Wave2									9

Figure 9. Confusion table for the action data set of [12], where our algorithm has achieved mean accuracy of 92.6%.

The confusion table is shown in Figure 9. It can be observed that only 6 out of a total of 81 videos were misclassified in these experiments. Two of the misclassified videos were from the *Jump Forward* action, which were incorrectly labelled as *Run* action. While two other videos were misclassified as *Jumping in Place*. The *Run* and *Side Gallop* action have one misclassification each. The observation we would like to make over here is that these are isolated errors, mostly for those actions which have quite a bit of similarity with each other, as is the case with when confusing running with walking, or jumping forward with running.

In order to test the robustness of our method with respect to the number of available joint tracks, we performed a second set of experiment by selecting only a subset of the 5 reference joints. In the first run, the head joint is removed from the list of reference point, and we achieved a mean accuracy of 81.2%. Most of the new errors were observed in bending and jumping actions. In the second run, we removed the left hand joint from the set and achieved an accuracy of 86.1%. We consider this a satisfactory performance, as we were able to maintain the action recognition accuracy up to a reasonable degree even if one of the reference time series is missing. This shows that the proposed approach is not very sensitive to occlusion of individual body joints. At the same time, we observed that the classification accuracy for actions that are heavily dependent on the removed body joint (e.g. head in the case of bending) suffers more. But for actions like walking and running that involve multiple joints (two feet & two hands), removing one of these joints does not severely effect the overall classification accuracy.

6. Conclusion

In this paper we introduced a framework which characterizes the nonlinear dynamics of human actions by using the theory of chaotic systems. Using this framework, we extracted a set dynamical and metric invariants of the strange attractor of the dynamical system, and used it for action recognition. Experimental validation of the feasibility and potential merits of carrying out action recognition using this framework is demonstrated on motion capture and real videos of human actions.

Acknowledgement: This research was funded by the U.S. Government VACE program.

References

[1] F. Taken, "Detecting Strange Attractors in Turbulence," Lecture Notes in Mathematics, ed D. A.Rand & L. S. Young, 1981.

[2] A. M. Fraser et. al., "Independent Coordinates for Strange Attractors from Mutual Information," Phys. Rev., 1986.

[3] M. Perc, "The Dynamics of Human Gait," European Journal of Physics, 26, 2005.

[4] M. B. Kennel et. al, "Determining Embedding Dimension for Phase Space Reconstruction using A Geometrical Construction," Phys. Rev. A, 45, 1992.

[5] M. T. Rosenstein et. al., "A Practical Method for Calculating Largest Lyapunov Exponents from Small datasets," Physica D, 65, 1993.

[6] L. Campbell et. al., "Recognition of Human Body Motion Using Phase Space Constraints," In Proc. CVPR, 1995.

[7] A. Bissacco et. al, "Recognition of Human Gaits,", IEEE CVPR, 2001.

[8] L. Ralaivola et. al., "Dynamical Modeling with Kernels for Nonlinear Time Series Prediction", NIPS, 2004.

[9] J. M. Wang et. al., "Gaussian Process Dynamical Models,", NIPS, 2005.

[10] B. North et. al., "Learning and classification of complex dynamics," In IEEE PAMI, 22(9), 2000.

[11] V. Pavlovic et. al., "Impact of Dynamic Model Learning on Classification of Human Motion," In CVPR, 2000.

[12] M. Blank et. al., "Actions as Space-Time Shapes", ICCV, 2005.

[13] A. Yilmaz et. al., "Actions Sketch: A Novel Action Representation", IEEE CVPR, 2005.

[14] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", IEEE CVPR , 1997.

[15] E. Shechtman et. al., "Space-Time Behavior Based Correlation", IEEE CVPR, 2005.

[16] A. F. Bobick et. al., "An Appearance-Based Representation of Action", IEEE CVPR, 1996.

[17] V. Parameswaran et. al., "Using 2D Project Invariance for Human Action Recognition", IJCV, 66(1), 2006.

[18] Y. Yacoob et. al., "Parameterized Modeling and Recognition of Activities", CVIU, 1999.

[19] G. Mori et. al., "Recovering Human Body Configurations: Combining Segmentation and Recognition", IEEE CVPR, 2004.

[20] H. Jiang et. al., "Successive Convex Matching for Action Detection", IEEE CVPR, 2006.

[21] K.M. Cheung et. al., "Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture", IEEE CVPR, 2003.

[22] T. S. Mahmood et. al., "Recognition Action Events from Multiple View Points", EventVideo01, 2001.

[23] I. Laptev et. al., "Space Time Interest Points", IEEE CVPR, 2003.

[24] A. A. Efros et. al., "Recognizing Action at a Distance", IEEE ICCV, 2003.

[25] FutureLight, R&D division of Santa Monica Studios.

[26] G. P. Williams, "Chaos Theory Tamed", pp 275-277.

[27] L. Cao, "Practical Method for Determining the Minimum Embedding Dimension of a Scalar Time Series", Physica D, 1997.