Computer Vision and Image Understanding 123 (2014) 1–13

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Action recognition using global spatio-temporal features derived from sparse representations ☆

Guruprasad Somasundaram, Anoop Cherian, Vassilios Morellas, Nikolaos Papanikolopoulos *

Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 9 October 2012
Accepted 10 January 2014
Available online 22 January 2014

Keywords:
Global spatio-temporal features
Action classification
Activity recognition

ABSTRACT

Recognizing actions is one of the important challenges in computer vision with respect to video data, with applications to surveillance, diagnostics of mental disorders, and video retrieval. Compared to other data modalities such as documents and images, processing video data demands orders of magnitude higher computational and storage resources. One way to alleviate this difficulty is to focus the computations to informative (salient) regions of the video. In this paper, we propose a novel global spatio-temporal self-similarity measure to score saliency using the ideas of dictionary learning and sparse coding. In contrast to existing methods that use local spatio-temporal feature detectors along with descriptors (such as HOG, HOG3D, and HOF), dictionary learning helps consider the saliency in a global setting (on the entire video) in a computationally efficient way. We consider only a small percentage of the most salient (least self-similar) regions found using our algorithm, over which spatio-temporal descriptors such as HOG and region covariance descriptors are computed. The ensemble of such block descriptors in a bag-of-features framework provides a holistic description of the motion sequence which can be used in a classification setting. Experiments on several benchmark datasets in video based action classification demonstrate that our approach performs competitively to the state of the art.

1. Introduction

The increasing popularity of video hosting websites such as Youtube and Dailymotion has brought out a deluge of video data, demanding smart algorithms for their efficient processing. Among several potential applications that video data could facilitate, an important one that has witnessed significant interest from the computer vision community has been that of *action recognition* in video sequences. Although conceptually similar to static image data, action classification in videos pose several challenges. While issues such as intra-class variations, scale, affine transformations, noise, and clutter are common to videos and image data, we also need to tackle additional difficulties due to object/camera motion, dynamic backgrounds, and diversity in the speed at which actions are performed. Extra computational and storage resources needed for processing videos is another important impediment for developing explicit data-driven approaches.

Although there has been plenty of research in this area, not many approaches take advantage of the complexity or lack there-of (redundancy) in the video data. Laptev [1] introduced the space time interest point detector approach to finding the unique spatio-temporal features in a manner similar to finding Harris corners in images. They model significant local phenomena in the video based on spatio-temporal scale-space extrema. Alternatively, one could think of the "interesting" regions as globally significant, i.e., over the entire video sequence. The main intuition of our approach is to exploit the structural redundancy (for example, dynamic but periodic backgrounds, moving camera, etc.) in the video; avoiding such non-informative regions of the video will leave us with portions with significant foreground motion that is different from the spatio-temporal variations in the rest of the video. Liu et al. [24] describe a feature selection approach to address this issue which involves computing multiple static and motion features followed by rule based pruning to select useful static features and a page rank metric on similarity graphs of motion features. This is followed by mutual information based optimization of action class vocabularies. Consequently, the vocabularies are discriminatively trained. The corresponding histogram features are then classified using a heterogeneous adaboost training of static and motion features. Whereas Liu et al. compute known static and motion features and prune them, we propose a new feature detection approach which optimally selects spatio-temporal features based on minimum description length principles. Additionally, we do not aim to optimize the individual class based

☆ This paper has been recommended for acceptance by Jordi Gonzàlez.

* Corresponding author. Fax: +1 612 625 0572.
E-mail addresses: guru@cs.umn.edu (G. Somasundaram), cherian@cs.umn.edu (A. Cherian), morellas@cs.umn.edu (V. Morellas), npapas@cs.umn.edu (N. Papanikolopoulos).

http://dx.doi.org/10.1016/j.cviu.2014.01.002
1077-3142/

vocabularies to keep the framework as generalizable as possible as such an effort can become daunting with a large number of action classes such as in the UCF 50 dataset.

In order to determine which spatio-temporal variations are informative, we take inspirations from information theory by modeling the complexity of each spatio-temporal patch (a sub-window of the image sequence). Towards this end, we would like to estimate the *Kolmogorov complexity* [2], which defines the minimum length of a code that can represent the given data; the more the redundancy in data, fewer bits are enough to represent it. Even though this complexity cannot be computed in practice, we can attempt to approximate it using the idea of minimum description length (MDL); in which the encoding scheme is restricted to only a predefined family of encoding models. We propose to use a dictionary learning and sparse coding model for MDL, which we show affords a simple and efficient algorithm for estimating the self-similarity of video data. In fact, we can observe in practice that many signals are redundant and it is fair to assume that such sparsity exists if a suitable basis dictionary is chosen. Most patches can potentially be represented sparsely with a suitable basis, however despite efforts there might be a few patches which can not be represented sparsely. Such patches might require additional representation "budget" or in other words their description lengths are longer than expected. These patches are likely to be more informative (less-redundant). The exact values of the estimates are not important since our interest is to rank the top $M\%$ salient patches, for a relatively small M.

This paper is organized as follows. In the next section, related work for the problem of human action classification based on spatio-temporal features is discussed. In Section 3, we provide the theoretical motivation for measuring the complexity (description length) of spatio-temporal patches. We formulate the problem of measuring the complexity as the representation error (ℓ_2 norm distortion) based on dictionary learning and sparse representation. In Section 3.4, we describe our algorithm for classifying human actions using our global spatio-temporal descriptors in a bag-of-features framework. Finally in Section 4, we present experimental results on several benchmark datasets for video based action recognition.

2. Related work

Action recognition emerge as an important problem in a variety of computer vision applications ranging from visual surveillance [3], video understanding [4,5], learning social interactions [6], and diagnostics of mental disorders [7] to name a few. Over the past decade, the problem of action recognition in videos has been looked at from multiple perspectives, with a diverse set of solutions suggested. Bashir et al. [8] demonstrate the use of HMMs with spatio-temporal curvature representations and PCA decompositions for trajectory based human action classification. Another powerful representation method involves describing actions as a sequence of shapes [9], and has gained popularity due to its invariance properties. Qiu et al. [10] propose a Gaussian process based dictionary objective function for efficient modeling of actions and for learning new actions. They use a discriminative dictionary learning approach and a probabilistic sparse coding scheme. Such frameworks do not generalize easily to a large number of classes. For example, for the UCF 50 dataset, learning discriminative dictionaries for 50 classes is cumbersome. In contrast, our approach uses a generative model for each video and discrimination is achieved at the classifier level. Our approach lends itself to many classification frameworks beyond bag of words and support vector machines. Liu et al. [11] propose the use of quantized spatio-temporal volumes and quantized spin images for capturing the shape deformation of actors in a graph based framework for action recognition. Wang et al. [12] propose trajectory descriptors in a bag-of-features framework. They use a KLT tracker on SIFT features between frames to extract the trajectories. Other popular methods for action recognition in videos include, but not limited to 3D gradients [13], 3D SIFT descriptors [14], velocity profiles [15], and convolutional neural networks [16]. In [17–19], surveys of some of the most popular methods can be seen.

Our paper describes a representation scheme akin to space time interest point (STIP) models [1], on which several other approaches build upon [20,21]. Interestingly almost all of these methods [1,22,23] have employed a local spatio-temporal scale-space extrema detection approach. One important factor deterring the consideration of these descriptors at a global scale (such as the entire video) is the high computational cost. In contrast, we argue that significant features informative for action recognition can be obtained from the spatio-temporal regions of the video sequence if we use the right set of tools and models.

There have been several approaches for action recognition that use sparse representations, similar to ours. Guo et al. propose a framework based on sparse representation of region covariance descriptors, computed on optical flow features [25] and silhouette tunnels [26]. Castrodad and Sapiro [27] propose a deep-layered discriminative approach for classifying human actions based on learned basis vectors or action primitives for each action category. These approaches try to capture the most non-redundant or informative spatio-temporal features representative of a certain class (top-down). In contrast, we determine in a bottom-up manner the global spatio-temporal features corresponding to the most non-redundant (least self-similar) regions of a video sequence. The discriminative modeling in our classification step is performed using a support vector machine. There are several other types of interest region detectors for video sequences which are usually an extension of 2D interest point detection methods to spatio-temporal sequences. A few examples are Harris 3D [1], Cuboids [22], Hessian [28], and dense sampling [29–31]. Detailed evaluation surveys of local spatio-temporal feature detectors as well as descriptors are provided in [32–34].

Recently, there have been approaches in literature which have reported excellent performance on the datasets evaluated in this paper. Gilbert et al. [35] describe a hierarchical representation of features in a multi-stage approach to capture the most distinctive components of actions. Wu et al. [36] use optical flow to find motion, thereby avoiding the need for object detection and creating robust trajectories. These motion features are then used in an SVM for performing action recognition. Raptis [37] describes another trajectory clustering approach for obtaining mid-level descriptions of the most significant action components useful for recognition. However, we compare our results with our feature detectors in a bag-of-features SVM framework. This category of methods has been studied extensively in parallel with the other approaches as it aims to solve the underlying action representation problem in an unsupervised way. No a priori information about class labels are used to mine the most distinctive features in the bag-of-features. However this information is consumed during the SVM optimization. Hence performance benefits are typically a result of a sound feature detection/ description approach.

3. Theory

In this section, we detail our approach from a theoretical standpoint. We draw motivation from the theory of Kolmogorov complexity and entropy. We aim to determine the most informative spatio-temporal regions in a video sequence as defined by its description length. The more complex (or longer) the description

length of a spatio-temporal patch, the more informative or complex is the patch. We will begin with a brief overview of prior work connecting information theory and image saliency, which helps us build the necessary fundamentals on which our sparsity based saliency computation framework is based. Next, we will formulate our problem formally, preceding which we detail our approach.

3.1. Information theory and saliency

Visual saliency has been looked at from the perspective of information theory several times in the past. In [38], local entropy is suggested as a means to find salient image patches. Inspired by biological vision [39] defines saliency as the minimum uncertainty under perceptional distortion associated with an image region given its local neighborhood. This uncertainty is quantified in terms of the conditional entropy; computing which is difficult in practice. Assuming an image patch and its surrounding patches are distributed as multivariate Gaussian, the conditional entropy is approximated via the lossy coding length. In [40], it is argued that saliency should be measured with respect to the likelihood of a given patch given its surrounding patches. The paper suggests an information maximization framework using the Shannon's self-information measure as captured by the negative log-likelihood of a patch given its neighboring patches. The connections of this measure to biological underpinnings of visual saliency is hypothesized and empirically validated. Saliency is quantified in terms of entropy gain in [41]. They propose incremental coding length, a principle by which energy is distributed in the visual attention system. According to this principle, features corresponding to unexpected visual cues will have high entropy gain and therefore get high energy.

The method introduced in this paper is also inspired from information theory. Similar to [40], we use the negative log-likelihood as a measure of self-information. That is, if $\mathbf{x}$ is a patch in the image, and if $p(\mathbf{x})$ defines the probability of occurrence of this patch given the statistics of patches in the image, then we define the saliency of this patch as $-\log p(\mathbf{x})$ which quantifies the 'surprise' in seeing $\mathbf{x}$. Unlike [40], which uses independent component analysis in a neural network framework to approximate $p(\mathbf{x})$, we resort to the principles of coding theory, specifically the concept of *Kolmogorov complexity* [42]. This principle characterizes the minimum number of bits required to encode given data. Intuitively, as regularity in the data increases, the number of bits required to represent it decreases. Unfortunately, the Kolmogorov complexity is not a computable quantity [43] and thus various approximations are sought. One such important approximation, which we will be using in this paper, is the idea of minimum description length (MDL).

3.2. Minimum description length

Introduced in [44], the main idea of MDL is to restrict the search for the best data model to minimally represent the data to a feasible family of models. MDL seeks the best model from this family using some predefined criteria. Formally, given a set of candidate models $\mathcal{M}$, and a data vector $\mathbf{x}$, MDL seeks the model $M \in \mathcal{M}$ that best approximates $\mathbf{x}$ with respect to a given *coding assignment function* ℓ. That is,

$$\widehat{M} = \arg\min_{M \in \mathcal{M}} \ell(\mathbf{x}, M). \tag{1}$$

In the context of our saliency model described in the last section, the coding function takes the form $\ell(\mathbf{x}, M) = -\log p(\mathbf{x}, M)$, where instead of the marginal $p(\mathbf{x})$, we approximate it in terms of a suitable probability model M. Applying Bayes theorem, we can write the joint model $p(\mathbf{x}, M)$ as $p(\mathbf{x}, M) = p(\mathbf{x}|M)p(M)$, using which we can rewrite our coding model as:

$$\ell(\mathbf{x}, M) = -\log p(\mathbf{x}|M) - \log p(M). \tag{2}$$

The first part on the RHS of (2) captures how well the model describes the data, while the second part describes the complexity of the model itself.

Among several different coding functions available, we will use the standard Gaussian loss function to model ℓ. That is, assuming the data can be encoded by a given basis dictionary $\mathcal{D}$ and a coefficient vector α, the reconstruction error follows a Gaussian distribution: $p(\mathbf{x}|\mathcal{D}, \alpha) \sim \mathcal{N}(\mathcal{D}\alpha, \beta^2)$, where β is the standard deviation. Further, assuming parsimony of data representation, we will assume only a few dimensions in the coefficient vector are used, and is captured by the model prior $p(M)$. Such a sparsity assumption goes well with the idea of MDL and has been previously employed in the context of image denoising [45–47] in which the underlying signal part of the data vectors is assumed to be representable by a few columns of $\mathcal{D}$, while the noise part cannot be represented by any combination of the columns. Incorporating these assumptions and writing α explicitly in terms of its sparsity k, (1) leads to:

$$\hat{\alpha} := \min_{\alpha} \frac{1}{\beta^2} \|\mathbf{x} - \mathcal{D}\alpha\|_2^2 \quad \text{subject to} \quad \|\alpha\|_0 \leqslant k, \tag{3}$$

where $\|\alpha\|_0$ stands for the number of non-zero dimensions in α and k controls the sparsity of $\mathbf{x}$ in $\mathcal{D}$. In this context, we define the descriptor length of the data vector $\mathbf{x}$ as that $\|\alpha\|_0$ that minimizes the objective in (3).

The performance of this representation is evaluated by the compressibility measure representing the lowest achievable distortion of $\mathbf{x}$ in the learned basis $\mathcal{D}$. That is, if Σ_k denotes the set of all k-sparse coefficient vectors in $\mathcal{D}$, we define the error in the k-sparse representation $\sigma_k(\mathbf{x})$ as:

$$\sigma_k(\mathbf{x}) := \inf_{s \in \Sigma_k} \|\mathbf{x} - s\| \tag{4}$$

for some suitable distortion measure $\|\cdot\|$. For a signal that does not have approximability with a k-sparse representation, we can state it has a higher complexity given the describing basis $\mathcal{D}$. Even though the complexity is an absolute quantity and practically indeterminate, we fix the size of $\mathcal{D}$ and then measure the complexity of data relative to its description length (sparsity) in $\mathcal{D}$ by way of the error incurred with an allowed description (representation) length. Our main intuition is that this error will be high for salient data points. We will discuss the mathematical details of this idea next.

3.3. Formulation

There are two important impediments to the direct use of the encoding scheme suggested in (3), namely (i) it is a non-convex combinatorial problem and is known to be NP-hard, and (ii) the dictionary is assumed to be given. The former issue is well-known in compressive sensing [48], and is usually circumvented by replacing the combinatorial ℓ_0 norm by its closest convex counterpart – the ℓ_1 norm. For the second issue, dictionary learning methods have been suggested in which the dictionary $\mathcal{D}$ is learned from the data itself. Making these changes to our objective, and assuming the dictionary $\mathcal{D} \in \mathbb{R}^{d \times n}$ has $d_1, d_2, \ldots, d_n$ as its columns, we have the following standard dictionary learning and sparse coding problem variant of (3):

$$\min_{\alpha, \mathcal{D}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathcal{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1, \tag{5}$$

$$\text{s.t.} \quad \|d_j\|_2 \leqslant 1, \quad j = 1, 2, \ldots, n, \tag{6}$$

where $\mathbf{x}_i$, $i = 1, 2, \ldots, N$ represents the data vectors, and λ is a regularization constant. The inequality constraints on the dictionary atoms are added to avoid degenerate cases while learning the

dictionary. This objective is non-smooth and non-convex, but can be solved efficiently using the K-SVD algorithm [49].

Given a dictionary D, the sparse representation error of a data point $\mathbf{x}$ is given by

$$\mathcal{R}(\mathbf{x}, \mathcal{D}, \alpha) = \|\mathbf{x} - \mathcal{D}\alpha(\mathbf{x}, \mathcal{D})\|_2. \quad (7)$$

Here $\alpha(\mathbf{x}, \mathcal{D})$ is the sparse representation for the pair $(\mathbf{x}, \mathcal{D})$, and can be done fast using orthogonal matching pursuit (OMP) [50] for which we can leverage the already existing efficient implementations [51,52]. Note that OMP is a greedy algorithm and we resort to it for reasons related to efficiency rather than correctness or exact recovery.

3.4. Approach

We propose the use of sparse representations on spatio-temporal patches to determine the saliency of those patches. We argue that these globally salient spatio-temporal regions are very informative while modeling action sequences. That is, the saliency of each spatio-temporal volume is measured with respect to the rest of the video sequence itself rather than just the local neighborhood. This is unlike many contemporary approaches for detecting spatio-temporal features such as the Harris 3D [53] and STIP (spatio-temporal interest points) [1]. However, we share a similarity with these approaches in our effort to make the features as robust to spatio-temporal scale variations as possible. We perform convolution of the spatio-temporal patches with a spatio-temporal (3-dimensional) Gaussian kernel at two different scales to form a 3-level Gaussian pyramid (including the native scale). Adding more scales can improve the robustness of the approach, however it increases the processing time of each spatio-temporal window.

3.4.1. Feature detection (salient region detection)

Each video sequence is first arranged into many sliding temporal windows with w number of frames in each window. This window is then arranged in a Gaussian pyramid as mentioned before. Each pyramid level is then densely sampled into spatio temporal patches of size $b \times b \times w$ as shown in Fig. 1. The spatio-temporal patches are then vectorized to form the signal matrix X and each signal is a column and is b^2w dimensional. Three such signal matrices are produced corresponding to the three-level Gaussian pyramid. Then a dictionary (basis) is learned separately for each scale. These dictionaries are then concatenated column wise into a single multiscale dictionary. Then we proceed to measure the sparse representation error by performing orthogonal matching pursuit to determine the best k atoms from the dictionary to represent each spatio-temporal patch in the native scale. The best L atoms that are chosen could thus be originating from different scales. The residual error in such a representation is robust to variations in scale within the assumed limits. The multiscale scale dictionary spans the entire vector space of the spatio-temporal window under the sparsity consideration. In other words, the dictionary minimizes the average error in representation of every spatio-temporal block while using only k of its columns to represent each block. This implies that not all spatio-temporal blocks will have low representation error. The relative residual error for each patch (ranging from 0 to 1) is used to then rank the top $M\%$ salient patches. Note that this is not a hard threshold but a relative threshold which varies across each temporal window. The higher the residual error, the more self-dissimilar the patch is with respect to the entire spatio-temporal window and the more salient it is. After these top regions are selected we compute the corresponding spatio-temporal descriptors.

Alternatively, instead of thresholding the patches based on the ranked ordering of saliency, we could compute the descriptors for

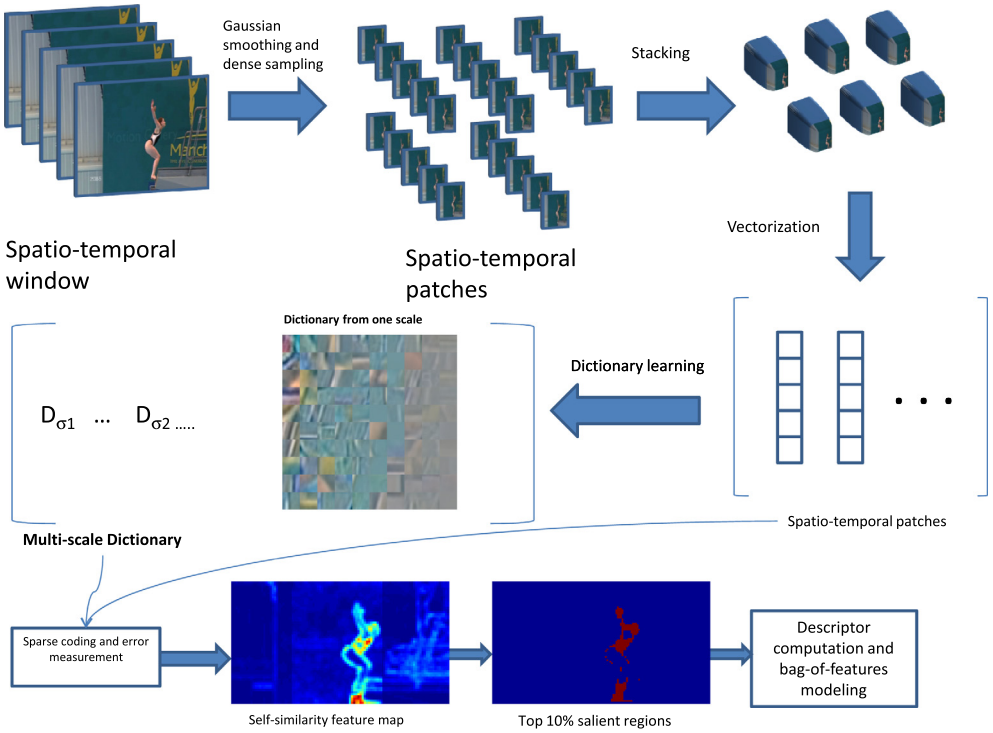

Fig. 1. An illustration of our approach. The video sequence is first arranged into temporal windows containing a certain number of frames. Each window is then decomposed into spatio-temporal patches which are vectorized and a dictionary specific to that window is learned. Then the spatio-temporal patches are represented using sparse set of coefficients with respect to the dictionaries and the representation errors are used as the saliency values.

every patch. While computing the bag-of-features histogram, the weights can be computed relative to their corresponding spatio-temporal saliency value (or residual error), defined as:

$$\theta_i = \frac{R_i}{\sum_{j=1}^{P} R_j}, \tag{8}$$

where θ_i is the weight of the ith patch and R_i is its residual error. In our experiments, we compare this soft-weighting scheme with the hard thresholding approach of selecting the top $M\%$ most salient patches. We observed a performance improvement in two datasets with this soft-weighting scheme.

3.4.2. Descriptor computation

After the salient regions are detected, we compute two types of descriptors on them, namely (i) HOG (histogram of oriented gradients [54]) and (ii) the region covariance descriptor [55] (extended for spatio-temporal volumes). For the former, HOG descriptors are computed on each $3 \times 3 \times 2$ sub-patch individually and then concatenated with re-normalization to form the descriptor of the entire spatio-temporal patch.

Region covariance descriptors first compute a set of features for every pixel in a spatio-temporal patch. This descriptor of this patch amounts to computing the covariance matrix of all such features in this patch. For a ith pixel at space–time coordinates (x, y, t) in a given spatio-temporal patch, we use the feature vector f_i given by:

$$f_i(x,y,t) = \left[x, y, t, \left|\frac{\partial I(x,y)}{\partial x}\right|, \left|\frac{\partial I(x,y)}{\partial y}\right|, \left|\frac{\partial^2 I(x,y)}{\partial x^2}\right|, \left|\frac{\partial^2 I(x,y)}{\partial y^2}\right|, \left|\frac{\partial^2 I(x,y)}{\partial x \partial y}\right|, \left|\frac{\partial I(x,y,t)}{\partial t}\right|\right]^T, \tag{9}$$

where I is the gray scale intensity at (x, y, t), and the partial derivatives represent the first and second order gradients along the x, y, and time dimensions respectively. Given this feature set at each (x, y, t) locations, we can compute the covariance descriptor of a particular spatio-temporal patch as:

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(f_i - \mu)(f_i - \mu)^T, \tag{10}$$

where n is the number pixels in the spatio-temporal patch volume and μ is the average feature vector, $\mu = \frac{1}{n}\sum_{i=1}^{n} f_i$.

Covariance descriptors are not euclidean objects, but belongs to a curved manifold geometry. As a result, we cannot use the conventional euclidean straight-line distance to compare covariance matrices, rather we need to use curved geodesic distances adhering to its geometry. Among several metrics available for computing the similarities of these descriptors, we use the log-euclidean distance which amounts to projecting the positive definite covariance descriptors to the flat euclidean distance using a matrix logarithm operator. After this projection, these projected descriptors could be compared using the euclidean distance [56].

3.4.3. Bag of features

The bag of features approach is commonly used for feature based object and action classification [32,57,34]. For each dataset consisting of different action classes, we detect the top $M\%$ salient regions and compute the descriptors in these regions. We collect about $100K$ feature descriptors of each type (HOG, Region Covariance) and perform K-means clustering to generate a vocabulary. Using these vocabularies, a histogram based on matching features for each action sequence is then computed. This histogram feature is then used as the feature descriptor for each action sequence in a support vector machine framework. For the SVM learning, different kernels are tested, such as linear, Gaussian radial basis function (RBF), polynomial, and exponential χ^2 kernels. For two histogram vectors x and y and a constant scalars γ and c, each of these kernels is defined as follows:

$$K_{linear}(x,y) = x^T y + c, \tag{11}$$

$$K_{polynomial}(x,y) = (\gamma x^T y + c)^d, \tag{12}$$

where d is the degree of the polynomial.

$$K_{RBF}(x,y) = \exp(-\gamma \|x - y\|_2^2), \tag{13}$$

$$K_{\exp-\chi^2}(x,y) = \exp\left(-\frac{1}{v}\chi^2(x,y)\right), \tag{14}$$

where

$$\chi^2(x,y) = \frac{1}{2}\sum_{i=1}^{p}\frac{(x_i - y_i)^2}{x_i + y_i}, \tag{15}$$

where we assume the feature vector has p dimensions, each dimension captured by the respective subscript i, and v is the mean χ^2 distance between all the training samples [58]. In general, histogram similarities are better captured by the exp-χ^2 kernels. However, in some cases we noticed that the polynomial, or Gaussian RBF kernels performed comparably. The best choice of the kernel was established through cross validation with each training set. On an average, we noticed the exponential χ^2 kernel showed about 3–4% performance boost over the next best kernel on the validation data. The SVM decision function f has the following form (for one class case): for a new data point y and training points given by x_i and their respective learned weights represented by w_i,

$$f(x) = \sum_i w_i y_i K(x_i, x) - T. \tag{16}$$

We classify the new data point y to a given class if the decision value is positive. We evaluate datasets which are multi-class and we use multi-class SVMs [59] and report the performance with respect to each class.

4. Experiments and results

In this section, we provide a brief description of the datasets we used for evaluation. Then, the process of selecting the best parameters for each dataset is discussed. Finally, we provide details of the performance of our method on all the datasets.

4.1. Datasets

We use the KTH actions dataset [23], the UCF sports action dataset [60], the Hollywood movie actions dataset [61], and the more recent and exhaustive UCF50 dataset [62]. In this way, we have tested our approach on datasets of varying number of actions and scene complexity.

4.1.1. KTH actions dataset

The KTH actions dataset has six action classes, namely walking, jogging, running, boxing, waving, and clapping. The dataset contains 2391 action sequences, each action performed several times by 25 subjects in four different scenarios. The sequences have a frame rate of 25 fps and a resolution of 160×120 pixels. We used the training + validation set for training (16 subjects) and report the performance on the test set (9 subjects). Some sample actions from this dataset, corresponding spatio-temporal saliency map (self-similarity), and the top 10% salient regions are shown in Fig. 2.

4.1.2. UCF sports actions dataset

The UCF sports action dataset consists of sequences collected from television channels such as BBC and ESPN. The dataset

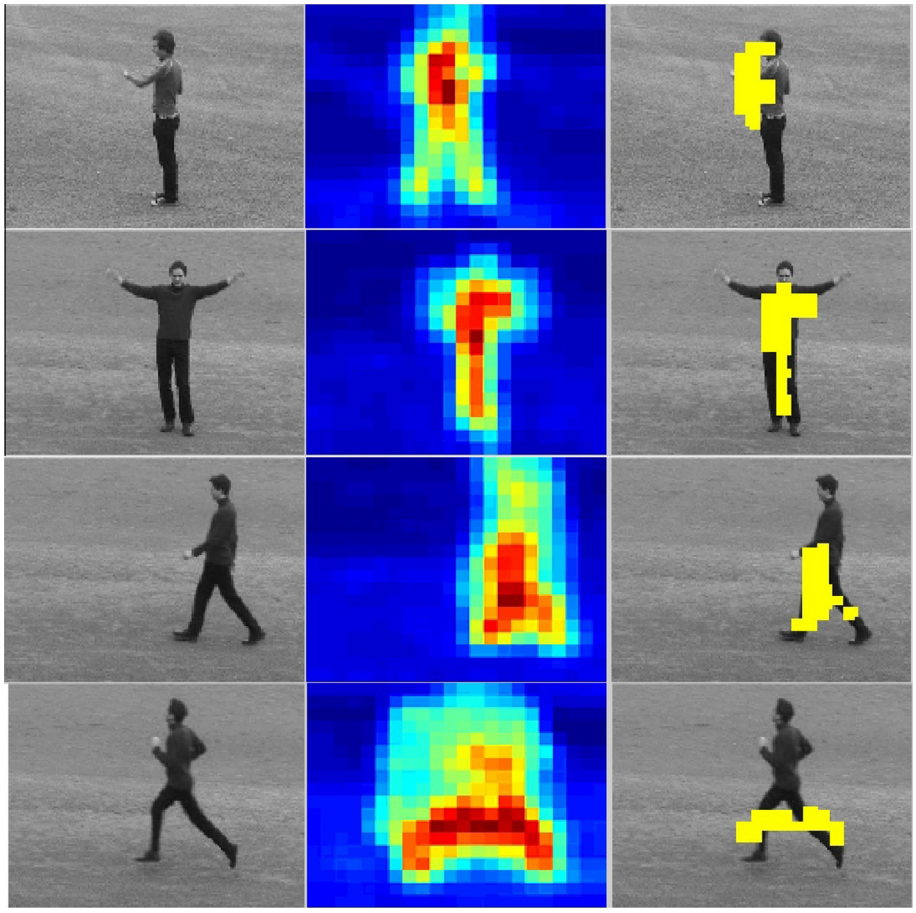

Fig. 2. An illustration of some action sequences from the KTH actions dataset and the corresponding salient features. From left to right: an action frame, saliency map, and the top 10% salient regions (used in feature computations).

contains 150 video sequences at a resolution of 720×480. The actions included are diving, Golf swinging, kicking, lifting, horseback riding, running, skating, swinging, and walking. These sequences were down-sampled to a resolution to match the KTH actions dataset (160×120) in order to reduce the computational burden, as well as to limit within the range of the spatial scale choices. Also, we generated additional samples using mirrored versions of the video sequences. Further, since no training or test set separation is provided, we report the average performance for leave-one-out cross validation. This is done by training the classifiers on all-but-one sequences and testing on the one left out sequence. Some action sequences and corresponding salient features from this dataset are shown in Fig. 3.

4.1.3. Hollywood movie actions dataset

The Hollywood movie actions dataset consists of more than 400 clippings from 32 movies representing the following actions: answering phone, getting out of car, handshaking, hugging, kissing, sitting down, sitting up, and standing up. We used the "train-clean" and "test-clean" sequences of this dataset for training our classifiers and testing. The training set consists of 219 action samples which contain manually provided labels obtained from twelve movies. The test set contains of 211 labeled action samples from 20 movies. Some sample action sequences and corresponding salient regions are shown in Fig. 4.

4.1.4. UCF 50 actions dataset

The UCF 50 actions dataset is a realistic dataset consisting of videos of 50 actions hosted on Youtube. The categories are: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot,Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo. This dataset is considered challenging due to variations in camera motion, object pose, clutter, etc. Some action sequences and corresponding salient features from this dataset are shown in Fig. 5.

4.2. Parameter selection and sensitivity

In order to obtain the best performance out of our approach, we need to determine the optimal choices for all parameters. By scaling all video datasets to the same size, we limit the ranges over which we need to search for the best choices. Specifically, the main parameters are: the spatial patch size (e.g., 14×14–20×20), the temporal window size (e.g., 4 frames in a window to 10 frames in a window), and lastly the choice of M for the selection of the top $M\%$ salient features. Even though we determine the features using a multiscale approach, the same choices of parameters may not be optimal for all datasets. For example, the number of frames in a temporal window depends on how fast the actions are performed. In the KTH dataset, the actions are more structured and performed at a uniform rate. This cannot be expected in real sports

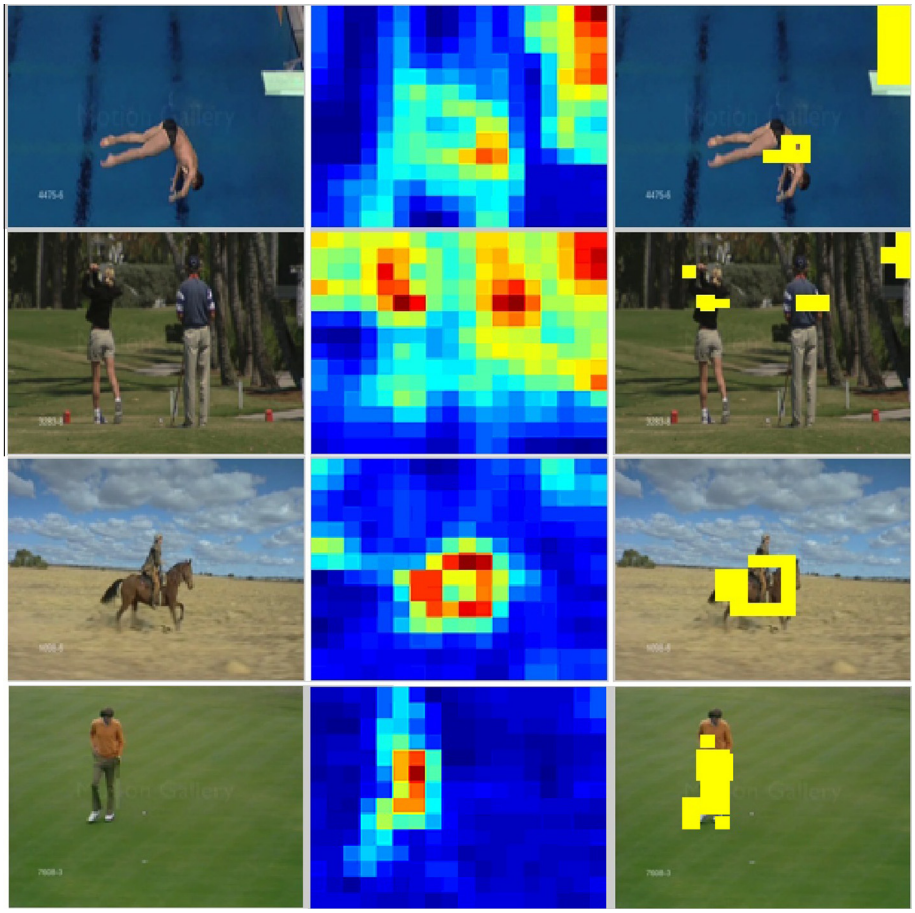

Fig. 3. An illustration of some action sequences from the UCF sports actions dataset and the corresponding salient features. From left to right: an action frame, saliency map and the top 10% salient regions.

action sequences or movie actions. In fact, one choice may not work for all the different sequences. But as with many other methods, we try to determine the parameter choices that maximize the performance for each dataset individually.

A common parameter selection for every dataset is the choice of the size of the dictionary D and the representation length L. There is a trade-off between these parameters and often times one can achieve a lower representation error by increasing either of these parameters. For analyzing the sensitivity to these parameters, we fix L and vary the size of the dictionary D and observe the error distribution of patches. For this experiment, we used 24 sequences from the KTH dataset (4 sequences from each action), 27 sequences from the UCF sports actions dataset (3 sequences from each action). We measured the representation error of each spatio-temporal patch ($14 \times 14 \times 4$) for varying sizes of learned dictionaries (98 elements, 196 elements, 294 elements, and 392 elements). The distribution of errors are shown in Figs. 6 and 7 respectively. There are a couple of inferences we could make from these distributions. First, the distributions corresponding to the different sizes of dictionaries are more or less similar. This indicates that even though the actual errors might be different for different dictionary sizes, the rank ordering of the patches based on error is likely to be similar. We also verified this by computing the intersection of the sets of the top $M\%$ patches for varying values of $M\%$ for the chosen subset of sequences. These averages are shown in Table 2. Second, they give us an insight about the two datasets and the applicability of hard thresholding or soft-weighting of the saliency maps. The KTH dataset is highly bimodal, with a large number of near zeros saliency values indicating that a hard thresholding might give better performance since the weights corresponding to most patches will be near zero. However, the UCF dataset sequences follow a fat-tail distribution thereby making soft-weighting or a higher value of M a desirable choice. This is further confirmed in our classification results.

For the KTH and the Hollywood movie actions dataset, we determined the optimal choices using the training (and validation) set. For the UCF sports action dataset, 40 sequences ($\approx$35% of the data) were used as a validation set for parameter selection. For the UCF 50 dataset, we used one video per group from each action to do parameter estimation. The best choices are listed in Table 1. Note that the parameter M was selected only for hard-thresholding approach. We searched a range of spatial windows from 14×14 to 20×20 pixels, temporal windows from 4 frames per window to 10 frames per window all in steps of 2. For each combination, we determined the top $10\%, 20\%, 30\%, 40\%$ and 50% salient regions and computed the corresponding feature descriptors in our bag-of-feature framework. The cross-validation accuracy for these choices on this subset of sequences are shown in Table 3. We notice that the best choices do not vary by much between datasets. Also the dense soft weighting scheme was also used for comparison. The dense soft weighting scheme provided marginally better results for the UCF sports action dataset and an improved performance with the Hollywood dataset. The dense soft weighting also did markedly better on the UCF 50 dataset. The best choices of patch sizes are more or less the same and they are also in agreement with other approaches [32]. There were only minor

Fig. 4. An illustration of some action sequences from the Hollywood movie actions dataset and the corresponding salient features. From left to right: an action frame, saliency map and the top 10% salient regions.

differences in the temporal window sizes between the different datasets. We believe that the reason for the UCF sports, UCF 50 actions and Hollywood movie action sequences requiring more salient features in the model is because these actions (Golf Swing, Answering Phone, Getting out of Car, etc.) required additional information about the scene and the interaction with the objects present in the scene for accurate classification. The information gathered from background was useful in classifying the actions. For instance snow and water are key differentiating features in skiing and rowing. Note that the best choices were obtained using 10-fold cross-validation on the validation set.

Finally we also determine the size of the feature vocabulary to be used in the bag of features framework for each feature type. In most cases, we noticed the best results for a choice between 1500 and 3000 feature words. Smaller vocabularies (1500–2000 words) produced best results for the region covariance descriptor whereas larger vocabularies with 3000 words were more suited for HOG feature descriptors. This is likely because of the inherent lower dimensionality (45 dimensions) of the covariance descriptor.

4.3. Results and discussion

Using the optimal parameter choices discussed in the previous Section, we train the SVM classifiers for each dataset. For the KTH and Hollywood movie actions dataset the training set and validation set were used to determine the best kernel choices for SVMs as well as their parameters (penalty factor, distance scaling variable, etc.). For the UCF sports action dataset 40 sequences representing all action classes were used in cross-validation for determining the kernel parameters (penalty factor, distance scaling variable, etc.). For the KTH actions dataset best results were obtained using the Gaussian radial basis function, whereas for the UCF sports actions, UCF 50 as well as the Hollywood movie actions datasets, we obtained best performance with the exponential χ^2 kernel described previously. We compare the results of our approach using the HOG descriptor with other methods and we also report our performance with the region covariance descriptor. Due to the lack of availability of individual implementations of other feature detection methods (alone to be used with region covariance descriptor) we do not compare the results of using the covariance descriptor with other methods. Finally, we combine the best region covariance descriptor based classifier and the best feature HOG classifier for each dataset as follows. We obtain the best bag-of-features histogram corresponding to each feature and then concatenate them together. This histogram is then renormalized and used as a combined feature [63] for classification. This method performed best for the Hollywood movie actions dataset and comparably for the other datasets.

The results of performance for each dataset are shown in Tables 4–7. Note that for the KTH and the UCF sports actions dataset, the accuracies corresponding to each feature descriptor are shown. However for the Hollywood actions dataset the accuracies shown are for the combined feature method only as it performed the best. The corresponding accuracies shown for the method of Laptev et al. [61] are also results from the best feature combinations. We notice that soft-weighting scheme produced better re-

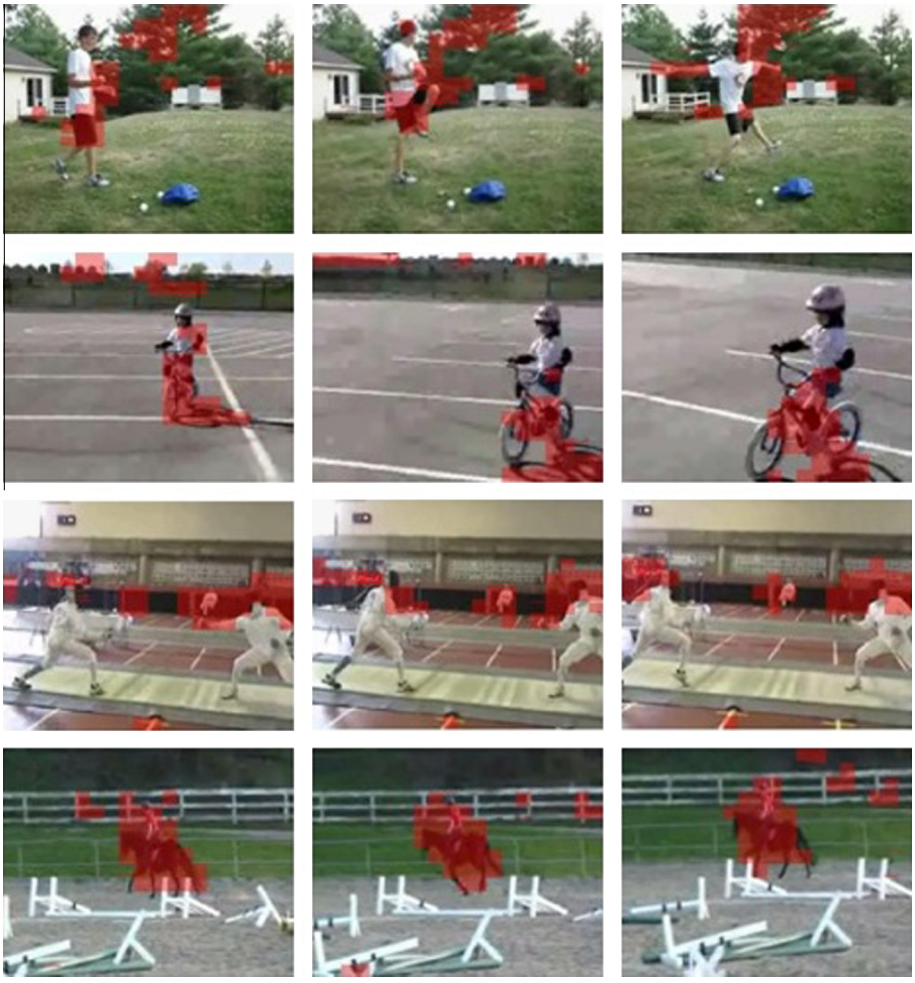

Fig. 5. An illustration of some action sequences from the UCF 50 actions dataset and the corresponding salient features. 3 frames from the same video overlaid with the top 10% salient regions.

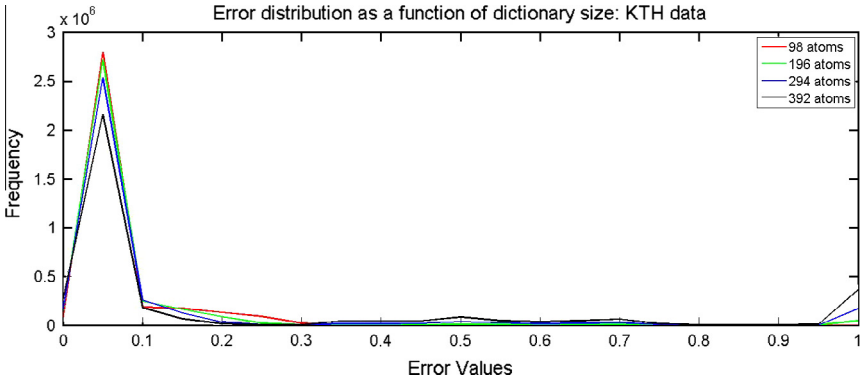

Fig. 6. Distribution of errors of patches corresponding to 24 KTH dataset sequences for varying values of dictionary sizes. The distributions are very bimodal and fairly uniform across the different choices.

sults for the UCF sports actions, UCF50 dataset and the Hollywood Movie Actions dataset. We obtained better results by not rejecting any regions based on the saliency and this phenomenon is also reflected in the higher threshold required to achieve a high performance using the hard-thresholding approach. However, the caveat is that the computation of a lot more descriptors are needed and consequently the processing time is increased.

From these results, we can probably infer that hard thresholding produces good results for static camera situations while soft weighting is better suited for dynamic backgrounds and more complex actions. Note that in our analysis we do not include any optical flow based feature descriptors. Even though these descriptors provide good performance, they are very expensive to compute. On the other hand the HOG and the covariance descriptors

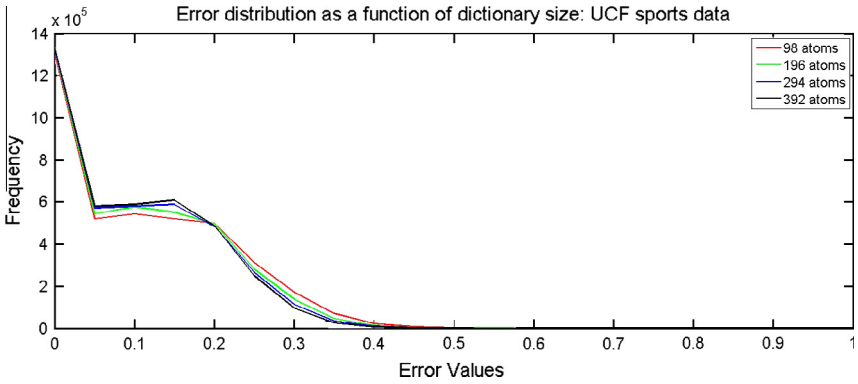

Fig. 7. Distribution of errors of patches corresponding to 24 UCF sports dataset sequences for varying values of dictionary sizes. The distributions are very fat-tail and fairly uniform across the different choices.

Table 1
Best parameter choices for each dataset determined through cross-validation on the training set. Covariance descriptors were used for this evaluation.

Dataset size	Patch size	Temporal window	[%] Salient considered
KTH actions	16×16	4	10
UCF sports actions	18×18	6	30
Hollywood movie actions	16×16	6	30
UCF50	16×16	4	40

Table 2
Average percentage overlap between saliency maps across different choices of $M\%$ hard thresholding for varying sizes of dictionaries (98, 196, 294, and 392 elements).

Dataset	Average overlap for $M\%$ salient considered				
	10	20	30	40	50
KTH actions (%)	71	76	81	84	91
UCF sports actions (%)	62	68	73	85	94

Table 3
Crossvalidation accuracy vs. $M\%$ salient patches considered on the validation set. The best choices of patch and window sizes were used from Table 1. Covariance descriptors were used for this evaluation.

Dataset	10%	20%	30%	40%	50%
KTH actions (%)	89.11	88.34	86.04	85.81	85.76
UCF sports actions (%)	84.31	84.71	86.71	86.03	85.91
Hollywood movie actions (%)	44.27	44.91	46.04	45.81	45.36
UCF 50	69.3	69.7	74.3	79.1	75.9

Table 4
Average accuracies for different methods on the KTH actions dataset. Performance on the test set is shown. Accuracies of other methods obtained from Wang et al. [32]

Feature detector/descriptor	Accuracy (%)
Harris 3D + HOG	80.9
Cuboid + HOG	82.3
Hessian + HOG	77.7
Dense + HOG	79
Our method + HOG	**85.3**
Our method + Covariance	**88.2**
Our method + Combined Descriptor	**90.1**
Our method + HOG + Soft	**83.4**
Our method + Covariance + Soft	**86.6**
Our method + Combined Descriptor + Soft	**89.6**

The numbers in bold indicate the best results.

Table 5
Average accuracies for different methods on the UCF sports actions dataset. (Leave-one-out cross-validation.) Accuracies of other methods obtained from Wang et al. [32].

Feature detector/descriptor	Accuracy (%)
Harris 3D + HOG	71.4
Cuboid + HOG	72.7
Hessian + HOG	66
Dense + HOG	77.4
Our method + HOG	**80.2**
Our method + Covariance	**85.92**
Our method + Combined Descriptor	**84.1**
Our method + HOG + Soft	**82.6**
Our method + Covariance + Soft	**87.3**
Our method + Combined Descriptor + Soft	**86.04**

The numbers in bold indicate the best results.

Table 6
Average accuracies for different methods on the Hollywood movie actions dataset. Performance on the test set is shown.

Action	STIP Laptev et al. (CVPR '08) (%)	Our method + combined descriptor (%)	Our method + combined descriptor + soft (%)
AnswerPhone	**32.1**	17.1	22.4
GetOutCar	**44.5**	28.3	28.1
HandShake	**32.3**	31.1	31.6
HugPerson	40.6	**53.1**	52.7
Kiss	53.3	**60.2**	59.6
SitDown	**38.6**	26.3	33.5
SitUp	**18.2**	14.1	16.1
StandUp	50.5	**58.3**	57.6
Overall	**38.38**	36.1	37.7

The numbers in bold indicate the best results.

are relatively inexpensive. We improve the state-of-the-art in the realm of feature detection approaches for bag of features classification approaches in the KTH actions and the UCF sports action dataset. We perform better compared to other bag of words models on the UCF 50 dataset. It is also comparable to the state of the art in leave group out classification (76.9% http://crcv.ucf.edu/data/UCF50.php). This method uses fusion of multiple descriptors such as MBH which is an approach based on extracting motion boundaries and dense trajectories that require computing optical flow. In addition, this method also uses scene text descriptors in their fusion framework. In contrast, our feature description is low-level. Additionally, we could use our features in conjunction with the other higher level feature detectors for improved performance.

Table 7
Average accuracies for different bag of words methods on the UCF 50 actions dataset. (Leave-group-out cross-validation).

Feature detector/descriptor	Accuracy (%)
Color + Gray PCA [64]	41.3
GIST BoF [64]	38.8
Dense cuboids + HOG [65]	64.4
Dense cuboids + HOF [65]	65.9
Our method + HOG	63.2
Our method + Covariance	64.9
Our method + combined descriptor	**66.3**
Our method + HOG + Soft	**68.4**
Our method + covariance + soft	**69.2**
Our method + combined descriptor + soft	**70.1**

The numbers in bold indicate the best results.

Table 8
Average processing speed comparison for different feature detector and descriptor combinations.

Feature detector + descriptor	Processing speed (frames per second)	No. of descriptors/ frame
Harris 3D + HOG	1.6	31
Hessian + ESURF	4.6	19
Cuboid	0.9	44
Dense + HOG3D	0.8	643
Our method + HOG + hard	2.5	62
Our method + HOG + soft	0.6	336

Since our goal is to motivate this work as a low level feature detector, such an approach was avoided. We also perform comparably with the approach of Laptev et al. in the Hollywood dataset.

4.4. Complexity

Our implementations were primarily carried out in MATLAB with some optimized components in C++ (mex interfaced). On an average, our method could process ≈2.5 frames per second while computing one feature descriptor only and ≈0.4 frames per second while computing both the HOG descriptors as well as region covariance descriptors. Since features are computed for each spatio-temporal window and not for each frame individually, we typically obtain a dense set of features. This is further influenced by the choice of the threshold for top $M\%$ saliency consideration. We also notice a severe drop in performance while performing soft weighting since a lot more descriptors need to be computed. This can be viewed as a speed vs. accuracy trade-off specifically for the UCF sports, UCF 50 and Hollywood datasets. Table 8 shows processing rate comparisons for different methods obtained from [32] with our method.

5. Conclusion

To conclude, we summarize our contributions in this paper. We propose a novel spatio-temporal feature detector based on the sparse representation length of the spatio-temporal patches measured via the residual error. We establish the theoretical motivation to determine spatio-temporal features in this manner. The patches are ranked according to their spatio-temporal saliency determined by the error magnitudes. These features are also determined in a multi-scale approach thereby making the feature robust to variation in spatial and temporal scales. We then compute the region covariance and HOG descriptors corresponding to the salient spatio-temporal patches. These features are used in a bag-of-features approach with an SVM classifier framework. We improve upon the state-of-the-art bag of features models in two (KTH actions and UCF sports actions) of the four datasets evaluated and we perform comparably on the UCF 50 and Hollywood movie actions dataset. We obtain competitive performance without the computation of expensive features such as optical flow. Our method is computationally efficient and theoretically well founded. Our main contribution is a novel feature detection approach. We propose a generic action classification framework based on bag of features which is generalizable and can be adapted to large number of action classes. However, for a smaller set of action classes despite the good performance of our proposed framework, one could envision further optimizing the framework for additional performance boosts. This could involve an additional level of feature pruning, customized discriminatively trained dictionaries for each action class, optimization of bag of features vocabularies, etc. We are currently investigating such optimization methods to improve our current classification performance especially for smaller sets of actions such as in real world applications.

Acknowledgments

This material is based upon work supported in part of the Minnesota Department of Transportation and by the National Science Foundation through grants #IIP-0443945, #CNS-0821474, #IIP-0934327, #CNS-1039741, #SMA-1028076, and #CNS-1338042.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2014.01.002.

References

[1] I. Laptev, On space-time interest points, International Journal of Computer Vision, vol. 64, Springer, 2005, pp. 107–123.
[2] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: International Conference on Computer Vision, IEEE Computer Society, 2001.
[3] S. Danafar, N. Gheissari, Action recognition for surveillance applications using optic flow and SVM, in: Asian Conference on Computer Vision, Springer, 2007, pp. 457–466.
[4] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[5] A. Fathi, A. Farhadi, J.M. Rehg, Understanding egocentric activities, in: IEEE International Conference on Computer Vision, IEEE, 2011, pp. 407–414.
[6] A. Fathi, J.K. Hodgins, J.M. Rehg, Social interactions: a first-person perspective, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1226–1233.
[7] J. Rehg, G. Abowd, A. Rozga, M.R.M. Clements, L. Presti, S. Sclaroff, I. Essa, Decoding children's social behavior, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[8] F.I. Bashir, A.A. Khokhar, D. Schonfeld, S. Member, S. Member, Object trajectory-based activity classification and recognition using hidden markov models, in: IEEE Transactions on Image Processing, vol. 16, 2007.
[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: Transactions on Pattern Analysis and Machine Intelligence, vol. 29, 2007, pp. 2247–2253.
[10] Q. Qiu, Z. Jiang, R. Chellappa, Sparse dictionary-based representation and recognition of action attributes, in: IEEE International Conference on Computer Vision, IEEE, 2011, pp. 707–714.
[11] J. Liu, S. Ali, M. Shah, Recognizing human actions using multiple features, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008.
[12] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 3169–3176.
[13] A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3D-gradients, in: Proceedings of the British Machine Vision Conference, 2008.
[14] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, in: International Conference on Multimedia, ACM, 2007, pp. 357–360.
[15] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: International Conference on Computer Vision, IEEE, 2009, pp. 104–111.
[16] G.W. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, in: European Conference on Computer Vision, Springer, 2010, pp. 140–153.

[17] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing, vol. 28, Elsevier, 2010, pp. 976–990.
[18] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al., Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference 16 (2007).
[19] J. Aggarwal, M.S. Ryoo, Human activity analysis: a review, ACM Computing Surveys, vol. 43, ACM, 2011, p. 16.
[20] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1948–1955.
[21] B. Chakraborty, M.B. Holte, T.B. Moeslund, J. Gonzàlez, Selective spatio-temporal interest points, Computer Vision and Image Understanding, vol. 116, Elsevier, 2012, pp. 396–410.
[22] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, 2005, pp. 65–72.
[23] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, Proceedings of the 17th International Conference on Pattern Recognition, vol. 3, IEEE, 2004, pp. 32–36.
[24] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, 2009, pp. 1996–2003.
[25] K. Guo, P. Ishwar, J. Konrad, Action recognition using sparse representation on covariance manifolds of optical flow, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2010, pp. 188–195.
[26] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by covariance matching of silhouette tunnels, in: Brazilian Symposium on Computer Graphics and Image Processing, IEEE, 2009.
[27] A. Castrodad, G. Sapiro, Sparse modeling of human actions from motion imagery, International Journal of Computer Vision, vol. 100, Springer, 2012, pp. 1–15.
[28] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: European Conference on Computer Vision, Springer, 2008, pp. 650–663.
[29] J. Gall, A. Yao, N. Razavi, L. Van Gool, V. Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, IEEE, 2011, pp. 2188–2202.
[30] Q. Le, W. Zou, S. Yeung, A. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 3361–3368.
[31] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3501–3508.
[32] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC 2009 British Machine Vision Conference, 2009.
[33] L. Shao, R. Mattivi, Feature detector and descriptor evaluation in human action recognition, in: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, 2010, pp. 477–484.
[34] T. de Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, D. Windridge, An evaluation of bags-of-words and spatio-temporal shapes for action recognition, in: IEEE Workshop on Applications of Computer Vision, Springer, 2011, pp. IEEE–351.
[35] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, IEEE, 2011, pp. 883–897.
[36] S. Wu, O. Oreifej, M. Shah, Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories, in: IEEE International Conference on Computer Vision, 2011, pp. 1419–1426.
[37] M. Raptis, Discovering discriminative action parts from mid-level video representations, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1242–1249.
[38] T. Kadir, M. Brady, Saliency, scale and image description, International Journal of Computer Vision, vol. 45, Springer, 2001, pp. 83–105.
[39] Y. Li, Y. Zhou, J. Yan, Z. Niu, J. Yang, Visual saliency based on conditional entropy, in: Asian Conference on Computer Vision, Springer, 2010, pp. 246–257.
[40] N. Bruce, J. Tsotsos, Saliency based on information maximization, in: Advances in Neural Information Processing Systems, 2005, pp. 155–162.
[41] X. Hou, L. Zhang, Dynamic visual attention: searching for coding length increments, in: Advances in Neural Information Processing Systems, 2008, pp. 681–688.
[42] N. Vereshchagin, P. Vitányi, Kolmogorov's structure functions with an application to the foundations of model selection, in: Foundations of Computer Science, IEEE, 2002, pp. 751–760.
[43] V. Nannen, A Short Introduction to Model Selection, Kolmogorov Complexity and Minimum Description Length (MDL), arXiv preprint arXiv:1005.2364.
[44] J. Rissanen, Modeling by shortest data description, Automatica, vol. 14, Elsevier, 1978, pp. 465–471.
[45] J. Rissanen, MDL denoising, IEEE Transactions on Information Theory, vol. 46, IEEE, 2000, pp. 2537–2543.
[46] I. Ramírez, G. Sapiro, An MDL framework for sparse coding and dictionary learning, IEEE Transactions on Signal Processing, vol. 60, IEEE, 2012, pp. 2913–2927.
[47] N. Saito, Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion, Wavelets in Geophysics, vol. 4, Academic, New York, 1994, pp. 299–324.
[48] D.L. Donoho, Compressed sensing, IEEE Transactions on Information Theory, vol. 52, IEEE, 2006, pp. 1289–1306.
[49] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing, vol. 15, IEEE, 2006, pp. 3736–3745.
[50] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415, http://dx.doi.org/10.1109/78.258082.
[51] R. Rubinstein, M. Zibulevsky, M. Elad, Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit, in: CS Technion, 2008.
[52] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: International Conference on Machine Learning, ACM, 2009, pp. 689–696.
[53] I. Sipiran, B. Bustos, Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes, in: The Visual Computer, Springer, 2011, pp. 1–14.
[54] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[55] O. Tuzel, F. Porikli, P. Meer, Region covariance: a fast descriptor for detection and classification, in: European Conference on Computer Vision, Springer, 2006, pp. 589–600.
[56] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-euclidean metrics for fast and simple calculus on diffusion tensors, Magnetic Resonance in Medicine, vol. 56, Wiley Online Library, 2006, pp. 411–421.
[57] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 2169–2178.
[58] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, in: IEEE Conference on Computer Vision and Pattern Recognition Workshop, IEEE, 2006, p. 13.
[59] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, in: ACM Transactions on Intelligent Systems and Technology, vol. 2, 2011, pp. 27:1–27:27. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.
[60] M.D. Rodriguez, J. Ahmed, M. Shah, Action MACH: a spatio-temporal maximum average correlation height filter for action recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008.
[61] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[62] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, in: Machine Vision and Applications, Springer, 2012, pp. 1–11.
[63] N. Larios, H. Deng, W. Zhang, M. Sarpola, J. Yuen, R. Paasch, A. Moldenke, D. Lytle, S. Correa, E. Mortensen, et al., Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects, Machine Vision and Applications, vol. 19, Springer, 2008, pp. 105–123.
[64] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2556–2563, http://dx.doi.org/10.1109/ICCV.2011.6126543.
[65] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense Trajectories and Motion Boundary Descriptors for Action Recognition, Research Report RR-8050, INRIA (August 2012). <http://hal.inria.fr/hal-00725627>.

Guruprasad Somasundaram received his Bachelors in Electronics & Communication Engineering from Anna University, India in 2005. He received his M.S in Electrical Engineering in 2008, M.S. in Computer Science in 2010, and his PhD in Computer Science in 2012 all from the University of Minnesota. His research interests lie in the domains of Computer Vision, Pattern Recognition, Machine Learning and Robotics. His focus is in the area of object detection and classification. His current and past projects include detecting image saliences, vehicle and pedestrian classification and counting, medical image processing and activity recognition.

Anoop Cherian received his B.Tech (honours) degree in computer science and engineering from the National Institute of Technology, Calicut, India in 2002, M.S. in computer science in 2010 and PhD in computer science in 2012 from the University of Minnesota. He is currently a postdoctoral researcher in the LEAR project group at INRIA Rhone-Alpes, France. From 20022007, he worked as a software design engineer at Microsoft. His research interests include machine learning, and computer vision. He is the recipient of the Best Student Paper award at the Intl. Conf. on Image Processing (ICIP) in 2012.

Vassilios Morellas received his Diploma of Engineering in Mechanical Engineering, from the National Technical University of Athens in 1983. He received his M.S. in Mechanical Engineering from Columbia University in 1988, and PhD in Mechanical Engineering from the University of Minnesota in 1995. Vassilios Morellas' research interests are in the area of geometric image processing, machine learning, robotics and sensor integration to enhance automation of electromechanical systems. He is the Program Director in the department of Computer Science and Engineering and Executive Director of the NSF Center for Safety Security and Rescue. Prior to his current position he was a Senior Principal Research Scientist at Honeywell Laboratories where he developed technologies in the general areas of access control, security and surveillance and biometrics with emphasis on the problem of tracking of people and vehicles across non overlapping cameras. Past research experience also includes work on Intelligent Transportation Systems where he developed innovative technologies to reduce run-off-the-road accidents.

Nikolaos Papanikolopoulos received his Diploma of Engineering in Electrical and Computer Engineering, from the National Technical University of Athens in 1987. He received his M.S. in 1988 and PhD in 1992 in Electrical and Computer Engineering from Carnegie Mellon University. Professor Papanikolopoulos specializes in robotics, computer vision and sensors for transportation uses. His research interests include robotics, sensors for transportation applications, computer vision, and control systems. As the director of the Center for Distributed Robotics and a faculty member of the Artificial Intelligence and Robotic Vision Laboratory, his transportation research has included projects involving vision-based sensing and classification of vehicles, and the recognition of human activity patterns in public areas and while driving.