Moving Vistas: Exploiting Motion for Describing Scenes

Nitesh Shroff, Pavan Turaga and Rama Chellappa
Department of Electrical and Computer Engineering
Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742
{nshroff,pturaga,rama}@umiacs.umd.edu

Abstract

Scene recognition in an unconstrained setting is an open and challenging problem with wide applications. In this paper, we study the role of scene dynamics for improved representation of scenes. We subsequently propose dynamic attributes which can be augmented with spatial attributes of a scene for semantically meaningful categorization of dynamic scenes. We further explore accurate and generalizable computational models for characterizing the dynamics of unconstrained scenes. The large intra-class variation due to unconstrained settings and the complex underlying physics present challenging problems in modeling scene dynamics. Motivated by these factors, we propose using the theory of chaotic systems to capture dynamics. Due to the lack of a suitable dataset, we compiled a dataset of 'in-the-wild' dynamic scenes. Experimental results show that the proposed framework leads to the best classification rate among other well-known dynamic modeling techniques. We also show how these dynamic features provide a means to describe dynamic scenes with motion-attributes, which then leads to meaningful organization of the video data.

1. Introduction

Scene recognition is an area of active research in computer vision where the goal is to identify an image as belonging to one of several scene classes such as mountains, beaches, or indoor-office, etc. Recognition of scenes by humans is usually explained in one of two ways. The first approach suggests scene recognition proceeds in a hierarchical manner by processing and understanding the objects in the space and their contextual relationships [17]. This line of thought has been used extensively in computer vision applications such as in [27, 16, 12]. Alternately, in many cases humans also recognize scenes by understanding holistically the properties of a scene [2] instead of fine object level descriptions. This line of thought has also resulted in several features and algorithms for scene analysis [20, 11].

While there exists a rich literature for modeling static scenes, much less attention has been devoted to studying if and how motion in the scene can be exploited for describing
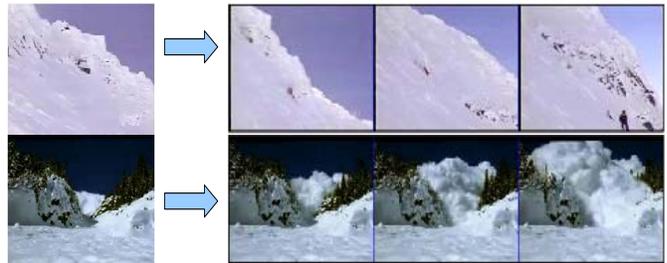

Figure 1. Consider the 2 frames on the left of the arrows. Both suggest a snow-clad mountain scene. But when temporal evolution of the scene is considered, as shown on the right, further information about 'avalanche' in the bottom video is revealed.

scenes. Motion information is useful both at the object-level and the global-level. In this paper, we investigate how the motion of the scene-elements themselves can provide more descriptive representations of videos than simply using their static appearance.

For instance, consider the 2 frames on the left of the arrow in figure 1. Both the frames suggest a snow clad mountain scene. But the sequence of frames (to the right of the arrows) gives further information that the bottom video is an avalanche scene. Here, the motion of the scene-elements (in this case, the snow) provides better fine-grained description of the scene. Similar examples can be imagined such as differentiating haphazard traffic from smooth traffic, a sea wave from a whirlpool, and numerous other examples. As these examples illustrate, a scene may be composed of the same elements in similar relative spatial positions, but the manner in which they move in relation to each other can be exploited to obtain a better, fine-grained distinction in scene categories.

Purely from a recognition perspective, a number of experimental studies have shown that motion information helps by enhancing the recovery of information about shape [30], edges [25], views [22] etc. In this paper, we propose to use scene-motion to *enhance the description of a scene*, and not merely as a method to improve static scene recognition. We show via examples and experiments in this paper that motion indeed provides informative cues about the scene. We demonstrate these ideas via recognition ex-

periments, and labeling of videos with semantically meaningful attributes.

Related Work There has been significant interest in recent years to recognize scenes. Towards this direction, several approaches have emphasized utilizing the presence of objects to recognize the scene. The co-occurrence of objects and actions with scene classes [16, 18] has been utilized to recognize the scenes. In a similar approach, relationships among locations of observed objects are modeled by prior distributions [27].

A parallel theme has been to use global cues like color histogram, power spectrum, global frequency with local spatial constraints [11, 28, 31, 20]. Local features have also been used to build a global descriptor of the image [9, 15]. The key idea in using these global cues is that a holistic approach, without analysis of its constituent objects, provides effective cues about its semantic category. Oliva and Torralba further incorporated the idea of using attribute-based descriptions of scenes [20]. Recently, both local and global discriminative information was fused together [23] to recognize indoor scenes.

These approaches have focused mainly on analyzing scenes from a static image. But none of these approaches have exploited motion for analyzing these scenes. Motion information, as we discussed, provides crucial cues about the dynamics of the scenes. Several methods to model dynamics have been considered in vision literature - some well-known [6, 19], some not so well-known [21]. Here, we briefly review them.

Doretto *et al*. [6] proposed linear dynamical systems (LDS) as models for dynamic textures. The LDS model is shown to be well-suited to capture second-order stationary stochastic processes and an effective generative model. This basic LDS model and its proposed extensions [3, 33] have been used for segmentation and recognition of dynamic textures and human activity recognition [32]. However, the first-order Markov property and linearity assumption in the model make it restrictive for modeling unconstrained dynamic scenes. We will show in our experiments that this is indeed the case.

The patch based bag-of-words approach has been used to model objects [26], scenes [9, 15], and human actions [5, 19] from videos. This approach uses local patches to build global representations. Due to its local nature, it can robustly handle dynamic backgrounds and moving cameras. When applied to scenes, this formalism does not distinguish scenes that locally may look similar, but globally appear different, e.g. consider avalanche vs. iceberg collapse.

Overview of our Approach In this work, we focus on ways to model the dynamics of the scenes and how this information may be utilized to describe scenes better. Some challenges that make application of standard methods difficult are: a) Scenes are unconstrained and 'in-the-wild', and b) the underlying physics of motion is either too complicated or very little is understood of them. Thus, this renders many of the standard methods of dynamic modeling unsuitable for this task. Consider, for example the last row in figure 2, which shows 6 different examples of videos from the same class 'Tornado'. The large variation in the appearance including scale, view, illumination, background can be easily perceived.

Motivated by these challenges, we model dynamic scenes using a chaos-theoretic framework. Chaos theory is concerned with understanding dynamical systems whose structure is unknown. Further, in such systems small changes in initial conditions result in huge variations in the output. But, the underlying process is not entirely random – infact, there exists a deterministic component which cannot be characterized in a closed form. These properties make this framework attractive in the study of in-the-wild dynamic scenes. This framework has been applied to gait modeling by Perc [21] and human action recognition by Ali *et al*. [1]. Apart from these applications, this framework has not been adopted widely in computer vision literature. This is probably because most problems in computer vision involving motion such as structure-from-motion, human actions, dynamic textures are usually constrained in nature, for which simple models such as second-order Gauss-Markov, or a constant-velocity model suffice.

Contributions We study the role of dynamics of scene elements for improved representation of scenes. Then, we address the problem of finding accurate and generalizable ways for characterizing the hard-to-model dynamics of unconstrained scenes via chaotic systems. We show how these dynamic features lead to better recognition of the videos using a dataset of 'in-the-wild' dynamic scenes. Finally, we introduce dynamic attributes and show how they lead to meaningful organization of scenes.

Outline Section 2 discusses the role that motion information and its attributes can play in scene representation. In section 3, we propose a framework based on chaotic systems to characterize unconstrained scene dynamics. In section 4, we discuss the dynamic scene dataset and provide extensive experimental results.

2. Motion Attributes

Without taking recourse to object-level descriptions, motion information of a scene can be described from a global perspective by attributes such as

Degree of Busyness: Dynamic scenes can be characterized by the amount of activity happening in the video. Consider scenes such as sea-waves or a traffic scene. These scenes are cluttered with a high degree of detailed motion patterns. On the other hand, scenes such as a waterfall appear largely unchanging and the motion typically occurs in a small portion of the scene.

Degree of Flow Granularity: Scenes that contain motion can also be distinguished based on the granularity of the scene's structural elements that undergo motion. Structural elements such as the falling rocks in a landslide are coarse in granularity, whereas the waves in an ocean are of a fine granularity.

Degree of Regularity: This attribute characterizes the quality of motion in terms of its regularity or irregularity. Even though traffic scenes may consist of the same level of busyness and granularity, chaotic traffic exhibits a high degree of irregular or random motion, whereas smooth traffic scenes exhibit a regular motion pattern.

As these examples illustrate, motion information can be exploited significantly to augment the spatial descriptions of scenes. Later in experiments, we will show how these motion-attributes can be learnt using dynamic features to provide semantically meaningful organization of videos.

3. Modeling Dynamic Scenes

Most models describing the dynamics of a system, assume that there exists an underlying mapping function f that synthesizes the next observation from the current observation, i.e. $y(t) = f(y(t-1)) + n(t)$. The structure of the function f encodes the dynamics of the underlying system. By making assumptions on the structure such as in the case of LDS [6], model fitting can be efficiently solved. However, in our application, the 'wild' and uncontrolled setting does not lend itself to making simplifying assumptions about the system. However, even without making any specific assumptions on the function f, using the theory of chaotic systems, it is possible to derive certain 'invariants' of f purely from the sequence of observations $y(t)$. We next describe the chaotic system framework that provides a solution to this problem.

3.1. Chaotic Invariants

Fundamental to chaos theory lies the fact that all variables in a deterministic dynamical system influence one another and thus are generically connected. Thus, every subsequent point of a given measurement is the result of an entangled combination of influences from all other system variables [21]. Due to space constraints, we briefly mention the steps below and refer the reader to [21, 1] for further details.

Consider a one-dimensional time series $\{x_t\}_{t=1}^n$. The first step is the reconstruction of attractor dynamics from the given time series. An *attractor* of a dynamical system is defined as the region of phase space that over the course of time (iterations), attract all trajectories emanating from some range of starting conditions [34]. The reconstruction is achieved by constructing the set of vectors that result from the concatenation of m time delayed version of x_t i.e., the vector $[x_1, x_{\tau+1}, ..., x_{(m-1)\tau+1}]$. These vectors are called the state-variables. The space of all state-variables is called the phase space. Here, m is the embedding dimension and τ is the embedding delay.

The embedding delay τ is calculated from the given time series using the mutual information [10]. For this, the range of the given values $[min(x_t), max(x_t)]$ is first divided into equal bins.

$$I(\tau) = -\sum_{s=1}^b \sum_{q=1}^b P_{s,q}(\tau) log \frac{P_{s,q}(\tau)}{P_s(\tau)P_q(\tau)} \quad (1)$$

where P_s and P_q denote the probabilities that the variable x_t assumes a value inside the sth and q^{th} bin, and $P_{s,q}$ is the joint probability that x_t is in bin s and $x_{t+\tau}$ in bin q. Then τ is chosen as the first local minima of $I(\tau)$. Subsequent to this, the optimal embedding dimension m is calculated using the false nearest neighbor algorithm [24]. This algorithm assumes that the phase space of a deterministic system folds and unfolds smoothly with no sudden irregularities appearing in its structure [21]. Given these values of τ and m for a time series, we form the matrix with the ith row representing i^{th} phase space vector $p(i)$. This set of m-dimensional points $p(i)$ is the reconstructed phase space.

$$X = \begin{pmatrix} x_1 & x_{\tau+1} & \cdots & x_{(m-1)\tau+1} \\ x_2 & x_{\tau+2} & \cdots & x_{(m-1)\tau+2} \\ \vdots & \vdots & \ddots & \end{pmatrix} \quad (2)$$

This embedding into m-dimensional phase space recreates the dynamics of the system. This is then represented using the metric and dynamical *invariants* of the system – the Lyapunov exponent, correlation integrals and correlation dimension. These *invariants* of a system's attractor quantify the properties that are invariant under smooth transformations of the phase space.

The Lyapunov exponent characterizes the level of chaos in the system. It describes the dynamics of a trajectory evolution by characterizing the average rate of convergence or divergence of two neighboring trajectories in the phase space. Positive exponents mean that the trajectories are diverging and hence is a representative of the chaotic dynamics. It quantifies the sensitive dependence on initial conditions by showing the average rate at which two close points separate with time. To calculate the Lyapunov exponent, we first estimate the maximum divergence around an arbitrarily chosen reference point $p(i)$. We then determine all the points $p(k)$ which are within ϵ distance of $p(i)$. These neighboring points are used as the starting point of nearby trajectories. Then the average distance of all these trajectories to the reference trajectory is computed as a function of relative time Δn.

$$D_i(\Delta n) = \frac{1}{r} \sum_{s=1}^r |x_{k+(m-1)\tau+\Delta n} - x_{i+(m-1)\tau+\Delta n}| \quad (3)$$

Figure 2. Dynamic Scene Dataset consisting of 13 classes with 10 videos per class. Top 2 rows show an example from 12 out of the 13 classes in the order Avalanche, Iceberg Collapse, Landslide, Volcano eruption, Chaotic traffic, Smooth traffic, Forest fire, Waterfall, Boiling water, Fountain, Waves and Whirlpool. The bottom row shows frames from 6 videos of the 13th class Tornado. Notice the large intra-class variation in the dataset. Large variations in the background, illumination, scale and view can be easily seen. Similar variations are present in each of the classes.

where s counts the number of neighboring points of $p(i)$. There are r such neighboring points that fulfill $||p(i) - p(k)|| < \epsilon$. This $ln(D_i(\Delta n))$ is then averaged over c such arbitrarily chosen reference points.

$$S(\Delta n) = \frac{1}{c} \sum_{i=1}^{c} ln(D_i(\Delta n)) \quad (4)$$

The maximal Lyapunov exponent is then calculated as the slope of this graph $S(\Delta n)$ versus (Δn).

Correlation integrals and correlation dimension, on the other hand, give an estimate of the system complexity. Correlation integral $C(\epsilon)$ is a metric invariant which characterizes the density of points in the phase space. This is done by calculating the fraction of pairs of points in the reconstructed phase space that are in the ϵ neighborhood of each other.

$$C(\epsilon) = \frac{2}{N(N-1)} \sum_{s=1}^{N} \sum_{t=s+1}^{N} H(\epsilon - ||p(t) - p(s)||) \quad (5)$$

where H is the Heaviside function. Correlation dimension is then calculated as the slope of the regression of $log\, C(\epsilon)$ versus $log\, \epsilon$. As we show in the experiments, these invariants serve as strong discriminative features. It is important to note that these descriptors of the dynamics has been extracted directly from the given time series without making specific assumptions about the system.

Implementation Details: In actual experiments, we first extract the 960-dimensional Gist descriptor per video-frame. Each of the dimensions in the sequence thus obtained is considered to be an individual time-series. For each of these time-series, the chaotic invariants are computed. These chaotic invariants include Lyapunov exponent, correlation dimension and correlation integral $C(\epsilon)$ for 8 values of ϵ. These forms a 10 dimensional vector for each time series and hence provides a 9600 dimensional vectorial representation of each video.

4. Experiments

We now demonstrate through experiments how capturing scene motion can be effectively used to improve dynamic scene recognition. We also show how videos can be labeled with motion-attributes derived from the scene-dynamics. First, we briefly describe the dataset that is used.

4.1. Dataset

Due to the lack of a suitable dataset, we compiled a dataset of 13 classes with 10 videos per class. Each of these videos has been downloaded from video hosting websites like 'Youtube'. As there was no control over the video capturing process, the dataset has large variations in terms of illumination, rate, view and scale. Also, there is a variation in resolution and camera dynamics. These variations along with the underlying physics of the process have ensured that the intra-class variation is very high. Various classes in this dataset are: Avalanche, Boiling Water, Chaotic Traffic, Forest Fire, Fountain, Iceberg Collapse, Landslide, Smooth Traffic, Tornado, Volcanic Eruption, Waterfall, Waves and Whirlpool. Figure 2 shows an example frame from one randomly chosen video of each class. The last row in this figure shows 6 examples from the 13th class 'Tornado'. Notice the large intra-class variation in the classes. This dataset can be downloaded from the project page http://www.umiacs.umd.edu/users/nshroff/DynamicScene.html

Before we proceed with the experiments, we note that pure static appearance based classification on randomly chosen frames from these videos will cause confusion between classes such as {'chaotic traffic' and 'smooth traffic'}, {'avalanche' and 'iceberg'},{'waterfall' and 'fountain'},{'landslide' and 'volcanic eruption'},{'whirlpool' and 'waves'}. Further, using only the dynamics of the scene will cause a confusion between classes like {'avalanche','landslide','volcanic eruptions'},{'tornado' and 'whirlpool'},{'boiling water' and 'forest fire'}.

	Avalanche	Boiling Water	Chaotic Traffic	Forest Fire	Fountain	Iceberg	Landslide	Smooth Traffic	Tornado	Volcano	Waterfall	Waves	Whirlpool
Avalanche	50	0	0	0	10	0	10	0	0	10	10	0	10
Boiling Water	0	30	0	10	20	0	30	0	0	0	0	10	0
Chaotic Traffic	0	0	30	0	0	0	40	20	0	0	0	0	10
Forest Fire	0	10	10	40	20	0	20	0	0	0	0	0	0
Fountain	0	0	0	0	50	0	30	20	0	0	0	0	0
Iceberg	10	0	0	0	0	40	0	0	10	10	0	10	20
Landslide	20	0	10	10	10	0	20	0	0	0	10	10	10
Smooth Traffic	0	0	20	0	10	0	20	40	0	10	0	0	0
Tornado	0	0	0	0	10	0	0	10	70	10	0	0	0
Volcano	10	0	0	0	0	0	10	20	30	0	10	0	20
Waterfall	10	0	10	10	10	0	30	20	0	0	10	0	0
Waves	0	10	0	10	0	0	0	0	0	0	0	70	10
Whirlpool	20	0	10	0	10	0	0	0	10	0	0	10	40

(a) Static

	Avalanche	Boiling Water	Chaotic Traffic	Forest Fire	Fountain	Iceberg	Landslide	Smooth Traffic	Tornado	Volcano	Waterfall	Waves	Whirlpool
Avalanche	30	0	20	0	0	0	10	0	0	0	10	10	10
Boiling Water	10	30	0	10	10	0	0	0	10	10	0	20	0
Chaotic Traffic	0	0	50	10	0	0	10	20	0	0	0	0	10
Forest Fire	0	0	10	30	50	0	0	0	0	0	0	0	10
Fountain	0	10	0	10	20	0	10	0	10	0	30	0	10
Iceberg	10	0	10	0	0	10	10	0	10	20	0	20	10
Landslide	10	0	40	0	0	20	10	0	0	0	0	0	20
Smooth Traffic	0	0	60	0	0	0	10	20	0	10	0	0	0
Tornado	0	0	10	0	0	0	10	0	60	10	10	0	0
Volcano	0	0	20	0	0	0	0	0	0	70	10	0	0
Waterfall	0	10	0	0	10	0	0	0	0	30	30	20	0
Waves	0	0	0	0	0	0	10	0	0	0	10	80	0
Whirlpool	0	0	10	0	0	0	30	10	0	0	10	10	30

(b) Dynamics

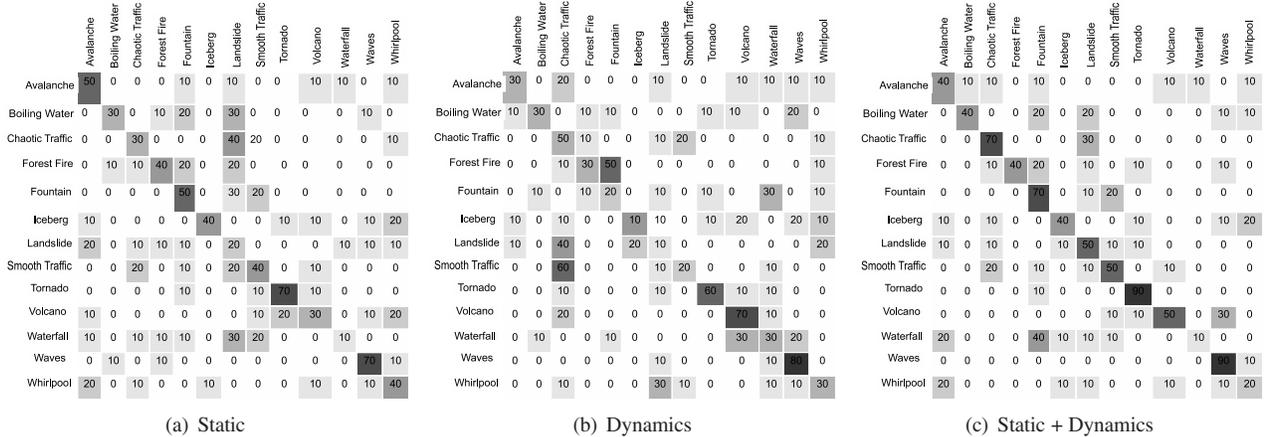|	Avalanche	Boiling Water	Chaotic Traffic	Forest Fire	Fountain	Iceberg	Landslide	Smooth Traffic	Tornado	Volcano	Waterfall	Waves	Whirlpool
Avalanche	40	10	10	0	10	0	0	0	0	0	10	10	10
Boiling Water	0	40	0	0	20	0	20	0	0	0	0	10	10
Chaotic Traffic	0	0	70	0	0	0	30	0	0	0	0	0	0
Forest Fire	0	0	10	40	20	0	10	0	10	0	0	10	0
Fountain	0	0	0	0	70	0	10	20	0	0	0	0	0
Iceberg	10	0	10	0	0	40	0	0	10	0	0	10	20
Landslide	10	0	10	0	0	10	50	10	10	0	0	0	0
Smooth Traffic	0	0	20	0	10	0	10	50	0	10	0	0	0
Tornado	0	0	0	0	10	0	0	0	90	0	0	0	0
Volcano	0	0	0	0	0	0	0	10	10	50	0	30	0
Waterfall	20	0	0	0	40	10	10	10	0	0	10	0	0
Waves	0	0	0	0	0	0	0	0	0	0	0	90	10
Whirlpool	20	0	0	0	0	10	10	0	0	10	0	10	20

(c) Static + Dynamics

Figure 3. Confusion Tables for the three top performing algorithms. Rows represent the ground truth label and columns represent the predicted label for the given video.

4.2. Recognizing Dynamic Scenes in the Wild

In this section, we present experiments for dynamic scene recognition performed on the dataset described in section 4.1. We compare the performance of the chaos framework with the well-known methods – LDS model and a bag-of-words (BoW) model. We restrict ourselves to these methods, as they are also 'global' in nature and do not require segmentation of the scene into objects. We first compare the performance using a Nearest Neighbors (NN) classifier. This serves as a transparent method for comparing various algorithms without any impact of hidden tunable parameters such as in the case of Support Vector Machines (SVMs). Next, we show the improvement obtained by using SVMs. In the following, we describe the implementation details of the other methods against which we compare.

Linear Dynamic Systems: [6] is a parametric model for spatio-temporal data and can be represented by:

$$x(t + 1) = Ax(t) + w(t) \;\; w(t) \sim N(0, R) \qquad (6)$$
$$z(t) = Cx(t) + v(t) \;\; v(t) \sim N(0, Q) \qquad (7)$$

where $x(t)$ is the hidden state vector, $z(t)$ is the observation vector, $w(t)$ and $v(t)$ are noise components and modeled as normal with 0 mean and covariance R and Q respectively. Here, A is state-transition matrix and C is the observation matrix. Let $[z(1), z(2), ...z(\tau)] = U\Sigma V^T$ be the singular value decomposition of the data matrix for τ observations. Then the model parameters are calculated [6] as $\hat{C} = U$ and $\hat{A} = \Sigma V^T D_1 V (V^T D_2 V)^{-1}\Sigma^{-1}$, where $D_1 = [0\ 0;\ I_{\tau-1}\ 0]$ and $D_2 = [I_{\tau-1}\ 0;\ 0\ 0]$. The distance metric used was based on subspace angles [4]. As a baseline, we first use the per frame intensity image as the feature vector. This gives a low performance – 13% – due to the large unconstrained photometric variations. Therefore, to better understand the performance of this model, we also use global scene features that are relatively invariant to photometric changes. In this case, we used the GIST

[20] [1] descriptor extracted from each frame of the video as the observation vector and use it to learn the LDS model.

Bag-of-words: In this formalism, a video is represented as a vector in an N-dimensional Euclidean space by using a N-bin histogram of visual words. These words are learnt by clustering descriptors extracted around interest points [5] [2] from the video. These local patches are then used to build a global representation of the video. These patches were extracted for each video with the detection parameters set to $\sigma = 2$ and $\tau = 2.5$. We then build our codebook by clustering these flattened cuboids of intensity values into $N = 1000$ words. Each video was then represented using the histogram of these words normalized by the total number of words in the video. Variations in representing the cuboids using the covariance features [29] and gradients were also tried. Best results were obtained with intensity-based descriptor and have been shown in table 1.

Mean Gist: We also use the mean GIST computed from the sequence of images as a global spatial descriptor of the video. This provides a 960 dimensional representation per video. The mean GIST can be seen as a simple fusion of frame-wise static representations of the video.

Chaotic Invariants: As mentioned in section 3.1, each video was represented using a 9600 dimensional vector. This was obtained [3] by concatenating Lyapunov exponent, correlation dimension and correlation integral for 8 values of ϵ for each dimension of the GIST vector. In our implementation, the values of epsilon used were: $\{0.085, 0.05, 0.03, 0.025, 0.01, 0.008, 0.007, 0.005, 0.003, 0.002, 0.001\}$.

Spatio-Temporal Fusion: Chaotic invariants by themselves provide only the dynamical invariants, but do not encode the spatial appearance. Hence, we fuse the global spa-

---

[1]Code available at http://people.csail.mit.edu/torralba/code/spatialenvelope/
[2]Code available at http://vision.ucsd.edu/~pdollar
[3]Code available at http://www.physik3.gwdg.de/tstool/HTML/index.html, http://www.mpipks-dresden.mpg.de/~tisean/TISEAN_2.1/docs/indexf.html

tial information provided by mean GIST and the dynamic information provided by the chaotic invariants into a single feature vector. This provides a simple way for fusing the spatial and temporal cues. The distance metric then uses a weighted combination of the static and global distances.

We perform a leave-one-video-out recognition experiment to compare the performance of various algorithms discussed above and the results are shown in table 1. We can see that modeling dynamics using just the chaotic invariants leads to a reasonable performance. We obtain comparable performance using the mean GIST. However, the best performance is obtained by fusing the chaotic invariants and mean Gist. We see that LDS shows poor performance compared to chaotic invariants. This is because the latter makes no assumption about the model and just uses the observed data to obtain invariants. Further, since scenes have valuable spatial information, using just the dynamic cues would not suffice. Therefore, augmenting the static features with the dynamic features leads to significantly improved classification rates. Confusion tables for the three best performing algorithms are shown in figure 3.

Class	LDS (GIST)	Bag of Words	Mean (GIST)	Dynamics (Chaos)	Static + Dynamics
Tornado	70	10	70	60	90
Waves	40	50	70	80	90
Chaotic Traffic	10	20	30	50	70
Fountain	0	10	50	20	70
Iceberg Collapse	20	30	40	10	50
Landslide	20	40	20	10	50
Smooth Traffic	10	0	40	20	50
Volcanic Eruption	0	30	30	70	50
Avalanche	70	30	50	30	40
Boiling Water	70	0	30	30	40
Forest Fire	0	30	40	30	40
Whirlpool	20	30	40	30	40
Waterfall	0	30	10	30	10
Overall	25%	24%	40%	36%	**52%**

Table 1. Comparison of classification rates of different methods using the dynamic scene dataset. The recognition accuracy is improved significantly by using the fusion of global static features (Mean GIST) and dynamic features. It is interesting to note when modeling dynamics with simplified models (LDS, Bag-of-words) the performance is worse than the static features.

Improved Classification In this experiment, we study the effect of better classifiers on the recognition performance of dynamic scenes. For the best performing algorithm (static + dynamics), we perform the same experiment with a linear SVM. Figure 4 shows the comparative performance of this classifier with the nearest neighbor (NN) classifier. As seen, the performance improves from overall recognition of 52% (NN) to 58% (linear SVM with $C = 1000$). This suggests that better classifiers can be designed to improve the performance further, but this is not the primary goal of this paper.

4.3. Attribute-based semantic organization

Attribute-based descriptions have been shown to possess useful properties such as, a) describing unknown classes, b) learning models of object categories from textual descrip-
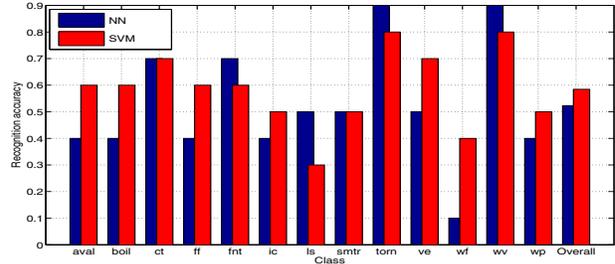

Figure 4. Performance comparison with two classifiers Nearest Neighbor(NN) and SVM. Shown here is the classification rates for each of the individual classes. Red bars (SVM) show an improved classification in most of the classes. The last column shows the overall performance which is improved from 52% (NN) to 58% (SVM). Figure best viewed in color.

tions, and c) detecting unusual attributes of known classes. This has been demonstrated for face recognition [13], object recognition [7, 14] and static-scene recognition [20]. Here, we show how the motion-attributes discussed in section 2 can be used to provide semantically meaningful organization of videos.

Each of these dynamic attributes – Busyness, Granularity, and Regularity – can be estimated from video features using regression techniques. Although any of the standard regression techniques can be used, in our experiments we use a linear regression model. Given a vector representation x of a video, the attribute value v is given by

$$v = x^T w + n \quad (8)$$

where w is the parameter vector. This parameter vector is then learnt by minimizing the mean squared error.

$$\hat{w} = X^+ v \quad (9)$$

where v is the vector of attribute values for the training videos. X is the matrix with its rows as the feature vectors of the training videos and X^+ is its pseudo-inverse. Training videos are given attribute labels from -1 (lowest) to 1 (highest). The vector v thus obtained is then used to learn the parameter vector $\hat{w}$. This is then used for predicting the attribute value $\hat{v}$ of a new video. The predicted values were then divided into 5 equal bins. In figure 5(a), randomly chosen videos from each of the bins are shown. The attribute value v increases as we go down the axis. First row is a video of 'waterfall' which has only a small portion of the image in motion while the bottom row ('waves') which has large motion almost all over the image is the most busy.

A similar procedure is used for granularity and regularity attribute. Figure 5(b) shows similar organization over the granularity axis. The top row 'chaotic traffic' has moving structural elements that are very coarse (vehicles) whereas the bottom row 'forest fire' has very fine moving structural elements (fire). Similarly, figure 6 shows the organization on the regularity axis. The top row shows an 'iceberg collapse' which has a very regular gravity-driven motion in the vertical direction while the video in the bottom row
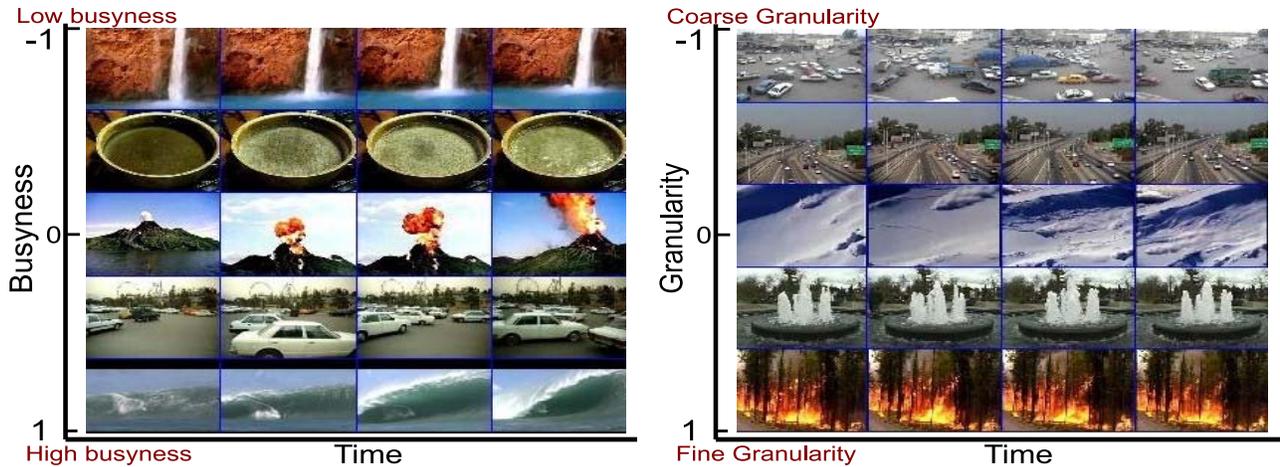
Figure 5. Organization of dynamic scenes by the degree of (a) busyness (left) and (b) granularity (right). Videos were divided into 5 equal bins and a randomly chosen video from each bin has been shown. x-axis shows the evolution over time. (Left) Top row is a 'waterfall' video where the motion is not-so-busy and is confined spatially to a small part of the video. Highly busy bottom row is a 'waves' video where the motion is present over almost all of the frame. (Right) Top row with coarse granularity video corresponds to the 'chaotic traffic' class whose moving elements are coarse (vehicles). While the bottom row is a video from 'forest fire' which has a very fine granularity (fire).

'tornado' exhibits motion that appears highly random, thus scores low on the 'regularity' axis.

We now present experiments that illustrate how scenes that share similar spatial attributes can be distinguished based on their dynamic attributes. We show that spatial attributes augmented with the dynamic attributes lead to a much better separation of scenes. Consider the 2 classes 'waves' and 'whirlpool', which share very similar spatial attributes like 'naturalness','roughness' [20] etc. But when the dynamic attributes 'regularity' and 'busyness' are considered, a clear separation can be seen in figure 7(a). Simple linear and quadratic classifiers easily separates the 2 classes. Similar separation can be seen for the case of 'chaotic traffic' and 'smooth traffic' in figure 7(b) and for 'avalanche' and 'iceberg' in figure 8.
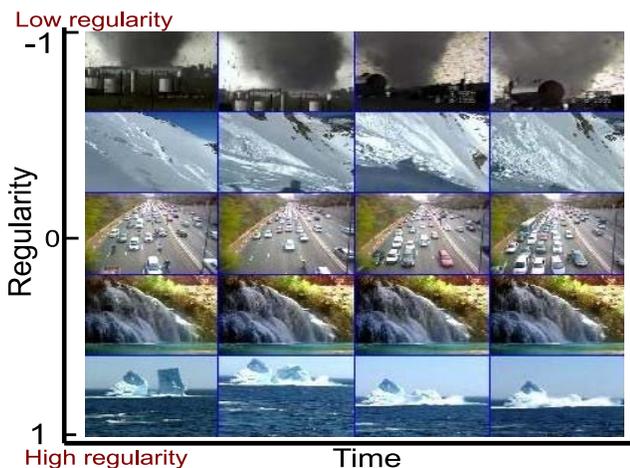

Figure 6. Organization of dynamic scenes according to the degree of regularity. Top row with very irregular and random motion is a video from the 'tornado' class. The bottom row corresponds to a video of 'iceberg' class with regular and constrained motion of an iceberg collapsing in the vertical direction under the effect of gravity.
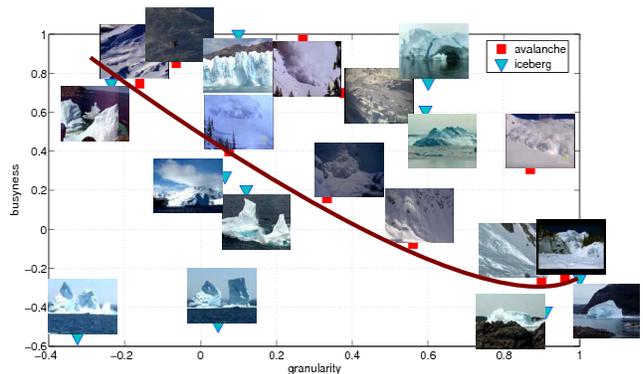

Figure 8. Separation of the classes 'avalanche' and 'iceberg'. These 2 categories have very similar spatial attributes and are difficult to separate using them. But a clear separation (red line) can be seen between the 2 classes on the 2 temporal attributes 'granularity' and 'busyness'. It should be noted that the red line has been drawn for the sake of visual illustration. A simple quadratic classifier easily separates the 2 classes. Note that 3 videos from 'iceberg' class, although with low granularity, end up falling in the wrong side because of being highly busy.

5. Conclusion and Discussion

In this paper, we studied the problem of categorizing 'in-the-wild' dynamic scenes. We showed that capturing dynamics within scenes leads to better representation of scenes. Further, we showed that using motion attributes leads to a meaningful organization of the videos. State-of-the-art algorithms to capture dynamics seem to perform poorly. We propose a method based on chaotic systems that outperforms other standard techniques, but overall performance is reminiscent of the early attempts at unconstrained object recognition [8]. Results show significant improvement by fusing static and dynamic features. Experimental results are promising. But, the categorization of dynamic scenes in the unconstrained setting poses a very challenging task due to its large intra-class variations and hence requires

(a) Waves vs. Whirlpool   (b) Chaotic vs. Smooth Traffic
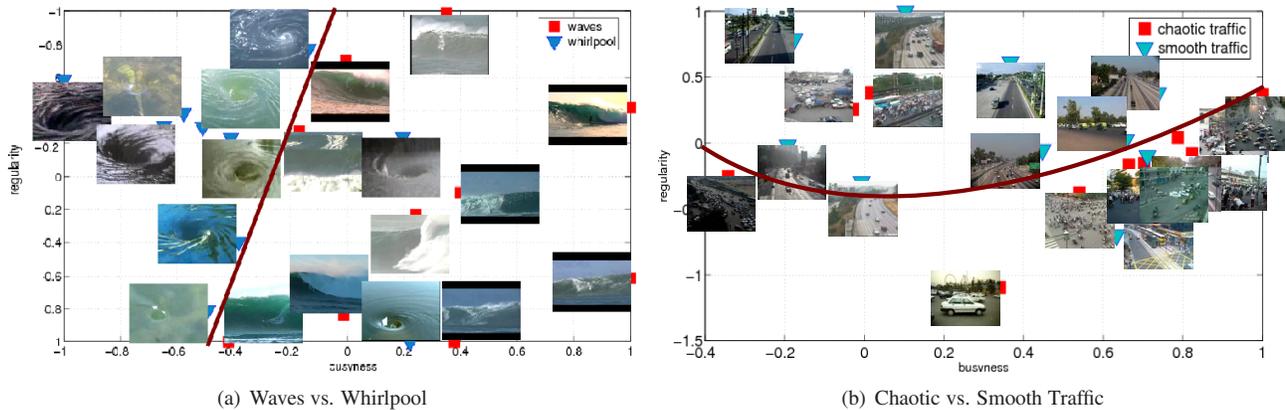
Figure 7. Separation of the spatially similar classes using motion attributes. Both pairs of classes have very similar spatial attributes and thus are difficult to separate using them. A clear separation (red line) can be seen between the 2 classes on the 2 temporal attributes 'regularity' and 'busyness'. It should be noted that the red line has been drawn for the sake of visual illustration. Simple classifiers (linear and quadratic respectively) easily separate the 2 classes. Note that in (a) 2 examples of the 'whirlpool' class and in (b) 2 examples from each class fall on the wrong side.

further work in this direction.

References

[1] S. Ali, A. Basharat, and M. Shah. Chaotic Invariants for Human Action Recognition. In *ICCV*, 2007.

[2] I. Biederman. Aspects and extensions of theory of human image understanding. *Computational processes in human vision: An interdisciplinary perspective*, pages 370–428, 1988.

[3] A. Chan and N. Vasconcelos. Layered dynamic textures. *Advances in Neural Information Processing Systems*, 2006.

[4] K. De Cock and B. De Moor. Subspace angles and distances between ARMA models. In *Proc. of the Intl. Symp. of Mathematical Theory of Networks and Systems*, volume 1, 2000.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[6] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 2003.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[8] L. F. Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative Model Based Vision*, 2004.

[9] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[10] A. Fraser and H. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, 1986.

[11] M. Gorkani and R. Picard. Texture orientation for sorting photos "at a glance". *TR-292, MIT, Media Labortory, Perceptual Computing Section*, 1994.

[12] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008.

[13] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009.

[14] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[16] L. Li, R. Socher, and L. Fei-Fei. Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Automatic Framework. In *CVPR*, 2009.

[17] D. Marr. Vision: A computational approach, 1982.

[18] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[19] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 2008.

[20] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 2001.

[21] M. Perc. The dynamics of human gait. *European journal of physics*, 26(3):525–534, 2005.

[22] G. Pike, R. Kemp, N. Towell, and K. Phillips. Recognizing moving faces: The relative contribution of motion and perspective view information. *Visual Cognition*, 4(4):409–438, 1997.

[23] A. Quattoni and A. Torralba. Recognizing Indoor Scenes. In *CVPR*, 2009.

[24] M. Rosenstein, J. Collins, C. De Luca, et al. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D*, 65(1-2):117–134, 1993.

[25] N. Rubin and N. Albert. Real-world scenes can be easily recognized from their edge-detected renditions: Just add motion! *Journal of Vision*, 1(3):37, 2001.

[26] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object category in image collection. In *ICCV*, 2005.

[27] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, 2005.

[28] M. Szummer and R. Picard. Indoor-outdoor image classification. In *1998 International Workshop on Content-Based Access of Image and Video Databases*, page 42, 1998.

[29] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006.

[30] S. Ullman. *The interpretation of visual motion*. MIT press Cambridge, MA, 1979.

[31] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10(1):117–130, 2001.

[32] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on PAMI*, 2005.

[33] R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamic textures. In *CVPR*, 2005.

[34] G. Williams. *Chaos theory tamed*. Taylor & Francis Ltd, 1997.